End-to-End Security Solutions for WebSphere on z/OS

2005 ITSO zSeries Workshop Tour
Deploying and Managing WebSphere on z/OS
Foulques de Valence

© 2005 IBM Corporation

Notices

This information was developed for products and services offered in the U.S.A.

Note to U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to: IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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 and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

2 © 2005 IBM Corporation

Redbooks IBM

Trademarks

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

IBM eServer™ Redbooks (logo)™

ibm.com®	z/OS®	zSeries®	AIX®	ClearCase®	Cloudscape™		CICS®	CICSPlex®	DB2
Connect™	DB2®	DFS™	DRDA®	Informix®	IBM®	IMS™	MQSeries®	MVS™	
	Perform™	Rational®	RACF®	S/390®	SAA®	TME®	VTAM®	WebSphere®	

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

3 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Introduction

End-to-End Security Solutions for WebSphere on z/OS

4 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

What is security?

- IT security objectives specified in ISO Standard 7498-2:
  - **Identification**
    - This is the ability to assign an identity to the entity accessing the system.
    - "user ID", UID, or "principal" in the J2EE security model
  - **Authentication**
    - This is the process of validating the identity claimed by the accessing entity.
    - Authentication information generally called "credentials": accessor's name and password, "token" provided by a trusted party, such as a Kerberos ticket, an x.509 certificate, or LTPA token.
  - **Authorization**
    - This is the process of checking whether an asserted (already authenticated) identity has access to a requested resource.
  - **Integrity**
    - Integrity ensures that transmitted or stored information has not been altered in an unauthorized or accidental manner.
  - **Confidentiality**
    - This refers to the concept that an unauthorized party cannot obtain the meaning of the transferred or stored data.
  - **Auditing**
    - With auditing, you capture and record security-related events, so that they can be exposed and analyzed after the fact.
  - **Non-repudiation**
    - This is a legal term that demands legal evidence that a party performed some action, so that it cannot reasonably be denied.

5 © 2005 IBM Corporation ON DEMAND BUSINESS

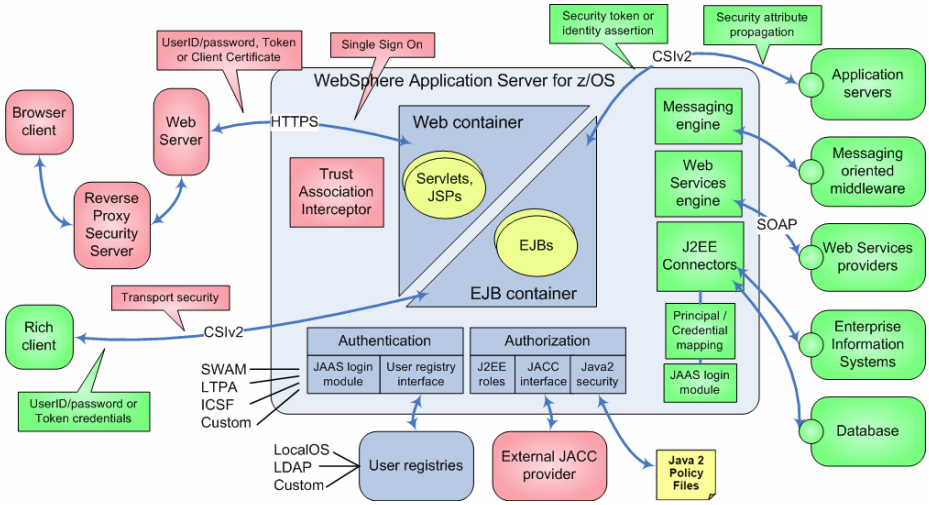
Redbooks IBM

Where Do You Start?

- A strategy for understanding WebSphere v6 security:
  - It's not like WebSphere 3.5 security.
  - It's a little like WebSphere V4.0.1 security.
  - It's about the same as WebSphere V5 security.
  - Read the security chapter (at least) of the Java 2 Enterprise Edition (J2EE) specs.
    - J2EE 1.4
    - Java Servlet 2.4
    - Enterprise JavaBeans (EJB) 2.1
    - http://java.sun.com/j2ee/1.4/docs/#general
  - Identify security functionality specified in J2EE specs.
  - Explore the WebSphere V6 manuals, expecting to find that functionality implemented.
- **WebSphere v6 security builds on the v5 foundation.**
- Incremental enhancements to Java standards
- Relatively few conversion/migration security issues.
- If you know WebSphere v5, you'll like v6.
  - If you don't know v5, there's a lot to learn.
  - We'll cover the most important aspects in class.

6 © 2005 IBM Corporation ON DEMAND BUSINESS

WebSphere z/OS v6 Security - Open Architecture



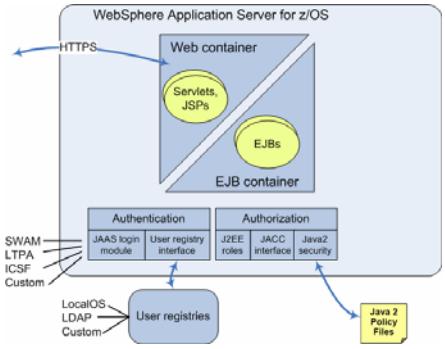
© 2005 IBM Corporation

Part 1: WebSphere for z/OS v6 security features overview

1.1. **Authentication**

1.2. **User registries**

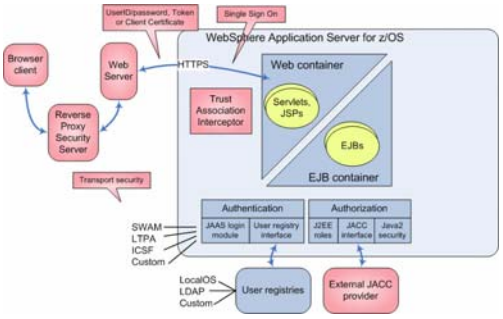
1.3. **Authorization**



© 2005 IBM Corporation

Part 2: WebSphere for z/OS front-end security solutions

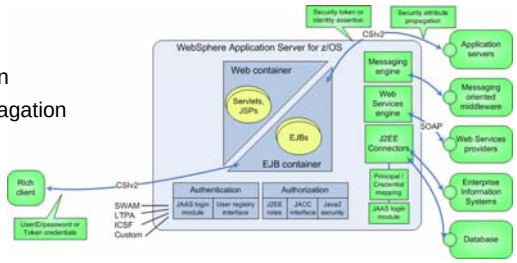
- **2.1. Web Application Security**
  - 2.1.1. Authentication
  - 2.1.2. Authorization
- **2.2. IBM HTTP Server on z/OS**
- **2.3. LDAP on z/OS**
- **2.4. JACC**
- **2.5. Tivoli Access Manager integration**
- **2.6. Single Sign On**
- **2.7. Transport Security**



9 © 2005 IBM Corporation ON DEMAND BUSINESS

Part 3: WebSphere for z/OS back-end security solutions

- **3.1. EJB Application Security**
  - 3.1.1. Authentication and CSIv2
  - 3.1.2. Authorization
- **3.2. Security attribute propagation**
  - 3.2.1. Horizontal attribute propagation
  - 3.2.2. CSIv2 standard Identity Assertion
  - 3.2.3. CSIv2 and vertical attribute propagation
  - 3.2.4. JAAS Login Modules
- **3.3. EIS Security**
  - 3.3.1. JCA Security
  - 3.3.2. Accessing CICS z/OS
  - 3.3.3. Accessing IMS z/OS
  - 3.3.4. Accessing DB2 z/OS
  - 3.3.5. TAM GSO Principal mapping
- **3.4. Web Services Security**



10 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

are welcome anytime

11 © 2005 IBM Corporation ON DEMAND BUSINESS

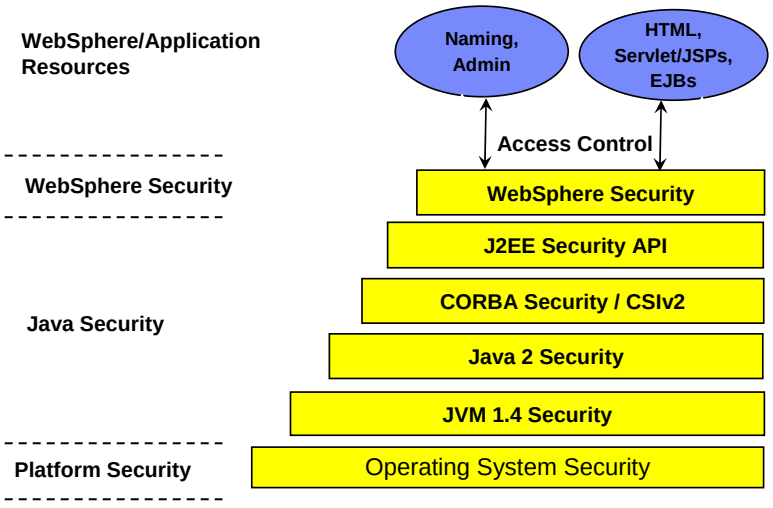
Redbooks IBM

Part 1

WebSphere for z/OS v6 security features overview

12 © 2005 IBM Corporation ON DEMAND BUSINESS

Security Layers



© 2005 IBM Corporation

J2EE 1.4 Security Features

- **Java 2 Security:** Access to System Resources
  - Enforce access control, based on the location of the code and who signed it – Not based on the principal
  - Defined in a set of Policy files
  - Enforced at runtime
- **JAAS Security:** Authentication and Authorization
  - Enforce access control based on the current Principle or Subject
  - Defined in Application Code
  - Enforced programmatically
  - Used for any type of Java code – Stand-alone Java application, Applet, EJB, Servlet, and so on
- **J2EE Security Roles:** Authorization of J2EE application artifacts
  - Role based security – Roles defined in the J2EE EAR file
  - Defined in application configuration settings (Deployment Descriptors)
  - Enforced by runtime, programmatically, or both
- **CSIv2:** Used for Authenticating EJBs, replacing IBM proprietary SAS and z/SAS protocols

© 2005 IBM Corporation

© 2005 IBM Corporation

Redbooks IBM

WebSphere v6 Security

- The security configuration and setting is cell wide in Network Deployment cell
  - DMgr, all Node Agents and all Servers have the same security configuration applied
    - Authentication mechanism, registry, etc.
  - Some security settings can be overridden on individual Application Servers
    - Turning off Application security
- Global Security must be enabled

15 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 1: WebSphere for z/OS v6 security features overview

1.1. Authentication

1.2. User registries

1.3. Authorization

16 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

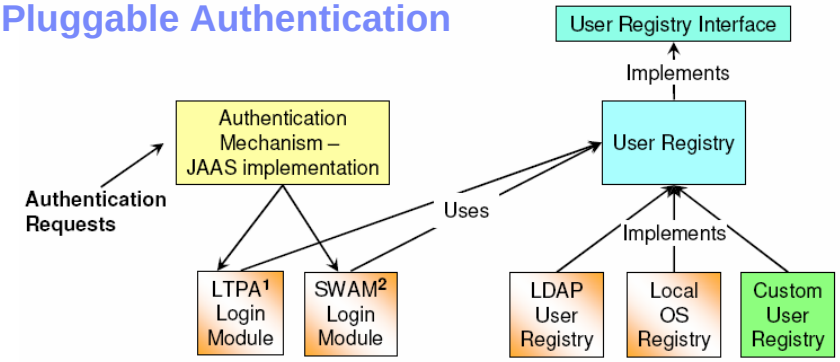
Authentication

- Authentication is the process of establishing whether a client is valid in a particular context
  - Client can be either an end user, a machine, or an application
- An *authentication mechanism* defines rules about security information and the format of how security information is stored in both credentials and tokens
  - Whether a credential is forwardable to another process
- Authentication Mechanism uses User (Authentication) Registry (where user ID/password, and other attributes are stored) to check the client authentication
  - WebSphere supports several User Registries - Local OS, LDAP and Custom Registry

17 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Pluggable Authentication

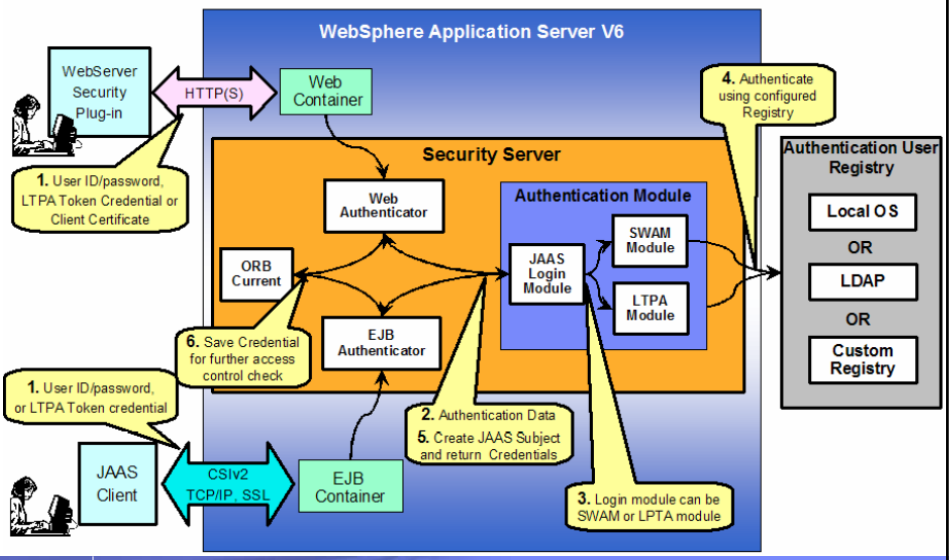


[1] LTPA = "Lightweight Third Party Authentication" mechanism
[2] SWAM = "Simple WebSphere Authentication Mechanism"

- Authentication requires Authentication Mechanism and an appropriate User Registry
  - Only one Authentication Mechanism and User Registry can be enabled at a time
  - However, Custom User Registry could check for users from multiple disparate registries for each registry query, if needed

18 © 2005 IBM Corporation ON DEMAND BUSINESS

Authentication Mechanism: Flow



© 2005 IBM Corporation

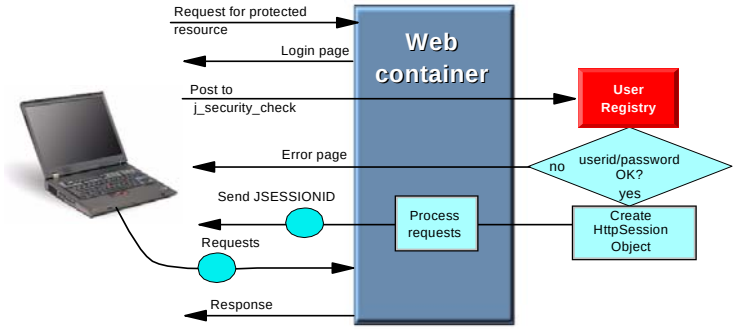
Supported Authentication Mechanisms

Authentication Mechanism	Intended Use and Supported Package
Simple WebSphere Authentication Mechanism (SWAM) Not available and not needed in WebSphere Application Server v6 Network Deployment and higher packages	▪For simple, non-distributed, single application server environments ▪Does not support *forwardable* credentials or Single Sign On (SSO) ▪Caller identity is not forwarded from client on one server to EJB on another server - What gets forwarded in unauthenticated credential which may fail on the receiving server
Lightweight Third Party Authentication (LTPA) mechanism Available on all platforms and packages	▪For distributed, multiple application server environments (WebSeal, Domino…) ▪Support *forwardable* credentials or Single Sign On (SSO) through cryptography ▪Requires all the servers authentication registry to be a centrally shared registry like LDAP
Integrated Cryptographic Service Facility (ICSF) Only on z/OS platforms	▪For distributed, multiple application server environments ▪Supports *forwardable* credentials or Single Sign On (SSO) ▪Supports all WebSphere supported Authentication Registry

© 2005 IBM Corporation

Simple WebSphere Authentication Mechanism

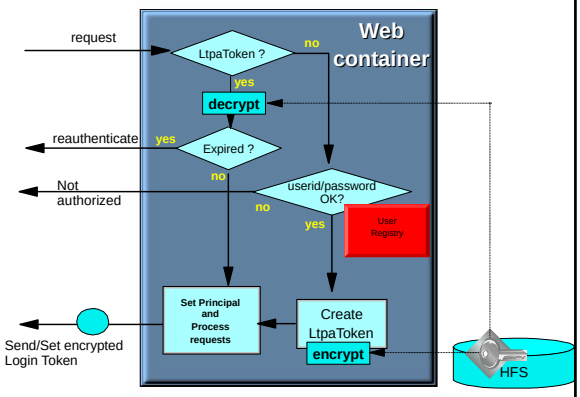
- Only supported for base server configuration. Because it does not support a forwardable token:
  - it does not support Single Sign-On
  - it keeps the form based authentication data in an HttpSession (not a LtpaToken cookie):



© 2005 IBM Corporation

Lightweight Third Party Authentication (LTPA)

- Intended for distributed, multiple application server and machine environments
- Supports forwardable credentials and SSO
- LTPA protocol uses cryptographic keys (LTPA keys) to encrypt and decrypt user data that passes between the servers
  - If Servers are in different cells, the LTPA keys need to be shared
    - Generate, Export and Import LTPA keys in the administrative console
  - All servers in the domain must be synchronized
- LTPA generates a security token for authenticated users, which can be used to represent that authenticated user on subsequent calls to the same or other servers within a Single Sign On (SSO) domain.

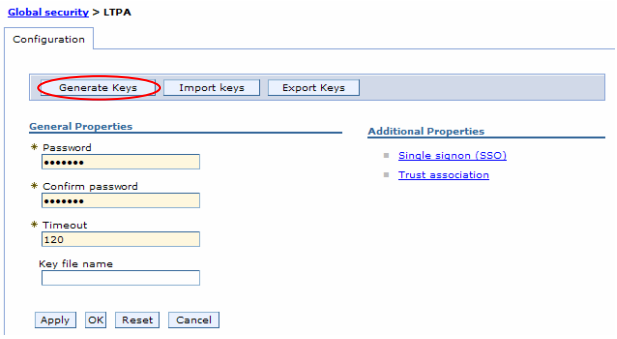


© 2005 IBM Corporation

Redbooks IBM

LTPA Authentication Mechanism – Keys Mngt

- Once a new set of keys is generated and saved, the key propagation is dynamic.
- All the processes running at that time (cells, node agents, application servers) are updated with the new set of keys.



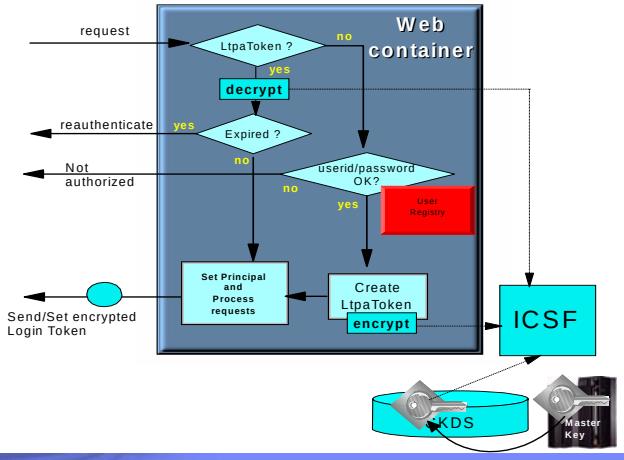
23 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

ICSF Authentication Mechanism

- ICSF can be configured as an alternative to LTPA on the z/OS platform to generate a security token for authenticated users.

IBM intends to deprecate the ICSF authentication mechanism. It is recommended that you migrate to LTPA.



24 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 1: WebSphere for z/OS v6 security features overview

1.1. Authentication

1.2. User registries

1.3. Authorization

25 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

User Registries

- WebSphere Application Server V6 Supports the following user registries for all Authentication Mechanisms:
  - **LocalOS** (Operating System)
  - **LDAP** (Lightweight Directory Access Protocol)
  - **Custom Registry** - ability for you to plug-in your own registry
- In a Network Deployment cell, only one user registry can be active at any given time
  - While only one configured registry can be active, a Custom user registry can be developed to access multiple registries.

26 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

User Registry: Local Operating System (OS) – z/OS

- Uses Local OS authentication registry users and groups
  - RACF, TopSecret or ACF/2
- Security database can be shared across the sysplex so works well in a multi-server 'distributed' environment.
  - 'distributed' here means across different LPARs
- Compatible with z/OS Enterprise Information Systems (CICS, IMS, DB2)
  - Able to flow original identity to backend EIS

27 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

User Registry: LDAP

- LDAP servers act as a repository for user and group information
- WebSphere Application Server calls the LDAP server to get the user and group information
  - This support is provided by using different user and group filters
- LDAP server configuration requires you to specify:
  - Valid Server user name (ID), the user password, the server host and port, the base distinguished name (DN)
  - If LDAP server does not support anonymous binds, then specify the bind DN and the bind password
- Incompatible with z/OS Enterprise Information Systems (CICS, IMS, DB2)
  - Additional steps required to access these systems with connection identity
  - Unable to flow original identity to EIS and must be mapped to a local OS identity

28 © 2005 IBM Corporation ON DEMAND BUSINESS

User Registry: Custom Registry

- Allows you to plug in your own Registry whose support is not implemented by WebSphere Application Server Security
- Written as a Java program that implements WebSphere Application Server supplied **com.ibm.websphere.security.UserRegistry** interface
  - The implementation should not be dependent on WebSphere Application Server resources (for example, datasource and so on)
- To configure Custom Registry, you need to provide the following:
  - Full class name of the Custom Registry implementation
  - Valid Server user id and password
  - Any custom properties required by the implementation

29 © 2005 IBM Corporation

Part 1: WebSphere for z/OS v6 security features overview

1.1. Authentication

1.2. User registries

1.3. Authorization

30 © 2005 IBM Corporation

Redbooks IBM

Authorization

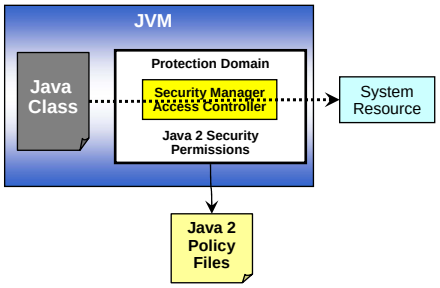
- Authorization is the process that verifies a client has the appropriate privileges to perform an operation
  - Information can be stored many ways
    - Access-control list, capability lists
- J2EE uses role based authorization
  - During assembly, permissions to call methods are given to various roles
  - Roles define a set of permissions within an application
  - During deployment users and groups are assigned to these roles

31 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Java 2 Security

- Provides an access control mechanism to manage the application's access to system level resources
  - File I/O, Network Connections (Sockets), Property files, etc…
  - Policy-based
- Policies define a set of permissions available from various signers and/or code locations
  - Stored in Policy files
- All Java code runs under a security policy
  - Grants access to certain resources



- Java code needs access to certain System Resources
- Java code will need to get the permission from Java 2 Access Control
- Access Control looks at the Java 2 Policy file(s) to determine if the requesting Java code has the appropriate permission

32 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Example: resources protected by Java 2 Security

- File Access
  - canRead(), FileInputStream(), RandomAccessFile(), isDirectory(), isFile(), length(),
  - canWrite(), FileOutputStream(), mkdir(), renameTo(), createTempFile()
  - delete(), deleteOnExit()
- Network Access
  - send(), receive(), getLocalAddress(), getHostName(), getLocalHost(), getAllByName()
- Java VM
  - ClassLoader(), loadLibrary(), checkPermission(), checkLink(), checkExit()
- Program Threads
  - stop(), resume(), suspend(), interrupt(), setPriority(), setName(), setDaemon()
- System Resources
  - getPrintJob(), setProperty(), getProperty(), setDefault(), getFont(), getEventQueue()
- Security Aspects
  - getFields(), getMethods(), getConstructors(), setPublicKey(), addCertificate()

33 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Java 2 Security policy files in WebSphere

STATIC Policies

DYNAMIC Policies

Policy File	Default Location	Description
java.policy	<PROFILE_HOME>/java/jre/lib/security/java.policy	Default permissions granted to all classes
server.policy	<PROFILE_HOME>/properties/server.policy	Default permissions granted to all the product servers.
client.policy	<PROFILE_HOME>/properties/client.policy	Default permissions for all of the product client containers and applets on a node
filter.policy	<PROFILE_HOME>/ config/cells/cell_name	Filters OUT policy that are specified in other policy files – this allows System administrator to provide protection globally, even if other policy files may give the permission
spi.policy	<PROFILE_HOME>/ config/cells/cell_name /nodes/node_name/spi.policy	For the Service Provider Interface (SPI) or 3rd party resources embedded in the product.
library.policy	<PROFILE_HOME>/ config/cells/cell_name /nodes/node_name/library.policy	For the shared libraries (Java library classes) used by applications. Default is empty.
app.policy	<PROFILE_HOME>/ config/cells/cell_name	Default permissions granted to all J2EE applications running on a specific node
ra.xml	rar_file_name/META-INF/was.policy.RAR	Default permission for the specific Resource Adapter, embedded in the RAR file
was.policy	Within each application – policies provided by developer	Describes the policy for that application – if none provided, a default one will be created

34 © 2005 IBM Corporation ON DEMAND BUSINESS

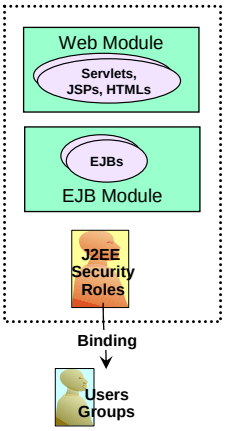
Enabling Java 2 Security

- Java 2 Security enabled when Global Security is turned on
  - Java 2 Security can be manually disabled, while keeping Global Security on
  - Test existing applications before turning on Java 2 Security
- Adding Java 2 Security to installed applications
  - For individual applications, the policy file, "**was.policy**", is placed in the META-INF folder of the Enterprise Application
  - Default one created during install process, if one is not already present
- Adding/modifying was.policy
  - Add was.policy file to EAR using AST or Rational® tools and reinstall
    - Recommended and less error prone
  - Place the was.policy file in the appropriate location in the file system where the application was installed
    - <WAS_PROFILE_DIR>\config\application\<application>.ear\META-INF\
- Tool for creating/modify policy files: **policyTool** – Part of JDK, in java/jre/bin/policytool
  - Advised to use the tool rather than editing by hand

35 © 2005 IBM Corporation ON DEMAND BUSINESS

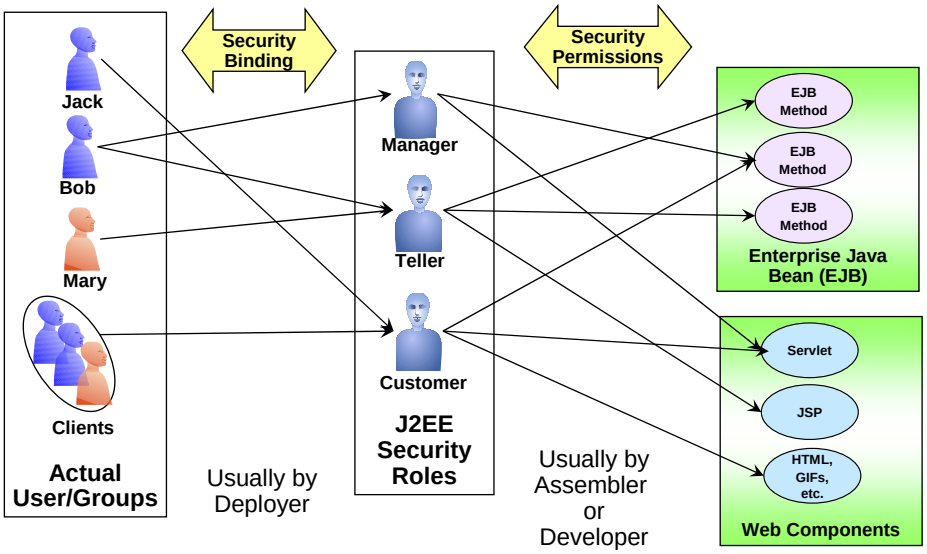
J2EE Security Roles: Application Authorization

- Authorization is performed using J2EE Security Roles
  - Allows developer to specify security at an abstract level
- Security roles are applied to the Web and EJB application components
  - EJB methods or Web URIs
- Security can be specified in the following ways:
  - Declaratively at assembly time, through the deployment descriptors
  - Programmatically using standard APIs at development time
- Binding users and groups to J2EE security roles is usually done at application deploy (install) time
  - On z/OS, EJBROLE Profiles need to be added for the required roles and users given access to these profiles when SAF Authorization is used



36 © 2005 IBM Corporation ON DEMAND BUSINESS

Securing J2EE Application Artifacts



37 © 2005 IBM Corporation

Roles: WebSphere bindings and SAF bindings

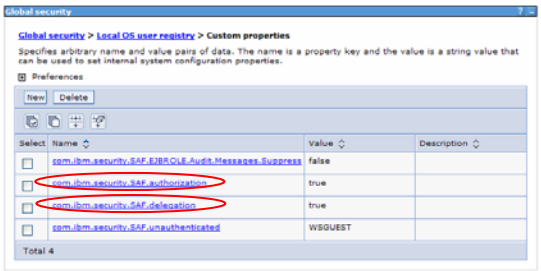
- **WebSphere bindings** (com.ibm.security.SAF.authorization=false)
  - Access to Servlets or EJB methods is based upon the 'role' (job title, function, etc.) of the user or caller.
  - Roles are associated with Servlets or EJBs at assembly time.
  - Roles are stored in the Application's .ear file: application.xml
  - Which users and groups have which roles is also stored in the Application's .ear file: ibm-application-bnd.xmi
  - Roles are managed by the application developer and the application deployer.
  - RACF only provides user and group information.
- **RACF Role Based Authorization** (com.ibm.security.SAF.authorization=true)
  - Access to Servlets or EJB methods is based upon the 'role' (job title, function, etc.) of the user or caller.
  - Roles are associated with Servlets or EJBs at assembly time.
  - Roles are represented in the Application's .ear file: application.xml
  - Which users and groups have which Roles is determined in RACF by profiles in the EJBROLE class.
  - If a user is in the access list of an EJBROLE profile, he has that role.
  - If a group is in the access list of an EJBROLE profile, users in that group have that role.
  - Roles are managed through RACF.

38 © 2005 IBM Corporation

Role Mapping using SAF bindings

- Method authorization is done using RACF EJBROLE profiles.
  - com.ibm.security.SAF.authorization=true
- To enable using the APPLDATA segment of the RACF EJBROLE profile for the identity to be used for the RunAs role
  - com.ibm.security.SAF.delegation=true

- Defines ejbCaller as an EJB role in domain DOMAIN1
  - RDEFINE EJBROLE DOMAIN1.ejbCaller APPLDATA('USER1')
- Permits access to the EJB role to members of group TRDREJB
  - PERMIT DOMAIN1.ejbCaller CLASS(EJBROLE) ID(USER1) ACCESS(READ)



© 2005 IBM Corporation

Java Authentication and Authorization Service (JAAS)

- Set of Java 2 Security APIs used to establish identity and perform authorization:
  - Authentication determines who is currently executing the code, regardless of where the code is running
  - Authorization of the users, based on JAAS security policies to specify what access rights are granted to executing code
- JAAS authentication is based on Pluggable Authentication Module
  - Allows applications to remain independent from underlying authentication technologies
- User authentication is represented by a Subject

© 2005 IBM Corporation

© 2005 IBM Corporation

Redbooks IBM

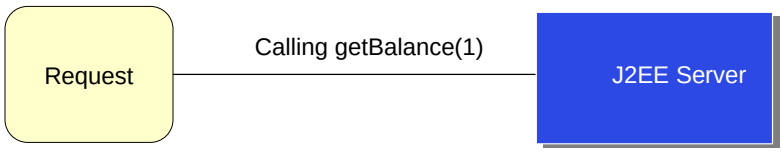
JAAS Subject

- Information about the currently authenticated user
- Basically a container of Java Sets
  - Principals
  - Credentials
- Data is generally read-only – only login modules can modify Subject
- Subject is populated by login modules as part of WAS authentication
  - WSPrincipal – basically a Java Principal
  - WSCredential – object with various security attributes about user – groups, userid, realm, etc.
  - Optional custom data from your login modules

41 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

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

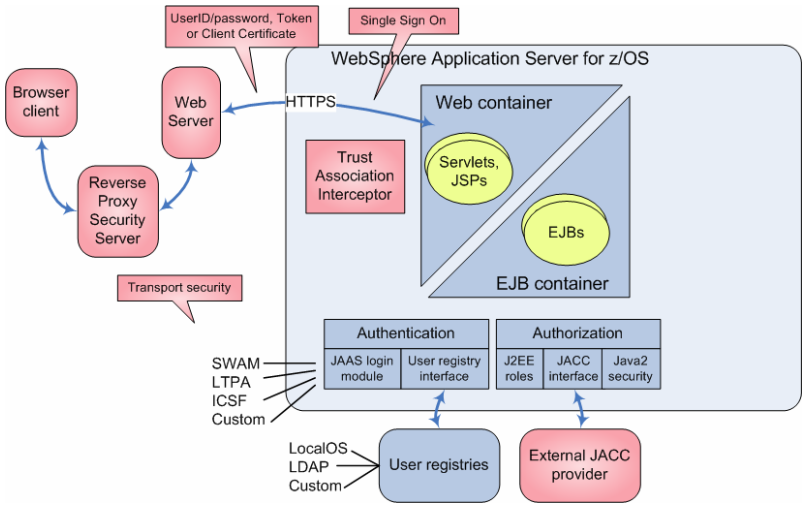
42 © 2005 IBM Corporation ON DEMAND BUSINESS

Part 2

WebSphere for z/OS front-end security solutions

© 2005 IBM Corporation

Part 2: WebSphere for z/OS front-end security solutions



© 2005 IBM Corporation

Redbooks IBM

Part 2: WebSphere for z/OS front-end security solutions

2.1. Web Application Security

2.1.1. Authentication

2.1.2. Authorization

2.2. IBM HTTP Server on z/OS

2.3. LDAP on z/OS

2.4 JACC

2.5. Tivoli Access Manager integration

2.6. Single Sign On

2.7. Transport Security

45 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Configuring Web Components Security

- **Authentication method** : specify how to obtain authentication information for the Web module
  - Basic authentication,
  - Client certificate authentication
  - Form-based authentication
- **Data constraints**: allows you to specify the required transport guarantee that defines the communication between the client and the Web application
  - None – no transport guarantee requires
  - Integral – ensures data cannot be changed in transit – SSL used
  - Confidential – ensures data cannot be viewed in transit – SSL used
- **Web resource collection** to be protected
  - Web resources is a set of URL patterns and HTTP methods
  - For static resources (HTMLs), valid HTTP methods are GET and POST
  - For dynamic resources (Servlet or JSP), valid HTTP methods are GET, POST, PUT, DELETE, HEAD, OPTION, TRACE
  - Assembler authorize different J2EE Security roles to access Web resources

46 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 2: WebSphere for z/OS front-end security solutions

47 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Types of Authentication for Web Applications

- **Basic**
  - Application server sends back a 501 challenge to the Web client (browser) allowing the client to pop up user ID, password dialog to the client
- **Form based**
  - Allows Web developer to provide a custom form login for the authentication challenge
- **Client certificate**
  - The client certificate is sent to the Application server using SSL secured connection

48 © 2005 IBM Corporation ON DEMAND BUSINESS

Authentication solution 1: **Basic Authentication**

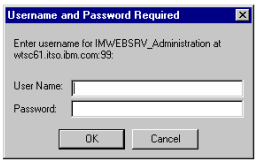
1. User clicks on link to protected page

Request: GET http://server/restricted.html

2. Server checks authority and rejects request

Response: Status 401
Realm "IMWEBSRV_Administration"

3. Browser pop-up window prompts user for userId and password

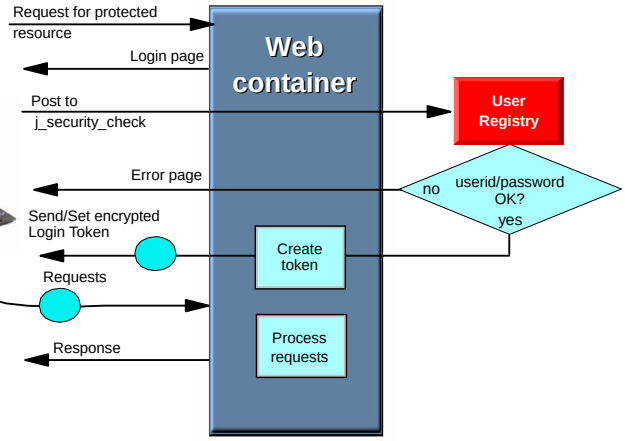


4. Browser resends request with userid and password in request header

Request: GET http://server/restricted.html

49 © 2005 IBM Corporation ON DEMAND BUSINESS

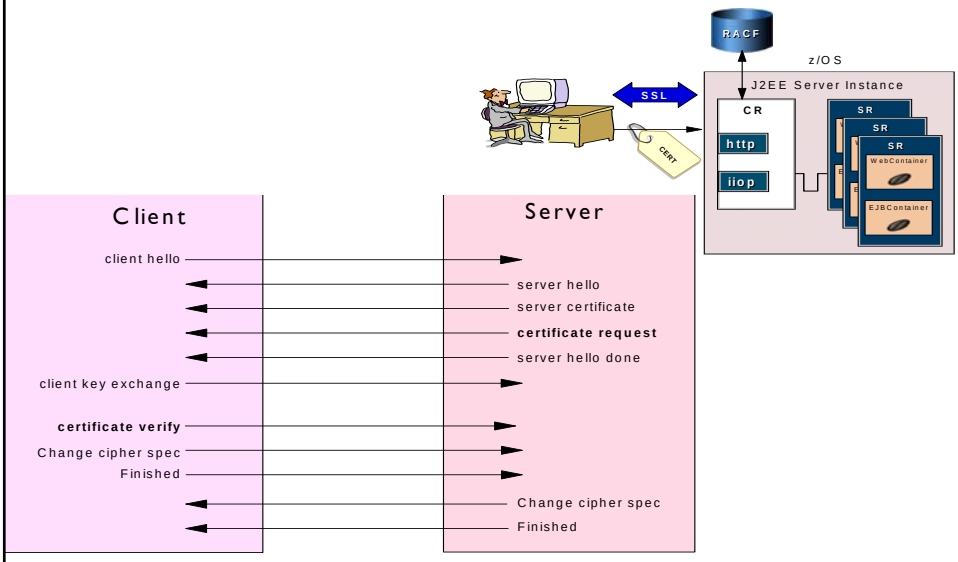
Authentication solution 2: **Form-based login**



The Login Token is typically a LtpaToken cookie but not necessarily.

50 © 2005 IBM Corporation ON DEMAND BUSINESS

Authentication solution 3: **Certificate-based Authentication**

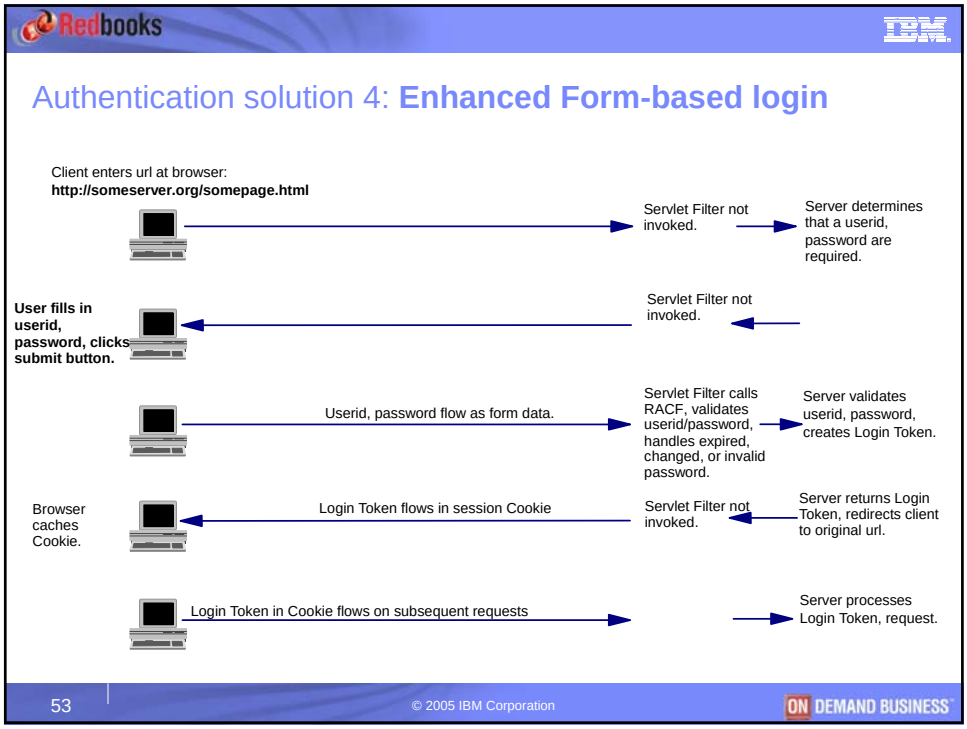


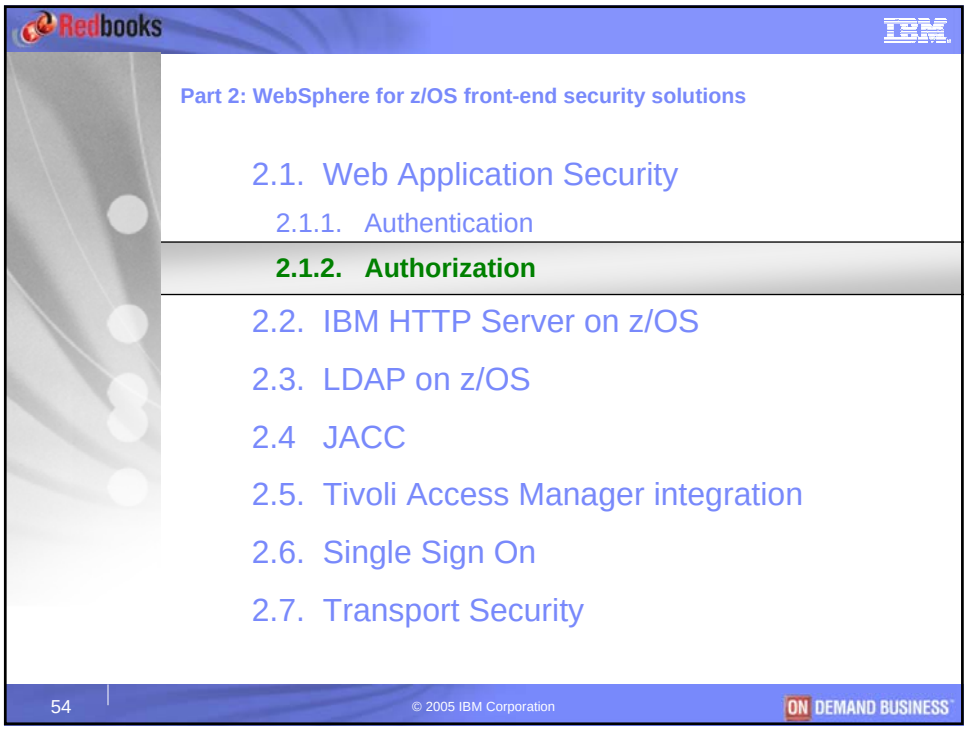
© 2005 IBM Corporation

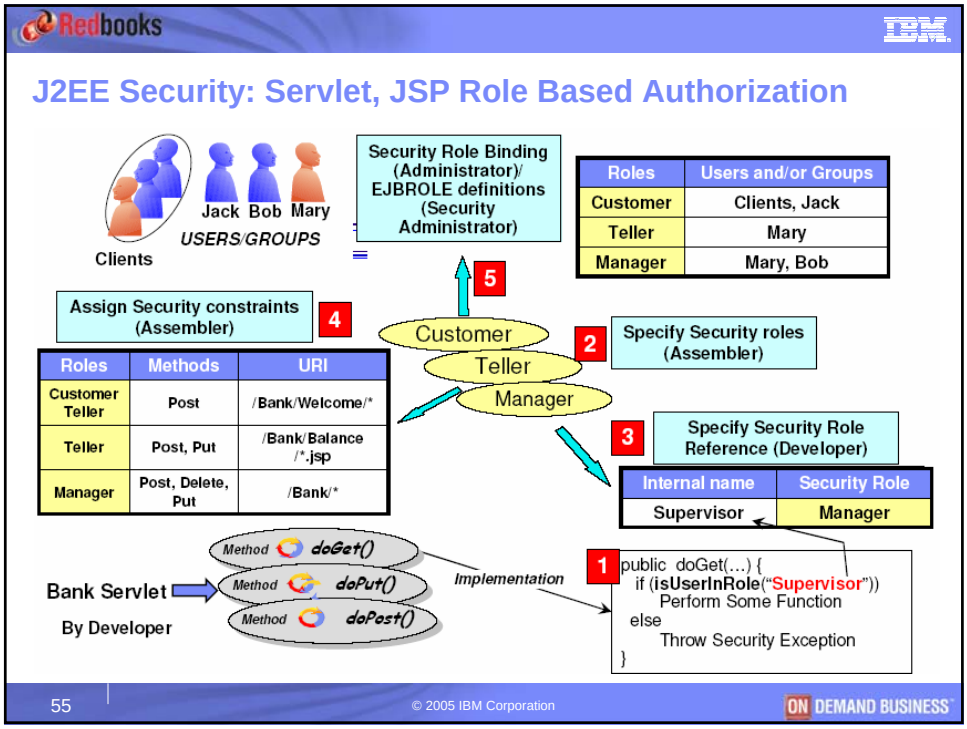
Form-based login limitation and solution

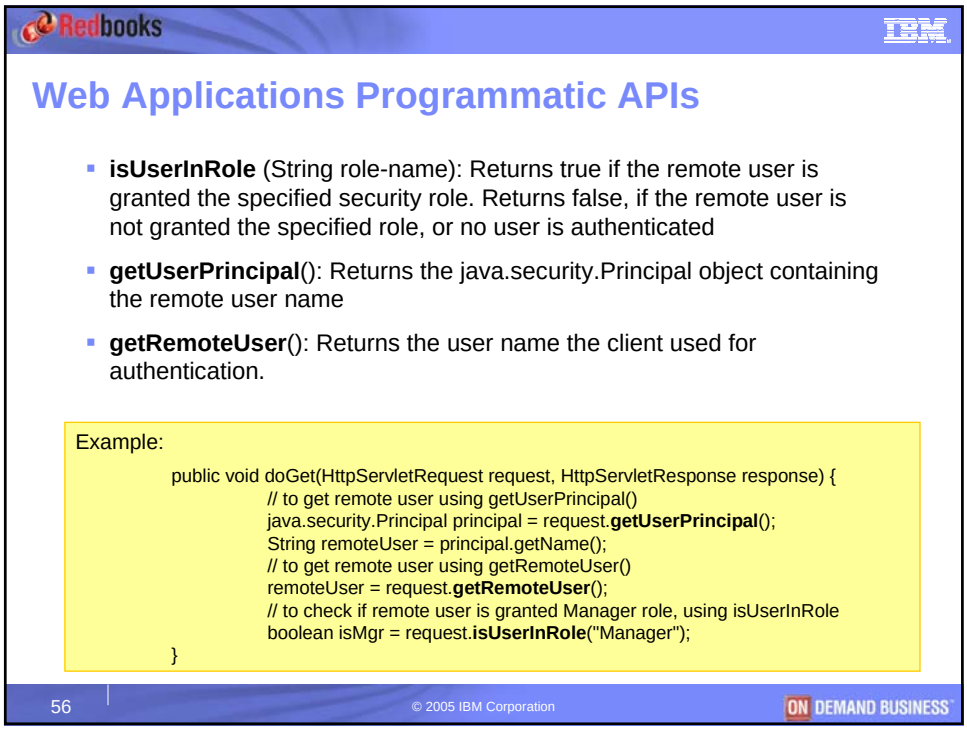
- zSeries customers have identified several limitations in Form Based Authentication:
  - It doesn't handle expired passwords (Not mentioned in the spec).
  - The 'error page' is static.
  - Doesn't provide enough status info.
- **Enhanced Form Based Authentication** was developed to solve Form Based Authentication problems:
  - It handles expired passwords.
  - It handles user-initiated password changes.
  - It provides information on authentication failures.
- Enhanced Form Based Authentication uses a Servlet Filter and is an adjunct to Form Based Authentication.
  - Enhanced Form Based Auth requires Form Based Auth.
- A **Servlet Filter** is Java code that is executed by the Web Container before and/or after a servlet.
- Techdoc TD101255:
  - http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101255

© 2005 IBM Corporation


Redbooks
IBM
Authentication solution 4: Enhanced Form-based login
Client enters url at browser:
http://someserver.org/somepage.html
Servlet Filter not invoked.
Server determines that a userid, password are required.
User fills in userid, password, clicks submit button.
Servlet Filter not invoked.
Userid, password flow as form data.
Servlet Filter calls RACF, validates userid/password, handles expired, changed, or invalid password.
Server validates userid, password, creates Login Token.
Browser caches Cookie.
Login Token flows in session Cookie
Servlet Filter not invoked.
Server returns Login Token, redirects client to original url.
Login Token in Cookie flows on subsequent requests
Server processes Login Token, request.
53
© 2005 IBM Corporation
ON DEMAND BUSINESS


Redbooks
IBM
Part 2: WebSphere for z/OS front-end security solutions
2.1. Web Application Security
2.1.1. Authentication
2.1.2. Authorization
2.2. IBM HTTP Server on z/OS
2.3. LDAP on z/OS
2.4 JACC
2.5. Tivoli Access Manager integration
2.6. Single Sign On
2.7. Transport Security
54
© 2005 IBM Corporation
ON DEMAND BUSINESS





Redbooks IBM

Web Applications Programmatic APIs

- **isUserInRole** (String role-name): Returns true if the remote user is granted the specified security role. Returns false, if the remote user is not granted the specified role, or no user is authenticated
- **getUserPrincipal**(): Returns the java.security.Principal object containing the remote user name
- **getRemoteUser**(): Returns the user name the client used for authentication.

Example:

```
public void doGet(HttpServletRequest request, HttpServletResponse response) {
        // to get remote user using getUserPrincipal()
        java.security.Principal principal = request.getUserPrincipal();
        String remoteUser = principal.getName();
        // to get remote user using getRemoteUser()
        remoteUser = request.getRemoteUser();
        // to check if remote user is granted Manager role, using isUserInRole
        boolean isMgr = request.isUserInRole("Manager");
}
```

56 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Changing Identity: Run-As

- The Web application Servlet or JSP has ability to change identity when calling downstream processes or EJBs
  - This is called "Run-As" identity
- The following are the 2 "Run-As" options:

Run-As options	Description
Client Identity	▪Bean takes on the same identity as the caller
Another Specified Role	▪Bean takes on identity of a specified user within the specified role ▪The specified role is part of the deployment descriptor and performed by the assembler ▪The specific user in the "Run-As" role is usually specified at deploy time

- Run-As does not change the identity of the z/OS thread (TCB level). WebSphere manages RunAs identities internally (Java Principal)

57 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Application Security Tasks and Roles

Tasks	Role	Tools used	Files modified
Define J2EE Security Roles	Assembler	Rational Tools, AST	Application Deployment. Descriptor, application.xml
Security check using programmatic API	Developer	Rational Tools, AST	Java code
Specifying Security permission or constraints	Assembler	Rational Tools, AST	ejb-jar.xml web.xml
Specify Security Role Reference	Assembler	Rational Tools, AST	Module level IBM Binding files: ibm-ejb-jar-bnd.xmi Ibm-web-bnd.xmi
Specify Security Role binding to users, groups or both	Administrator	Application Server (production) or Rational tool (dev.), AST	Server security.xml file (production) or ibm-application-bnd.xml (for development) or JACC provider
Define EJBROLEs and grant users/groups access	z/OS Security Administrator	SAF based security product	SAF based database
Specifying Authentication type	Administrator	Application Server	Server security.xml file

58 © 2005 IBM Corporation ON DEMAND BUSINESS

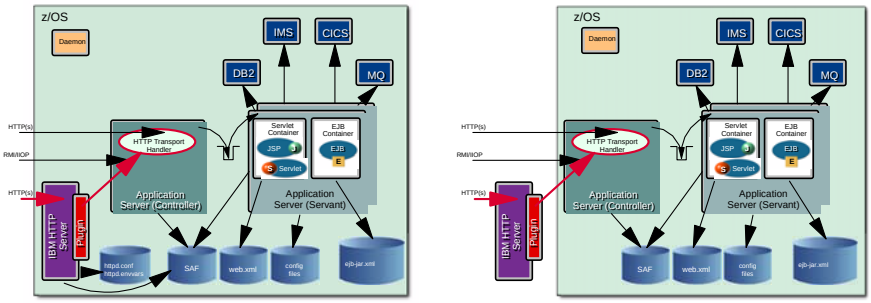
Redbooks IBM

Part 2: WebSphere for z/OS front-end security solutions

59 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

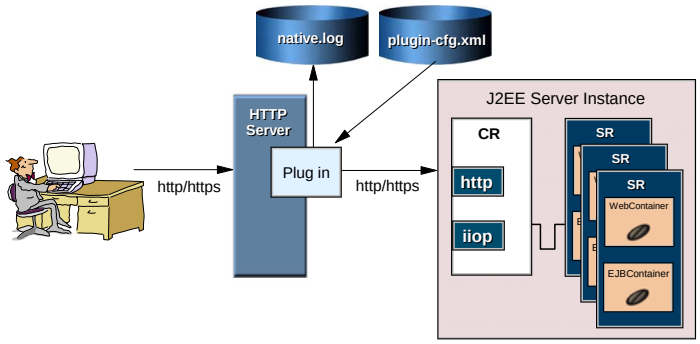
Add a Web server



- Web server may be on any platform
- If Web server authenticates, identity not automatically propagated
  - Basic: Userid/password on HTTP header will be authenticated twice
  - Form: Web server doesn't do form, its authentication ignored
  - Certificate: Client certificate forwarded in private header

60 © 2005 IBM Corporation ON DEMAND BUSINESS

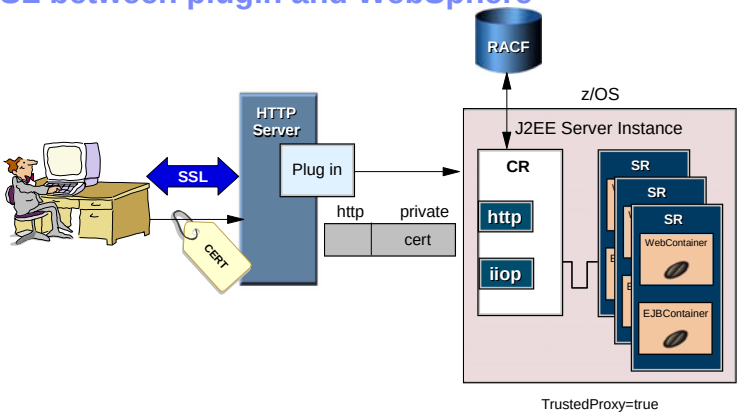
IBM HTTP Server and WebSphere plugin



- IBM HTTP Server is better to serve static content (Flexible, FRCA, local cache…)
- Plugin configuration is automatically updated by WebSphere (application location, protocols security, server weight...)
- WebSphere plugin handles session affinity

61 © 2005 IBM Corporation

IBM HTTP Server with SSL authentication
No SSL between plugin and WebSphere



- Client certificate is transferred in HTTP private headers between the plugin and WebSphere
- TrustedProxy=true property of the HTTP transport allows the Web Container to use http private headers.

62 © 2005 IBM Corporation

Redbooks IBM

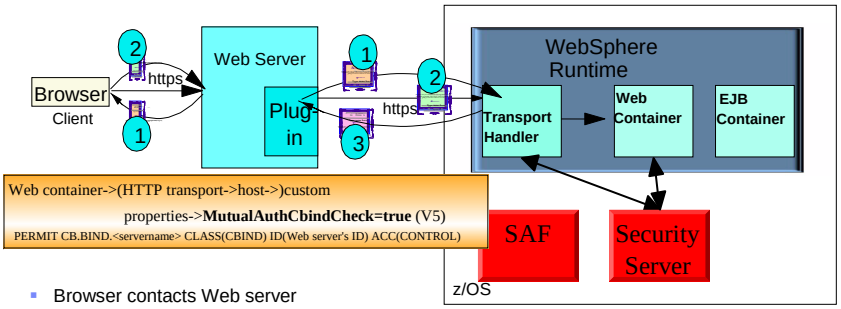
The private header story

- Private headers are a set of HTTP header that the plug-in adds to the HTTP request header.
- WebSphere removes this information from the header and then processes this information.
- Some key information carried from plug-in in private headers:
  - port number from URL (required for redirection and, for example, form based login)
  - client certificate (if provided)
- Configure Web container to accept private headers:
  - Web container->(HTTP transport->host->)custom properties->TrustedProxy=true
- You should limit connections to trusted servers

63 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

IBM HTTP Server with SSL authentication SSL between plugin and WebSphere



- Browser contacts Web server
  - SSL protocol responds with server certificate (1)
  - Browser responds with client certificate (2)
- Plug-in contacts transport handler
  - System SSL responds with WebSphere's certificate (3)
  - Plug-in responds with Web server's certificate (1) -> mapped to userid for CBIND check (works only with SAF)
  - Plug-in forwards HTTP request with client certificate (2) in private header -> mapped to userid for role-based authorization

64 © 2005 IBM Corporation ON DEMAND BUSINESS

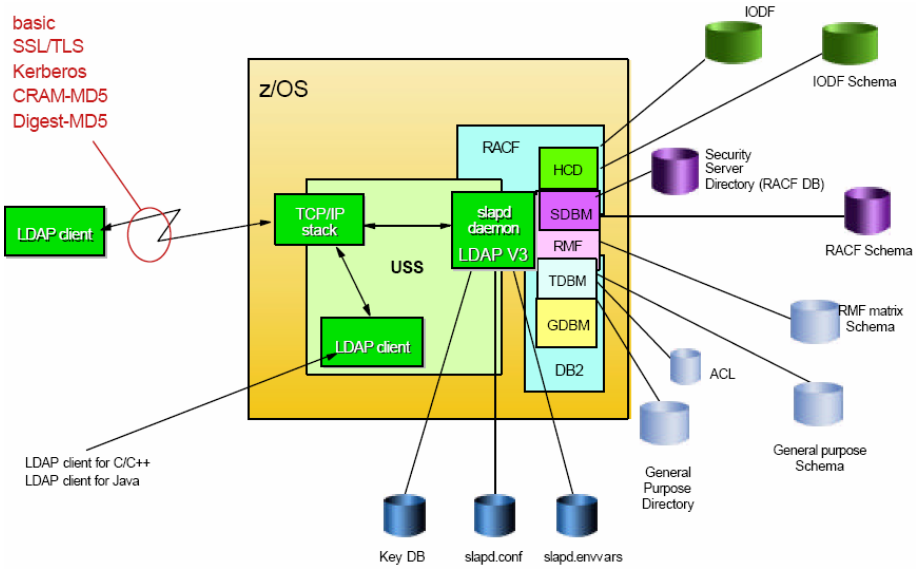
Redbooks IBM

Part 2: WebSphere for z/OS front-end security solutions

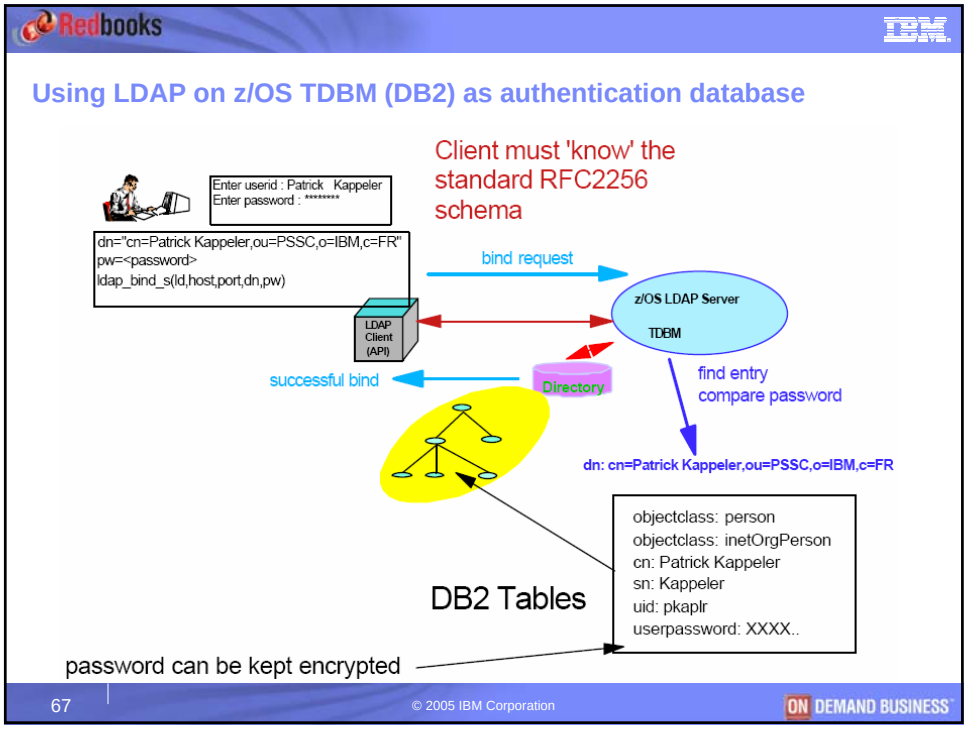
65 © 2005 IBM Corporation ON DEMAND BUSINESS

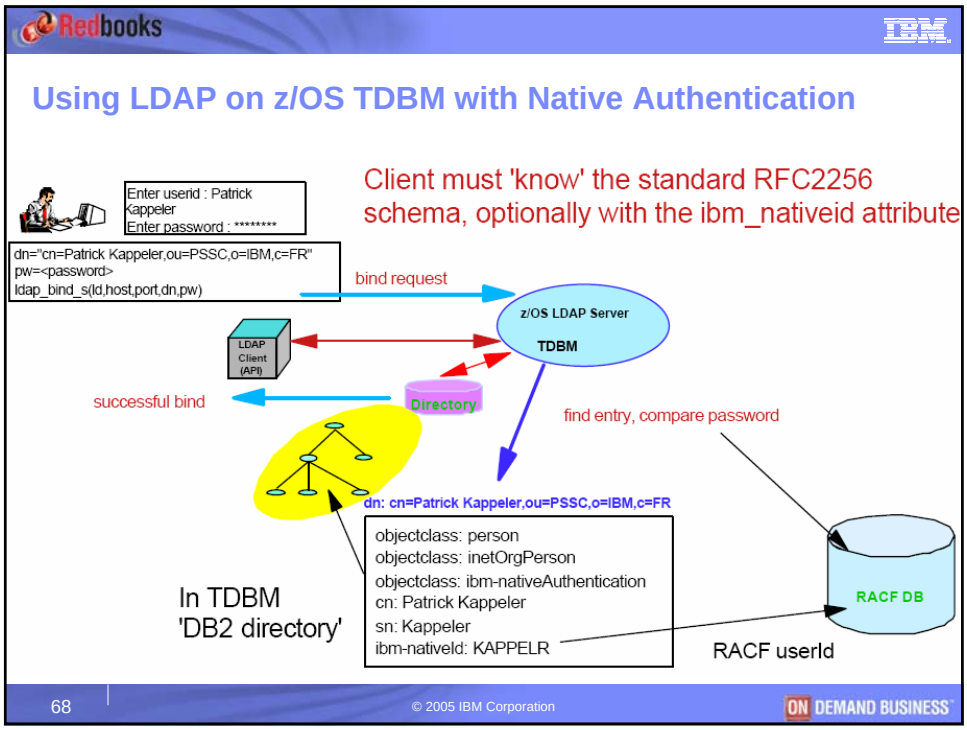
Redbooks IBM

LDAP on z/OS (z/OS v1.5)



66 © 2005 IBM Corporation ON DEMAND BUSINESS


Redbooks
IBM
Using LDAP on z/OS TDBM (DB2) as authentication database
Client must 'know' the standard RFC2256 schema
Enter userid : Patrick Kappeler
Enter password : ********
dn="cn=Patrick Kappeler,ou=PSSC,o=IBM,c=FR"
pw=<password>
ldap_bind_s(ld,host,port,dn,pw)
bind request
LDAP Client (API)
z/OS LDAP Server
TDBM
successful bind
Directory
find entry
compare password
dn: cn=Patrick Kappeler,ou=PSSC,o=IBM,c=FR
objectclass: person
objectclass: inetOrgPerson
cn: Patrick Kappeler
sn: Kappeler
uid: pkaplr
userpassword: XXXX..
DB2 Tables
password can be kept encrypted
67
© 2005 IBM Corporation
ON DEMAND BUSINESS


Redbooks
IBM
Using LDAP on z/OS TDBM with Native Authentication
Client must 'know' the standard RFC2256 schema, optionally with the ibm_nativeid attribute
Enter userid : Patrick Kappeler
Enter password : ********
dn="cn=Patrick Kappeler,ou=PSSC,o=IBM,c=FR"
pw=<password>
ldap_bind_s(ld,host,port,dn,pw)
bind request
LDAP Client (API)
z/OS LDAP Server
TDBM
successful bind
Directory
find entry, compare password
dn: cn=Patrick Kappeler,ou=PSSC,o=IBM,c=FR
objectclass: person
objectclass: inetOrgPerson
objectclass: ibm-nativeAuthentication
cn: Patrick Kappeler
sn: Kappeler
ibm-nativeId: KAPPELR
In TDBM 'DB2 directory'
RACF DB
RACF userId
68
© 2005 IBM Corporation
ON DEMAND BUSINESS

Redbooks IBM

Security: LDAP on z/OS Native Authentication

- TAM and WAS can both use LDAP on z/OS (IBM z/OS Security Server).
- LDAP Native Authentication (LNA) allows authentication to be done using RACF userids and passwords.
- LDAP Native Authentication requires to use a TDBM back-end (DB2 z/OS tables).

Why should you enable LDAP Native Authentication?

- You have the need for a central user registry (Single Sign-On).
- You want the ability to reuse RACF userids/passwords using an LDAP interface.
- You are looking to front-end WebSphere Application Server for z/OS with a security product like Tivoli Access Manager for e-business.

- LDAP Native Authentication uses the `ibm-nativeId` and/or `uid` attributes to map an LDAP userid to a RACF userid or uid.
- The LDAP server is accessible from WebSphere on all platforms

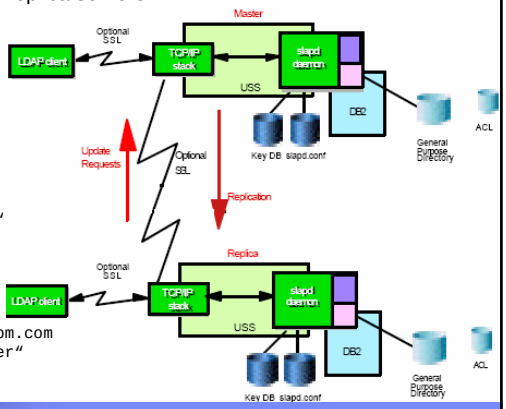
69 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

LDAP on z/OS replication

- LDAP on z/OS supports two types of replication:
  - **Peer- to-peer** replication: each LDAP peer server is a read-write server. Updates processed on one peer server are replicated to all the other peer servers.
  - **Read-only** replication: a single read-write LDAP server (the master) replicates the updates it processes to a set of read-only replica servers.

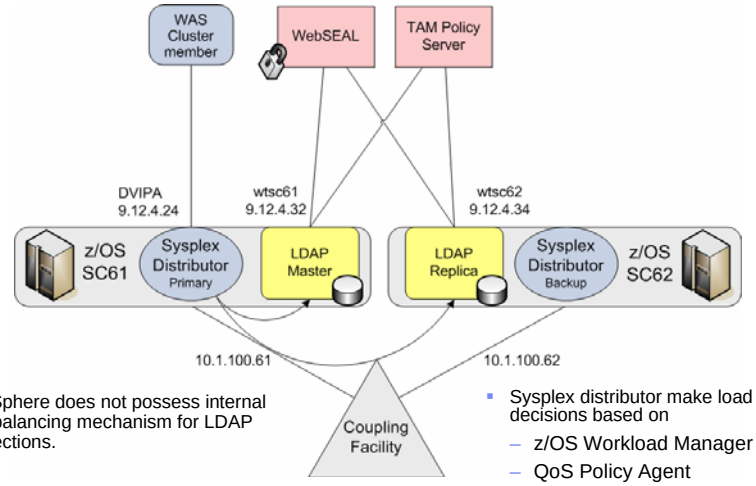
- Master SC61 server entry for replication:
  - `dn: cn=ReplicaSC62,o=ITSO`
    `objectclass: replicaObject`
    `cn: ReplicaSC62`
    `replicaHost: wtsc62.itso.ibm.com`
    `replicaPort: 3389`
    `replicaBindDn:cn=Replication User`
    `replicaCredentials:secret`
    `description:"LDAP Replica on SC62"`



- Replica SC62 server configuration:
  - `masterServer ldap://wtsc61.itso.ibm.com`
    `masterServerDN "cn=Replication User"`
    `masterServerPW secret`

70 © 2005 IBM Corporation ON DEMAND BUSINESS

LDAP on z/OS high availability solution



- WebSphere does not possess internal load balancing mechanism for LDAP connections.
- Sysplex distributor make load distribution decisions based on
  - z/OS Workload Manager (WLM)
  - QoS Policy Agent
- TAM components possess internal load balancing mechanisms.

71 © 2005 IBM Corporation ON DEMAND BUSINESS

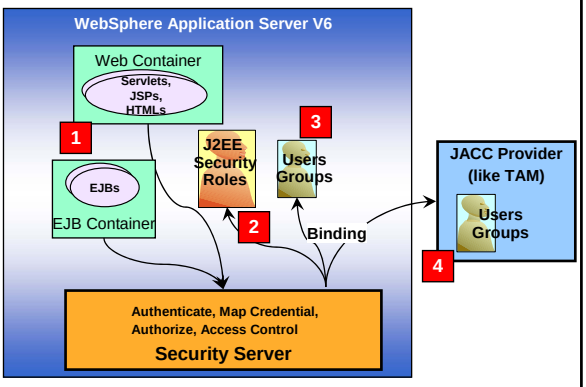
Part 2: WebSphere for z/OS front-end security solutions

72 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

JACC Introduction

- JACC allows applications servers to interact with third party authorization providers via standard interfaces to make authorization decisions
  - JACC defines permission classes for both the EJB and Web container
- Does not specify how to assign principals to roles
- The security policy, as well as the user or group bindings to the Security Role, are maintained by the JACC provider



73 © 2005 IBM Corporation ON DEMAND BUSINESS

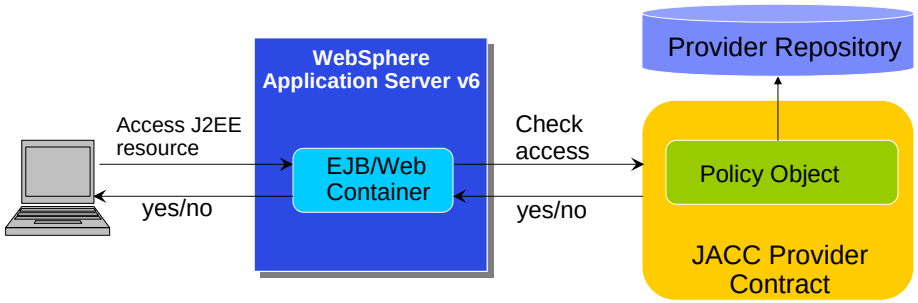
Redbooks IBM

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

74 © 2005 IBM Corporation ON DEMAND BUSINESS

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

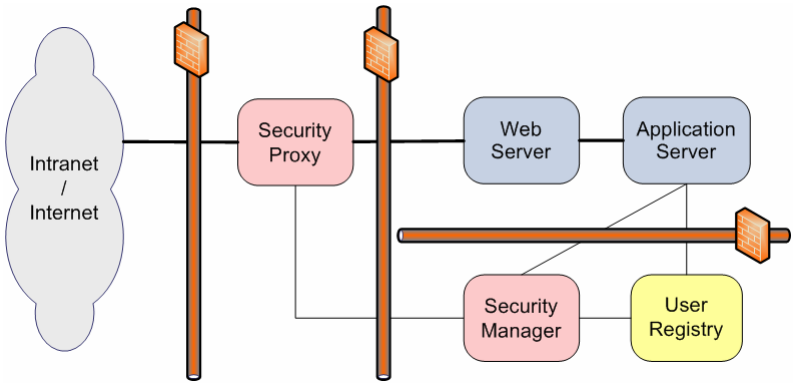
75 © 2005 IBM Corporation

Part 2: WebSphere for z/OS front-end security solutions

76 © 2005 IBM Corporation

Redbooks IBM

The Big Picture



77 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

What is Tivoli Access Manager (TAM) ?

An Integrated Security Platform for e-business

- Provides centralized security management
- Provides integrated, policy-based management
- Delivers single sign-on to Web-based applications
- Lets the right people in, controlling access to the right applications and data

Tivoli. software

78 © 2005 IBM Corporation ON DEMAND BUSINESS

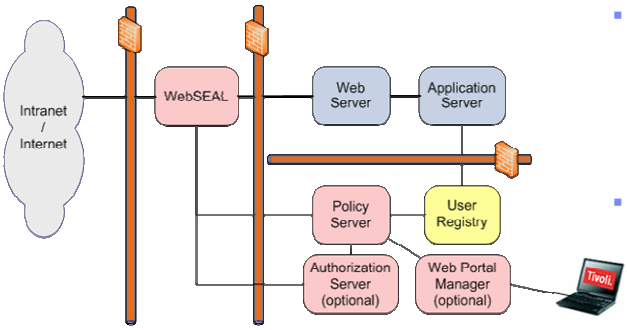
TAM features

- **Authentication**: TAM validates the user's identity.
- **Authorization**: TAM handles authorization through the use of the followings:
  - TAM authorization service
  - Access Control Lists (ACLs), Protected Object Policies (POP) and authorization rules
  - Standard-based authorization APIs such as aznAPI (C language) and JAAS (Java language)
  - External authorization service
- **Quality of protection**: TAM protects any information transmitted between client and server (SSL).
- **Scalability**: TAM uses replication of services, off-loading of authentication and authorization services.
- **Accountability**: TAM provides logging and auditing capabilities
- **Centralized management**: TAM provides an administration GUI (Web Portal Manager), a command line utility (pdadmin) and an administration API.

79 © 2005 IBM Corporation ON DEMAND BUSINESS

TAM components

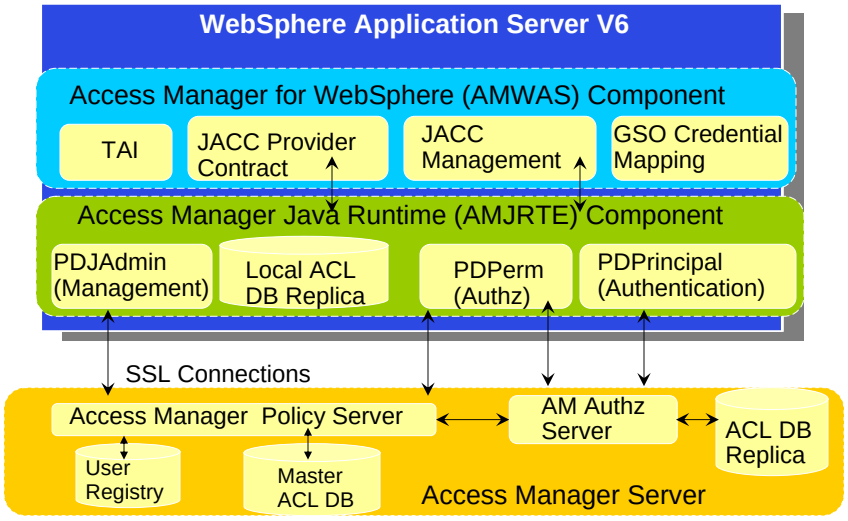
- **User registry**: database of the user identities.
- **Policy Server**: maintains the master authorization database for the secure domain.
- **Authorization Server** (optional): offloads access control and authorization decisions from the policy server. It maintains a replica of the authorization policy database.



- **WebSEAL**: Remote Proxy Security Server (RPSS). It performs authentication and junction authorization for incoming HTTP requests.
- **Web Portal Manager** (optional): Web-based graphical user interface (GUI) used for TAM administration.

80 © 2005 IBM Corporation ON DEMAND BUSINESS

TAM Components for WebSphere

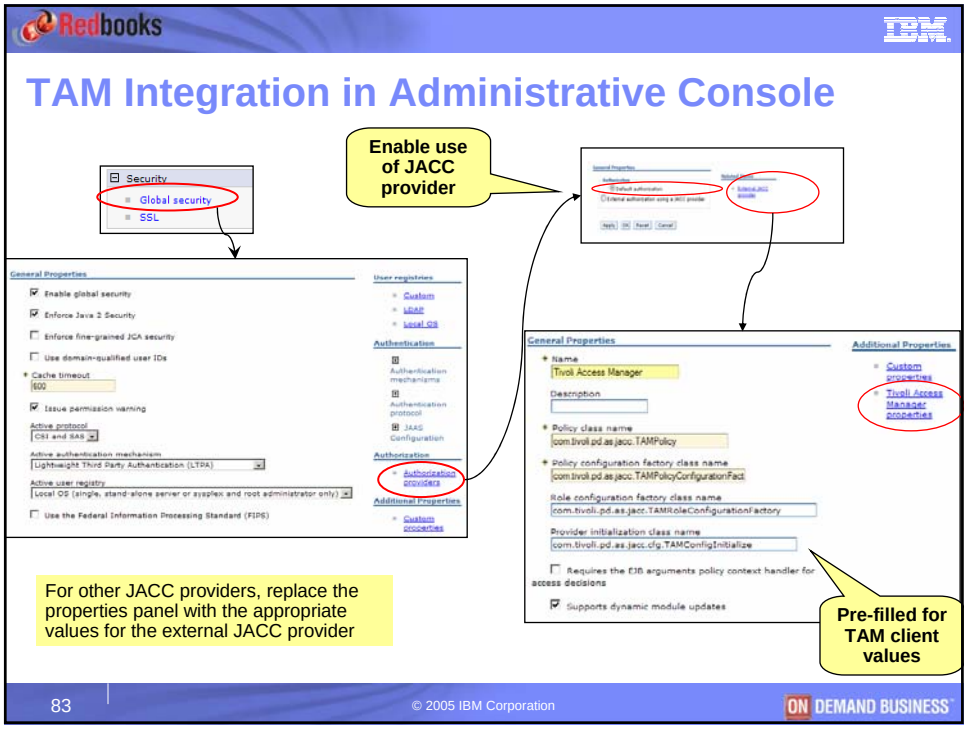


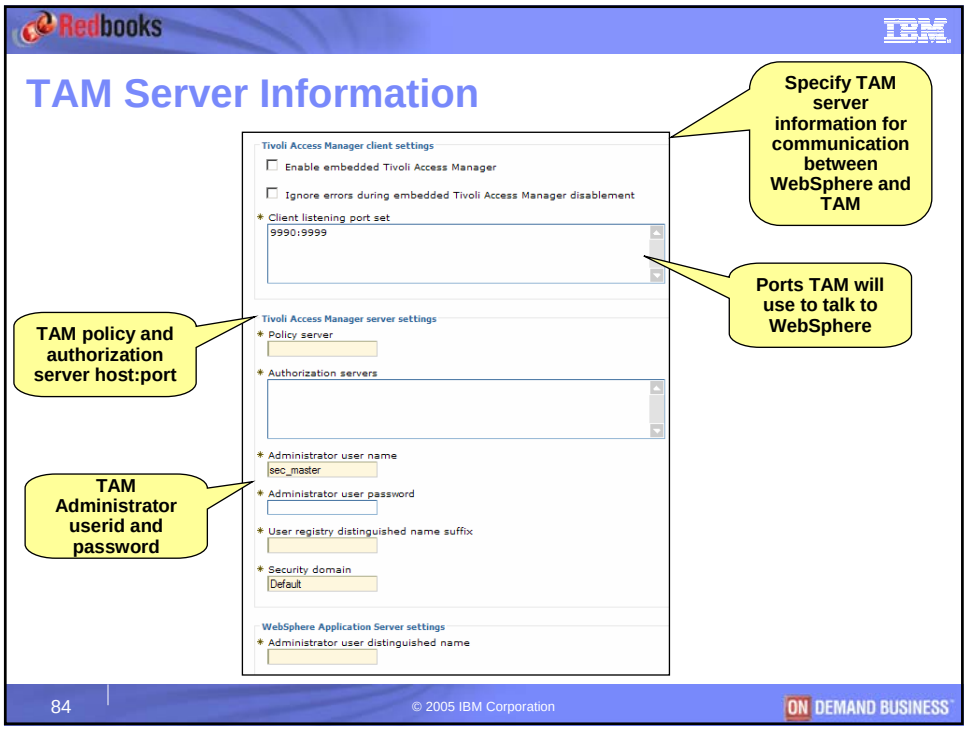
81 © 2005 IBM Corporation

TAM Integration

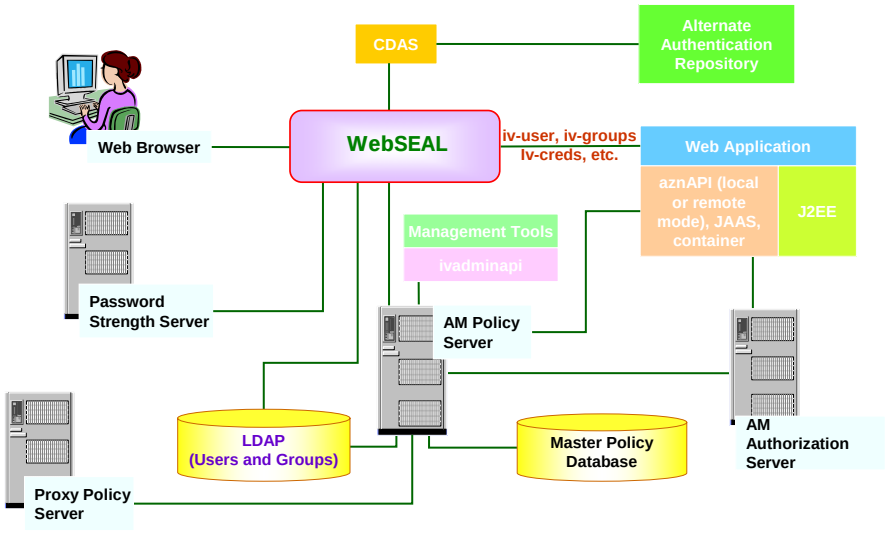
- TAM client pieces are embedded in the WebSphere Application Server V6
- TAM is the default JACC provider for WebSphere
- TAM Policy Server is included with WebSphere Application Server V6 Network Deployment package
- TAM client can be configured using the scripting or the Administration Console
- In addition to authorization, TAM server can also provide authentication functionality
- When TAM is used as the JACC provider, the GUI panels and the wsadmin scripting used to associate the Principals (users/groups) to roles directly communicate with the TAM server
- The TAM client can be configured using the scripting and the GUI management facilities of WebSphere
- Authentication can also be performed by the TAM server

82 © 2005 IBM Corporation


Redbooks
IBM
TAM Integration in Administrative Console
Enable use of JACC provider
Security
Global security
SSL
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
General Properties
Name
Tivoli Access Manager
Description
Policy class name
com.tivoli.pd.as.jacc.TAMPolicy
Policy configuration factory class name
com.tivoli.pd.as.jacc.TAMPolicyConfigurationFact
Role configuration factory class name
com.tivoli.pd.as.jacc.TAMRoleConfigurationFactory
Provider initialization class name
com.tivoli.pd.as.jacc.cfg.TAMConfigInitialize
Requires the EJB arguments policy context handler for access decisions
Supports dynamic module updates
Additional Properties
Custom properties
Tivoli Access Manager properties
For other JACC providers, replace the properties panel with the appropriate values for the external JACC provider
Pre-filled for TAM client values
83
© 2005 IBM Corporation
ON DEMAND BUSINESS


Redbooks
IBM
TAM Server Information
Specify TAM server information for communication between WebSphere and TAM
Tivoli Access Manager client settings
Enable embedded Tivoli Access Manager
Ignore errors during embedded Tivoli Access Manager disablement
Client listening port set
9990:9999
Ports TAM will use to talk to WebSphere
TAM policy and authorization server host:port
Tivoli Access Manager server settings
Policy server
Authorization servers
TAM Administrator userid and password
Administrator user name
sec_master
Administrator user password
User registry distinguished name suffix
Security domain
Default
WebSphere Application Server settings
Administrator user distinguished name
84
© 2005 IBM Corporation
ON DEMAND BUSINESS

Tivoli Access Manager Component Architecture



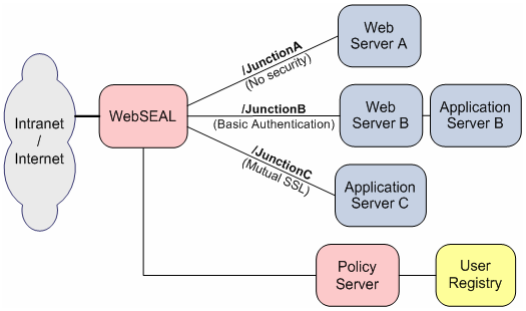
85 © 2005 IBM Corporation

WebSEAL junction

- A WebSEAL junction is a connection configuration between a front-end WebSEAL server and one or multiple back-end servers.

https://www.ibm.fr/JunctionB/ApplicationB

WebSEAL Hostname — WebSEAL junction — Back-end server Context-root

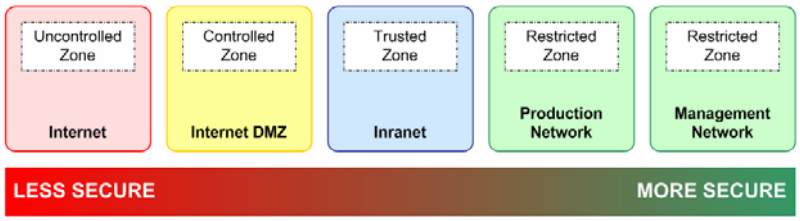


- Junctions support multiple authentication and delegation mechanisms: Basic Authentication, SSL…
- Junctions support multiple Single Sign-On (SSO) mechanisms: TAI, LTPA, GSO…

86 © 2005 IBM Corporation

Redbooks IBM

Security: Network zones



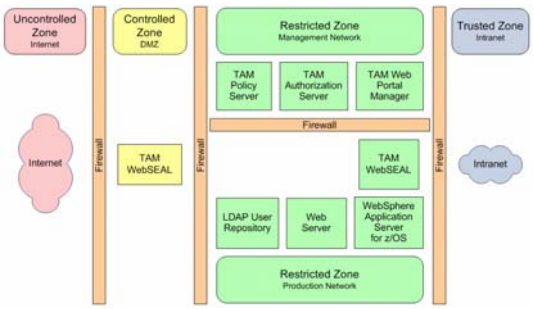
- **Internet DMZ** (controlled zone): It provides a buffer between the uncontrolled Internet and internal networks.
- **Intranet** (trusted zone): A trusted zone is one that is generally not heavily restricted in use, but an appropriate span of control exists to assure that network traffic does not compromise operation of critical business functions.
- **Production or management network(s)** (restricted zones): One or more network zones may be designated as restricted. They support functions to which access must be strictly controlled, and of course, direct access from an uncontrolled network should not be permitted.

87 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Security: Components placement

- **WebSEAL**: it should always be the sole HTTP/HTTPS contact point for a Web server from an Internet client.
- **TAM Policy Server, Authorization Server**: it should always be placed in a restricted (or at least a trusted) zone.
- **LDAP user registry**: The registry should be in a restricted zone to which access may be strictly controlled, or at least a trusted network. Firewall configurations should disallow any possibility of access to the user registry from the uncontrolled zones such as the Internet.
- **Web server**: it is recommended that the back-end Web servers does not reside in an Internet DMZ.
- **Application server**: it should be placed in the production network restricted zone.

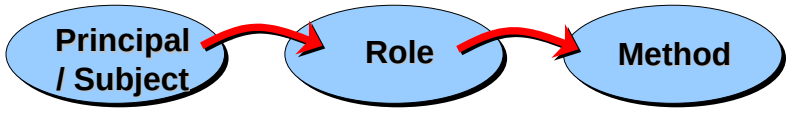


88 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Security: WAS, TAM and J2EE security

- Two mappings are very important to J2EE security:



•Principal/subject to Role
•Determined by Administrator
•Relatively dynamic

•Role to Method
•Determined by deployment descriptor

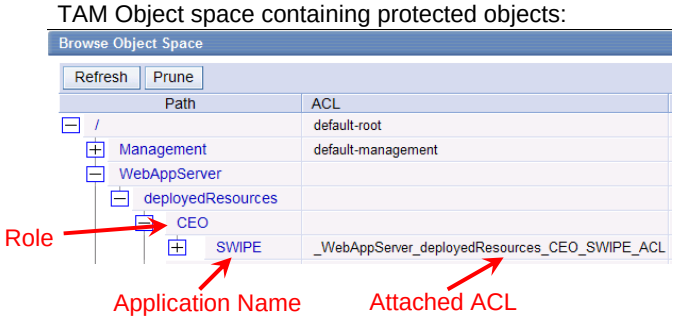
- WAS knows/handles Role-to-Method mapping
- WAS asks TAM for Principal/Subject-to-Role mapping
- WAS runs the AMWAS module for authorization

89 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Security: TAM J2EE Subject-to-Role mapping

TAM Object space containing protected objects:



- J2EE Roles are protected objects in TAM Object Space.
- ACLs are attached to protected objects.
- Users/groups authorizations are set in ACLs.
- CellName, HostName and ServerName can be specified for better granularity.

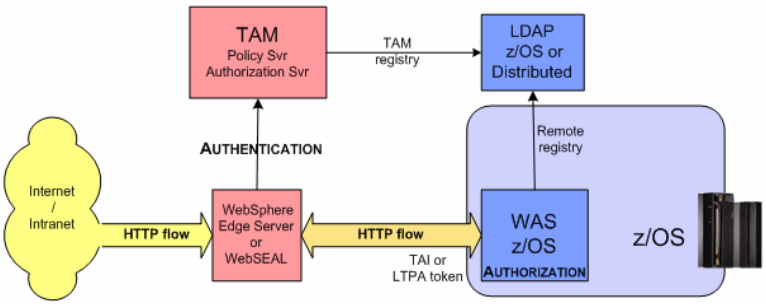
90 © 2005 IBM Corporation ON DEMAND BUSINESS

Sample integration solutions

- **Solution 1**: TAM authentication for WAS
  - TAM performs authentication only.
  - Single Sign-On occurs using the shared LDAP user registry.
- **Solution 2**: TAM authentication and LocalOS authorization for WAS
  - TAM performs authentication only.
  - Single Sign-On occurs using LDAP Native Authentication.
  - WAS for z/OS uses LocalOS (RACF) for authorization.
- **Solution 3**: TAM authentication and authorization for WAS
  - TAM performs authentication and J2EE authorization.
  - Single Sign-On occurs using the shared LDAP user registry.
- **Solution 4**: TAM authentication, authorization and Native Authentication for WAS
  - TAM performs authentication and J2EE authorization.
  - Single Sign-On occurs using the shared LDAP user registry.
  - LDAP Native authentication makes passwords be validated against RACF.

91 © 2005 IBM Corporation ON DEMAND BUSINESS

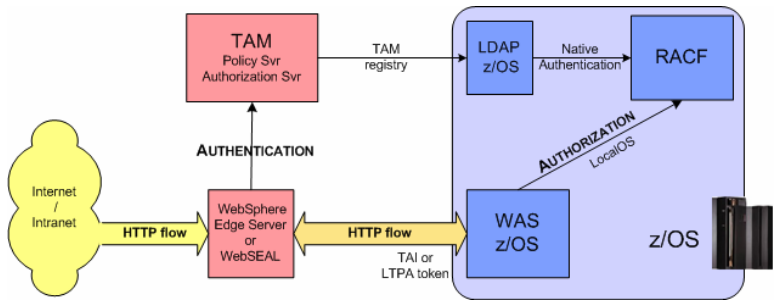
Solution 1: TAM authentication for WAS



- TAM allows cross-platform centralized authentication management.
- LDAP can be a central user registry for Single Sign-On solutions.
- The TAI or LTPA tokens allow Single Sign-On between WebSEAL and WebSphere Application Server.

92 © 2005 IBM Corporation ON DEMAND BUSINESS

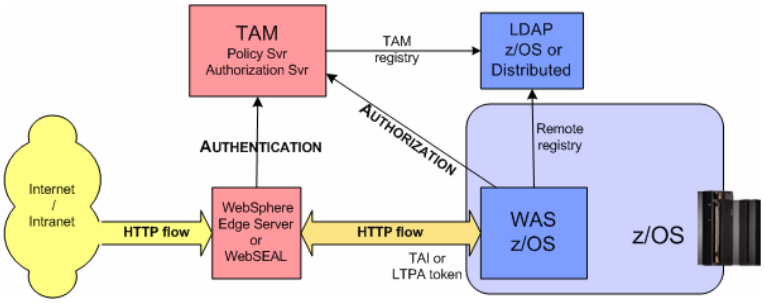
Solution 2: TAM authentication and LocalOS authorization for WAS



- WAS for z/OS uses LocalOS (RACF) for authorization.
- Native Authentication allows user registry passwords to be stored inside RACF where they are not accessible from outside.
- Native Authentication allows end users to enter their MVS userid and password when they access a URL that requires authentication.
- LDAP can be a central user registry for Single SignOn solutions.
- The TAI or LTPA tokens allow Single SignOn between WebSEAL and WebSphere Application Server.

93 © 2005 IBM Corporation ON DEMAND BUSINESS

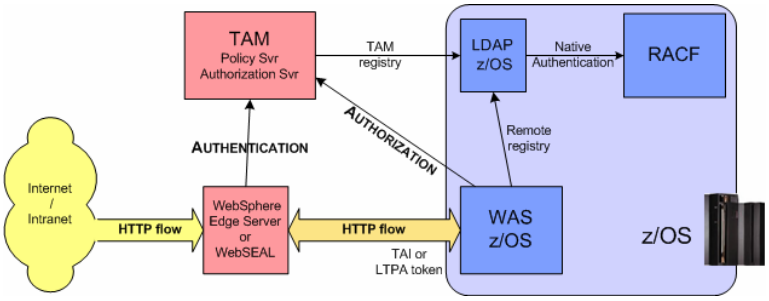
Solution 3: TAM authentication and authorization for WAS



- TAM allows cross-platform centralized authentication and authorization management.
- TAM allows cross-platform centralized users access to J2EE roles management.
- LDAP can be a central user registry for Single SignOn solutions.
- The TAI or LTPA tokens allow Single SignOn between WebSEAL and WebSphere Application Server.

94 © 2005 IBM Corporation ON DEMAND BUSINESS

Solution 4: TAM authentication, authorization and Native Authentication for WAS



- TAM allows cross-platform centralized authentication and authorization management.
- TAM allows cross-platform centralized users access to J2EE roles management.
- Native Authentication allows user registry passwords to be stored inside RACF.
- Native Authentication allows end users to enter their userid and MVS password when they access a URL that requires authentication.
- LDAP can be a central user registry for Single SignOn solutions.
- The TAI or LTPA tokens allow Single SignOn between WebSEAL and WAS.

95 © 2005 IBM Corporation ON DEMAND BUSINESS

TAM and WAS z/OS integration advantages

- Industry leading security provider
- Container-based security: EJB/servlet/JSP developers focus on business.
- Central administration: Manage WebSphere and non-WebSphere environments.
- Flexibility: More flexible user-to-role policies are possible.
- Transparency: Added value, yet no changes to J2EE code.
- Standards-based: J2EE
- Dynamic: Make user-to-role changes without Application Server restart
- Early user authentication with WebSEAL or WebSphere Edge Server TAM plug-in
- URI based access control

96 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 2: WebSphere for z/OS front-end security solutions

97 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Single Sign On (SSO)

- SSO is a mechanism which allows HTTP clients to authenticate with any server, and automatically be authenticated with any other server in the same Network Deployment cell or across cells
- Requires all servers to have LTPA authentication type
  - LTPA keys and User registry must be shared between servers in different the cells, for SSO to work
- The authenticating server issues an LTPA login token and sends it in HTTP cookie named “LtpaToken” as part of the HTTP response
  - Part of the cookie contains the supported domains
  - For each subsequent HTTP request to a server in the supported domain, the client sends the cookie – the server then uses the token to authenticate and authorize the request
- Since V5.1.1, multiple DNS domains are supported

98 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

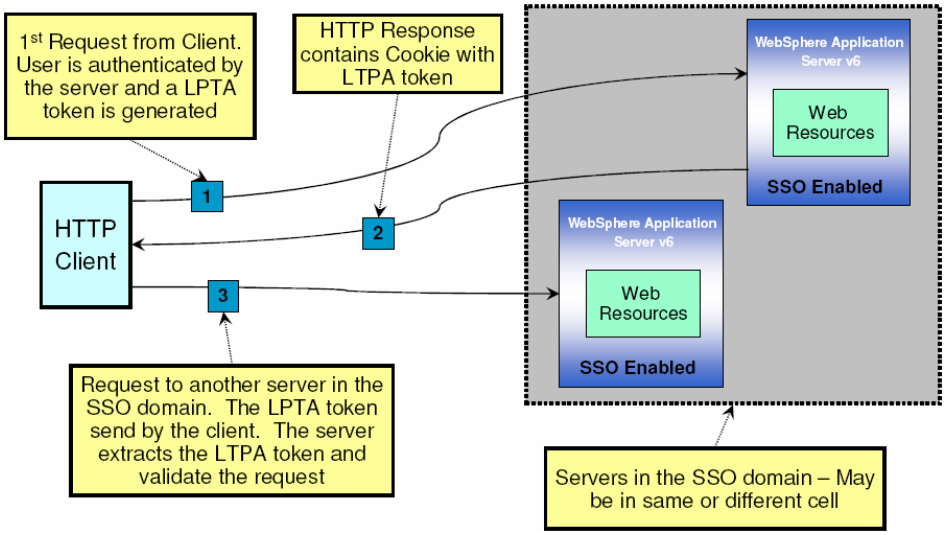
SSO Domain - Possible Domain values

Possible domain values	Comment and Example
Blank	No domain name send – SSO will only work with this server
Single domain name	Example: austin.ibm.com Any subset of this domain name will participate in SSO – for example – **hostA.austin.ibm.com**. Server **hostA.raleigh.ibm.com** will not work
"UseDomainFromURL"	SSO domain name value to the domain of the host that makes the request. For example, if an HTTP request comes from **clientA.raleigh.ibm.com**, WebSphere Application Server set the SSO domain name value to **raleigh.ibm.com**
Multiple domain names, separated by a valid delimiter (semi-colon, space, pipe, comma)	Example: austin.ibm.com;raleigh.ibm.com Any hosts from this domain can participate in this SSO
Multiple domain names and UseDomainFromURL	Example: austin.ibm.com;raleigh.ibm.com; UseDomainFromURL If the HTTP request URL host is rchland.ibm.com, then hosts in domain austin.ibm.com, raleigh.ibm.com and rchland.ibm.com can participate in this SSO

99 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

SSO: How it works



100 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

SSO between WebSphere z/OS and Distributed

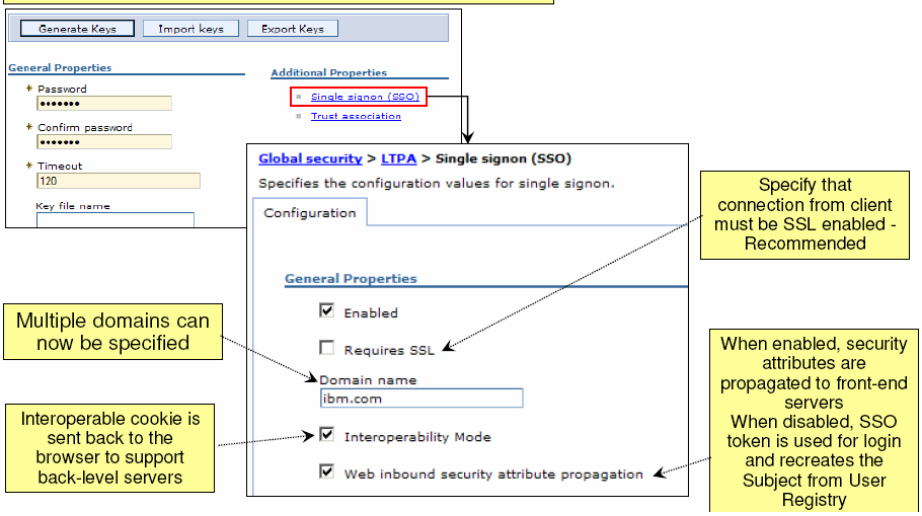
- WebSphere provides transparent Single Sign-On (SSO) among all platforms if
  - The same user registry is used by both servers
    - WebSphere on z/OS uses the same LDAP server as Distributed WebSphere (no Local OS)
  - Servers use LTPA authentication
    - SWAM does not provide for SSO across systems
    - ICSF authentication is z/OS specific
  - The same LTPA encryption keys are used
    - LTPA encryption key is shared between servers in different Cells
- SSO occurs at two levels
  - Web communication
    - Relies on sharing the SSO token (a.k.a. LTPA cookie) via the web browser
  - IIOP communication
    - Relies on sharing the authentication token (a.k.a. LTPA token) by sending it over the IIOP channel

101 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

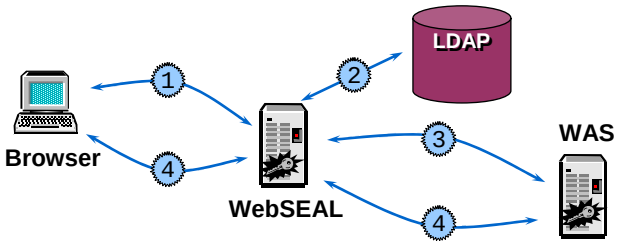
SSO: Administration

Security → Global Security → Authentication Mechanism → LTPA



102 © 2005 IBM Corporation ON DEMAND BUSINESS

SSO between WebSeal and WebSphere z/OS Solution 1 : Using LTPA token



- WebSEAL acts as an LTPA server
- WebSEAL and WAS share the same user registry
- LTPA cookie does not reach the user browser

- Proxy authenticates user and does coarse-grained authorization (URI level)
- Proxy forwards the http request including an encrypted LTPA token
- WAS decrypts the LTPA token and retrieve the user credentials
- WAS uses the credentials to make finer-grained authorization decisions

103 © 2005 IBM Corporation

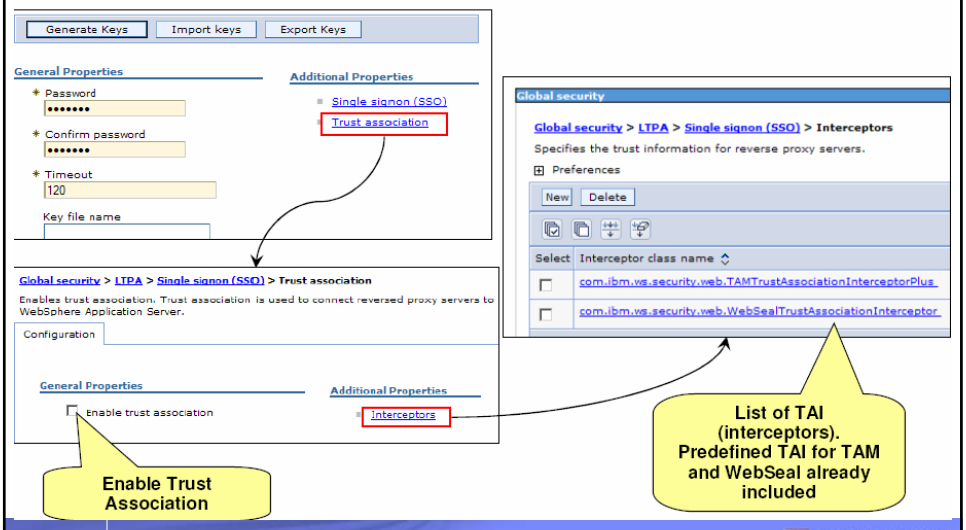
Trust Association

- Allows third party Reverse Proxy Security servers (RPSS such as WebSeal) to act as a front-end authentication server for Web Http requests into WebSphere Application Server
- WebSphere Application Server validates the RPSS using the Trust Association interceptors (TAI) of the proxy server
  - TAI is custom java code that you write or buy.
  - TAI is the answer to most custom authentication situations.
  - Two sample TAIs (TAI and TAI++) work with Tivoli Access Manager
- WebSphere Application Server can be set up to receive HTTP requests exclusively using the proxy server, or to accept HTTP requests directly as well

104 © 2005 IBM Corporation

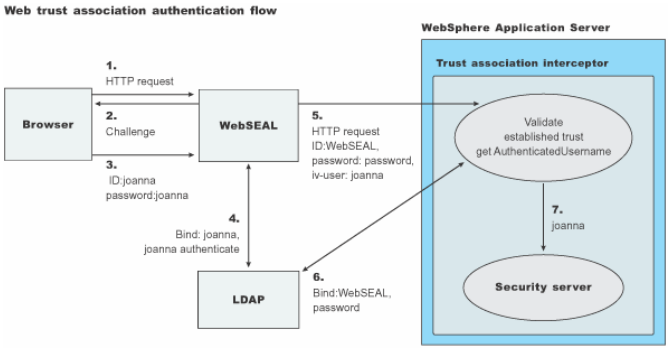
Trust Association – Administration

Security → Global Security → Authentication Mechanism → LTPA)



© 2005 IBM Corporation

SSO between WebSeal and WebSphere z/OS Solution 2 : Using Trust Association



- Trust relationship between WebSEAL and WAS implemented using mutually-authenticated SSL connection or Basic Authentication.

- Proxy authenticates user and does coarse-grained authorization (URI level)
- Proxy sends its own identity information as well as user information to WAS
- WAS calls TAI: Are you the right TAI to handle this request? Do you trust me? TAI converts user info to WebSphere credentials
- WAS uses the credentials to make finer-grained authorization decisions

© 2005 IBM Corporation

Redbooks IBM

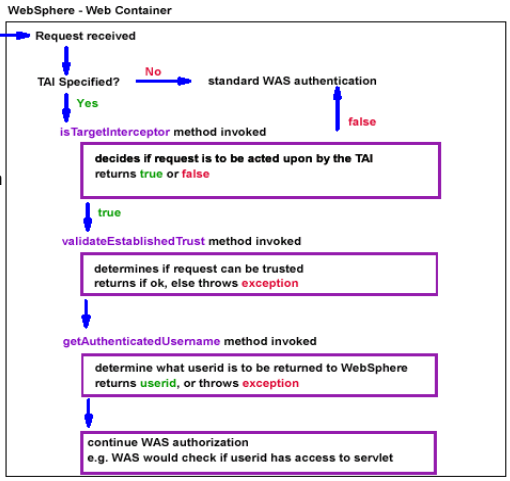
Trust Association Interceptor

Trust Association Interceptor

The Trust Association feature is a point in the **WebSphere authentication process** where an organization can insert their own code to achieve whatever authentication outcome they desire.

It can be **coded** so that only some requests are validated by the TAI. That is the TAI can decide that it will not perform a trust association operation on a request, in which case the authentication process returns to WebSphere for processing.

Otherwise, as the name implies, WebSphere is going to "trust" the result returned to it by the TAI.



107 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 2: WebSphere for z/OS front-end security solutions

108 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Secure Sockets Layer (SSL) in WebSphere v6

- SSL provides transport layer security with authenticity, integrity, and confidentiality, for a secure connection between a client and server in WebSphere v6
- SSL is used by multiple components within the Application Server
  - Built-in HTTP server within the Web Container for HTTP over SSL
  - ORB component using IIOP over SSL
  - Security LDAP client using LDAP over SSL to connect to LDAP user registry
- Administration:
  - You configure SSL settings using SSL repertoire
  - Once defined, you can use the SSL definitions where needed

109 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

SSL Repertoires

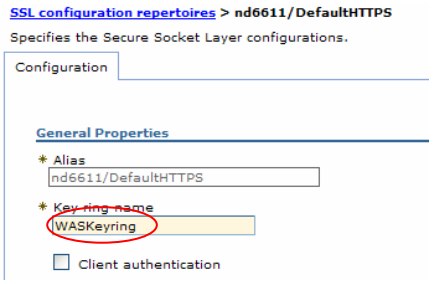
- Every SSL port is associated to an SSL “repertoire”
- **Two types of SSL "repertoires"** on z/OS have distinct configuration types:
  - **System SSL** used for HTTPS,IIOP and RMI connector
    - Used for HTTPS and IIOP communication
    - Always uses keys stored in SAF keyrings
    - MutualAuthCBindCheck=true property of the SSL repertoire enforces that:
      - All SSL connection to the Web Container must have a client certificate
      - The client certificate should map to a valid RACF user ID.
      - The mapped userid must have CONTROL access to CB.BIND.cluster_name, where cluster_name is the cluster short name for the target application servers.
      - If these conditions are not met, the connection is closed.
  - **JSSE** mandatory for SOAP/HTTPS requests including wsadmin
    - HFS .jks keystores
    - SAF keyrings

110 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

System SSL Repertoire – RACF Keyring

- System SSL repertoires are required to use SSL over HTTP and IIOP.



- The following command is an example of the command invoked during customization to create a RACF Keyring
  - ***RACDCERT ADDRING(WASKeyring) ID(ASCR1)***
- The following commands are examples of the commands invoked during customization to add the RACF CA and client certificates to the RACF Keyring
  - ***RACDCERT ID(ASCR1) CONNECT(RING(WASKeyring) LABEL('WebSphereCA') CERTAUTH)***
  - ***RACDCERT ID(ASCR1) CONNECT(RING(WASKeyring) LABEL('DefaultWASCert.BBOC001') DEFAULT)***

111 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

SSL Encryption – RACF Certificates

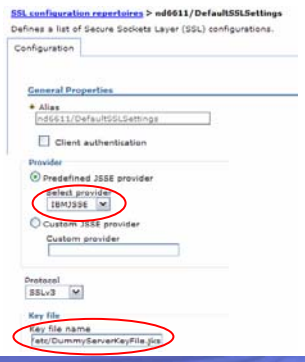
- The following command is an example of the command invoked during customization to generated a self-signed CA certificate for WebSphere using RACF
  - ***RACDCERT CERTAUTH GENCERT SUBJECTSDN(CN('WAS CertAuth for Security Domain') OU('CTFMVS09.WebSphere for zOS')) WITHLABEL('WebSphereCA') TRUST NOTAFTER(DATE(2010/12/31))***
- The following command is an example of the command invoked during customization to generate a client certificate using the above CA certificate for the WebSphere server's identity.
  - ***RACDCERT ID (ASCR1) GENCERT SUBJECTSDN(CN('ASCR1.BBOC001') O('IBM') OU('CTFMVS09')) WITHLABEL('DefaultWASCert.BBOC001') SIGNWITH(CERTAUTH LABEL('WebSphereCA')) NOTAFTER(DATE(2010/12/31))***
- The following command exports WebSphere's RACF generated CA certificate for sending to other systems.
  - ***RACDCERT CERTAUTH EXPORT(LABEL('WebSphereCA')) DSN(CERTAUTH.ARM)***
- Import WebSphere's CA certificate into a JSSE Keystore
  - ***keytool -import -v -trustcacerts -alias "WebSphereCA" -file CERTAUTH.ARM -keystore jsse.jks –storepass secret***

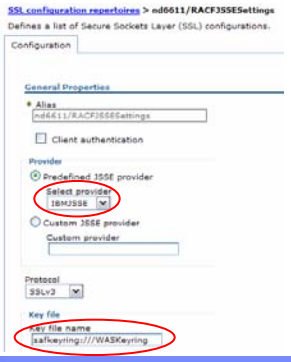
112 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

JSSE Repertoire – non-z/OS and RACF Keyring

- A Java Secure Socket Extension (JSSE) repertoire is mandatory for SOAP/HTTPS requests (wsadmin)
  - HFS .jks keystores (default configuration before WAS 5.02)
  - SAF keyrings now the default (set-up by customization dialogs)

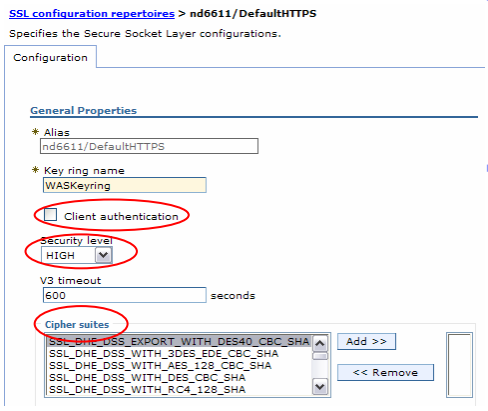




 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

SSL Repertoire configuration



- Client authentication
  - Specifies whether to request a certificate from the client for authentication purposes when making a connection.
- Security level
  - High specifies 128-bit ciphers only including digital signing.
  - Medium specifies 40-bit ciphers only including digital signing.
  - Low specifies digital signing ciphers only without encryption.
- Cipher Suites – Encryption methods

 © 2005 IBM Corporation ON DEMAND BUSINESS

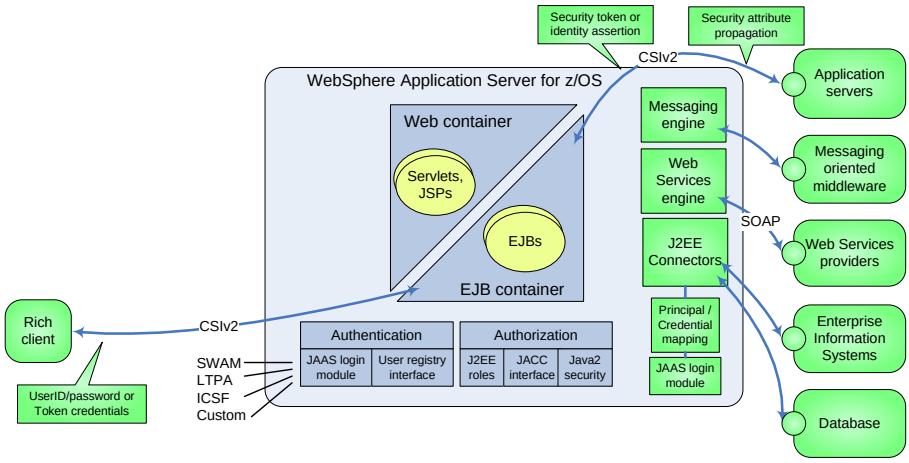
© 2005 IBM Corporation

Part 3

WebSphere for z/OS back-end security solutions

© 2005 IBM Corporation

Part 3: WebSphere for z/OS back-end security solutions



© 2005 IBM Corporation

© 2005 IBM Corporation

Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

117 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

118 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Authentication protocol for EJB client and server

- Authentication protocol determines the level of security and the type of authentication that needs to occur between the EJB client and the EJB for each request in a secure environment
  - It finds the appropriate authentication policy suitable for both the client and the server by coalescing of their configurations
- WebSphere for z/OS V6 supports 2 authentication protocols:
  - **CSIv2**: Common Secure Interoperability Version 2 RECOMMENDED
    - Defined by Object Management Group (OMG) and is part of the J2EE standards
  - **zSAS**: zSecure Authentication Service for backward compatibility
    - Used by previous levels of WebSphere Application Server
- EJB request and response uses Inter-ORB Protocol (IIOP) services
  - IIOP is a request-and-reply communications protocol used to send messages between two Object Request Brokers (ORBs)
- The EJB authentication protocol used by WebSphere V6 are add-on to IIOP services

119 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

CSIv2 Overview

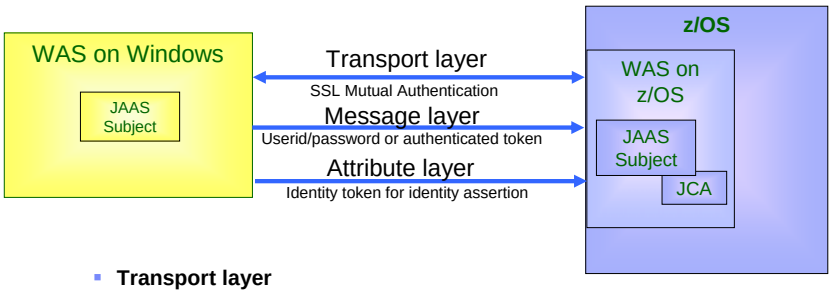
- CSIv2 defines the Security Attribute Service (SAS) that enables interoperable authentication, delegation and privileges
  - CSIv2 SAS supports SSL and interoperability across J2EE vendors (starting with J2EE 1.3 specification)
- Provides 3 layers of authentication, as shown in the table below:

Transport layer	Uses SSL client certificate as the identity	Attribute layer has the highest priority, followed by the message layer, and then the transport layer. If a client sends all three, only the identity token from the attribute layer is used
Message layer	Uses an user ID/password or an authenticated token with an expiration	
Attribute layer	Uses Identity token to support Identity assertion of an upstream server	

- CSIv2 features:
  - SSL Client Certificate Authentication
  - Message Layer Authentication
  - Identity Assertion
  - Security Attribute Propagation
  - Stateful and Stateless choices

120 © 2005 IBM Corporation ON DEMAND BUSINESS

CSIv2 authentication choices



- **Transport layer**
  - TCP/IP: transport layer authentication not possible
  - SSL: Client certificate authentication
- **Message layer**
  - Basic authentication
  - Security attribute propagation
- **Attribute layer**
  - Identity assertion

121 © 2005 IBM Corporation

Transport layer authentication: SSL Client Certificate

- An additional way to authenticate a client to a server using SSL client authentication
- Disable message layer on the client side security (user ID/password) option in the configuration, if the SSL certificate is the identity against which to invoke the method
- A credential is created by mapping the identity from the certificate to the user registry
  - For Local OS: The 1st attribute of the DN in the certificate is used to map to the user ID in the registry - Example: For DN "cn=Smith, ou=NewUnit, o=NewCompany, c=us", the user ID is "smith"
  - For LDAP: Either mapping the Subject field in the certification with the EXACT DN name or by matching attributes in the certificate to attributes of LDAP entries
- Advantage: Optimizes authentication performance, because an SSL connection is typically created anyway - Extra overhead of sending the client certificate is minimal
- Disadvantage: Complexity of setting up the keystore file on each client system

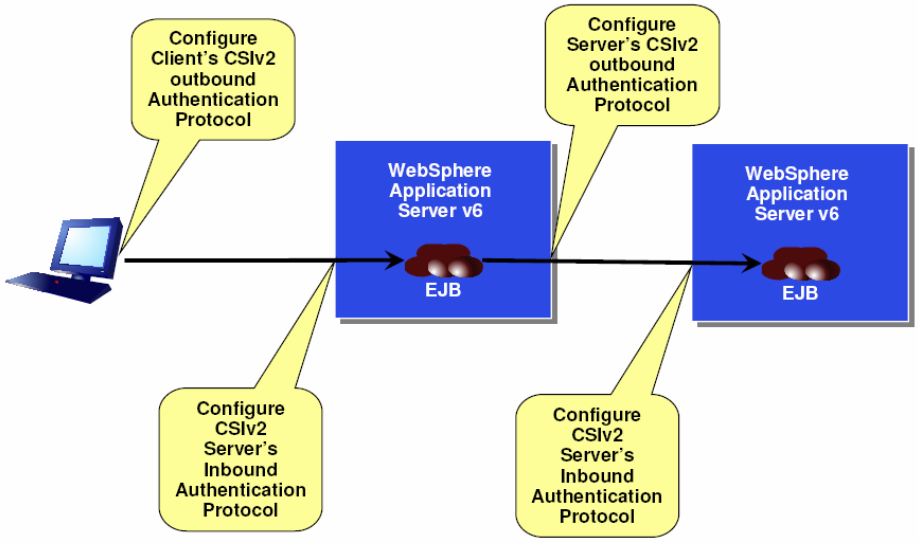
122 © 2005 IBM Corporation

Message Layer Authentication

- Defines the credential information and sends that authentication information, using a token, across the network to a receiving server
  - Token can be user ID/password or mechanism-specific format token, like LTPA token
    - Pure Java client uses basic authentication whereas a Servlet can use basic authentication or LTPA token
    - Cannot use SWAM authentication mechanism, since the tokens are not forwardable in SWAM
- The server knows the mechanism to use when reading and validating the token

123 © 2005 IBM Corporation

CSIv2 configuration



124 © 2005 IBM Corporation

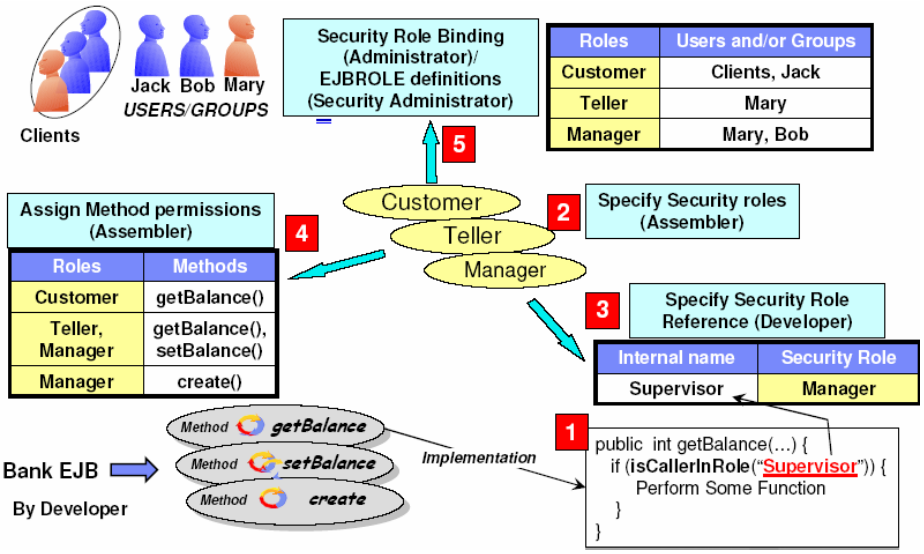
Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

125 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

J2EE Security: EJB Role Based Authorization



126 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

EJB Applications Programmatic APIs

- **IsCallerInRole** (String role-name)
  - Returns true if the bean caller is granted the specified security role
  - If the caller is not granted the specified role, or if the caller is not authenticated, it returns false
  - If the specified role is granted **Everyone** access, it always returns true
  - Must have security role reference defined in the deployment descriptor
- **getCallerPrincipal()**:
  - Returns the java.security.Principal object containing the bean caller name
  - If the caller is not authenticated, it returns a principal containing UNAUTHENTICATED name

Example:

```
public void myEJBmethod() {
        …
        // to get bean's caller using getCallerPrincipal()
        java.security.Principal principal = context.getCallerPrincipal();
        String callerId= principal.getName();
        // to check if bean's caller is granted Mgr role
        boolean isMgr = context.isCallerInRole("Mgr");
        …
}
```

127 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Changing Identity: “Run-As” Option

- EJB methods have the ability to change identity when calling downstream processes or EJBs
  - There are several different “Run-As” identities that you can choose from
  - Run-As specification applies to all the methods of the EJB
    - With IBM extension, you can specify different “Run-As” options for different methods within the same EJB
  - Does not change the identity of the z/OS thread
  - The role identity is specified either in WebSphere or in RACF EJBROLE profile

“Run As” options	Description
Client Identity	▪Bean takes on the same identity as the caller
Another Specified Role	▪Bean takes on identity of a specified user within the specified role ▪The specified role is part of the deployment descriptor and performed by the assembler ▪The specific user in the “Run-As” role is usually specified at deploy time
Server Identity	▪Bean takes on the identity of the user under which the server is running ▪This is an IBM extension to the specification

128 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

129 © 2005 IBM Corporation ON DEMAND BUSINESS

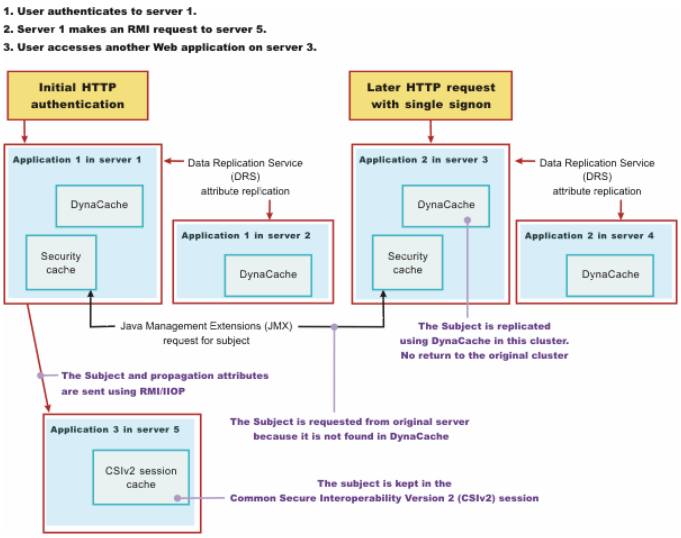
Redbooks IBM

Security Attribute Propagation

- Security attribute propagation enables propagation of security attributes (user identity, authenticated Subject contents and security context information) between Application servers
  - Alternatively, servers would have to query the User Registry or a custom login module to get the attributes – can be expensive from performance view point
  - Attributes might include original caller identity, location, IP address, dynamic group and so on
  - Previous versions of WAS propagated only the user name of the authenticated user, but ignored other security attribute information that other servers may need
- Different attribute propagation styles:
  - **Horizontal propagation**: across front-end servers for Web Applications using DynaCache and JMX
  - **Vertical propagation**: to downstream for EJBs using RMI-IIOP

130 © 2005 IBM Corporation ON DEMAND BUSINESS

Security Attribute Propagation – How it works



- **Initial Login**: authenticating the user information. The user provides credential (userid/password...). WebSphere validates the user against the user registry and looks up secure attributes that represent the user access rights.
- **Propagation Login**: validating the user information, typically a LTPA token, and then deserializing a set of tokens that constitute both custom objects and token objects.

131 © 2005 IBM Corporation ON DEMAND BUSINESS

Security Attribute Propagation – Tokens

- Subject-based tokens
  - **Authentication token (old LtpaToken)**
    - Contains the identity of the user only.
    - Converted to a cookie and sent to browser. This is equivalent to the old LtpaToken for backwards compatibility.
  - **Single Sign On (SSO) token (new LtpaToken2)**
    - Converted to a cookie and sent to browser. This represents the unique authentication. Named LtpaToken2 by default.
    - Contains stronger encryption and enables you to add multiple attributes to the token.
    - Contains the authentication identity and attributes that are used for contacting the original login server and the unique cache key for looking up the Subject.
  - **Authorization token**
    - Contains most of the authorization-related security attributes that are propagated. It is used by WebSphere to make J2EE Authorization decisions.
- Thread-based token
  - **Propagation token**
    - Not user specific and thus not part of Subject.
    - Represents thread context. Used to implement chaining support
- Can even implement custom versions of the above tokens
  - The Token framework serves as a way to notify WebSphere that you want these tokens propagated in a particular way.

132 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

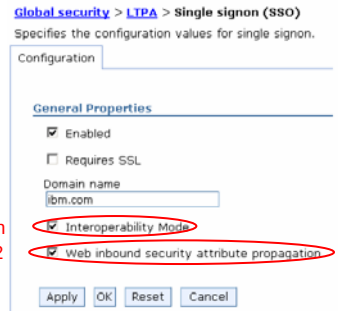
Part 3: WebSphere for z/OS back-end security solutions

133 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Horizontal Security Attribute Propagation

- Used if you need to gather dynamic security attributes set at the original login server that cannot be regenerated at the new front-end server
- The serialized information of the security attributes are automatically propagated to all the servers within the same Data Replication service (DRS)
- Benefits:
  - Do not need to perform any remote user registry calls because the application server can regenerate the Subject from the serialized information
- Configuration: Enabled on Single Sign On (SSO) panel
  - Select the **Web inbound security attribute propagation** option



Authentication Token LtpaToken
SSO Token LtpaToken2

134 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Horizontal Propagation - DynaCache

- WAS creates a private security cache in DynaCache
- Subjects are placed in cache and replicated in replication domain (e.g., the application server cluster)
- Using tokens from DynaCache results in a propagation login
- Cache lifetime
  - Lifetime of DynaCache entry is same as SSOToken lifetime (2 hours by default)
  - JMX SecurityAdmin clearAuthCache affects both local cache and DynaCache Sometimes Subject isn't in DynaCache

Horizontal Propagation - JMX

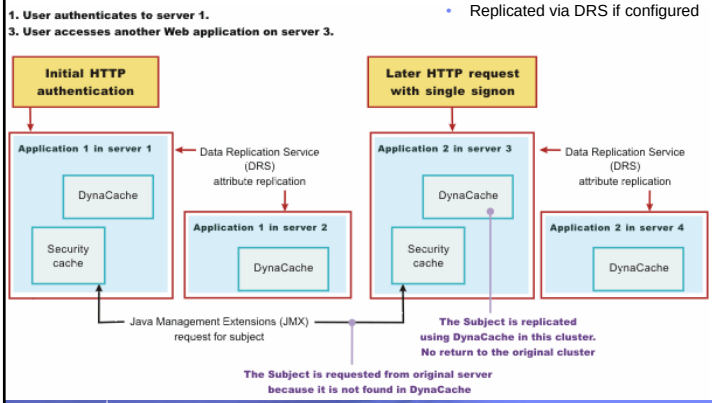
- JMX is fallback
  - If application server can't find Subject in cache, uses JMX to query server that created (and hopefully still has) Subject
  - WAS makes a secure JMX admin call to application server. If cross cell,
    - The two servers must share common security infrastructure – registries, SSL keys, etc.
    - Make sure your cell's security server id has admin access to remote cell
- If JMX fails, WAS falls back to an initial login to recreate the Subject
  - Your login modules will be called
  - User will not have to reauthenticate!

135 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Horizontal Propagation scenario

- A user authenticates
- Subject is created
  - Placed in application server cache
  - Placed in DynaCache
    - Converted to serialized tokens
    - Replicated via DRS if configured



- SSO Token is created that represents this Subject. It contains
  - User's uniqueid and timestamp
  - Optional custom key information
  - Application server's JMX admin endpoint
- User then accesses another application server via Web
- WAS security searches for authentication information, using the SSO Token as the key, as follows
  - Local security cache for instantiated Subject
  - DynaCache for Tokens
  - JMX for Tokens

136 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

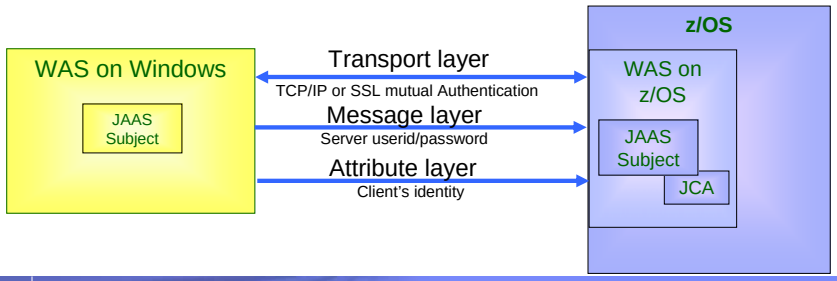
Part 3: WebSphere for z/OS back-end security solutions

137 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

CSIv2 Identity Assertion

- **Identity assertion** is a mechanism which allows the propagation of an already authenticated identity from one server to another. ID assertion is based on a **trust** relationship between the sender of the identity and the receiver. The receiver can assume that the identity has been authenticated, because he trusts the sender.
- CSIv2 allows client identity to be asserted to a downstream server .
  - This can be beneficial when there is no common authentication token that can be sent at the message layer for interoperability
- Trust is established between servers prior to sending the client identity
  - Client certificate authentication
  - Basic authentication



138 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

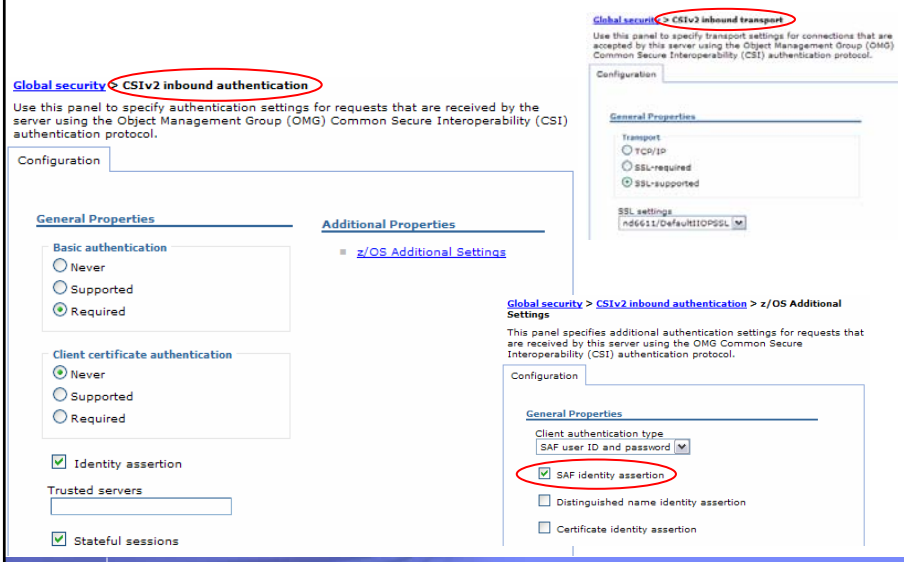
CSIv2 Identity Assertion mechanism

- Outbound server
  - authenticates to the Inbound server to establish trust:
    - **Client Certificate Authentication**
      - Outbound server's client certificate must be verifiable by the Inbound server (CA must be connected to the Inbound server's KeyRing)
      - Outbound server's certificate must be mapped to an identity in the Inbound server registry
    - **Basic Authentication**
      - Outbound server identity and password must be in the Inbound server's registry
  - provides the attribute layer asserted identity only (e.g. RACF ID, LDAP DN, certificate…)
- Inbound server
  - receives the outbound server identity and the asserted identity and does the following
    - Checks if the outbound server (from the outbound server identity) is in its list of trusted servers and if so, authenticates the upstream server. WebSphere for z/OS check if the outbound server identity has CONTROL access to the CBIND class.
    - **Accepts the asserted identity** and creates credentials by querying the registry - No validation is performed on asserted identity (no password, token, etc)
    - For Stateful server, this checking is done only once - subsequent requests are made through a session ID

139 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

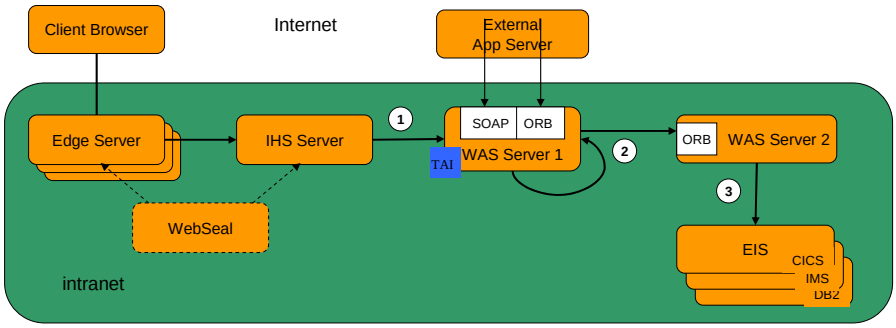
CSIv2 Identity Assertion configuration



140 © 2005 IBM Corporation ON DEMAND BUSINESS

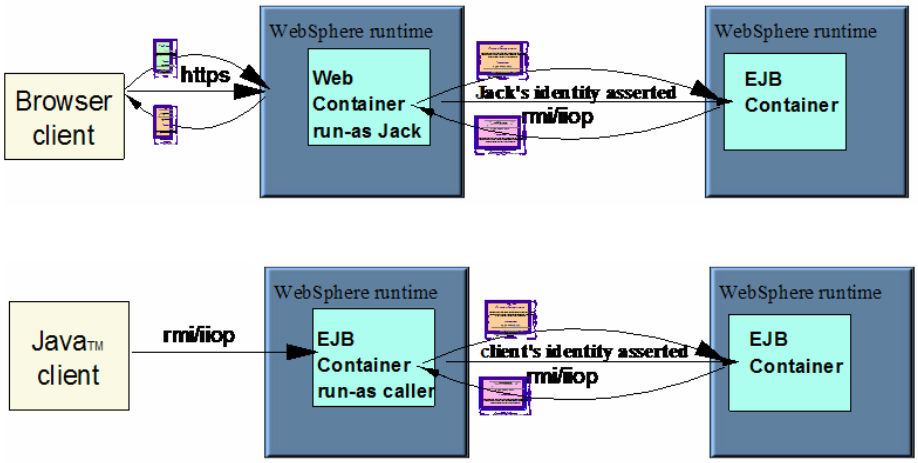
Solution: End-to-end security using Identity Assertion

1. Authorization or private headers
   - May be interpreted by TAI
   - TAI returns identity meaningful to configured user registry
2. CSIv2 identity assertion
3. Thread identity/thread security/JAAS authentication alias/JAAS custom mapping module



141 © 2005 IBM Corporation ON DEMAND BUSINESS

Solutions: More CSIv2 Identity Assertion examples



142 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

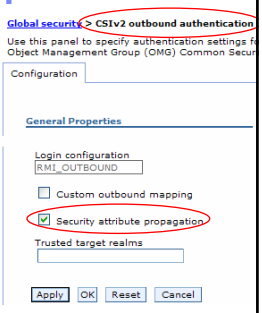
Part 3: WebSphere for z/OS back-end security solutions

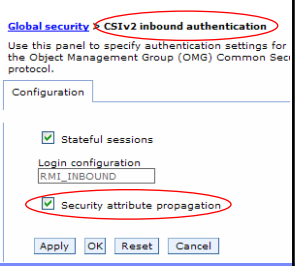
143 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Vertical Security Attribute Propagation

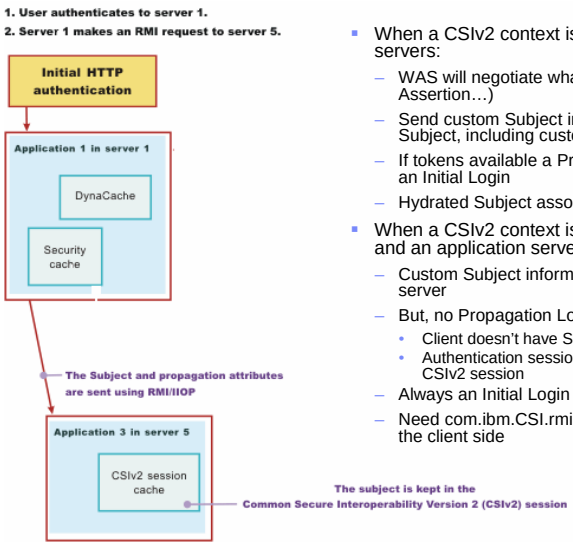
- Vertical security attribute propagation enables propagation of security attributes to downstream server
  - CSIv2 Identity Assertion propagates only the user name of the authenticated user
  - **Vertical security attribute propagation includes attributes such as user identity, authenticated Subject contents and security context information**
    - Alternatively, servers would have to query the User Registry or a custom login module to get the attributes – can be expensive from performance view point
    - Attributes might include original caller identity, location, IP address, dynamic group and so on
- Benefits:
  - Can eliminate the need for user registry calls to get the security attributes, at each remote hop along an invocation
  - Enables third-party providers to plug in custom tokens
  - Provides the ability to have multiple tokens of the same type within a Subject created by different providers





144 © 2005 IBM Corporation ON DEMAND BUSINESS

Downstream Propagation mechanism

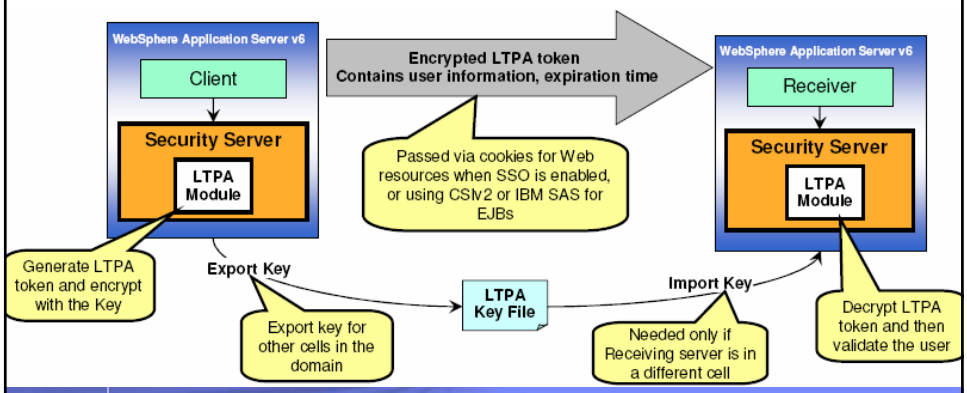


- When a CSIv2 context is established between application servers:
  - WAS will negotiate what is supported (CSIv2 Identity Assertion…)
  - Send custom Subject information and tokens from caller Subject, including custom tokens to downstream server
  - If tokens available a Propagation Login performed, otherwise an Initial Login
  - Hydrated Subject associated with CSIv2 session
- When a CSIv2 context is established between a J2EE client and an application server:
  - Custom Subject information is propagated from client to server
  - But, no Propagation Login
    - Client doesn't have SSO Token (exists only on server)
    - Authentication session between client and server is just the CSIv2 session
  - Always an Initial Login
  - Need com.ibm.CSI.rmiOutboundPropagationEnabled=true on the client side

145 © 2005 IBM Corporation ON DEMAND BUSINESS

LTPA Authentication Process and Calls

- Intended for distributed environments - Supports forwardable credentials and SSO
- When using LTPA, a token (called LTPA token) is generated with user information, an expiration time and is signed by the keys
- LTPA protocol uses cryptographic keys to perform data integrity (signing) and data confidentiality (encrypting) on user data, that passes between the servers



146 © 2005 IBM Corporation ON DEMAND BUSINESS

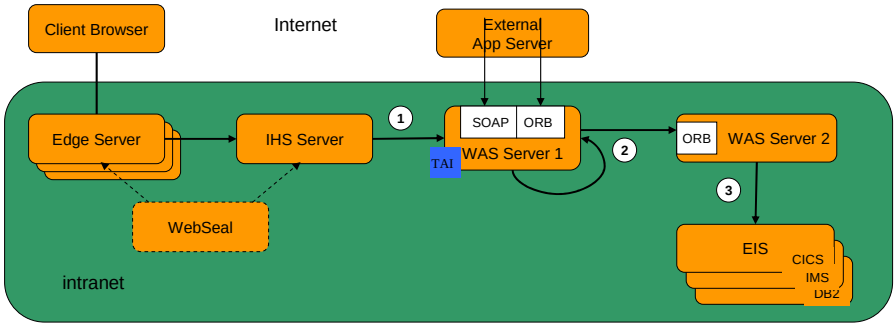
Security of Propagation Data

- If Subjects contain sensitive data, be careful
  - Network Security
    - DynaCache is replicated over network
      - Network transport is unencrypted by default
        - Of course, Authentication Token and SSO Token are inherently encrypted
      - Consider enabling DRS encryption
  - Application/Internal Security
    - Private cache is shared across entire application server (and replication domain)
    - WAS internal API provides cache access
    - Not accessible to application code *IF* Java 2 security is enabled

147 © 2005 IBM Corporation ON DEMAND BUSINESS

Solution: End-to-end security using LTPA token

1. Authorization or private headers
   - May be interpreted by TAI
   - TAI returns identity meaningful to configured user registry
2. LTPA token
3. Thread identity/thread security/JAAS authentication alias/JAAS custom mapping module



148 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

149 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

JAAS Login Modules and Login Configurations

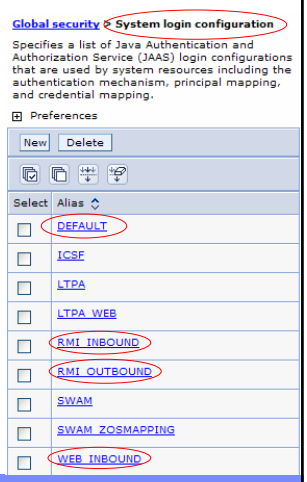
- **JAAS Login Modules**
  - Provide login function
  - Validate user identity, alter Subject, etc.
  - Standard implementation of javax.security.auth.spi.LoginModule
  - Custom login modules can
    - Affect authentication process before or after the WebSphere system login module
    - Make additional authentication decisions or add information to the Subject
- **JAAS Login Configurations**
  - Contain login modules
  - Login modules called in well defined order within configuration and can affect other login modules
  - Success or failure of login depends on login modules
  - Specific configurations are used in well-defined situations
  - Apply to all WAS authentication, not just web authentication

150 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WebSphere Login Configurations

- System login configurations
  - **WEB_INBOUND** – called when inbound web request needs authentication. Presence of SSO token implies authentication not needed. Called after any TAIs.
  - **RMI_INBOUND** – called when an inbound IIOP request needs to authenticate
  - **RMI_OUTBOUND** – called when an outbound IIOP request is being made. Can alter the outbound Subject information to handle foreign domains or unusual requirements.
  - **DEFAULT** – used when none of the above apply. E.g., default SOAP (admin authentication) and WAS internal authentication.
- Application login configurations
  - Applications use explicitly for authenticating on the server or from client
  - "Authentication" for special features, like web services and J2C identity mapping
  - Can add custom configurations

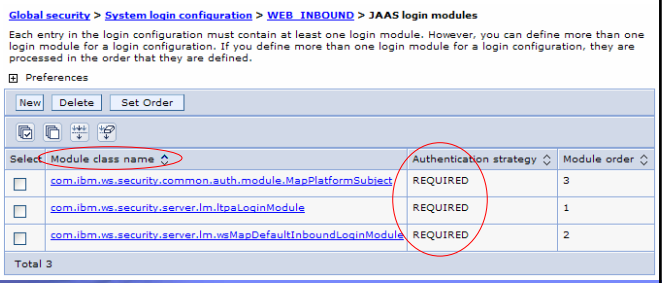


151 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

JAAS Login Module

- Classname – the login module to load
- Authentication Strategy
  - **REQUIRED** – Module must succeed or entire login will fail. You will most often use this.
  - **REQUISITE** – Module must succeed or entire login will fail. If it does fail, later modules will not be called.
  - **SUFFICIENT** – Module not critical to success, but if it succeeds, other login modules will be skipped. This will in most cases break WAS.
  - **OPTIONAL** – Module can succeed or fail without affecting login success or failure



152 © 2005 IBM Corporation ON DEMAND BUSINESS

JAAS Login Module Development – Key Concepts

- **Key Methods/Phases**
  - *Initialize*() – called when module first loaded. Options are passed from properties configured in WAS.
  - *Login*() – called when authentication required. Use callbacks to get at authentication data.
  - *Commit*() – called when authentication is successful. Store updates into Subject.
  - *Abort*() – called when authentication fails. Destroy any security information.
  - *Logout*() – called when authentication is to be destroyed. Remove stuff from Subject.
- **Shared state**
  - A Java Map scratch pad area where login modules can put stuff temporarily, instead of in Subject
  - Better to use shared state during login phase. Then, update Subject during commit.
- **Callback handler**
  - used by login module to get authentication information from the environment
  - WAS defines many handlers (refer to InfoCenter), such as
    - HTTP request/response handlers
    - Token handlers
    - Java standard required userid and password handlers
    - And more

153 © 2005 IBM Corporation ON DEMAND BUSINESS

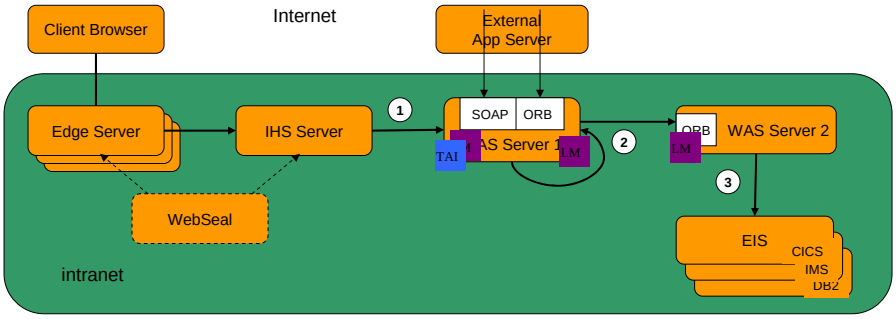
TAI versus Login Module

- The TAI interface, while proprietary to IBM, is easier to use
- A TAI can suppress the web login challenge (login modules can't)
- A TAI is called only once upon initial user login and won't be called again until the user's SSO Token expires, while a login module will be called
  - Whenever the user's credentials expire from the security cache (perhaps as often as every security cache timeout seconds)
  - When accessing another server for the first time
- A login module can handle web requests as well as RMI requests if configured into the appropriate configurations
- Opinion
  - Use TAI if goal is web authentication since simpler
  - Use login module if goal is RMI authentication
  - Use login module if you want common code for all types of authentication (Web, RMI, WebServices, Admin, etc.).

154 © 2005 IBM Corporation ON DEMAND BUSINESS

Solution: End-to-end security showing Login Modules

1. HTTP TAI and/or WEB_INBOUND login module builds Subject (rather than just returning an identity)
   - May add credential information (e.g. auditing data)
2. RMI_OUTBOUND login module propagates credential in token
3. RMI_INBOUND login module adds credential from token to subject



155 © 2005 IBM Corporation ON DEMAND BUSINESS

Providing SAF identity to WebSphere

- Synchronize registries (e.g. Windows, SAF) so that userid appears in both (LDAP Native Authentication can help)
- Use TAI to map non-SAF identity to SAF identity
  - Can be done through some lookup or by having the SAF identity as an attribute in the LDAP schema and passed as data in the HTTP request header
- Use WEB_INBOUND login configuration to map non-SAF identity to SAF identity (similar to TAI, but more "standard" and more difficult to code)
- Use RMI_INBOUND login configuration to map non-SAF identity to SAF identity (if first z/OS component is EJB)

156 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

157 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Where do Connectors Fit in J2EE?

- **Components**
  - These are Servlets, JSP's and EJB's with other Java classes as helpers and utilities. These are executable.
- **Containers**
  - This is where the components are executed. There are two types of containers, Web containers (where servlets and JSP's are executed) and EJB containers (where EJB's are executed). A container on MVS is not an address space.
- **Connectors**
  - These are the adapters a developer uses to get to databases and transactions. They include the CICS Transaction Gateway, IMS Connect, JDBC drivers, and JMS.

158 © 2005 IBM Corporation ON DEMAND BUSINESS

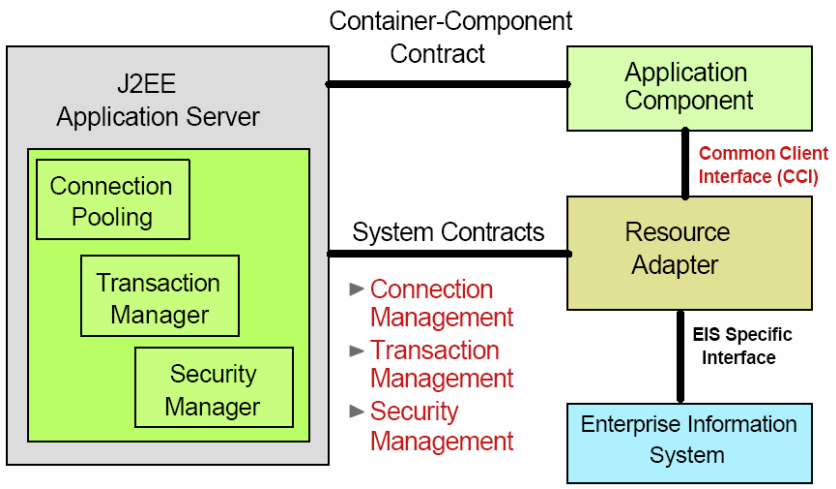
Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

159 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

J2EE Connector Architecture



160 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

J2EE Connector Security: Overview

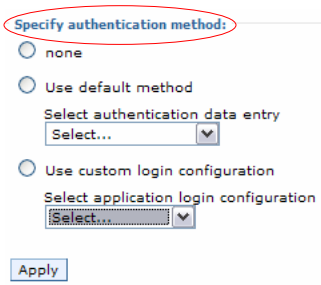
- J2EE components use "logical" names called resource references to refer to resource manager connection factories. A resource-ref element is added to the deployment descriptor which is scoped to the application component
- Indicate in res-auth element whether your application performs resource sign-on programmatically, or whether the container manages all authentication for this resource
  - **res-auth=Application**
  - **res-auth=Container**
- As of Version 6.0, resource authentication for *res-auth* settings of *Container* is preferably specified on the resource-reference mapping page.
  - Specification of container-managed authentication on a data source or connection factory is deprecated.

161 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

J2EE Connector Security Binding

- **res-auth=Application**: the authentication data is taken from, in order:
  - user id and password passed to getConnection(...)
  - component-managed auth alias on the Connection Factory or DataSource
  - Custom Properties UserName and Password on the DataSource
- **res-auth=Container**: the authentication data is taken from:
  - The authentication method defined at the "Map resource references to resources" panel
    - Authentication Data entry if any
    - DefaultPrincipalMapping Login Configuration by default and com.ibm.mapping.authDataAlias property
    - Custom Login Configuration such as one using the com.tivoli.pd.as.gso.AMPrincipalMapperLoginModule, which uses the GSO lockbox function in TAM server
  - Mapping-configuration and authentication alias defined at the Connection Factory panel, which is deprecated in V6.

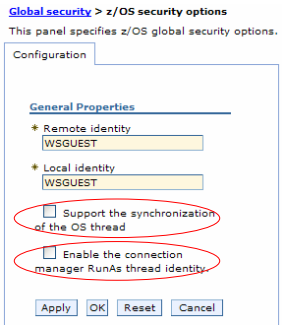


162 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Thread Identity versus Thread Security

- **Thread Identity support**
  - The ability to pass the identity of the Java principal within the Subject (J2EE security identity) through a JCA connector to an EIS
  - Thread identity support is enabled when:
    - A local connection is used between the application server and the EIS (CICS, IMS…)
    - Res-auth=Container is specified for the resource reference defined in the deployment descriptor
    - The connection factory does not specify a JAAS Authentication Alias
    - You are using an SAF-based user registry
- **Thread Security support**
  - Refers to the WebSphere on z/OS unique ability to allow the switching of the security context of servant region's Task Control Block (TCB level ACEE) to the current Java Thread Identity.
  - **Sync-to-OS Thread**:
    - Applies when the application requires access to system resources (e.g. HFS files, TCP/IP sockets, etc). The application request this function and the container will change the OS thread identity to the J2EE identity
  - **Connection Manager RunAs Identity**:
    - Applies to the connections to backend EIS such as DB2 and IMS JDBC. It allows the changing of the OS thread identity to the J2EE identity.
    - Thread security support for DB2 and IMS JDBC is enabled when:
      - Connection Manager RunAs Identity Enabled
      - A local connection is used between the application server and the EIS (CICS, IMS…)
      - Res-auth=Container is specified for the resource reference defined in the deployment descriptor
      - The connection factory does not specify a JAAS Authentication Alias
      - You are using an SAF-based user registry



163 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

JCA resource adapter and JDBC provider support

Connectors	Thread identity support	OS thread security
IMS Connector - local `ConnectionFactory` configuration	ALLOWED	Not supported
IMS Connector - remote `ConnectionFactory` configuration	NOTALLOWED	Not supported
CTG CICSECIConnector - local `ConnectionFactory` configuration	ALLOWED	Not supported
CTG CICSECIConnector - remote `ConnectionFactory` configuration	NOTALLOWED	Not supported
IMS JDBC Connector - local `ConnectionFactory` configuration (By default, IMS JDBC only supports this type of configuration.)	REQUIRED	True
RRA DB2 for z/OS local JDBC provider - data sources configured to the local DB2	ALLOWED	True
RRA DB2 Universal JDBC Driver Provider using Type 2 connectivity	ALLOWED	True
RRA DB2 Universal JDBC Driver Provider using Type 4 connectivity	NOTALLOWED	Not supported
WebSphere MQ JMS Provider: Connection Factory (TransportType = BINDINGS)	ALLOWED	True
WebSphere MQ JMS Provider - Connection Factory (TransportType = CLIENT)	NOTALLOWED	Not supported
WebSphere JMS Provider (such as Integral JMS Provider): Connection Factory	NOTALLOWED	Not supported

- The level of support can be:
  - **ALLOWED**, which indicates thread identity for connection ownership is allowed for this configuration.
  - **NOTALLOWED**, which indicates thread identity for connection ownership is not allowed for this configuration.
  - **REQUIRED**, which indicates thread identity for connection ownership is required.

164 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

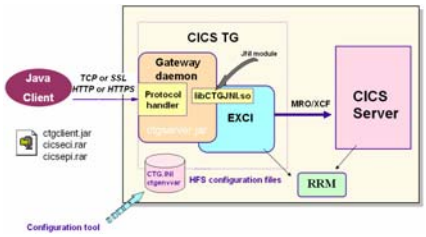
Part 3: WebSphere for z/OS back-end security solutions

165 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WebSphere and CICS Transaction Gateway

- Connectivity from WebSphere to CICS requires an additional component or gateway, ***CICS Transaction Gateway*** *(CTG),*
  - CTG converts a request from a Java client and forwards the request as a DPL request to a CICS region using either ECI or EXCI
  - CTG is a Java based application which can run as an OMVS task on z/OS, as a process on distributed platforms and within WebSphere on any platform when installed as a Resource Adapter
  - CTG on z/OS can authenticate a userid/password with the local SAF database (userids must have OMVS segments).
  - CTG on z/OS can authorize a user's access to a CICS region using SAF SURROGAT resources.
  - A CTG task on z/OS is trusted by a CICS region much more than a CTG process on a distributed platform.



166 © 2005 IBM Corporation ON DEMAND BUSINESS

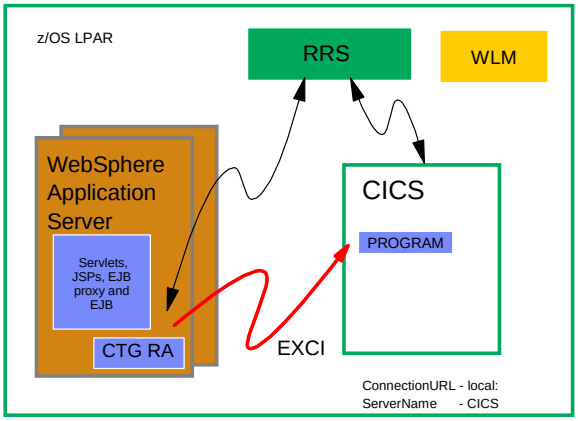
CICS Identity Projection Summary

- **Thread Identity** support for connection identity for **local** connections when using the default principal mapping module
- **Thread Security** not applicable
- **J2C reauthentication** supported
- **Identity assertion** of a WebSphere provided identity to CICS is possible
  - Requires WebSphere accessing CICS via a CTG on z/OS
  - Requires disabling CTG on z/OS userid/password authentication
  - Only when CICS attachment security from CTG on z/OS to CICS is set to ***Identify***
  - Challenge – If you disable CTG userid/password authentication how can you ensure some level of a trust between WebSphere and CTG or WebSphere and CICS?
    - Mutual Authentication of client certificates between WebSphere and CTG or CICS
    - SAF SURROGAT checking between WebSphere and CICS
    - SAF FACILITY checking of the EXCI pipe name between WebSphere and CICS
    - SAF FACILITY checking of the DFHAPPL name between WebSphere and CICS

167 © 2005 IBM Corporation ON DEMAND BUSINESS

WebSphere for z/OS accessing a local CTG

- CTG running in the WebSphere on z/OS address space can optionally verify an userid and password
- Attachment Security from an embedded CTG on z/OS to CICS can be either **Local** or **Identify**
- Identity assertion from WebSphere is available if attachment security is **Identify** and CTG on z/OS user authentication is not enabled
- Full 2PC support for CICS in a WebSphere global transaction

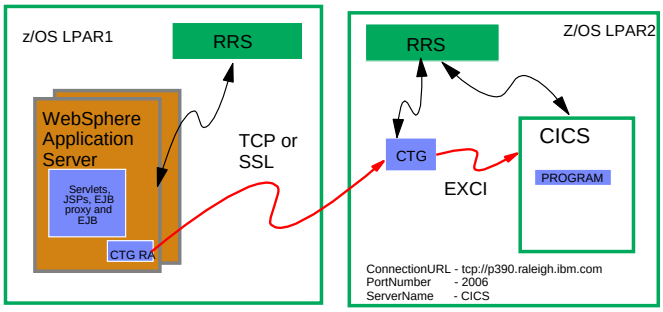


- CICS attachment security:
  - **Identify**: A user ID is flowed on every request, but no password is expected, because CICS trusts the user ID as having been already authenticated. The userid is either:
    - The user named in the ECIRequest object.
    - The user ID of the thread under which the ECI request runs.
  - **Local**: The CTG does not flow a user identifier; only the link user ID (if specified) is used. If no link user ID is supplied, all requests are run under the CICS default user ID.

168 © 2005 IBM Corporation ON DEMAND BUSINESS

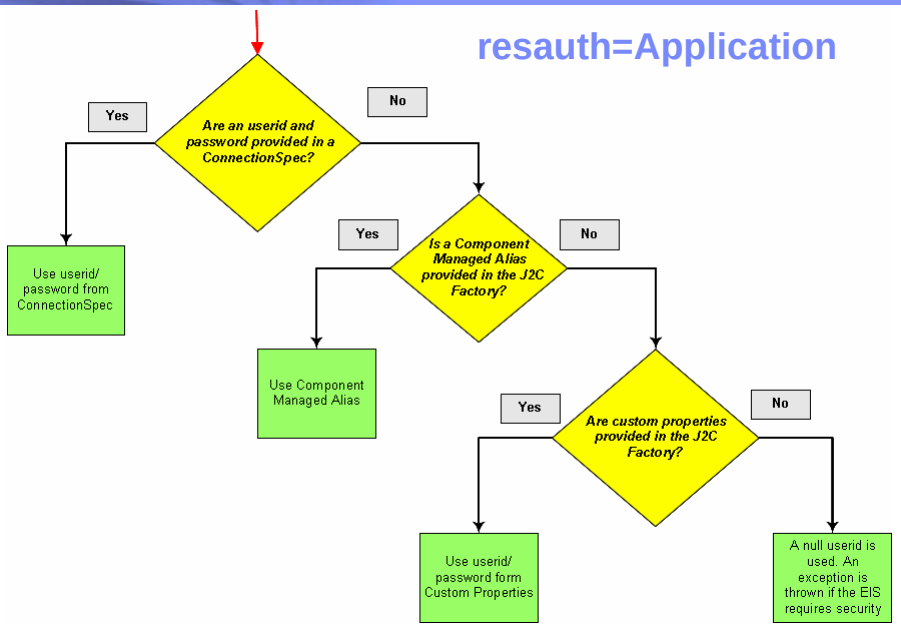
WebSphere for z/OS accessing a Remote CTG on z/OS

- TCP or SSL from WebSphere to CTG on z/OS
- With a SSL connection, access to CTG on z/OS can be controlled by Mutual Authentication of digital certificates.
- CTG on z/OS can optionally verify a userid and password with RACF
- Attachment Security from CTG on z/OS to CICS can either be **Local** or **Identify**
- Identity assertion from WebSphere is available if attachment security is **Identify** and CTG on z/OS user authentication is not enabled
- 1PC support but CICS can participate as a last participant in a WebSphere global transaction

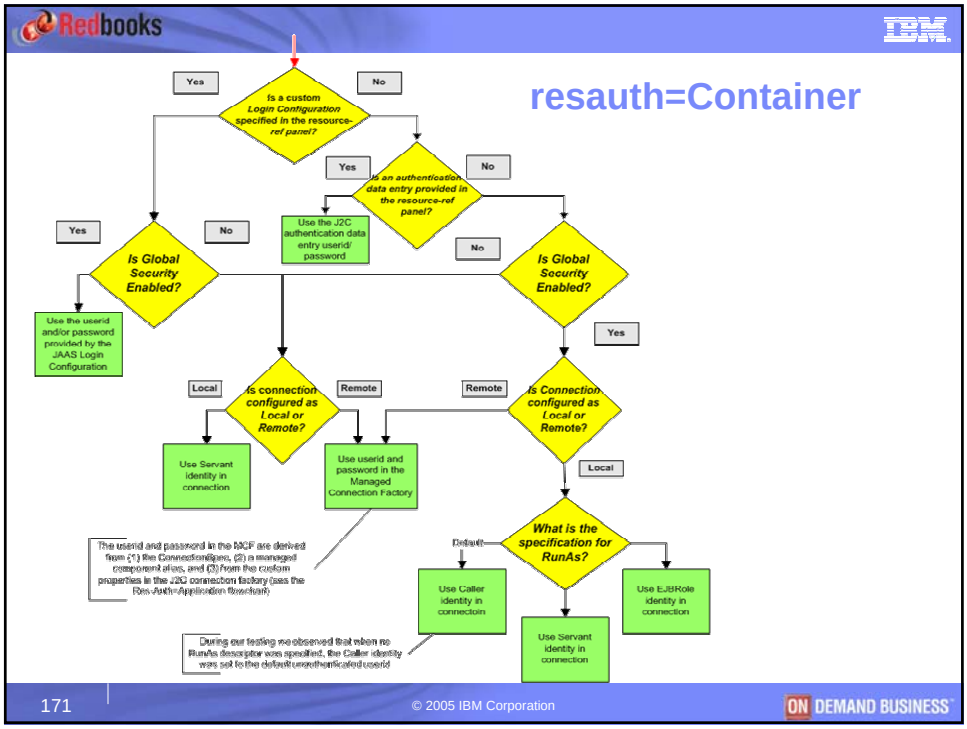


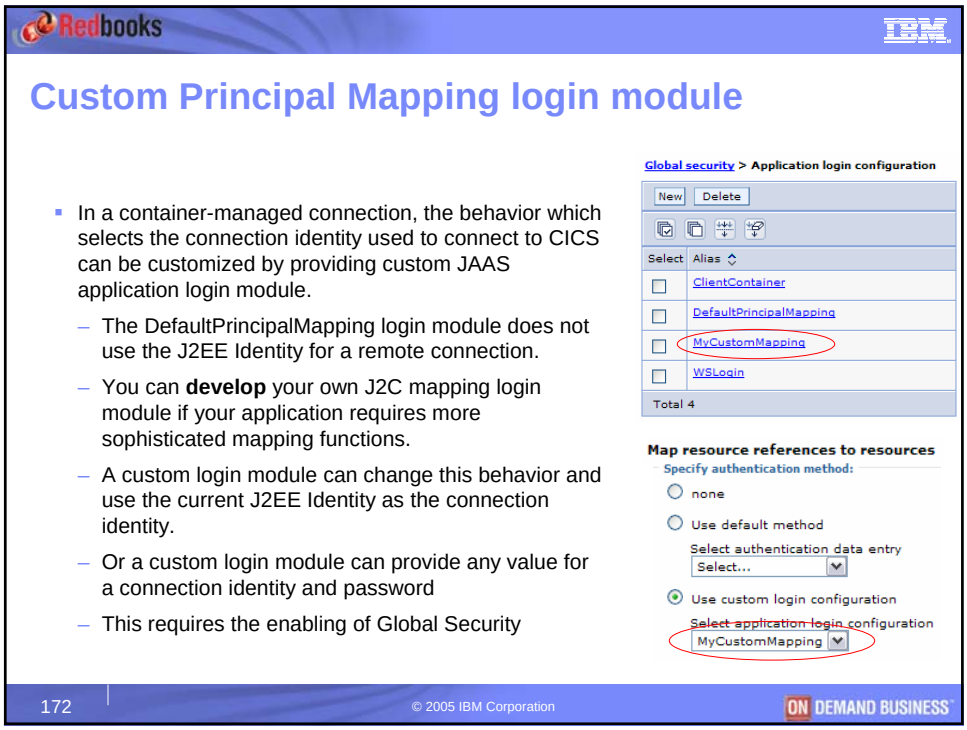
169 © 2005 IBM Corporation

resauth=Application



170 © 2005 IBM Corporation


Redbooks
IBM
resauth=Container
Is a custom Login Configuration specified in the resource-ref panel?
Yes
No
Is an authentication data entry provided in the resource-ref panel?
Yes
No
Use the J2C authentication data entry userid/ password
Is Global Security Enabled?
Yes
No
Use the userid and/or password provided by the JAAS Login Configuration
Is Global Security Enabled?
No
Yes
Is connection configured as Local or Remote?
Local
Remote
Use Servant identity in connection
Use userid and password in the Managed Connection Factory
Is Connection configured as Local or Remote?
Remote
Local
What is the specification for RunAs?
Default
Use Caller identity in connection
Use Servant identity in connection
Use EJBRole identity in connection
171
© 2005 IBM Corporation
ON DEMAND BUSINESS


Redbooks
IBM
Custom Principal Mapping login module
In a container-managed connection, the behavior which selects the connection identity used to connect to CICS can be customized by providing custom JAAS application login module.
The DefaultPrincipalMapping login module does not use the J2EE Identity for a remote connection.
You can develop your own J2C mapping login module if your application requires more sophisticated mapping functions.
A custom login module can change this behavior and use the current J2EE Identity as the connection identity.
Or a custom login module can provide any value for a connection identity and password
This requires the enabling of Global Security
Global security > Application login configuration
New
Delete
Select
Alias
ClientContainer
DefaultPrincipalMapping
MyCustomMapping
WSLogin
Total 4
Map resource references to resources
Specify authentication method:
none
Use default method
Select authentication data entry
Select...
Use custom login configuration
Select application login configuration
MyCustomMapping
172
© 2005 IBM Corporation
ON DEMAND BUSINESS

Redbooks IBM

CTG Supports J2C Reauthentication

J2C reauthentication reuses an existing connection in the pool

- When the connection identity changes
- Avoids the overhead of establishing a new connection when the J2EE identity changes

When a connection request is made to an EIS ***without J2C Reauthentication***, the container

1. Checks to see if a connection already exists in the pool
2. If a connection to the EIS is already in the pool with the same identity, the connection is reused.
3. Otherwise a new connection is created with all of the inherent overhead.

When a connection request is made with a **CICS J2C connection Factory**, the container

1. Checks to see if a connection for this factory already exists in the pool
2. If a connection to the EIS is already in the pool and is available *regardless* of the connection's identity, the connection to CICS is reused.
3. Otherwise a new connection to CICS is created with all of the inherent overhead.

J2C reauthentication requires container managed security and a sharing scope of sharable.

173 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

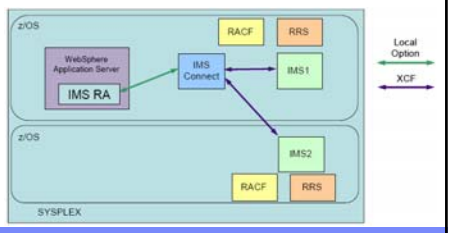
Part 3: WebSphere for z/OS back-end security solutions

174 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WebSphere and IMS

- Connectivity from WebSphere to IMS requires gateway components:
  - **IMS Connector for Java** (IC4J) is a Java application which runs within WebSphere on any platform and installed as a Resource Adapter
  - **IMS Connect** converts a request from IC4J and forwards the request to IMS using IMS's Open Transaction Management Architecture (OTMA) format.
    - IMS Connect is native z/OS application which listens on TCP/IP ports for remote request over TCP/IP and XCF for local z/OS request.
    - IMS Connect can authenticate an userid/password with the local SAF database
    - IMS Connect can authorize a local user's access to IMS Connect using SAF FACILITY resources.



175 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

IMS Identity Projection Summary

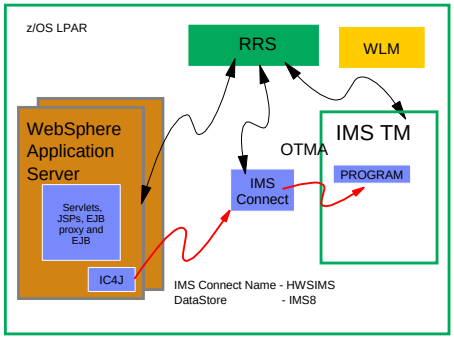
- **Thread Identity** support for connection identity for **local** connections using default principal mapping module
- **Thread Security** not applicable
- **J2C reauthentication** supported
- **Identity assertion** of a WebSphere provided identity to IMS is possible (two options)
  - Requires disabling all IMS Connect authentication checking (RACF=N). or
  - Requires the use of an IMS Connect trusted user exit which can bypass authentication checks for trusted clients
  - Challenge – If you disable IMS Connect authentication either totally or selectively, how can you ensure some level of trust between WebSphere and IMS Connect?
    - Mutual Authentication of client certificates between WebSphere and a remote IMS Connect
    - SAF FACILITY checking of access between WebSphere on z/OS and a local IMS Connect
    - Include specific information in the request which an IMS Connect trusted user exit can verify and use to bypass further authentication of the request.

N.B If authentication has not been disabled or bypassed, IMS Connect will try to authenticate a request with a SAF regardless of the level of trust between WebSphere and IMS Connect

176 © 2005 IBM Corporation ON DEMAND BUSINESS

WebSphere for z/OS accessing a local IMS Connect

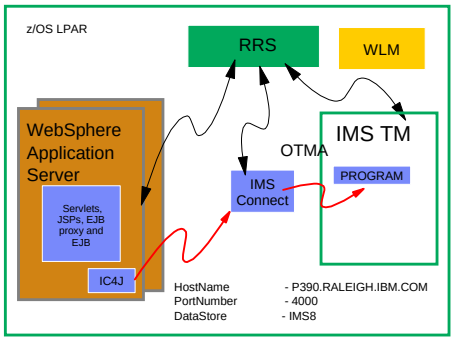
- IMS Connector for Java (IC4J) accesses IMS Connect task via XCF facilities
- Access to IMS Connect from WebSphere can be controlled by RACF FACILITY resource
- Access to IMS TM from IMS Connect can be also controlled by RACF FACILITY resources
- Full 2PC support for IMS in a WebSphere global transaction



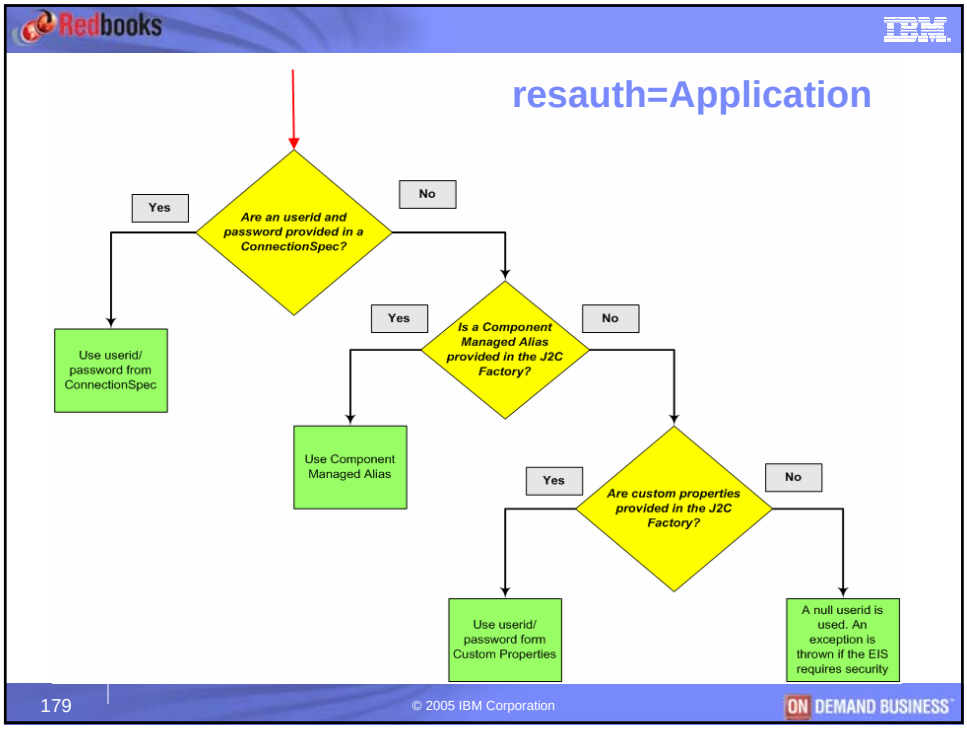
177 © 2005 IBM Corporation ON DEMAND BUSINESS

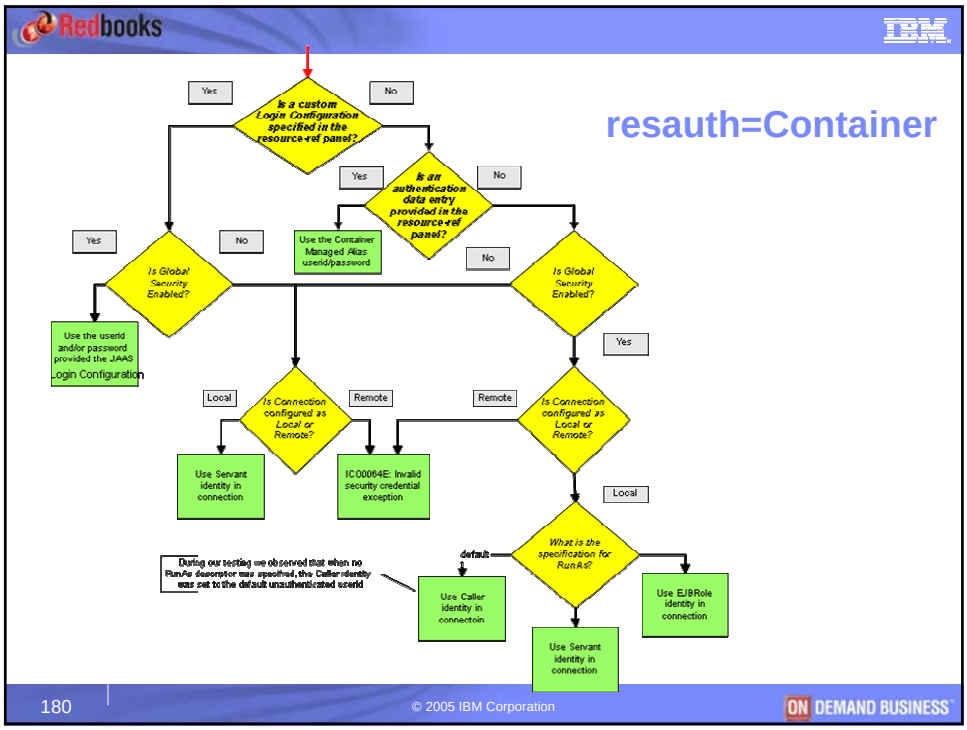
WebSphere for z/OS accessing a remote IMS Connect

- IMS Connector for Java (IC4J) accesses IMS Connect task via TCP or SSL
- With a SSL connection, access to IMS Connect can be controlled by Mutual Authentication of client certificates
- Access to IMS TM from IMS Connect can be also controlled by RACF FACILITY resources
- Full 2PC support for IMS in a WebSphere global transaction



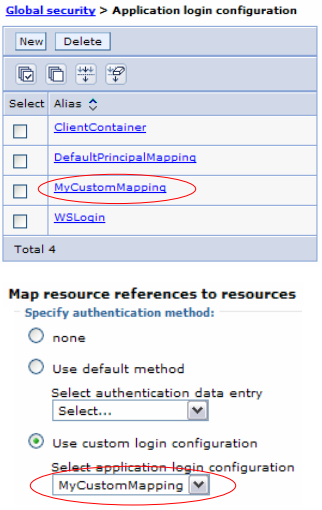
178 © 2005 IBM Corporation ON DEMAND BUSINESS


Redbooks
IBM
resauth=Application
Are an userid and password provided in a ConnectionSpec?
Yes
No
Use userid/ password from ConnectionSpec
Is a Component Managed Alias provided in the J2C Factory?
Yes
No
Use Component Managed Alias
Are custom properties provided in the J2C Factory?
Yes
No
Use userid/ password form Custom Properties
A null userid is used. An exception is thrown if the EIS requires security
179
© 2005 IBM Corporation
ON DEMAND BUSINESS


Redbooks
IBM
resauth=Container
Is a custom Login Configuration specified in the resource-ref panel?
Yes
No
Is an authentication data entry provided in the resource-ref panel?
Yes
No
Use the Container Managed Alias userid/password
Is Global Security Enabled?
Yes
No
Use the userid and/or password provided the JAAS Login Configuration
Is Global Security Enabled?
Yes
Is Connection configured as Local or Remote?
Local
Remote
Use Servant identity in connection
IC00064E: Invalid security credential exception
Is Connection configured as Local or Remote?
Remote
Local
What is the specification for RunAs?
default
Use Caller identity in connectoin
Use EJBRole identity in connection
Use Servant identity in connection
During our testing we observed that when no RunAs descriptor was specified, the Caller identity was set to the default unauthenticated userid
180
© 2005 IBM Corporation
ON DEMAND BUSINESS

Custom Principal Mapping login module

- In a container-managed connection, the behavior which selects the connection identity used to connect to IMS can be customized by providing custom JAAS application login module.
  - The DefaultPrincipalMapping login module does not use the J2EE Identity for a remote connection.
  - You can **develop** your own J2C mapping login module if your application requires more sophisticated mapping functions.
  - A custom login module can change this behavior and use the current J2EE Identity as the connection identity.
  - Or a custom login module can provide any value for a connection identity and password
  - This requires the enabling of Global Security



181 © 2005 IBM Corporation ON DEMAND BUSINESS

IMS Supports J2C Reauthentication

J2C reauthentication reuses an existing connection in the pool
- When the connection identity changes
- Avoids the overhead of establishing a new connection when the J2EE identity changes

When a connection request is made to an EIS ***without J2C Reauthentication***, the container
1. Checks to see if a connection already exists in the pool
2. If a connection to the EIS is already in the pool with the same identity, the connection is reused.
3. Otherwise a new connection is created with all of the inherent overhead.

When a connection request is made with a **IMS J2C connection Factory**, the container
1. Checks to see if a connection for this factory already exists in the pool
2. If a connection to the EIS is already in the pool and is available *regardless* of the connection's identity, the connection IMS is reused.
3. Otherwise a new connection to IMS is created with all of the inherent overhead.

J2C reauthentication requires container managed security and a sharing scope of sharable.

182 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

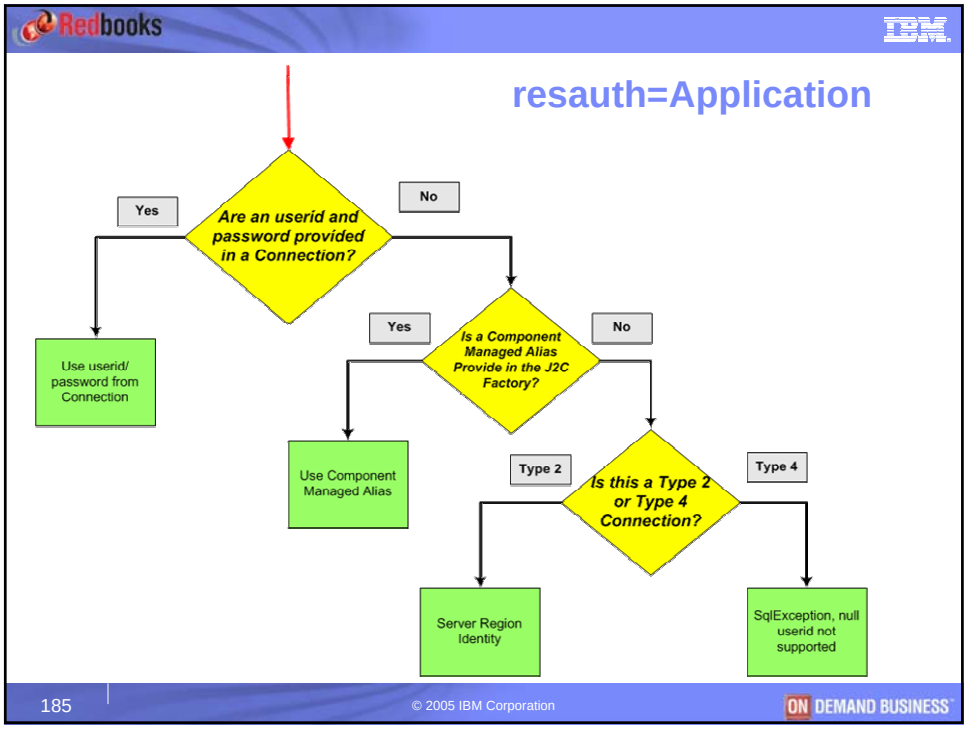
183 © 2005 IBM Corporation ON DEMAND BUSINESS

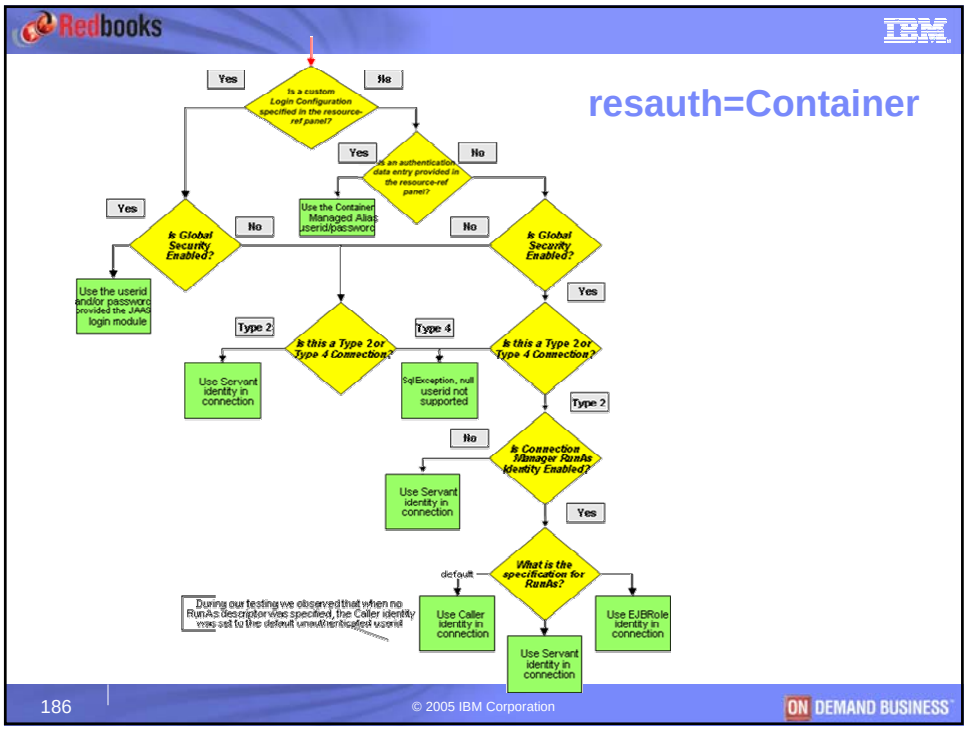
Redbooks IBM

DB2 Identity Projection Summary

- **Thread Identity** support for connection identity for **local** connections using default principle mapping module
- **Thread Security** support for connection identity for **local** connections
- **J2C reauthentication** – WebSphere v6 supports database reauthentication when a custom DataStoreHelper is provided.
- **Identity assertion** – no
  - resetDB2Connection(user,pw) method is used to switch identity of a connection and it requires userid and password.

184 © 2005 IBM Corporation ON DEMAND BUSINESS

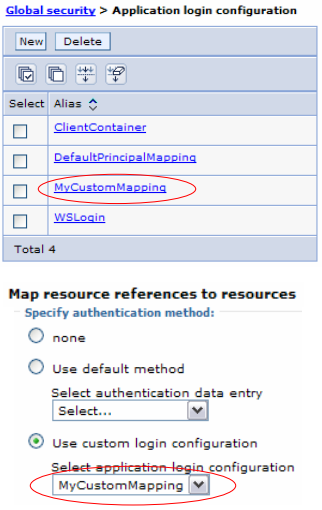

Redbooks
IBM
resauth=Application
Are an userid and password provided in a Connection?
Yes
No
Use userid/ password from Connection
Is a Component Managed Alias Provide in the J2C Factory?
Yes
No
Use Component Managed Alias
Is this a Type 2 or Type 4 Connection?
Type 2
Type 4
Server Region Identity
SqlException, null userid not supported
185
© 2005 IBM Corporation
ON DEMAND BUSINESS


Redbooks
IBM
resauth=Container
Is a custom Login Configuration specified in the resource-ref panel?
Yes
No
Is an authentication data entry provided in the resource-ref panel?
Yes
No
Use the Container Managed Alias userid/password
Is Global Security Enabled?
Yes
No
Use the userid and/or password provided the JAAS login module
Is Global Security Enabled?
No
Yes
Is this a Type 2 or Type 4 Connection?
Type 2
Type 4
Use Servant identity in connection
SqlException, null userid not supported
Is this a Type 2 or Type 4 Connection?
Type 2
Is Connection Manager RunAs Identity Enabled?
No
Yes
Use Servant identity in connection
What is the specification for RunAs?
default
Use Caller identity in connection
Use Servant identity in connection
Use EJBRole identity in connection
186
© 2005 IBM Corporation
ON DEMAND BUSINESS

Redbooks IBM

Custom Principal Mapping login module

- In a container-managed connection, the behavior which selects the connection identity used to connect to DB2 can be customized by providing custom JAAS application login module.
  - The DefaultPrincipalMapping login module does not use the J2EE Identity for a remote connection.
  - You can **develop** your own J2C mapping login module if your application requires more sophisticated mapping functions.
  - A custom login module can change this behavior and use the current J2EE Identity as the connection identity.
  - Or a custom login module can provide any value for a connection identity and password
  - This requires the enabling of Global Security

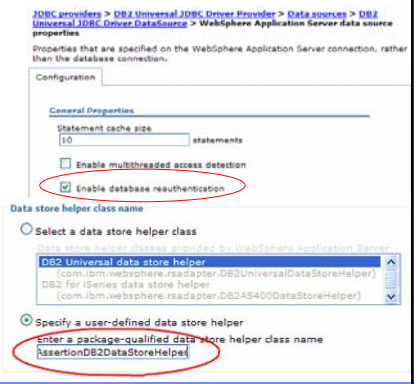


187 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

J2C reauthentication using a DataStoreHelper

- In WebSphere v6, the RelationalResourceAdapter that is used for all relational database access has been enhanced to support **reauthentication**.
  - Reauthentication avoids the overhead of establishing a new connection when the J2EE identity changes.
- Configuration:
  - Configure resource-reference res-auth=**Container**
  - **Develop** a custom DataStoreHelper with methods that alter the database connection identity information.
  - Configure the DataSource to use **Reauthentication** and to use the custom **DataStoreHelper**.
- When the maximum number of connections is reached and a new request for a connection with a new identity comes in to the connection pool, the pool manager selects any of the connections in its pool and call the **doConnectionSetupPerTransaction**() DatastoreHelper method.
- DB2 V8 APAR PQ99707 on z/OS includes performance enhancements to make reauthentication substantially more efficient.
- For more information see developerWorks article "Database identity propagation in WAS v6"
  - http://www.ibm.com/developerworks/websphere/techjournal/0506_barghouthi/0506_barghouthi.html



188 © 2005 IBM Corporation ON DEMAND BUSINESS

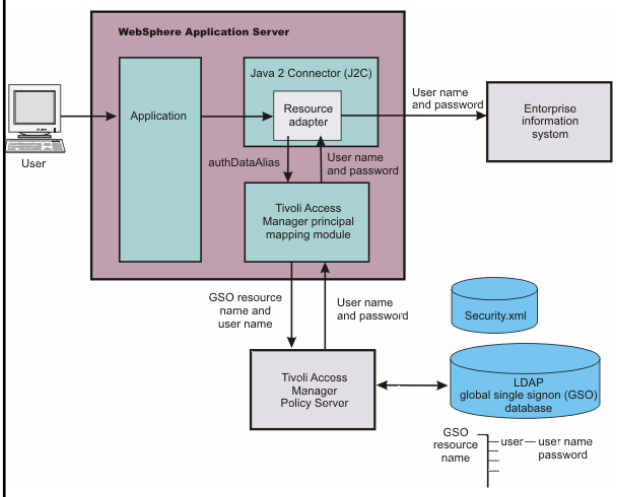
Redbooks IBM

Part 3: WebSphere for z/OS back-end security solutions

189 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

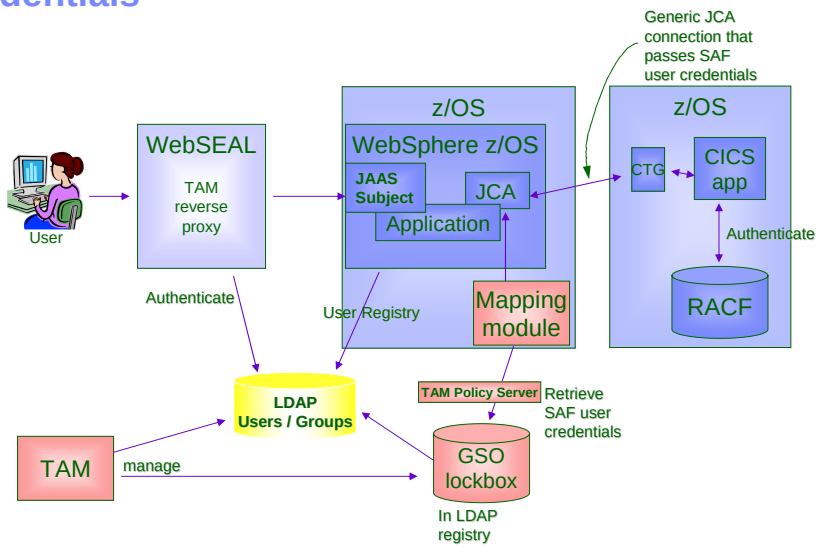
TAM Global Sign On Principal Mapping



- User credentials retrieved from **TAM GSO lockbox** and associated with JCA request
- Configuration:
  - Requires TAM custom Principal Mapping Module for JCA
  - Configure each user as a GSO user in TAM
  - Set the GSO resource to match that of the JCA resource
  - Configure userid and password appropriate for JCA EIS

190 © 2005 IBM Corporation ON DEMAND BUSINESS

Solution: End-to-end propagation of User credentials



© 2005 IBM Corporation

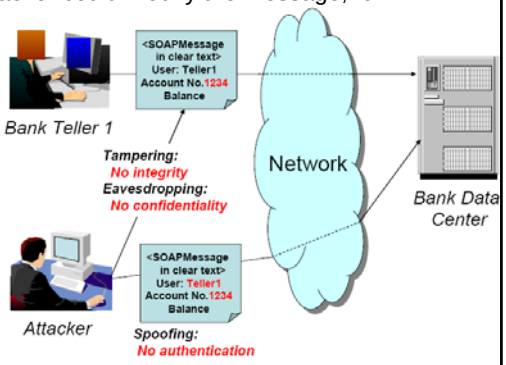
Part 3: WebSphere for z/OS back-end security solutions

© 2005 IBM Corporation

Redbooks IBM

Web Services security exposure

- **Spoofing, no authentication**: An attacker could send a modified Simple Object Access Protocol (SOAP) message to the service provider, pretending to be a bank teller, to obtain confidential information, or to withdraw money from another customer's account.
- **Tampering, no integrity**: The SOAP message is intercepted between the Web service requester and provider. An attacker could modify the message, for example, to deposit the money into another account by changing the account number.
- **Eavesdropping, no confidentiality**: Without data encryption, SOAP messages are sent in clear text, and the information contained in the message can be intercepted or read by an attacker. Confidential customer or bank information can get into the wrong hands.

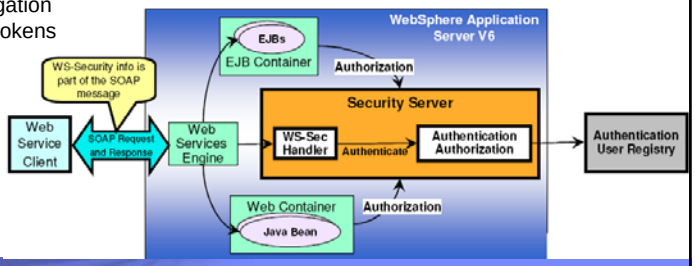


193 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WS-Security: Overview

- WS-Security is a message level standard that defines how to secure SOAP messages, using
  - XML Digital Signature:
    - Digitally sign the SOAP XML document, providing integrity, authenticity, and signer authentication - JSR 105 to address this programmatically
  - XML Encryption:
    - Process for encrypting data and representing the result in XML providing confidentiality – JSR 106 to address this programmatically
  - XML Canonicalization:
    - Provides normalized XML document that can be digitally signed and verified
- Credential propagation through security tokens

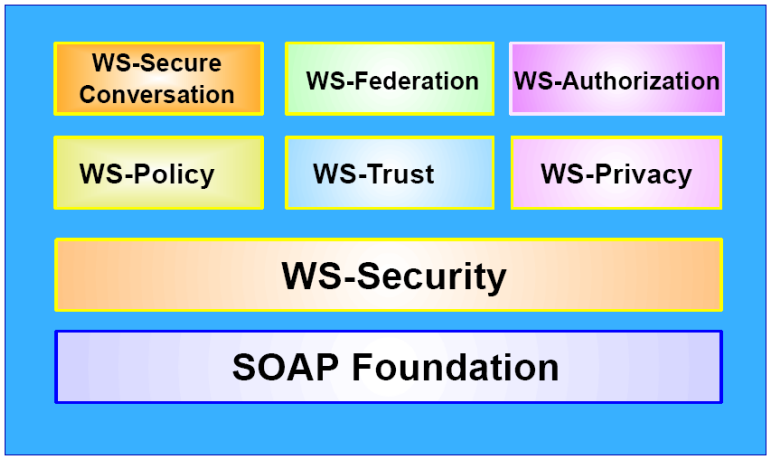


194 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WS-Security specifications

- WS-Security defines a set of extensions to the SOAP standards



195 © 2005 IBM Corporation ON DEMAND BUSINESS

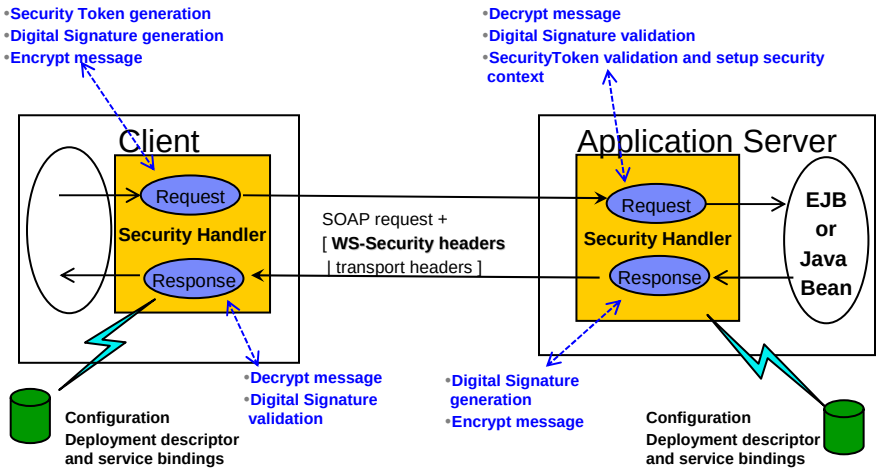
Redbooks IBM

WS-Security specifications

- **WS-Policy**: Addresses the capabilities and constraints of security or business policies on intermediaries and endpoints, for instance, required security tokens, supported encryption algorithms, and privacy rules
- **WS-Trust**: Describes a framework for trust models that enable Web services to securely interoperate
- **WS-Privacy**: Describes a model for how Web service providers and requestors state privacy preferences and organizational privacy practice statements
- **WS-Secure Conversation**: Describes how to manage and authenticate message exchanges between parties, including security context exchange and establishing and deriving session keys
- **WS-Federation**: Describes how to manage and broker the trust relationships in a heterogeneous federated environment, including support for federated identities
- **WS-Authorization**: Describes how to manage authorization data and authorization policies

196 © 2005 IBM Corporation ON DEMAND BUSINESS

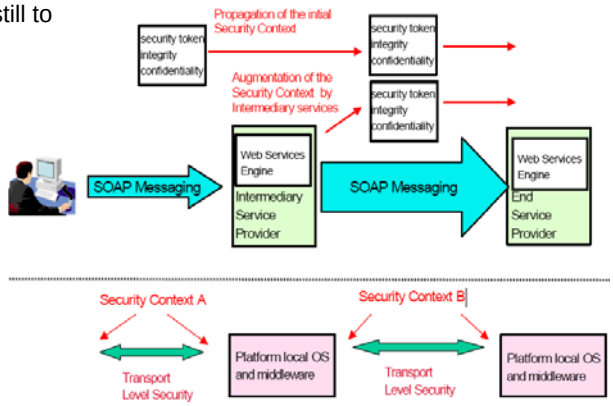
WS-Security High Level Architecture



197 © 2005 IBM Corporation ON DEMAND BUSINESS

Web Services security protocol layers

- Web services messaging relies on two protocol layers. Security can be implemented at each of these layers:
  - **The SOAP or Message layer**: The WS-Security specifications indicate how SOAP XML messages can carry security assertions and contexts. Some of these specifications are still to be finalized.
  - **The Transport layer**: HTTP, RMI/IIOP, WebSphere MQ, and so on typically carry authentication information in headers, with optional additional security provided by encapsulation in the SSL/TLS protocol.



198 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Web Services Transport layer security

- SSL is the most popular way to encrypt communication between business partners over the Internet.
- It simply creates a secure pipeline between two nodes and encrypts all traffic flowing between the nodes.
  - SSL provides a straightforward way to provide **confidentiality**.
  - It also includes a built-in communication **integrity** check.
  - Connection layer **authentication** is achieved by the client always authenticating the server, and optimally being authenticated by the server, through the exchange of X.509 certificates.
- HTTPS (SSL over HTTP) has the following advantages:
  - It can be used to provide a very fast and secure transport for Web services.
  - It provides authentication through either HTTP Basic Authentication or a client X.509 certificate.
  - It provides integrity between the client and server by using asymmetric key cryptography to establish authenticity of server and client and to securely share a secret key.
  - It provides confidentiality between the client and server through efficient shared key cryptography.
  - It has good support for a broad array of hardware accelerators.
  - It is mature and similarly implemented by most vendors, and therefore, is subject to few interoperability problems.
- JMS: SSL can be used between messaging engines.

199 © 2005 IBM Corporation ON DEMAND BUSINESS

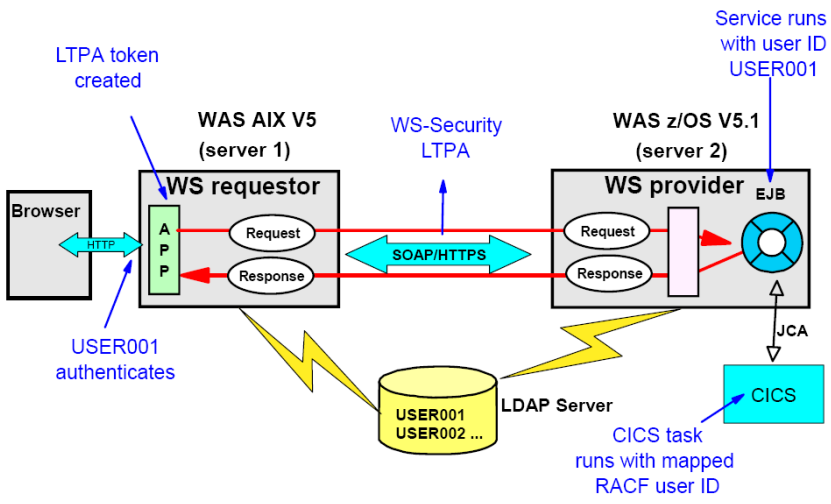
Redbooks IBM

Web Services Message level security

- WS-Security provides a general purpose mechanism for associating security tokens with messages.
  - Typical tokens in WebSphere-based Web services are user name and password, X.509 certificates, and LTPA tokens.
- WS-Security supports the following authentication mechanisms via the insertion of a security token:
  - **Basic Authentication**: The security token includes the user name and password information, and is generated as <wsse:UsernameToken> with <wsse:Username> and <wsse:Password>.
  - **Signature**: The security token includes the X.509 certificate of the signer of the data and is generated as <ds:Signature> with <wsse:BinarySecurityToken>.
  - **ID assertion**: ID assertion includes a user name only, since the identity is asserted, and is generated as <wsse:UsernameToken> with <wsse:Username>.
  - **Custom**: This mechanism includes a custom-defined token.
  - **LTPA**: Use of an LTPA token is a WebSphere-specific customer token, generating a <wsse:UsernameToken> with <wsse:Username>

200 © 2005 IBM Corporation ON DEMAND BUSINESS

Web Services solution 1: Direct access with LTPA authentication



© 2005 IBM Corporation

Web Services solution 1: Direct access with LTPA authentication

1. Server 1: USER001 authenticates when invoking the Web application
2. Server 1: The Callback Handler extracts the LTPA token from the current JAAS Subject. The Web services security runtime inserts the extracted token into the SOAP header
3. Server 1: The Web service requester invokes a Web service on server 2 using the LTPA authentication method.
4. Server 2: The JAAS login configuration validates the LTPA token and sets it as the current JAAS Subject for the EJB application. This enables the EJB application to run with the user ID USER001; all methods of the EJB application are defined with RunAs set to Caller.
5. Server 2: The EJB application uses the ECI resource adapter to make a call to the target CICS COBOL application. A JAAS authentication alias is specified on the connection factory, so the user ID that flowed to CICS is the RACF user ID that is associated with that JAAS alias.
6. CICS: A Link user ID is preset on the CICS EXCI connection. The CICS COBOL program runs under the transaction ID specified by the TPNName property of the ECIInteractionSpec. All CICS resource authorization checks, including transaction authorizations, are performed against both the flowed and Link user IDs.

Trust is established between the WAS profiles using HTTPS and by use of LTPA token authentication. Both servers are configured with the same LTPA key, with the same user registry, have single signon enabled.

© 2005 IBM Corporation

Redbooks IBM

Web Services solution 1: Direct access with LTPA authentication Configuration

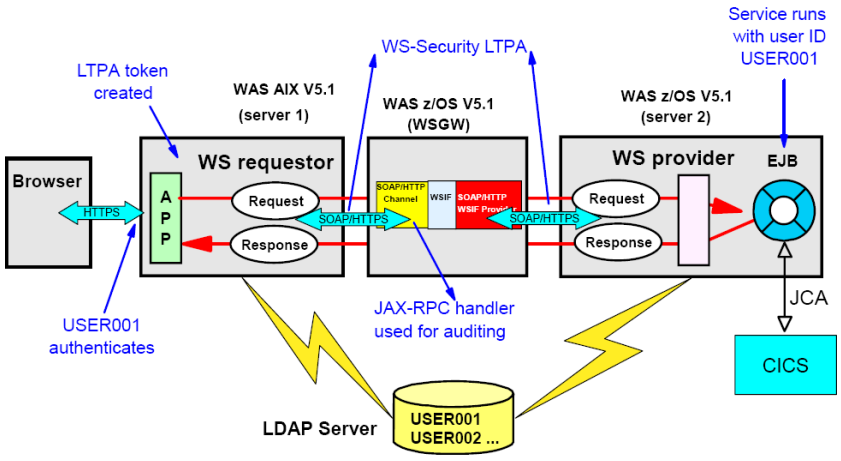
Version 5.x application

- Web service requester Server1 **login binding**:
  - Authentication method: LTPA
  - Token local name: LTPA
  - Token value type URI: http://www.ibm.com/websphere/appserver/totentype/5.0.2
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.LTPATokenCallbackHandler
- Web service provider Server2 **login mapping**:
  - Authentication method: LTPA
  - Configuration name: WSLogin
  - Token local name: LTPA
  - Token value type URI: http://www.ibm.com/websphere/appserver/totentype/5.0.2
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.WSCallbackHandlerFactoryImpl

203 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Web Services solution 2: Indirect access with WSGW and LTPA authentication



204 © 2005 IBM Corporation ON DEMAND BUSINESS

Web Services solution 2: Indirect access with WSGW and LTPA authentication

- Security advantages of using the Web Services Gateway:
  - Web service requestors do not need to know the network location of the target services.
  - The gateway can be used as a single point of control for defining service security settings.
  - The gateway provides a central point for logging and auditing.
- Web Services Gateway configuration:
  - The **service security** settings consist of service security configuration and the target service security configuration. Each gateway service is individually configured with its own service security configuration.
  - The **login mapping** specifies the configuration for validating security tokens within incoming messages coming from service requestors.
  - The **login binding** specifies the configuration for generating security tokens within outgoing messages that are sent to service providers.

Version 5.x application

205 © 2005 IBM Corporation ON DEMAND BUSINESS

Web Services solution 2: Indirect access with WSGW and LTPA authentication

1. Server 1: Same as scenario 1.
2. Server1: The Web service requester invokes a Web service on server 2 using the LTPA authentication method.
3. WSGW: A gateway JAX-RPC handler can interact with messages as they pass between the service requester and the gateway, and between the gateway and the target service. For example the handler can print SOAP headers to a message log for auditing purposes.
4. WSGW: The JAAS login configuration validates the LTPA token and sets it as the current JAAS Subject for the WSGW application. The Callback Handler extracts the LTPA token from the current JAAS subject in the WSGW application and the Web services security runtime inserts the token in the SOAP header of the message that is sent to the target service on server 2.
5. Server 2: The JAAS login configuration validates the LTPA token and sets it as the current JAAS Subject for the EJB application. This enables the EJB application to run with the user ID USER001; all methods of the EJB application are defined with RunAs set to Caller.
6. Server 2: Same as scenario 1.

The gateway server is configured with the same LTPA key as the other application servers, has single signon enabled, and uses the same user registry.

206 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Web Services solution 2 Configuration

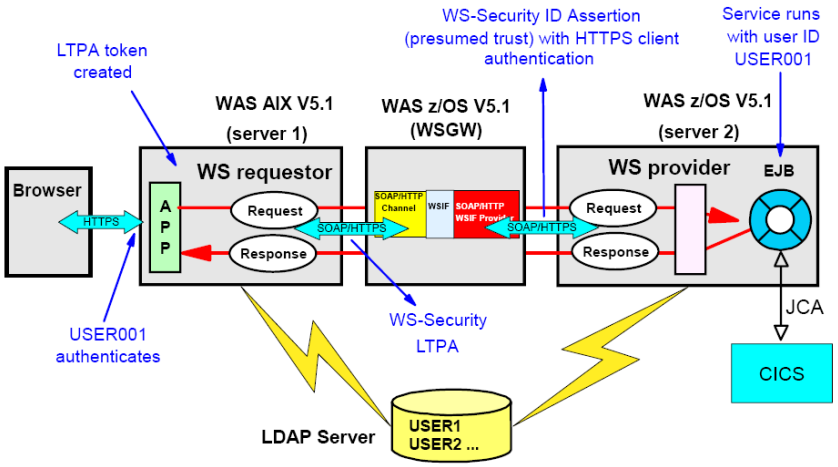
Version 5.x application

- Web service requester Server1 **login binding**:
  - Authentication method: LTPA
  - Token local name: LTPA
  - Token value type URI: http://www.ibm.com/websphere/appserver/totentype/5.0.2
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.LTPATokenCallbackHandler
- WSWG **login mapping**:
  - Authentication method: LTPA
  - Configuration name: WSLogin
  - Token local name: LTPA
  - Token value type URI: http://www.ibm.com/websphere/appserver/totentype/5.0.2
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.WSCallbackHandlerFactoryImpl
- WSWG **login binding**:
  - Authentication method: LTPA
  - Token local name: LTPA
  - Token value type URI: http://www.ibm.com/websphere/appserver/totentype/5.0.2
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.LTPATokenCallbackHandler
- Web service provider Server2 **login mapping**:
  - Authentication method: LTPA
  - Configuration name: WSLogin
  - Token local name: LTPA
  - Token value type URI: http://www.ibm.com/websphere/appserver/totentype/5.0.2
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.WSCallbackHandlerFactoryImpl

207 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Web Services solution 3: Indirect access with WSGW and ID assertion



208 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Web Services solution 3: Indirect access with WSGW and ID assertion

- Why identity assertion?
  - For target service authentication method to be independent of how the service requestor authenticates with the gateway.
  - For example when LTPA authentication is not available on the WS Requestor side.
  - When using ID assertion, the gateway invokes a service from a downstream server on behalf of the client, but does not pass on the client authentication information.
- WS-Security supports the following trust modes with a downstream server:
  - **Basic Authentication**: The asserting server authenticates itself, sending a user name and password in the SOAP header, in addition to the transmitted asserted identity.
  - **Digital signature**: The asserted identity is transmitted digitally signed by the asserting server, along with the asserting server x.509 certificate. This provides both for data integrity of the transmitted asserted identity and for authentication of the sender.
  - **Presumed trust**: In this case, communication flows inside a secure channel or uses a secure transport protocol so that the asserting server does not need to provide authentication data at the SOAP message level. Typically this can be achieved using HTTP as the transport protocol with SSL/TLS client (the client is the asserting server) authentication.

209 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Web Services solution 3: Indirect access with WSGW, ID assertion and presumed trust

1. Server 1: Same as scenario 1.
2. Server 1: The Web service requester invokes a Web service on server 2 using the LTPA authentication method.
3. WSGW: The JAAS login configuration validates the LTPA token, and sets it as the current JAAS Subject for the WSGW application. The WSGW removes the authentication information from the incoming SOAP message and replaces it with a UsernameToken element, which contains the user name USER001. The WSGW sends it's server certificate to the target server.
4. Server 2: The target server control region receives the WSGW server certificate and performs the CBIND check to ensure that the request is being received from a trusted party.
5. WSGW: The Web Services runtime sends the SOAP message with the UsernameToken element.
6. Server 2: The Web services security runtime sets USER001 as the current security context in the JAAS subject, so that the EJB application runs with user ID.

210 © 2005 IBM Corporation ON DEMAND BUSINESS

Web Services solution 3 Configuration:

Version 5.x application

- Web service requester Server1 **login binding**:
  - Authentication method: LTPA
  - Token local name: LTPA
  - Token value type URI: http://www.ibm.com/websphere/appserver/totentype/5.0.2
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.LTPATokenCallbackHandler
- WSWG **login mapping**:
  - Authentication method: LTPA
  - Configuration name: WSLogin
  - Token local name: LTPA
  - Token value type URI: http://www.ibm.com/websphere/appserver/totentype/5.0.2
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.WSCallbackHandlerFactoryImpl
- WSWG **login binding**:
  - Authentication method: IDAssertion
  - ID Type: User name
  - Trust Mode: Not specified
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.WSCallbackHandlerFactoryImplAuthentication
- Web service provider Server2 **login mapping**:
  - Authentication method: IDAssertion
  - ID Type: User name
  - Trust Mode: Not specified
  - Configuration name: system.wssecurity.IDAssertion
  - Callback handler classname: com.ibm.wsspi.wssecurity.auth.callback.WSCallbackHandlerFactoryImplAuthentication

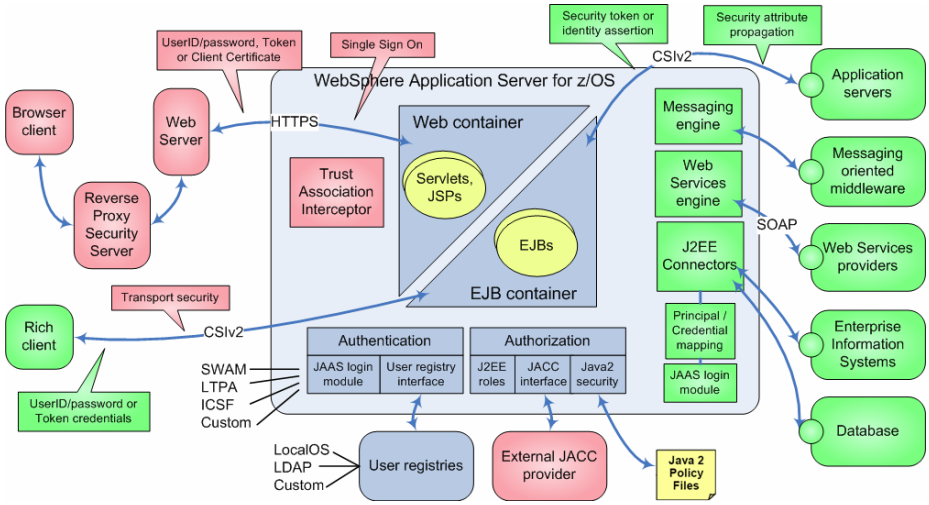
211 © 2005 IBM Corporation ON DEMAND BUSINESS

Conclusion

End-to-End Security Solutions for WebSphere on z/OS

212 © 2005 IBM Corporation ON DEMAND BUSINESS

End-to-End Security Solutions for WebSphere on z/OS



213 © 2005 IBM Corporation ON DEMAND BUSINESS

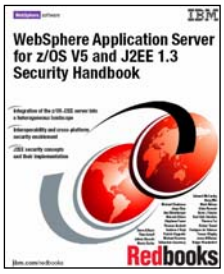
214 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Resources

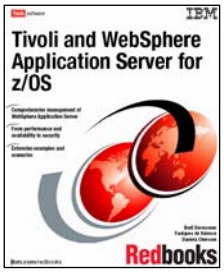
- WebSphere Application Server v6.0 Infocenter
  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/
- Redbooks

SG24-6760-00



SG24-6086-01

ZG24-6752-00



SG24-7062-00

215 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

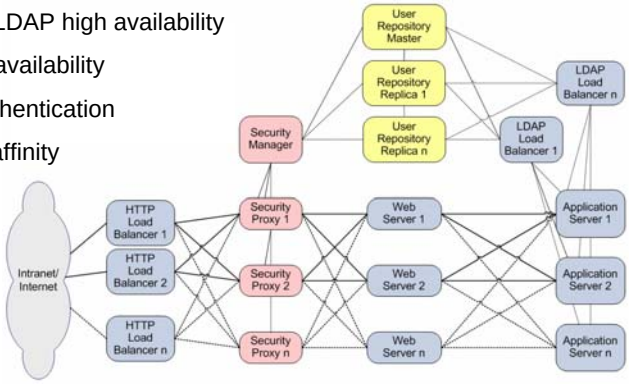
216 © 2005 IBM Corporation ON DEMAND BUSINESS

Backup Slides

© 2005 IBM Corporation

TAM and WebSphere z/OS
High availability, Scalability, Affinity

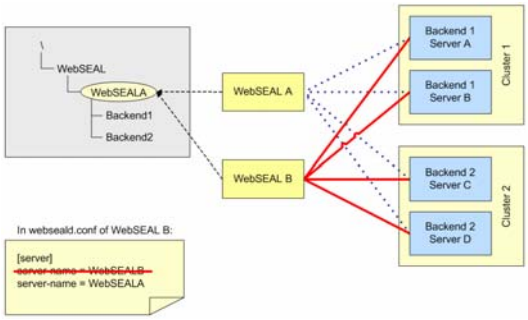
- Replicated WebSEAL
- Policy server is not a SPOF
- WAS for z/OS cluster
- TAM, WebSEAL and LDAP high availability
- WAS and LDAP high availability
- WebSEAL failover authentication
- WebSEAL and WAS affinity



© 2005 IBM Corporation

Replicated WebSEAL

- For high availability and scalability purposes, WebSEAL servers can be duplicated. These are called Replicated WebSEAL.



- When you replicate front-end WebSEAL servers, each server must contain an exact copy of the Web space, the junction database.
- Replicated WebSEAL allows failover of client sessions between WebSEAL servers.

219 © 2005 IBM Corporation ON DEMAND BUSINESS

Policy Server is not a SPOF

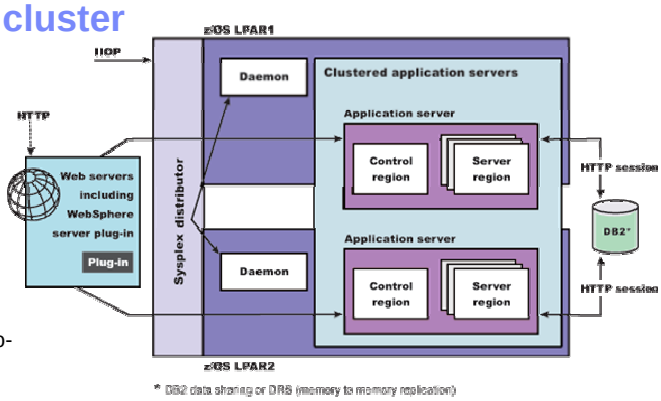
- The TAM Policy Server is not a Single Point Of Failure (SPOF).
- WebSEAL can still perform authentication when the Policy Server is down because WebSEAL uses a local authorization database replica.
- TAM automatically replicates the primary authorization policy database that contains the policy rules and credentials when a new application component, configured in local cache mode, or a TAM resource manager (such as WebSEAL or an Authorization Server) is configured.
- Update notification from the policy server (whenever a change has been made to the master authorization policy database) triggers the caching process to update all replicas.
- The only portion of TAM that cannot be duplicated within the same secure domain is the Policy Server.
- You can, however, have a second server in stand-by to provide manual fail-over capabilities as a first aid response.
- In general, the most effective way to have a redundant Policy Server is to configure an original and standby Policy Server in an HACMP (or similar) environment.

220 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WAS for z/OS cluster

- A WAS for z/OS horizontal cluster spans on two LPARs.
- A WAS for z/OS horizontal cluster duplicates Daemons, Control and Servant regions.
- HTTP sessions can be shared using memory-to-memory replication or stored in DB2.



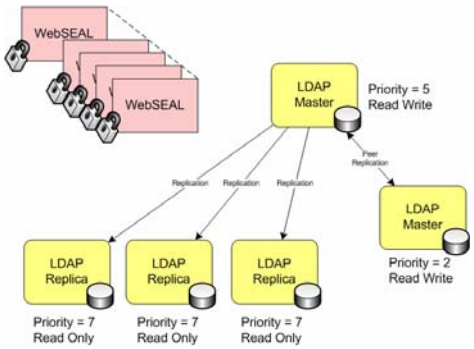
- A dynamic virtual IP address (DVIPA) is defined through the z/OS Sysplex Distributor as the daemon IP name for the cell.
- This IP address enables WLM-balanced routing and fail over between the LPARs for IIOP requests.
- A static IP address is required for each node as an auxiliary HTTP transport name for the cell. This enables directed HTTP routing for sessional HTTP requests.

221 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

TAM, WebSEAL and LDAP High Availability

- TAM Policy server and WebSEAL support hierarchical preference values to allow access to a multiple LDAP servers (with failover to the other servers).
- These preference values are called Priorities.

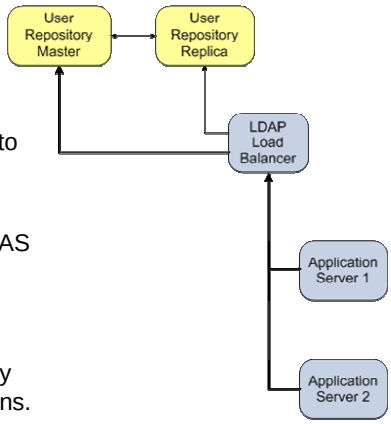


- For WebSEAL servers, you would put higher priority values to access Replica LDAP servers because WebSEAL only does read-only access to LDAP.
- For a Policy server, you would put higher priority values to access Master LDAP servers to be able to write to LDAP.

222 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WAS and LDAP high availability



- WAS does not possess any mechanism to access multiple LDAP servers (Master, Replica).
- In order to use LDAP high availability, WAS needs a load balancer that can route requests to the LDAP master or LDAP replicas.
- The Load Balancer should support affinity or WAS shouldn't reuse LDAP connections.

223 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WebSEAL failover authentication

- **Failover authentication**: method that enables an authenticated session between a client and WebSEAL to be preserved when the WebSEAL server becomes unavailable. This prevents the end-user to have to log-in again if the WebSEAL server fails.
- It enables the client to connect to another WebSEAL server, and create an authentication session containing the same user session data and user credentials.
- Failover authentication uses **failover cookies**.
- The failover cookie contains client-specific data, such as user name, cookie-creation time stamp, original authentication method, and an attribute list.
- When the replicated WebSEAL server receives this cookie, it decrypts the cookie, and uses the user name and authentication method to regenerate the client's credential.

224 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

WebSEAL and WAS Affinity

- **WebSEAL** supports back-end server affinity through the use of **Stateful Junctions**.
- Each back-end server has a unique identifier called UUID.
- Using a stateful junction, this UUID is stored in a cookie and sent with all HTTP requests. The cookie's Server UUID information ensures that the HTTP requests are consistently routed to the same back-end server.
- **HTTP Server running WAS plug-in** supports server affinity through the use of **CloneID** attributes.
- Each WAS cluster member has a unique CloneID attribute.
- When affinity is switched on, HTTP requests possess an URL JSESSIONID attribute or a JSESSIONID cookie which contain the CloneID value.
- The WAS plug-in ensures that HTTP requests are consistently routed to the same WAS cluster member using the CloneID value.

225 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

zSeries – Designed for On Demand Solutions

Performance
- Fast, consistent and predictable
- 64-bit Architecture
- Balanced system design
- End-to-end performance management
- SSLs/sec

Open
- Embracing standards for ease of integration

Resilient
- Superior reliability and security
- Self-healing, self-protecting
- Multi-site business continuity solutions

Scalablity
- Scale up, scale out to meet unpredictable demand
- Capacity On Demand
- Variable Workload Charge (VWLC) Software

Efficiency
- Share resources for greater utilization and reduced costs
- End-to-end prioritization
- Outstanding utilization rates
- Energy, floor-space, networking, administration costs

Virtual
- Cost-effective consolidation and integration
- 100s of virtual blades
- Network in a box - HiperSockets

226 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Balanced Systems Design

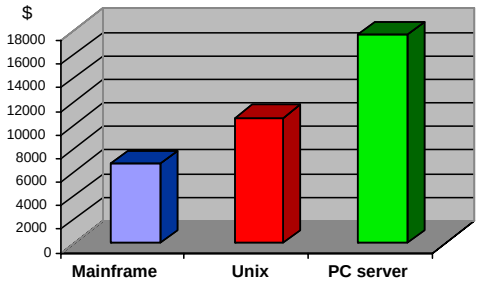
zSeries Servers designed for end-to-end transaction throughput

- **To achieve a balanced system, you need**
  - High microprocessor performance
  - Very Large I/O Bandwidth
  - High speed connectivity
  - Very high speed intelligent storage servers
  - Many paths to your data
  - Performance Management Software
- **Balance is important**
  - High performance
  - Maximum throughput
  - Avoid bottlenecks
  - High utilization and low TCO

227 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

Mainframe cost advantage stronger than ever



Total annual cost per user over a five-year period
Source: ***The Dinosaur Myth*****, www.arcati.com, 2004**

Arcati Research

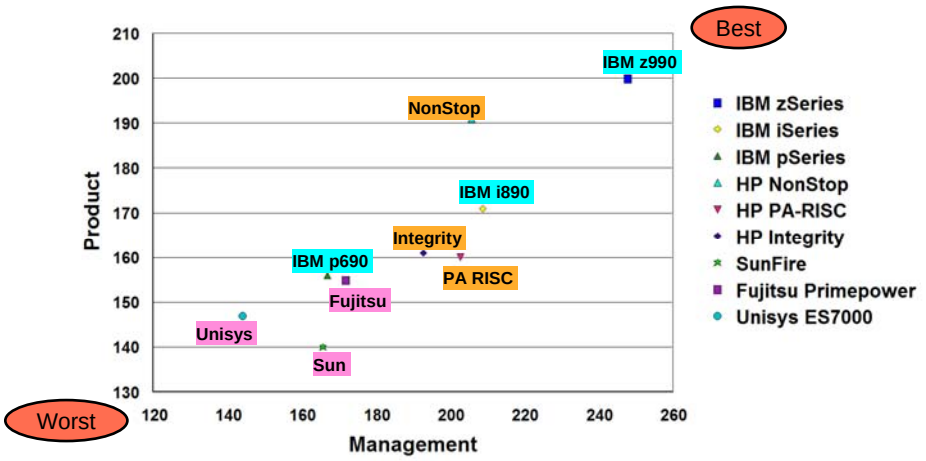
"We believe that simple Cost of Ownership comparisons between the mainframe and distributed platforms are often misleading - dangerously so. Many distributed costs are surprisingly well hidden within the enterprise; but with the mainframe data center, what you pay is what you get.

And what you get is unparalleled scalability, very high utilization levels, and mature centralized management. This in turn significantly reduces the need for technical support, simplifies change management, and allows a more flexible approach to business continuity. "

Mark Lillycrop, Arcati Research

228 © 2005 IBM Corporation ON DEMAND BUSINESS

Gartner ASEM - Product and Manageability



Source: Gartner, ASEM Enterprise Server Update 2004, 22nd October 2004

229 © 2005 IBM Corporation ON DEMAND BUSINESS

zSeries customers recognize highly differentiated value across the zSeries platform

- Extremely High Availability and Overall Reliability
- Massive end-to-end Scalability
- Capacity on Demand
- Rock Solid Security and Privacy
- Advanced Virtualization Capabilities
- Highly Manageable, Responsive and Autonomic via Workload Manager (WLM) and Intelligent Resource Director (IRD)
- Utilizes Open and Industry Standards
- World-class Integrated Support
- Higher Utilization and Balanced System Design

zSeries average system utilization often exceeds 80%, and zSeries servers are designed to handle sustained peak workload utilization of 100% without service level degradation to high priority workloads.

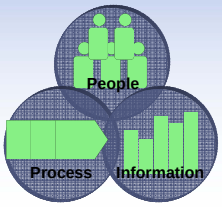
230 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

zSeries Software for the On Demand Operating Environment

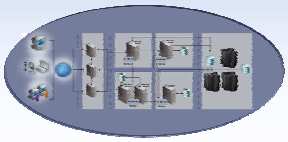
Integration

Business flexibility through integration of people, processes and information within and beyond the enterprise



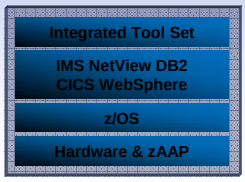
Infrastructure Management

Optimize IT through integration of management tools and information across the end to end systems infrastructure



Platform Readiness
(Technology, Sub-capacity pricing (WLC))

Resilient, Available and Secure - the underpinnings of an On Demand Infrastructure



231 © 2005 IBM Corporation ON DEMAND BUSINESS

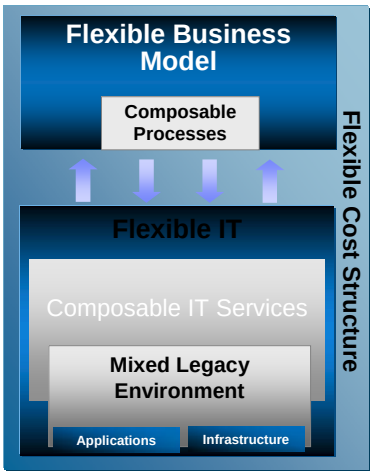
Redbooks IBM

Integration

On Demand Business

Requires a World Class Integration Platform

- On demand businesses requires **flexible business processes…**
- Flexible business processes require **flexible IT…**
- Flexible IT requires a world-class platform for **integration of applications and data** across the entire enterprise
- *IBM offers an **unmatched integration platform** for the on demand world*



232 © 2005 IBM Corporation ON DEMAND BUSINESS

Redbooks IBM

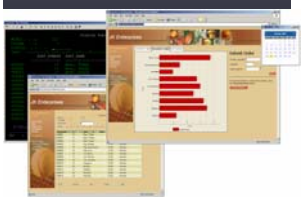
Integration

Transforming and Reusing Existing IT Assets

Three Approaches to Modernizing Legacy Applications

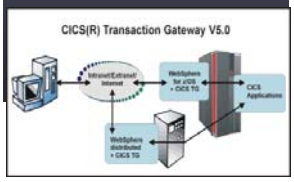
Transform the User Experience

Improve the user interface and workflow for quick return on investment



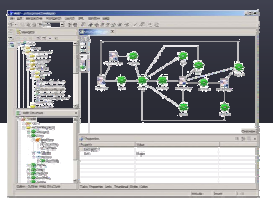
Transform the Application Connectivity

Integrate legacy applications throughout the enterprise using Web services and Java*TM connectors



Transform the Application Architecture

Update and extend mission-critical applications as services, leveraging their core value in new ways



233 © 2005 IBM Corporation ON DEMAND BUSINESS