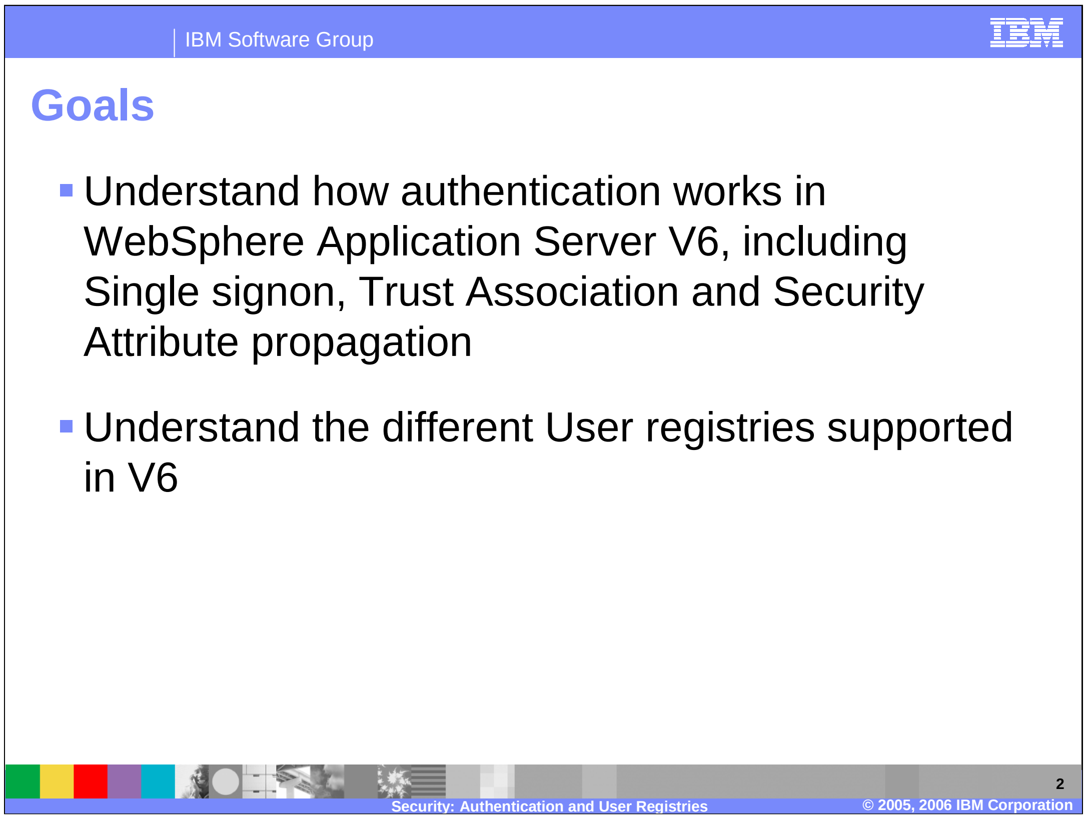
IBM Software Group

IBM® WebSphere® Application Server V6

Security Authentication Details and User Registries

business on demand.

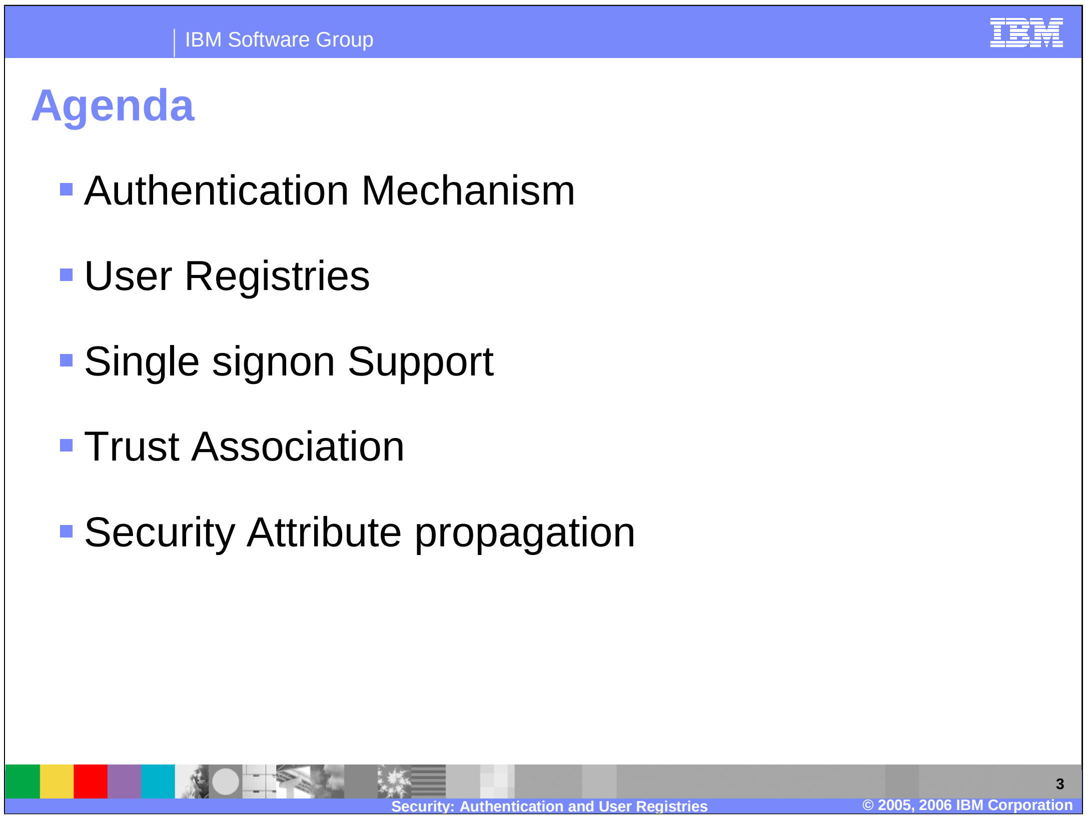
This presentation will focus on the Security Authentication details.



Goals

- Understand how authentication works in WebSphere Application Server V6, including Single signon, Trust Association and Security Attribute propagation
- Understand the different User registries supported in V6

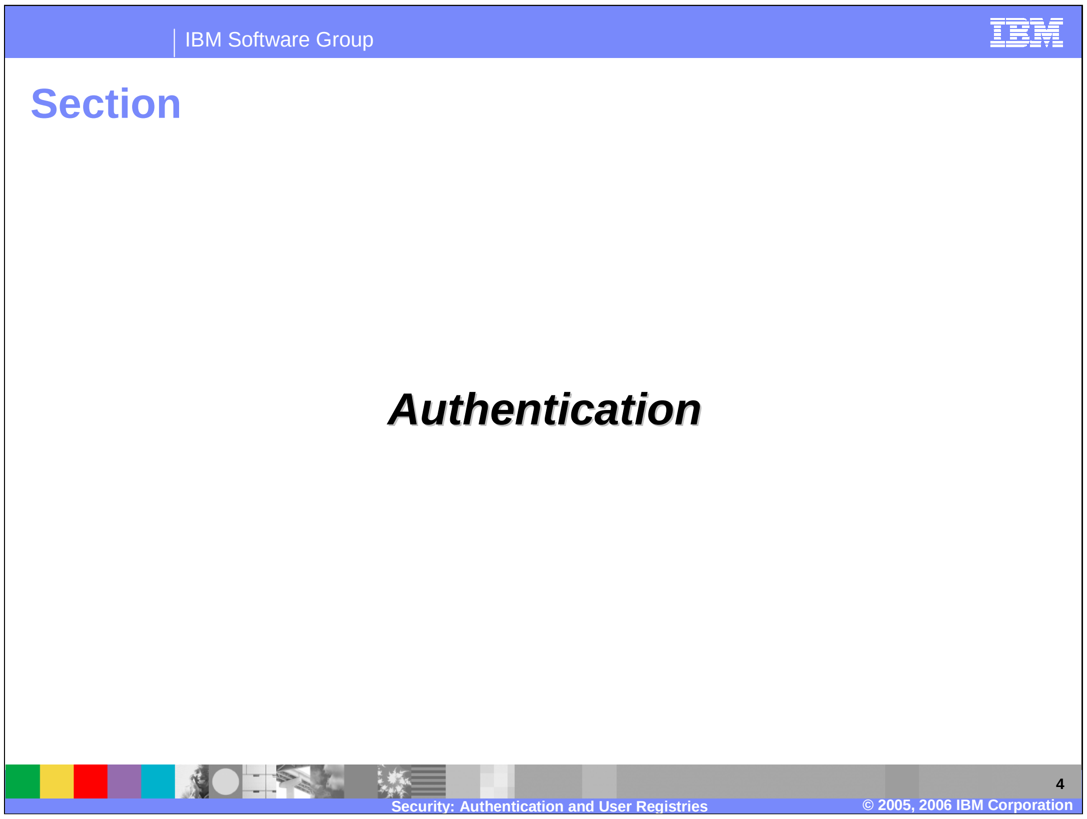
The goals for this presentation are to understand the Authentication mechanism, Authentication User Registries and other authentication functions like single signon, and how to use Trust Association to let third party vendors perform authentication of incoming requests.



Agenda

- Authentication Mechanism
- User Registries
- Single signon Support
- Trust Association
- Security Attribute propagation

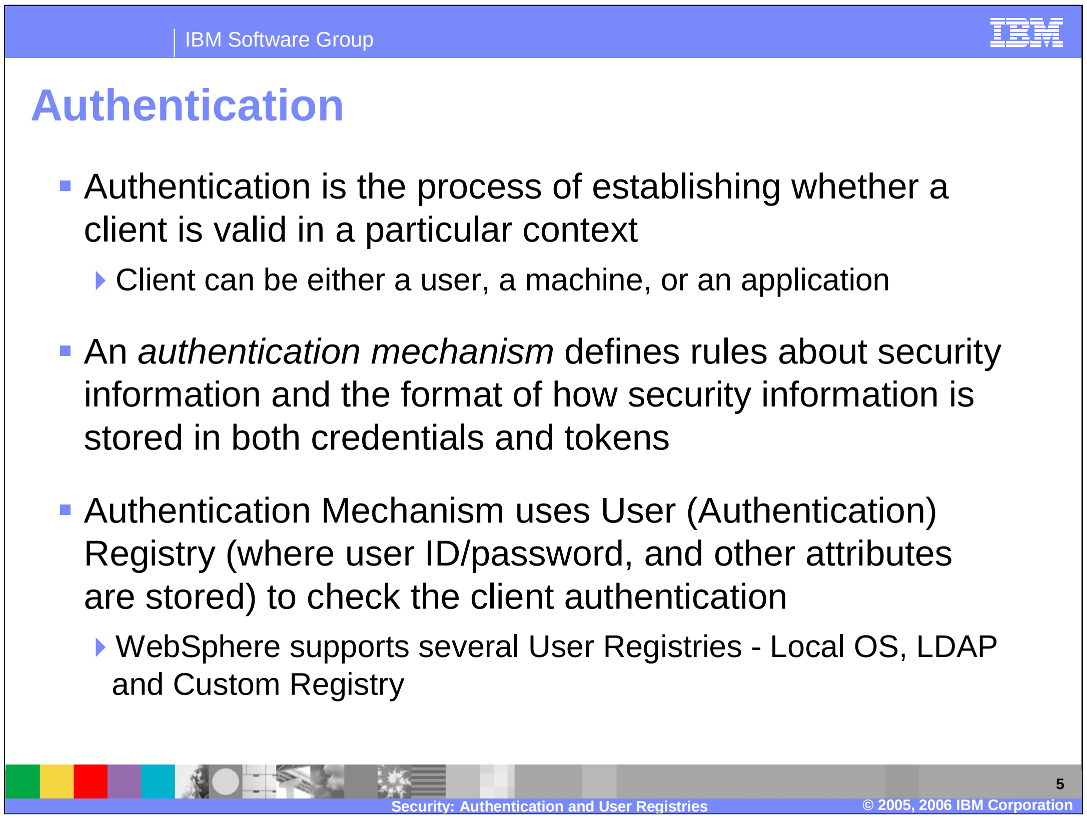
The agenda for this presentation is as listed on this page.



Section

Authentication

The next section will discuss Authentication details.



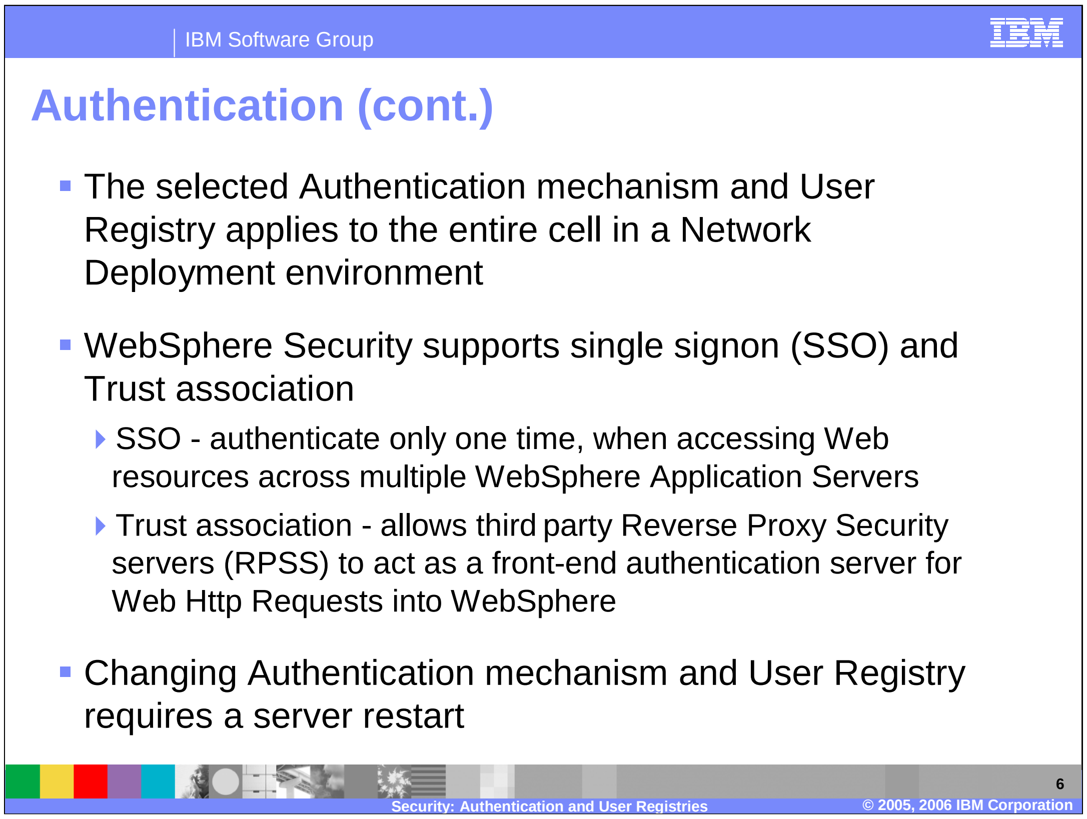
Authentication

- Authentication is the process of establishing whether a client is valid in a particular context
  - Client can be either a 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
- 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

Authentication establishes client identity. It uses an external user registry that holds information about the clients, like user id, password and other attributes.

WebSphere Application Server V6 supports different authentication mechanisms. These authentication mechanisms save the authentication information differently, and that dictates whether you can forward the Authentication information to other servers.

The support of Authentication mechanism and user registries in V6 are similar to the support in V5.



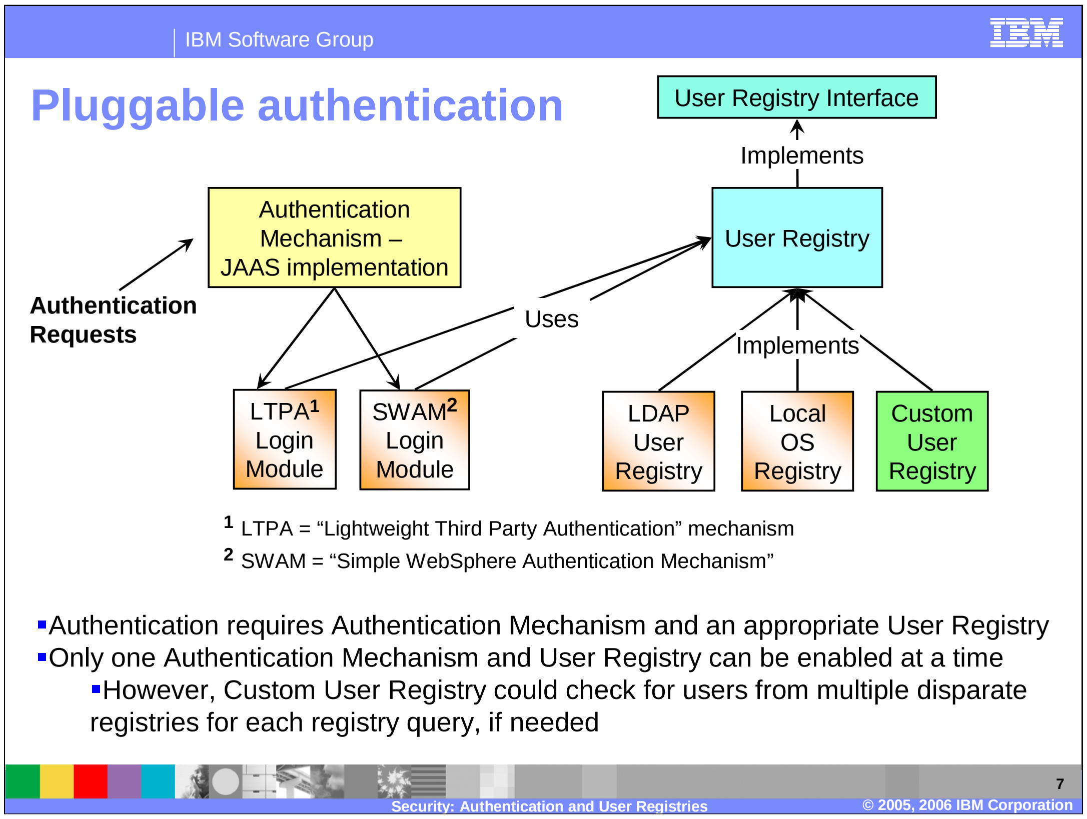
Authentication (cont.)

- The selected Authentication mechanism and User Registry applies to the entire cell in a Network Deployment environment
- WebSphere Security supports single signon (SSO) and Trust association
  - SSO - authenticate only one time, when accessing Web resources across multiple WebSphere Application Servers
  - Trust association - allows third party Reverse Proxy Security servers (RPSS) to act as a front-end authentication server for Web Http Requests into WebSphere
- Changing Authentication mechanism and User Registry requires a server restart

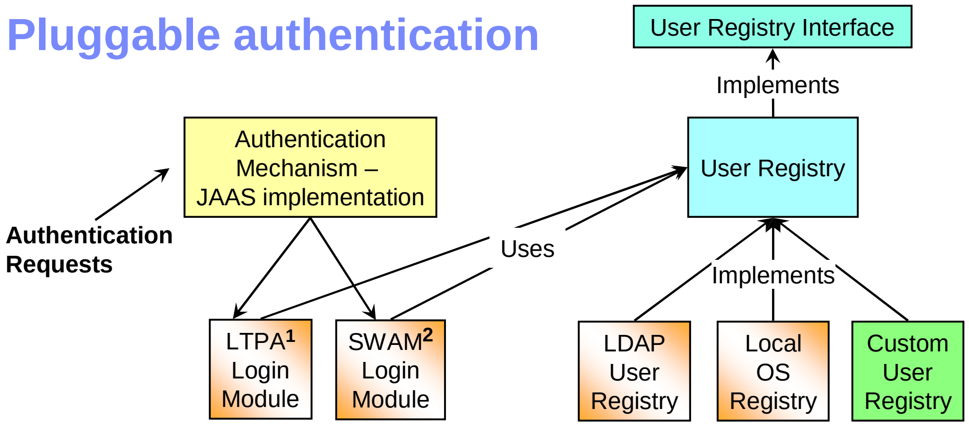
Even though several authentication types and user registries are supported in V6, you can select only one authentication type and one user registry to be used for the entire Network Deployment Cell.

In addition, single signon and Trust association are also supported under certain authentication mechanisms. Details are explained later.

Any changes to Security configuration requires a server restart.



Pluggable authentication

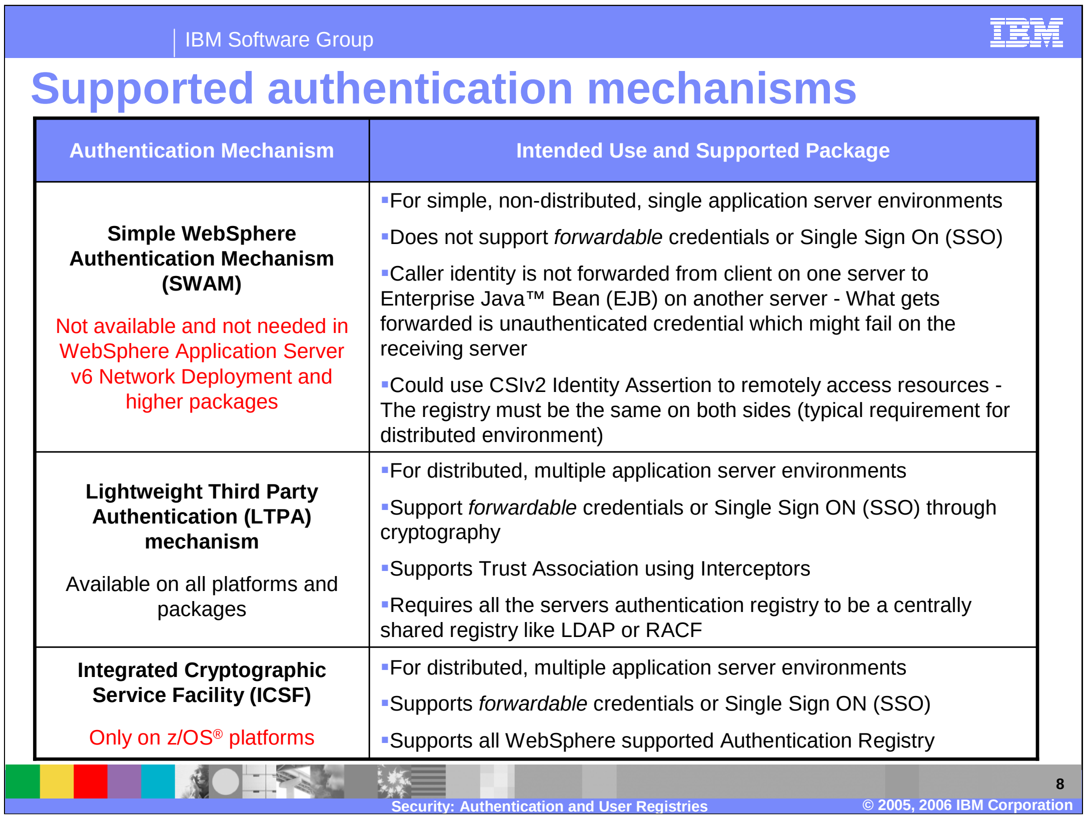


[1] LTPA = “Lightweight Third Party Authentication” mechanism

[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

This slide shows the architecture of the Pluggable Authentication, where there are 2 Authentication mechanisms (LTPA and SWAM). Each one of them can use any one of the supported User registries (Local OS, LDAP or Custom Registry). Details of the Authentication mechanism and user registry are in the next few slides.



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 Enterprise Java™ Bean (EJB) on another server - What gets forwarded is unauthenticated credential which might fail on the receiving server ▪Could use CSIv2 Identity Assertion to remotely access resources - The registry must be the same on both sides (typical requirement for distributed environment)
Lightweight Third Party Authentication (LTPA) mechanism Available on all platforms and packages	▪For distributed, multiple application server environments ▪Support *forwardable* credentials or Single Sign ON (SSO) through cryptography ▪Supports Trust Association using Interceptors ▪Requires all the servers authentication registry to be a centrally shared registry like LDAP or RACF
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

SWAM and LTPA are the two Authentication mechanisms available on the Distributed, iSeries® and zSeries® platforms.

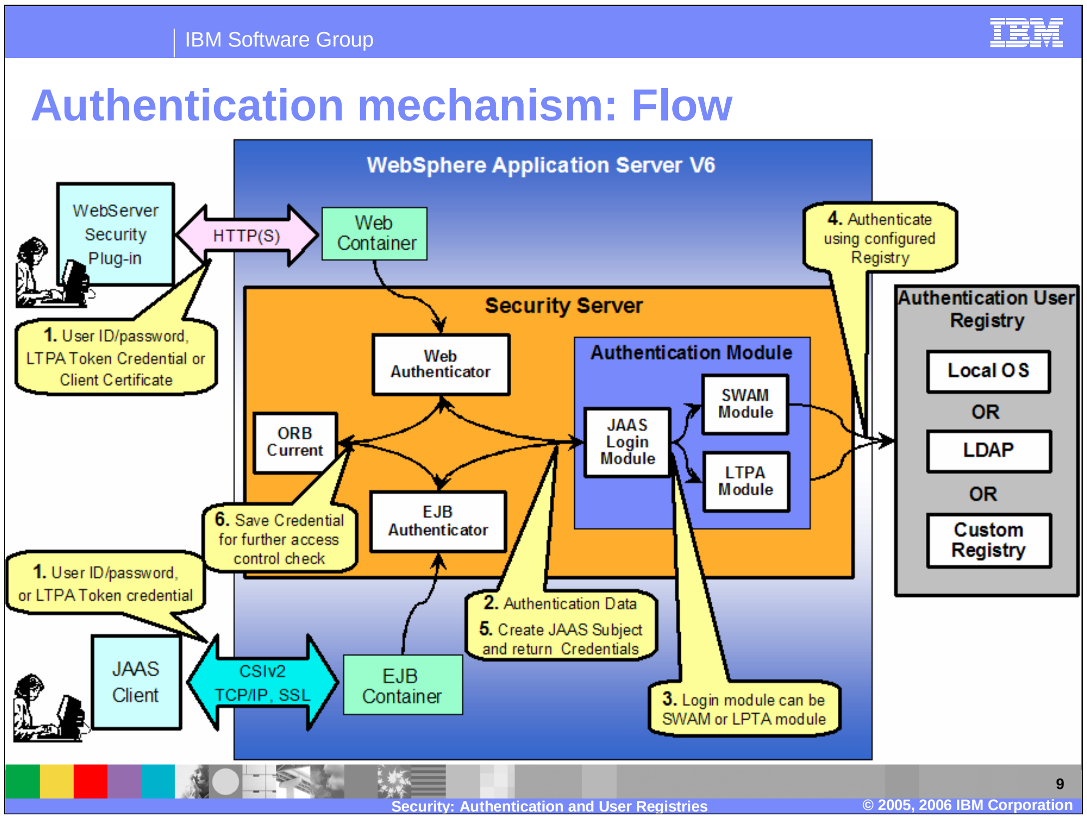
SWAM is used mainly in a single stand-alone Application Server environment, whereas LTPA is used in a distributed multiple server environment.

When using LTPA across multiple servers, you would need to have a user registry that is accessible by all the servers. LDAP is a good option in this environment or the SAF registry in the case of z/OS.

SSO and Trust association are only supported when using LTPA.

LTPA has all the capabilities of SWAM, plus more. In a Network Deployment Cell, there is no support for SWAM and none required.

ICSF is being deprecated in V6. It will allow you to read SSO tokens generated from ICSF but will no longer generate them. Support will eventually move to LTPA to support ICSF-generated keys.


IBM Software Group

Authentication mechanism: Flow

WebSphere Application Server V6
WebServer Security Plug-in
HTTP(S)
Web Container
4. Authenticate using configured Registry
Security Server
Authentication User Registry
1. User ID/password, LTPA Token Credential or Client Certificate
Web Authenticator
Authentication Module
SWAM Module
JAAS Login Module
LTPA Module
ORB Current
Local OS
OR
LDAP
OR
Custom Registry
EJB Authenticator
6. Save Credential for further access control check
1. User ID/password, or LTPA Token credential
2. Authentication Data
5. Create JAAS Subject and return Credentials
JAAS Client
CSIv2 TCP/IP, SSL
EJB Container
3. Login module can be SWAM or LPTA module

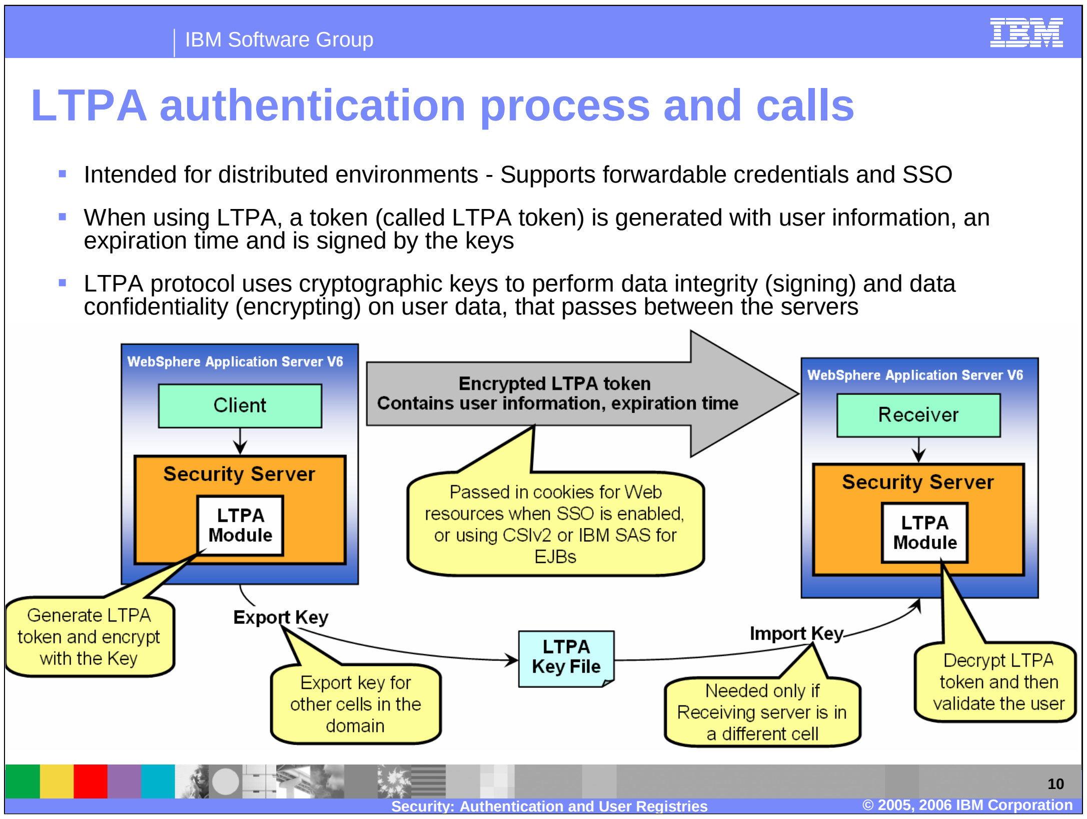
9

Security: Authentication and User Registries

© 2005, 2006 IBM Corporation

This picture shows a more detailed flow of the authentication process for both Web clients and EJB clients.

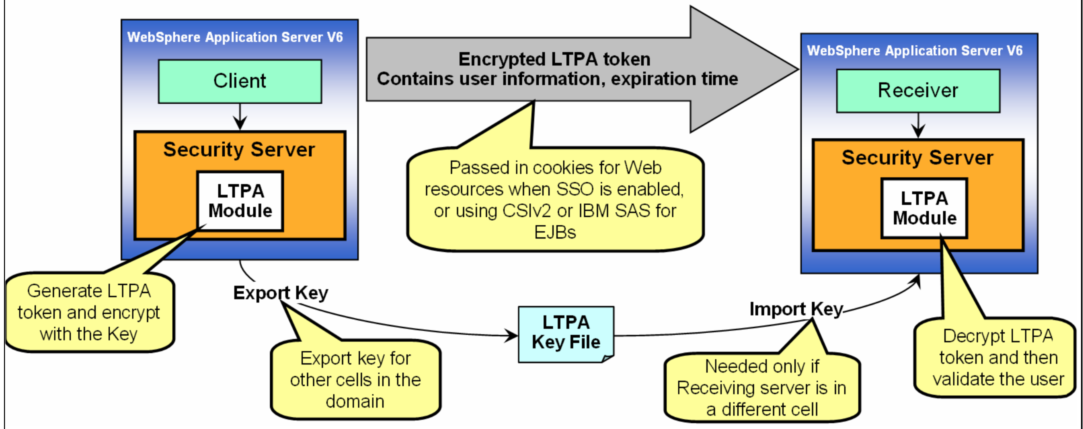
When the clients send authentication information, the Authentication module (SWAM or LTPA) will then authenticate the user using the User Registry, if the authentication information matches an entry in the user registry. The Authentication module will create the JAAS subject for the client identity. The Subject is then saved for further access. If required, the token for the Subject is sent back, through a cookie for the Web client, or over the CSIv2 authentication protocol for the EJB client.



IBM Software Group

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

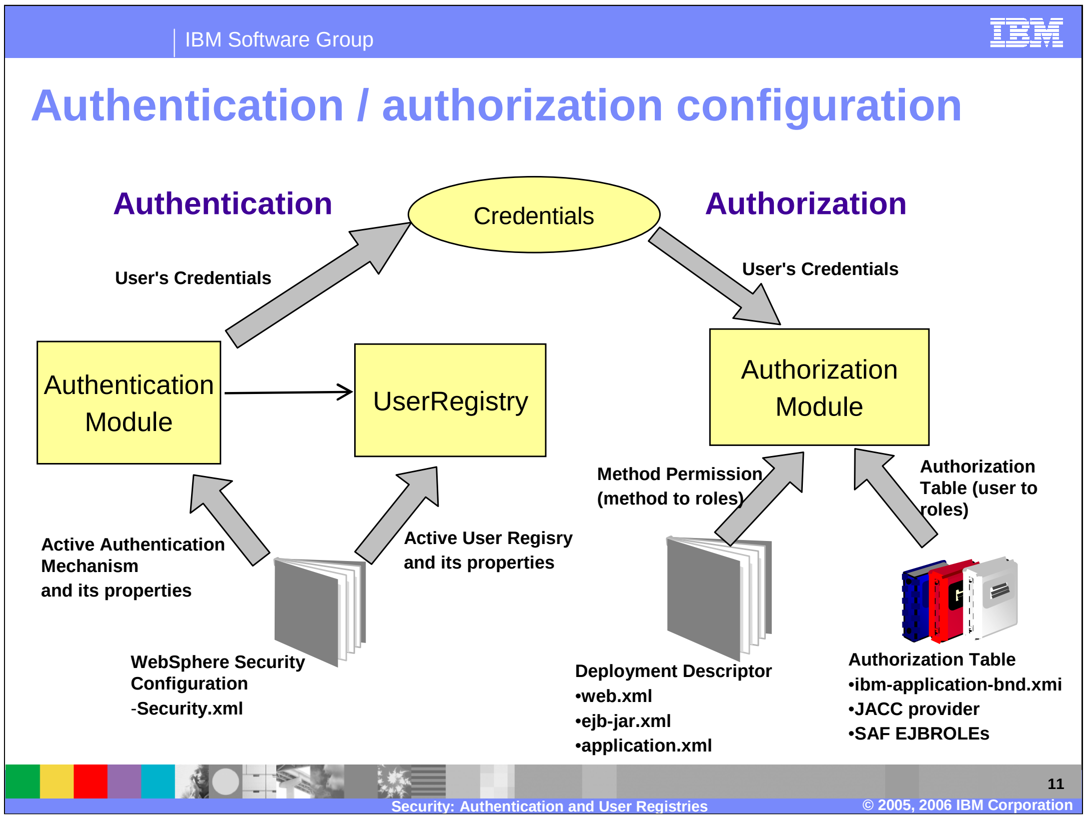

WebSphere Application Server V6
Client
Security Server
LTPA Module
Encrypted LTPA token
Contains user information, expiration time
Passed in cookies for Web resources when SSO is enabled, or using CSIv2 or IBM SAS for EJBs
WebSphere Application Server V6
Receiver
Security Server
LTPA Module
Generate LTPA token and encrypt with the Key
Export Key
Export key for other cells in the domain
LTPA Key File
Import Key
Needed only if Receiving server is in a different cell
Decrypt LTPA token and then validate the user

10

Security: Authentication and User Registries

© 2005, 2006 IBM Corporation

This page shows the detailed flow of LTPA Authentication. As indicated before, LTPA supports forwardable credentials and SSO. The 2 servers participating in the LTPA authentication must have the same LTPA keys. This is true for servers within the same cell. For servers in different cells, you could export the LTPA key from the Sender (client) and import it to the Receiver. These keys can then be used to decrypt the LTPA token being sent by the sender to the receiver. Data passed between servers using LTPA protocol is signed for integrity and encrypted for confidentiality.



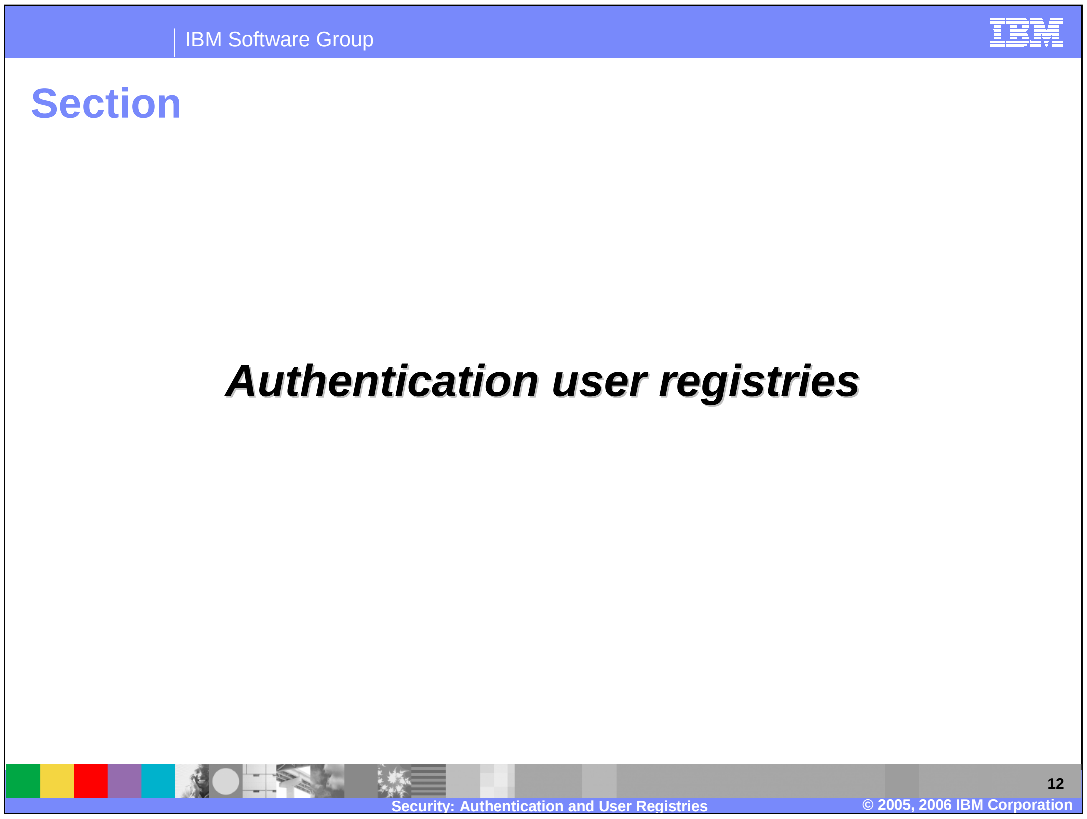
Authentication / authorization configuration

This slide shows the different configuration required for Authentication and Authorization.

Within V6 Server, you have to select and configure an authentication mechanism (SWAM or LTPA) and user registry. This information is saved in the security.xml within WebSphere configuration.

With the applications, you have to define the J2EE Security Roles and the method permissions (methods to roles). These Security roles are part of the Application Deployment descriptor, whereas the permissions are part of either the Web or the EJB Deployment descriptor. Finally, the security Role to the user/group binding needs to be specified. You can use the default binding that uses the IBM binding file within the application EAR, a third party JACC provider or SAF EJBROLEs if on the z/OS platform.

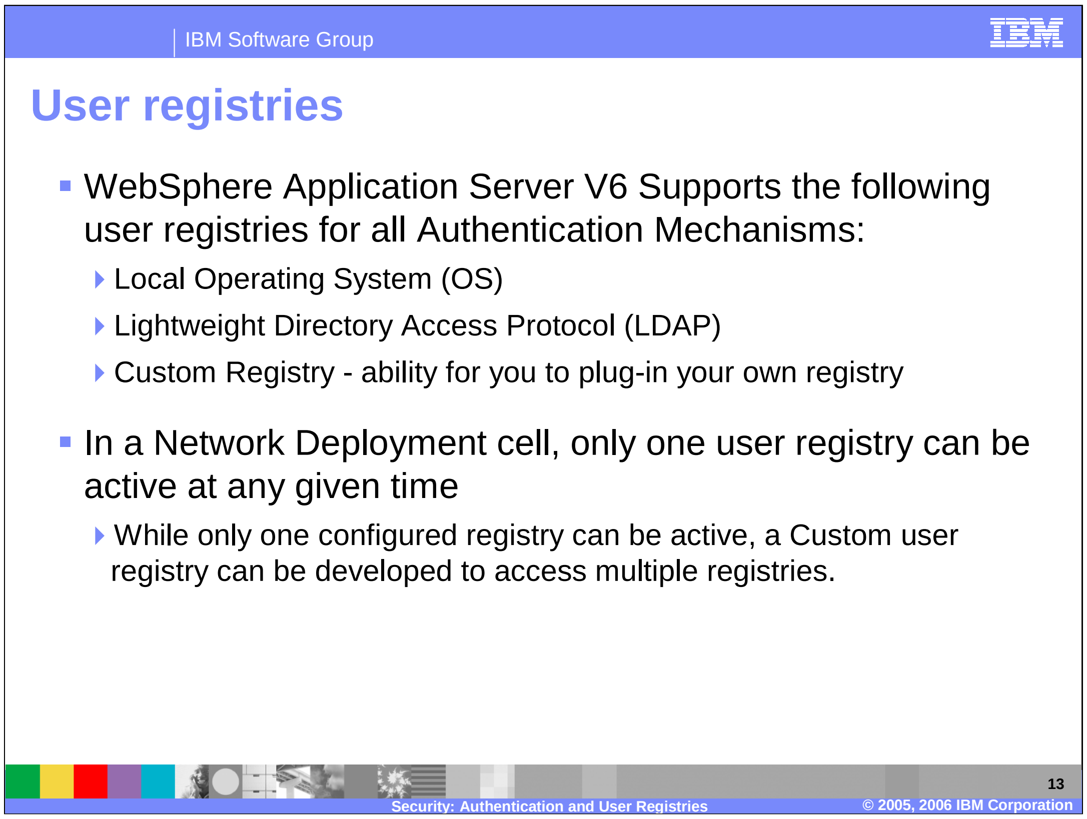
The Authentication module creates the User Credentials which is then used by the Authorization module to check if the J2EE Security Roles for that user/group has the necessary permission.



Section

Authentication user registries

The next section will discuss the Authentication User Registries


IBM Software Group

User registries

- WebSphere Application Server V6 Supports the following user registries for all Authentication Mechanisms:
  - Local Operating System (OS)
  - Lightweight Directory Access Protocol (LDAP)
  - 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.

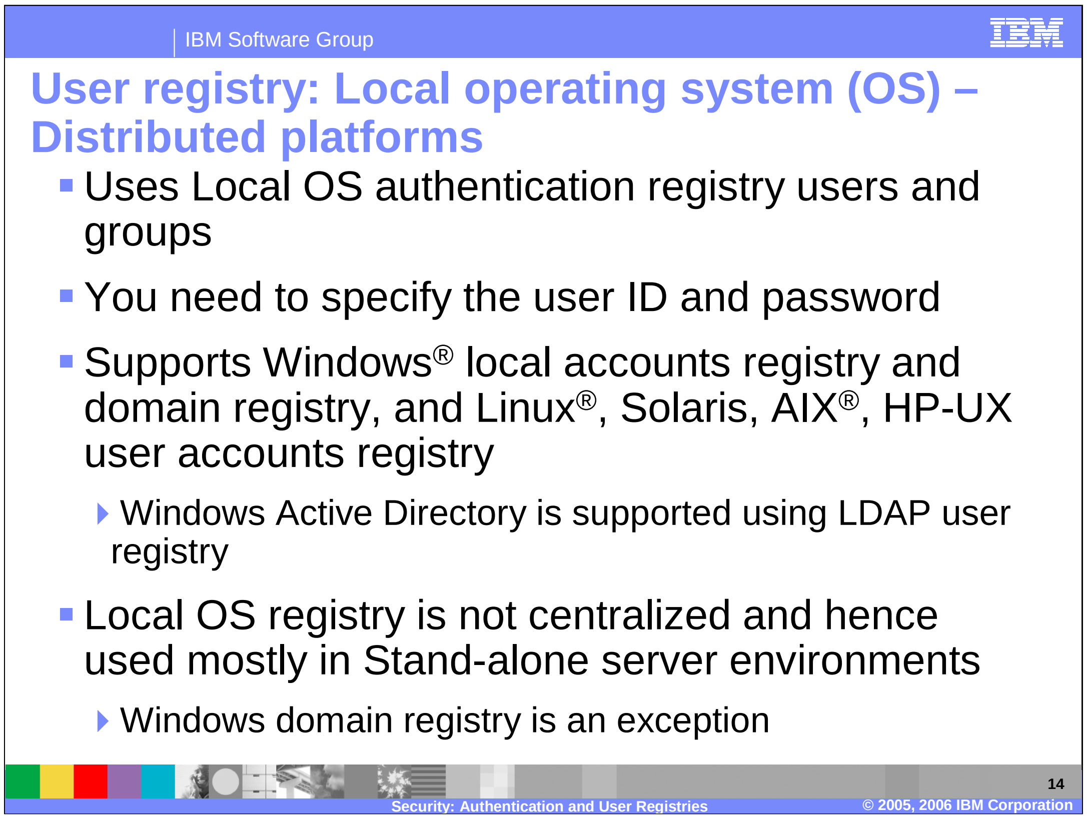
13

Security: Authentication and User Registries © 2005, 2006 IBM Corporation

WebSphere Application Server V6 supports three User Registries as listed on this page.

In each of the User registries, a valid security user ID and password needs to be supplied. On Distributed platforms, this security id is authenticated with the registry during Server startup, meaning that the security id must be valid in the User Registry. It also has administrative privileges and is used to log into the administrative console after security is turned on. Later, you can add additional administrator users. On the z/OS platform, the SERVER/STARTED and CBIND classes are used during Server startup instead. The userid that is defined in the STARTED profile needs to have the correct access to the SERVER and CBIND class in the local registry.

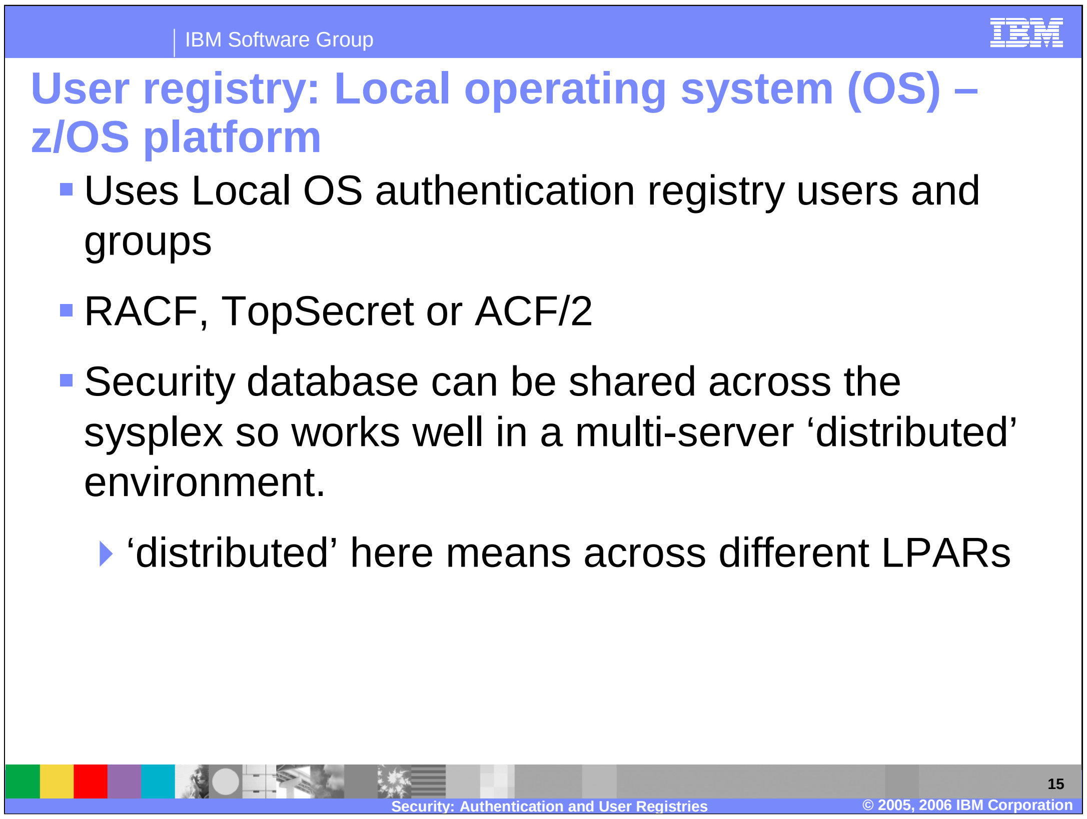
Note that this security user ID is different than the process-id that the server runs under. It is the process-id that requires permission to access the User registry and not the security user ID.



User registry: Local operating system (OS) – Distributed platforms

- Uses Local OS authentication registry users and groups
- You need to specify the user ID and password
- Supports Windows® local accounts registry and domain registry, and Linux®, Solaris, AIX®, HP-UX user accounts registry
  - Windows Active Directory is supported using LDAP user registry
- Local OS registry is not centralized and hence used mostly in Stand-alone server environments
  - Windows domain registry is an exception

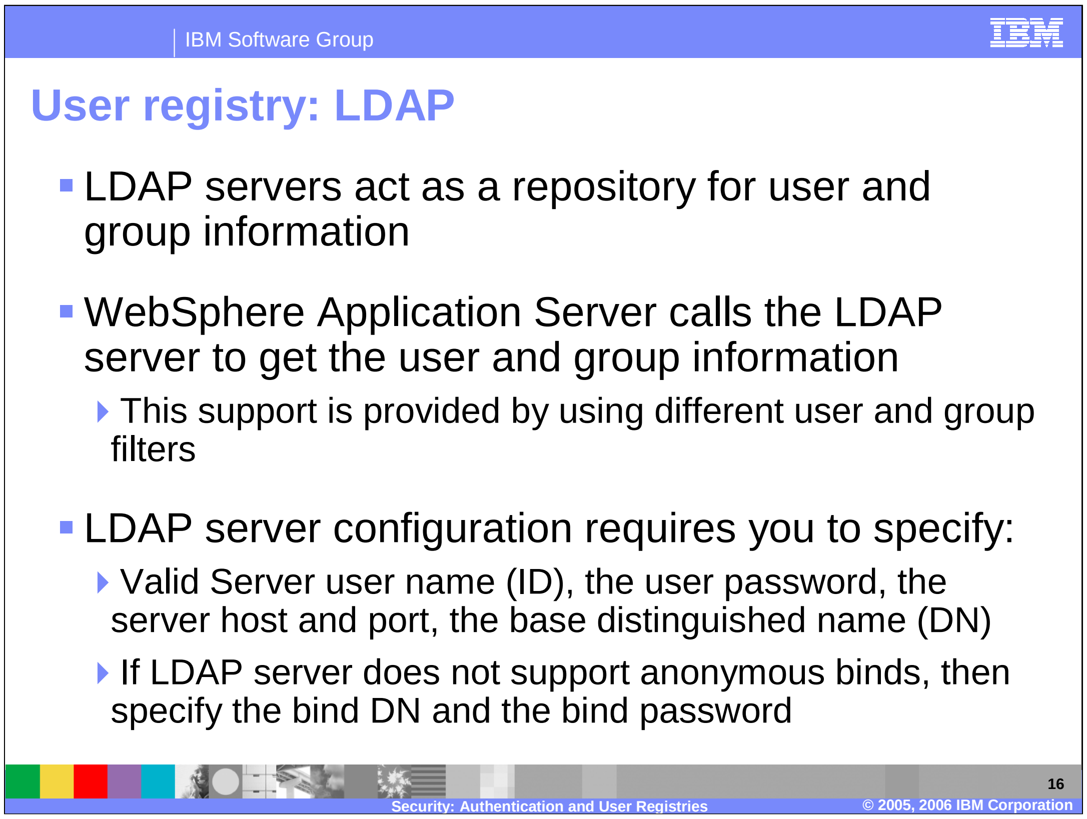
Local OS cannot be used in a multi-server distributed environment, since you need to share the user registry across the servers.



User registry: Local operating system (OS) – z/OS platform

- 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

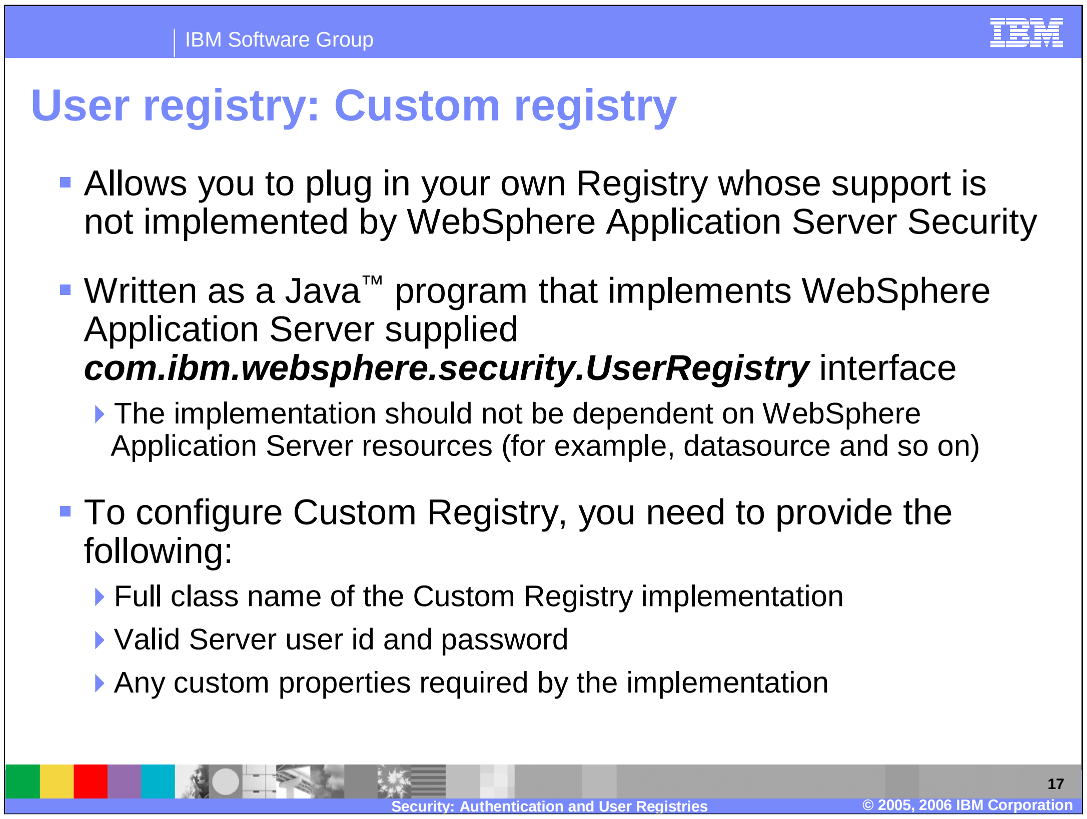
Unlike the distributed platforms, Local OS on the z/OS platform can be used in a multi-server 'distributed' environment, since the security database can be easily shared across the sysplex.



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

LDAP is widely used in a distributed environment where multiple servers need access to a central User registry**.** This is an option on z/OS as well.



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

Note that the Custom Registry class should be placed in the WebSphere Extension class loader path, since the WebSphere runtime requires this class. Hence, this should not be placed in a location that is loaded by class loaders below the Extension class loader hierarchy.

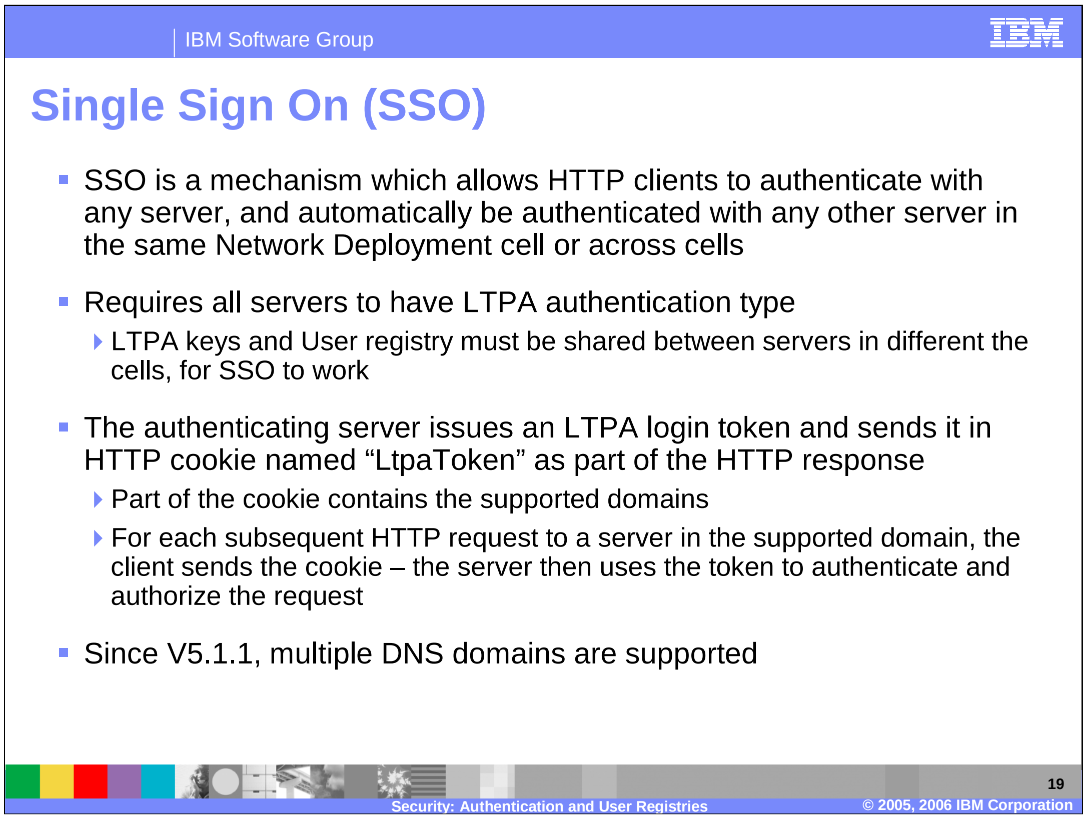
Application Server resources are loaded later than the Custom Registry class, and hence using Resources within Custom Registry will not work.

A sample of file-based custom Registry is supplied with WebSphere, called, FileRegistrySample – refer to WebSphere Application Server V6 Information Center for more details.

Section

Single Sign On support

The next section will discuss single signon Support.


IBM Software Group

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

19

Security: Authentication and User Registries

© 2005, 2006 IBM Corporation

SSO is widely used where the client needs to authenticate only once across multiple Web applications running across multiple Applications Servers. These servers can be part of the same cell or different cell or could be a stand-alone application server.

SSO is only supported with LTPA.

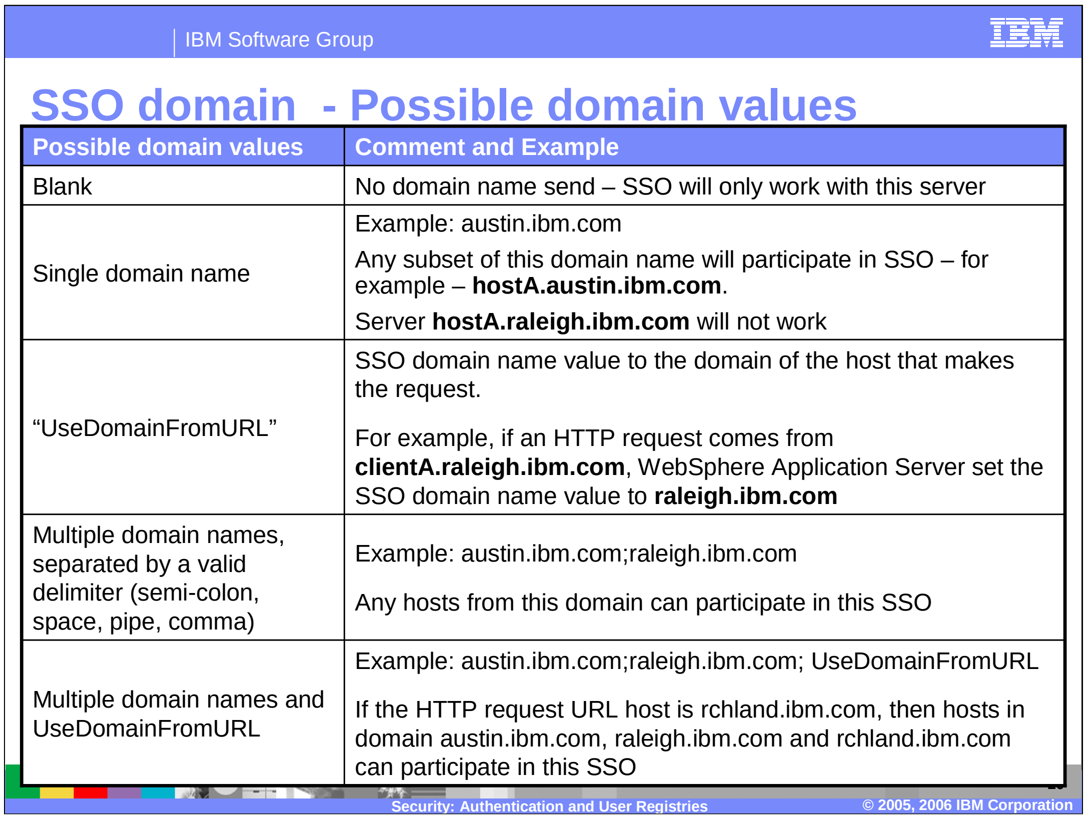
The authenticating server issues an LTPA login token and sends it in an HTTP cookie named "LtpaToken" in the HTTP response. This cookie is then sent by the client to each eligible server with each subsequent HTTP request. The server then uses the token to authenticate and authorize the request.

In V5, these servers needed to be part of the same DNS domain, that had to be specified in the SSO configuration. New, since V5.1.1, the limitation of only servicing one domain per WebSphere deployment is removed. Now, multiple domains can be specified. More details on this on the next page.

Different operating systems use different schemes for domain names, based on User registry.

If using Local OS as the user registry, on the Windows platform, the realm name is the domain name, if a domain is in use, or the machine name. On the UNIX® platforms, the realm name is the same as the host name.

When using LDAP as the user registry, the realm name is the host:port of the LDAP server


IBM Software Group

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

Security: Authentication and User Registries © 2005, 2006 IBM Corporation

Domain name is specified as part of SSO configuration and is included in the HTTP cookie

The HTTP client sends the LTPA login token in the subsequent HTTP requests in cookies, to servers in the specified domain. This allows the server to automatically authenticate the client**.**

There are 5 possible domain value options

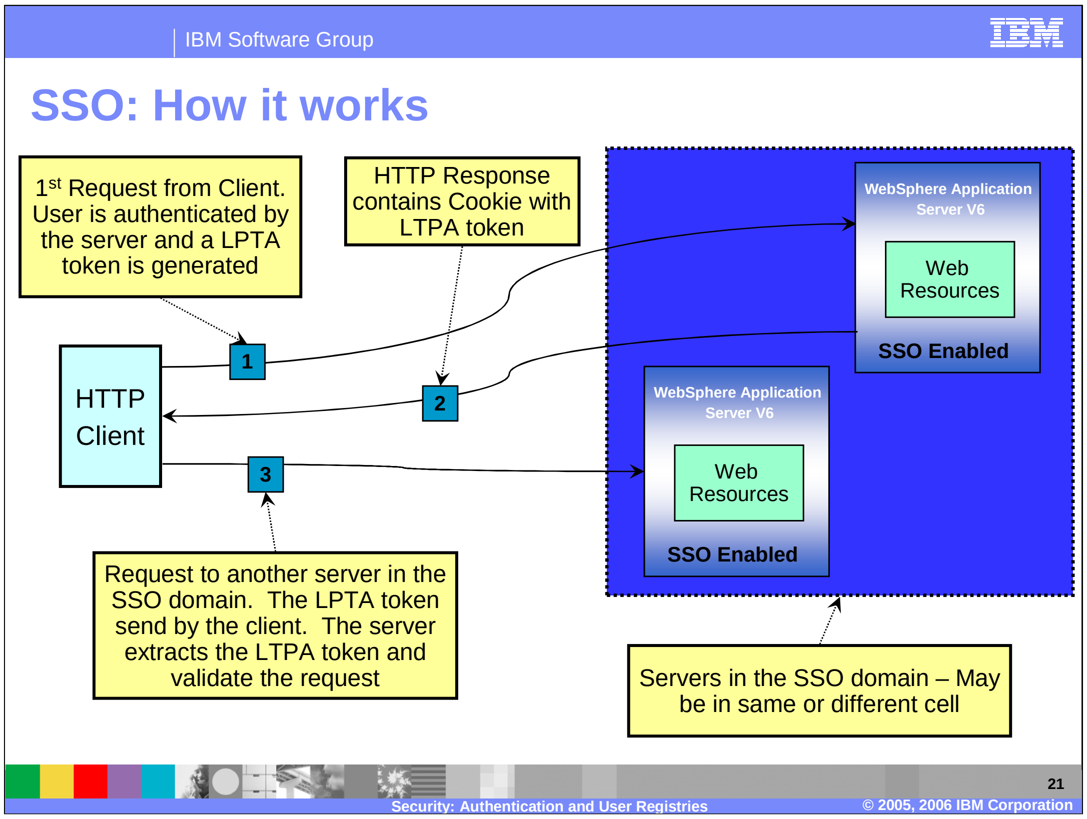
(1) When no domain name is specified, then only the authenticating server will receive the LTPA login token in subsequent HTTP requests.

(2) If a single domain name is specified, then any server that is part of the DNS domain will receive the LTPA login token. As an example, if the domain name is austin.ibm.com, then a server with host name hostA.austin.ibm.com will receive the LTPA login token from the HTTP client, if the request is being sent to that server. However, requests to hostB.raleigh.ibm.com will not receive the LTPA login token.

(3) If "UseDomainFromURL" is specified, then the domain name is generated from the URL of the incoming HTTP request. As shown in the table, if a request comes from clientA.raliegh.ibm.com, the generated domain name will be raleigh.ibm.com. Any subsequent request to a server in that domain will receive the LTPA login token.

(4) Multiple domain names, with each name separated by a delimiter. Semi-colon, space, pipe or comma can be used as a delimiter. This allows applications running on servers in different domains to participate in SSO.

(5) Last is the combination of multiple domain names and "UseDomainFromURL". The generated domain name is added to the multiple domain name list.


IBM Software Group

SSO: How it works

1st Request from Client. User is authenticated by the server and a LPTA token is generated
HTTP Response contains Cookie with LTPA token
WebSphere Application Server V6
Web Resources
SSO Enabled
1
2
HTTP Client
3
WebSphere Application Server V6
Web Resources
SSO Enabled
Request to another server in the SSO domain. The LPTA token send by the client. The server extracts the LTPA token and validate the request
Servers in the SSO domain – May be in same or different cell

21

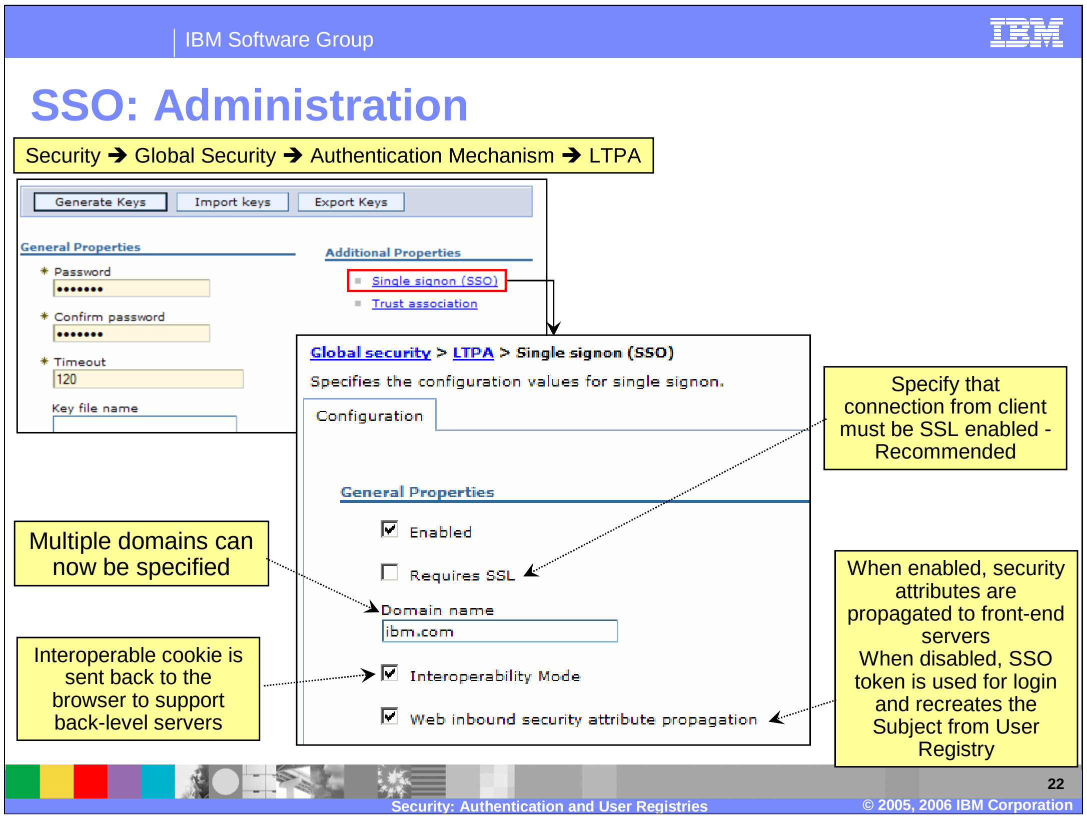
Security: Authentication and User Registries

© 2005, 2006 IBM Corporation

The picture shows the steps of how SSO works. Here there are two servers (which could be in different Network Deployment Cells) running different Web applications. With the SSO enabled in the LTPA authentication mechanism, the first request to a server in a SSO domain will return an LTPA token back to the client. The client will then send the LTPA token to any subsequent requests to any Server in the same SSO domain. The server will then use the LTPA token to validate the request without the need of re-challenging the client again.

The authenticating server issues an LTPA login token and sends it in an HTTP cookie named “LtpaToken” in the HTTP response.

This cookie is then sent by the client to each eligible server with each subsequent HTTP request. The server then uses the token to authenticate and authorize the request.



SSO: Administration

Security ➔ Global Security ➔ Authentication Mechanism ➔ LTPA

* Password: •••••••
* Confirm password: •••••••
* Timeout: 120
Key file name:

Global security > LTPA > Single signon (SSO)

Specifies the configuration values for single signon.

[✓] Enabled
[ ] Requires SSL
Domain name: ibm.com
[✓] Interoperability Mode
[✓] Web inbound security attribute propagation

Specify that connection from client must be SSL enabled - Recommended

Multiple domains can now be specified

Interoperable cookie is sent back to the browser to support back-level servers

When enabled, security attributes are propagated to front-end servers
When disabled, SSO token is used for login and recreates the Subject from User Registry

This panel shows the Administrative console panel where you enable SSO using the LTPA authentication mechanism. It is simple to enable SSO. While doing this, the SSL requirement between the client and the server could be enabled. It is recommended that SSL be enabled.

The domain names are also set in this panel. The different values of the domain names were explained few pages back.

V6 (similar to V5.1.1) supports 2 different cookies added to the HTTP response. These are called LtpaToken and LtpaToken2.

LtpaToken is used for SSO behavior in V5.1 and previous versions of the Application server. This token type is interoperable with those previous releases. This is generated when the **Interoperability mode** is enabled.

LtpaToken2 is used for security attribute propagation. The token type allows for multiple attributes specified in the token (mostly containing information to contact the original login server). These are used on servers on V5.1.1 and V6 release. This token is generated when **Web inbound security attribute propagation** option is enabled.

This option is used if you want information added during the login at a specific front-end server to propagate to other front-end servers. The SSO token does not contain any sensitive attributes, but does understand where the original login server exists in cases where it needs to contact that server to retrieve serialized information.

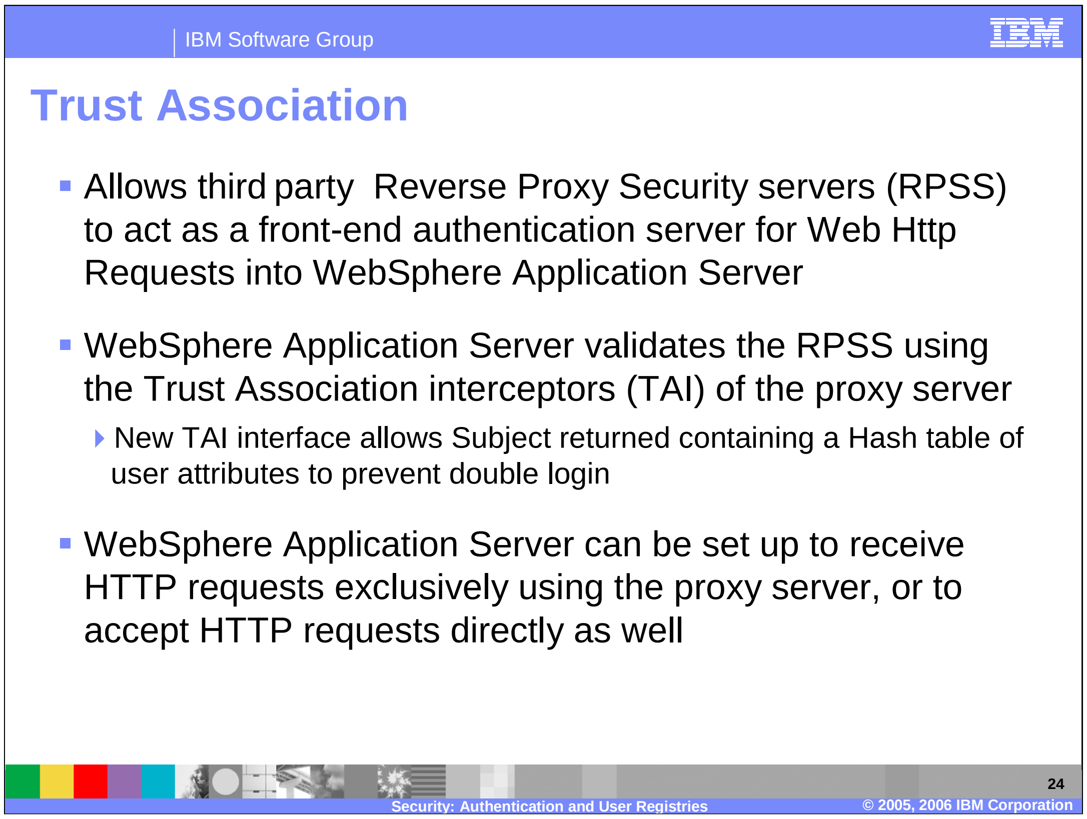
So, in short, when **Web inbound security propagation** is enabled, LtpaToken2 is generated. When **Interoperability mode** is enabled, LtpaToken is generated.

When both modes are enabled, both LtpaToken and LtpaToken2 are generated.

Section

Trust Association

The next section will discuss Trust Association

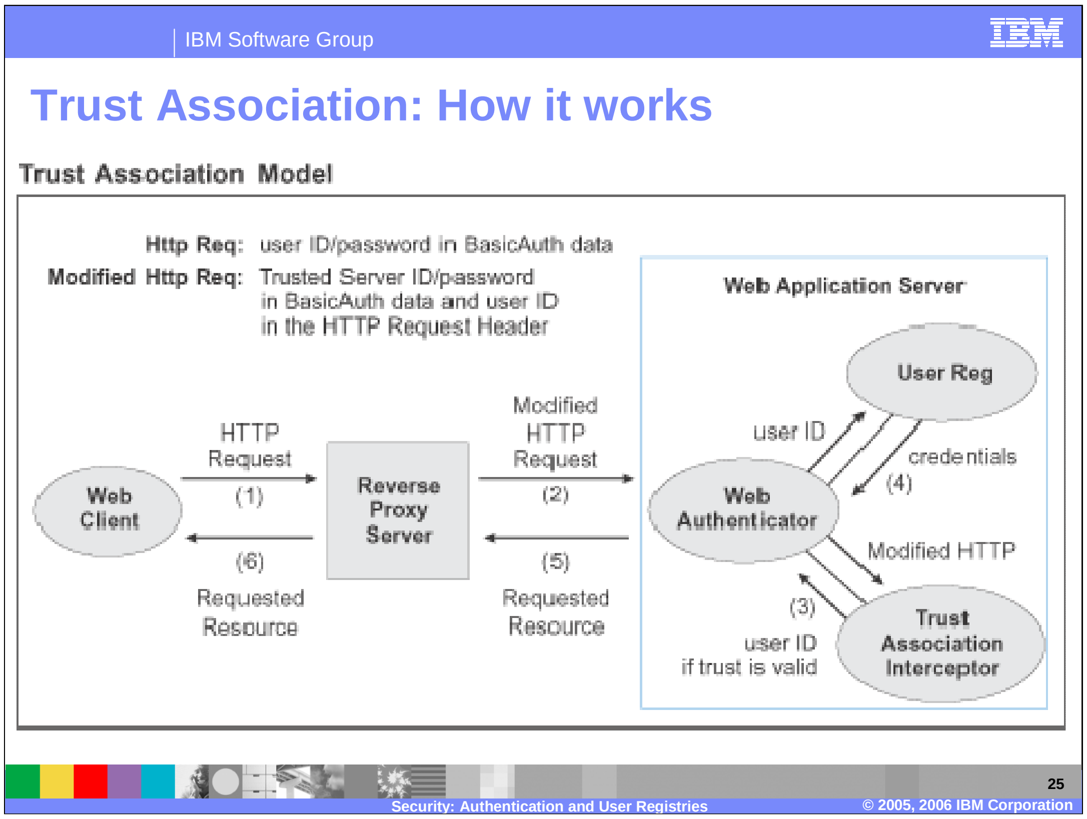


Trust Association

- Allows third party Reverse Proxy Security servers (RPSS) 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
  - New TAI interface allows Subject returned containing a Hash table of user attributes to prevent double login
- WebSphere Application Server can be set up to receive HTTP requests exclusively using the proxy server, or to accept HTTP requests directly as well

Trust Association applies for Web requests where some other server, like Reverse Proxy Security servers, will authenticate the request and forward the request to the Application Server. The Application Server will only validate the Reverse Proxy Security server through the Trust Association Interceptor. The RPSS server id must be a valid id within the Application Server User registry.

Examples of Reverse Proxy Security servers are IBM® Tivoli® Security Manager - WebSeal for Policy Director, and so on.


IBM Software Group

Trust Association: How it works

25

Security: Authentication and User Registries

© 2005, 2006 IBM Corporation

The picture shows the steps of how the Trust Association works in the following steps:

1) The web client sends the HTTP request to the proxy server.

2) The proxy server authenticates the user and sends a modified HTTP request that contains the user id and the proxy server’s ID and password. This information is sent in the HTTP request header.

3) The Trust Association Interceptor (TAI) is called by the Application Server to validate if the trust is valid. The modified HTTP request is passed to the TAI

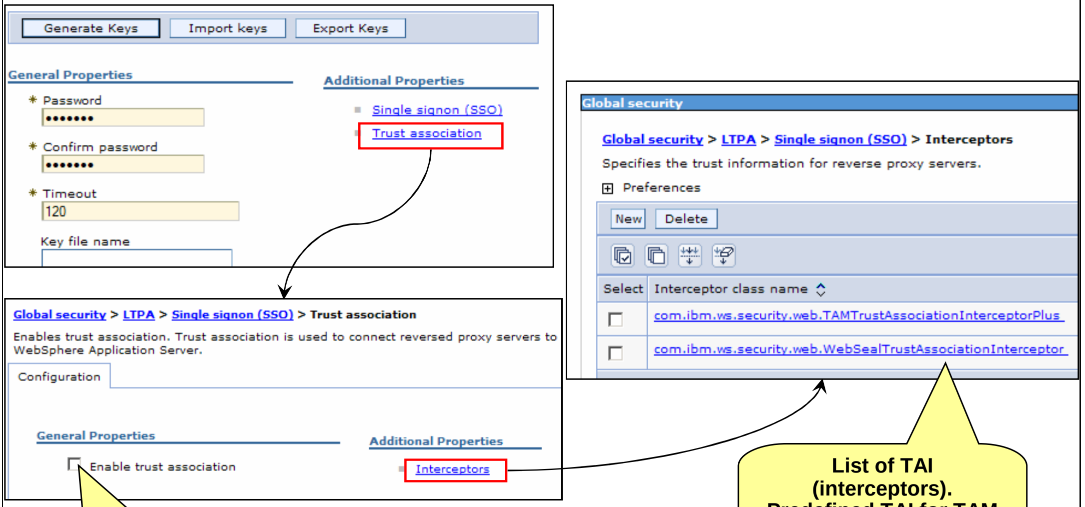
4) If TAI returns OK, then the Application server will create the user credential, looking at the user registry.

5) Application server sends the response back to the Proxy server.

6) The proxy server sends the response to the client.

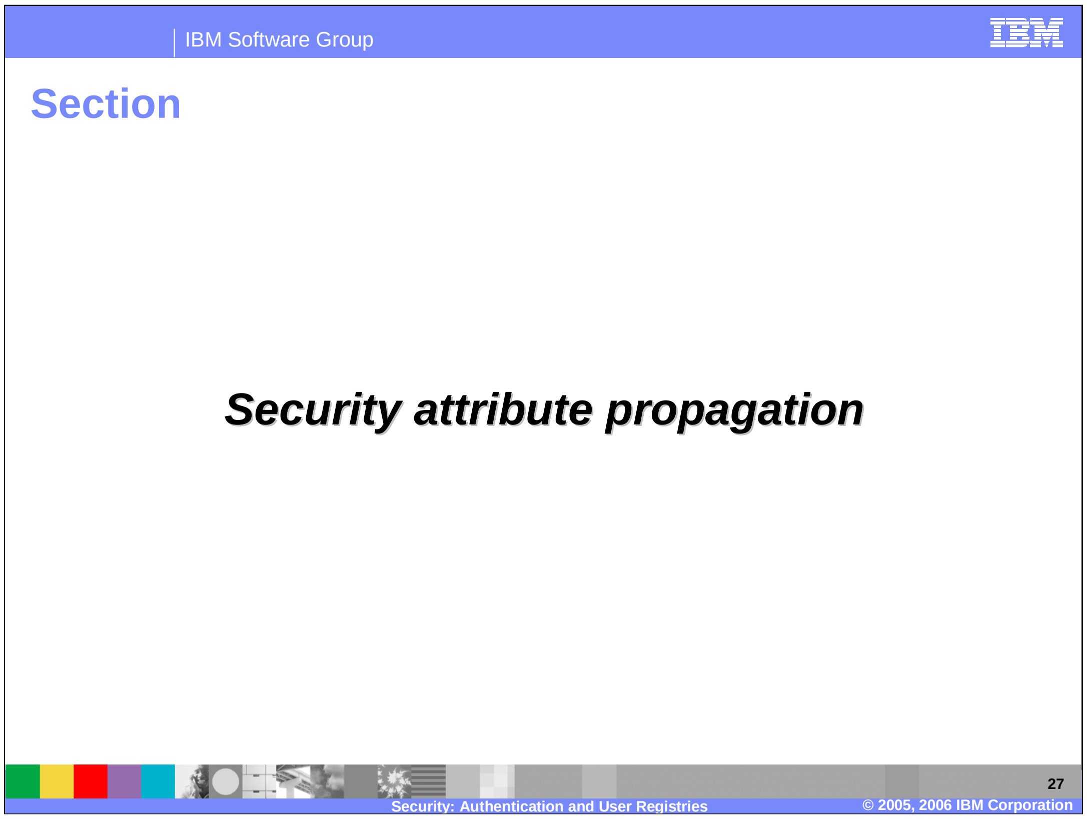
Trust Association – Administration

Security ➔ Global Security ➔ Authentication Mechanism ➔ LTPA)



Generate Keys
Import keys
Export Keys
General Properties
Additional Properties
Password
Confirm password
Timeout
120
Key file name
Single signon (SSO)
Trust association
Global security > LTPA > Single signon (SSO) > Trust association
Enables trust association. Trust association is used to connect reversed proxy servers to WebSphere Application Server.
Configuration
Enable trust association
Interceptors
Global security
Global security > LTPA > Single signon (SSO) > Interceptors
Specifies the trust information for reverse proxy servers.
Preferences
New
Delete
Select
Interceptor class name
com.ibm.ws.security.web.TAMTrustAssociationInterceptorPlus
com.ibm.ws.security.web.WebSealTrustAssociationInterceptor
Enable Trust Association
List of TAI (interceptors). Predefined TAI for TAM and WebSeal already included

This panel shows the Administrative console panel where you enable Trust Association using the LTPA authentication mechanism. It is simple to enable Trust Association. You will need to specify the Trust Association Interceptor, or TAI. Note that TAIs for WebSeal and TAM are already bundled with WebSphere Application Server V6 and listed in the panels.



IBM Software Group

Section

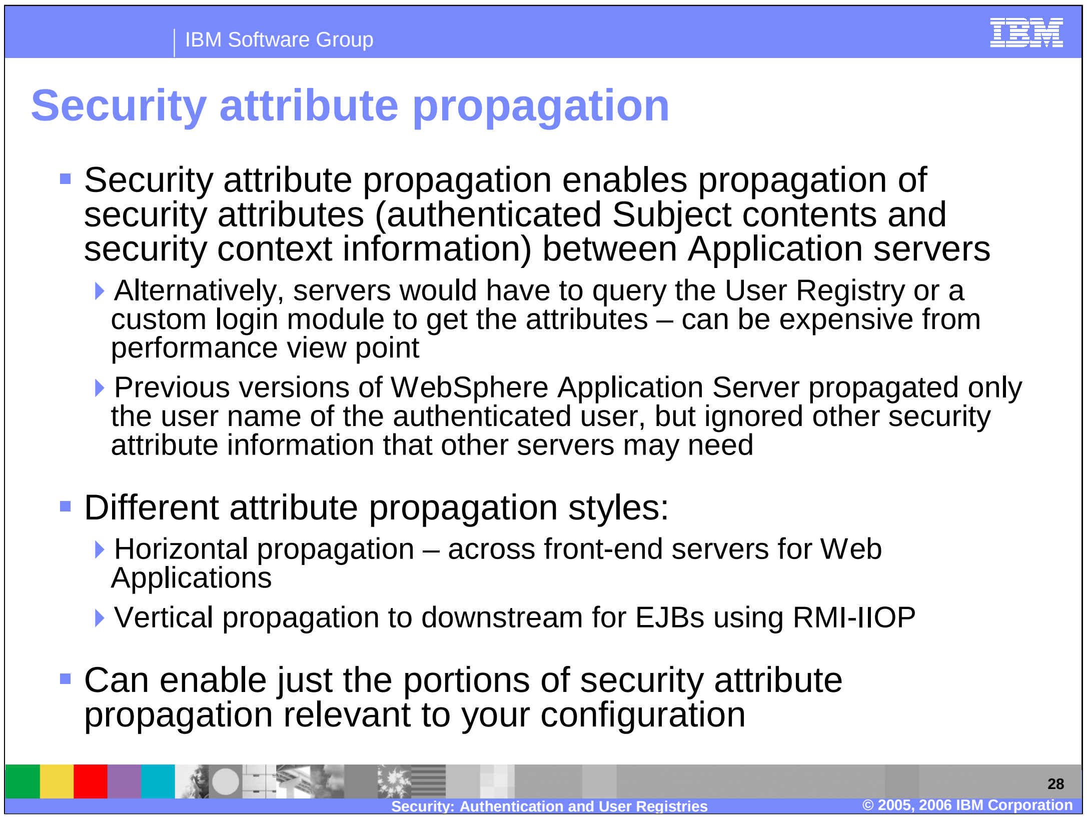
Security attribute propagation

27

Security: Authentication and User Registries

© 2005, 2006 IBM Corporation

The next section will discuss Security attribute propagation.



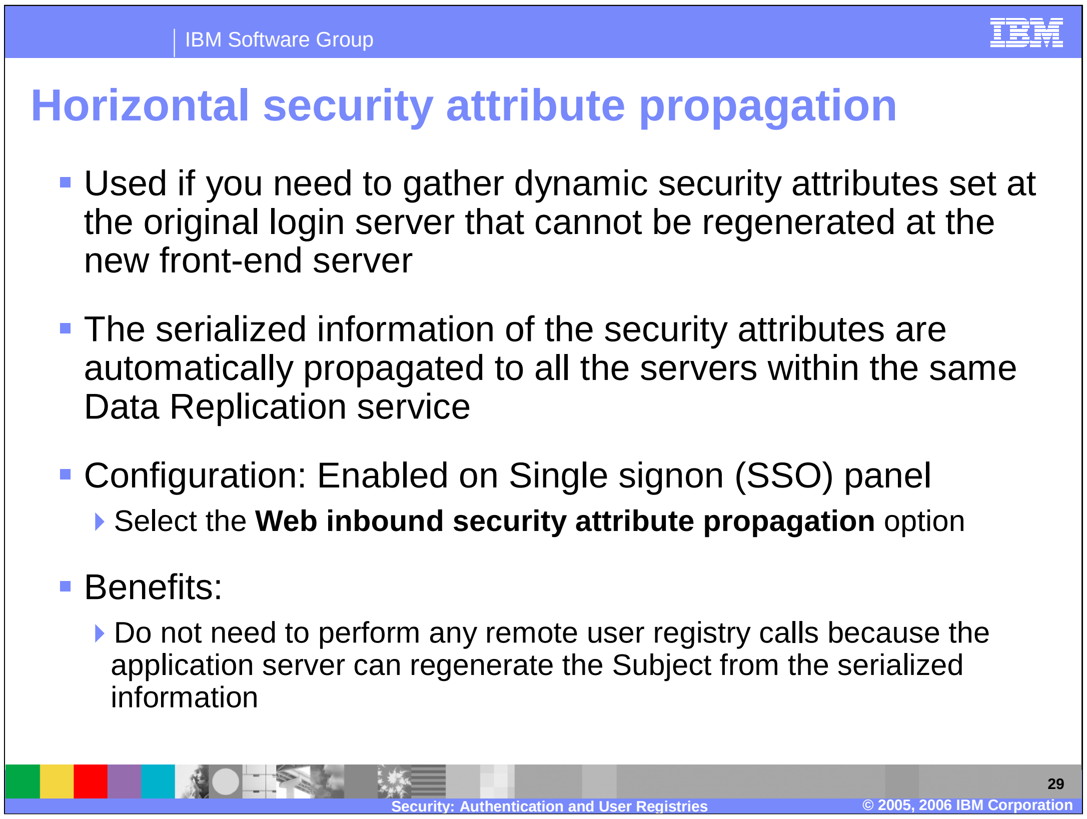
Security attribute propagation

- Security attribute propagation enables propagation of security attributes (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
  - Previous versions of WebSphere Application Server 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
  - Vertical propagation to downstream for EJBs using RMI-IIOP
- Can enable just the portions of security attribute propagation relevant to your configuration

Security attribute propagation enables WebSphere Application Server to transport security attributes from one server to another in your configuration. WebSphere Application Server might obtain these security attributes from either an enterprise user registry, which queries static attributes, or a custom login module, which can query static or dynamic attributes.

Custom tokens added by other servers and front-end are not used by WebSphere Application Server for authorization or authentication.

The server propagates only the objects within the Subject that it can serialize. The server propagates custom objects on a best-effort basis.


IBM Software Group

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
- Configuration: Enabled on Single signon (SSO) panel
  - Select the **Web inbound security attribute propagation** option
- Benefits:
  - Do not need to perform any remote user registry calls because the application server can regenerate the Subject from the serialized information

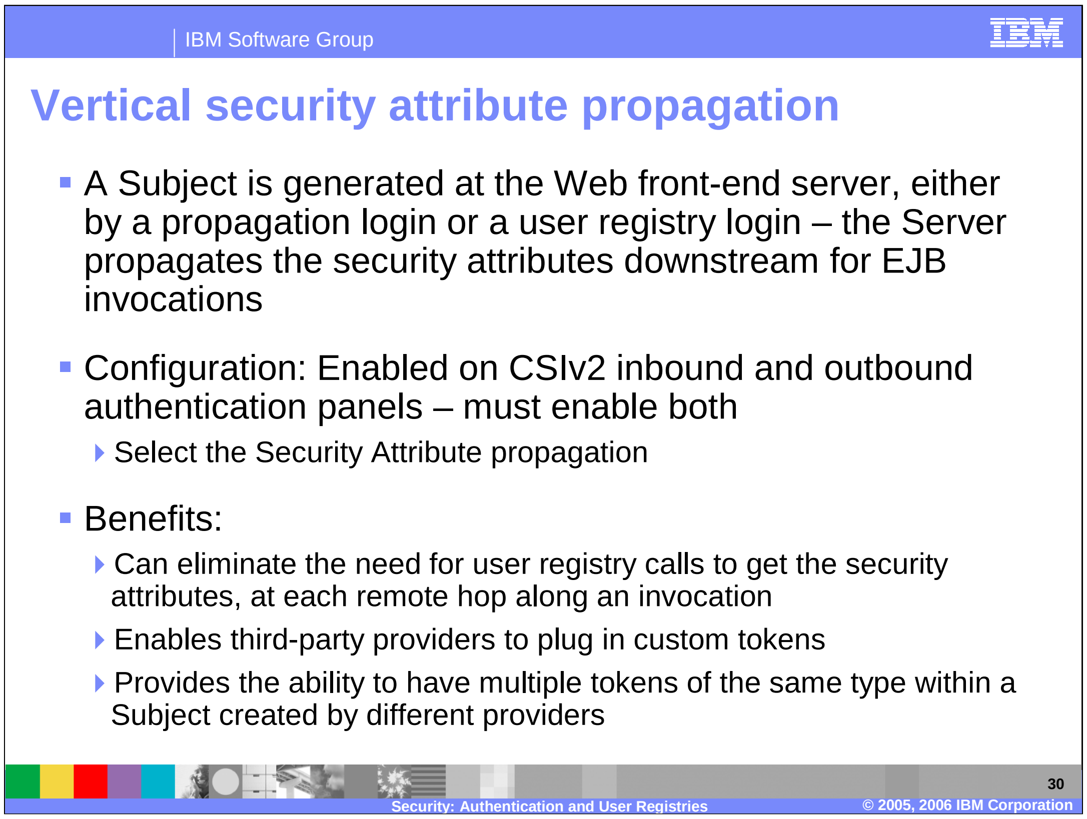
29

Security: Authentication and User Registries

© 2005, 2006 IBM Corporation

The horizontal security attribute propagation is used with SSO where the security attributes are propagated to all the front end servers that participate in the SSO domain. If the servers are part of the same Data replication service, the security attributes are automatically propagated to other servers.
This option is enabled in the SSO panel of the Administration console.

The key benefit is the servers do not need to go back to the user registry to get the attributes.

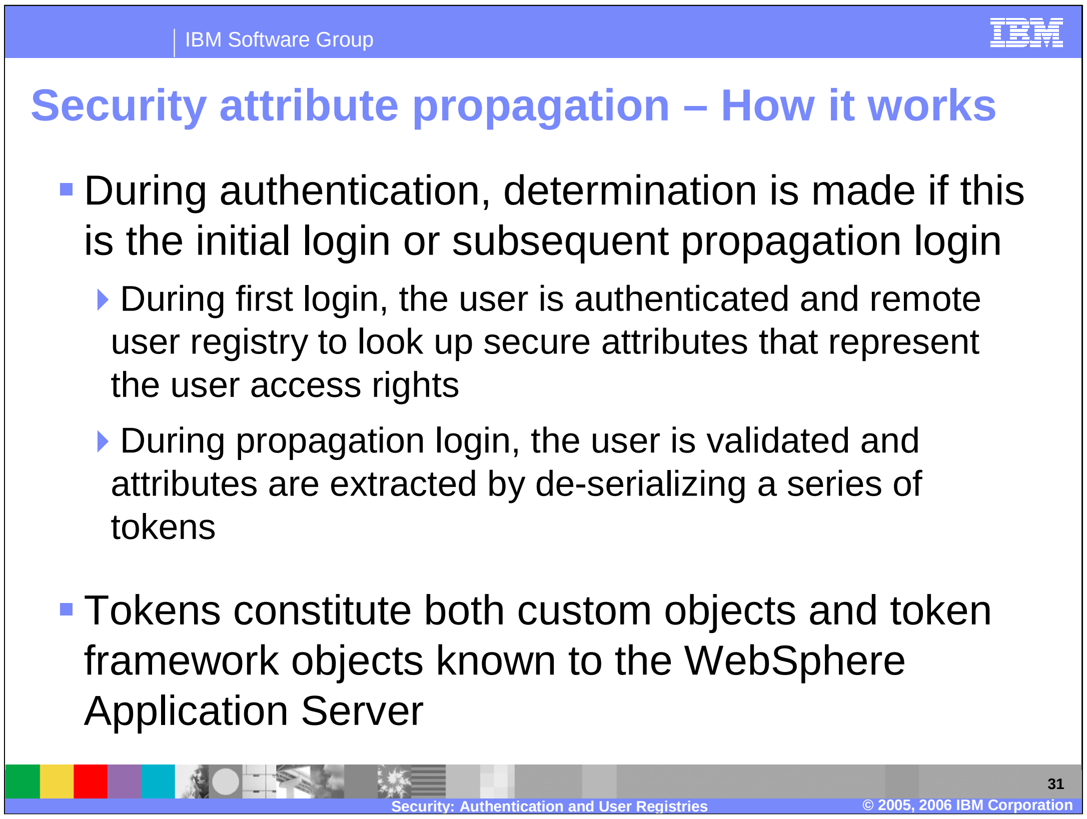


Vertical security attribute propagation

- A Subject is generated at the Web front-end server, either by a propagation login or a user registry login – the Server propagates the security attributes downstream for EJB invocations
- Configuration: Enabled on CSIv2 inbound and outbound authentication panels – must enable both
  - Select the Security Attribute propagation
- 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

Vertical security propagation of attributes is propagating of attributes downstream for EJB invocations. The downstream servers can now get the attributes rather than retrieving them by going to the user registry.

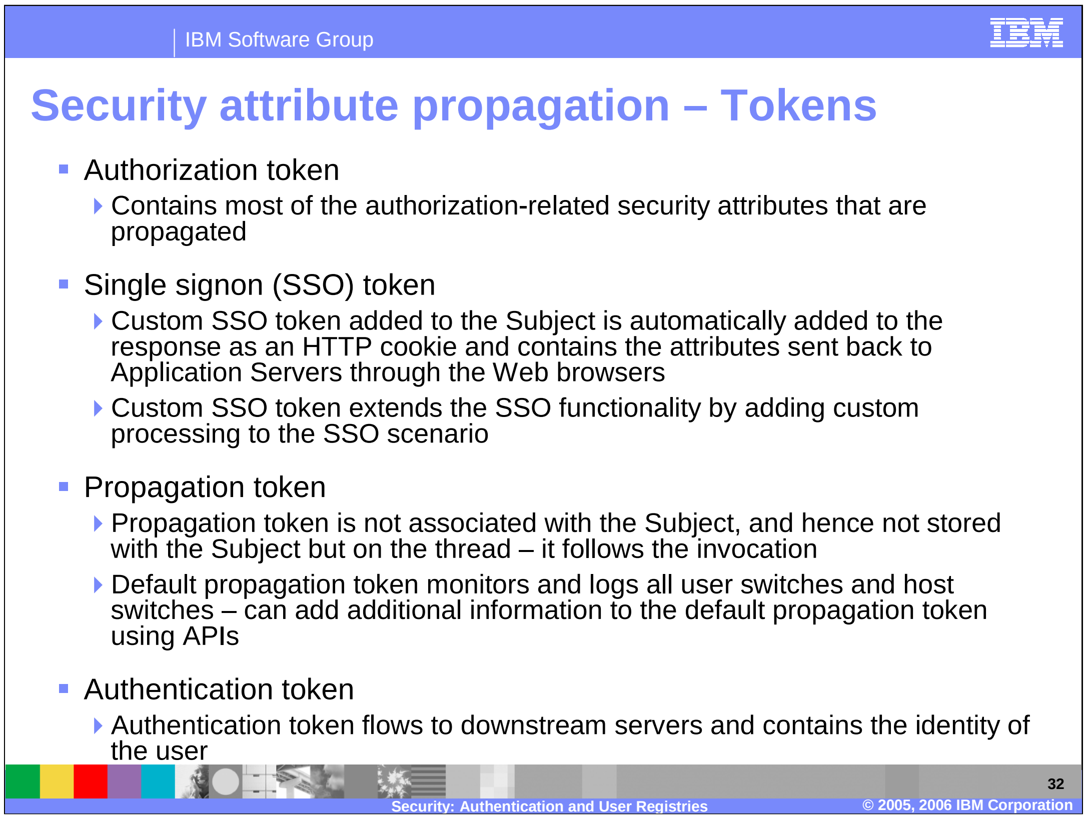
CSIv2 is the authentication mechanism used for EJBs, and hence the security propagation is enabled in the CSIv2 inbound and outbound panels in the Administrator console.



Security attribute propagation – How it works

- During authentication, determination is made if this is the initial login or subsequent propagation login
  - During first login, the user is authenticated and remote user registry to look up secure attributes that represent the user access rights
  - During propagation login, the user is validated and attributes are extracted by de-serializing a series of tokens
- Tokens constitute both custom objects and token framework objects known to the WebSphere Application Server

During the initial login, the security attributes are serialized and stored in tokens. During subsequent logins, referred to as the propagation login on this page, the security attributes are extracted by de-serializing the tokens containing the attributes. There are several types of tokens, described in the next page.


IBM Software Group

IBM

Security attribute propagation – Tokens

- Authorization token
  - Contains most of the authorization-related security attributes that are propagated
- Single signon (SSO) token
  - Custom SSO token added to the Subject is automatically added to the response as an HTTP cookie and contains the attributes sent back to Application Servers through the Web browsers
  - Custom SSO token extends the SSO functionality by adding custom processing to the SSO scenario
- Propagation token
  - Propagation token is not associated with the Subject, and hence not stored with the Subject but on the thread – it follows the invocation
  - Default propagation token monitors and logs all user switches and host switches – can add additional information to the default propagation token using APIs
- Authentication token
  - Authentication token flows to downstream servers and contains the identity of the user

32

Security: Authentication and User Registries

© 2005, 2006 IBM Corporation

The 4 different types of tokens used in the Security Attribute propagation are described here.

Authorization token:

Service providers can use custom authorization token implementations to isolate their data in a different token; perform custom serialization and de-serialization; and make custom authorization decisions using the information in their token at the appropriate time. For information on how to use and implement this token type, see Default PropagationToken and Implementing a custom PropagationToken in the V6 Information Center.

Single signon token:

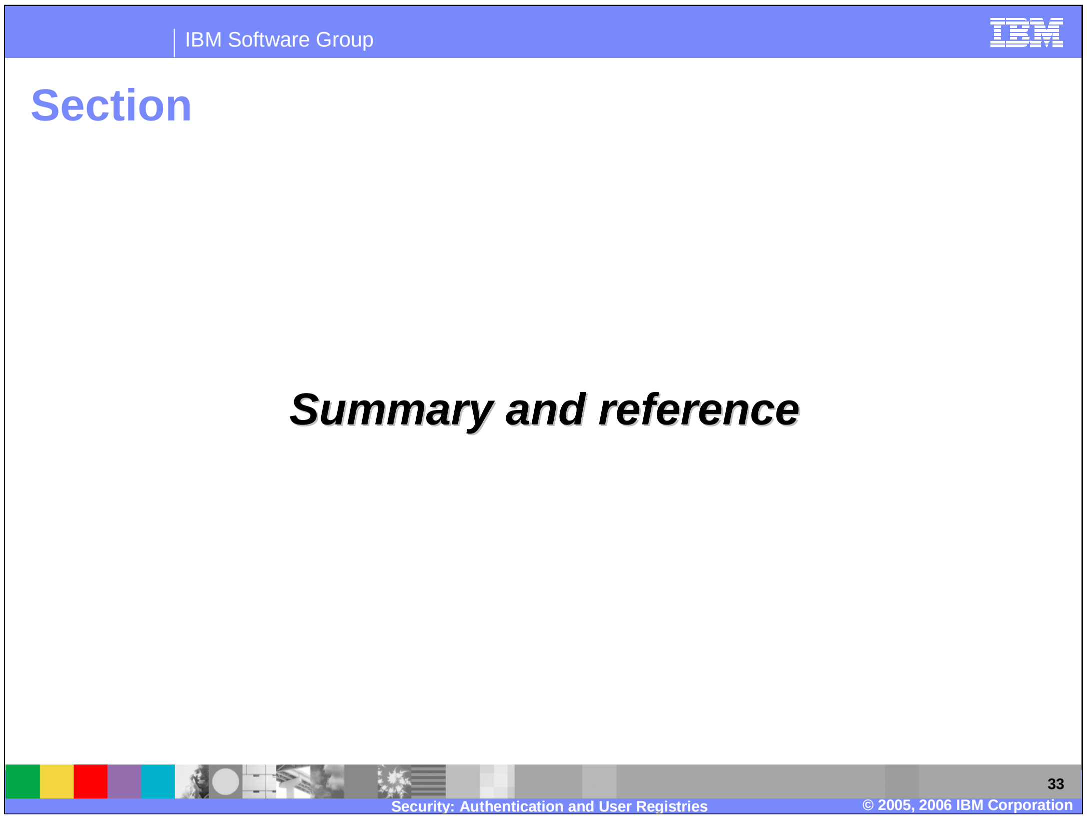
Single signon token is added to the Subject. This is sent as an HTTP cookie to all the servers. WebSphere Application Server defines a default SingleSignonToken in a cookie named LtpaToken2. When using SSO token, it is recommended to use SSL.

Propagation token:

This allows servers in the invocation hop to add information that can be retrieved down stream. This token is not associated with the Subject, since the subject may change. Instead, it is stored with the thread. The default propagation token monitors and logs all user switches and host switches. You can add additional information to the default propagation token using the WSSecurityHelper APIs. To retrieve and set custom implementations of a propagation token, you can use the WSSecurityPropagationHelper class.

Authentication token:

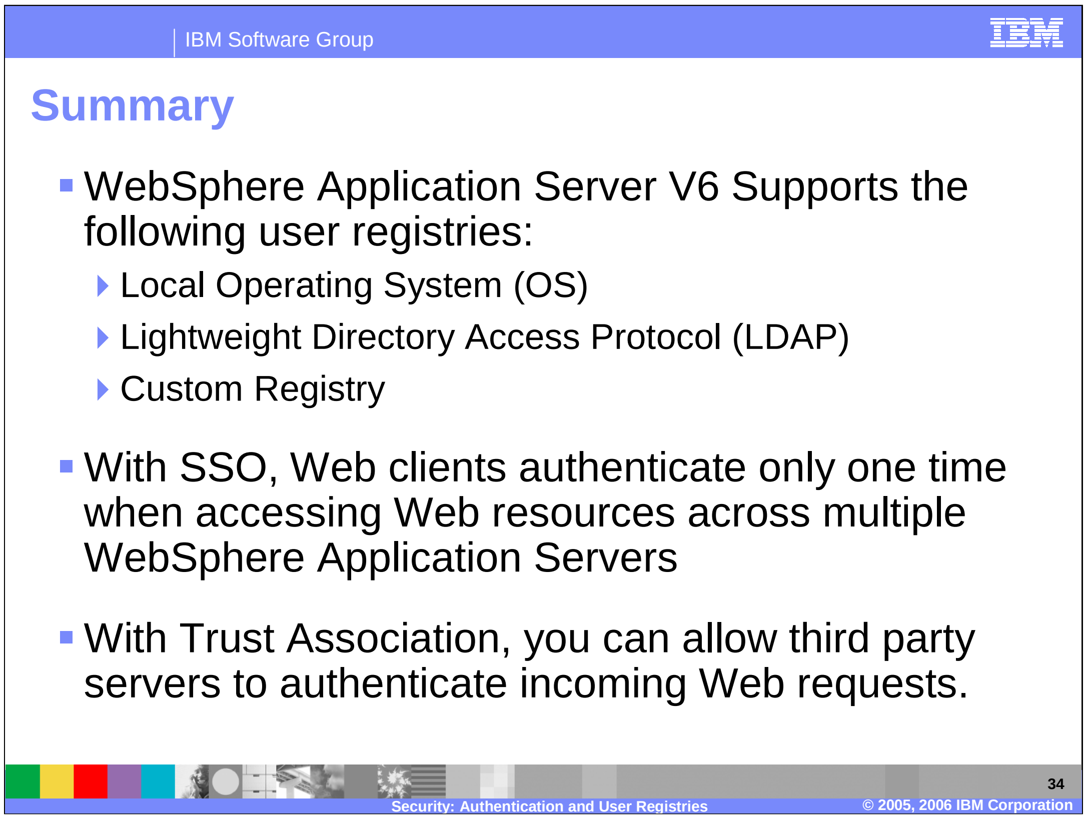
The authentication token flows to downstream servers and contains the identity of the user. This token type is typically reserved for internal WebSphere Application Server purposes. You can add this token to the Subject and the token is propagated using the getBytes method of the token interface. Custom authentication token is used solely for the purpose of the service provider that adds it to the Subject. WebSphere Application Server



Section

Summary and reference

The next section will discuss the Summary of this presentation.



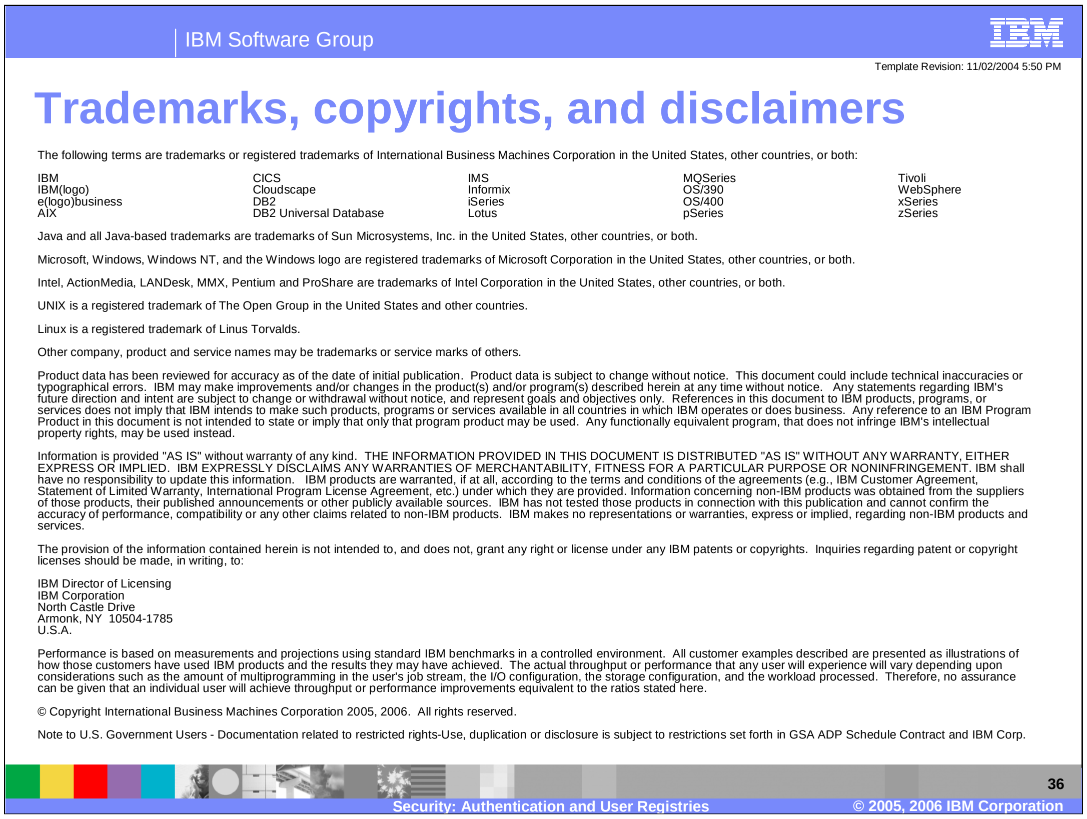
Summary

- WebSphere Application Server V6 Supports the following user registries:
  - Local Operating System (OS)
  - Lightweight Directory Access Protocol (LDAP)
  - Custom Registry
- With SSO, Web clients authenticate only one time when accessing Web resources across multiple WebSphere Application Servers
- With Trust Association, you can allow third party servers to authenticate incoming Web requests.

In summary, this presentation has focused on WebSphere Application Server V6, which provides rich support for Authentication, including SSO and Trust Association. It supports many User registries that can be used based on your needs.

References: Information Center links

- Authorization tokens:
  - Default propagation token
  - Implementing a custom propagation token
- Authentication tokens:
  - Default authentication token
  - Implementing a custom authentication token
- Security attribute propagation:
  - http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/csec_secattributeprop.html



Trademarks, copyrights, and disclaimers