Powerful and secure infrastructures with

WebSphere Application Server for z/OS

WebSphere V5 and J2EE 1.3: Security Overview

Redbooks

International Technical Support Organization

Holger Wunderlich
wunderl@us.ibm.com
(thanks to my residents)

Trademarks

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

Agenda

- Introduction, terminology and overview
- J2EE security concepts
- New in J2EE 1.3 and WebSphere Application Server V5
  - Increased flexibility
  - Improved interoperability
  - Java 2 security
  - JAAS

Terminology

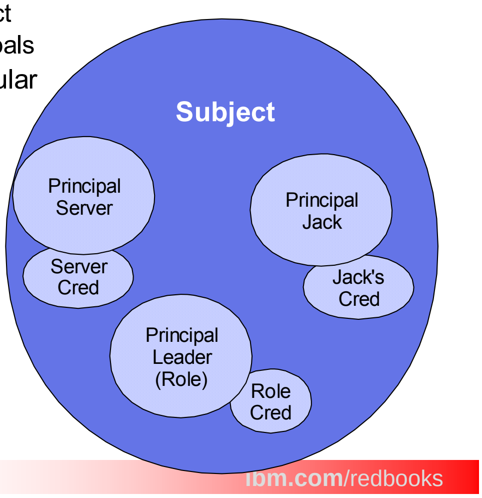
- Basic security terminology
  - Identification
    - Examples: userId, distinguished name
  - Authentication
    - How do we know you're really Adam?
  - Authorization
    - OK Adam, you may freely eat of every tree in the garden; but of the tree of the knowledge of good and evil you shall not eat.
  - Confidentiality
    - Protection or messages and other data from observation by unauthorized entities
  - Integrity
    - Assurance that a message has not been altered in transmission
  - Non-repudiation
    - The woman whom you gave to be with me, she gave me fruit from the tree, and I ate--blame her!

Security APi’s reference

- CertPath 1.0 from J2EE V1.4
- Java Cryptography Extension (JCE) 1.2.1
- Java Authentication and Authorization Services (JAAS) 1.0
- Java Secure Socket Extension (JSSE) 1.0.2
- Public Key Cryptography Standards 1.0
- SOAP-Sec 1.0
- XML Digital Signature 0.9.0
- Common Secure Interoperability Version 2 (CSIv2)

Some Java terminology

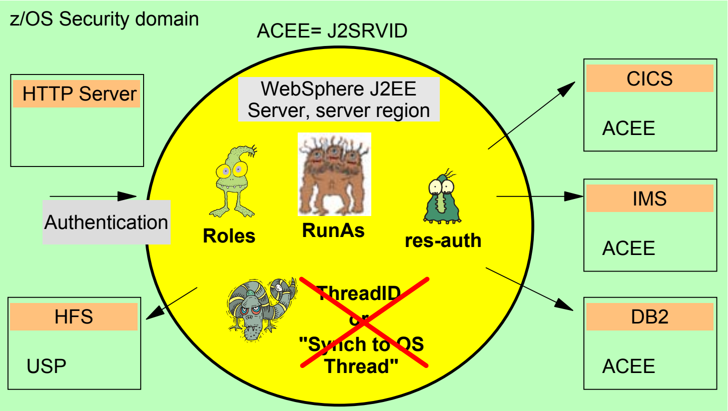
- Subject an abstraction for a user or caller
  - ironically, the subject is an object
  - may contain one or more principals
- The principal is a user in a particular context--one principal at a time may be active on the thread
- A credential is what the security mechanism returned after the user authenticated



J2EE Security Model

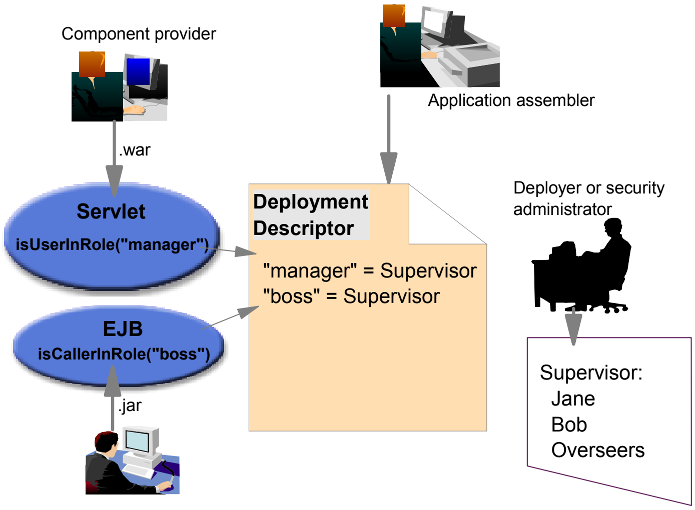
- Declarative
  - Specified outside the program code - in the deployment descriptor
  - Controls how authentication is performed and who is authorized to which resources
  - Roles, run-as, res-auth
  - Java 2 security
- Programmatic
  - Program code performs authentication and/or authorization
  - JAAS

The monsters still with us



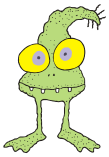
- Within the J2EE server, security identities can be set/changed but the ACEE stays equal to the RACF userid of the J2EE application servant region.
- ACEE can no longer be changed via "Sync to OS Thread" option

Roles



Roles

- Roles are defined in the deployment descriptor
  - Web components (URL / method)
  - EJB components
- RACF
  - profile in classes GEJBROLE
  - or EJBROLE control use of the role
  - The APPLDATA field in the profile specifies a RACF userid to be associated with the role when RunAs(role)
  - Although the classes names are GEJBROLE/EJBROLE, the profiles protect all resources, not just only EJBs
- Customer User Registry
  - Roles defined in authorization bindings



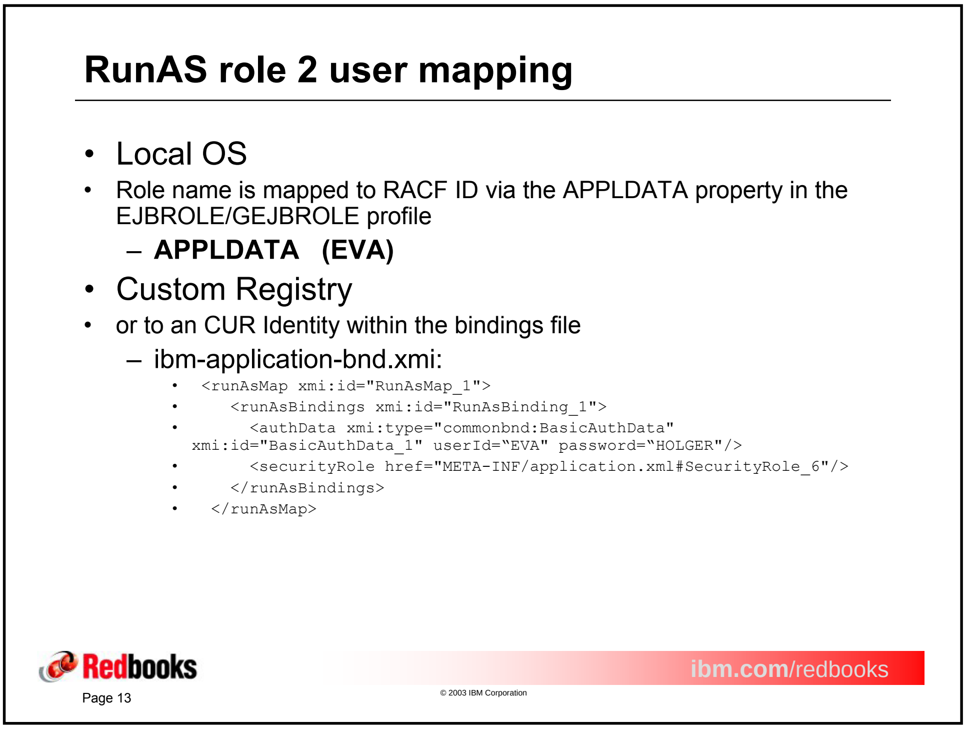
Roles

Run-as RunAS are two monsters with many heads

- RunAs sets the principal
- used to run this method and for downstream propagation
  - Caller (the default)
    - Run this method with the identity of the
    - user who instantiated me
  - Server (EJB Container only)
    - Run this method with the identity of the
    - server on which I was instantiated
  - Role
    - Run this method with the RACF ID associated with this
    - roles RACF EJBROLE/GEJBROLE profile

Identity propagation

- run-as
  - run-as
  - Included in J2EE 1.3 (EJB 2.0, Servlet 2.3) specification
  - Defaults to caller
  - Can be specified as a role-name
  - Found in Web application or EJB deployment descriptor
  - Applies to component and all its methods
- runAs
  - IBM WebSphere extension to the programming model
  - Introduced in V4, supported in V5
  - Applies to EJB only
  - Defaults to caller
  - Can be specified as Server or a role-name
  - Found in IBM extension to EJB deployment descriptor
  - Can be applied separately to each method

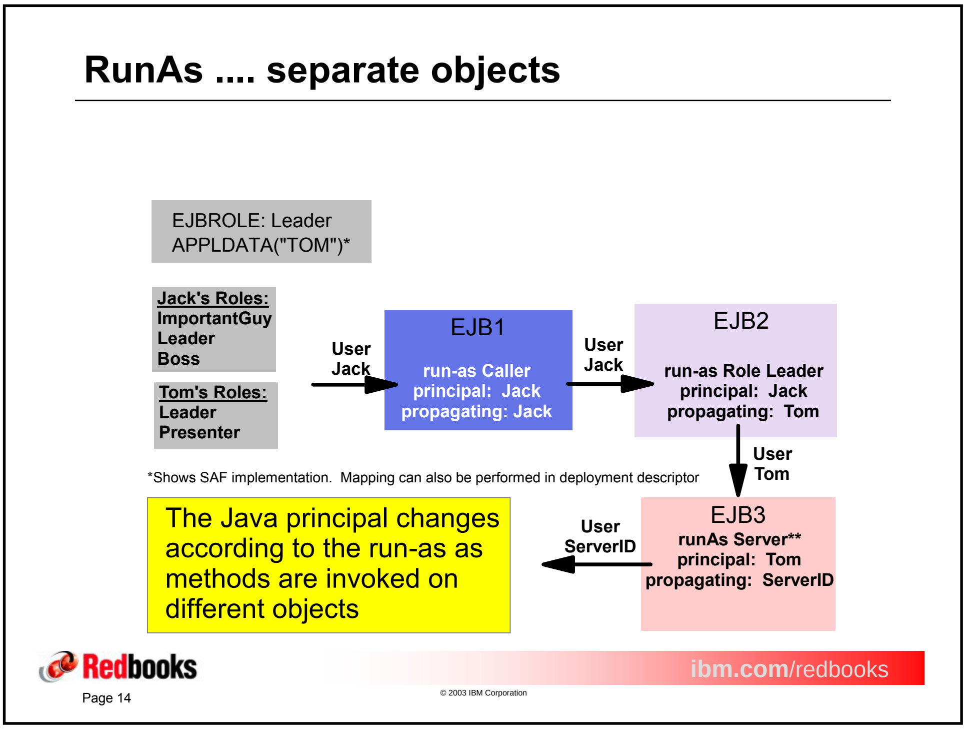


RunAS role 2 user mapping

- Local OS
- Role name is mapped to RACF ID via the APPLDATA property in the EJBROLE/GEJBROLE profile
  - **APPLDATA (EVA)**
- Custom Registry
- or to an CUR Identity within the bindings file
  - ibm-application-bnd.xmi:

```
•   <runAsMap xmi:id="RunAsMap_1">
•      <runAsBindings xmi:id="RunAsBinding_1">
•        <authData xmi:type="commonbnd:BasicAuthData"
  xmi:id="BasicAuthData_1" userId="EVA" password="HOLGER"/>
•        <securityRole href="META-INF/application.xml#SecurityRole_6"/>
•      </runAsBindings>
•    </runAsMap>
```

Redbooks ibm.com/redbooks



RunAs .... separate objects

EJBROLE: Leader
APPLDATA("TOM")*

*Shows SAF implementation. Mapping can also be performed in deployment descriptor

The Java principal changes according to the run-as as methods are invoked on different objects

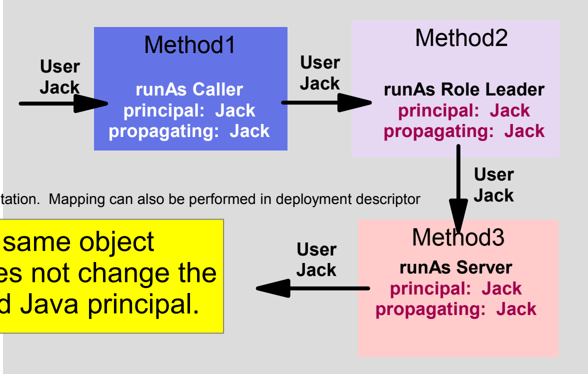
Redbooks ibm.com/redbooks

Identity propagation: same object

EJBROLE: Leader
APPLDATA("TOM")*

Jack's Roles:
ImportantGuy
Leader
Boss

Tom's Roles:
Leader
Presenter

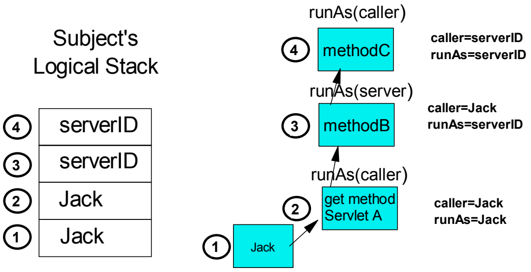


*Shows SAF implementation. Mapping can also be performed in deployment descriptor

Within the same object RunAs does not change the propagated Java principal.

RunAS stack

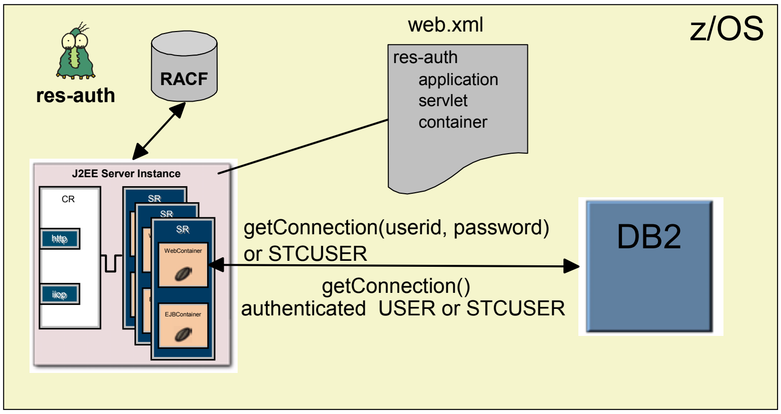
RunAs "stack"



caller identity is important for:
-methodPermissions
-isCallerinRole(x)
-getCallerPrincipal()

RunAs Identity is important for:
-downstream authorizations
-outbound identity
-syncToOSThread when enabled

Res-auth



Container: container supplies userid (and password if required) from execution environment or configuration settings

Application/Servlet: application must specify userid and password

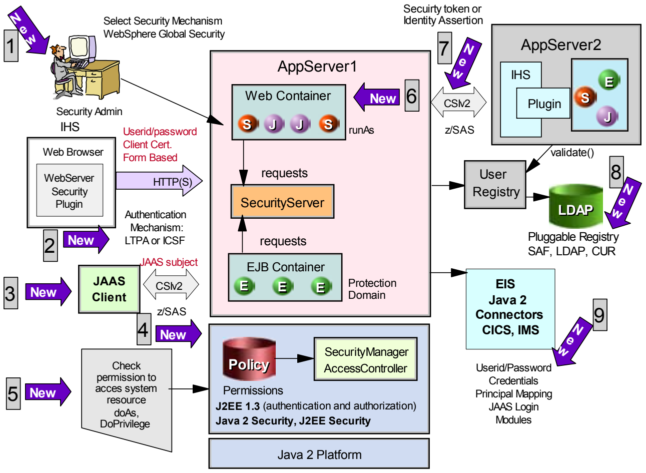
Summary of J2EE security concepts

- People often work in various roles
- Application developer defines roles required for use
- Application assembler defines what identity should be used when running (RunAs)
- Application assembler can map application roles to organizational roles using role references
- Security administrator defines roles as profiles in the RACF EJBROLE class
- Permits users & groups to EJBROLE profiles
- Security can be container based (declarative) or application based (programmatic)

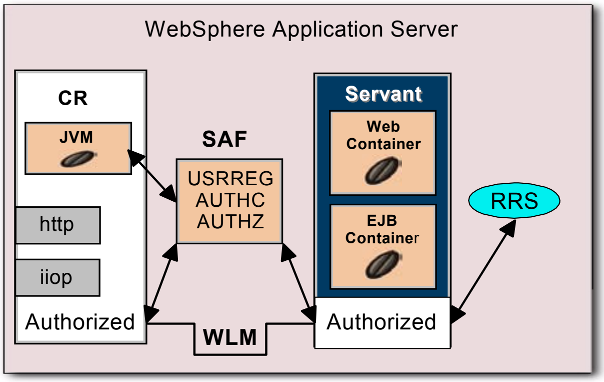
New in security for V5

- J2EE 1.3 compliant
- Role based Authorization Enhancements for J2EE 1.3
- Java 2 Security
  - Policy based access to System resources
- JAAS programming model support
  - Provides means for restricting running code based on authenticated user
- Support for CSIv2 security protocol (conformance level 0)
  - Provides Interoperability between different vendor's application servers
  - SAS (for WAS Distributed) and z/SAS (for WAS z/OS) protocol supported for backward compatibility (4.0.x and before)
- JAAS, Java 2 Security and CSIv2 required by J2EE 1.3 CTS
- WebSphere Security Administration
  - Role Based Authorization for Administrative tasks
  - Role Based Authorization for Naming
  - All Security configuration handled by admin console Security Center
    - No longer necessary to edit SAS properties files, and other security files
- Consistency and commonality between WebSphere on Distributed platforms and z/OS platform

WebSphere Security: The big picture

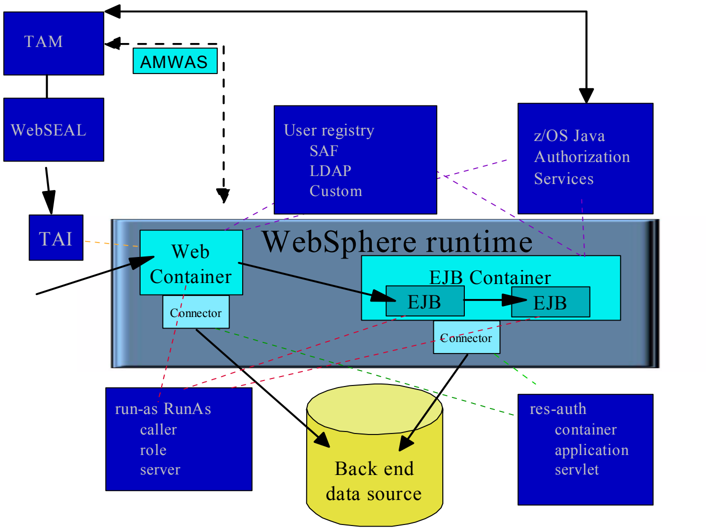


z/OS Implementation



Fenced Server Regions

V5 Authc/Authz & delegation options



Connector security

Connectors	ThreadIdentity	ThreadSecurity
IMS Connector local configuration	ALLOWED	Not Supported
IMS Connector remote configuration	NOTALLOWED	Not Supported
CTG CICSECIConnector local configuration	ALLOWED	Not Supported
CTG CICSECIConnector remote configuration	NOTALLOWED	Not Supported
IMS JDBC Connector	REQUIRED	True
RRA DB2 390 Local JDBC Provider	ALLOWED	True

<table>
<tr><td colspan="8">Container-managed alias specified?</td></tr>
<tr><td colspan="4">NO</td><td colspan="4">YES</td></tr>
<tr><td colspan="4">Connector Allows or Requires Thread Identity?</td><td colspan="4">Connector Requires Thread Identity?</td></tr>
<tr><td>NO</td><td colspan="3">YES</td><td>NO</td><td colspan="3">YES</td></tr>
<tr><td rowspan="6">Processing is dependent on connector:
• may throw exception
• may default to connector user/pswd custom properties</td><td colspan="3">Connector Requires OS Thread Security?</td><td rowspan="6">Use specified alias</td><td colspan="3">Connector Requires OS Thread Security?</td></tr>
<tr><td>NO</td><td colspan="2">YES</td><td>NO</td><td colspan="2">YES</td></tr>
<tr><td rowspan="4">Use RunAs user identity associated with current thread</td><td colspan="2">Server Sync-To-Thread enabled?</td><td rowspan="4">Use RunAs user identity associated with current thread</td><td colspan="2">Server Sync-To-Thread enabled?</td></tr>
<tr><td>NO</td><td>YES</td><td>NO</td><td>YES</td></tr>
<tr><td>Use Server identity</td><td>Use RunAs user identity associated with current thread</td><td>Use Server identity</td><td>Use RunAs user identity associated with current thread</td></tr>
</table>

Redbooks

ibm.com/redbooks

It finally talks to others!

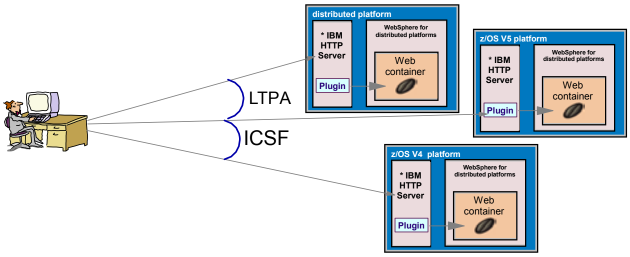
- Two suites of function available
  - zSAS
    - Compatible with z/OS and OS/390 V4 and V5 servers
    - Compatible with WebSphere AE V4 clients (basic authentication over SSL)
    - Compatible with WebSphere AE V4 servers (server-wide configured userid/password only)
  - CSIv2
    - Compatible with all WebSphere V5 servers
    - Provides support for asserted identity
- These are available concurrently

Redbooks

ibm.com/redbooks

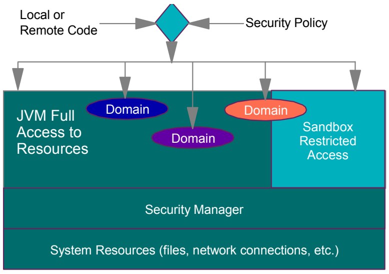
"Authentication mechanisms"

- SSO
  - SWAM
    - No security token created
  - LTPA
    - Security token created by WebSphere using configured keys
  - ICSF
    - Security token created by ICSF using keys from SAF keyring
- Interoperability/migration



Java 2 Security

- Generally code based vs. Subject- Based Authorization
- Access Control
  - Policy
    - defines permissions
    - grant to code based on the location and signer(s)
    - Dynamic, multi-level
      - Cell, Server, Application, Connector
- Each class belongs to one Protection Domain
  - Protection Domain
    - Code Source
      - Location (URL)
      - Signers
    - Permissions
- Filters prevent abuse by applications

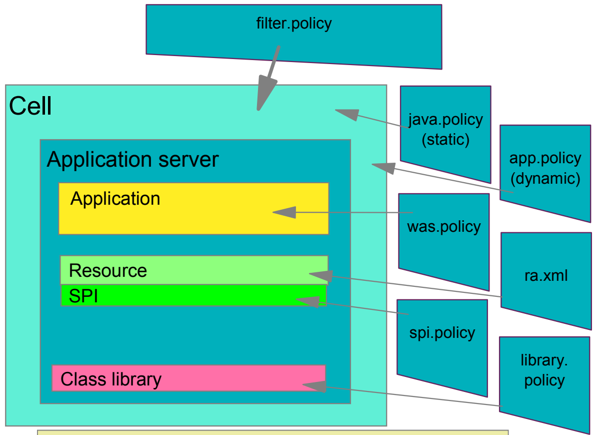

Local or Remote Code
Security Policy
JVM Full Access to Resources
Domain
Domain
Domain
Sandbox Restricted Access
Security Manager
System Resources (files, network connections, etc.)

Why?

Servant identity must have permission to all OS resources required by application

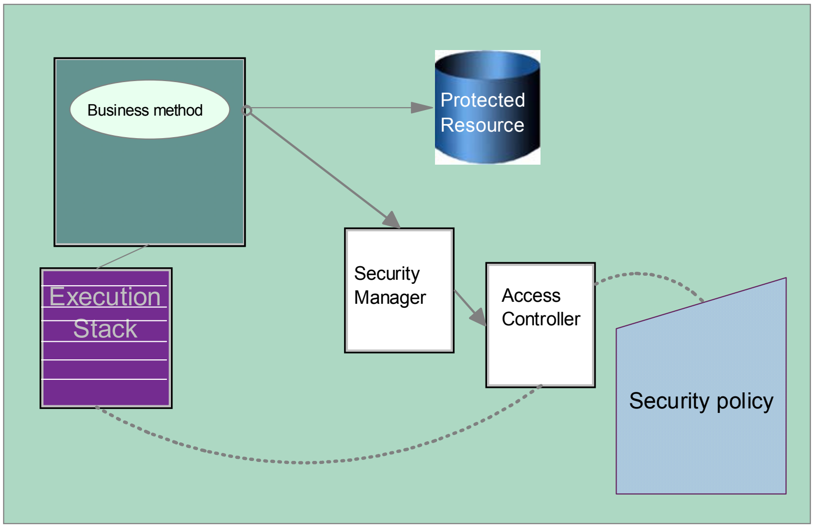
Java 2 Security allows for differentiation among applications

Policy files



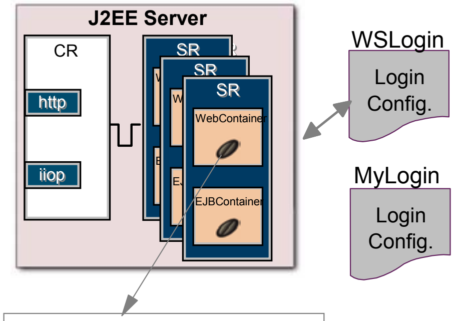
```
grant codebase "file:/u/tomhac/lib/*", {
    permission java.io.FilePermission "log.txt", "write";
};
```

Security Manager/Access Controller



JAAS

- Container (declarative) and application (programmatic) security
- Multiple login configurations
  - Each having one or more modules
    - Required
    - Requisite
    - Sufficient
    - Optional
  - Default is WSLogin
- Replaces SSOAuthenticator
  - Deprecated, but still supported
- Introduces Subject-based permissions
  - Class x can open file y when running under Subject with principal z.

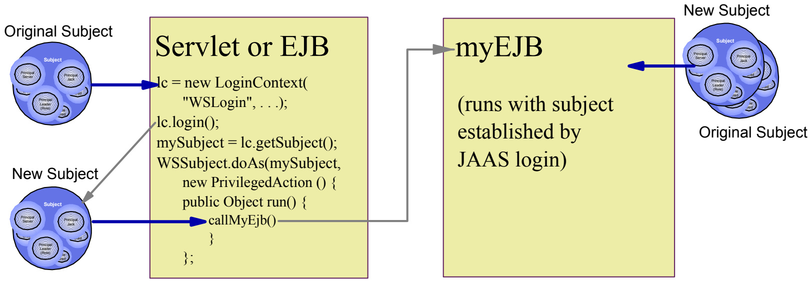


```
lc= new LoginContext (
  "WSLogin,"
  new WSCallbackHandlerImpl
     (userid, "myrealm", password));
```

```
grant codebase "file:/u/tomhac/lib/*",
        Principal myPrincipal "TOMHAC" {
    permission java.io.FilePermission "log.txt", "write";
};
```

doAS() makes JAAS login useful

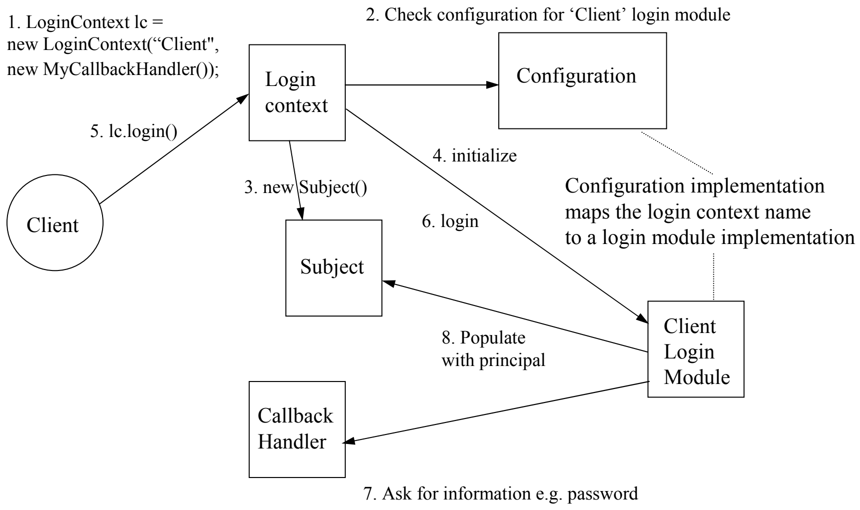
- Establishes new subject on the thread
  - Java Subject won't survive a doPrivileged() call
    - No way to call an EJB using the new subject
  - WebSphere WSSubject invented to overcome this deficiency



```
lc = new LoginContext(
      "WSLogin", . . .);
lc.login();
mySubject = lc.getSubject();
WSSubject.doAs(mySubject,
      new PrivilegedAction () {
      public Object run() {
          callMyEjb()
          }
      };
```

(runs with subject established by JAAS login)

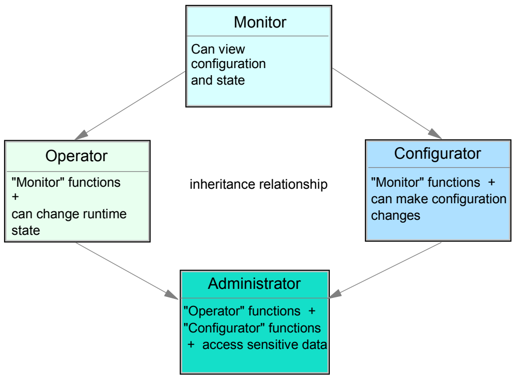
JAAS Components


1. LoginContext lc =
new LoginContext(“Client",
new MyCallbackHandler());
2. Check configuration for ‘Client’ login module
Login context
Configuration
5. lc.login()
4. initialize
3. new Subject()
Client
Configuration implementation
maps the login context name
to a login module implementation
6. login
Subject
Client Login Module
8. Populate
with principal
Callback Handler
7. Ask for information e.g. password

doAs() vs. run-As

	run-as	doAs()
type	declarative	programmatic
scope	method (runAs) or component (run-as)	run() method
values	caller (default) role server (runAs only)	any Subject (or WSSubject)
propagated?	Yes	WSSubject only

Admin security



Monitor

Can view configuration and state

Operator

"Monitor" functions + can change runtime state

inheritance relationship

Configurator

"Monitor" functions + can make configuration changes

Administrator

"Operator" functions + "Configurator" functions + access sensitive data

Role Based
LOCAL OS only