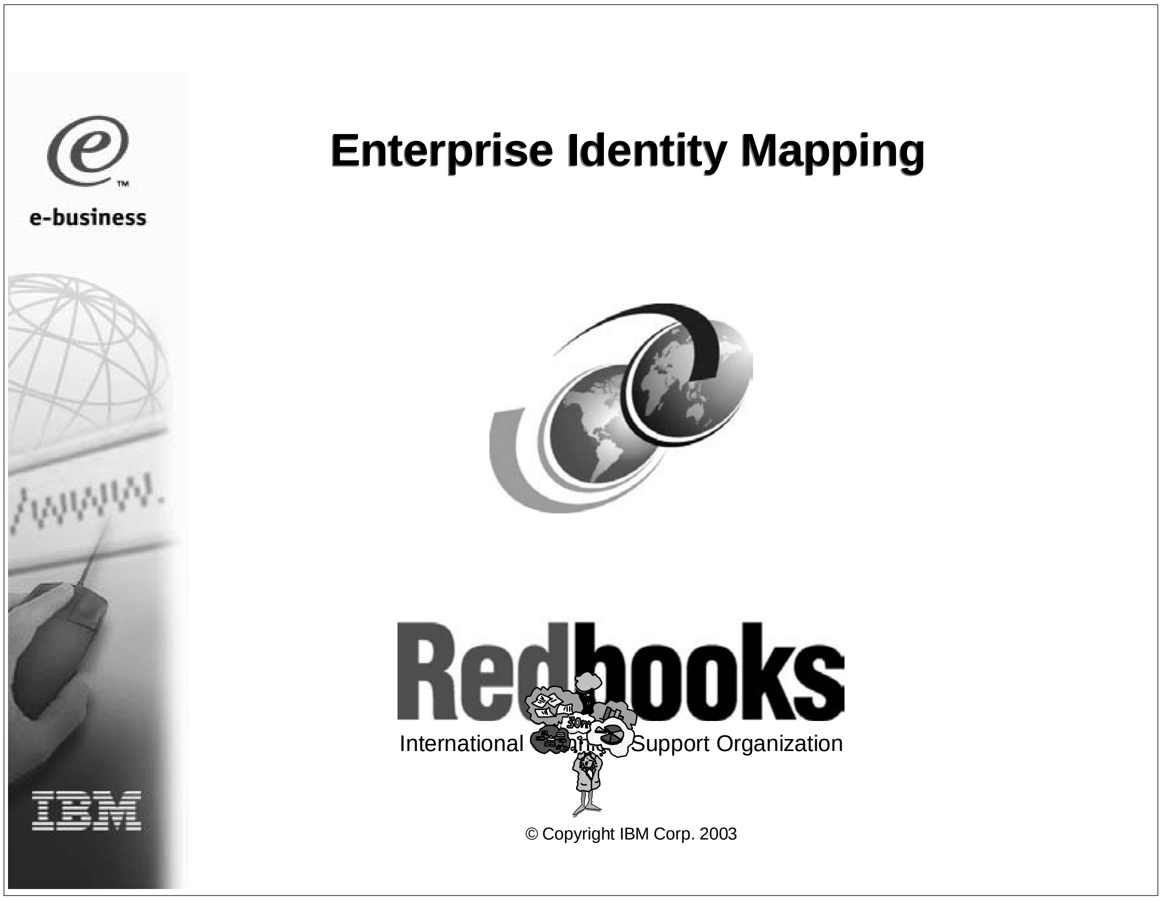
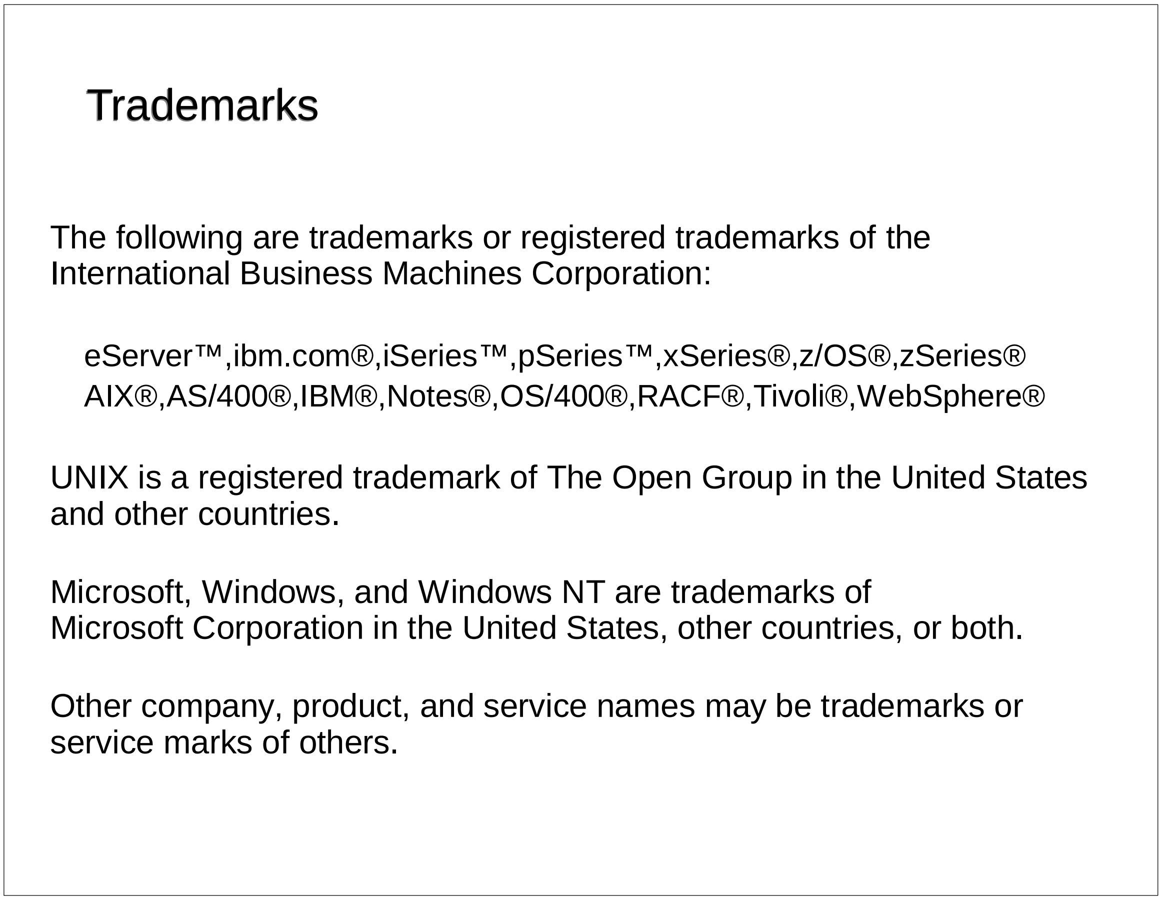


Enterprise Identity Mapping

International Technical Support Organization



Trademarks

The following are trademarks or registered trademarks of the International Business Machines Corporation:

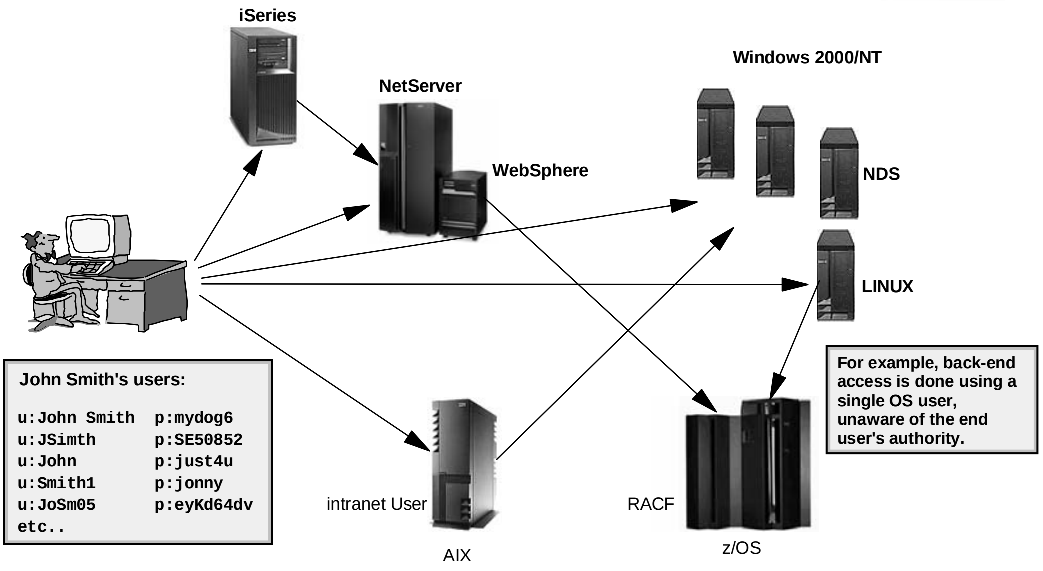
eServer™,ibm.com®,iSeries™,pSeries™,xSeries®,z/OS®,zSeries®
AIX®,AS/400®,IBM®,Notes®,OS/400®,RACF®,Tivoli®,WebSphere®

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

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

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

Typical Environment Today



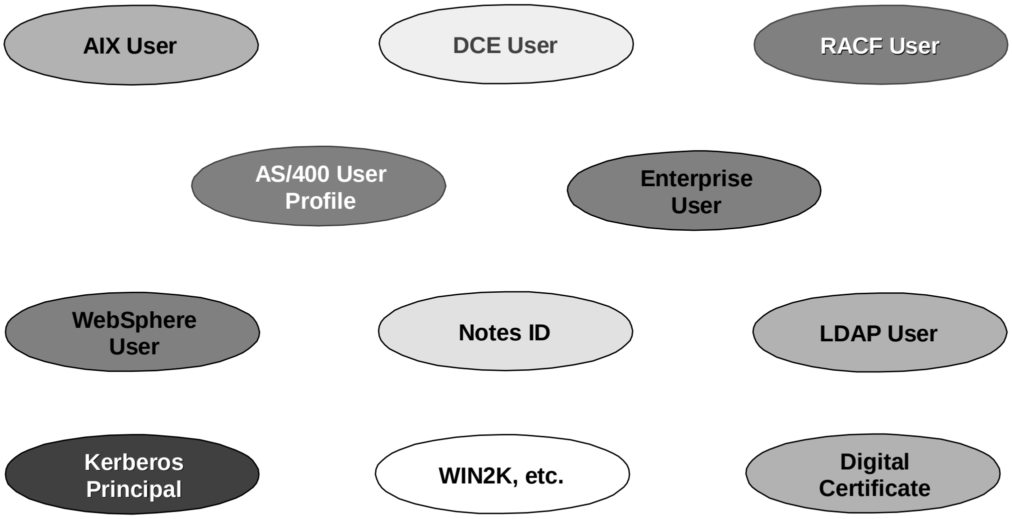
John Smith's users:

u:John Smith	p:mydog6
u:JSimth	p:SE50852
u:John	p:just4u
u:Smith1	p:jonny
u:JoSm05	p:eyKd64dv
etc..	

For example, back-end access is done using a single OS user, unaware of the end user's authority.

:

Multiple User Registries Problem



AIX User

DCE User

RACF User

AS/400 User Profile

Enterprise User

WebSphere User

Notes ID

LDAP User

Kerberos Principal

WIN2K, etc.

Digital Certificate

Administrative Nightmare !!

Single Signon ?

Enterprise "Trust Scope" ?

X-model transactions ?

Problem Description

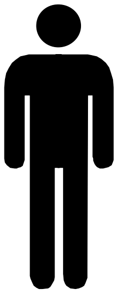
All eServer platforms (and many SWG products) have unique mechanisms for managing users (called User Registries). Most user registries are closely associated with an authorization mechanism.

- user registry = that set of users known to and/or trusted by a specific instance of an operating system or application
- authorization mechanism = those tools and interfaces that allow an administrator to assert ***who may access which resources in which way***.

In today's world of Data/Transaction Servers, UNIX and NT servers, Web Services, this becomes a severe problem for all classes of users.

Current approaches to solving, alleviating the problem focus on specific classes of users.

User Identity Problems for the End User

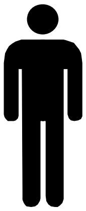


John Smith

z/OS	**JOHN**
OS/400	**JOHNSMITH**
AIX	**js**
LINUX	**js**
LDAP	**cn=John Smith,c=us**
Kerberos	**jsmith/admin@k390.ibm.com**

Many user IDs and passwords

User Identity Problems for the Administrator

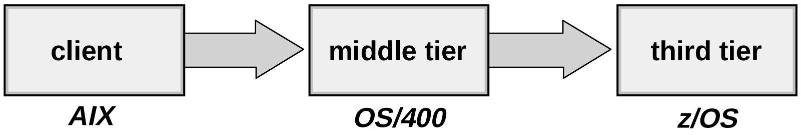


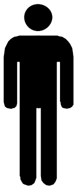
John Smith

z/OS	JOHN
OS/400	JOHNSMITH
AIX	js
LINUX	js
LDAP	cn=John Smith,c=us
Kerberos	jsmith/admin@k390.ibm.com

Keeping user IDs and
user info up to date!

User Identity Problem for the Multi-tiered Application Developer



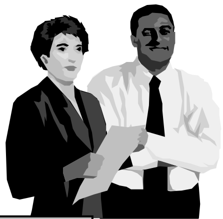


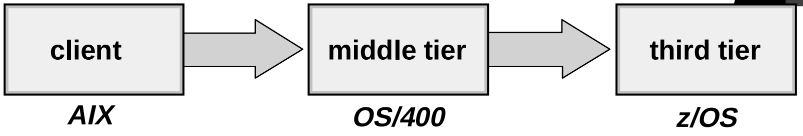
John Smith

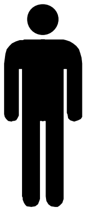
z/OS	JOHN
OS/400	JOHNSMITH
AIX	js
LINUX	js
LDAP	cn=John Smith,c=us
Kerberos	jsmith/admin@k390.ibm.com

Authentication, Authorization, Auditing

User Identity Problems for the Security Administrator and Auditor







z/OS	JOHN
OS/400	JOHNSMITH
AIX	js
LINUX	js
LDAP	cn=John Smith,c=us
Kerberos	jsmith/admin@k390.ibm.com

John Smith

Analysis of security policy

Existing Solutions Don't Go Far Enough...

- Standardized naming conventions
  - least common denominator
  - manual updates
- Cross platform administration tools
  - Ex. Tivoli
  - Address admin problems but not runtime or auditing
- Pick one registry and standardize on it
  - Which one? Convert all servers and apps to use it?
- Invent a new user registry
  - Avoids the cost system specific code
  - Adds yet another registry
  - Cannot leverage system specific access control

Enterprise Identity Mapping

Accept the fact that multiple registries (IBM and non-IBM) will exist in the enterprise

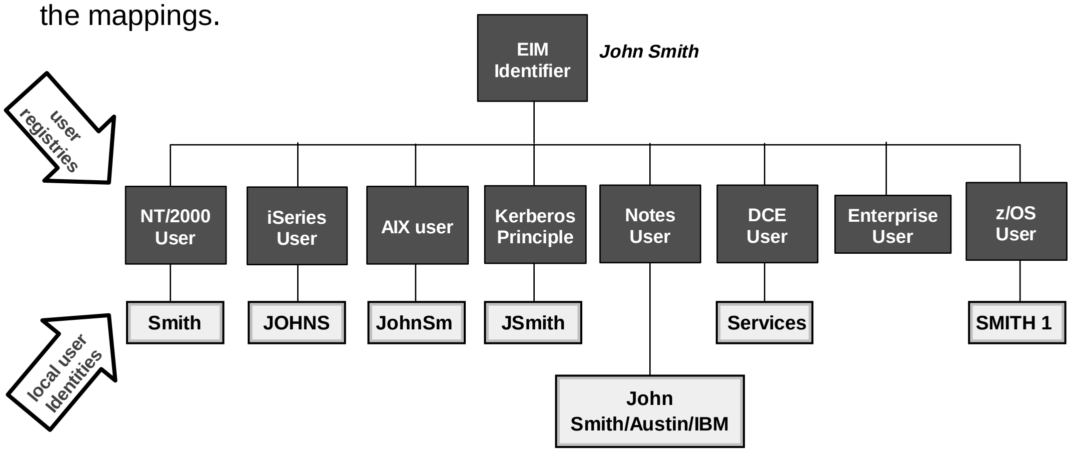
Make it easy for customers to associate a user's multiple identities in the enterprise and to manage those associations

Use IBM's platform breadth of software offerings to differentiate eServer platforms while providing a complete solution for heterogeneous environments

Develop this in such a way that it can be extended to other facets of cross-platform management

Enterprise Identity Mapping

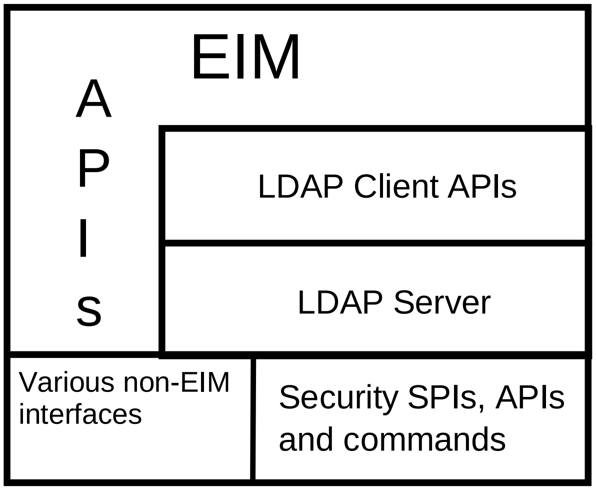
- **EIM defines** associations between an identifier and user ids in registries that are part of OS platforms, applications, and middle-ware.
- The identity associations (*mappings*) are stored in a well known location, e.g. LDAP, with common services across platforms to access the mappings.



- Addresses needs of applications and platforms to "translate" identity when crossing platform and registry boundaries.

EIM Architecture

EIM API Architecture



EIM

APIs

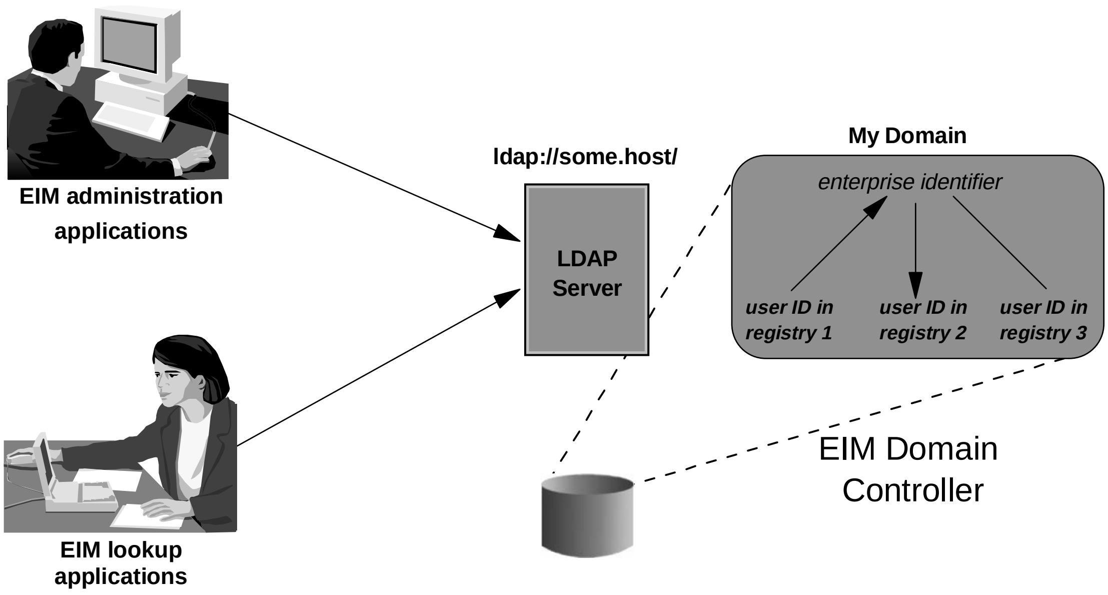
LDAP Client APIs

LDAP Server

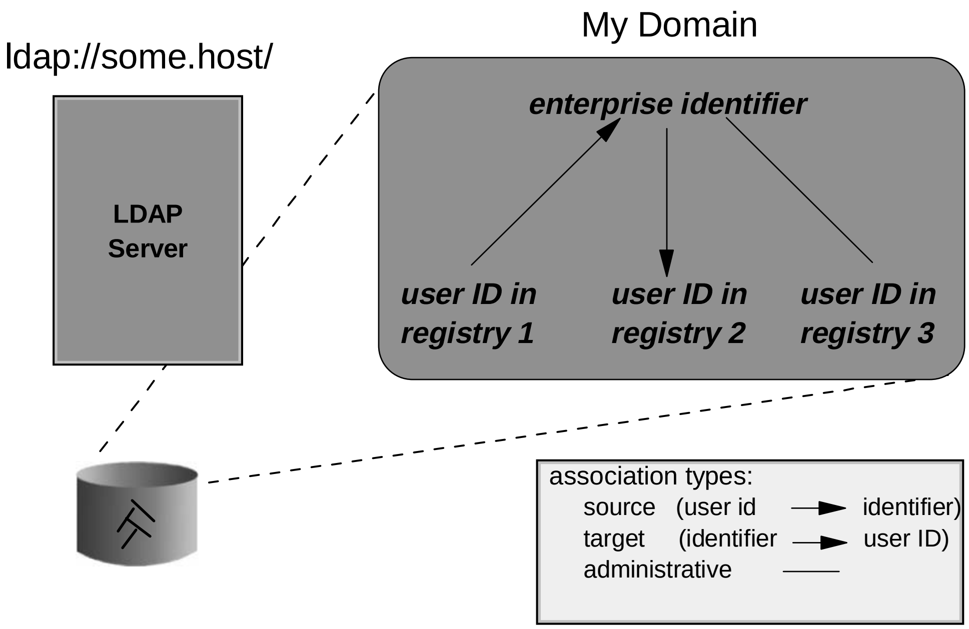
Various non-EIM interfaces

Security SPIs, APIs and commands

EIM eServer View



EIM Domain Controller



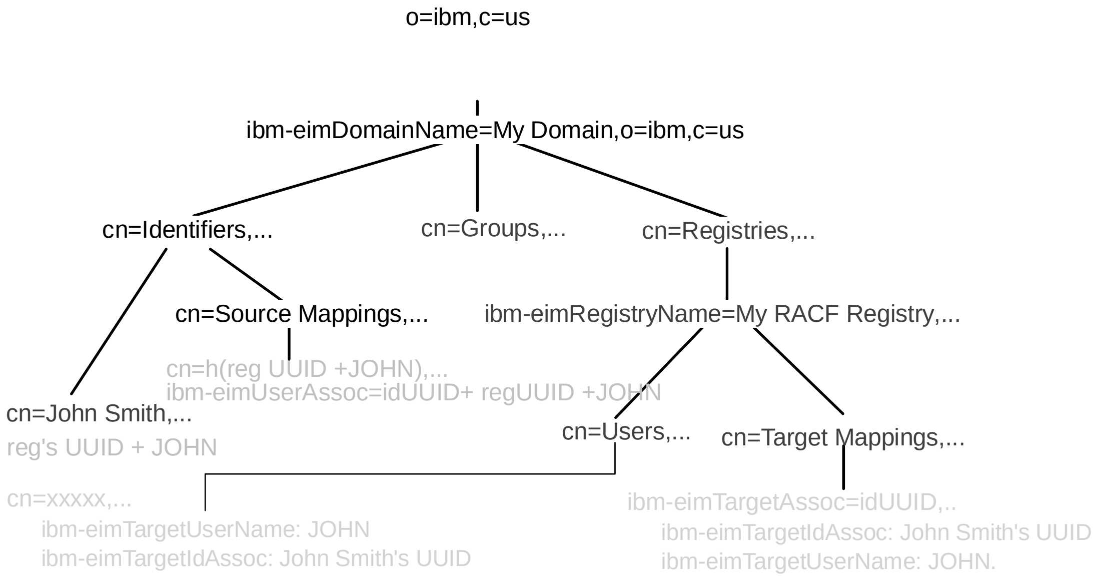
association types:
source (user id → identifier)
target (identifier → user ID)
administrative ——

EIM Schema Additions

- Attributes:
  - ibm-entryUUID
  - ibm-eimDomainName
  - ibm-eimAdditionalInformation
  - ibm-eimAdminUserAssoc
  - ibm-eimDomainVersion
  - ibm-eimRegistryAliases
  - ibm-eimRegistryEntryName
  - ibm-eimRegistryName
  - ibm-eimRegistryType
  - ibm-eimSourceUserAssoc
  - ibm-eimTargetIdAssoc
  - ibm-eimTargetUserName
  - ibm-eimUserAssoc

- Objectclasses:
  - ibm-eimDomain
  - ibm-eimIdentifier
  - ibm-eimRegistry
  - ibm-eimSystemRegistry
  - ibm-eimApplicationRegistry
  - ibm-eimRegistryUser
  - ibm-eimSourceRelationship
  - ibm-eimTargetRelationship

LDAP Directory Information Tree for an EIM Domain



o=ibm,c=us

ibm-eimDomainName=My Domain,o=ibm,c=us

cn=Identifiers,...

cn=Groups,...

cn=Registries,...

cn=Source Mappings,...

ibm-eimRegistryName=My RACF Registry,...

cn=h(reg UUID +JOHN),...
ibm-eimUserAssoc=idUUID+ regUUID +JOHN

cn=John Smith,...
reg's UUID + JOHN

cn=Users,...

cn=Target Mappings,...

cn=xxxxx,...
ibm-eimTargetUserName: JOHN
ibm-eimTargetIdAssoc: John Smith's UUID

ibm-eimTargetAssoc=idUUID,..
ibm-eimTargetIdAssoc: John Smith's UUID
ibm-eimTargetUserName: JOHN.

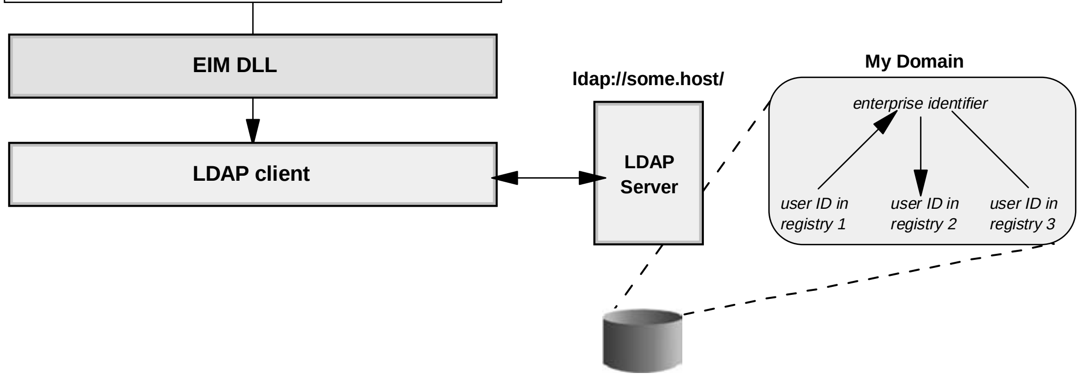
SourceRelationship objects are created for each source relationship that exists. Large number of objects....

For each registry, a tragetRelationship object is created for each target relationship.

EIM Client Applications - Administration or Lookup

C/C++ Lookup Application

```
/* obtain an identity, ex. principal @ realm */
call eimCreateHandle (...)
call eimConnect(...)

call eimGetTargetFromSource(...)
/* assert the new identity */
/* access local resources */

call eimDestroyHandle(...)
```



EIM DLL

LDAP client

ldap://some.host/

LDAP Server

My Domain

enterprise identifier

user ID in registry 1

user ID in registry 2

user ID in registry 3

EIM APIs

- EIM "handle" operations - common
  - Manages a token which is an instance of the EIM services. Similar in concept to other services in which the invoker is responsible for hanging-on to a "handle"
- Domain operations - EIM Admin
  - Creates a EIM domain, establishes the EIM "domain" controller...
- Registry operations - EIM Admin
  - System or application registries join EIM instance
- EIM Identifier operations - EIM Admin
  - Manages a "anchor" point for a enterprise user
- EIM Core Mapping operations - run-time
  - Supports determination of user's ID across disparate registries
- System operations - System/EIM Admin
  - Connection to an EIM domain
- User Management operations - Admin
  - Definition of this set of services is in progress
  - Direction is to define XML markup(s) which describe:
    - Users within registries and defines data passed on API
  - Allows add/modify/delete of users across multiple registries

Coded by application or registry security function that requires EIM services

EIM services implemented over LDAP, no new protocol

Access to an EIM Domain Controller

- Bind credentials for the LDAP server
  - bind distinguished name and password
  - digital certificate for client
  - kerberos principal and password
- All administrative users must have an EIM authority
  - EIM Administrator
  - EIM Registries administrator or registry administrator
  - EIM Identifier administrator
  - or is the ldap administrator
- All lookup users
  - EIM Mapping Operations

EIM Authorities

	LDAP Admin	EIM Admin	EIM Registries admin *	Identifier Admin	Mapping Operations
Domain	create	delete, change, list			
Registry		add, remove, change, list	change, list	list	list
Registry User		change, list	change, list	list	list
Registry Alias		change, list, retrieve	change, list, retrieve	list, retrieve	list, retrieve
Identifier		add, remove, change, list	list	add, change, list	list
Association		add, remove, list all types	add, remove target; list all types	add, remove admin, source; list all types	list
LookUp		all types	all types	all types	all types

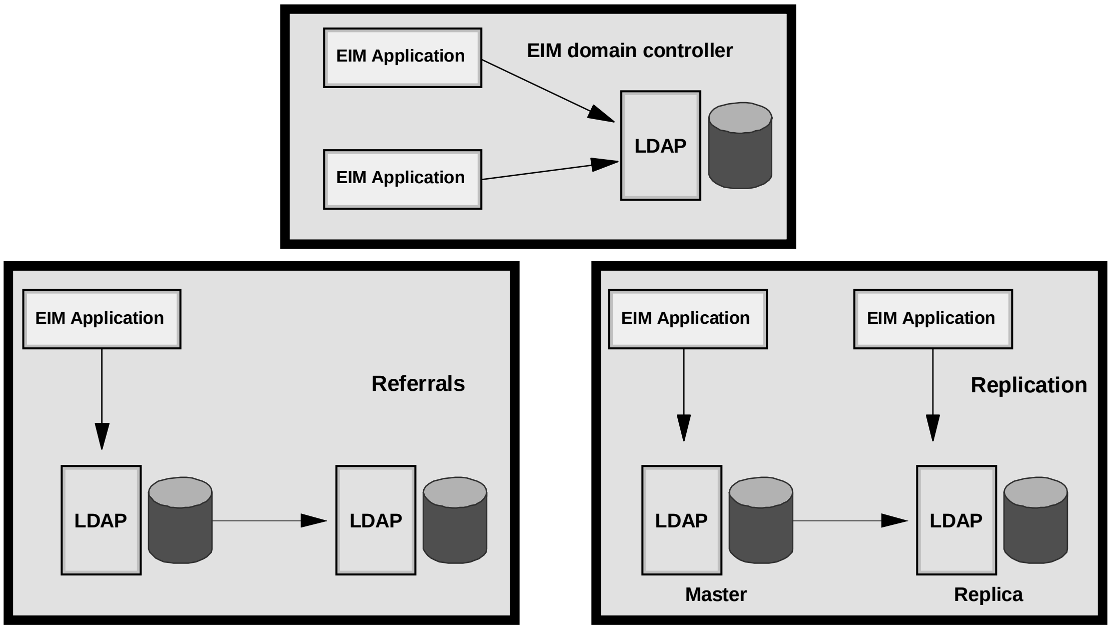
*** There is also registry_admin_ group for each registry**

EIM Availability

eServer EIM Support Today

Platform	EIM Domain Controller	EIM Client	IBM EIM Admin Tools
OS/400 on iSeries	OS/400 V5R2	OS/400 V5R2	OS/400 V5R2
z/OS on zSeries	z/OS V1R4 LDAP	z/OS V1R4 LDAP SPE OW57137	z/OS V1R4 LDAP SPE OW57137
AIX on pSeries		AIX R5.2	
Windows 2000 on xSeries		Download + IBM Directory 4.1 Client	
LINUX - SLES8 on PPC64 - Red Hat 7.3 on i386 -SLES7 on zSeries		Download + IBM Directory 4.1 client or OpenLDAP v2.0.23 client	

Basic EIM Configurations

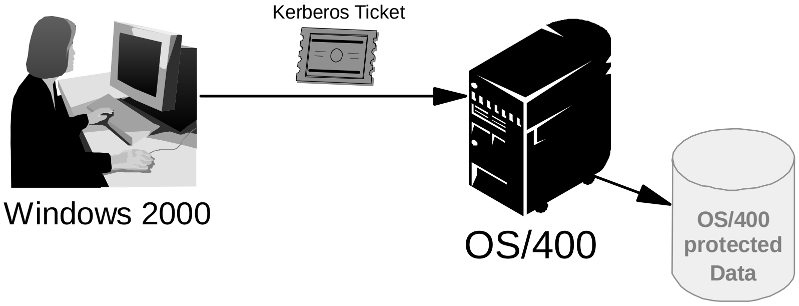


Benefits to Application Providers

Authentication vs. Authorization

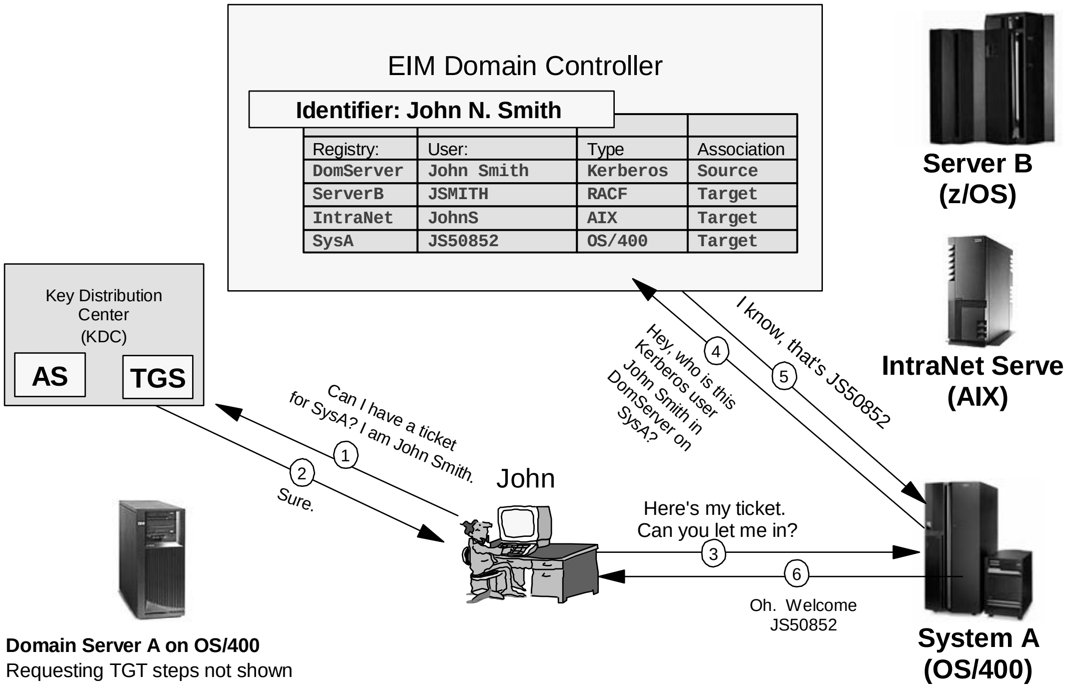
Client application says
"I am 'patriciaboats@MYCOM.WIN2KDOMAIN1' and here's proof. "

Kerberos addresses authentication only



OS/400 says
"I know who you are over **there**; but I need to know who you are over **here** to determine what you can access over here."

Kerberos + EIM = Single Sign-on

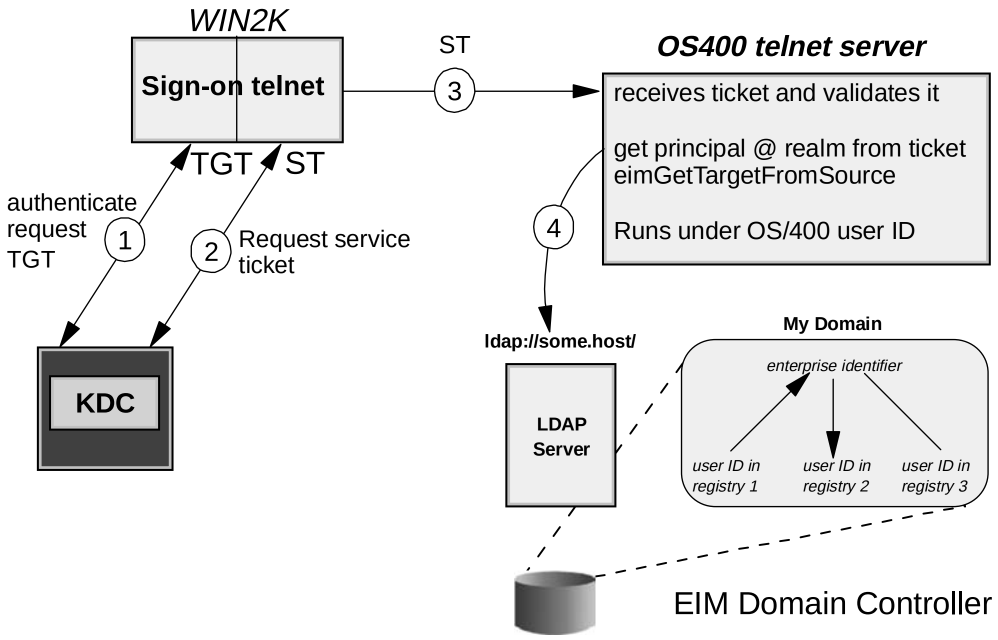


Identifier: John N. Smith

Registry:	User:	Type	Association
DomServer	John Smith	Kerberos	Source
ServerB	JSMITH	RACF	Target
IntraNet	JohnS	AIX	Target
SysA	JS50852	OS/400	Target

Domain Server A on OS/400
Requesting TGT steps not shown

OS/400's Single Sign-on w/Kerberos and EIM



WIN2K

Sign-on telnet

TGT ST

authenticate request TGT

1

2 Request service ticket

ST

3

OS400 telnet server

receives ticket and validates it

get principal @ realm from ticket
eimGetTargetFromSource

Runs under OS/400 user ID

4

KDC

ldap://some.host/

LDAP Server

My Domain

enterprise identifier

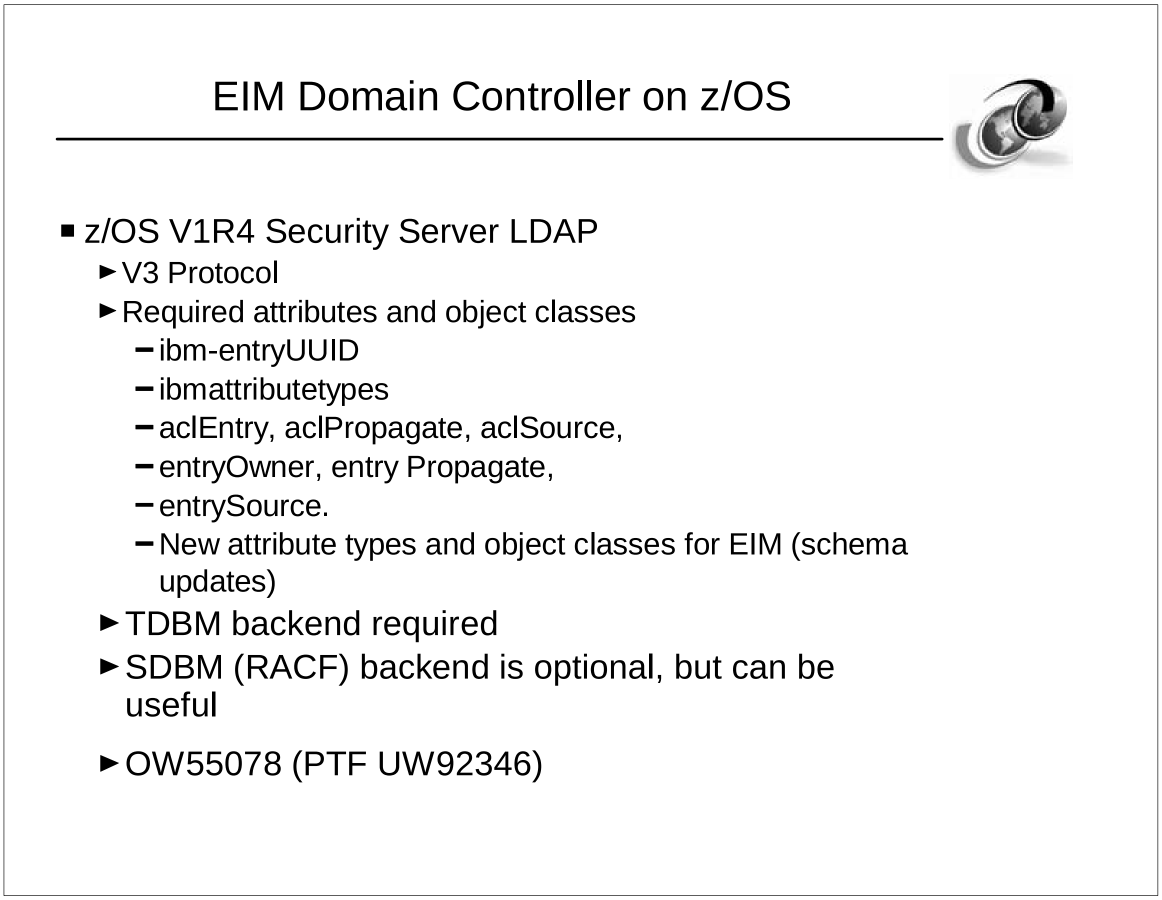
user ID in registry 1

user ID in registry 2

user ID in registry 3

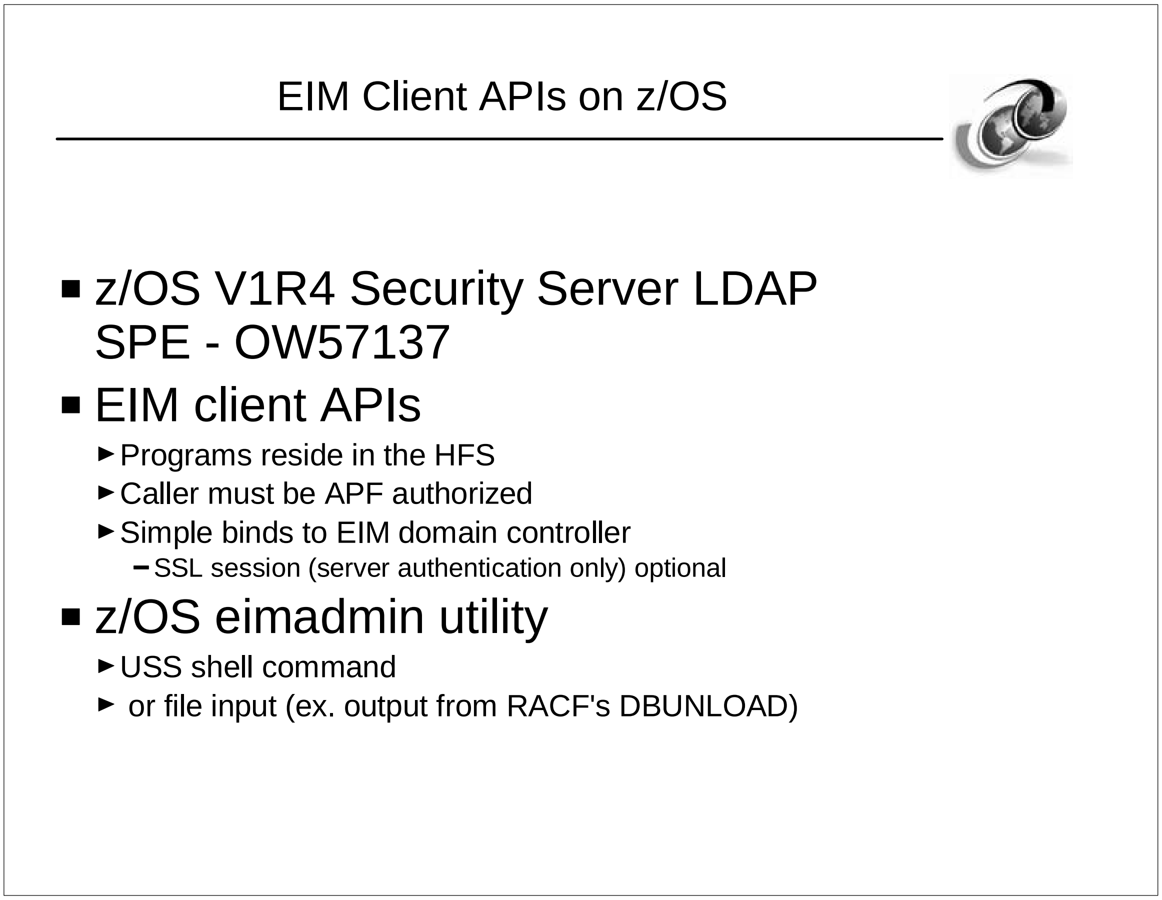
EIM Domain Controller

Enterprise Identity Mapping on z/OS



EIM Domain Controller on z/OS

- z/OS V1R4 Security Server LDAP
  - V3 Protocol
  - Required attributes and object classes
    - ibm-entryUUID
    - ibmattributetypes
    - aclEntry, aclPropagate, aclSource,
    - entryOwner, entry Propagate,
    - entrySource.
    - New attribute types and object classes for EIM (schema updates)
  - TDBM backend required
  - SDBM (RACF) backend is optional, but can be useful
  - OW55078 (PTF UW92346)



EIM Client APIs on z/OS

- z/OS V1R4 Security Server LDAP SPE - OW57137
- EIM client APIs
  - Programs reside in the HFS
  - Caller must be APF authorized
  - Simple binds to EIM domain controller
    - SSL session (server authentication only) optional
- z/OS eimadmin utility
  - USS shell command
  - or file input (ex. output from RACF's DBUNLOAD)

Getting Started with EIM on z/OS...

Preparation

The LDAP server hosting the domain is configured and started Entries for the suffix (i.e. o=ibm,c=us) are defined

Bind DN for the LDAP administrator is defined

Create the EIM administrator bind DN

1. In the file eimadministrator.ldif enter:

```
DN: cn=eim administrator,o=ibm,c=us
objectclass: top
objectclass: person
sn: eim administrator
cn: eim administrator

userpassword: secret
```

2. Issue the command

```
ldapadd -h ldap://some.big.host -D cn=ldap administrator -w secret
-f eimadministrator.ldif
```

Getting Started with EIM on z/OS...

1. Create the domain

```
eimadmin -aD  -d 'ibm-eimDomainName=My Domain,o=ibm,c=us'
              -h ldap://some.big.host:389
              -b 'cn=ldap administrator' -w passwd
```

2. Assign an EIM administrator

```
eimadmin -aC  -d 'ibm-eimDomainName=My Domain,o=ibm,c=us'
              -q 'cn=eim administrator,o=ibm,c=us'
              -c admin
              -h ldap://some.big.host:389
              -b 'cn=ldap administrator' -w passwd
```

Getting Started with EIM on z/OS...

3. Add system registries

```
     eimadmin -aR -r 'RACF on SYS1' -y RACF
                  -d 'ibm-eimDomainName=My Domain,o=ibm,c=us'
                  -h ldap://some.big.host
                  -b 'cn=eim administrator,o=ibm,c=us' -w passwd
```

4. Add identifiers

```
     eimadmin -aI -i'John Smith'
                  -d 'ibm-eimDomainName=My Domain,o=ibm,c=us'
                  -h ldap://some.big.host
                  -b 'cn=eim administrator,o=ibm,c=us' -w passwd
```

Getting Started with EIM on z/OS...

5. Add associations between identifiers and user IDs

```
     eimadmin -aA -r 'RACF on SYS1'
                  -u JOHN -i 'John Smith'  -t SOURCE -t TARGET
                  -d 'ibm-eimDomainName=My Domain,o=ibm,c=us'
                  -h ldap://some.big.host
                  -b 'cn=eim administrator,o=ibm,c=us' -w passwd
```

6. Give an end user EIM mapping lookup authority

```
     eimadmin -aC  -d 'ibm-eimDomainName=My Domain,o=ibm,c=us'
                   -q 'cn=John Smith,o=ibm,c=us'
                   -c mapping
                   -h ldap://some.big.host:389
                   -b 'cn=eim administrator' -w passwd
```

z/OS V1R4 Security Server RACF Support for EIM

- Security administrator has ability to
  - Define default EIM domain by system or by server
  - Define default LDAP bind information for the EIM domain
- Enhanced commands and profiles
  - ADDUSER, ALTUSER, LISTUSER
  - RDEFINE, RALTER, LISTUSER
  - IRR.PROXY.DEFAULTS FACILITY class profile
  - IRR.EIM.DEFAULTS LDAPBIND class profile
- Other updates
  - r_admin callable service
  - Database unload
  - SMF records, SMF unload
  - Templates

EIM Configuration on z/OS

- System-wide settings

```
   RDEFINE FACILITY IRR.PROXY.DEFAULTS
   EIM(DOMAINDN('ibm-eimDomain=My Domain,o=ibm,c=us'))
   OPTIONS(ENABLE))
   PROXY(LDAPHOST(ldap://some.big.host)
   BINDDN('cn=EIM Lookup') BINDPW('secret'))
-or-
   RDEFINE LDAPBIND IRR.EIM.DEFAULTS
   EIM(DOMAINDN('ibm-eimDomain=My Domain,o=ibm,c=us'))
   OPTIONS(ENABLE))
   PROXY(LDAPHOST(ldap://some.big.host)

   BINDDN('cn=EIM Lookup') BINDPW('secret'))
```

EIM Configuration on z/OS

- Server Specific Settings

```
RDEFINE LDAPBIND APPDOMAIN
EIM(DOMAINDN('ibm-eimDomain=Application Domain,o=ibm,c=us' ))
OPTIONS(ENABLE))
PROXY(LDAPHOST(ldap://another.big.host)
BINDDN('cn=EIM Application Lookups') BINDPW('secret'))

ADDUSER SERVERID EIM(LDAPPROF(APPDOMAIN))
```

- Can use same method to assign domain/bind info to an administrator's user ID

EIM Configuration on z/OS

- Assigning a name to SAF registry
  - ►Activate the name

```
RALTER FACILITY IRR.PROXY.DEFAULTS
            EIM(LOCALREGISTRY('RACF on SYS1'))
SETROPTS EIMREGISTRY or ipl the system
```

  - ►Deactivate the name

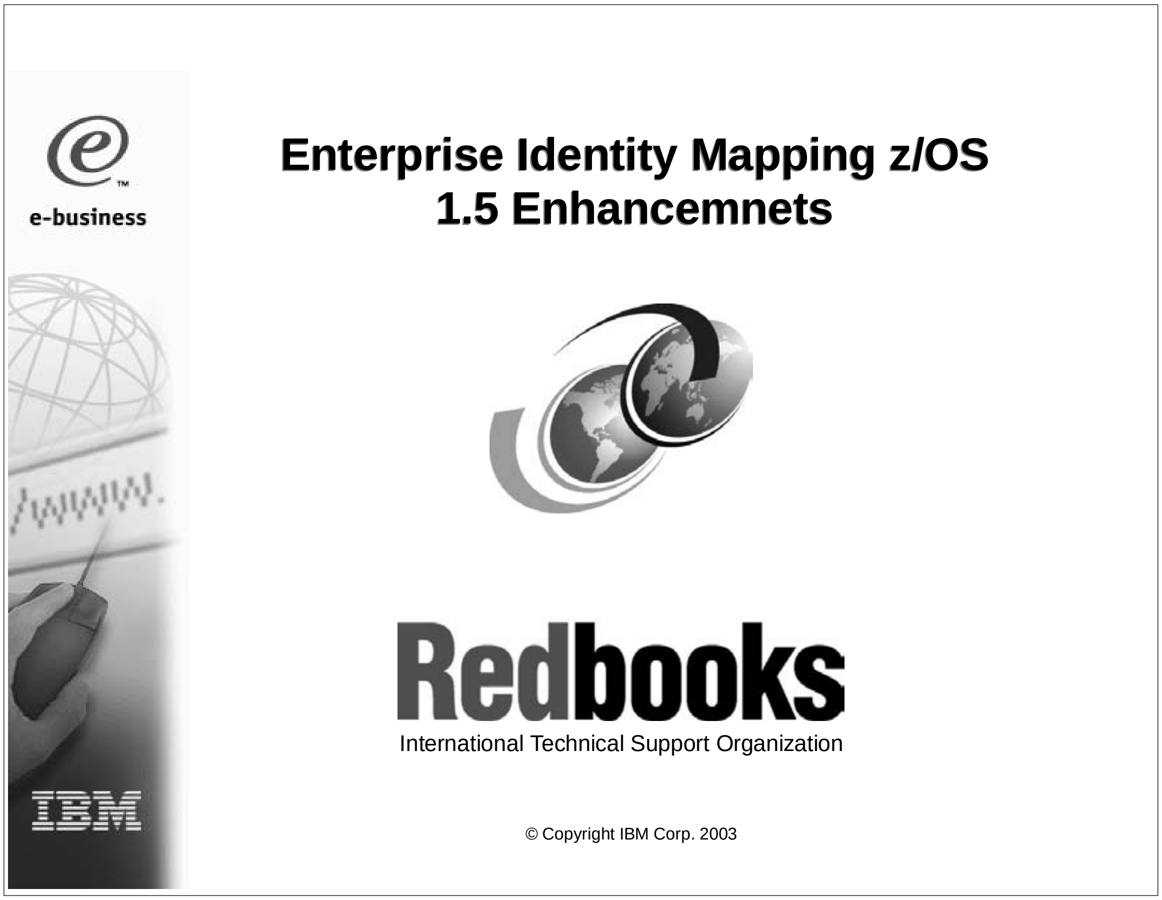
```
RALTER FACILITY IRR.PROXY.DEFAULTS
            EIM(NOLOCALREGISTRY)

SETROPTS EIMREGISTRY or IPL the system
```

How the EIM APIs Use the information in the RACF profiles

- eimGetTargetFromSource, eimGetTargetFromIdentifier, eimGetAssociatedIdentifier
  - ldaphost, domaindn, binddn, bindpw, enable/disable
  - Locates this info by searching in the following order:
    - Parameter list
    - Callers user profile -> LDAPBIND profile
    - IRR.EIM.DEFAULTS profile in the LDAPBIND class
    - IRR.PROXY.DEFAULTS profile in the LDAPBIND class
- Local registry name from IRR.PROXY.DEFAULTS
  - eimGetTargetFromSource
  - eimGetIdentifierFromSource
  - eimGetAssociatedIdentifiers

What is Enterprise Identity mapping?

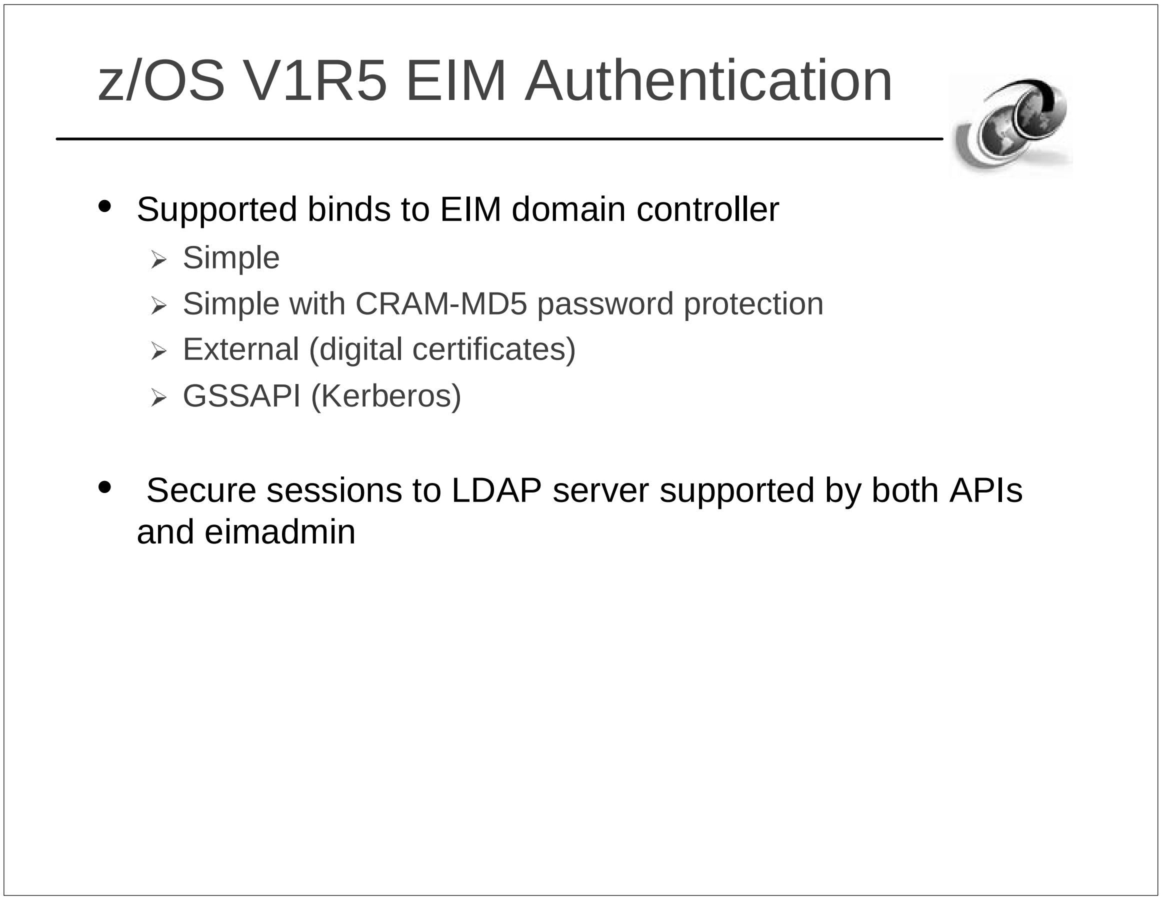
- EIM is a cornerstone to solving the complete set of problems with managing user IDs
  - End users
  - Administrators
  - Application developers
  - Security administrators and auditors
- LDAP application that centralizes mappings between user IDs and an enterprise wide identifier
- EIM domain controllers initially available on
  - z/OS and OS/400
- EIM client APIs available on all eServer platforms, LINUX and Windows 2000
- z/OS EIM available now



Enterprise Identity Mapping z/OS 1.5 Enhancemnets

Redbooks

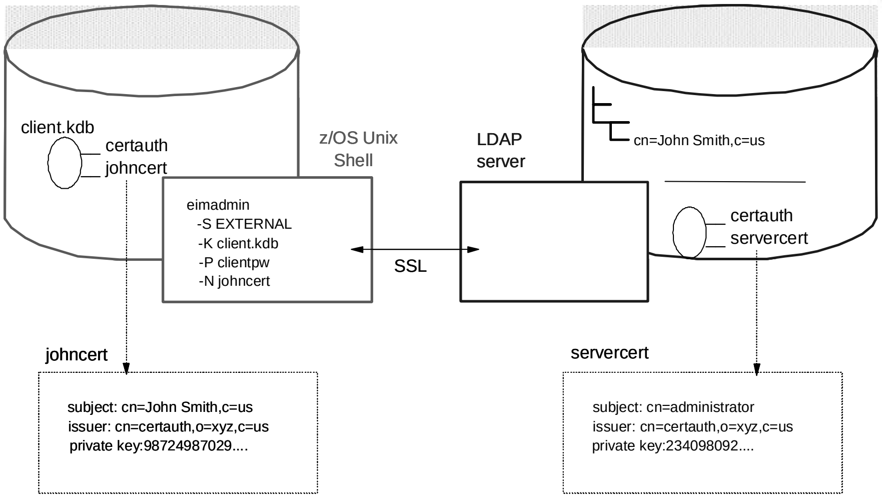
International Technical Support Organization



z/OS V1R5 EIM Authentication

- Supported binds to EIM domain controller
  - Simple
  - Simple with CRAM-MD5 password protection
  - External (digital certificates)
  - GSSAPI (Kerberos)
- Secure sessions to LDAP server supported by both APIs and eimadmin

Digital Certificate Authentication



client.kdb

certauth

johncert

z/OS Unix Shell

```
eimadmin
   -S EXTERNAL
   -K client.kdb
   -P clientpw
   -N johncert
```

SSL

LDAP server

cn=John Smith,c=us

certauth

servercert

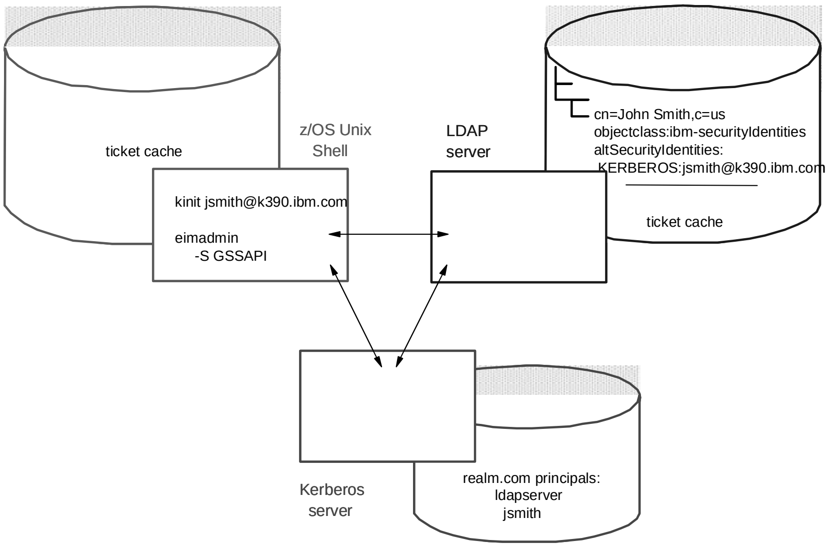
johncert

subject: cn=John Smith,c=us
issuer: cn=certauth,o=xyz,c=us
private key:98724987029....

servercert

subject: cn=administrator
issuer: cn=certauth,o=xyz,c=us
private key:234098092....

Kerberos Authentication



ticket cache

z/OS Unix Shell

```
kinit jsmith@k390.ibm.com

eimadmin
   -S GSSAPI
```

LDAP server

cn=John Smith,c=us
objectclass:ibm-securityIdentities
altSecurityIdentities:
KERBEROS:jsmith@k390.ibm.com

ticket cache

Kerberos server

realm.com principals:
ldapserver
jsmith

Additional Enhancements

- Online help for utility
  - man pages for eimadmin

- Translated messages
  - Japanese

Updated APIs

- eimChangeDomain
- eimConnect
- eimConnectToMaster
- eimCreateDomain
- eimDeleteDomain
- eimListDomains

Updated Utility

New eimadmin options

```
[-K keyFile
     [-P keyFilePassword ]
     [-N certificateLabel ] ]
[-S connectType ]
```