Configuring Access controls in the OS/390 and z/OS LDAP Server (Vanguard Session 72)

Tim Hahn
IBM z/OS Directory Development
hahnt@us.ibm.com

Disclaimer

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as-is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment and therefore, the results which may be obtained in other operating environments may vary significantly.

Users of this document should verify the applicable data for their specific environments. It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country or not yet announced by IBM. Such references or information should not be construed to mean that IBM intends to announce such IBM products, programming, or services.

Permission is hereby granted to Vanguard Security Expo to publish an exact copy of this paper in the Solutions proceedings. IBM retains the title to the copyright in this paper, as well as the copyright in all underlying works. IBM retains the right to make derivative works and to republish and distribute this paper to whomever it chooses in any way it chooses.

Trademarks

The following are trademarks of the IBM Corporation. An asterisk following the name denotes a registered trademark.

ACF/VTAM*
ADSTAR*
Advanced Function Printing
Advanced Peer-to-Peer Networking
AIX*
AIX/6000
APL2*
APPN
Approach
AS/400*
C/VM
C/370
Callup
CICS
CICS/VSE*
Common User Access
Current
CUA
DataJoiner
DataPropagator
DB2*
DB2 Connect
DB2/2
DB2/6000
DFS
DFSMS
DFSMS/VM
DirMaint
DisplayWrite*
Distributed Relational Database Architecture
Domino
DRDA*
Enterprise Systems Connection Architecture
Enterprise Systems Architecture/390
ES/9000*
ESCON*
GDDM*
Hardware Configuration Definition
IBM*
IBM Business Partner
IBMLink
IMS
Language Environment*
Lotus Notes
Lotus SmartSuite
MQ
MQ Series
Multiprise
MVS*
MVS/ESA
MVS/SP
MVS/XA
Net.Data
NetView*
Notes
NotesPump
OfficeVision*
OfficeVision/VM
Open Blueprint
OSA
OS/2*
OS/390
Parallel Sysplex
PowerPC
PR/SM
PROFS*
QMF
RACF
RAMAC
RISC System/6000*
RS/6000
SQL/DS
SQL Master
System/390*
S/370
S/390*
S/390 Multiprise
S/390 Parallel Enterprise Server
TalkLink
Time and Place
Ultrastar
VisualAge
VisualGen
VisualLift
Visual Warehouse
VM/ESA*
VM/XA
VSE/ESA
VTAM*
Wordpro

The names listed below are trademarks or registered trademarks and are the properties of their respective companies.

ANSI
Apple
Beyond Software
C++
CATIA
CSS
DEC
DirectPC
EnterpriseWeb/VM
EnterpriseWeb Calendar
Enterprise View
Ethernet
Eudora
Gateway
Hewlett-Packard
HP
IEEE
ITAA
Java
KERBEROS
LAN Manager
Macintosh
Mortice Kern Systems InterOpen
NCR
NCE
NetWare
Network File System
Novell
NFS
Open Software Foundation
OSF, Motif
Outlook
POSIX
SAS
SnapShot
Sterling Software
Sun Microsystems
SunOS
ULTRIX
UNIX
VAX
VM:Webserver
Windows
Windows NT
XPG4
X-Windows

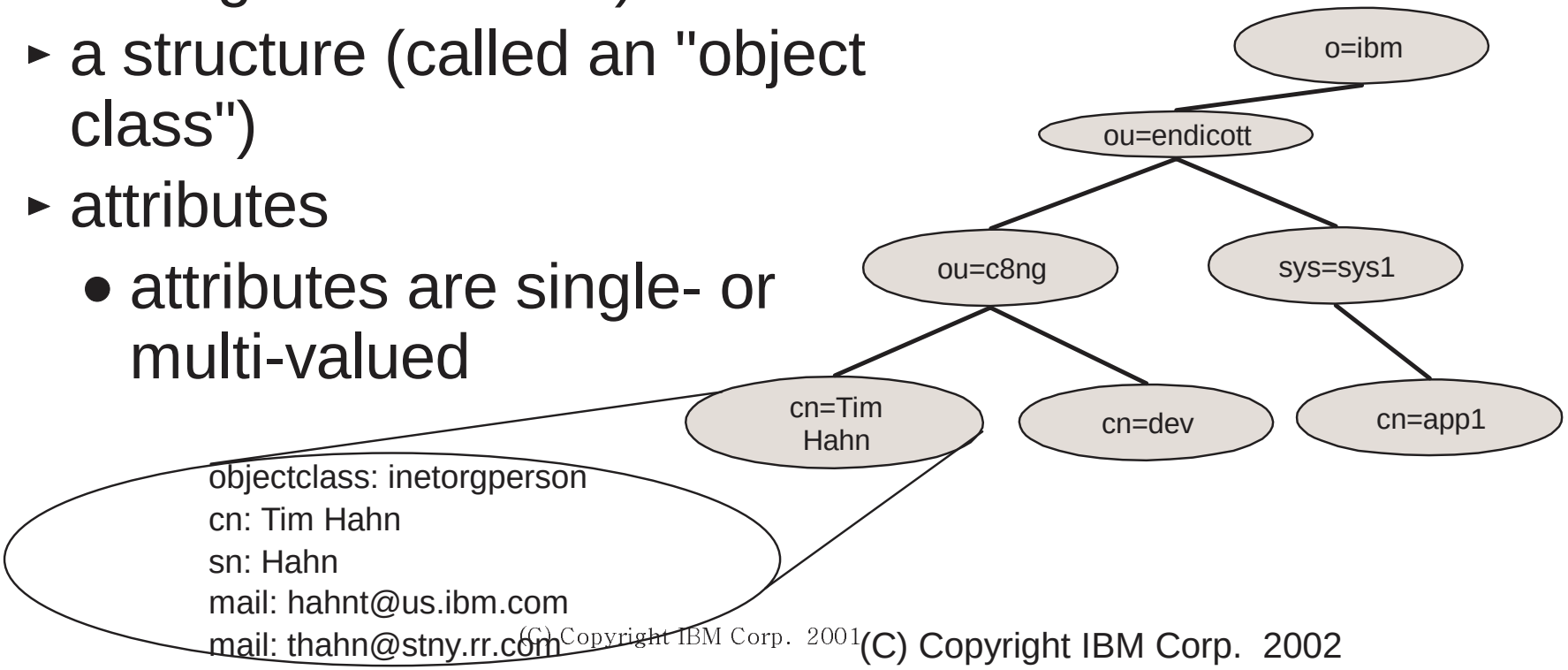
All statements regarding IBM's future intent are subject to change without notice, and represent goals and objectives only.

What are we going to talk about?

- Directory Information model
  - Hierarchy of entries
  - Object classes
  - Attributes
- OS/390 LDAP Access Control model
  - Data elements
  - Check Algorithm and precedence
  - Usage
  - Examples
  - Differences between Server Releases

Directory Information Model

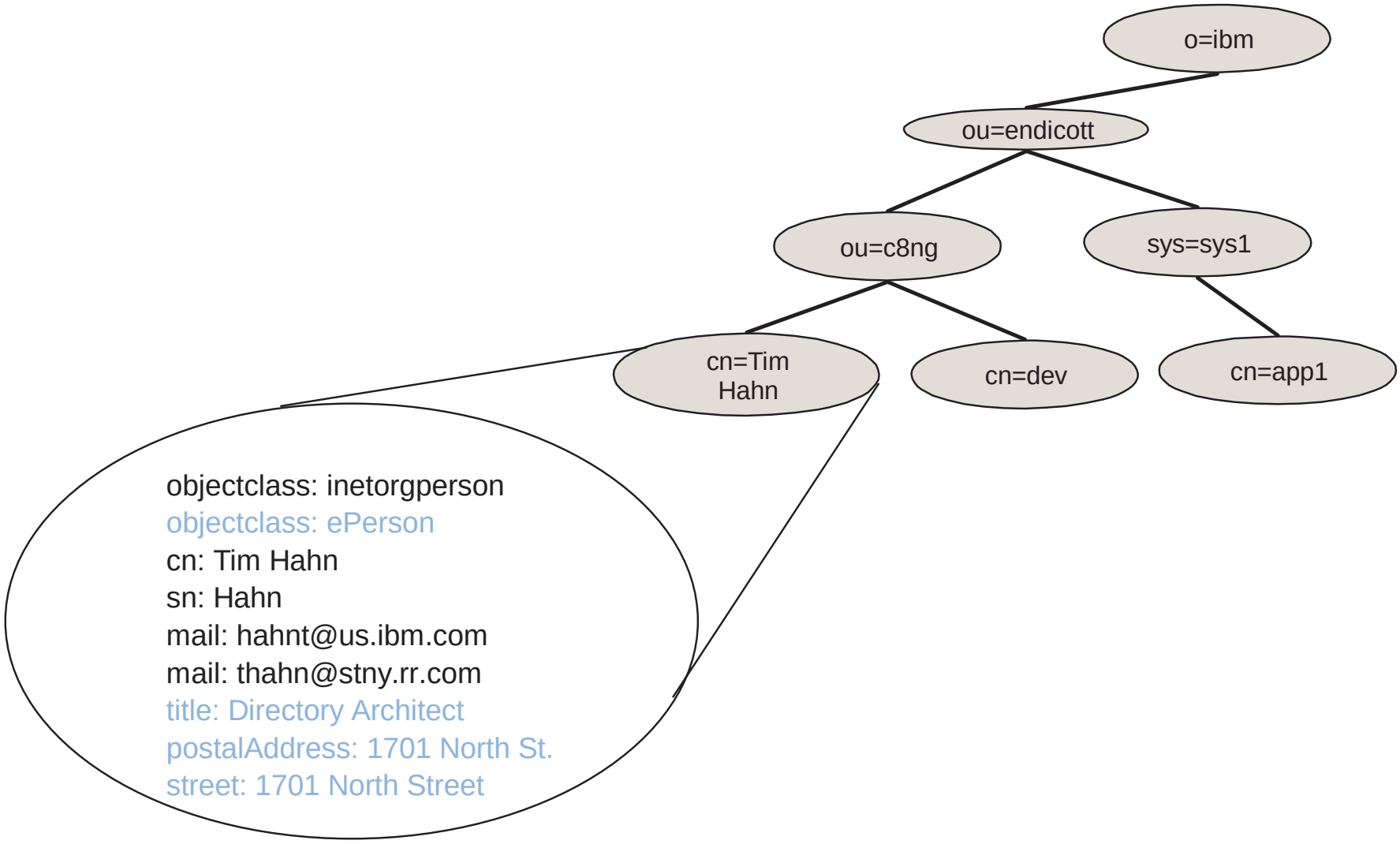
- An LDAP Directory is formed by a hierarchy of "entries"
- Each "entry" has:
  - a name (called a distinguished name)
  - a structure (called an "object class")
  - attributes
    - attributes are single- or multi-valued



Directory Information Model

- An Entry's "object class" defines
  - structure of an entry
  - attributes that MUST be present in an entry
  - attributes that MAY be present in an entry
- An individual Entry in the directory can take on the form of multiple object classes
  - The attributes in the entry are the UNION of those defined for individual object classes

Directory Information Model

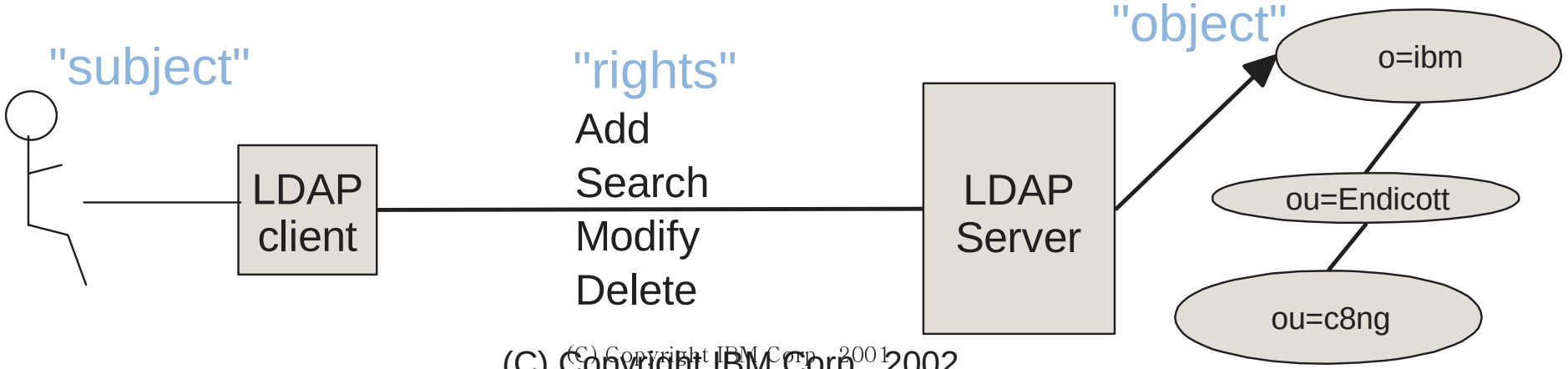


Directory Information Model

- Attributes are defined by their name, syntax, and matching rule(s)
  - Syntax refers to the type of data stored in attribute values
    - Examples: directoryString, binary, integer
  - Matching Rules define how equality and ordering comparisons are performed on attribute values
    - Examples: caseIgnoreMatch, caseExactMatch, octetStringMatch
- Different attributes within an entry may be more "sensitive" than others within an entry
  - Example: common name (cn) vs. uid vs. userPassword

Access Control of Directory Content

- The access control model for a directory must
  - treat the "subject" of the access control check as the identity/group set of the "bound" LDAP client
  - treat the "object" of the access control check as the entry (or attributes within an entry) in the directory tree
  - determine the set of "rights" required in order to perform any given LDAP operation
  - Check to see if the "subject" can access the "object", with the required "rights"



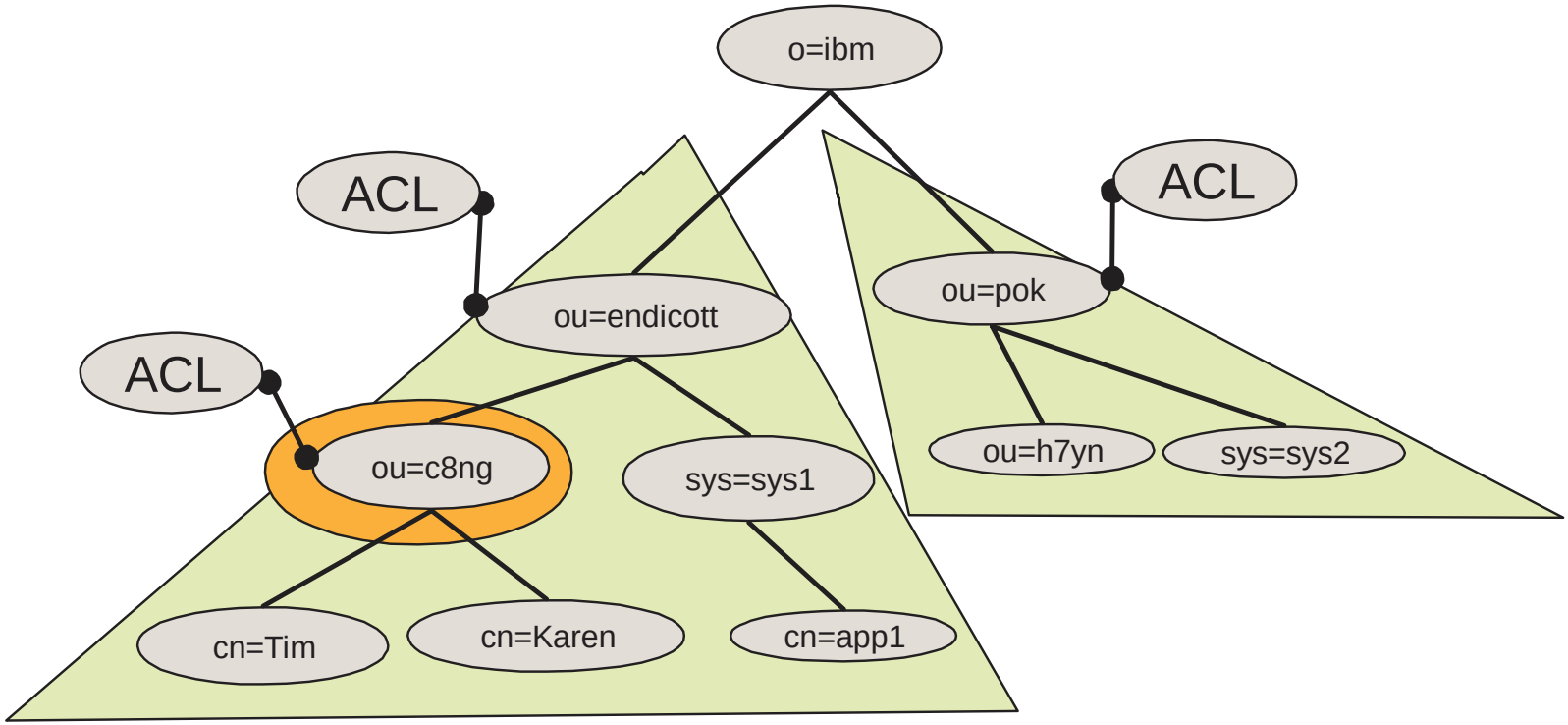
Access Control Implementation

- Provides "near" attribute-level granularity of permissions
  - attributes are "grouped" by "access class"
  - normal, sensitive, critical
  - the access class for an attribute is defined with extensions to the schema for the attribute
- "subject" is defined using distinguished names (in accordance to LDAP BIND protocol operations)
- "object" is defined in terms of entry name and attribute "access class" (normal, sensitive, critical, restricted, system)
- "rights" defined as read (r), write (w), search (s), compare (c)

Access Control Implementation

- Access control can be defined in one entry and made to apply to all entries below the entry (until overridden by another ACL definition)
- Each entry also has an "entry owner" which has complete access to the entry
- Entry owner can also be made to apply "down the tree"
- All access control information is viewable and updateable as additional attributes in an entry

Access Control Implementation



Access Control Attributes

- Access control is defined with directory attributes attached to entries in the directory
  - aclEntry
  - aclPropagate
  - aclSource
  - entryOwner
  - ownerPropagate
  - ownerSource
- aclSource and ownerSource are read-only attributes
  - calculated by the backend
  - never specified when defining ACLs

ACL Attributes

- aclEntry
  - Defines specific permissions for specific people or groups
  - Used if entryOwner does not apply
  - Format:
    - [access-id:|group:|role:]distinguishedName \ [<:normal:|:sensitive:|:critical:|:object:>permissions]*
    - permissions are [r|w|s|c]* or [a|d]* for :object:
  - Example:
    - aclEntry: cn=Tim, ou=c8ng, ou=Endicott, o=IBM:normal:rwsc:sensitive:rsc
    - aclEntry: cn=Karen, ou=c8ng, ou=Endicott, o=IBM:object:ad:normal:rwsc

ACL Attributes

- aclPropagate
  - single-valued attribute and defines if the aclEntry applies to only the entry or to the entry and the sub-tree of entries below the entry
  - format
    - TRUE|FALSE
    - Example:
      - aclPropagate: TRUE

Owner Attributes

- entryOwner
  - multi-valued attribute and defines the names of people or groups that are deemed "owners" of an entry (or sub-tree).
  - An owner has complete control of the entry, including changing the ACL
  - format:
    - [access-id:|group:|role:]distinguishedName
    - Example:
      - entryOwner: cn=Tim, ou=c8ng, ou=Endicott, o=IBM

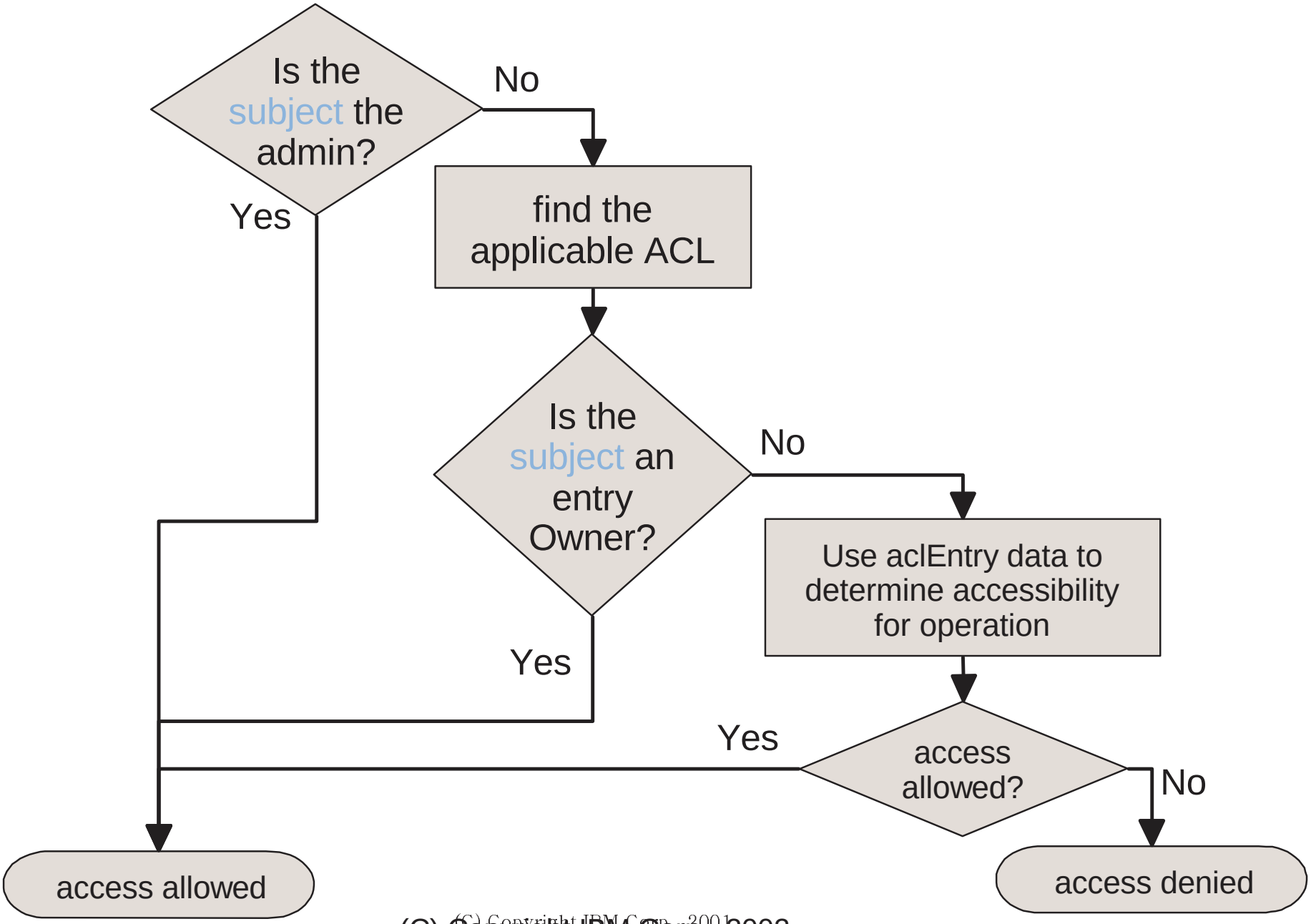
Owner Attributes

- ownerPropagate
  - single-valued attribute and defines if the entryOwner applies to only the entry or to the entry and the sub-tree of entries below the entry
  - format
    - TRUE|FALSE
    - Example:
      - ownerPropagate: TRUE

Special aclEntry "pseudo-DNs"

- cn=anybody
  - this is used when no specific ACL entry applies
- cn=authenticated (OS/390 R10 and later, SecureWay V3.1+)
  - this is used if the person has authenticated to the directory but no specific ACL entry applies
  - meant to allow more access than cn=anybody ACL entry
- cn=this (OS/390 R10 and later, SecureWay V3.1+)
  - this is used if the person has authenticated with the same name as the entry being accessed
  - used to grant individuals access to private attributes using a single ACL that applies to a sub-tree

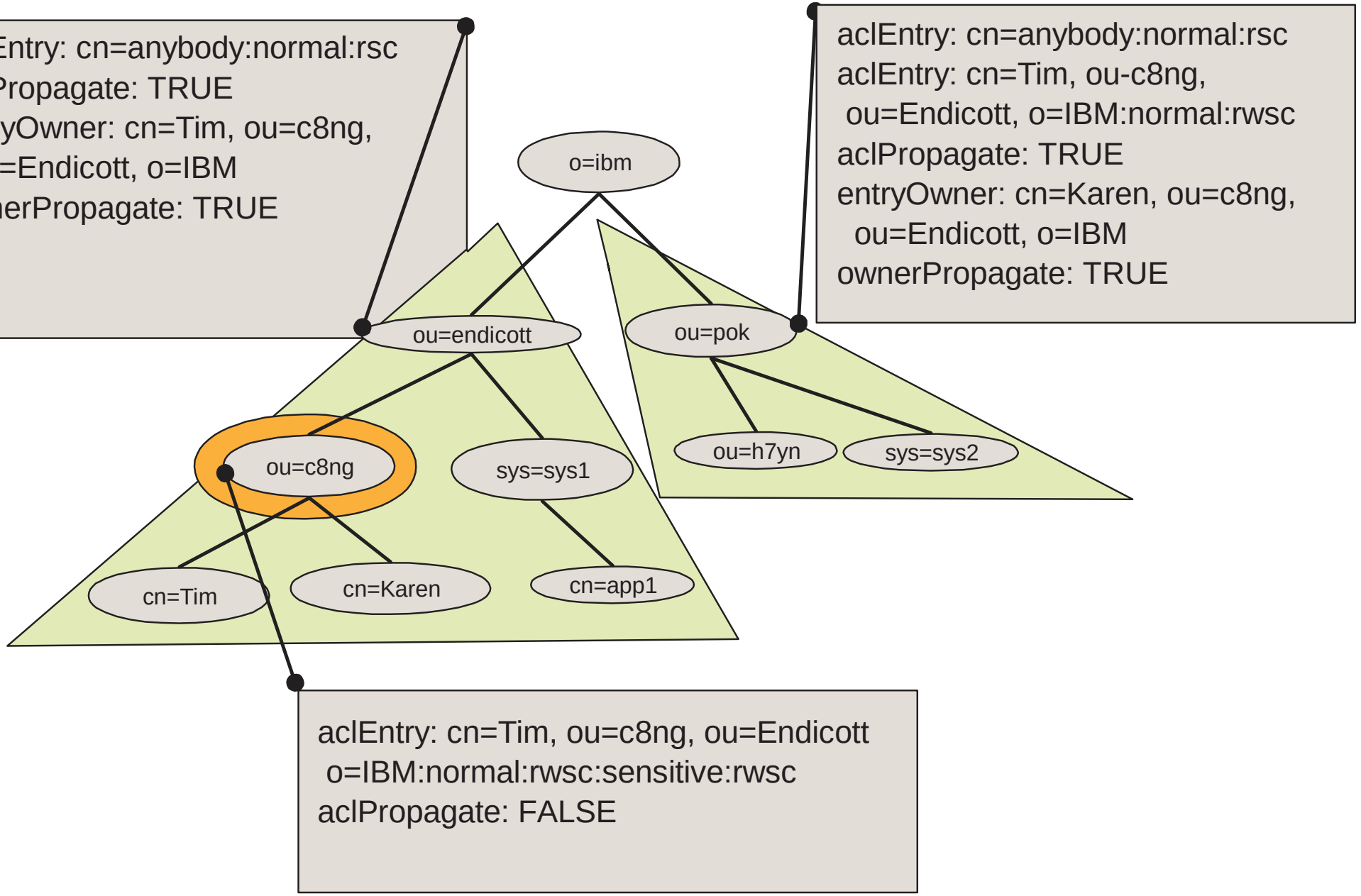
ACL Checking Algorithm



ACL Entry precedence

- more specific aclEntry values have higher precedence
- group and role type aclEntry values have their permissions unioned together
- precedence:
  - value specific to a bound user (including cn=this)
  - set of applicable group and role ACL entries
  - cn=authenticated
  - cn=anybody

Access Control Example



Creating and Modifying ACLs

- Use the ldapmodify command to modify ACLs
  - Accepts input in the form of LDIF files
  - edit a file to contain the modification
  - run the ldapmodify command with -f <filename>
- Individual ACL entries can be added and removed

ACL modify Example

- Command:

```
$ ldapmodify -h localhost \
 -D "cn=Tim, ou=c8ng, ou=Endicott, o=IBM" \
 -w XXXXX \
 -f aclmods.ldif
```

- aclmods.ldif file:

```
dn: ou=Endicott, o=IBM
changetype: modify
add: aclEntry
aclEntry: cn=Karen, ou=c8ng, ou=Endicott,
 o=IBM:normal:rwsc
aclEntry: cn=this:normal:rwsc:sensitive:rwsc:
 critical:rwsc
-
replace: aclPropagate
aclPropagate: TRUE
-
```

Listing ACLs

- Use the ldapsearch command to list ACL information
  - the ACL attributes must be explicitly requested in order to have the ACL information returned
  - ACL attributes are covered under access control checks as well so only users who are allowed to see ACL information can see it

ACL search example

- Command:

```
$ ldapsearch -h localhost \
 -D "cn=Tim, Hahn, ou=Endicott, o=IBM" \
 -w XXXXX \
 -b "ou=Endicott, o=IBM" -s base -L \
 "(objectclass=*)" \
 aclEntry aclPropagate aclSource \
 entryOwner ownerPropagate ownerSource
```

ACL search example *(continued)*

- Command Output:

```
dn: ou=Endicott, o=IBM
aclEntry: cn=Karen, ou=c8ng, ou=Endicott,
 o=IBM:normal:rwsc
aclEntry: cn=this:normal:rwsc:sensitive:rwsc:
 critical:rwsc
aclEntry: cn=anybody:normal:rsc
aclPropagate: TRUE
aclSource: ou=Endicott, o=IBM
entryOwner: cn=Tim, ou=c8ng, ou=Endicott, o=IBM
ownerPropagate: TRUE
ownerSource: ou=Endicott, o=IBM
```

Differences in ACL model by OS/390 Release

- The following ACL features apply only to TDBM in OS/390 R10 and later
  - cn=this
  - cn=authenticated
  - access-id, group, role keywords optional
  - add/modify of individual aclEntry values
  - ACL administration using ldapmodify
- ACL support in RDBM is the same for all OS/390 releases, including OS/390 R10

For More Information

- LDAP RFCs
  - http://sunsite.auc.dk/RFC/rfc/rfc2251.html- rfc2256.html
- z/OS LDAP Documentation
  - SC24-5923-02 z/OS V1R2.0 Security Server LDAP Server Administration and Use
    - http://publibz.boulder.ibm.com/epubs/pdf/glda2a11.pdf
  - SC24-5924-01 z/OS V1R2.0 SecureWay Security Server LDAP Client Programming
    - http://publibz.boulder.ibm.com/epubs/pdf/glda1a10.pdf
- Books
  - e-Directories: Enterprise Software, Solutions, and Services
    House, Hahn, Mauget, Daugherty
    ISBN: 0-201-70039-5
    - http://www.awl.com/cseng/titles/0-201-70039-5
  - Understanding LDAP
    - http://www.redbooks.ibm.com
- Contacting me
  - e-mail: hahnt@us.ibm.com