

CA ACF2™, IBM® RACF®, CA Top Secret®

What's the same – What's different?

Carla Flores – carla.flores@ca.com

Julie Bergh – jbergh@us.ibm.com

October 14, 2015

The “Big 3.” They all fundamentally do the same thing – protect z/OS and your data.

If you struggle to understand the similarities (or differences) we are here to help!

We will cover the basic structure of each security product and discuss how they secure in different ways and the similar ways.

Abstract

The “Big 3” z/OS Security Packages



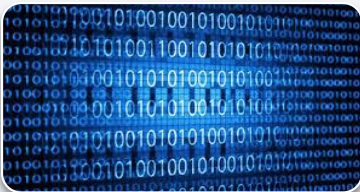
All use SAF

- System Authorization Facility
- Invoked for security access checks, passes the request along to the appropriate security system



All have Security Databases (“Directories”)

- Not X.500 directories but highly-efficient legacy systems
- Now accessible from X.500 via LDAP



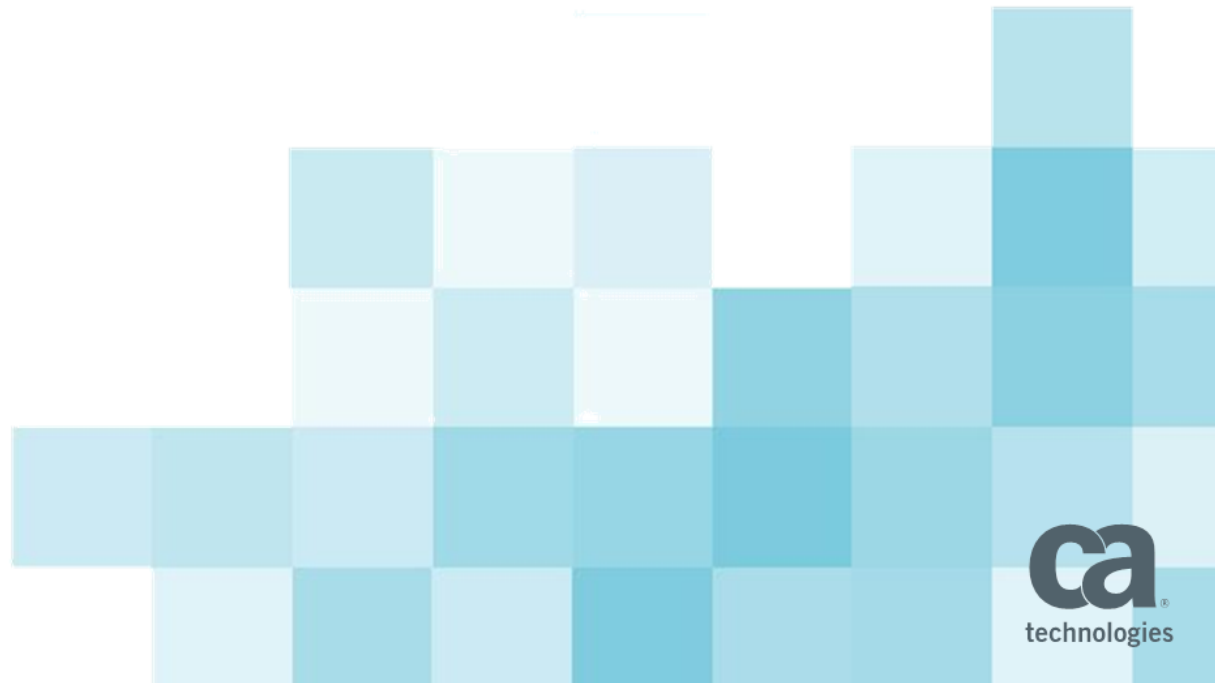
All protect the data as defined



All write to SMF for audit/reporting

CA ACF2™ for z/OS Basics

Carla A. Flores



CA ACF2

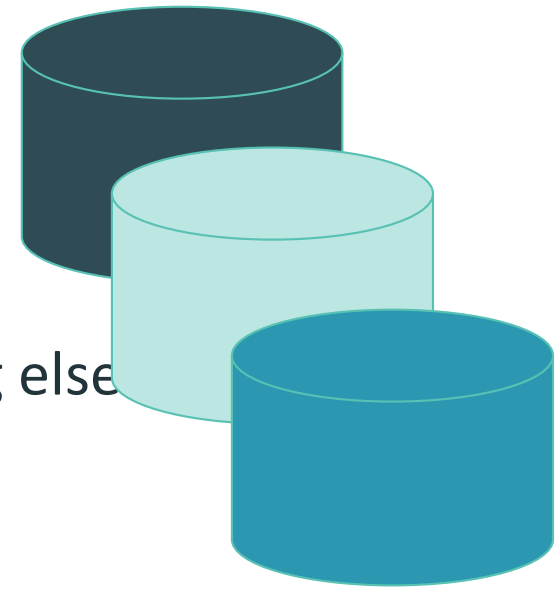
- **“Access Control Facility 2” aka “ACF2”**
 - Developed by SKK (Schrager, Klemens and Krueger) in 1978 and marketed by Cambridge
 - Cambridge was acquired by UCCEL, who was acquired by CA in 1987
- **Resource Oriented**
 - Resources are defined and permitted through rules
- **IDs are called “LIDs” (for Logon IDs)**
- **Logical grouping of IDs is by nature of the UID string**
 - UID string is user defined fields consisting of up to 24 bytes
- **Protection by default – if it isn’t defined, no access is given!**

CA ACF2 Security Modes

QUIET	LOG	WARN	RULE	ABORT
• System entry validation	• System entry validation • Access rule validation and logging • Access to data NOT prevented	• Same as LOG mode • Warn message issued to user	• System entry validation • Access rule validation • Selectable mode for each rule set based on \$MODE	• System entry validation • Access rule validation • Unauthorized access prevented • Violation message issued

CA ACF2 Structure – Three VSAM files

- **Logonid Database**
 - One record per logonid
 - Central source for most user data*
- **Rules Database**
 - Contains all data set access rules
- **Infostorage Database** – includes everything else
 - GSO (global system options)
 - Resource rules (all non-data-set access rules)
 - XREF (cross-reference records)
 - SCOPE (limit the authority a specially privileged user has)
 - SHIFT/ZONE (defines when access is permitted or prevented)
 - PROFILES (security information extracted by SAF)
- Can be synched using the Command Propagation Facility (CPF)



Logonid Database – User identification (UID string)

- 1-24 character long “pseudo field” constructed of logonid record fields such as department, location, job function and logonid
- Allows for grouping of users
- Often contains user-defined fields
- Multi-valued Logonid fields-allow multiple views of a single UID
- Example: @UID LOC, DIV, DEPT, JOBF, LID

CHFOPSCHTLC492

LOC = Chicago

DIV = Finance & Data Processing

DEPT = Operations

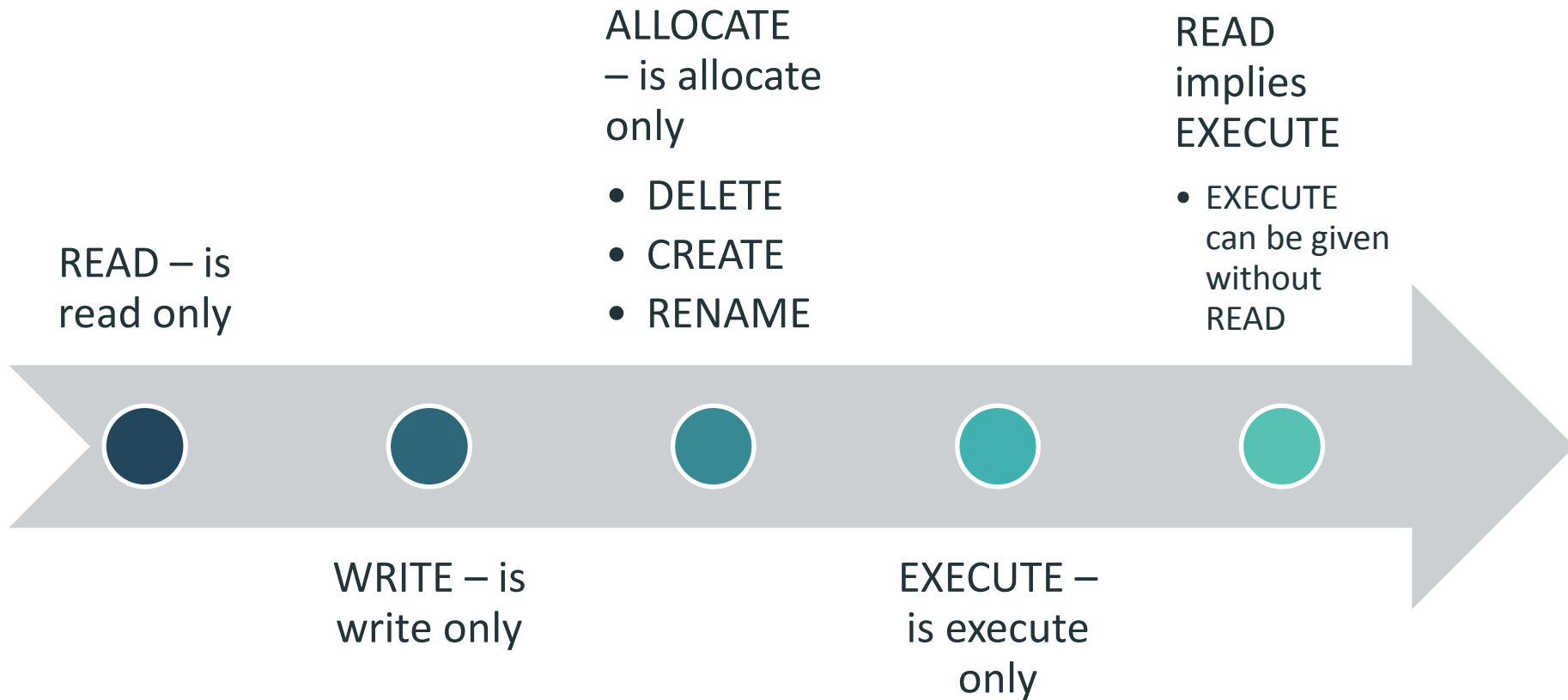
JOBF = Scheduler

LID = TLC492

Rules Database

- By default, CA ACF2 does not allow access to data unless rules authorize it
 - PROTECTION BY DEFAULT (in full Abort mode)
 - Direct result of the SHARE requirements
- One rule set exists for each DSN high-level index or resource
- Rule sets can exist for entire volumes of data in DASD or tape
- Key:
 - Up to 8 character to record on database is DSN high-level index for data set rule sets
 - Up to 40 character for resource rules
 - Nextkeys allow for chaining of rules that span (4K)
- Rule sets are compiled and stored much like programs

Access Rule Permissions



***Note: Each privilege in CA ACF2 must be given independently
Ex. WRITE does not give READ***

Sample Rule Set

\$KEY(SYS1)

BROADCAST UID(CHFSPSYS) R(A) W(A) A(L) E(A)

BROADCAST UID(*) R(A) W(A) ← Global Access Rule for Read and Write access

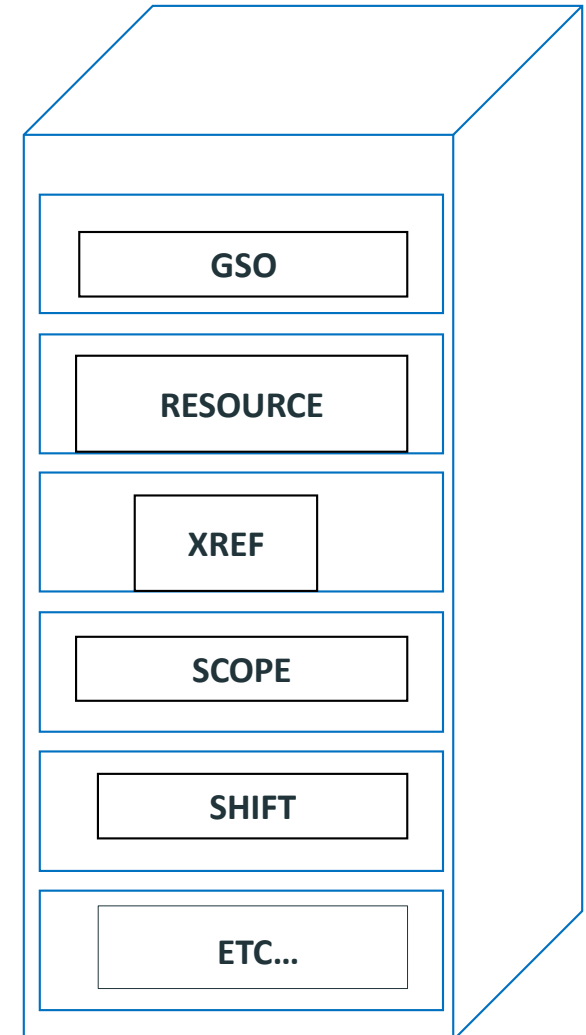
PARMLIB UID(CHFSPSYS) R(A) W(A) A(L) E(A)

PARMLIB UID(*) ← Global PREVENT rule so no one gets access

PROCLIB UID(CHFSPSYS) R(A) W(A) A(L) E(A)

Infostorage Database ~ Record Classes

- **GSO**
 - Global options used to initialize CA ACF2
- **Resource Rulesets**
 - Control the use of logical system resources (TSO, CICS, IMS, DB2, IDMS, Certs, User defined, etc).
- **XREF**
 - Allows for grouping of sources or resource rule
 - Treats groups as single entities
- **Scope**
 - Limit authority of privileged users
- **Shift**
 - Identify particular periods of time and dates



IBM® RACF® Basics

Julie Bergh



RACF

- Resource Access Control Facility
- The original mainframe security system (1976)
 - Prior, there was only UADS, the password file and dataset protection bits
 - Started IBM down the path to a consolidated user registry
- Uses dataset protection bits with discrete profiles; deleted with protected object
- Resource Oriented
- Generic profiles more policy-based, not attached to objects secured
- IDs are called “user IDs”
 - Note that term “UID” refers to the numeric z/OS UNIX identifier in the segment

RACF Modes

PROTECTALL (FAILURES | WARNING)

- When protect-all processing is active, the system automatically rejects any request to create or access a data set that is not RACF-protected.
 - The WARNING suboperand enables you to specify a warning message to the requestor in place of rejecting the request.
 - FAILURES specifies that RACF is to reject any request to create or access a data set that is not RACF-protected.

NOPROTECTALL

- Specifies that the system is not to check for RACF protection before it processes a request to create or access a data set.
 - NOPROTECTALL means that users can create and access data sets that are not RACF-protected.

RACF

Four kinds of security profiles:

- **User**
- **Group** - each user belongs to at least one Group
- **Dataset**
- **General Resource**
- **SETROPTS**

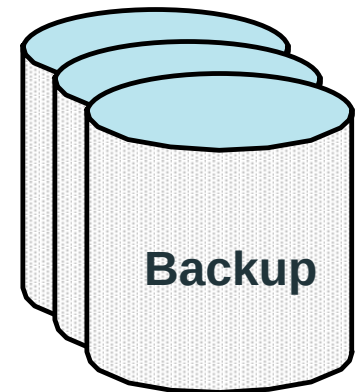
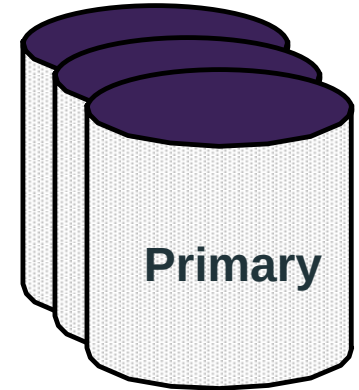
Note Both Dataset and General Resource profiles may be Discrete or Generic, and both have Access Lists

Security database

- **One (logical) primary and one secondary (backup)**
- **Can be split into several data sets to reduce contention**

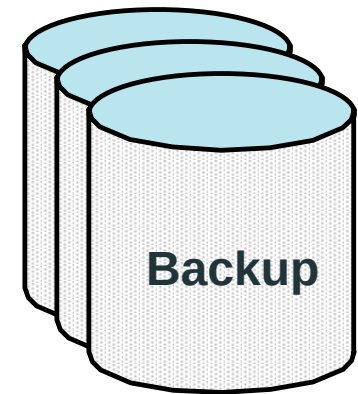
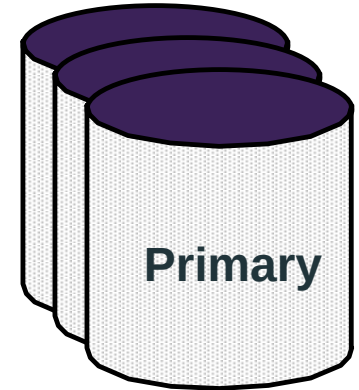
The RACF Database

- Proprietary format (precursor of VSAM). Non-relational.
- Defined by RACF Data Set Name Table (ICHRDSNT) and Range Table (ICHRRNG)
- Contains RACF profiles, system-wide options, templates, indices
- Can be shared across systems
- Built for speed
- Manage with batch utilities, RVAR command



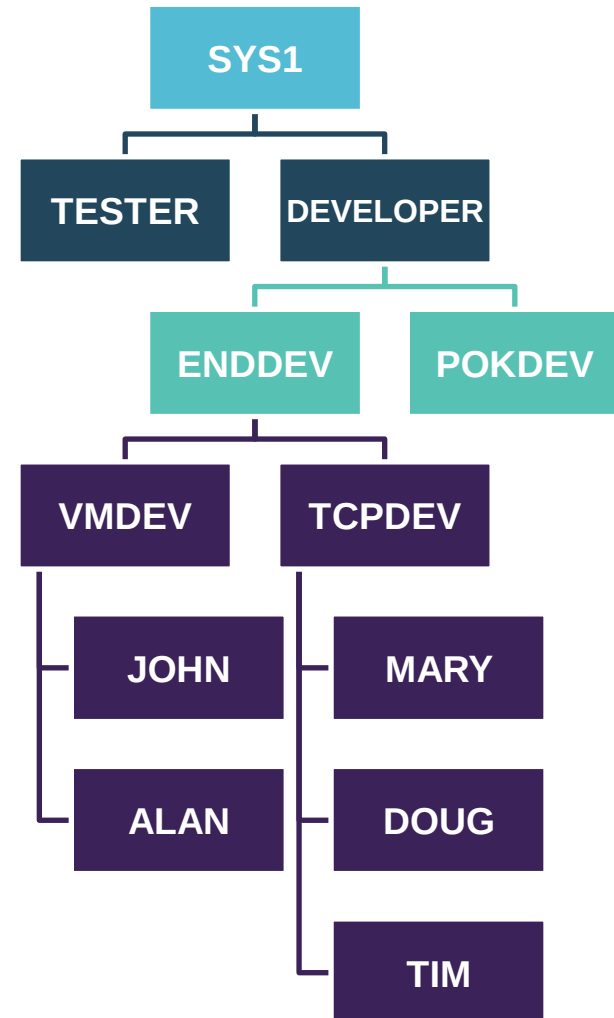
The RACF Database: Sharing

- DASD can be shared across LPARS/systems
- Coupling facility can be used to share in a Sysplex
- RACF Remote Sharing Facility (RRSF) can synchronize data across a network
- Can be accessed with LDAP protocols and synchronized to some extent that way (e.g. heterogeneous password synchronization with IBM Tivoli Directory Integrator)



RACF Group Tree Defines Profile Ownership Structure

- Give access rights to a group
- Connect users to one or more groups with different authorities
- Delegate group management
- Reduce administration effort



RACF Profiles

Represent users and groups

Protect “resources” (data sets, commands, services, programs, terminals, etc)

- General resource profiles are organized within “classes” (e.g. FACILITY, SERVAUTH, PROGRAM, etc) which are defined in the Class Descriptor Table (CDT)

Contain application-specific “segments” (e.g. OMVS, TSO, ICSF)

- Which contain “fields” (e.g. owner, access list, name, logging options)
 - Which contain data
 - “Custom fields” can be added dynamically, using the CFIELD general resource class, and the CFDEF segment

Mapped by “templates” (i.e. the schema)

Managed with TSO commands or RACF ISPF Panels (or Vendor add-on tools)

RACF Resource Profiles & Masking

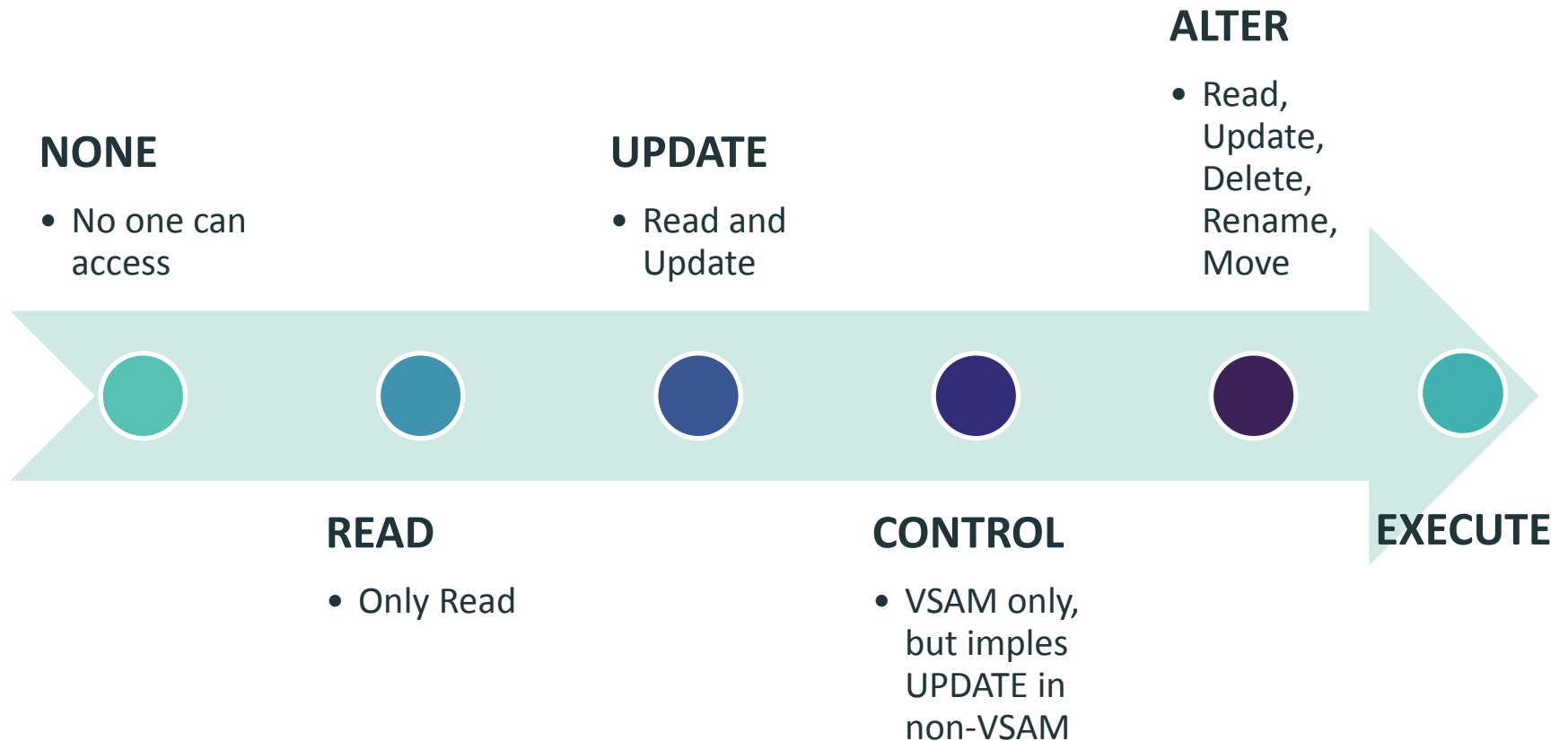
Are either discrete

- Protect a single resource with the same name as the profile

Or generic

- Use wildcarding to protect collections of resources
 - * to match to the end of a qualifier, or to match 1 or more qualifiers at the end of a name
 - ** to match 0 or more qualifiers
 - % to match a single character
 - & to specify a variable which defines a collection of unlike-named substitution strings (general resources only)
- A data set “fully qualified generic” profile has no generic characters, but protects the data set name regardless of the volume on which it resides

RACF Permissions



RACF General Resource Classes

IBM-defined classes defined in the CDT (ICHRRCDX)

- Customers cannot change
- Static – change requires IPL
- Sometimes (seldom) changes in the service stream when new function is added

Customer-defined general resource classes

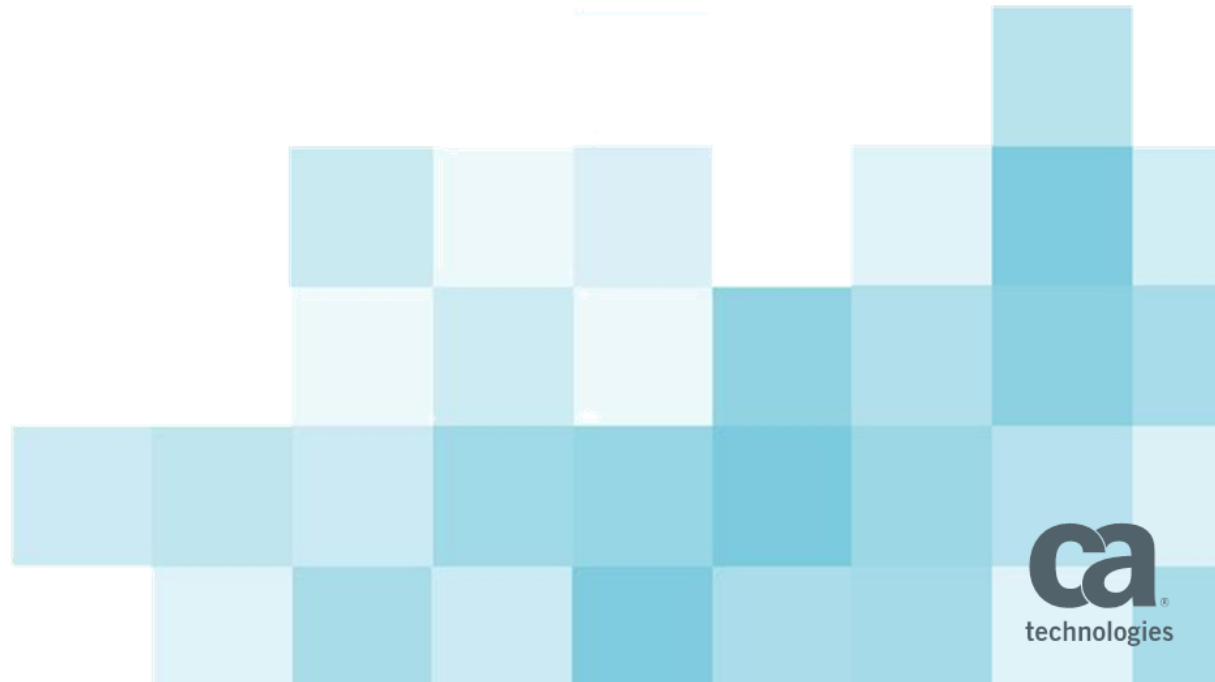
- ICHRRCDE – old, static method
- CDT general resource class, CDTINFO segment allows addition of new classes dynamically

Class-wide operations effected using SETROPTS command

- De/activate, read into storage (RACLIST), set logging options

CA Top Secret® for z/OS Basics

Carla Flores



CA Top Secret

- “Top Secret” aka “TSS”
- Developed by CGA Software Products Group in 1981
- Acquired by CA in 1985



CA Top Secret Security Modes

Dormant Mode

- Make sure the product is functional

Warn Mode

- Look for violations, patterns

Implement Mode

- Only what's explicitly secured

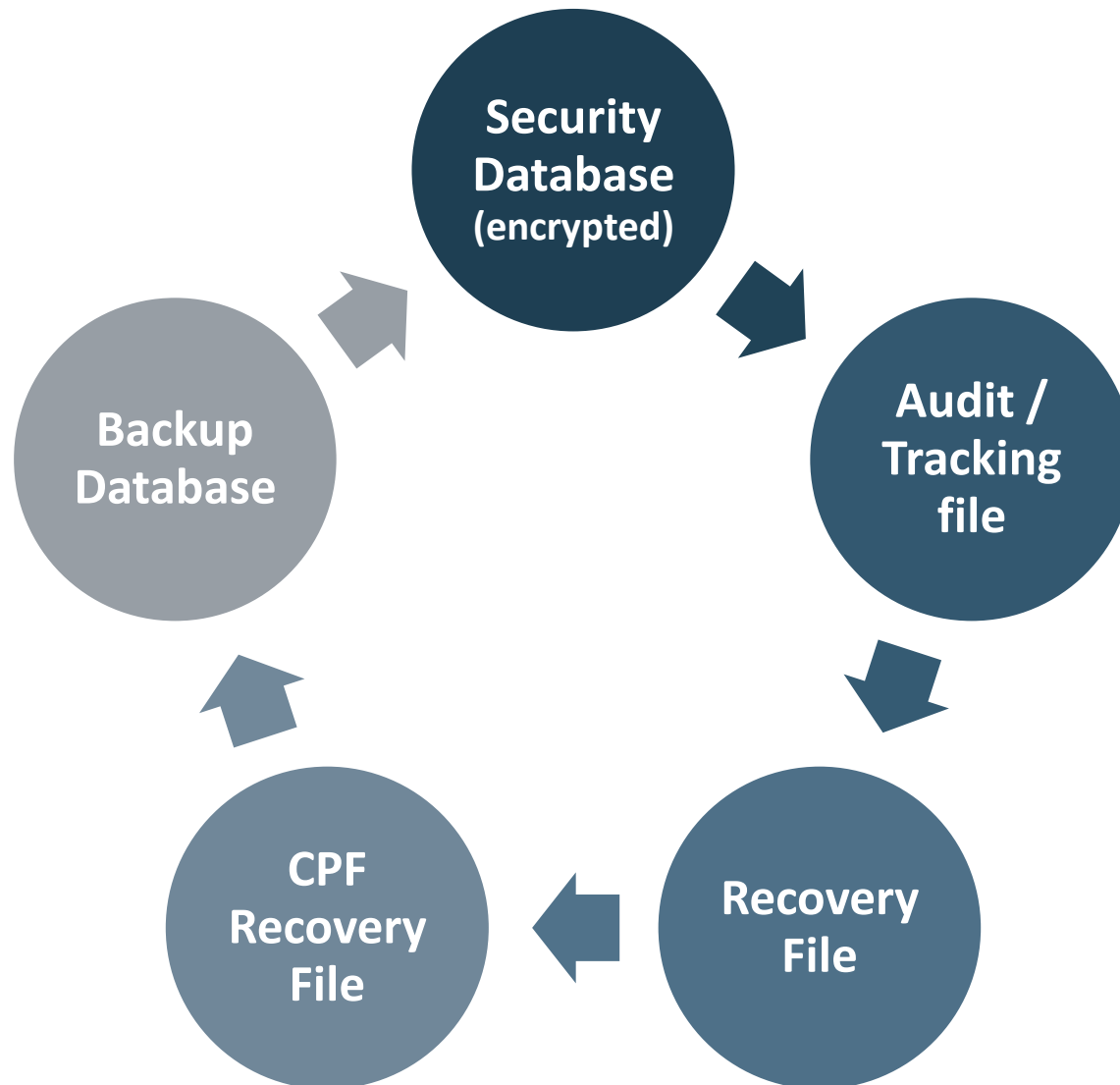
Fail Mode

- Mousetrap security

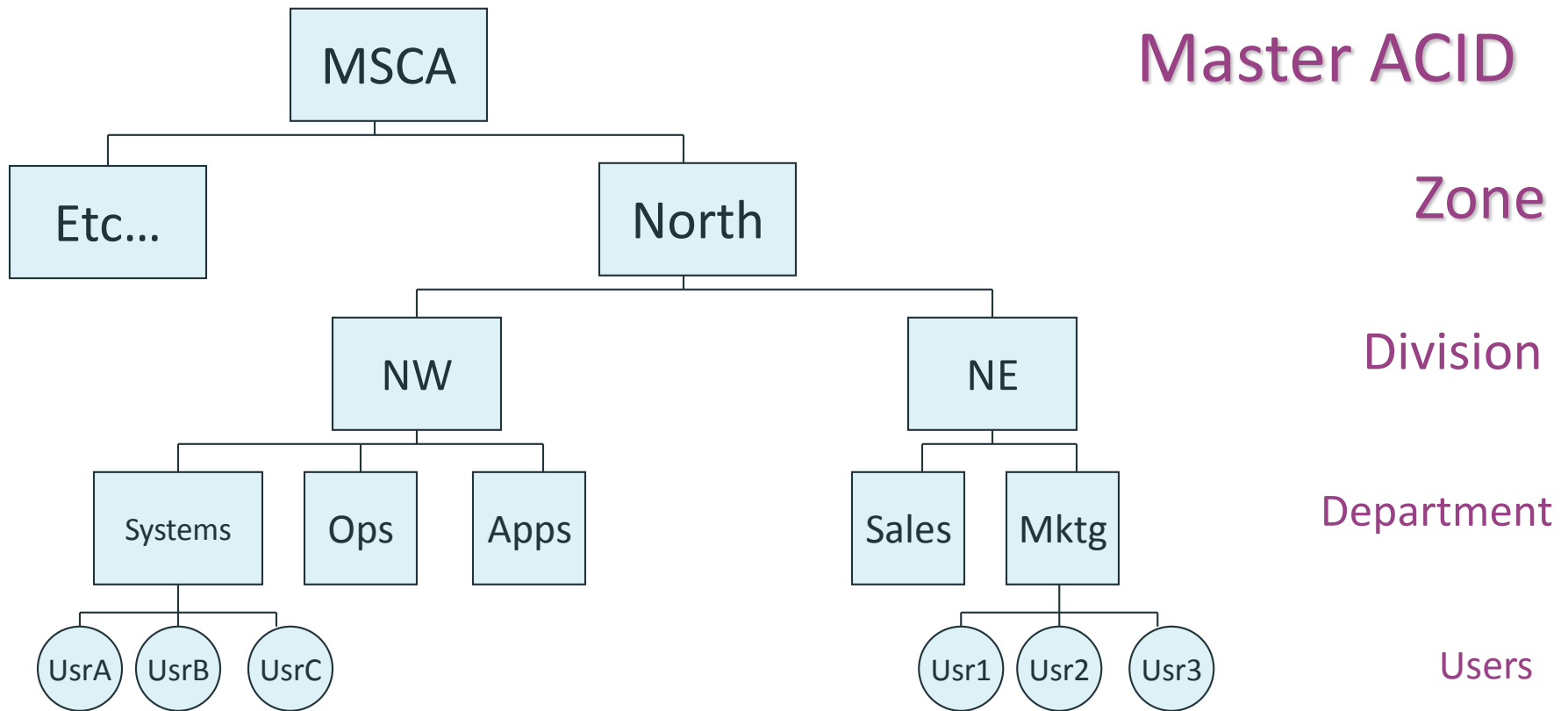
CA Top Secret

- Security database: 1 BDAM file, 1 VSAM file
- Can be synched using Command Propagation Facility (CPF)
- IDs are called “ACIDs” (pronounced **ay**-sids, for ACcessor IDs)
- Tree Structured
 - Everything (including ID’s) owned by someone
 - MSCA (Master Security Control ACID) is at the top of the tree
- User Oriented
 - Resources “owned” and “permitted”
- Facilities
 - Attributes, not resources
 - Not owned
 - May be added to ACIDs and PROFILES
 - Examples include: CICSPROD, CICSTEST, TSO, BATCH, STC

CA Top Secret Configuration - Data Sets



CA Top Secret Structure ~ Hierarchical Organization



CA Top Secret Structure

MSCA

- “Master Security Control ACID”
 - Owns Everything (“Root”)
 - For Installation, Maintenance
 - Encryption Key
 - Console messages issued for logons and failed logons
 - Never use it unless you have to

Other control ACID's

- SCA Central Security Control ACID
- LSCA Limited Central Security Control ACID
- ZCA Zone Control ACID
- VCA Divisional Control ACID
- DCA Departmental Control ACID

CA Top Secret Structure

Zones, Divisions, Departments

- Hierarchy
- Users can only belong to Departments (except for Control ACIDs)
- Departments can only belong to Divisions or the MSCA
- Divisions can only belong to Zones or the MSCA
- Zones only belong to the MSCA

ACIDs

- Any “node” in the hierarchical tree
 - Control ACIDs
 - Zones
 - Divisions
 - Departments
 - Users

Users

- Anything that can logon, whether front-line user, started task or Control ACID
- Access to resources by ownership or permission

CA Top Secret Structure

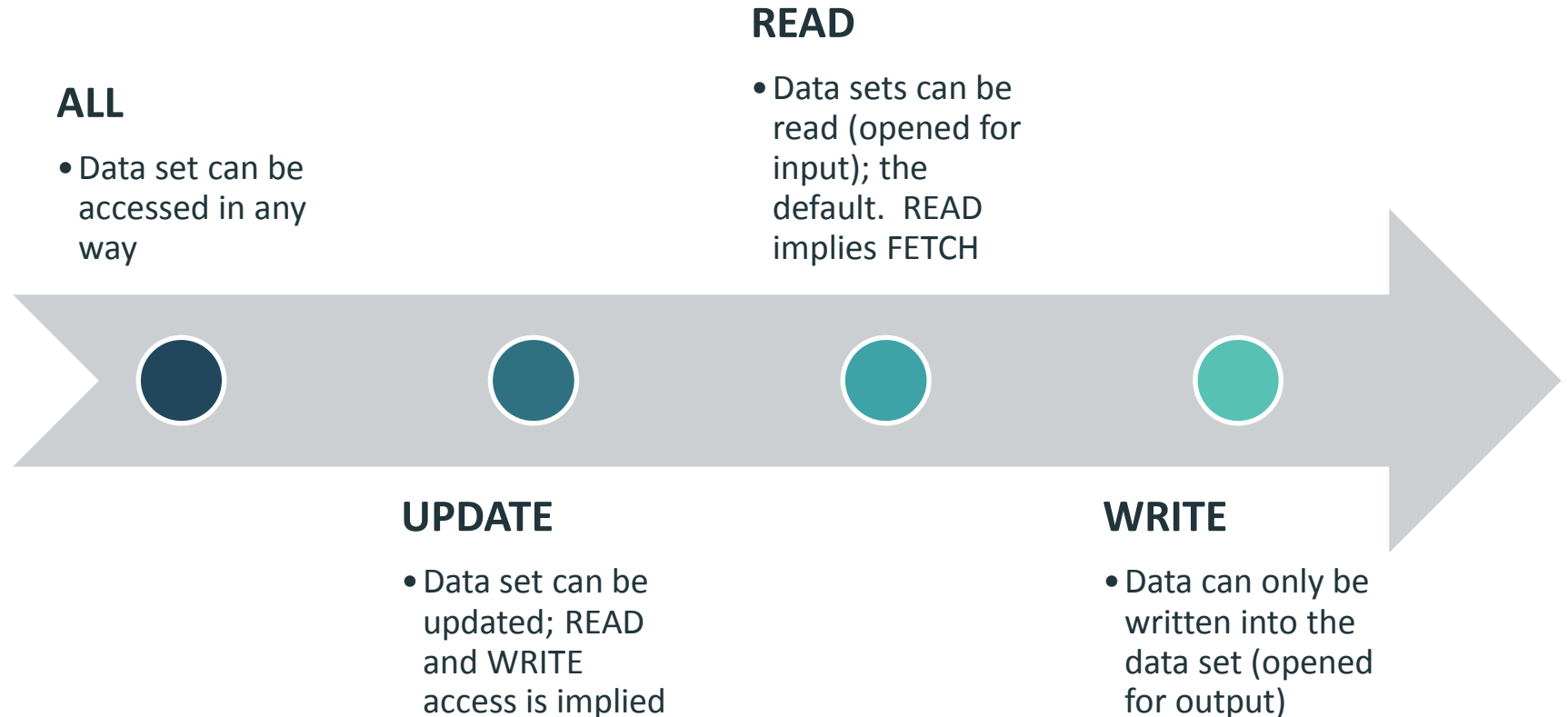
PROFILES

- Have access to resources and facilities, just like users
- A user can have many PROFILES
- Many users can have the same PROFILE
- An excellent way to give many users the same access, with the same changes
- Can be temporarily added

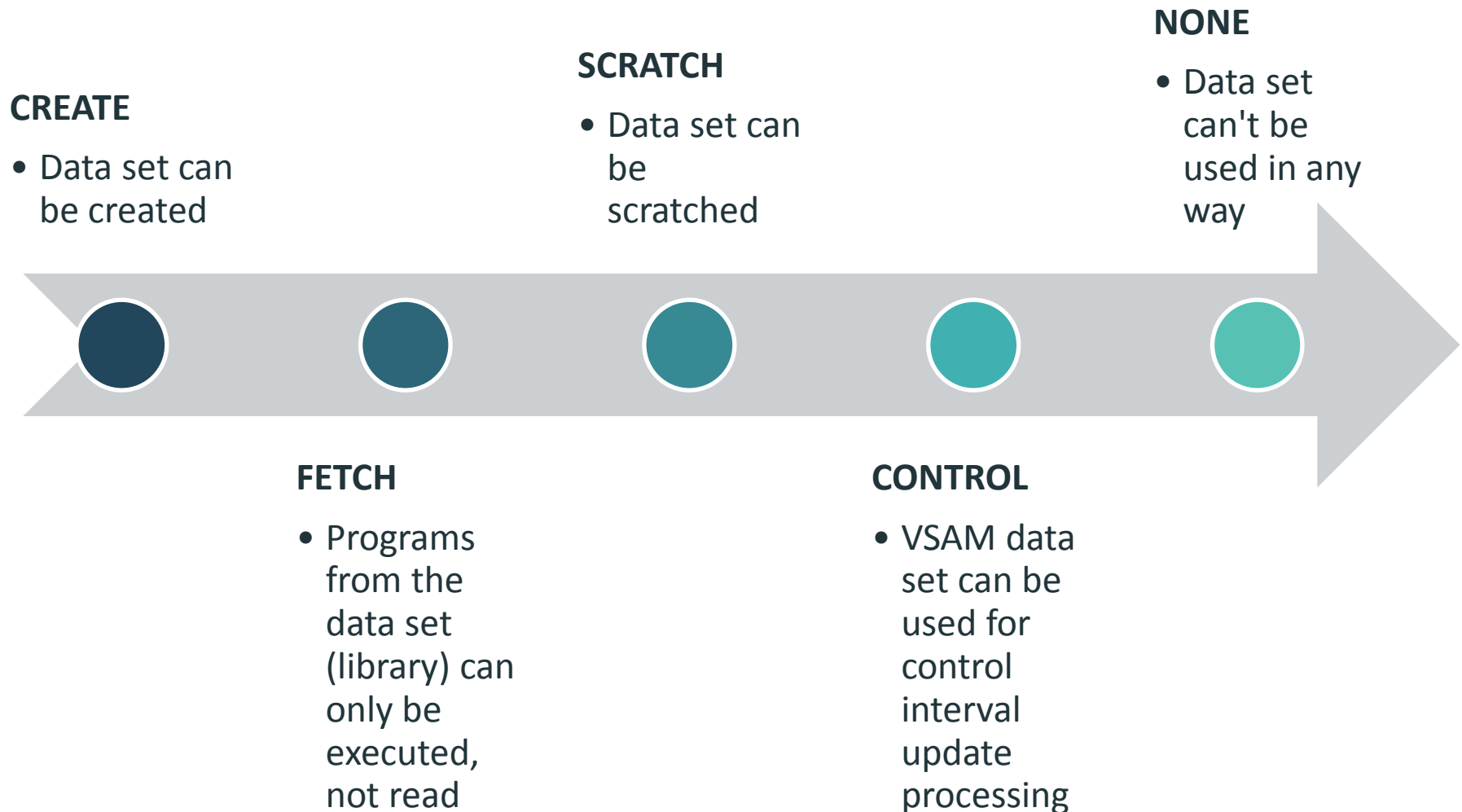
GROUPs

- Like profiles, but especially for use with UNIX System Services

CA Top Secret Structure - Permissions



CA Top Secret Structure - Permissions



CA Top Secret Structure - Other Records

NDT

- Node Descriptor Table
- Contains data for assigning Pass Tickets and Session Keys to applications.

RDT

- Resource Descriptor Table
- Stores both predefined and user-defined resources

FDT

- Field Descriptor Table
- Stores both predefined and user-defined fields.

SDT

- Stores internal, non-volatile data used to protect records, fields, screens, calendars, and other resources.

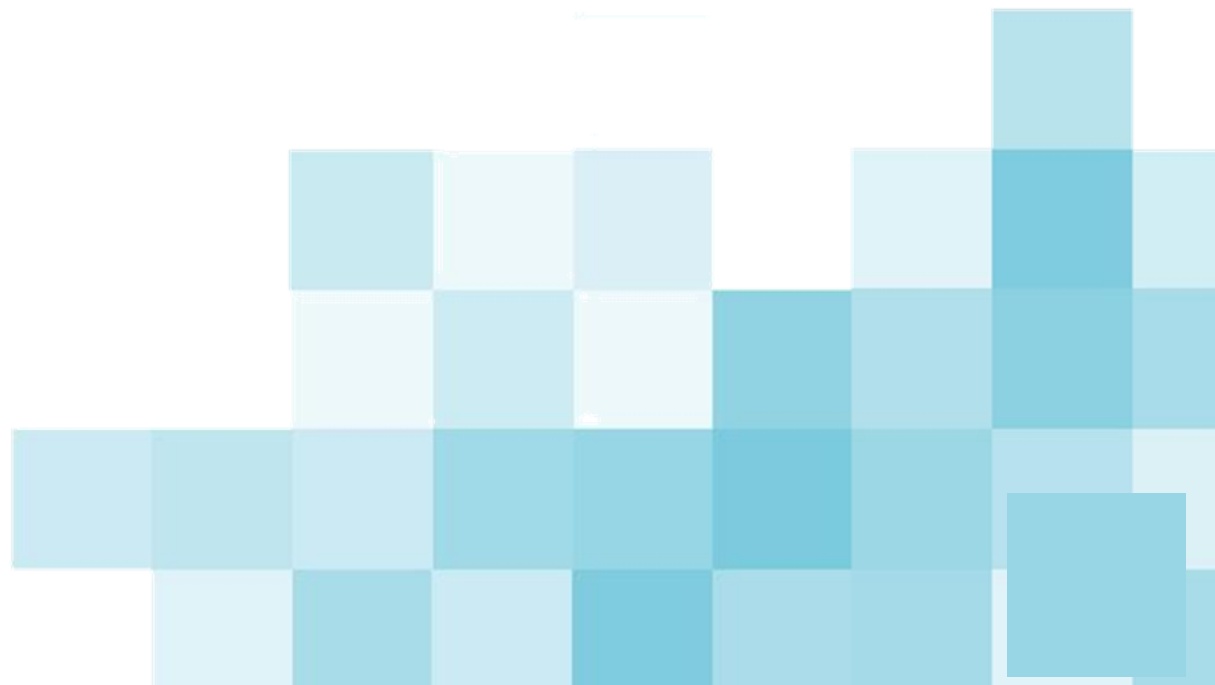
ALL

- Identifies resources that are globally accessible to all users.

STC

- Started Task Command
- Defines a started task command to CA Top Secret.

Q&A



Where to Find Out More...

- CA ACF2 Cookbook, CA Top Secret Cookbook and related manuals
 - Available on-line at www.support.ca.com
- IBM RACF Manuals and Red Books
 - Available on-line at www.ibm.com





Appendix

**Security Basics with examples
for each ESM**

Defining IDs

- RACF:

```
ADDUSER user_id DFLTGRP(group) PASSWORD(pwd)
```

- ACF2:

```
SET LID
```

```
INSERT logonid PASSWORD(pwd)
```

- TSS:

```
TSS CREATE(accessorid) DEPARTMENT(dept) PASSWORD(pwd)
```

Controlling System Entry

- Batch
 - RACF:
 - SETROPTS JES(BATCHALLRACF) forces all BATCH users to be defined to RACF
 - SETROPTS CLASSACT(JESJOBS)
 - PERMIT SUBMIT.*node.job.userid* CLASS(JESJOBS) ID(*userid*) ACCESS(READ)
 - ACF2:
 - Specify the JOBCK option of the GSO OPTS record
 - SET LID
 - CHANGE *logonid* JOB
 - TSS:
 - TSS ADDTO(*acid*) FACILITY(BATCH)

Controlling System Entry

- TSO
 - Master Catalog Alias
 - RACF:
 - ALTUSER *userid* TSO(ACCTNUM(*accnum*) PROC(*logonproc*))
 - ACF2:
 - SET LID
 - CHANGE *logonid* TSO
 - TSS:
 - TSS ADDTO(*acid*) FACILITY(TSO)

Revoking/Suspending Accounts

- RACF:
 - ALTUSER *userid* REVOKE
- ACF2:
 - SET LID
 - CHANGE *logonid* SUSPEND
- TSS:
 - TSS ADDTO(*acid*) SUSPEND

Access

- Defining Security for Datasets

- RACF:

- Discrete profile:

- ADDSD 'dsname' UACC(access)

- Generic profile:

- ADDSD 'dsname-incl-generic-char' UACC(access)

- **or**

- ADDSD 'dsname' UACC(access) GENERIC

- ACF2:

- \$KEY(high-level-qualifier)*

- dsname-extent UID(pattern-for-UIDs) R(A) and/or other accesses*

- TSS:

- TSS ADDTO(*acid*) DSNAME(*dsname*)

Access

- Permitting Access to Datasets
 - RACF:
 - PERMIT '*dsname-or-profile*' ID(*userid*) ACCESS(*access*)
 - ACF2:
 - \$KEY(*high-level-qualifier*)
 - dsname-extent* UID(*pattern-for-UIDs*) R(A) and/or other accesses
 - TSS:
 - TSS PERMIT(*acid*) DSNAME(*dsname*) ACCESS(*access*)

Access

- Grouping Access

- RACF:

- CONNECT *userid* GROUP(*group*)

- ACF2:

- SET LID

- CHANGE *logonid* DEPT(*dept*)

- TSS:

- TSS ADDTO(*acid*) PROFILE(*profilename*)

Passwords

- Changing a Password

- RACF:

- `ALTUSER(userid) PASSWORD(newpwd)`

- ACF2:

- `SET LID`

- `CHANGE logonid PASSWORD(newpwd)`

- TSS:

- `TSS REPLACE(acid) PASSWORD(newpwd)`

Digital Certificates

- Adding a certificate
 - RACF:
 - `RACDCERT ID(SYSMAN) ADD(MY.CERT) WITHLABEL('MIGRATED.KEY')
PCICC(*)`
 - ACF2:
 - `Set profile(user) div(certdata)`
 - `INSERT SYSMAN.cert DSN('MY.CERT') LABEL(MIGRATED.KEY) TRUST PCICC
PKDSLBL(*)`
 - TSS:
 - `TSS ADD(sysman) DIGICERT(EKMAUT01) HITRUST +
DCDSN('MY.CERT') +
LABLCERT('MIGRATED.KEY') PCICC +
LABLPKDS('IRR.DIGTCERT.CERTAUTH.EKMAUT01')`

Digital Certificates

- Export a certificate to z/OS dataset “MY.CERT”
 - RACF:
 - RACDCERT ID(SYSMAN) EXPORT(LABEL('SECURE.KEY')) DSN(MY.CERT)
FORMAT(CERTDER)
 - ACF2:
 - Set profile(user) div(certdata)
 - EXPORT SYSMAN LABEL(SECURE.KEY) DSN('MY.CERT') FORMAT(CERTDER)
 - TSS:
 - TSS EXPORT(sysman) DIGICERT(EKMAUT01) +
DCDSN('MY.CERT') +
FORMAT(CERTDER)

Displaying User Security Settings

- Listing a user's information
 - RACF:
LISTUSER *userid*
 - ACF2:
SET LID
LIST *logonid*
 - TSS:
TSS LIST(*acid*)

Admin Authority

- RACF:
 - SPECIAL, AUDITOR, OPERATIONS Attributes; scoped using group-versions
 - CLAUTH, Access and Profile Ownership
- ACF2:
 - ACCOUNT, SECURITY, LEADER, CONSULT, USER
 - Scoped by SCPLIST field defined in logonid record
- TSS:
 - ACID types: User, DCA, VCA, ZCA, LSCA, SCA