MVS Secured using CA-ACF2

Session ACF

MVS Secured using CA-ACF2

- CA-ACF2 architecture
- System entry validation
- Dataset Validation
- Resource Validation
- Auditing and Logging
- Administration
- Interfaces
- System Authorization Facility (SAF)

ARCHITECTURE

Components of CA-ACF2

- Runs as a subsystem
  - At SSI CA-ACF2 dynamically front-ends: TSO logon, data management routines, SVCs, SMF exit driver, SAF
- Has an STC
  - Checks that CA-ACF2 hooks are in place
  - Reads GSO infostorage records to set system options
  - Processes the CA-ACF2 commands
  - Process Cache functions

ARCHITECTURE

Components of CA-ACF2 continued

- Two SVCs
  - SVCA (system entry validation, resource validation, database access)
  - SVCS (dataset validation)
- JES2 exits / JES3 source mods and exits
- ACFFDR options module
- Three security files
  - Logonid records
  - Dataset rules
  - Infostorage records

ARCHITECTURE - Security Files

Logonid Record

- Identifies on-line, batch, and STC users to CA-ACF2
- Defines user privileges, attributes, and responsibilities
- LOGONID field record key
- 1024 byte record
- Requires ACCOUNT privilege to insert and delete
- ACCOUNT and/or SECURITY privilege can change

ARCHITECTURE - Security Files

Access Rule Sets

- Authorize access to datasets
- One rule set per DSN high level index or one rule set per VOLSER protected volume
- Key to record is high level index of dataset name or volume serial number
- Record length maximum of 4096 bytes
- Requires SECURITY privilege to compile and store access rules
- Owners may compile and store access rules for data which they own
- %CHANGE and/or %RCHANGE may compile and store specific access rules

ARCHITECTURE - Security Files

Multiple Infostorage Record Types

- Defines access authorizations for resources
- Allows for translation of source names
- Allows grouping of sources
- Allows grouping of resources
- Identifies scope of authority for administrators
- Determines time controls for system entry and data/resource access
- Contains system options for CA-ACF2
- Key to record up to 44 bytes
- Record length maximum of 4096 bytes
- Requires SECURITY privilege to maintain

ARCHITECTURE

GSO Records

- Used to customize CA-ACF2
- Can be dynamically modified
- Controlled by "Security Officer"
- Can be shared by multiple CPUs
- 27 GSO records
- Structured infostorage record

ARCHITECTURE - GSO Record Example

OPTS Record - CA-ACF2 Option Specifications

RECORD-ID	FIELDS
OPTS	BLPLOG/NOBLPLOG CMDREC/NOCMDREC CONSOLE(NOROLL/ROLL/NONE/WTP) CPUTIME(LOCAL/GMT) DATE(MDY/DMY/YMD) DDB/NODDB DFTLID(default-logonid) DFTSTC(ACFSTCID/logonid) INFOLIST(SECURITY,AUDIT/privilege list) JOBCK/NOJOBCK MAXVIO(10/nnn) MODE(ABORT/WARN/LOG/QUIET,RULE,no-rule,no-$mode) NOTIFY/NONOTIFY RPTSCOPE/NORPTSCOPE SAF[redacted] SAFTRACE/NOSAFTRACE SHRDASD/NOSHRDASD STAMPSMF/NOSTAMPSMF STC/NOSTC TAPEDSN/NOTAPEDSN UADS/NOUADS XBM/NOXBM

ARCHITECTURE

Simple Install with Basic Options

1. Unload tape

2. Tailor SMP jobs

3. Run SMP JOBS to receive, apply and accept CA-ACF2

4. Edit UM99901 to update the ACFFDR (change DSNs and SVCs)

5. Run SMP JOBS to receive, apply and accept usermod

6. Run DEFINE and INITIAL jobs to create the security files and add 1 authorized user to the LID file

7. Update SYS1.PARMLIB(IEFSSN00) add ACF2, ACF89SIP
   Update SYS1.PARMLIB(COMMND00) add ACF2
   Update SYS1.PARMLIB(JES2)/RESDSN(Jes3) & IATUX28,29,33

8. IPL with CLPA

ARCHITECTURE

Simple Install with Basic Options

9. Logon to TSO with authorized userid, proc(xxx) acct(xxx)

10. Exec the ACF command to change the mode to log and insert a dftlid with restrict and non-cncl

11. Run UADJOB to read uads and add users to the Logonid file

SYSTEM ENTRY VALIDATION

Overview

- SVCA validates system entry
  - Logonid record
  - ACVALD parameter list
- Called by TSO, JES, CICS, IMS, etc.
- Requirements:
  - Valid Logonid
  - Valid password
  - Valid input source
  - Valid shift
  - Valid authentication information

SYSTEM ENTRY VALIDATION

Additional Checks

- TSO
  - — UADS/NOUADS
  - — ACF2 fullscreen
  - — Automatic account and proc checking
- Batch
  - — Restricted LIDs
  - — Subauth and Program
- CICS, IMS, ROSCOE (MUSASS) etc.
  - — Jobfrom
  - — ACMCB/MLID
- Optional authority bit check for access to TSO, CICS, IMS

SYSTEM ENTRY VALIDATION

More Options

- GSO Records:
  - TSO
  - TSOKEYS
  - OPTS
  - PSWD
  - EXITS

See the Administrator's Guide

SYSTEM ENTRY VALIDATION

Distributed DataBase (DDB)

- Eliminates need for multi-Logonid definition across nodes
- Provides additional auditing information about users

SYSTEM ENTRY VALIDATION

Distributed DataBase (DDB)

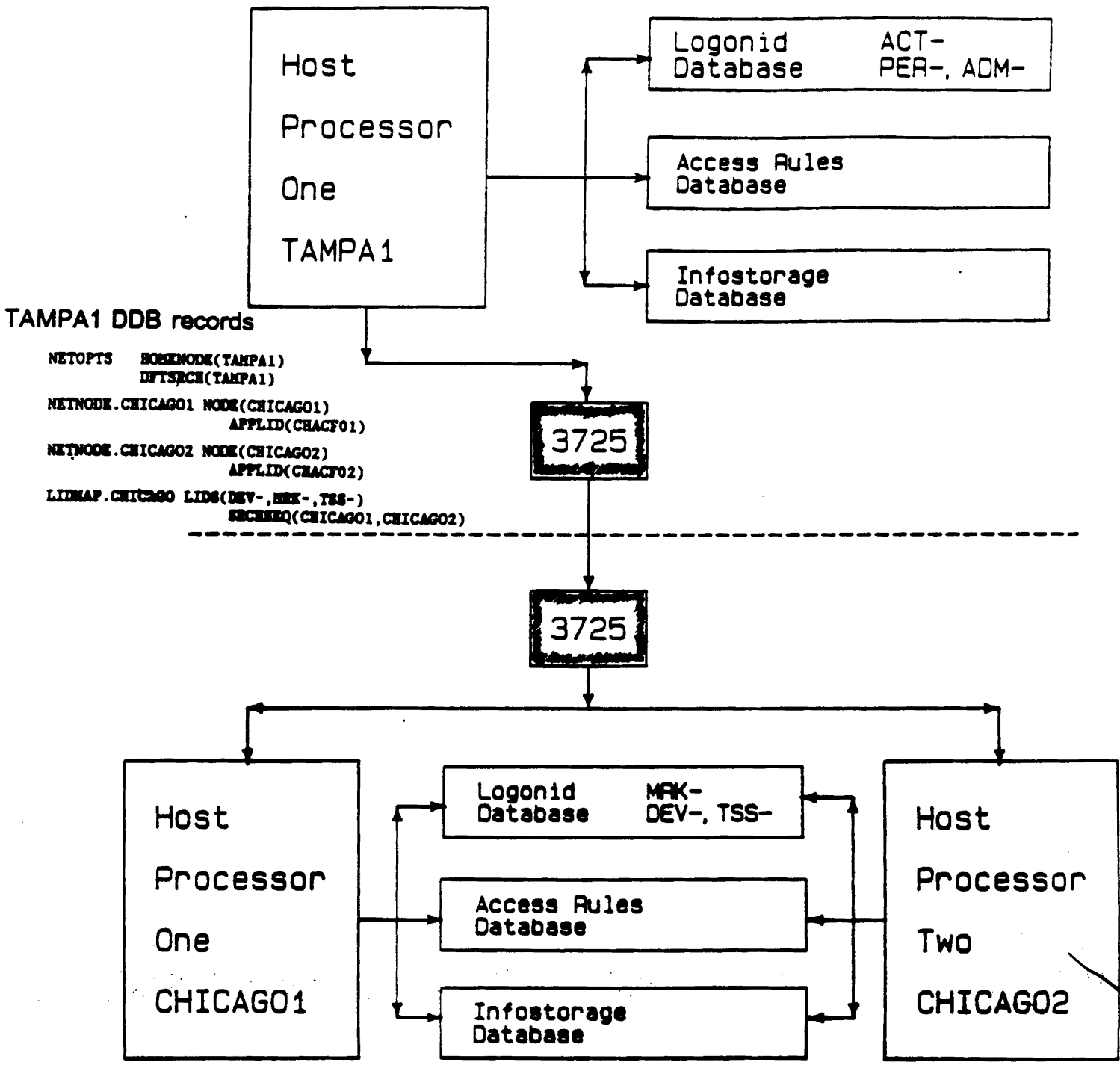
- Defining the DDB Network
  - Information storage records
    - Storage Class = "C"
    - Type Code = "NET"
    - Record Name = "NETOPTS", "NETNODE" or "LIDMAP"

NETOPTS	**Used to maintain information about the node, and information specific to DDB support. Each node must have one and only one NETOPTS record per Global Systems Option (GSO) SYSID.**
NETNODE	**Used to identify and maintain information about other nodes in the network. Each node must have at least one NETNODE record for every other node in the network.**
LIDMAP	**Used to maintain masks of Logonids and identify a node search sequence to be used in locating the node where matching Logonids are defined.**

SYSTEM ENTRY VALIDATION

Distributed DataBase (DDB)

- Sample Configuration



SYSTEM ENTRY VALIDATION

Extended User Authentication (EUA)

- Allows for additional authorization of system access
- Up to 8 EUA processing routines can be installed
- One EUA device per Logonid record
- EUA processing applies to system entry via TSO and CICS only

SYSTEM ENTRY VALIDATION

Extended User Authentication (EUA)

- EUA Processing

  — Logon

  — Preprompt- Sets up ACVALD parmlist and invokes SVCA.

  — SVCA - Validates system entry then checks the AUTHSUP byte in the Logonid record. If a bit matches an AUTHEXIT GSO record control is passed to the EUA processing program. If INFO bit is on in the AUTHEXIT GSO record a infostorage record is read and passed to the exit.

  — EUA program - Can prompt a user and get back response. Also can request svca insert, update or delete infostorage records.

SYSTEM ENTRY VALIDATION

EUA Implementation

- Identify AUTHSUPn bit to be used for EUA routine
- Create GSO AUTHEXIT record
- Create GSO APPLDEF record (optional)
- Set AUTHSUPn bit in selected Logonid records

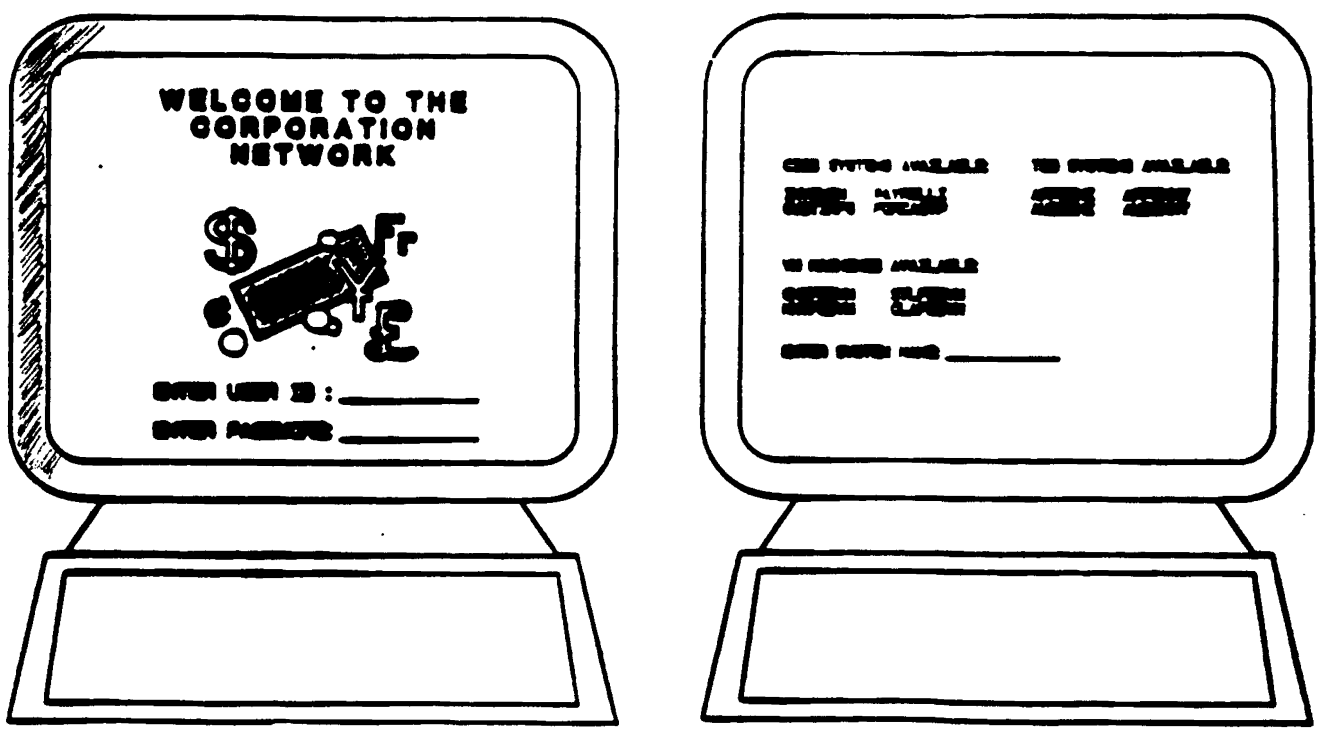
AUTHEXIT—EXTENDED USER AUTHENTICATION EXIT

RECORD-ID	FIELDS
AUTHEXITqual	LIDFIELD(attribute name) PROCPGM(processing-program-name) INFOSTG/NOINFOSTG

SYSTEM ENTRY VALIDATION

VTAM Common Signon Features

- Secure access via resource rules
- Perform multiple access validation with one call
- Application ability to use Information Storage
- **Cross-address space inheritance**



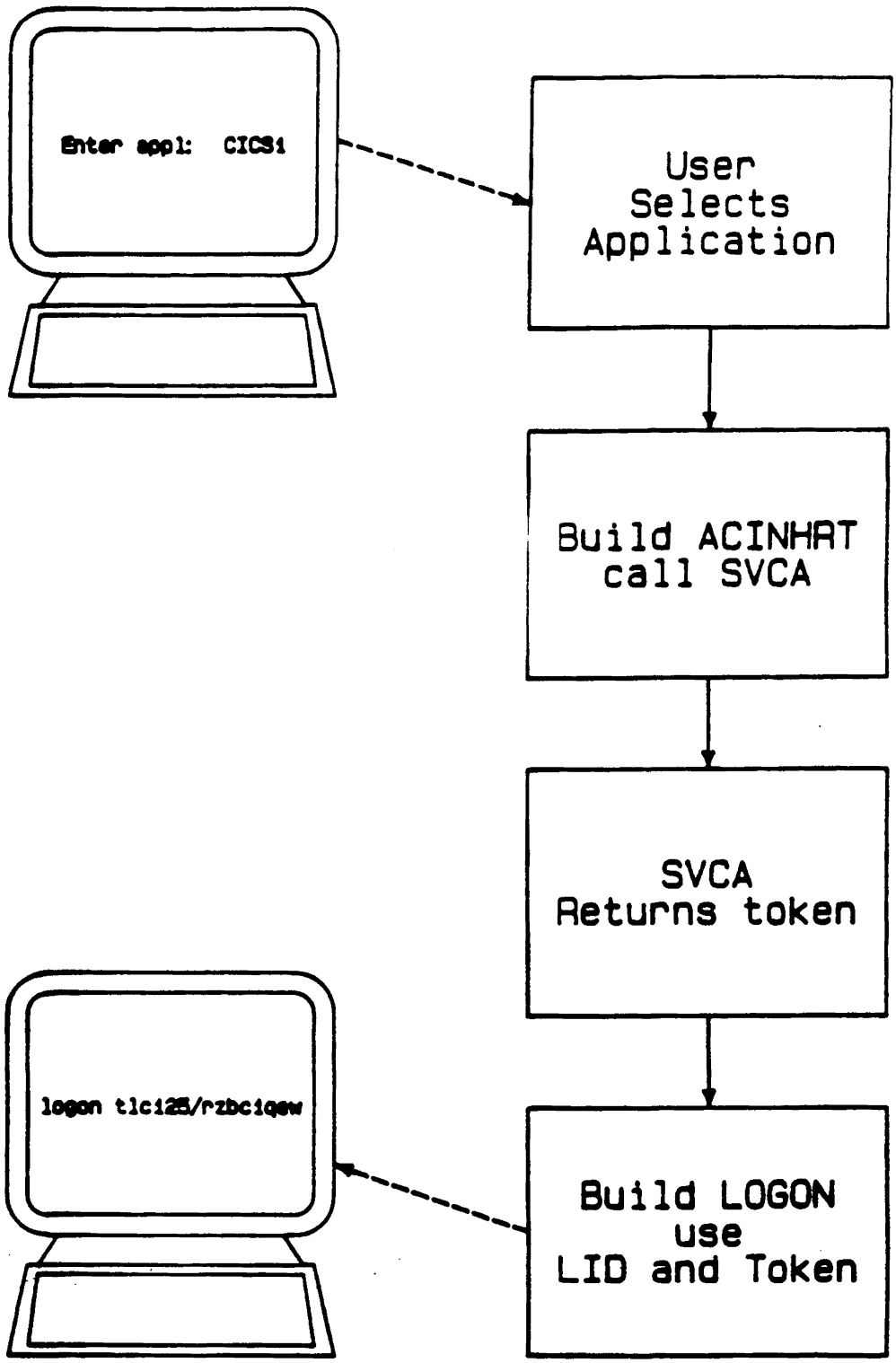
SYSTEM ENTRY VALIDATION

VTAM Common Signon Logon Inheritance

- No password saving required
- Pass a unique one-shot token in lieu of a password
- How to use
  - — Invoked via ACINHRT parameter block
  - — Interface specifies user's Logonid, terminal name, and name of application
- SVCA returns a one-shot inheritance indicator token
- Token used when user signon requested

SYSTEM ENTRY VALIDATION

VTAM Common Signon Logon Inheritance Flow



DATASET ACCESS VALIDATION

Overview

- SVCS validates access to DASD and tape
  - Access rule
  - ACDSV parameter list
- Called by data management intercept, SAF, etc.
- Validation process
  - User identification
  - Dataset name protection
  - Volume protection
  - Owner access
  - Access rule check

DATASET ACCESS VALIDATION

User Identification (UID) String

- Identifies users within groups
- Is constructed of Logonid record fields
- Often contains user-defined fields
- Format is defined in the ACFFDR
- Length is 1 - 24 characters
- Is used in rules to allow grouping of access

DATASET ACCESS VALIDATION

@UID Macro

- Located in the ACFFDR
- Tells CA-ACF2 how to build the UID string at session initiation
- Specifies the order in which the fields will be concatenated
- Example: @UID LOC,DIV,DEPT,JOBF,LID
  - LOC = 1st and 2nd characters
  - DIV = 3rd character
  - DEPT = 4th and 5th characters
  - JOBF = 6th through 8th characters
  - LID = 9th through 16th characters

New feature
"GROUP" attribute.
Allows dynamic update at logon time.

DATASET ACCESS VALIDATION

Access Rule Sets

- Rule Entry Format

$KEY(high-level-index)

"THE ENVIRONMENT":

```
dsn-mask  VOL(vol-mask)  UID(uid-mask)  SOURCE(name) -
          LIB(lib-mask)   PGM(pgm-mask)   DDN(ddn-mask) -
          SHIFT(shift-name)  UNTIL(mm/dd/yy) / FOR(days)
```

"ACCESS PERMISSIONS":

```
READ(A/L/P)   WRITE(A/L/P)   ALLOC(A/L/P)   EXEC(A/L/P)

      A = Allow with no logging
      L = Allow and log
      P = Prevent and log (default)
```

"MISCELLANEOUS KEYWORDS"

```
DATA(local-data)   NEXTKEY(alternate-$key)
```

EXAMPLE:

```
$KEY(SYS1)
 LINKLIB    UID(CHFSP)  READ(A)  WRITE(L)
 PARMLIB    UID(CHFSP)  READ(A)  WRITE(L)
```

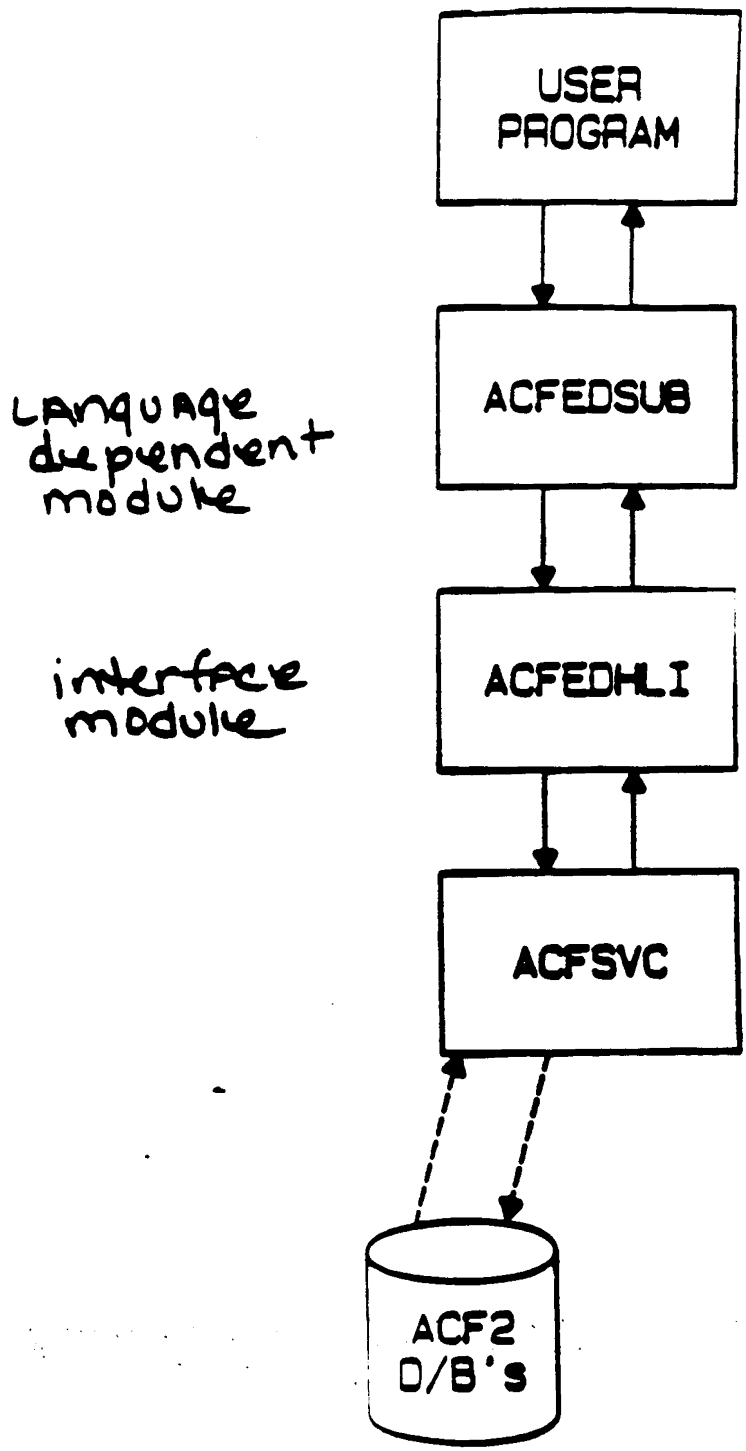
HIGH LEVEL INTERFACE (HLI)

Features

- HLI can be used to call CA-ACF2 to:
  - Validate system entry
  - Return Logonid of a user
  - Partially encrypt a password
  - Re-validate a password
  - **Validate Datset Access**
  - Validate generalized resource access
  - Build and/or delete a resource rule directory
  - Perform a CA-ACF2 SVCA or SVCS call
- Supports COBOL, FORTRAN, and PL/I
- CICS compatible
- Comprehensive error checking
- Maintains compatibility between releases

HIGH LEVEL INTERFACE (HLI)

Overview



RESOURCE VALIDATION

Resource Rules

- SVCA validates access to resources
  - — Resource rules
  - — ACGRSRC parameter list
- Called by TSO, CICS, any application
  - — TSO resources (accounts, procs, commands)
  - — CICS resources (transactions, files, programs)
  - — IMS resources (transactions, AGNs)
  - — IDMS resources (data areas, subschemas, programs)
  - — Any other "defined" resource
- Can interpret multiple rules with one SVCA call

RESOURCE VALIDATION

Format of Resource Rulesets

$KEY(resource-name-mask)

$TYPE(type-code)

$USERDATA(local-data)

%CHANGE uid1...uidn

THE ENVIRONMENT

UID(uid-mask) SOURCE(source-name) SHIFT(shift-name) -
UNTIL(mm/dd/yy)/FOR(days) SERVICE(read,add,update,delete)

ACCESS PERMISSIONS

ALLOW / LOG / PREVENT

ALLOW = Allow with no logging
LOG = Allow and log
PREVENT = Prevent and log (default)

MISCELLANEOUS KEYWORDS

DATA(local-data) VERIFY

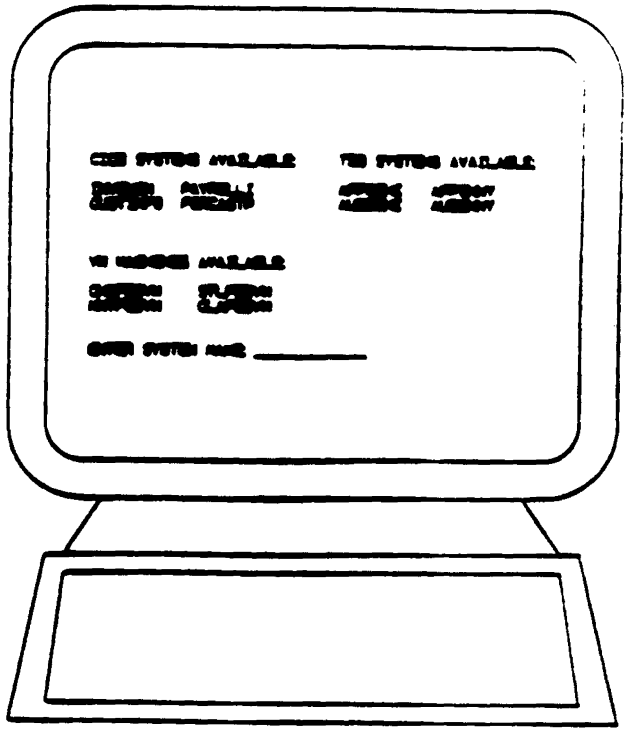
EXAMPLE:

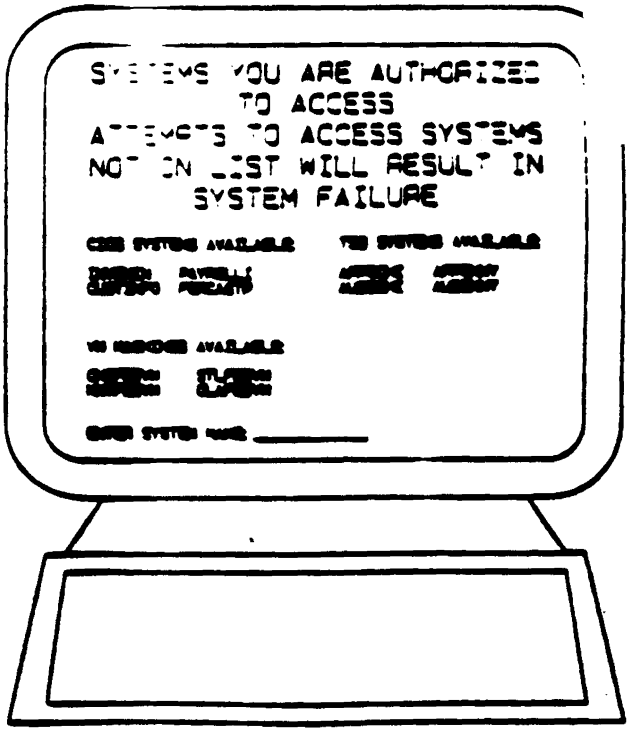
```
$KEY(PAYT) TYPE(CKC)
 UID(CHHPDMGR) ALLOW
 UID(CHHPDCLK) LOG
```

RESOURCE VALIDATION

VTAM Common Signon

- **Secure access via resource rules**
- **Perform multiple access validation with one call**
- Application ability to use Information Storage
- Cross-address space inheritance

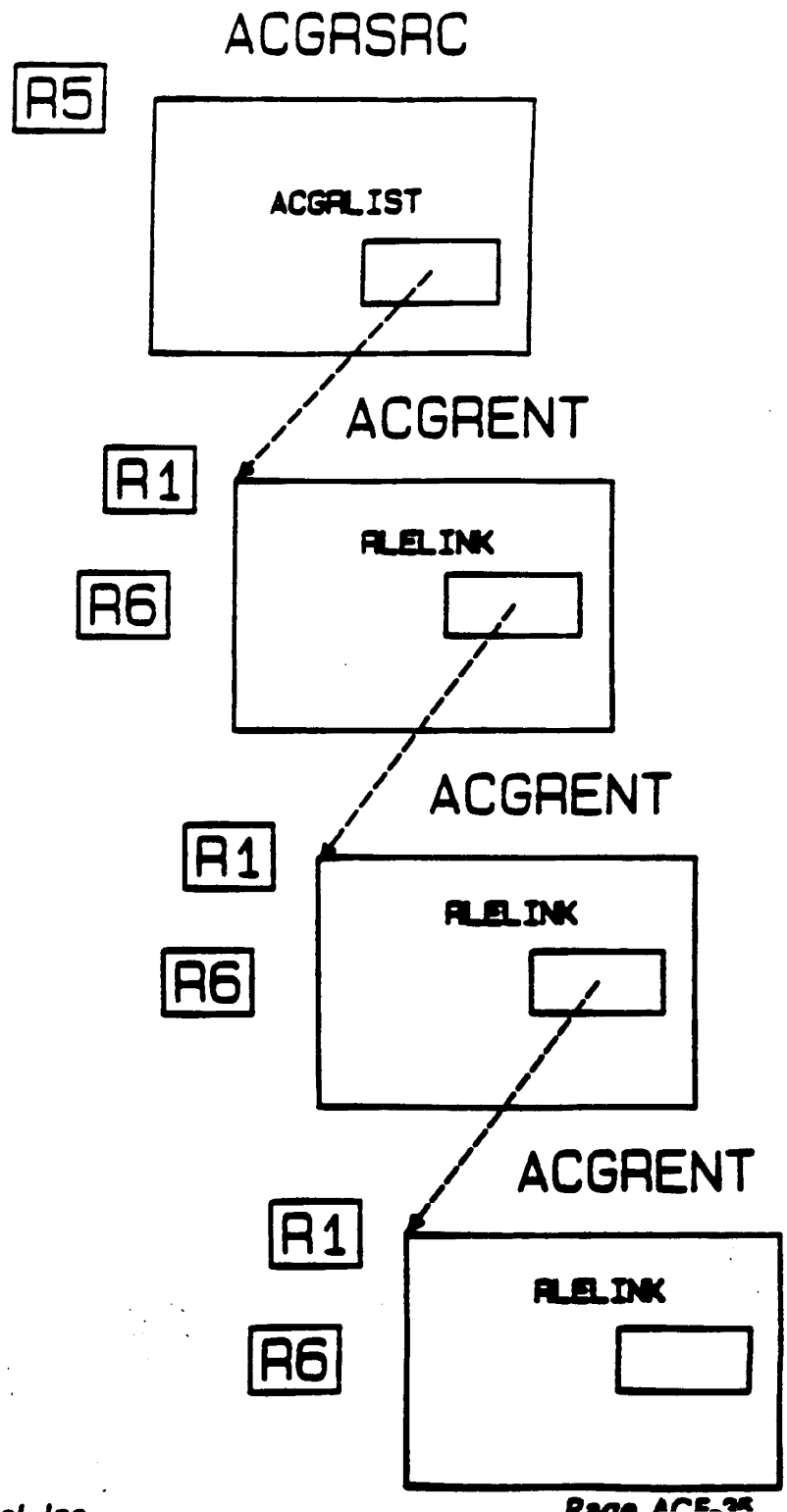




RESOURCE VALIDATION

VTAM Common Signon - Rule List Interpret

```
USING ACGRSRC, R5

GETMAIN R, LV=(0)
ST R1, ACGRLIST
LR R6, R1
USING ACGRENT, R6

(Fill in ACGRENT)

GETMAIN R, LV=(0)
ST R1, RLELINK
LR R6, R1

(Fill in ACGRENT)

GETMAIN R, LV=(0)
```



RESOURCE VALIDATION

Infostorage Records

- SVCA retrieves and stores data from the infostorage database to be used for validation
  - — Structured infostorage records
  - — Unstructured infostorage records
  - — ACNTRY parameter list
- Called by ACF2 command, SAF. CICS, any application
- Structured record
  - — GSO OPTs record
  - — GSO AUTHEXIT record
- Unstructured record
  - — Entry list (sources)
  - — Time/shift records
  - — Scopelist

RESOURCE VALIDATION

Structured Infostorage Records

- Extended User Authentication example

  1. Create RSB module to define the record

  2. ACF
     ```
     SET CONTROL(GSO)
     INSERT APPLDEF.OID CLASS(I/Identity)
                        TYPE(AUT/AUTHSUP)
                        APPLDIV(OID)
                        APPLDLEN(8)
                        DFTDRTN(ACFOOID)
                        RECID(ACFOIRSB/-)
                        RECIDLEN(8)
                        SELAUTH(ACCOUNT)
     END
     ```

  3. F ACF2,REFRESH(APPLDEF-)

- Record Key
  I-AUT-OID-xxxxxxxx

- ACF
  ```
  SET I(AUT) DIV(OID)
  INSERT xxxxxx OIDCARD(xxxx)
  END
  ```

RESOURCE VALIDATION

Structured Infostorage Records

- VTAM Common Signon example

1. Structured infostorage record

```
SET CONTROL(GSO)
APPLDEF.VCS      CLASS(3/VTAMNET)
                 TYPE(VCS/COMSIGN)
                 APPLDIV(-)
                 APPLDLEN(8)
                 DFTDRTN(ACFOODFT)
                 RECID(PARMSRGB/PARMS)
                 RSBLIB(RSB.LOADLIB)
                 RECIDLEN(8)
                 SELAUTH(SECURITY)
END
```

2. Record Key

```
3-VCS-xxxxxxxx-PARMS
```

3. ACF command support

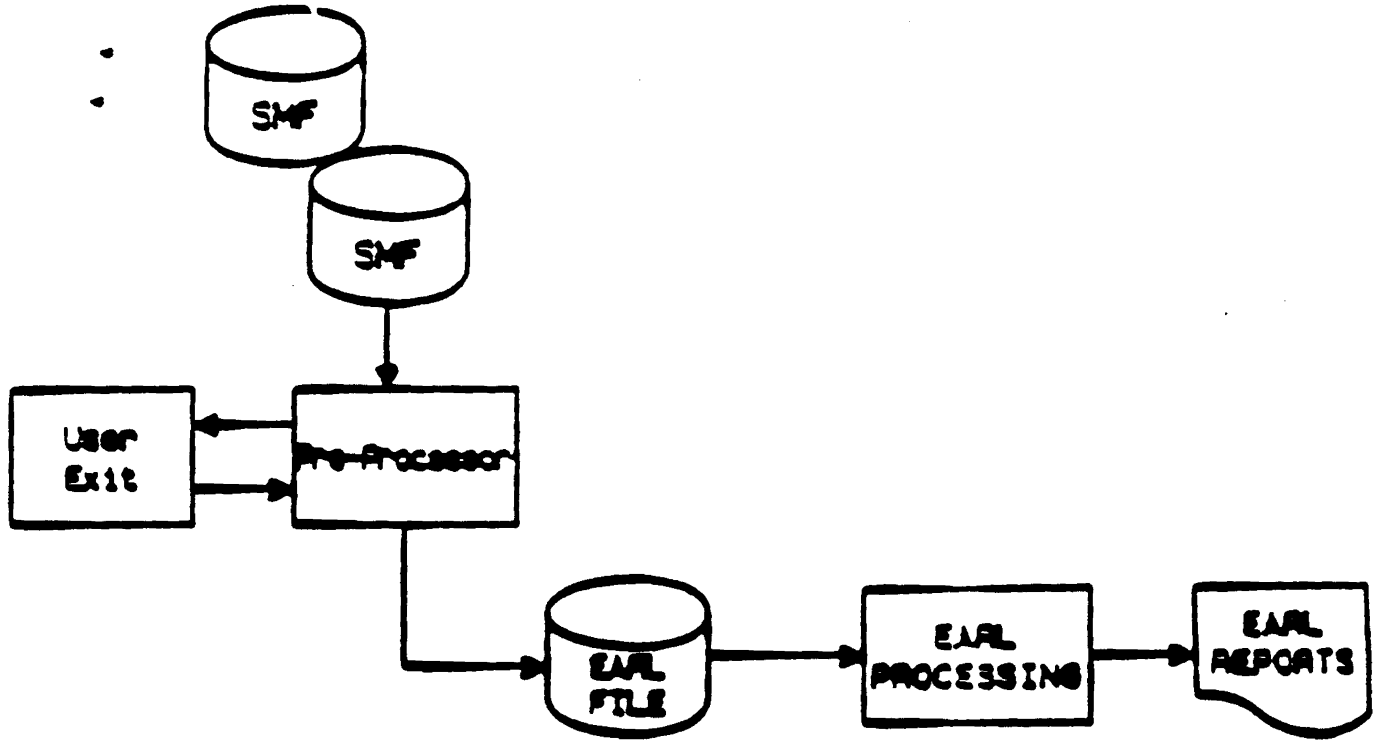
```
ACF
    SET VTAMNET(COMSIGN) DIV(TSO1)
    INSERT PARMS(xxx)
    CHANGE PARMS(xxx)
    DELETE PARMS(xxx)
END
```

AUDITING AND LOGGING - REPORTS

ACFRPTCR	TSO Command Statistics Log
ACFRPTDS	Dataset/Program Event Log
ACFRPTEL	Information Storage Update Log
ACFRPTIX	Dataset Index Report
ACFRPTJL	Restricted Logonid Job Log
ACFRPTLL	Logonid Modification Log
ACFRPTNV	Environment Report
ACFRPTPP	Pre-Processor
ACFRPTPW	Invalid Password/Authority Log
ACFRPTRL	Rule-Id Modification Log
ACFRPTRV	General Resource Event Log
ACFRPTRX	Logonid Access Report
ACFRPTSL	Selected Logonid List
ACFRPTXR	Cross-Reference Report

AUDITING AND LOGGING

CA-EARL Report Processing Overview



ADMINISTRATION

Scope List

- SCPLIST - The Scope List

  — This Logonid field, along with the authorization fields, reduces the scope of a user's administrative authority.

  — The "scope-list-name" points to a record on the Information Storage database that contains the CA-ACF2 elements within the range or "scope" of the privileged user.

  — The scope list record specifies a list of Dataset high level qualifiers, Logonids, UIDs, and Infostorage keys to be the authorized control limits of a privileged user.

ADMINISTRATION

Commands

- To create/change/display/delete:
  - Logonid records
  - Access rule sets
  - Generalized resource rules
  - Global system options
  - Entry lists
  - Shift records
  - Zone records
  - Scopelists
- Includes HELP members

ADMINISTRATION

Administration Using ISPF

```
---------- ACF2 SPF OPTION SELECTION MENU ----------

   SELECT OPTION====> 6_


   1  RULES     - PROCESS ACF2 ACCESS AND GENERALIZED RESOURCE RULES
   2  LOGONIDS  - ACF2 LOGONID CREATION/MAINTENANCE FACILITY
   3  SYSTEM    - ACF2 SHOW COMMANDS
   4  REPORTS   - ACF2 REPORT PROGRAM PROCESSOR
   5  UTILITIES- PROCESS ACF2 UTILITIES
   6  GSO       - GLOBAL SYSTEM OPTIONS SERVICES
   7  NET       - NETWORKING SYSTEM OPTIONS SERVICES
```

ADMINISTRATION

CA-ACF2 Console Commands

- Standard CA-ACF2 modify requests:

  F ACF2,BACKUP

  F ACF2,RESET(logonid)

  F ACF2,RELOAD(rule-id)

  F ACF2,REBUILD(directory-type)

  F ACF2,NEWXREF,TYPE(RGP/SGP)

  F ACF2,NEWSHIFT

  F ACF2,SETNORUL(jobname/ALL)

- CA-ACF2 GSO modify requests:

  F ACF2,SHOWSYS

  F ACF2,SETSYS(sysid)

  F ACF2,REFRESH(recid/ALL)
  Requires REFRESH bit privilege in Logonid

INTERFACES

CA-ACF2/CICS Features

- Dynamic terminal timeout
- Logonid inheritance for batch jobs submitted from the transient data queue
- HLI support
- EUA support
- Application Program Interface
- ACFM
  - Full ACF command support
  - Dynamically modify options
  - Display storage statistics
- Can co-exist with CICS transaction, resource, or external security
- Dynamic intercepts
- LU 6.2 support (APPC)

INTERFACES

CA-ACF2/CICS Resource Types

CKC Transaction level validation

CPC Program validation

CFC File validation

CTD Transient data validation

CTS Temporary storage validation

CMR MR●(IRC/ISC) validation

PSB PSB validation for DL/I calls

CDB DBD validation

INTERFACES

CA-ACF2/CICS MRO (IRC/ISC) Support

Provide Logonid Inheritance

- Tailor inheritance

```
MRO        SYSID    = (mask)
           DEFAULT  =
           IMPLSIGN = NO/YES
           SIGNEXIT = NO/YES
           RECEIVE  = NO/YES/FS/TR
           TRANSMIT = NO/YES/FS/TR
```

- Use resource rules to validate a user's access to/from a SYSID

```
CICSKEY RESOURCE=MROIN,OPTION=VALIDATE,TYP=CMR CICSKEY
RESOURCE=MROOUT,OPTION=VALIDATE,TYP=CMR
```

INTERFACES

CA-ACF2/CICS MRO (IRC/ISC) Support

Resource rule format:
$Key(ssss.ff.aaa) TYPE(CMR)

ssss = SYSID of receiving region outbound request or SYSID of transmitting region inbound request

ff = TR or FS

aaa = IN or OUT

UID = user making the request

- MRO inbound exit
- MRO outbound exit

INTERFACES

CA-ACF2/IMS Features

- Can co-exist with IMS security
- Signon, transaction, and AGN security
- Dynamic intercepts
- Structured into storage rules to store options
- XRF support
- IMS batch support

INTERFACES

CA-ACF2/DB2 Features

- Multi-vendor support
- Single registration point for userid and security information
- Resource rules used to secure access to tables, views, plans, and field/column access
- Share rules across multiple DB2 systems
- Separation of function security administrator/database administrator
- Security enhancements:
  - — Source checks
  - — Shift
  - — Mode
  - — Pre/post validation exits

INTERFACES

CA-ACF2/VAX Features

- Multi-hardware support
- Single point of registration for user and security information
- Central administration of UAF and NETUAF files accross VAX machines
- Changes at signon to UAF common fields are propagated across VAX machine
- System entry and file violations are logged to CA-ACF2 MVS
- CA-ACF2 command and reports
- Uses ENF for cross memory communication
- Uses CA-NET for VTAM communication

System Authorization Facility (SAF)

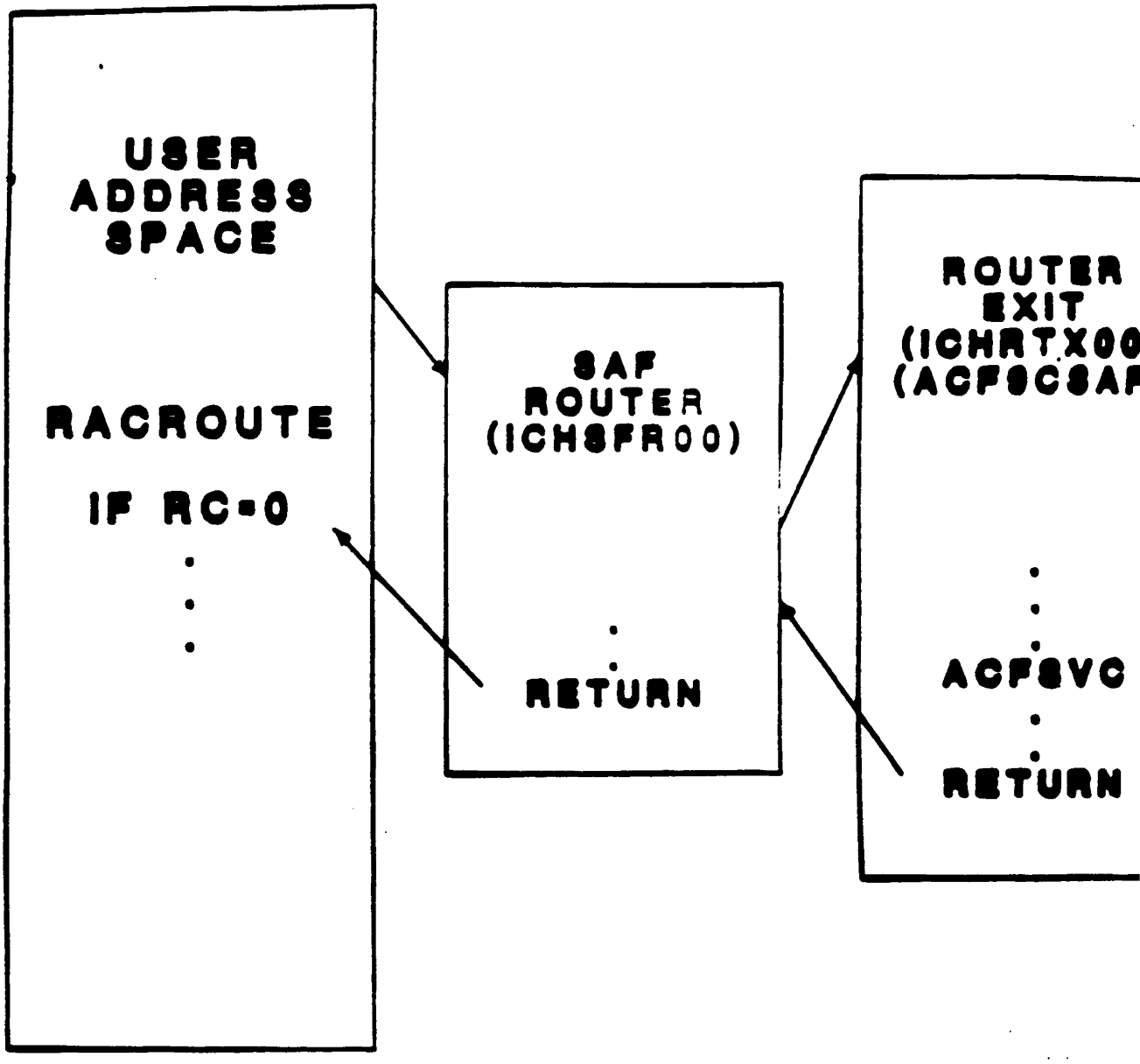
- SAF is a facility that provides a centralized control point for security processing. It is a single interface that can be used across all products.

- The main component of SAF is the MVS router, ICHSFR00. The router is always present in an MVS system.

- SAF is invoked through the use of the RACROUTE macro. The RACROUTE macro calls the MVS router. The RACROUTE parameter list describes the security function needed.

System Authorization Facility (SAF)

RACROUTE Macro Logic Flow Diagram



System Authorization Facility (SAF)

RACROUTE Macro

- IBM supplied macro replaces direct RACF SVC calls
- Builds necessary SAF/RACF parameter lists
- Invokes the SAF router routine (ICHSFR00)

```
RACROUTE,
 REQUEST=(AUTH/FASTAUTH/VERIFY/DEFINE/LIST/ENCRYPT/EXTRACT),
 CLASS=(DATASET/DASDVOL/TAPEVOL/resource),
 ENTITY=,
 USER=,
 PASSWORD=,
 ATTR=(READ/UPDATE/CONTROL/ALTER),
 SUBSYS=,
 REQSTOR=,
 ACEE=
```

- The following are the supported SAF requests:

RACINIT	**VERIFY**	**ACVALD validation call**
RACXTRT	**EXTRACT**	**ACVALD info call**
RACHECK	**AUTH**	**ACDSV or ACGRSRC**
RACDEF	**DEFINE**	**ACVALD**
FRACHECK	**FASTAUTH**	**ACFGINT rule interpret**
RACLIST	**LIST**	**ACGRSRC directory build**

System Authorization Facility (SAF)

CA-ACF2 SAF Facilities

Global CA-ACF2/SAF options in the GSO records:

OPTS SAF	Specify if SAF processing is to be done
OPTS SAFTRACE	Specify if diagnostic tracing is to be done
SAFPROT	Specify which SAF calls are to be validated
SAFSAFE	Specify which SAF calls to ignore
SAFMAPS	Define resource types to be used for SAF

Local CA-ACF2/SAF options in the Logonid record:

NOSAF	SAF validation is not to be done for this user
SAF-TRC	Tracing is to be done for this user

System Authorization Facility (SAF)

SAFTRACE Diagnostics

- GSO OPTS SAFTRACE enables SAF tracing
- Specify SAF-TRC in Logonids to be traced
- All tracing to security console
- Example:

```
ACF9C009        SAF  ENVIRONMENT -  RACHECK PCDPDQ $UCCISPF
ACF9C004        SAF  SUBSYSTEM +    SVC019
ACF9C005        SAF  CONTROL PT +   EXEC
ACF9C006        SAF  CLASS     -    DATASET
ACF9C007        SAF  ENTITY    -    ISP.R1M0.ISRCLIB
```

System Authorization Facility (SAF)

DFSMS SAF Calls

- Access to SMS STORAGE and MGMT classes
- New dataset allocation
  - SAF call extract resowner
    LID=HLQ DSN
    $RESOURCE
    HLQ DSN
  - SAF call extract default classes
    SMSINFO (xxxxxxxx) field in Logonid
    where xxxxxxxx is name of SMS record on infostorage database that specifies data class, storage class, mgmt class
  - SAF call validate access to storage class
    $KEY(storage class) TYPE(STR)
  - SAF call validate access to mgmt class
    $KEY(mgmt class) TYPE(MGM)

System Authorization Facility (SAF)

DFSMS SAF Calls

- Additional SAF calls

  — DFSMS functions and command
  CLASS = FACILITY
  $KEY(IBM defined resource name) TYPE(FAC)

  — ISMF functions
  CLASS = PROGRAM
  $KEY(program) TYPE(PGM)

System Authorization Facility (SAF)

MVS/ESA 3.1.3 SAF Calls

- Hiperspace

  — LSR Access
  Class = Facility
  $Key(CSR.BLSRHIPR.BLSR) type (FAC)
  *When using the default LSR SSN

  — DLF
  Class = DLFCLASS
  $Key (volser dataset name) type (SAF)

- JES 3.1.3

  — Sysout access
  Class = JESSPOOL
  Dataset access rule
  $Key(userid)
  JOBNAME.JOB#.DD#.DSN

  — Cancel authority
  Class = JESJOBS
  Dataset access rules
  $Key(userid)

- Operator commands

Class = OPERCMDs
$Key(command) type(saf)

- Console logon
  Support 5.2