The 'ABC's of z/OS Integrity

Paul R. Robichaux - prr@newera.com
http://www.newera-info.com/

The Mainframe today - Why?

92
of the top 100 worldwide banks

10
out of 10 of the world's largest insurers

23
of the top 25 US retailers

23
out of 25 of the world's largest airlines

Processing the world's transactions & data

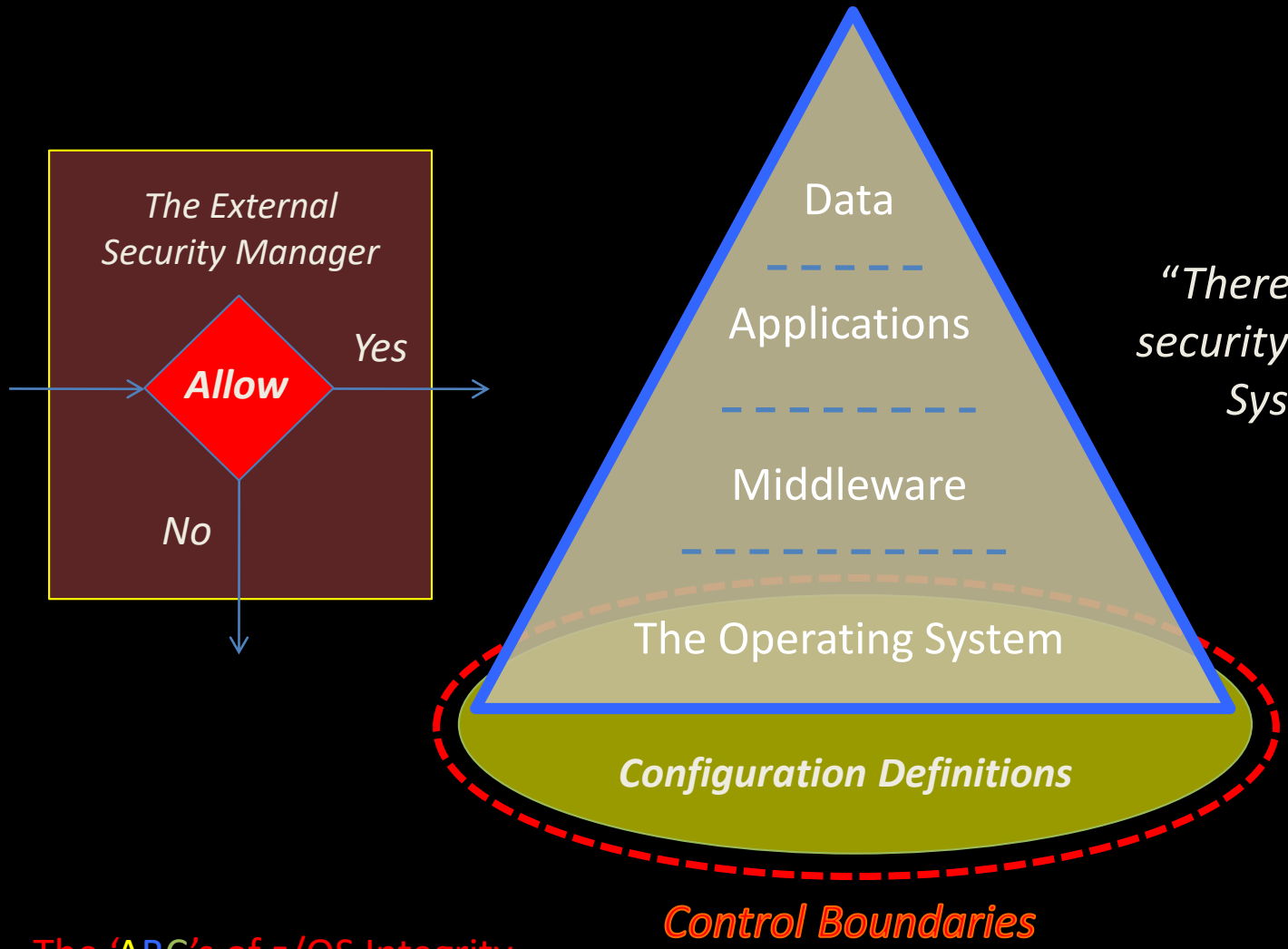
30 billion
business transactions processed on the mainframe per day

80 percent
of the world's corporate data resides or originates on mainframes

91 percent
of surveyed CIOs said that new customer-facing applications are accessing the mainframe

55 percent
of all enterprise applications need the mainframe to complete transactions

The Trusted Computing Base:



"There can be no system security without Operating System Integrity."
Barry Schrager

The IBM z/OS Integrity Statement:

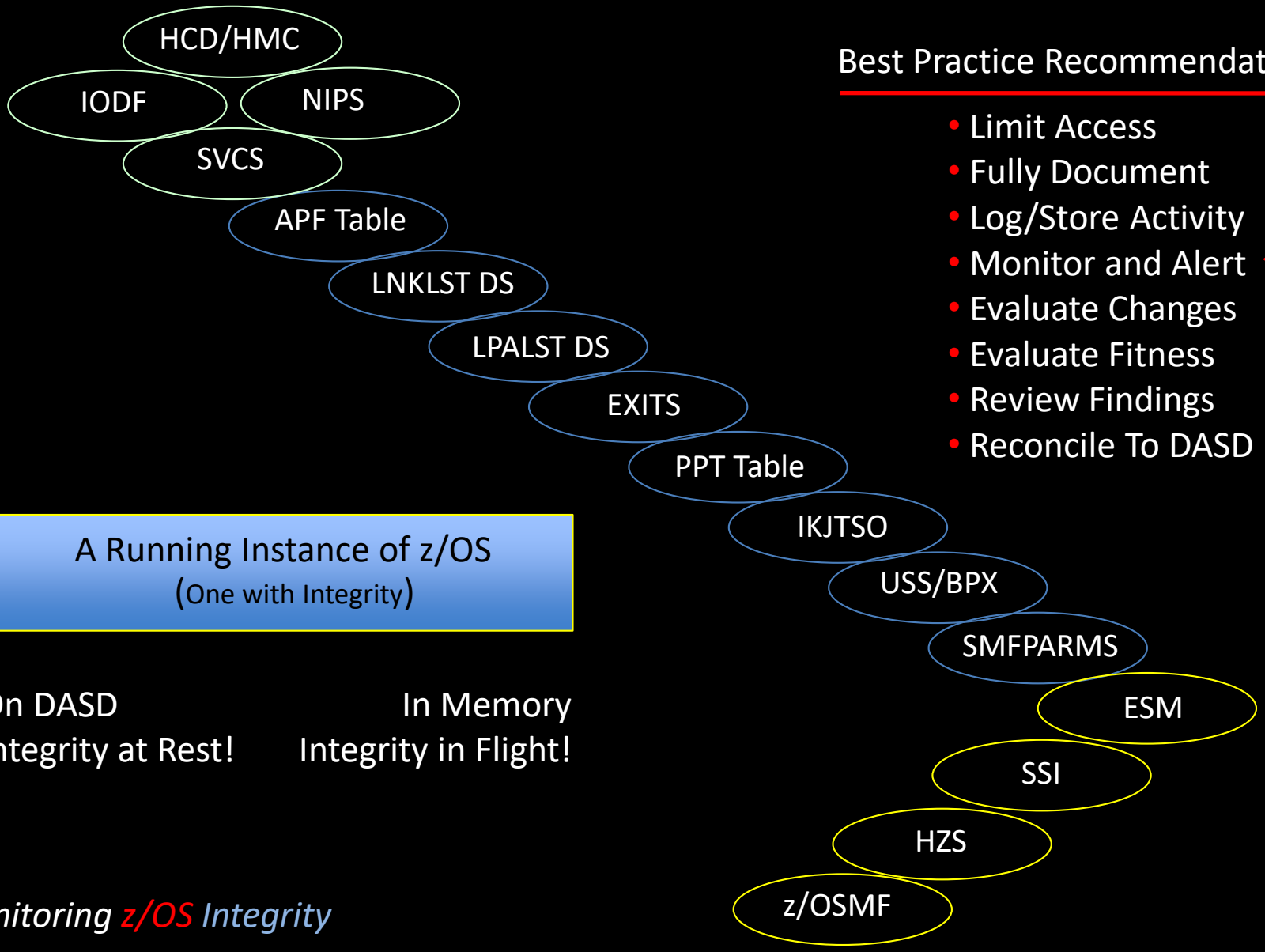
First issued in 1973, IBM's MVSTM System Integrity Statement, and subsequent statements for OS/390® and z/OS, has stood for over three decades as a symbol of IBM's confidence in and commitment to the z/OS operating system. IBM reaffirms its commitment to z/OS System Integrity.

z/OS operates in either of two states: *problem or supervisor state. In problem state a set of non-privileged instructions are available to a program. In supervisor state, programs are additionally able to use privileged instructions which are generally intended for supervisory functions. These functions may affect other users or the entire computer system. A general user is only allowed to access specific supervisory functions after thorough authorization checking by the operating system.*

The Authorized Program Facility (APF) is the primary mechanism provided for this purpose. It allows authorization of system-level programs that need to modify or extend the basic functions of the operating system.

Link

True Integrity Reinforces Business Objectives!



Best Practice Recommendations

- Limit Access
- Fully Document
- Log/Store Activity
- Monitor and Alert ✓
- Evaluate Changes
- Evaluate Fitness
- Review Findings
- Reconcile To DASD

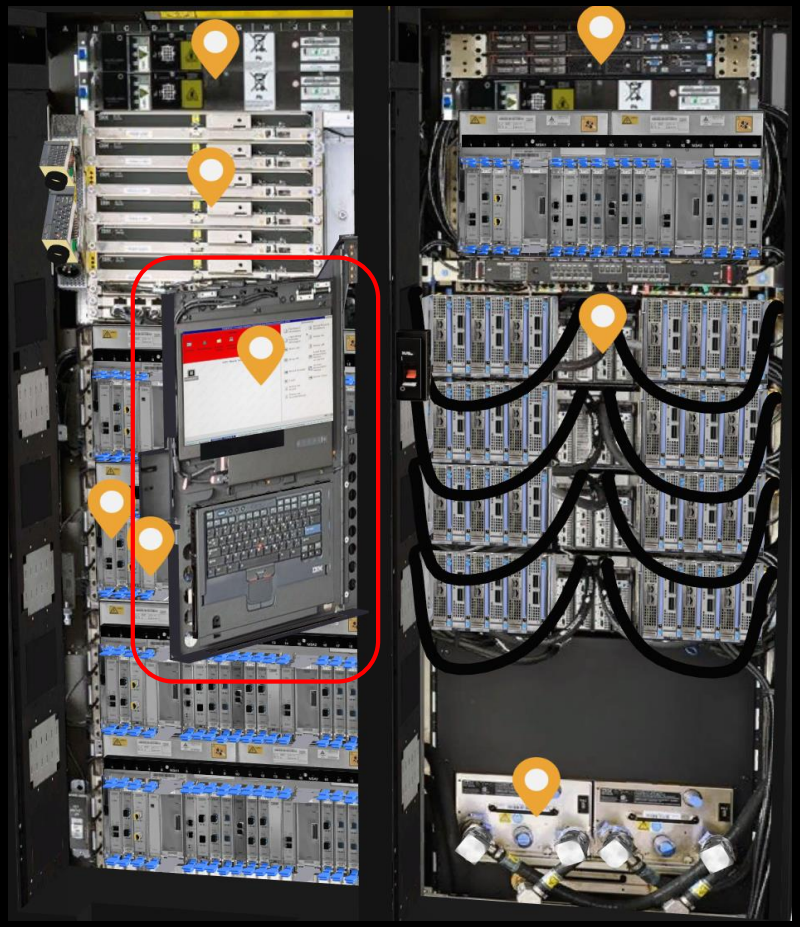
A Running Instance of z/OS
(One with Integrity)

On DASD
Integrity at Rest!

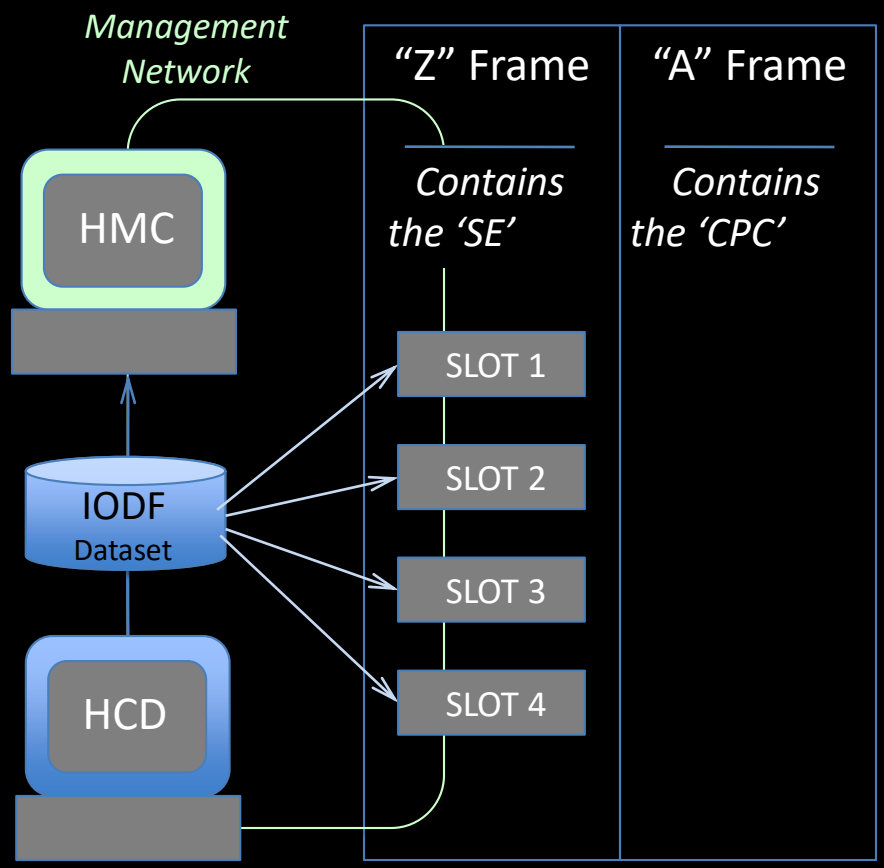
In Memory
Integrity in Flight!

The z/OS IPL Path:

Back-End of z/13 - System Element (SE)



A Single Central Processor Complex (CPC)

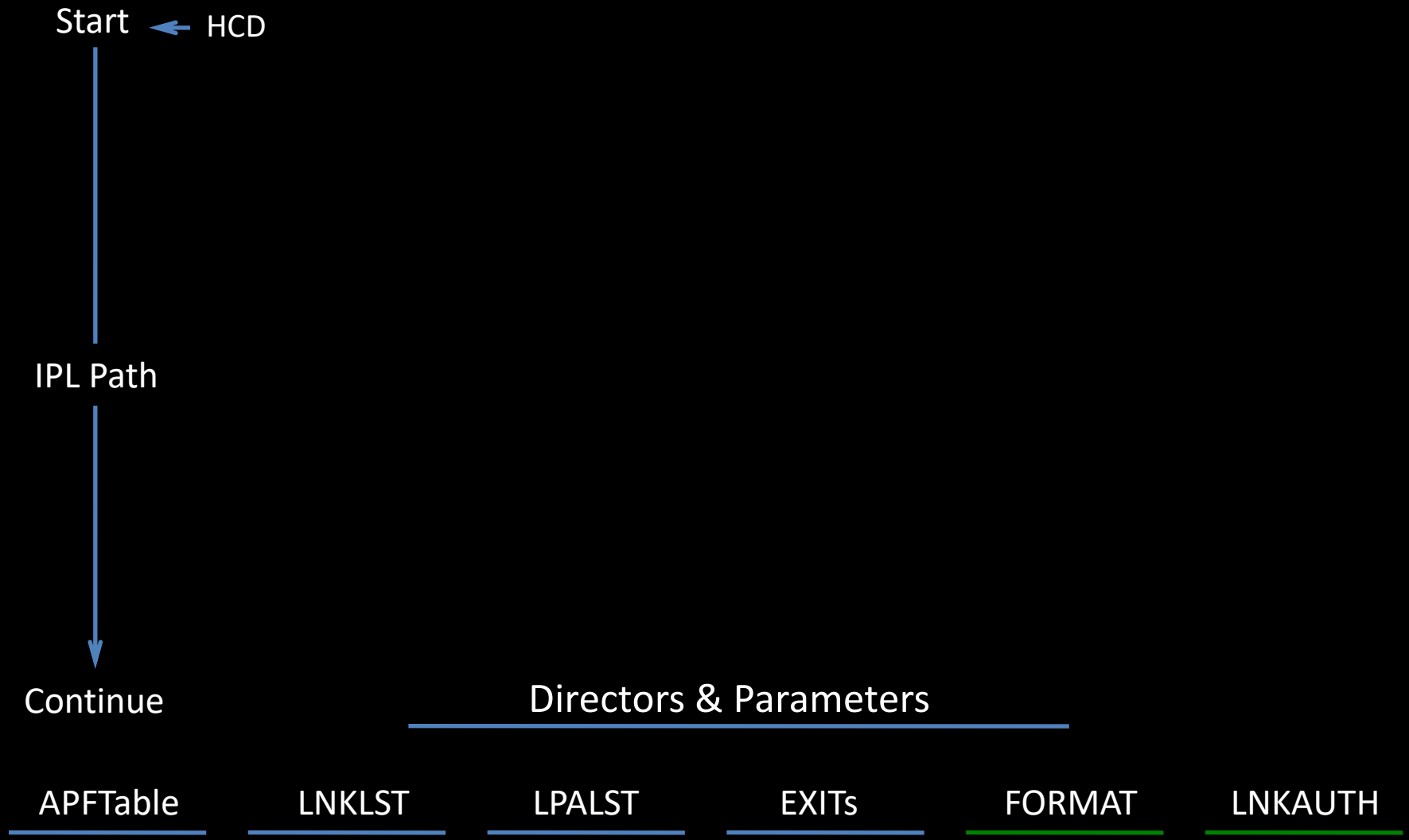


Link

Glossary of Terms:

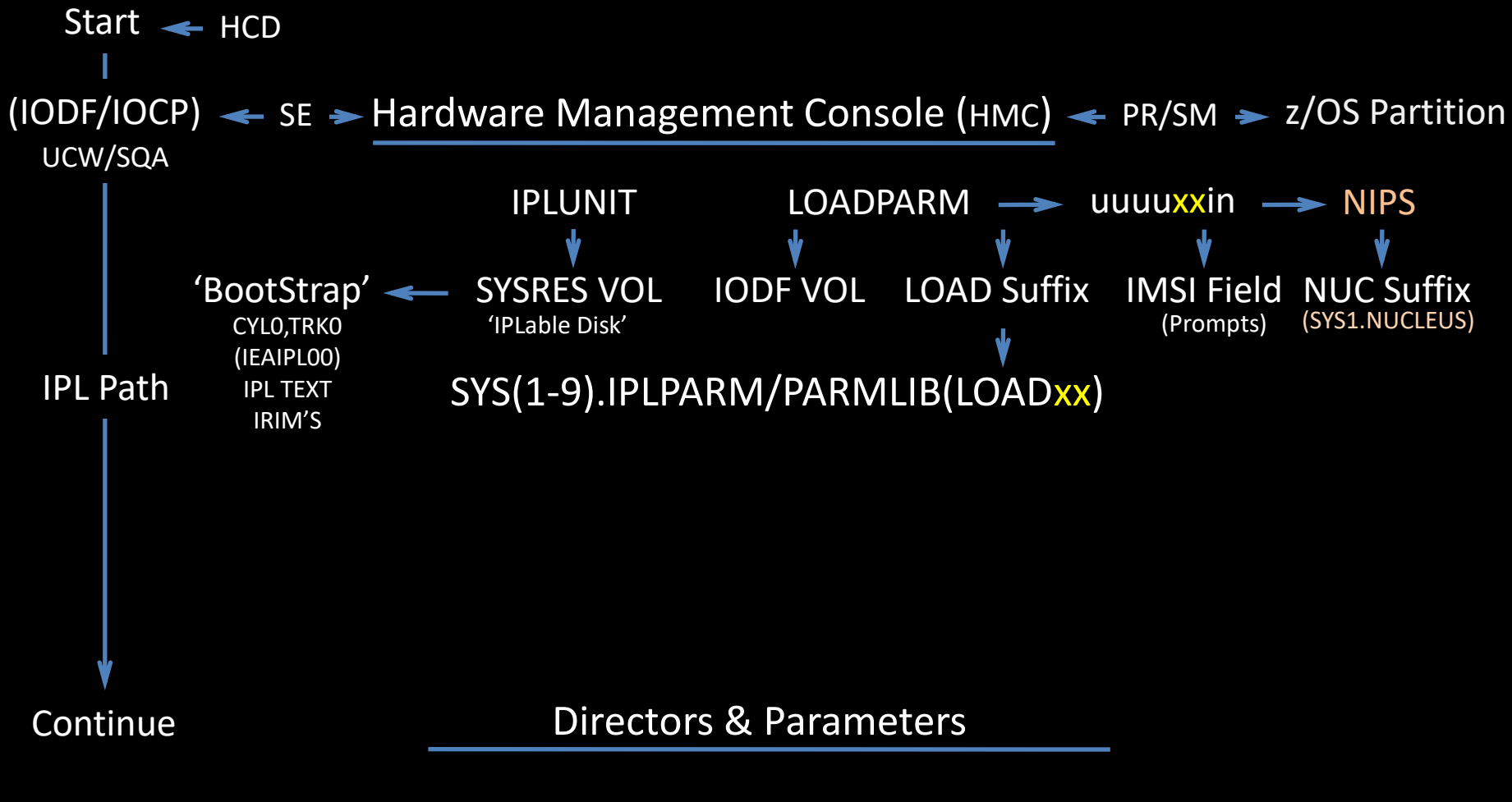
APF - Authorized Program Facility
ASID - The Numeric Address Space Identifier
BCP - The Base Control Program - Backbone of z/OS Reliability and Integrity
CPC - The Central Processing Complex
CLI - Compare Logical Intermediate - In snippet - test for change in State
DUCT - Dispatchable Unit Control Table - Control over the Authority State
EDT - Eligible Device Table
ESM - External Security Manager
HCD - Hardware Configuration Definition
HMC - Hardware Management Console
HSA - Hardware Storage Area
IMSI - Initialization Message Suppression Indicator
IOCP - I/O Configuration Program - Part of IODF
IODF - Input/Output Definition File - HCD - IOCP, OSCP and SWCP
IPK - Insert PSW Key - A privileged Instruction - See snippet
IRIM - IPL Resource Initialization Modules
MODESET - Change system status - alter PSW/PKM or State Indicator
NIPCON - A named Console Device used only during NIPS
NIPS - Nucleus Initialization Processing
OSCP - Operating System Control Program - Part of IODF
PPT - Program Properties Table
PR/SM - Processor Resource/System Manager
PKM - Program Status Word MASK - Control PSW Key Changes
PSW - Program Status Word - 0/7 protected & 8/15 not protected
RIM - Resource Initialization Modules
SE - System Element - 1 of 2 CPC specific Workstations
SPKA - Set Storage Protect Key from Address - A Privileged Instruction
SQA - System Query Area - A storage area in main memory
SVC - Supervisor Call - Named System Modules
SWCP - Switch Configuration Program
UCB - Unit Control Block - Part of Device Chain
UCW - Unit Control Work - Part of Device Chain
USS - Unix System Services

The z/OS IPL Path:

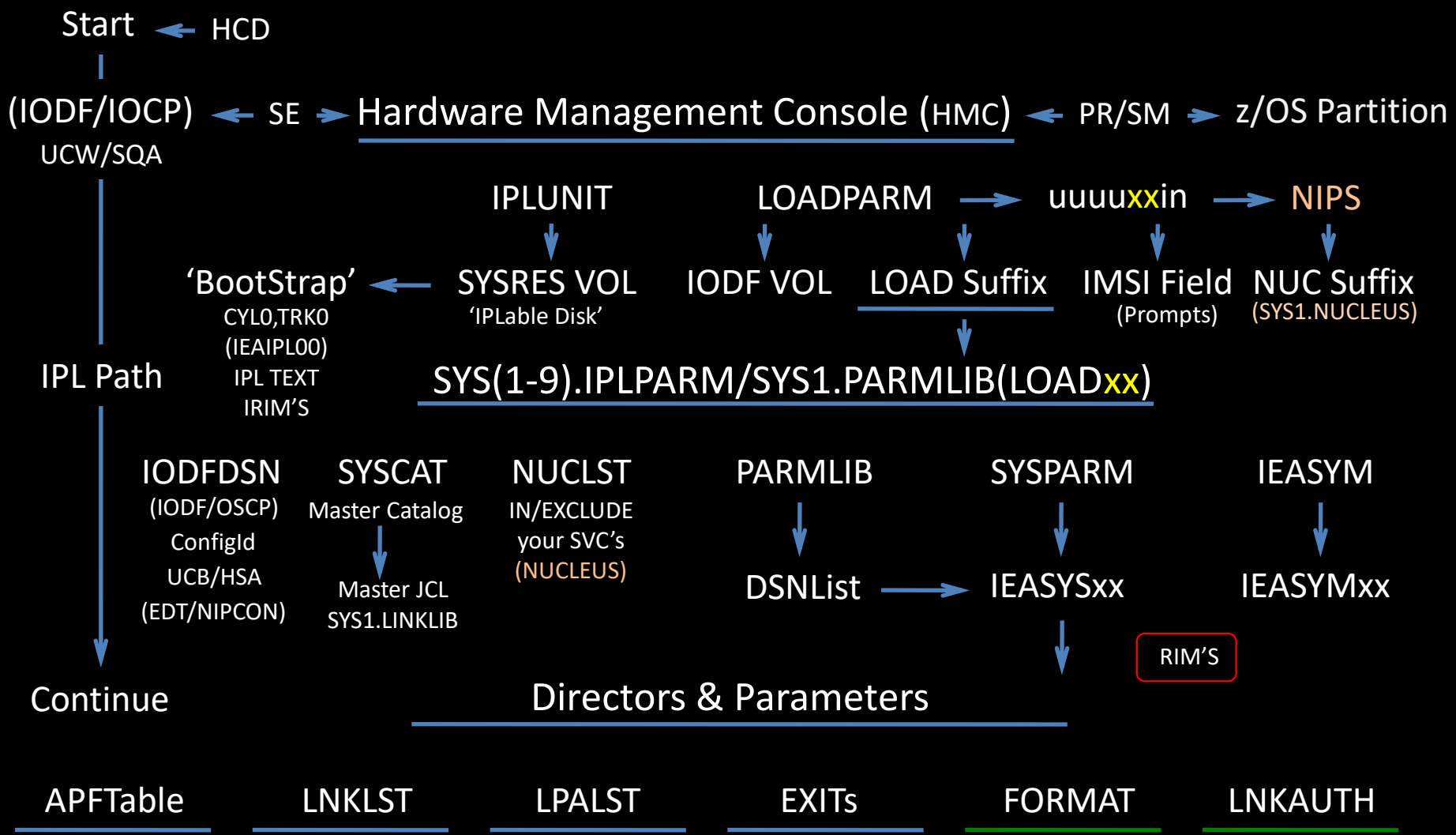


Link

The z/OS IPL Path:



The z/OS IPL Path:



Note: If a member exists more than once within the parmlib concatenation, the first occurrence is used.

The z/OS IPL Path:

IEASYS - Directors … (,L)

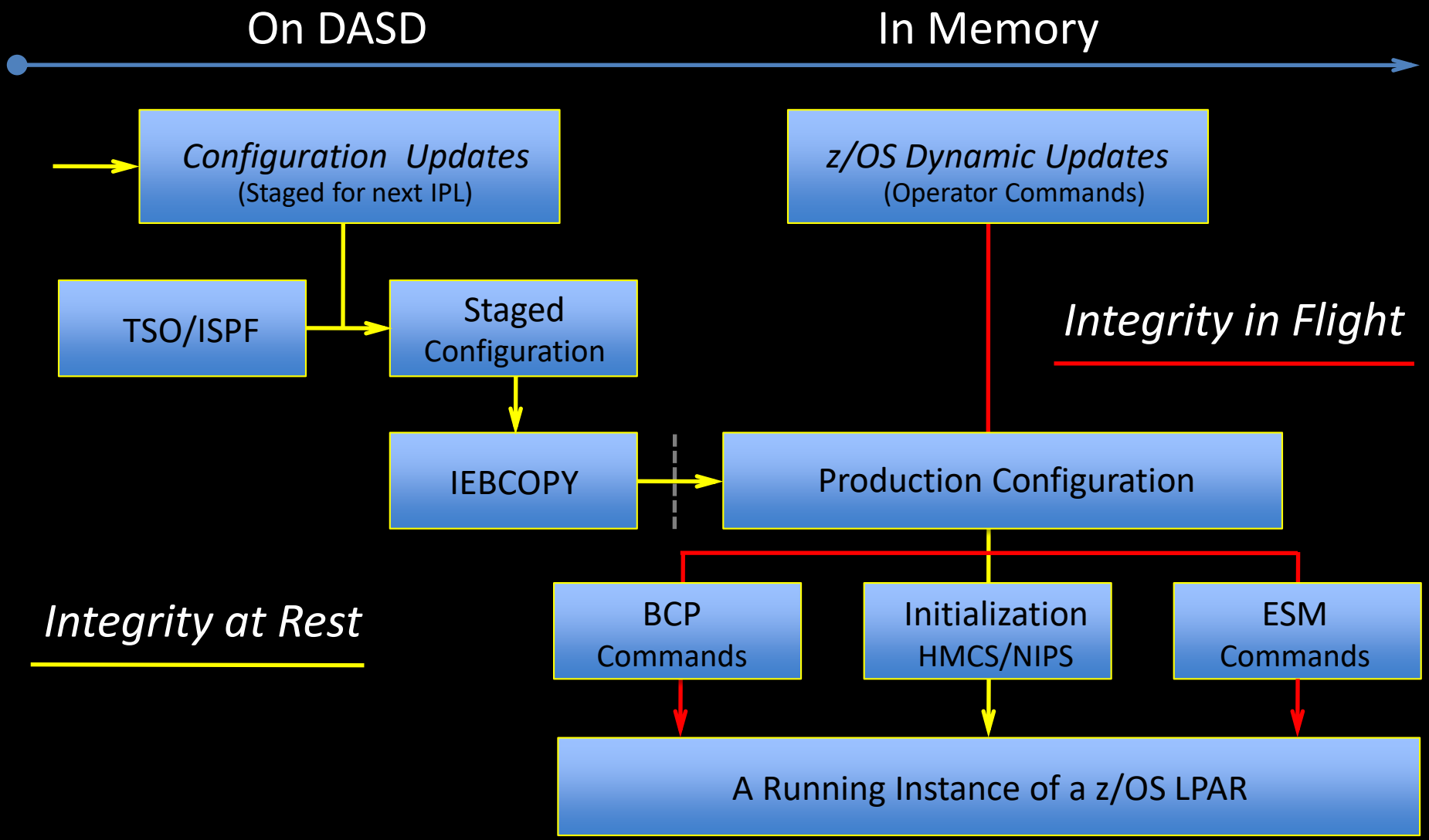
ALLOC=xx,xx	IKJTSO=xx
APF=xx	IOS=xx
AUTOR=xx,xx	IQP=xx,xx
AXR=xx,xx	IXGCNF=xx,xx
CATALOG=xx,xx	LNK=xx,xx
CEA=xx,xx	LPA=xx,xx
CEE=xx,xx	MLPA=xx,xx
CLOCK=xx,xx	MSTRJCL=xx
CMD=xx,xx	OMVS=xx,xx
CON=xx	OPT=xx
COUPLE=xx	PAK=xx
DEVSUP=xx,xx	PROD=xx,xx
DIAG=xx,xx	PROG=xx,xx
EXIT=xx	SCH=xx,xx
FIX=xx,xx	SMF=xx,xx
GRSCNF=xx	SMS=xx,xx
GRSRNL=xx,xx	SSI=xx,xx
GTZ=xx,xx	SVC=xx,xx
HZS=xx,xx	SYSP=OPR,xx,xx
IEFOPZ=xx,xx	UNI=xx & VAL=xx,xx

IEASYS - Parameters

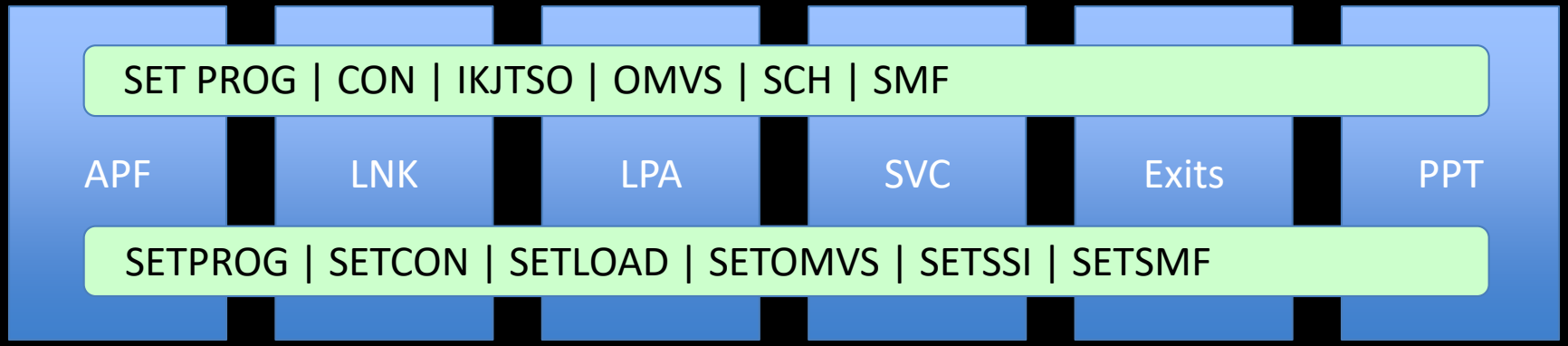
CLPA	NSYSLX
CMB	OPI
CSA	PAGE
CSCBLOC	PAGESCM
CVIO	PAGTOTL
DRMODE	PLEXCFG
DUMP	PRESCPU
GRS	RDE
HVCOMMON	REAL
HVSHARE	RER
HZSPROC	RSU
LFAREA	RSVNONR
LICENSE	RSVSTRT
LNKAUTH	SQA
LOGCLS	SYSNAME
LOGLMT	SYSP
LOGREC	VIODSN
MAXCAD	VRREGN
MAXUSER	WARNUND
NONVIO	ZAAPZIIP

Directors and Parameters that can be placed in an IEASYSxx member or specified by the operator.

System Configuration Management:



APF Authorization - Privileged Instructions:



The system runs in “problem state”. Meaning the set of privileged instructions is not available. Only when the “problem program” is in “supervisor state” can these privileged instructions be used. APF authorization of programs, however granted, permits their use.

APF Datasets are de-fined to the system at a very early stage of the IPL process. As a result the system has no knowledge of their actual existence and loads “as is”. Errors in naming lead to Post-IPL APF vulnerabilities if they are allocated.

LNKLST Datasets are APF Authorized by default. Or not, when the value of the IEASYS Parameter LNKAUTH is set to APFTAB. Only those Modules in either APF/LNKLST marked by the author as AC1 will actually be granted APF authorized access.

If a library is in the LNKLST concatenation but is not APF-authorized, the system will consider the library to be unauthorized for the duration of the job or step if the library is referred to through a JOBLIB or STEPLIB DD statement.

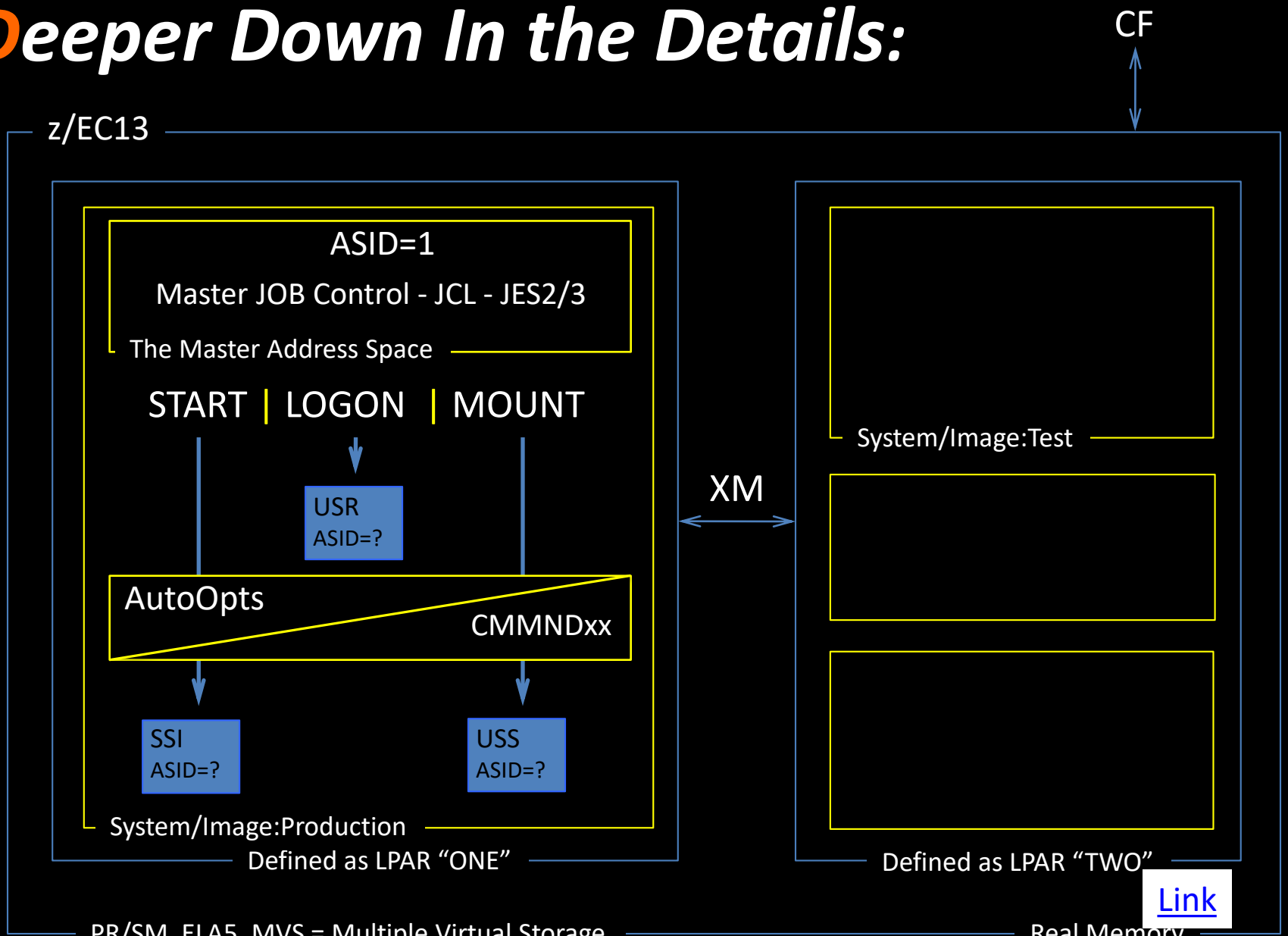
It is not necessary for the datasets in the LPALST to be APF-authorized. However, any module in the link pack area (pageable, modified, fixed, or dynamic LPA) is treated by the system as though it came from an APF authorized library.

PSW keys 0 - 7 are used by the z/OS base control program (BCP) and various subsystems and middleware. Key 0 is the master key. PSW keys 8 through 15 are assigned to users. The Program Properties Table can be used to modify expected PSW key values.

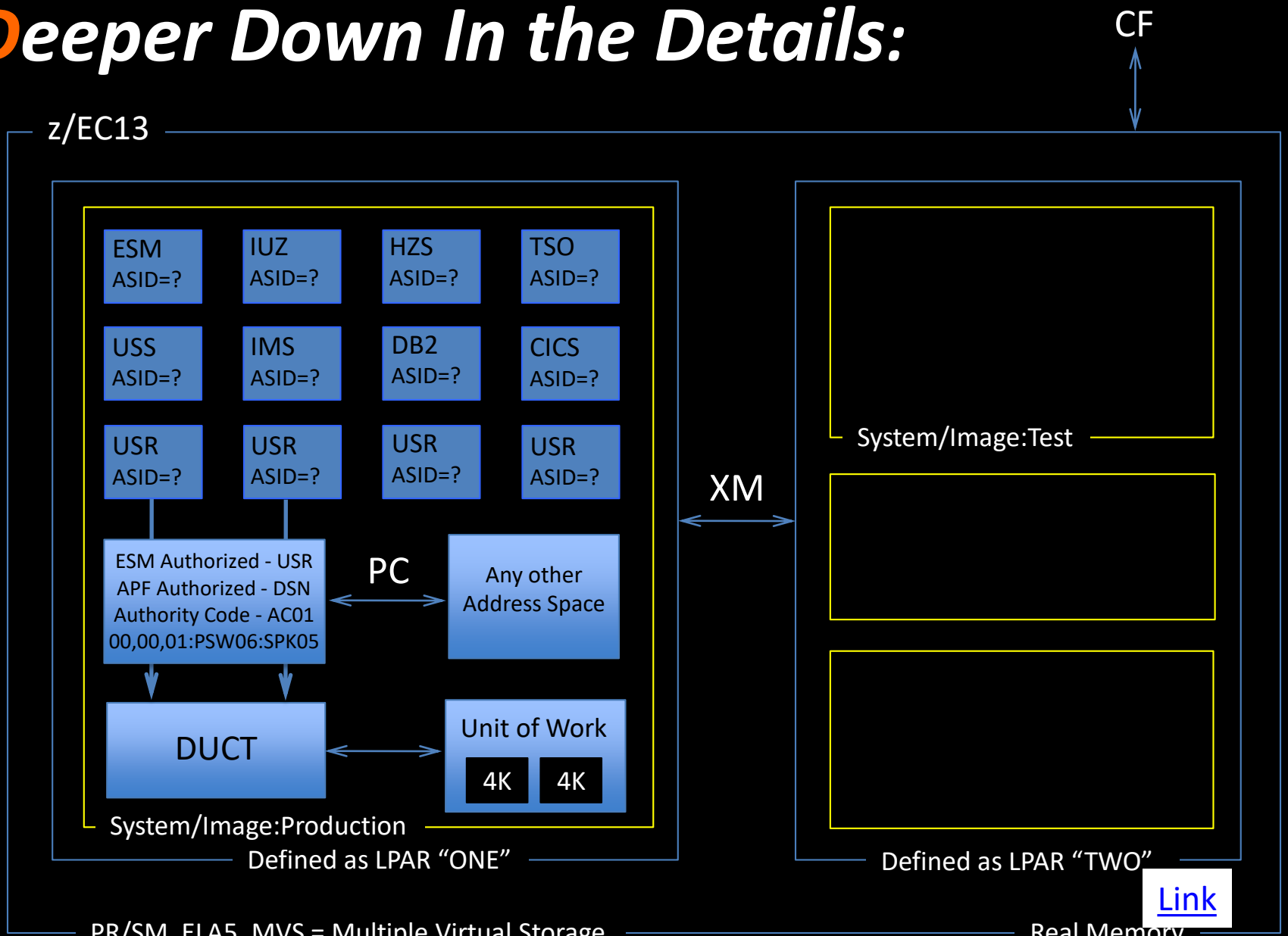
Link

Properly protect LNK and LPA data set to avoid system security and integrity exposures, just as you would any APF-authorized library

Deeper Down In the Details:



Deeper Down In the Details:



Deeper Down In the Details:

ISPF 3.4 - SYS1.LPALIB

Name	Prompt	Alias-of	Size	TTR	AC
BPXQRATT		BPXINLPA	0006FCB8	02A21A	01
CBRHDMAP			00000120	011921	00
CBRHMAT			00000770	011928	00

IEAFIXxX, IEALPAxx, LPALSTxx - PARMLIB

AC = link-edited as being APF eligible using editor SETCODE AC (1) control statement or not (0).

MODSET/PSW/SPKA - A Code Snippet

```
TESTAUTH STATE=YES,RBLEVEL=1   TEST STATE
STC    R15,STATE               SAVE
IT
IA     R2,0
SPKA   0(R2)                   REVERT KEY
CLI    STATE,0                 SUP. STATE
BE     RETURN2                 YES
MODESET MODE=PROB
```

Extract/Save state, MODESET MODE=SUP to issue the IPK instruction and extracts the current key, then back to MODE=PROB, saves the key. Then back to MODE=SUP to restore the caller to original PSW KEY.

In Summary: *To create a program capable of executing privileged instructions and accessing protected storage, that program needs to be linked by the author with AC 01 and placed in an APF authorized library. Next, the program must issue a MODESET macro to placed the program process in supervisor state. Next, using the SPKA instruction (Set PSW Key from Address), the PSW key is changed to 0-7. Finally, if the key associated with program matches a second key, the "Storage Protect Key" the program is granted access/update to the content of the 4K memory target - protected storage.*

Deeper Down In the Details:

A Typical z/OS System LNKLST

```
-- 34 Datasets - Name:LNKLST00 - System:ADCD113 - LNKAuth:LNKLST --
---------Active LNK Datasets--------- APF X Cat Type Volume SMSVol
SYS1.LINKLIB                          APF 1 YES PDS  ZDRES1 ------
SYS1.MIGLIB                           APF 1 YES PDS  ZDRES1 ------
SYS1.CSSLIB                           APF 1 YES PDS  ZDRES1 ------
SYS1.SIEALNKE                         APF 1 YES PLIB ZDRES1 ------
SYS1.SIEAMIGE                         APF 1 YES PLIB ZDRES1 ------
SYS1.SHASLNKE                         APF 1 YES PLIB ZDRES1 ------
SYS1.SERBLINK                         APF 1 YES PDS  ZDRES1 ------
ISF.SISFLOAD                          --- 1 YES PDS  ZDRES2 ------
ISF.SISFLINK                          --- 1 YES PDS  ZDRES2 ------
ISF.SISFMOD1                          --- 1 YES PDS  ZDRES2 ------
```

LNKLSTxx or PROGxx - PARMLIB

An ordered list of data sets processed as the LNKLST concatenation.

z/OS 2.1 - DISPLAY PPT

PgmName	NC	NS	PR	ST	ND	BP	Key	2P	1P	NP	NH	CP
DBNARCHV	.	Y	.	Y	.	.	1	.	.	.	.	.
DFHSIP	.	Y	.	.	.	.	8	.	.	Y	.	.
EPWINIT	Y	Y	.	.	Y	Y	0	.	.	Y	.	.
ERBMFMFC	.	Y	.	Y	Y	.	8	.	.	.	.	.
ERB3GMFC	.	Y	.	Y	Y	.	8	.	.	.	.	.
FNMMAIN	Y	Y	.	.	.	.	6	.	.	.	.	.
HASJES2A	Y	Y	Y	Y	Y	.	1	.	.	.	.	.
ICUMKG10	.	.	.	.	.	.	1	.	.	.	.	.
ICUMKM11	.	.	Y	Y	.	.	1	.	.	.	.	.
IRRSSM00	Y	Y	Y	Y	.	.	2	.	.	.	.	.

SCHEDxx - PARMLIB

Allows the installation to specify a list of programs that require special attributes.

Deeper Down In the Details:

SVC routines (NUC/LNK/LPA) receive control with PSW key zero and in supervisor state

SVC Table -IBM 000-199 / USER 200-255

SVCNum	Location	-Values-	Lib	-Module-	Typ	-Authorization-	ASC	Locks
IBM000	80FE6070	00008000	NUC	--IGC000	1	----------	---	LOCAL
IBM001	80FF4CD2	00008000	NUC	--IGC001	1	----------	---	LOCAL
IBM002	810C4080	00008000	NUC	--IGC002	1	----------	---	LOCAL
IBM003	81405730	00808000	NUC	--IGC003	1	----------	Yes	LOCAL
IBM004	814957CA	00008000	NUC	--IGC004	1	----------	---	LOCAL
IBM005	814957CA	00008000	NUC	--IGC005	1	----------	---	LOCAL
IBM006	81390F48	80008000	NUC	--IGC006	2	----------	---	LOCAL
IBM007	81398010	80008000	NUC	--IGC007	2	----------	---	LOCAL
IBM008	81391EF0	80008000	NUC	--IGC008	2	----------	---	LOCAL
IBM009	81387F58	80008000	NUC	--IGC009	2	----------	---	LOCAL
IBM010	814967C8	00008000	NUC	--IGC010	1	----------	---	LOCAL
IBM011	836E0C70	C0000000	LPA	IGC00011	3/4	----------	---	-----
IBM012	813955E0	80808000	NUC	--IGC012	2	----------	Yes	LOCAL
IBM013	83BF1000	C1808000	LPA	IGC00013	3/4	May be Assisted	Yes	LOCAL

EXIT - A Sample List on Named EXITS

0002	CSVDYNEX	E	0003	HZSADDCHECK	E
0005	IEASDUMP.GLOBAL	E	0006	IEASDUMP.LOCAL	E
0008	IEASDUMP.POSTDMP	E	0009	IXC_ELEM_RESTART	E
0011	ISGNQXIT	E	0012	ISGNQXITFAST	E
0014	ISGCNFXITSYSPLEX	E	0015	ISGNQXITBATCH	E
0017	ISGNQXITQUEUED2	E	0018	ISGENDOFLQCB	E
0020	ISGNQXITBATCHCND	E	0021	ISGDGRSRES	E
0023	IGGPRE00_EXIT	E	0024	IGGPOST0_EXIT	E
0026	REKEY_EXIT	E	0027	IEF_ALLC_OFFLN	E
0029	IEF_VOLUME_ENQ	E	0030	IEF_VOLUME_MNT	E
0032	IEF_ALLC_MOD	E	0033	IEF_ALLC_EVENT	E
0035	CEE_ABEND_EXIT	E	0036	CNZ_WTOMDBEXIT	E
0038	SYSIEASLIPAEXIT	E	0039	SYSSTC.IEFUSO	E
0041	SYSSTC.IEFU84	E	0042	SYSSTC.IEFU83	E

IEASVCxx- PARMLIB

Do not attempt to modify SVCs that are in the range of 0-199. Doing so will cause unpredictable results.

PROGxx/EXITxx - PARMLIB

Ensure EXITxx resides in SYS1.PARMLIB, because it can only access it from SYS1.PARMLIB.

Deeper Down In the Details:

IKJTSO - Authorized Programs/Commands	SMFPRM - System Management Facility
AUTHCMD NAMES(*cmd1,cmd2...)* AUTHPGM NAMES(*pgm1,pgm2...)* AUTHTSF NAMES(*name1,name2...)* *The TSO Service Facility provides a mechanism to invoke authorized commands, programs, or CLISTs (consisting of only authorized commands or programs) from unauthorized application programs. Usually, authorized commands, programs, or CLISTs can be invoked only from authorized environments.* *Eligible commands and programs are those having an entry in SYS1.PARMLIB, member IKJTSOxx.*	*SMF forms the basis for monitoring and automation utilities. Each SMF record has a number - its record type. Records written by software other than IBM products generally have a record type of 128 or higher. SMFPRM controls how much or how little SMF data is collected for each System - by its SMFID or SID.* *RACF type 80 records are written to record security issues, i.e. password violations, denied resource access attempts, etc. Other security systems such as ACF2 also use the type 80 and 81 SMF records.* *If NOPROMPT, an SMFPRMxx parameter, is set during an IPL, the operator command SETSMF cannot be used for the duration of that IPL.*
IKJTSOxx- PARMLIB	SMFPRMxx- PARMLIB

Link

Deeper Down In the Details:

Typical Sub-System Initialization - Following z/OS Initialization

Sample IEASYSxx member - Keyword SSN={aa }|{(aa,bb,...)}

The SSN parameter in IEASYSxx identifies the IEFSSNxx member that the system is to use to initialize the subsystems.

Sample IEFSSNxx member

SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN) INITPARM(ID=ZX)
SUBSYS SUBNAME(JES2) /* JES2 AS PRIMARY SUBSYSTEM */
PRIMARY(YES) START(YES)
BEGINPARALLEL
SUBSYS SUBNAME(RACF) INITRTN(IRRSSI00) INITPARM('#,M')
SUBSYS SUBNAME(IRLM)

RACF Specific IEFSSNxx Syntax Decoded

SUBSYS SUBNAME(ssname) INITRTN(routine) INITPARM(cmdpref[,scope])
where:
ssname is the 1-4 character subsystem name (required) - Very likely 'RACF'
routine is the initialization routine (required) - IRRSSI00 is the RACF routine
cmdpref is the 1-8 character command prefix (optional) - # is RACF default
scope is the command prefix scope for CPF (optional)
where: X for sysplex scope and/or M for system scope

The SETSSI MVS Operator Command - MVS.SETSSI.ACTIVATE.subname

SETSSI {ADD,{INITRTN|I}=initrtn[,{INITPARM|P}=initparm]] }

Deeper Down In the Details:

By Example - How TCE/OPER Balances Control with Productivity

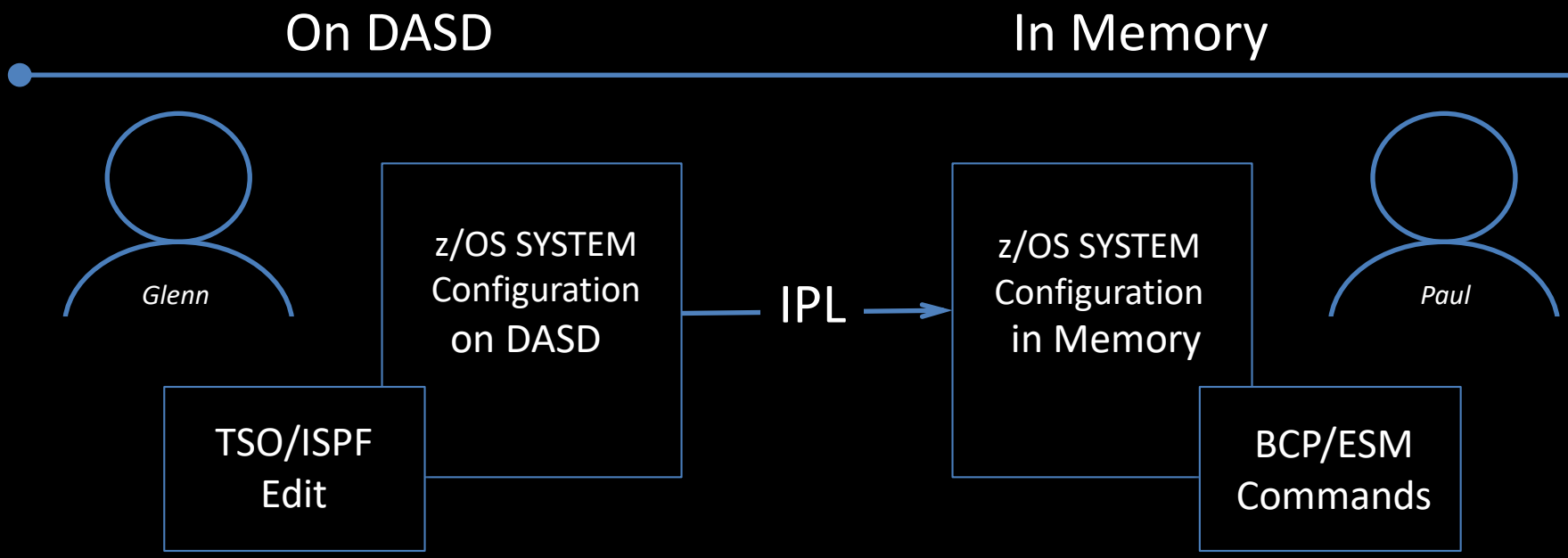
"…making a data set APF-authorized is not sufficient to bestow APF-authorization upon a program that is the target of EXEC PGM=. That requires AC=1. And that could be checked before adding such a data set to the LNKLST. That is a reason why, naturally, it is very important not to have modules mismarked as AC=1.

Putting such a data set into the LNKLST with LNKAUTH=LNKLST does, however, mean that if an authorized program asks to fetch such a module (perhaps to LINK to it), that fetch will be granted. That is a danger of marking any data set as APF-authorized that should not be.

FWIW, if you just want to see if your APF list completely has all of the LNKLST libraries, you could capture the output of DISPLAY PROG,LNKLST and DISPLAY PROG,APF then sort and compare. That will at least give you an idea (although the APF entries may show volume, and the LNKLST entries could have a data set alias whereas the APF entry is supposed to be the 'real' data set name)."

ibm-main@LISTSERV.UA.EDU - Peter Relson, z/OS Core Technology Design Team - relson@us.ibm.com

Lost Integrity Undermine System Controls!



"Integrity at Rest"

- Limit Access
- Take a Backup!
- Set Restore Points
- Fully Document
- Log/Store Activity
- Monitor and Alert ✓
- Evaluate Fitness
- Review Findings

Vs.

"Integrity in Flight"

- Limit Access
- Fully Document
- Log/Store Activity
- Monitor and Alert ✓
- Evaluate Changes
- Evaluate Fitness
- Review Findings
- Reconcile To DASD

E-Book Downloads at www.newera-info.com:

Thank You!

The 'ABC's of z/OS Integrity

Paul R. Robichaux - prr@newera.com
http://www.newera-info.com/