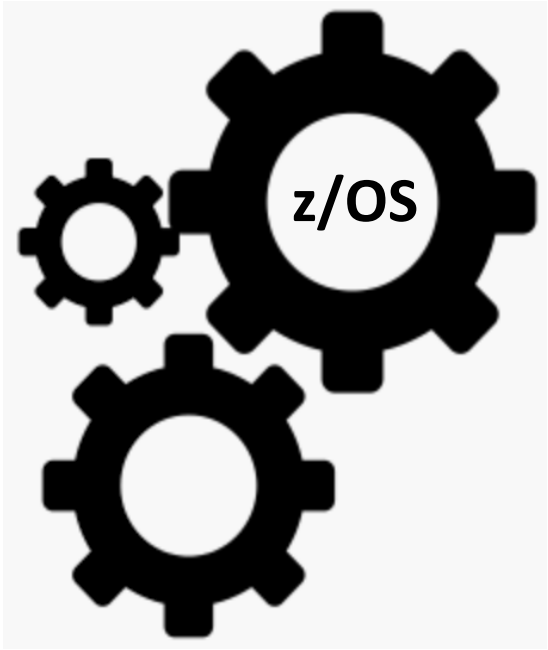


Let's Build a z Environment - 101

Session 23330

Tuesday, August 14 at 10:00-11:00 AM

STL CC, Room 242

Presented by Paul R. Robichaux

NewEra Software, Inc.

prr@newera.com

Abstract – Let's Build a z Environment!

The two presentations in this series focus on the building of a z Environment – Hardware, Software, Security – with the goal of establishing a 'Trusted Computing Base'. A z/OS System that can provide the reliability needed to meet demanding service levels, integrity and security objectives. All are necessary to execute mission critical applications. This is Intended for those new to z Systems or just beginning their careers with organizations that capitalize on systems anchored to the power and reliability of the IBM Mainframe.

In – 101 – the focus will be on the platform, in this case a z14, hardware divisions of the Central Processing Complex (CEC), its various channel pathways and related devices that define a UCW (Unit Control Work), the front half of the z System Device Chain. This segment continues with the definition of an associated Operating System configuration, its various I/O devices and related features that define a UCB (Unit Control Block), the back half of the z System Device. Detailing both the Power-On and IPL process will join UCWs and UCBs to form a fully addressable device across which data (encrypted or not) may flow to and from the CEC.

In – 102 – the focus will shift to a discussion of Multiple Virtual Storage (MVS), what is z/OS, how to get it, install it, support it and upgrade/migrate from release to release. The elements of the IPL Path – IPLPARM, IRIMS, IODF, SYSRES – to name just a few will be examined in detail as will the Post-IPL environment – APFLST, LNKLST, LPALST, SVCs, EXITs, PPT. The integrity of the environment will be described within the context of the IBM Integrity Statement and the Authorized Program Facility (APF). The session ends with a discussion concerning system vulnerabilities, their potential impact and sources of possible remediation.

Paul R. Robichaux is CEO and co-founder of NewEra Software, Inc. He served as the Chief Financial Officer of Boole and Babbage for the ten years immediately preceding his co-founding of NewEra in 1990. He holds a BS in Accounting and a Masters in Business Administration from a Louisiana State University, is a Certified Public Accountant and a frequent speaker at industry events.

The corporate mission of NewEra Software is to provide software solutions that help users avoid z/OS non-compliance, make corrections when needed and in doing so, continuously improve z/OS integrity and Security. http://www.newera.com

Let's Build a System z Environment - 101

Sysplex with two z14s and a z14 (CF). A total of 30 LPARs - An average size z/OS shop.

Null Z14 Hardware → Defining the z/14 Hardware

Load the z/OS Program → A Running z/OS LPAR

When the System is at Rest - 101

1. Why are we here?
2. Alphabet, Words, Pictures & Rules
3. The Mainframe Today
4. Virtualization
5. Physicalization
6. Crafting a Partition
7. Goals and Control Points
8. Unpacking the Configuration
9. Power-on-Reset and/or IPL
A. Defining a z/OS Image

When the System is in Flight – 102

1. What is MVS?
2. Delivery
3. Migration
4. On the IPL Path
5. Post-IPL State
6. z System Integrity
7. Program Authorization
8. System Vulnerabilities
9. Language Index
A. Recommended Reading/Viewing

1 - Why are we here?

"The world is in the midst of a transformation that is having a profound effect on us as individuals, in business, and in society at large. As we adapt to capitalize on these trends, we must come to understand that trust will be the valued currency that will drive our economies."

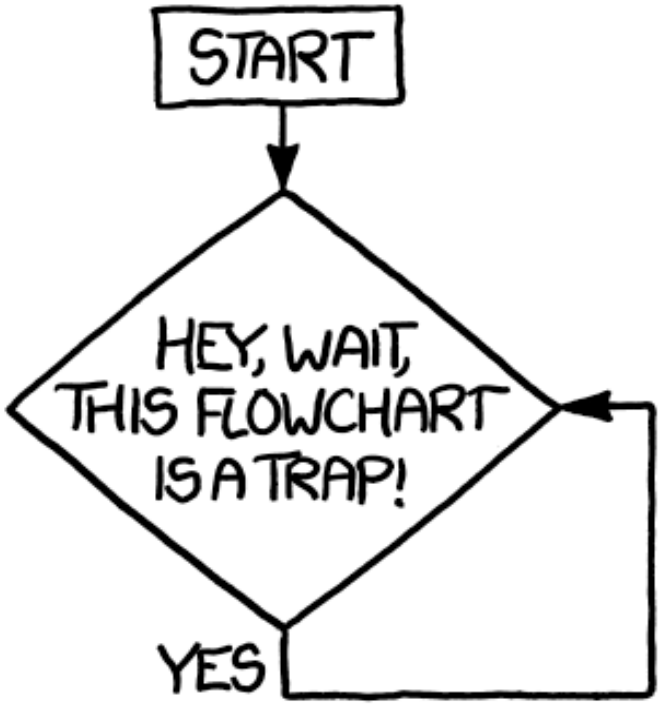
The Mainframe Language

```
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
DEB     - Data Extent Block build on OPEN of DCB (Data Control Block).
DUCT    - Dispatchable Unit Control Table - Control over the Authority State
EDT     - Eligible Device Table
ESM     - External Security Manager
```

SHARE
EDUCATE • NETWORK • INFLUENCE

Mainframe Words and Pictures

- Sometimes when mainframe words are used to describe an element, it is best to reduce your understanding of the commentary to a picture/flowchart.
- Sometimes when a picture/flowchart is presented as a mainframe process, it is best to reduce your understanding to mainframe words.
- It's always best to reconcile the two as one or the other might be a trap. Likely not both!



Rules to Live By

System Integrity and System Security are equally important.

Systems at Rest and those in Flight require different management controls.

When possible, employ separation of duties. Make responsibilities crystal clear.

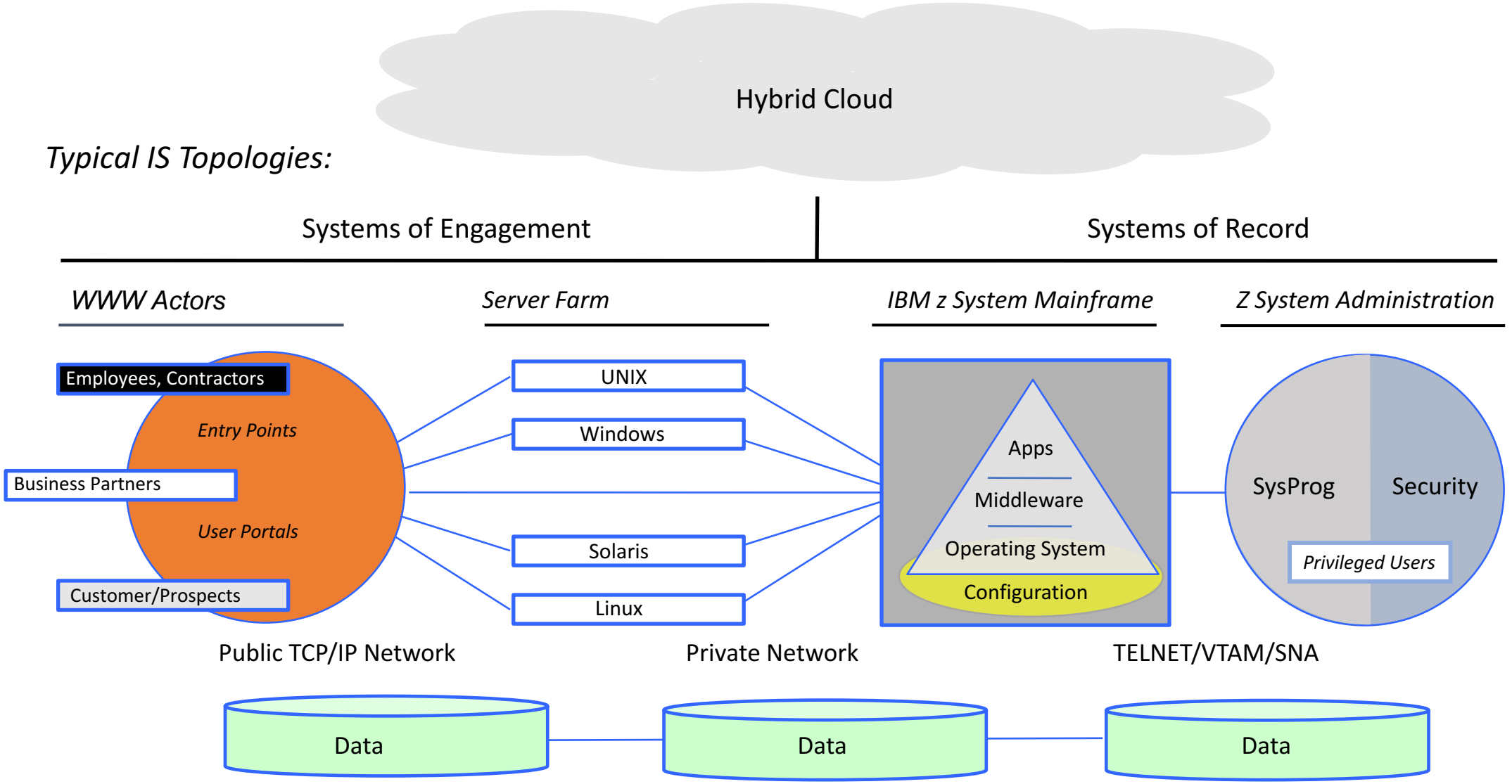
Always make a backup before you make a change.

Don’t misplace your master keys.

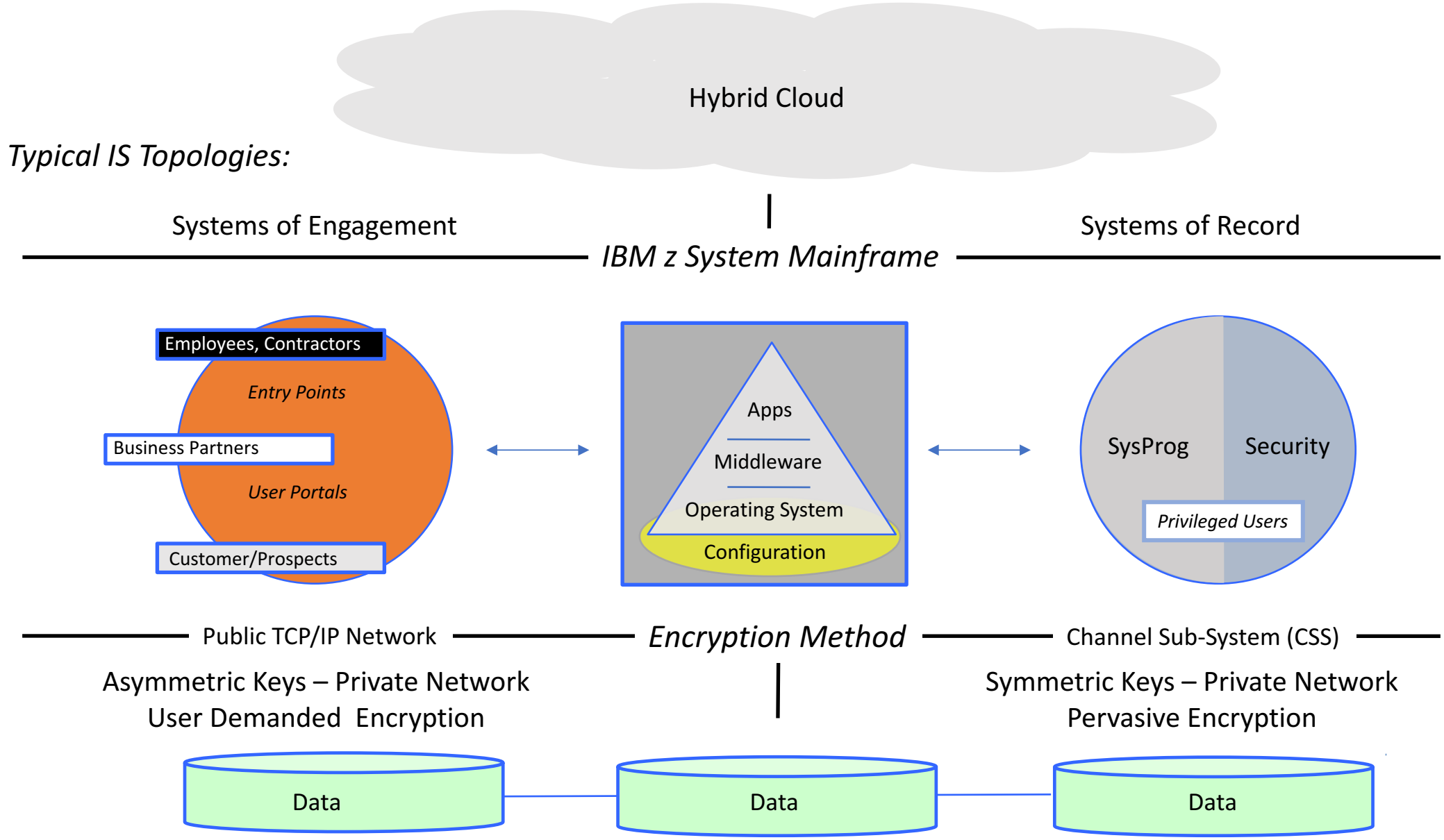
Compatibility above all

“It’s the Pipe”!

3 - The Mainframe - Today.

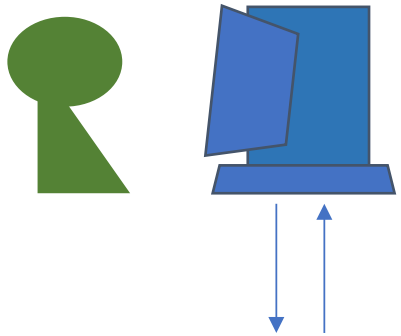


3 - The Mainframe - Tomorrow.

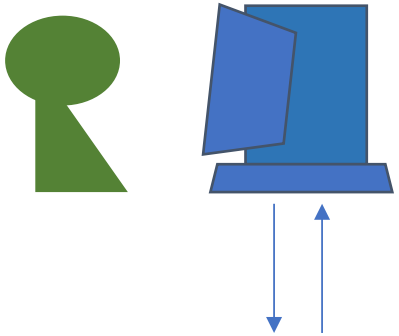


3 - The Mainframe Today.

Today, to the end-user it all appears to be the same. Really it's not! It's Real-Time Vs. Batch.



Client/Server
Systems of Engagement



Mainframe
Systems of Record

Real-Time Processing:

Transaction oriented, involving a continuous process of input and output of data. Hence, it processes in a short period of time. Examples of programs which use Real-Time processing, Bank ATMs, customer services, and Point of Sale (POS) Systems.

Every transaction is directly reflected in the master file as it occurs.

Batching Processing:

Focused on processing of a large volume of data all at once, i.e. in a batch. It is an extremely efficient way to handle data that is collected over a period of time. It reduces operational costs and JOBs can be scheduled at off-hours.

System Master files are likely outdated upon the completion of a "Batch-Run" as current transactions are not reflected after a data cutoff.

3 - The Mainframe Today.

zHyperLink Express

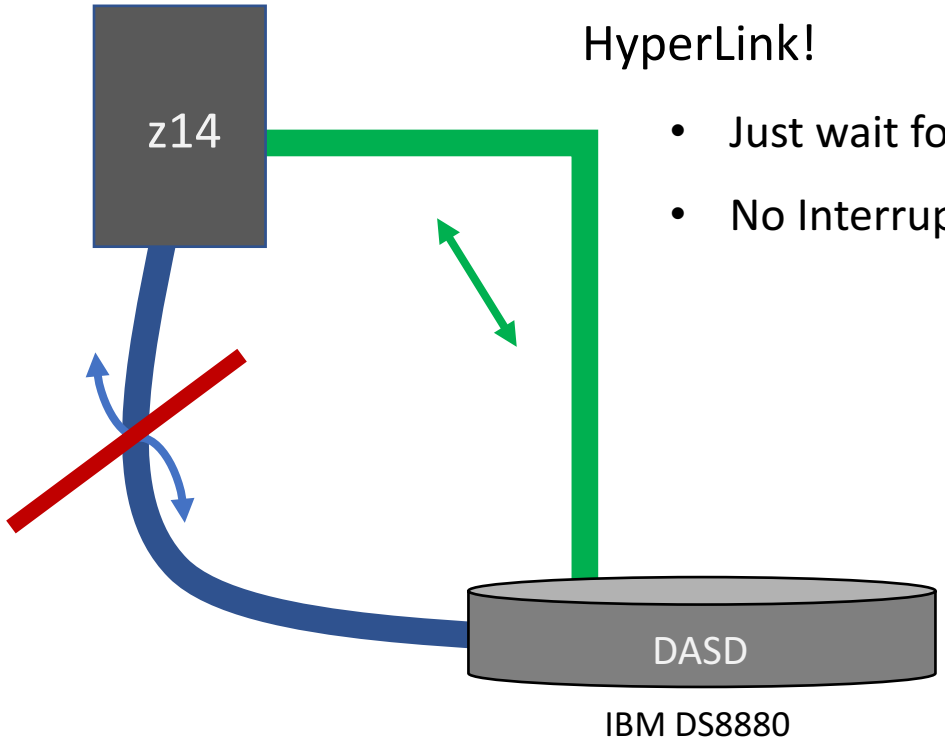
Reduces I/O latency between a z14 Mainframe and data storage (DASD) by interconnecting the z14 directly to DASD, specifically the IBM DS8880.

How?

- No need to suspend the running task
- Eliminates the need to wait for a reply
- Super Small I/O Service Time

Pipe Interrupted!

- Task Suspended
- Waiting for reply



IBM DS8880

HyperLink!

- Just wait for the reply
- No Interrupts

DASD – Direct Access Storage Device

3 - The Mainframe Today.

Hybrid Cloud

Typical IS Topologies:

Systems of Engagement — *IBM z System Mainframe* — Systems of Record

IBM LinuxONE/zR1/z14/z13
Hypervisor is KVM/zHPM

Vs.

IBM /zR1/z14/z13/z12
Hypervisor is PR/SM-MVS

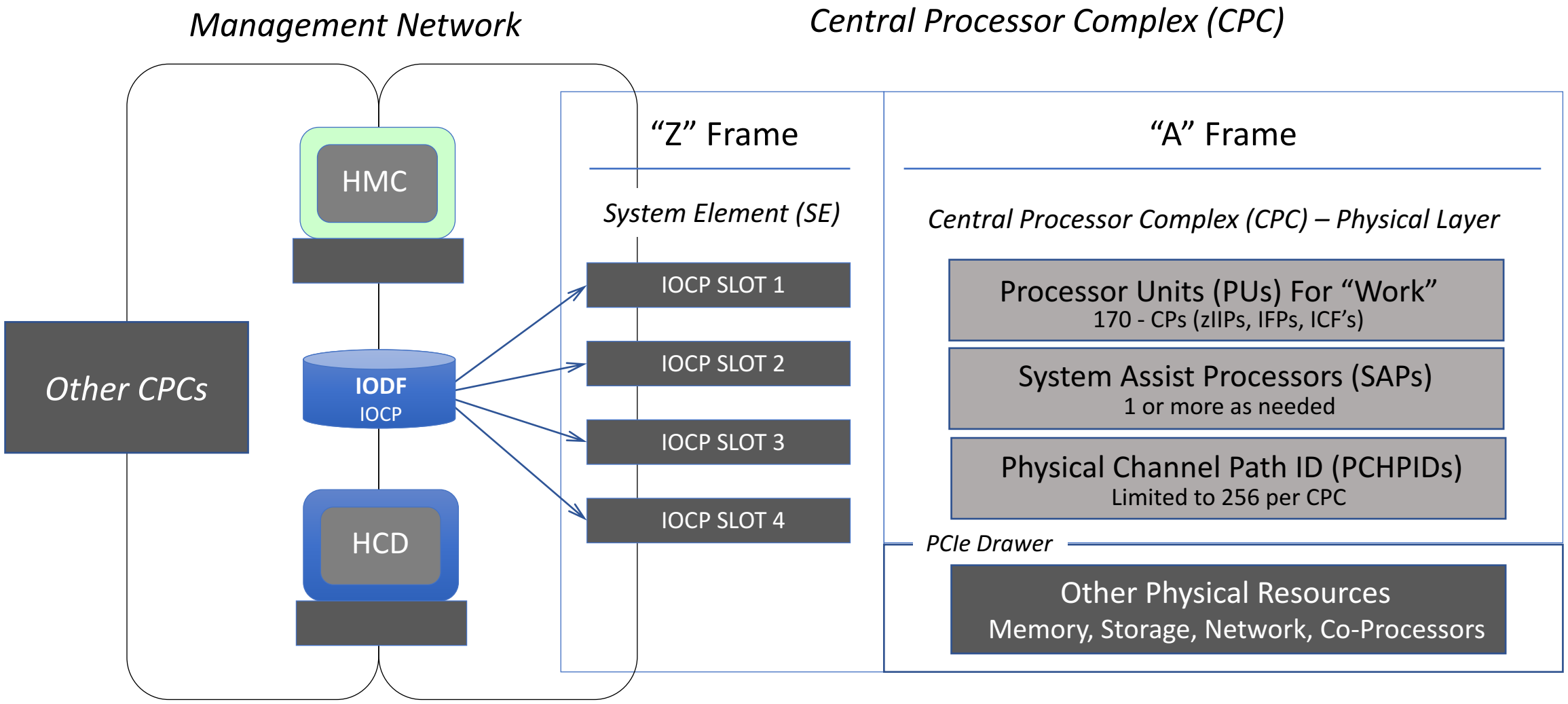
The 'Big Deal' here (DPM) is a change in the way we define the Virtualization of zResources

Dynamic Partion Management
Sets DPM Mode – Can't run z/OS

Vs.

Processor Resource/System Manager
Sets PR/SM Mode – Can't run KVM/zHPM

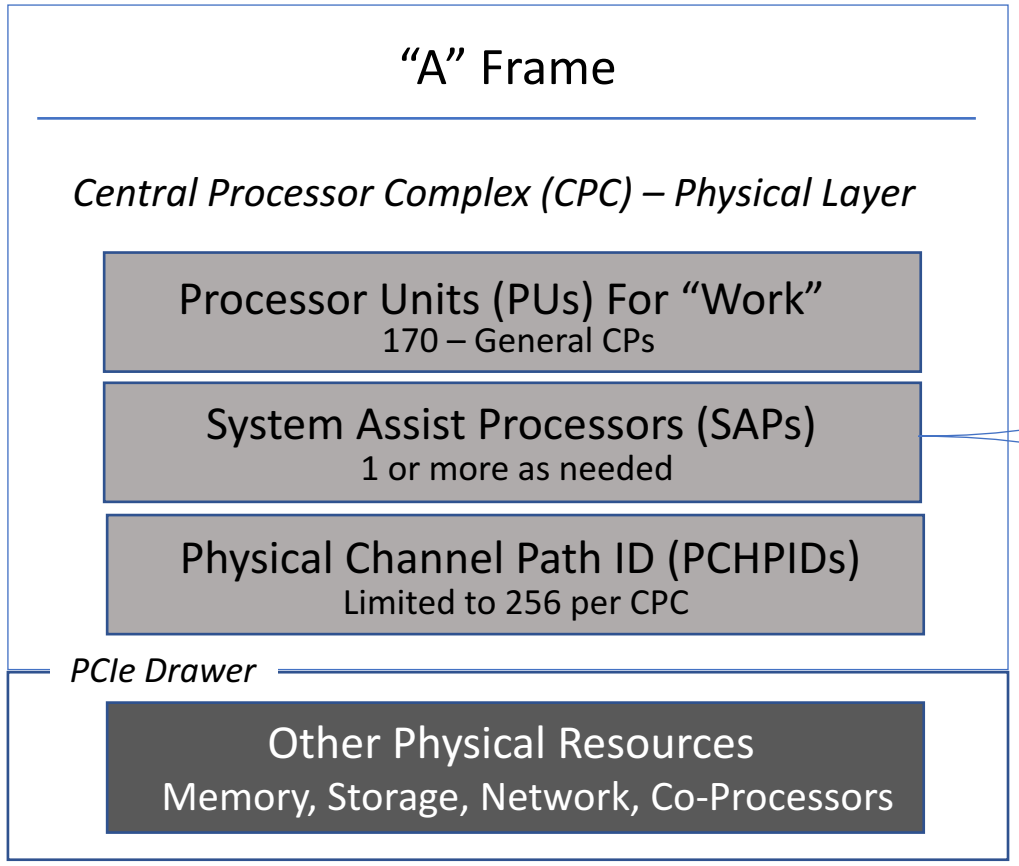
3 - The Mainframe Today.




```
IODF    - Input/Output Definition File - HCD - IOCP, OSCP and SWCP
IOCP    - I/O Configuration Program - Part of IODF — AKA the IOCDS
HCD     - Hardware Configuration Definition
HMC     - Hardware Management Console
```

3 - The Mainframe Today.

Central Processor Complex (CPC)



*Specialty z System Processing Units (PU)**

Designed to Help Reduce Operational Cost

SAP – System Assist Processor
Dedicated to supporting z System Input/Output and Channel Facility Operations.

zIIP – Integrated Information Processor
Dedicated to running specific DB2 processing loads, and for offloading other z/OS workloads.

zAAP – z Application Assist Processor
Dedicated to running specific Java and XML workloads under z/OS.

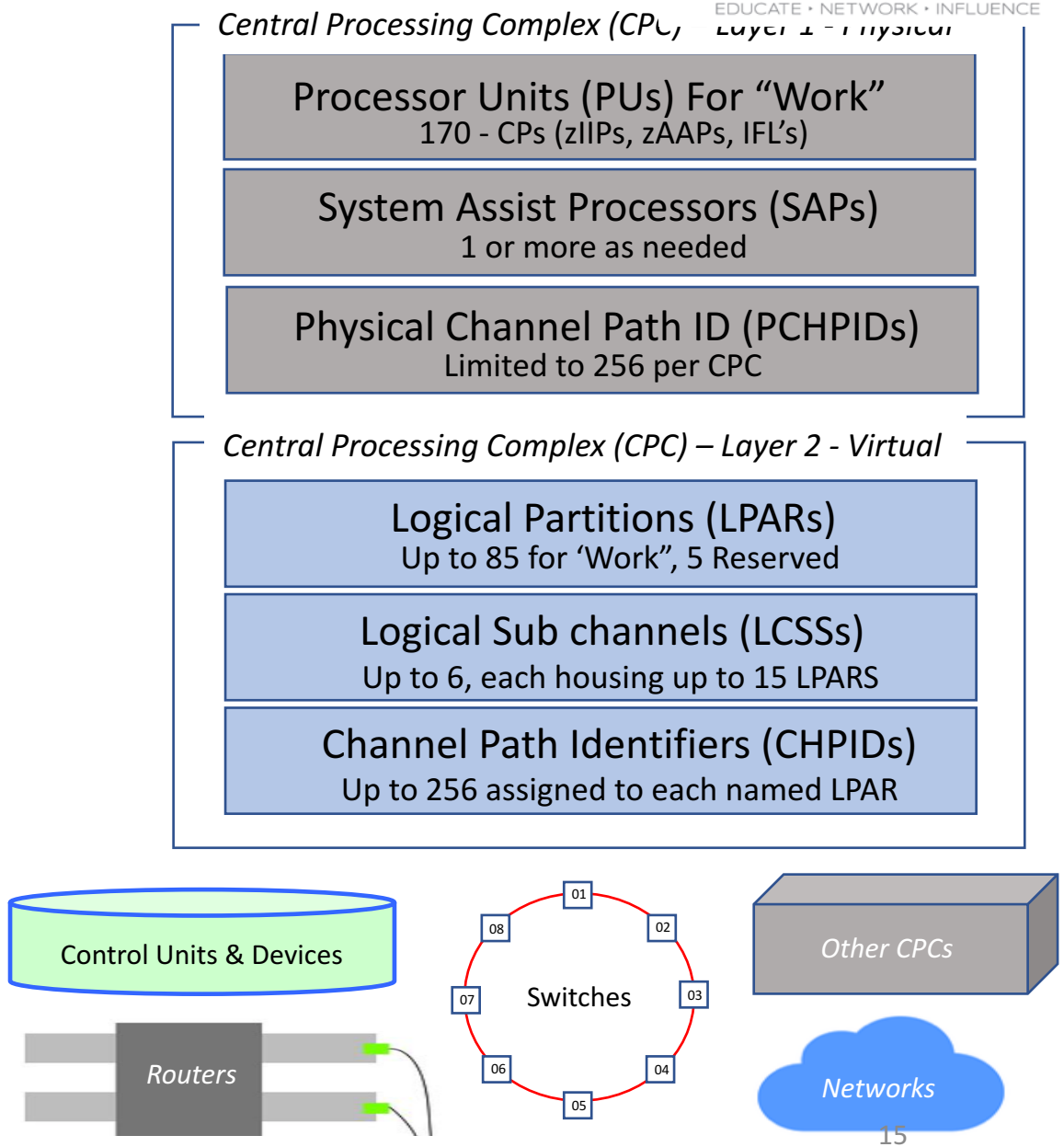
IFL – Integrated Facility for Linux
Dedicated to running specific Linux Workloads under z/OS.

* These processors do not contain microcode or hardware features that accelerate their workloads. Instead, by relieving the general CP of particular workloads, they often lead to a higher workload throughput at reduced license fees. - From Wikipedia

3 - The Mainframe Today.

Each IBM Mainframe (CPC) has a channel subsystem (CSS).

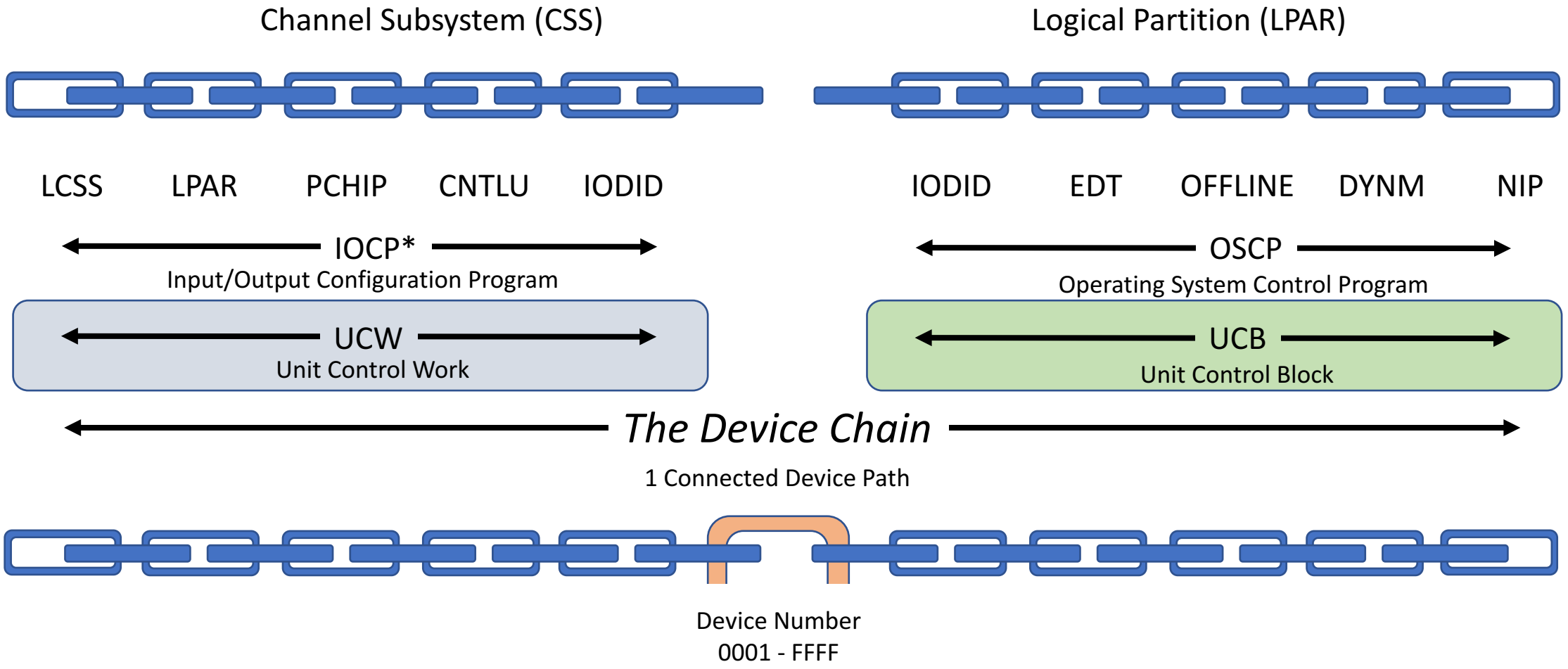
- The role of the CSS is to control communication of internal and external channels, control units and devices for the movement of data.
- The CSS is the very heart of moving data into and out of the CPC and doing so independently of the processors dedicated to the "Work" being performed by the CPC itself. This means that input/output (I/O) to/from the CPC is done asynchronously.
- When I/O operations are required, the CSS is passed the request from the main CPC processor. While awaiting completion of an I/O request, the CPC is able to continue processing other "Work". This is a critical performance factor in a system designed to handle massive numbers of concurrent transactions – Millions-Plus/Second.
- The processors that run the CSS system are called the system assist processors (SAP). There generally are more than one SAPs running the channel subsystem, depending on specific workload requirements.



"It's the Pipe"!

4 – Virtualization - The Device Chain: CSS Segments

Understanding How Data Flows – I/O Addressability



*Sometimes referred to as the IOCDS – Input/Output Configuration Dataset

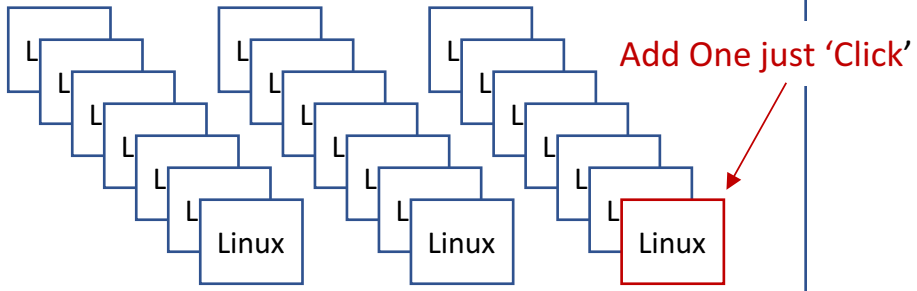
4 - Virtualization - One size fits all

At Power on Reset (POR)

Systems of Engagement | Systems of Record

ID NAME=SMART

One of 85 LPARs



Thousands of Linux Systems

Linux - z/VM - KVM

Set - DPM Mode

Many z/Linux Instances

Vs.

ID NAME=STRONG

One of 85 LPARs

LOGON - START - MOUNT

ASID1 Master	ASIDx JES	ASIDx VTAM	ASIDx TCP/IP	ASIDx ESM	ASIDx CICS
ASIDx UserId	ASIDx UserId	ASIDx UserId	ASIDx UserId	ASIDx UserId	ASIDx UserId

One z/OS System IMAGE – An LPAR

z/OS – z/VM

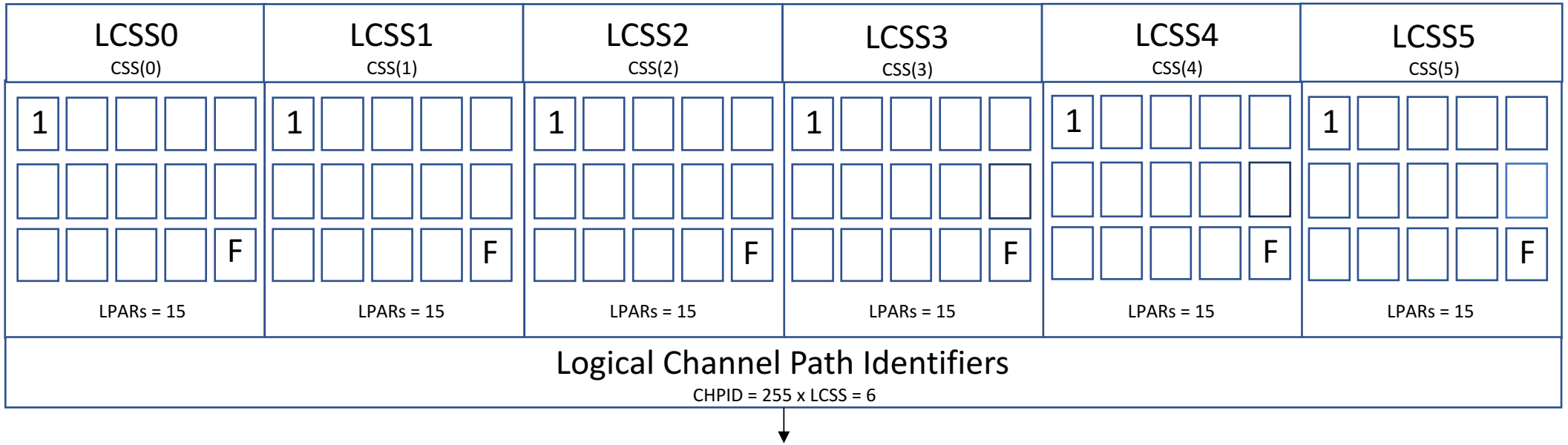
Set - PR/SM Mode

One z/OS Instance

Hypervisor - A process that separates a computer's operating system and applications from the underlying physical hardware. Usually done as software although embedded hypervisors can be created for things like mobile devices.

4 - Virtualization - z/OS Logical Partitions

z 14 Maxed Out

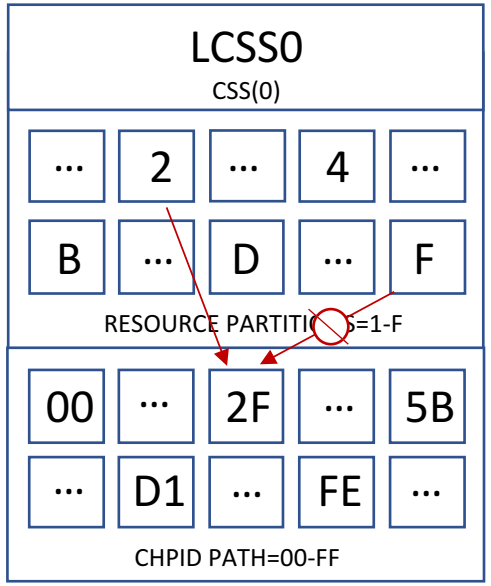


4 - Virtualization - Channel Paths

Types of Channel Paths

Dedicated

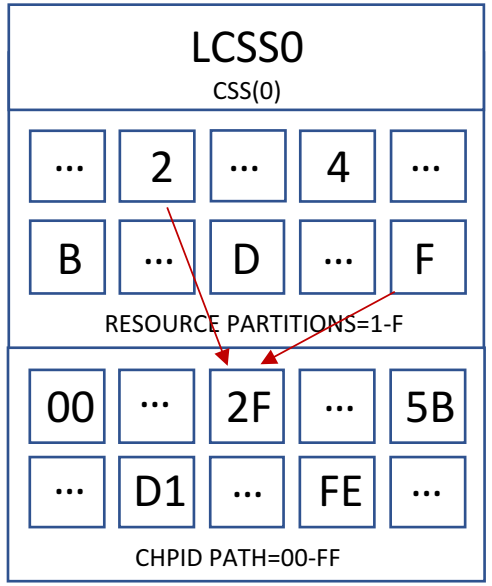
Only 1 LPAR has exclusive access to a named CHPID.



I/O Subsystem

Shared

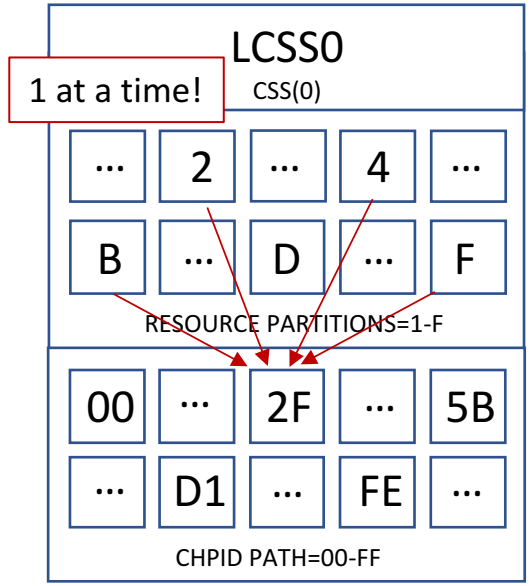
More than 1 LPAR has shared access to a CHPID.



I/O Subsystem

Reconfigurable

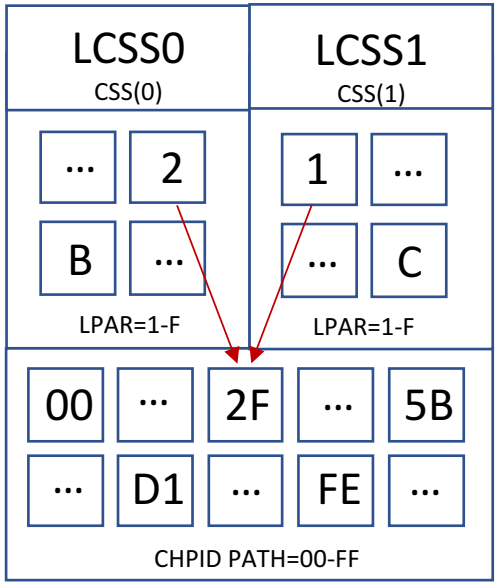
Only 1 LPAR at a time has access to a CHPID.



I/O Subsystem

Spanned

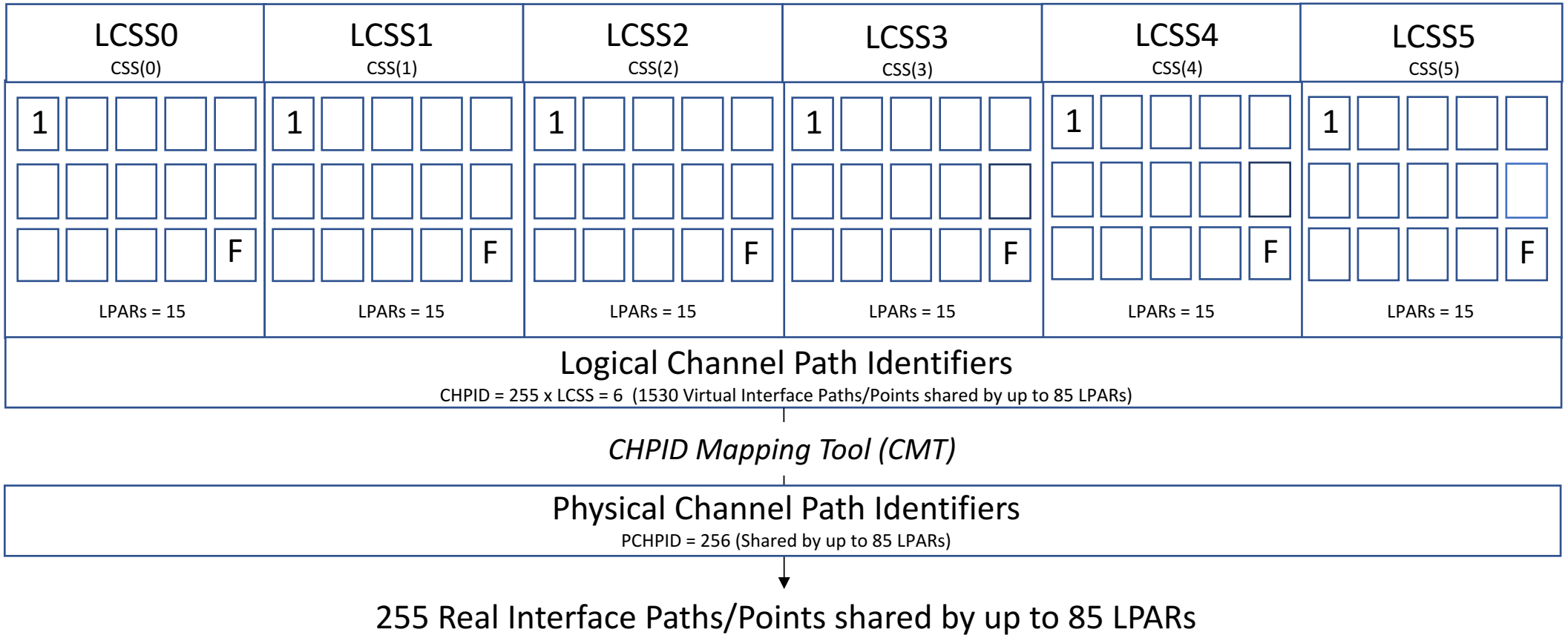
CHPID access by LPARs in different LCSS.



I/O Subsystem

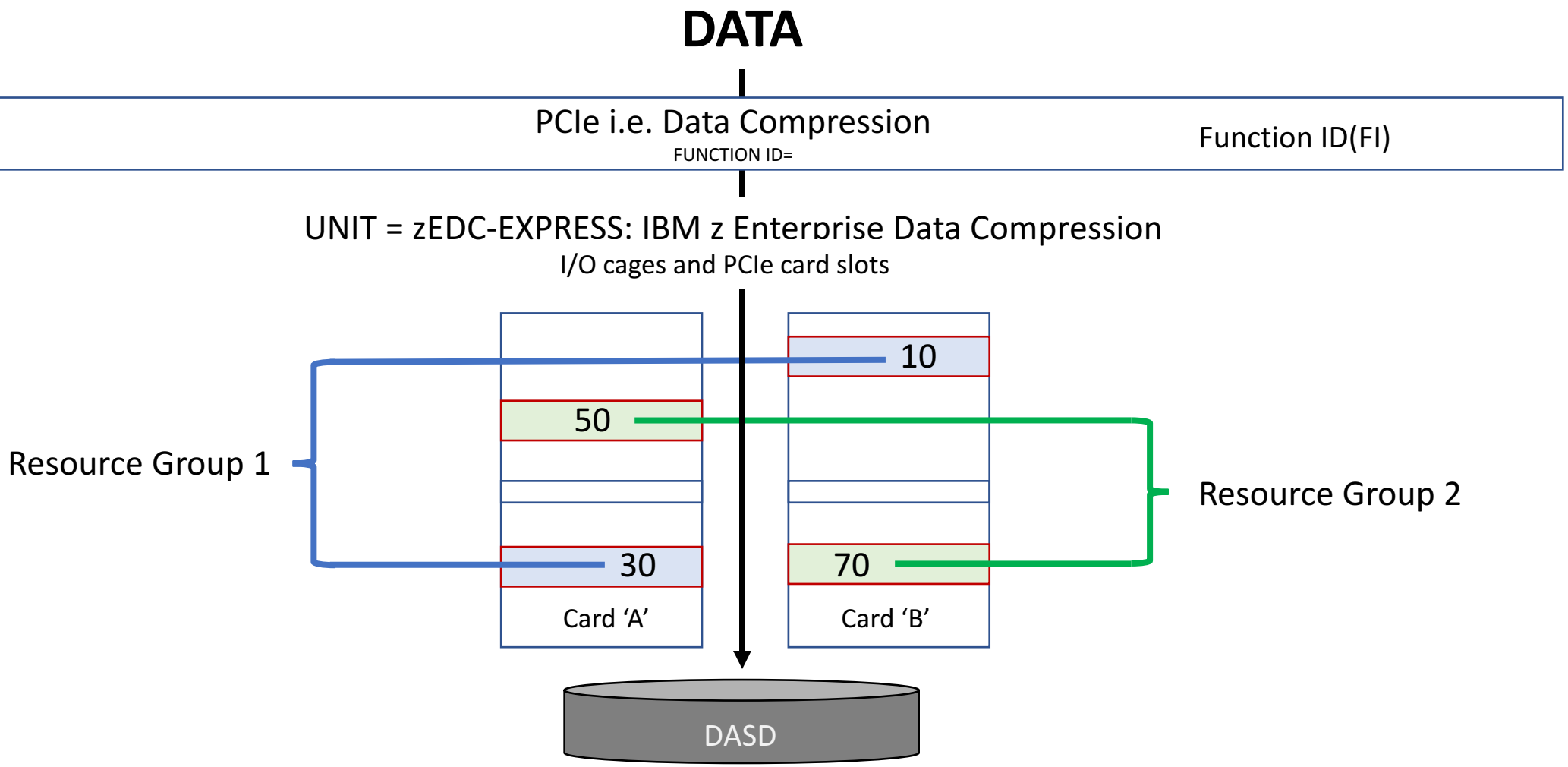
5 - Physicalization - Mapping to the Physical World

Mapping to Reality!



5 - Physicalization - Physical Redundancy

Redundancy will be your friend!



PCIe = Peripheral Component Interconnect Express

5 - Physicalization - Data Encryption

Pervasive DS Encryption - Don't Misplace Your Master Key!

ICSF/TKE – Make the Key
ICSF & HSM En/Decrypt DS

Compressed DATA

ESM – Authorize Access
Dataset, Key Label Binding

Master Key – Tamper Proof

Crypto Engines - Crypto Express5/6S
Symmetric Pervasive Dataset Encryption

Master Key – Key Label

HSM

DFSMS
Data Management

DASD

How to Make a Master Key:

- It takes 2 - 4 cooks to make a Master Key
- Each picks a secret Random String
- The Strings are called Key Parts
- Each enters their secret Key Part
- Each cook saves his/her Key Part
- XOR the pot and you have a Master Key
- Move the Master Key to the HSM
- Label the Key; you're good to go!

We have Separation of Duties:

- DS Administration can: Backup, Archive and Restore without worry of reading it.
- Key Label & DS are bound by ESM. DS Users, with authority, can read/write the Dataset at will. ICSF works with the HSM to encrypt/decrypt as needed.
- PDE does not apply to System, PDS, PDSE and certain other Datasets used at IPL.

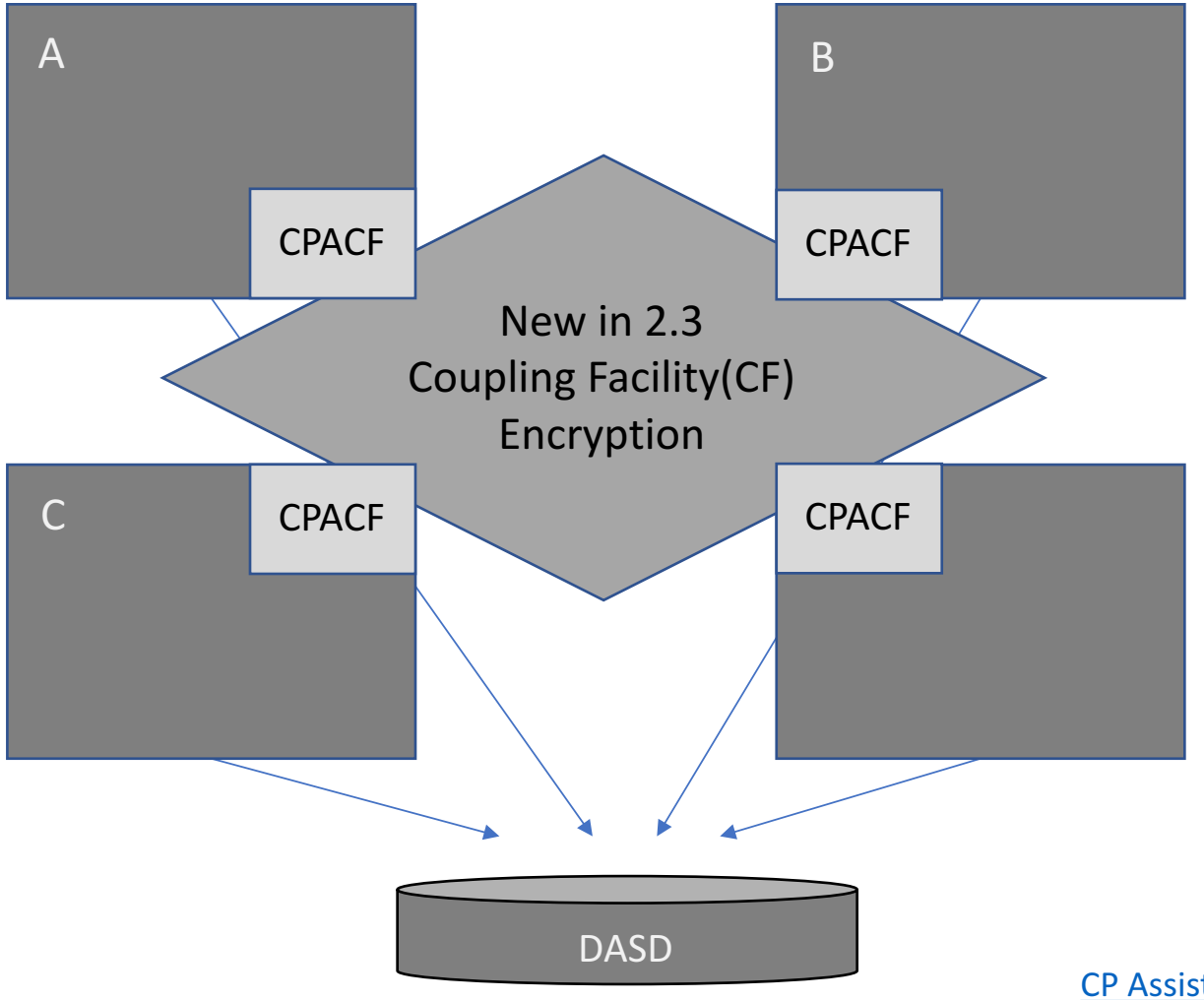
TKE - Trusted Key Entry Workstation
ICSF - Integrated Cryptographic Services Facility

HSM – Hardware Security Module
PDE – Pervasive Dataset Encryption

5 - Physicalization - Data Encryption

Pervasive DS Encryption - Don't Misplace Your Master Key!

A Coupling Facility (CF) allows direct host to host data communication.



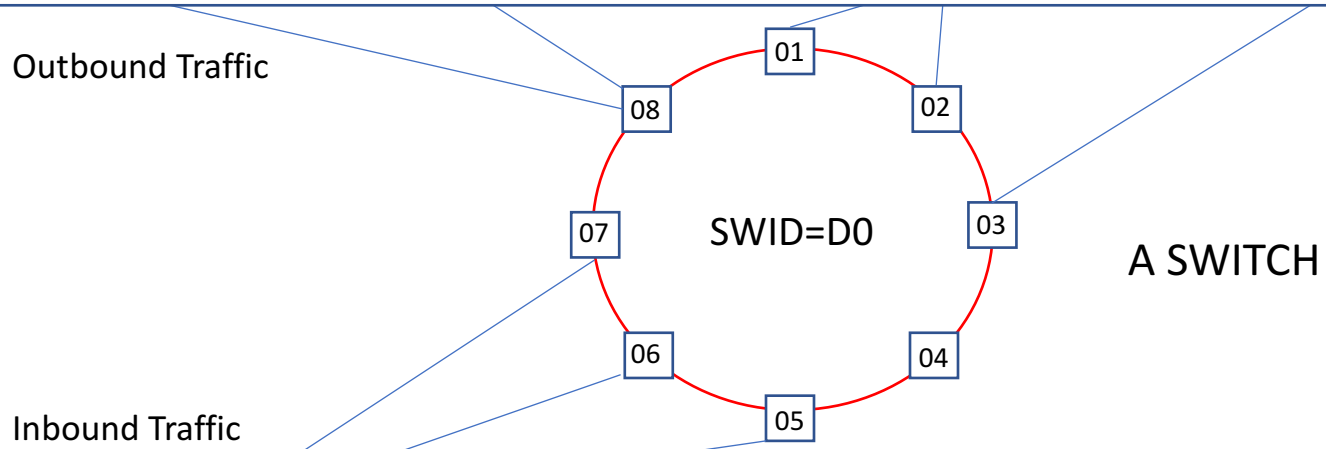
Data encrypted in a host remains encrypted until decrypted by a host.

CP Assist for Cryptographic Functions (CPACF)

5 - Physicalization - Path Multiplication

Contention will not be your friend!

LCSS0 CSS(0)	LCSS1 CSS(1)	LCSS2 CSS(2)	LCSS3 CSS(3)	LCSS4 CSS(4)	LCSS5 CSS(5)
LPARs = 15	LPARs = 15	LPARs = 15	LPARs = 15	LPARs = 15	LPARs = 15
CHPID = 255	CHPID = 255	CHPID = 255	CHPID = 255	CHPIP = 255	CHPID = 255
PCHPID = 255					



A SWITCH and Ports

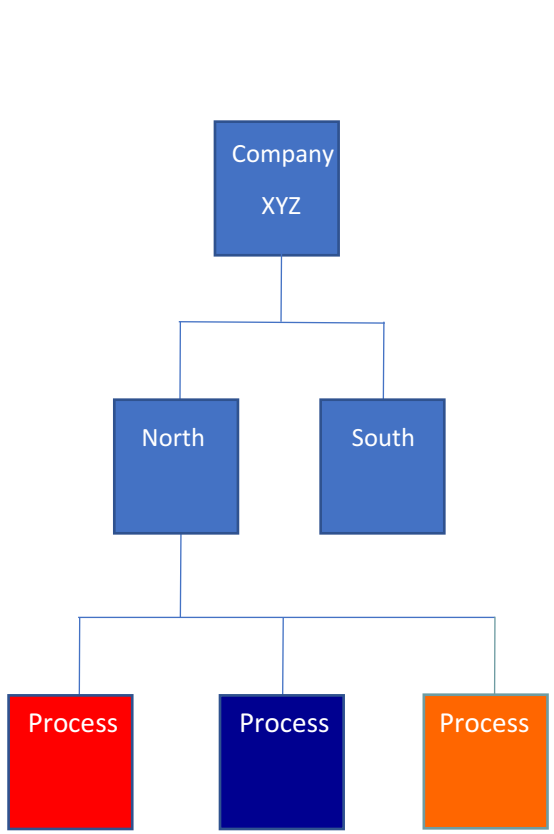
"Head of String"

CtlUnit=1000 — DEVICE=1100 — DEVICE=1200 — DEVICE=1300

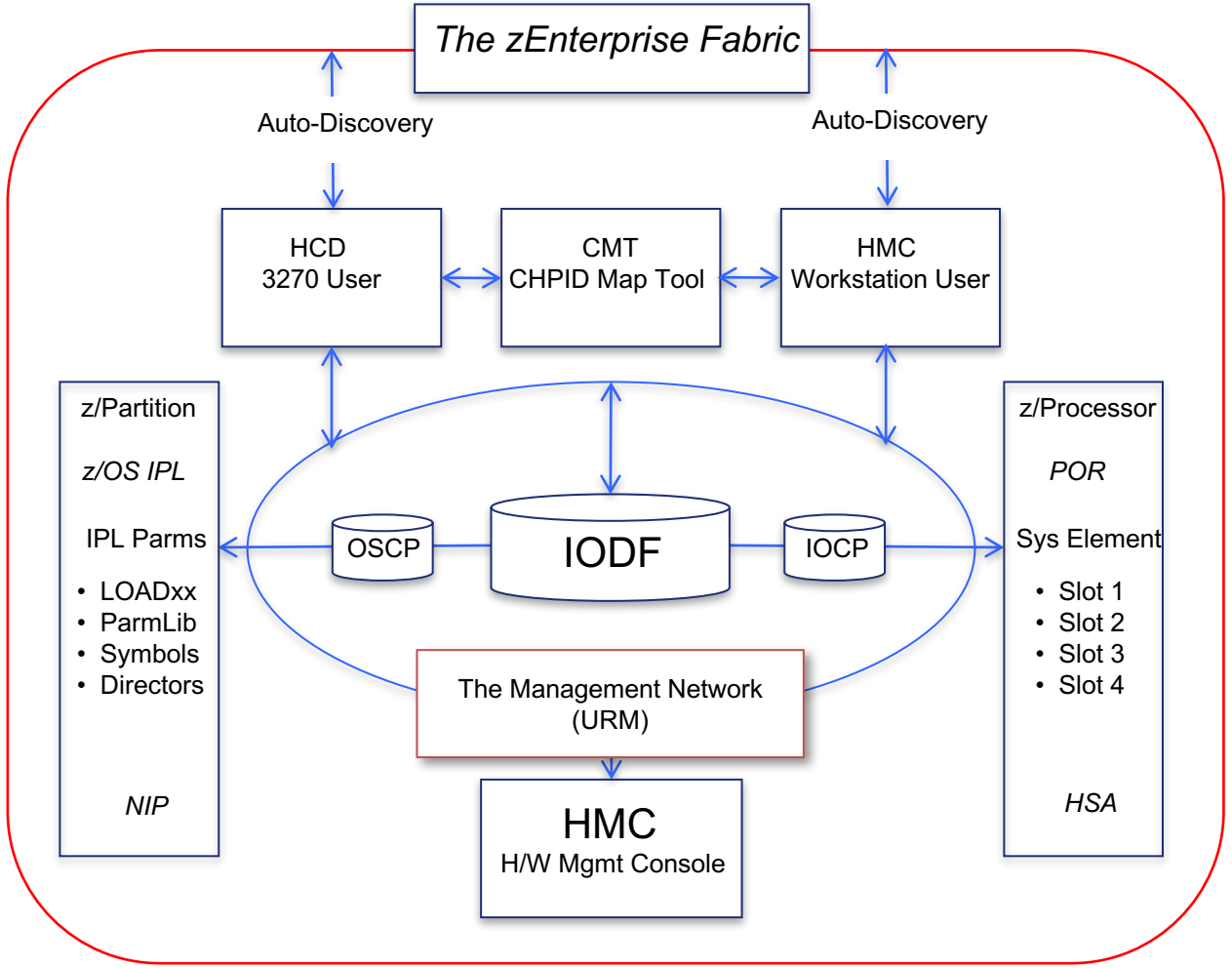
Control Unit with a String of DASD

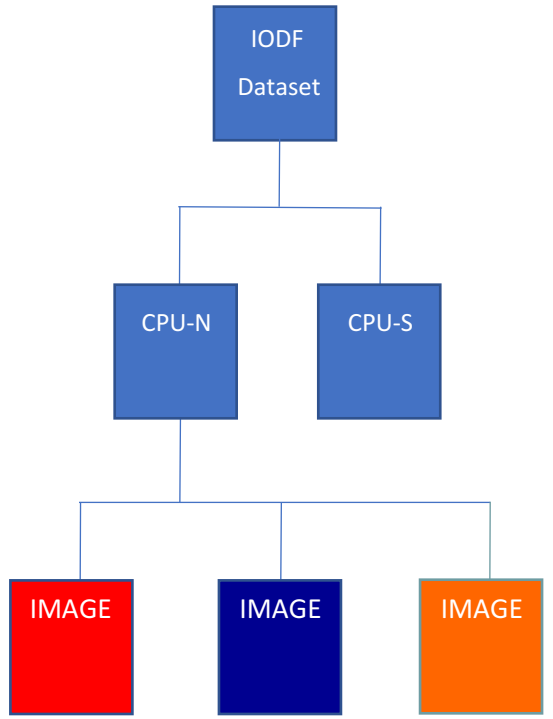
7 - Goals and Control Points - An Organization Mirror

Service Level Agreement (SLA)



Business and Control Objects





z/Platform, z Control Objects

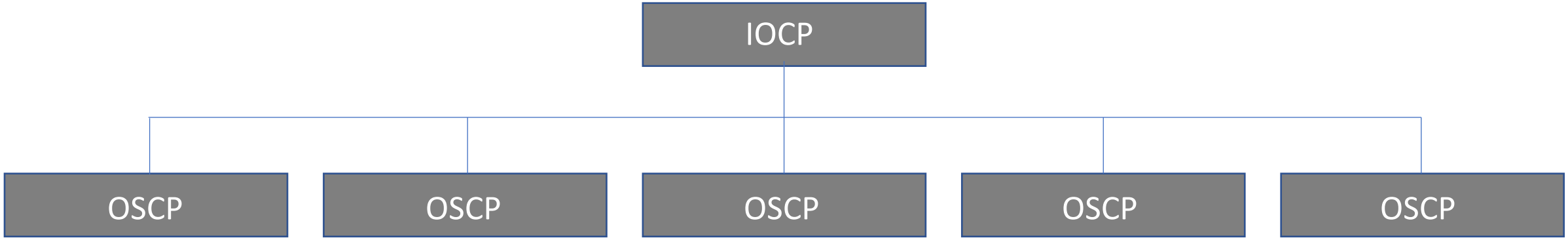
6 - Crafting a Partition - Defining OS Elements

It begins by giving the Partition an ID and a Name!

Each Partition must have its own I/O Configuration.

There may be more than one CEC defined in a single IODF. A single CEC can support up to 85 Partitions. Because of this “Multiplier Effect”, there will be many OSCP configurations defined, in parallel, with a named CEC in the same IODF.

This operational fact makes the identification and naming of each Partition a critical first step in defining the Partitions’ OS Elements.



7 - Goals and Control Points – Assured Separation

Access Vs. Candidate List!

Channel Paths

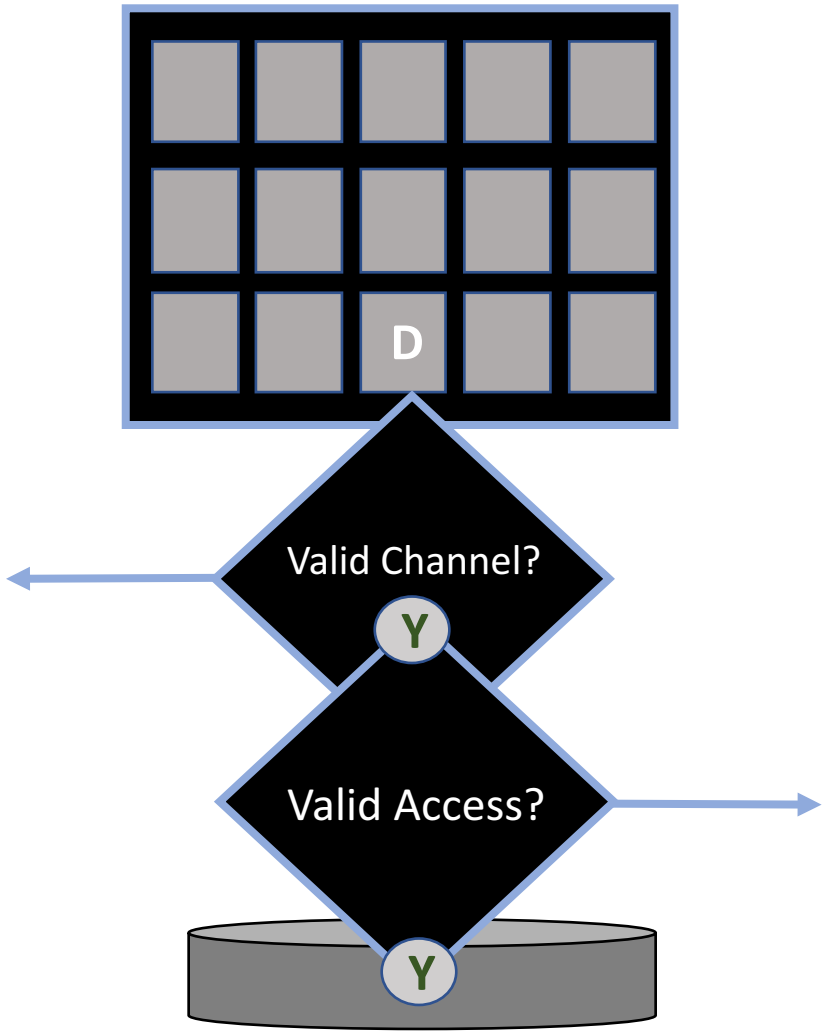
PART, or NOTPART – Not Optional

PART specifies the access list of LPARs that will have a named pair CHPID/PCHID configured online after a Power-on-Reset (POR), and

The related candidate list identifies the LPARs which can actually access the resulting channel path.

NOTPART specifies the negative access list of LPARs that will not have a named pair CSS/CHPID configured online after a Power-on-Reset POR.

The related candidate list of LPARs that cannot access the channel path or device(s) along it.



ON/OFFLINE

I/O Devices

PART or NOTPART - Optional

PART specifies a candidate list of LPARs that can access the device.

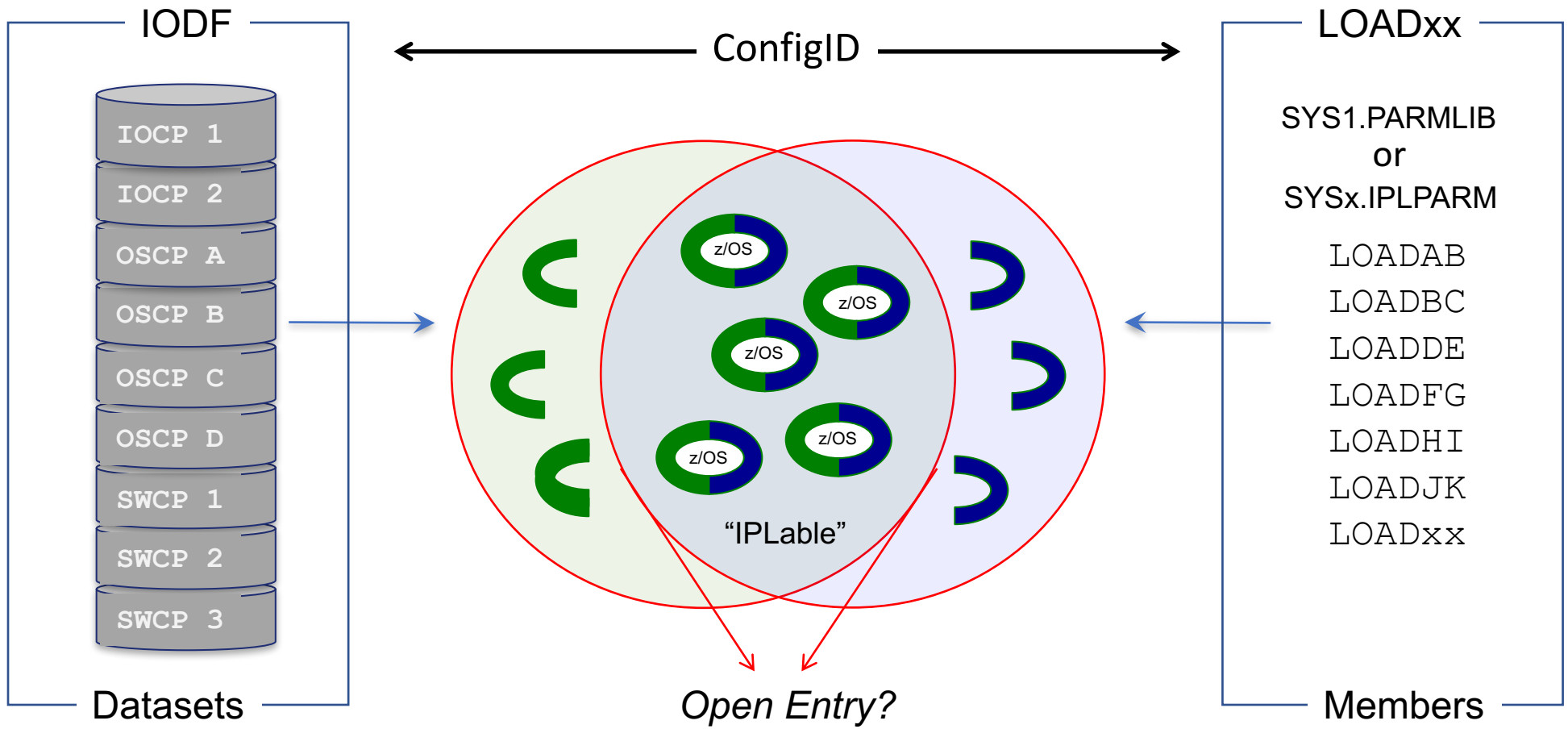
NOTPART specifies a negative candidate list of LPARs that cannot access the device. If not specified, all LPARs may access a named Device.

If PART or NOTPART is in the Device Statement, the CSS sub-keyword is required.

If a device has access to more than one CSS, the CSS sub-keyword indicates to which channel (CSS) subsystem the partition belongs.

7 - Goals and Control Points – Cleanup is Required

It's a *Best Practice* to identify and purge mismatched ConfigIDs.



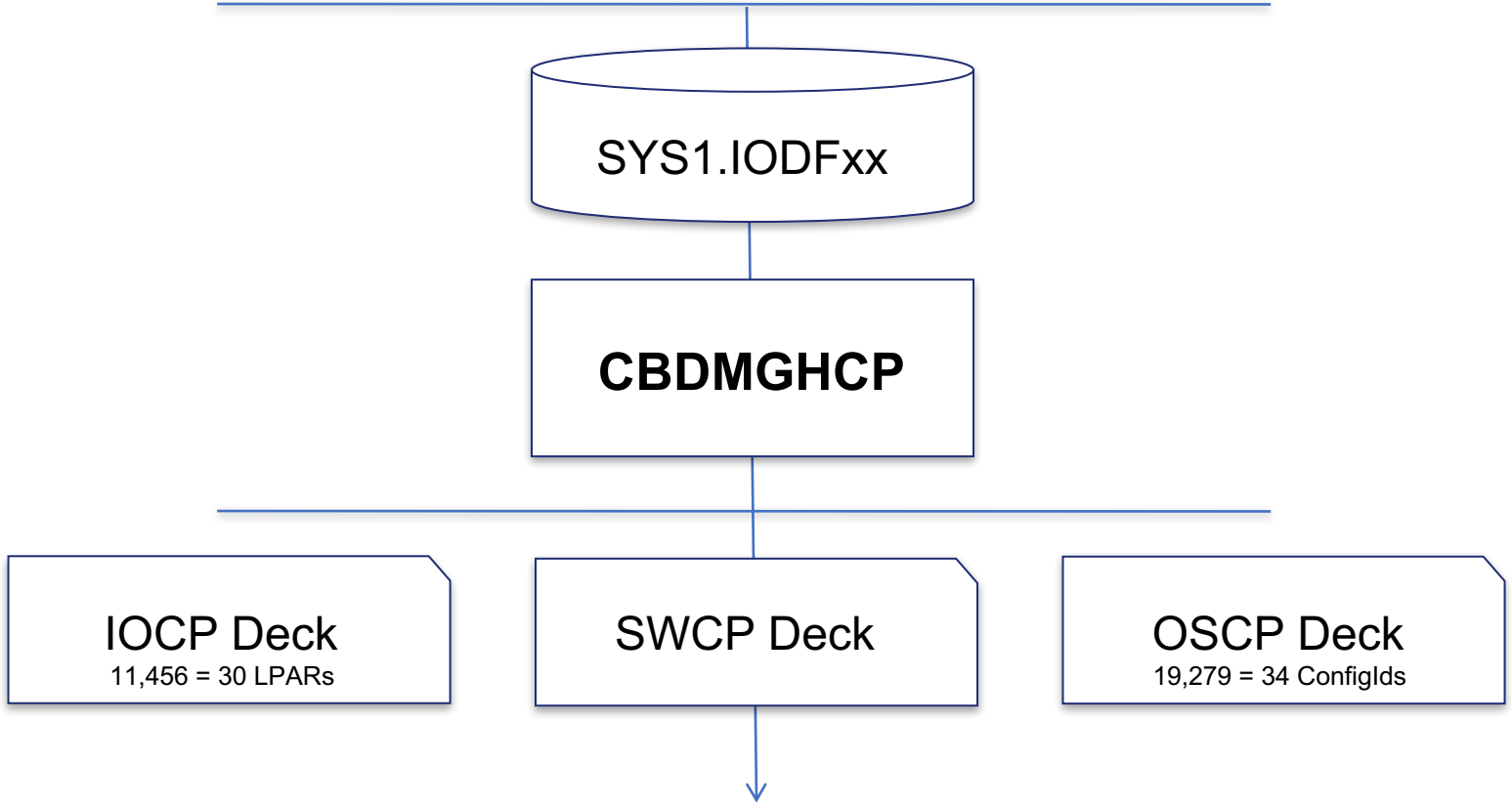
Orphaned OSCP CONFIGID - z/OS IPLable – OSCP CONFIGID matches with LOADxx Member - Orphaned LOADxx Members

7 - Goals and Control Point - The Unknown can be Known

Publication SC34-2669-03 – Hardware Configuration Definition User's Guide

Chapter 13. How to invoke HCD batch utility functions

You Can See it!



Chapter 12. How to migrate existing input data sets

8 - Unpacking the Configuration - Naming the CEC Configuration

The CEC Details

```
TITLE 'SYS1.IODFA1 - 2017-08-27 20:38:47  '
*
 ID NAME=CPA,UNIT=2827,MODEL=H66,DESC='IBM zEC12 in NL',        *
       SERIAL=05A5372827,MODE=LPAR,LEVEL=H130331,LSYSTEM=CPA,   *
       SNAADDR=(IBM390PS,CPA),                                  *
       SCR='CPA      .ü..◊à.ã..ßü...◊........17-08-2720:38:47SYS*
       1    IODFA1  '
```

UCW
Unit Control Work

8 - Unpacking the Configuration - Defining/Naming the LPARS

Logical Partition Details

```
RESOURCE PARTITION=((CSS(0),(CPAUS10A,A),(CPAUS102,2),(CPAUS10*
      6,6),(CPA0B,B),(CPA0C,C),(CPA0D,D),(CPA0F,F),(CPA01,1),(*
      CPA03,3),(CPA04,4),(CPA05,5),(CPA07,7),(CPA08,8),(CPA09,*
      9),(*,E))),MAXDEV=((CSS(0),65280,65535,65535)),          *
      DESCL=('CPAUS10A(PNB2DR)','CPAUS102 (PNB1DR)','CPAUS106 *
      (PNBZ)','CPA0B (SYSKDR)','CPA0C (SYSCDR)','CPA0D (SYSG)'*
      ,'HBUSCFDR (HBUS ICF)','CPA01 (HFN1DR)','CPA03 (HISV)','*
      CPA04 (HIS7)','CPA05 (SYSL)','CPA07 (HTSW)','CPA08 (SYSS*
      )','CPA09 (SYST-NL)'),                                   *
      USAGE=(OS,OS,OS,OS,OS,OS,CF,OS,OS,OS,OS,OS,OS,OS,CF/OS)
```

UCW
Unit Control Work

8 - Unpacking the Configuration - Setting up Channel Pathways

Logical and Physical Channels Paths

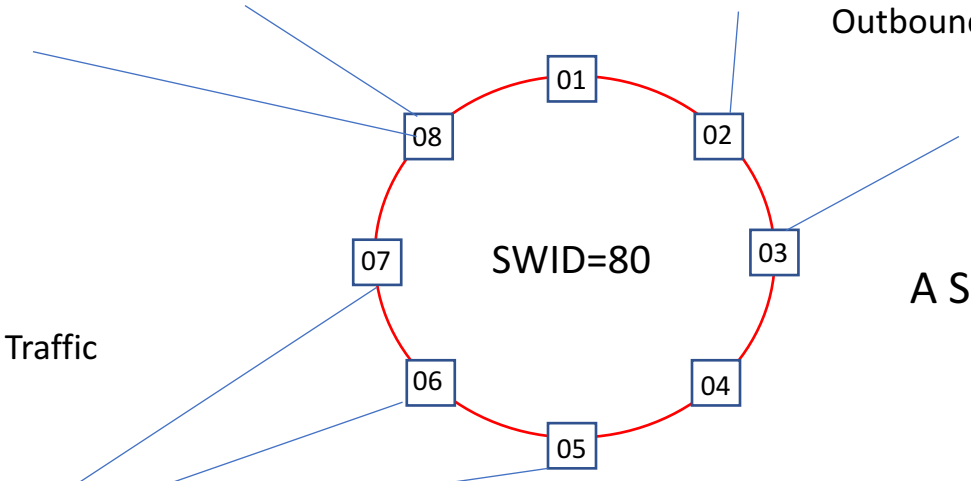
```
CHPID PATH=(CSS(0,1),00),                                          *
      PARTITION=((CSS(1),(0),(CPA11))),                            *
      NOTPART=((CSS(0),(CPA01,CPAUS102,CPA03,CPA04,CPAUS106,CP*
      A07,CPA08,CPA09,CPA0C,CPA0D),(CPA0F))),PCHID=588,            *
      TYPE=OSD,SWPORT(80,00)
CHPID PATH=(CSS(0,1),01),SHARED,                                   *
      PARTITION=((CSS(1),(CPA11),(=))),                            *
      NOTPART=((CSS(0),(CPA01,CPAUS102,CPA03,CPA04,CPA05,CPAUS*
      106,CPA07,CPA08,CPA0B,CPA0D),(CPA0F))),PCHID=58C,            *
      TYPE=OSD
CHPID PATH=(CSS(0,1),02),REC,                                      *
      PARTITION=((CSS(0),(CPAUS106,CPAUS10A),(CPA01,CPAUS102,C*
      PA03,CPA04,CPA05,CPA08,CPA09,CPA0B,CPA0C,CPA0D)),(CSS(1)*
      ,(0),(CPA11))),PCHID=5B4,TYPE=OSD
```

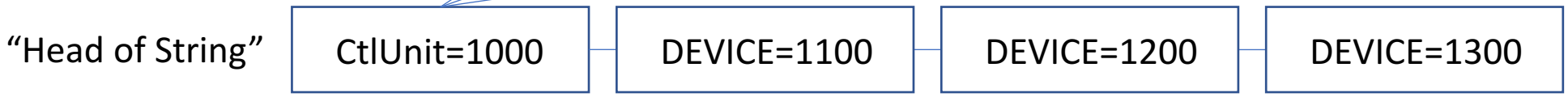
UCW
Unit Control Work

8 - Unpacking the Configuration – Eliminating Device Contention

The Switch

```
SWITCH SWID=80,ADDRESS=80,DESC='Great Switch of North',      *
      PORT=((00,FF)),UNIT=2032,SWPORT=((08,02,01),(07,06,05))
PORT ID=07,OCC
PORT ID=08,OCC
PORT ID=0A,OCC
PORT ID=0B,OCC
PORT ID=0C,OCC
PORT ID=0D,OCC
```





UCW
Unit Control Work

8 - Unpacking the Configuration – I/O the Final Hardware Link

Control Units and I/O Devices

```
CNTLUNIT CUNUMBR=0460,                                          *
      PATH=((CSS(0),A6,A9,C2,C8),(CSS(1),A6,A9,C2,C8)),         *
      UNITADD=((00,16)),                                        *
      LINK=((CSS(0),800F,820F,802F,810F),(CSS(1),800F,820F,802*
      F,810F)),SHARED=N,CUADD=0,                                *
      DESC='VH LIBID=C6666,LIBPORT=41,CUA=0',UNIT=3490,         *
      SWPORT=((80,07),(80,06),(81,0F),(82,0F))
IODEVICE ADDRESS=(0460,16),UNITADD=00,CUNUMBR=(0460),STADET=Y,*
      UNIT=3490
```

IODevice Mathematics: Read ADDRESS=(0460,16) as saying:

- Add 16 Devices of this type beginning with unit address 0460, attached to Named Control Unit

UCW
Unit Control Work

8 - Unpacking the Configuration – Naming the OS Configuration

Partition ConfigID and Name!

```
IOCONFIG ID=00,NAME=AMHDEV,TYPE=MVS,DESC='BUF DEV'
IODEVICE ADDRESS=(0070,15),UNIT=OSA,OFFLINE=NO,DYNAMIC=YES,    *
      LOCANY=YES,CUNUMBR=0070,SCHSET=(CSS(0),1)
IODEVICE ADDRESS=(007F,1),UNIT=OSAD,OFFLINE=NO,DYNAMIC=YES,    *
      LOCANY=YES,CUNUMBR=0070,SCHSET=(CSS(0),2)
```

I/O device definitions require Unit information modules (UIMs). A UIM contains the information and rules that HCD uses to process device definitions. When an IODF is created, HCD uses the UIMs to validate the device definitions entered and uses them during an IPL to build device related UCBs.

UIMs are provided with IBM® product software for devices that z/OS® supports. A generic UIM or a UIM from a similar IBM device is used for non-IBM® products.

SCHSET indicates the subchannel set (1,2,3,*) for normal base devices that have a special secondary device with the same address. The indicator provides direction at IPL such that the correct device(s) is online.

UCB
Unit Control Block

8 - Unpacking the Configuration - Configuring Device Specifics

Eligible Device Table (EDT)!

```
EDT ID=A1,DESC='BUF DEV EDT'
UNITNAME NAME=EAUTO,                                          *
      UNIT=((2100,16),(2110,16),(2120,16),(2130,16),(2140,16),*
      (2150,16),(2160,16),(2170,16),(2180,16),(2190,16),(21A0,*
      16),(21B0,16),(21C0,16),(21D0,16),(21E0,16),(21F0,16),(2*
      200,16),(2210,16),(2220,16),(2230,16),(2240,16),(2250,16*
      ),(2260,16),(2270,16))
```

The eligible device table (EDT) is an installation-defined and named representation of the I/O devices defined to the device chain that are eligible for allocation. At IPL, information in the IODF and UIMs is used to build the EDT.

- Generic Device: A generic device is an MVS-defined grouping of devices with similar characteristics, which determines how devices are used in an esoteric device group.
- Esoteric Device: A single virtual device which can be allocated without using a specific device number. Devices within an esoteric may or may not be from the same device group.

UCB
Unit Control Block

8 - Unpacking the Configuration - NIP Console - the Final OS Link

Nucleus Initialization Console (NIPCON)!

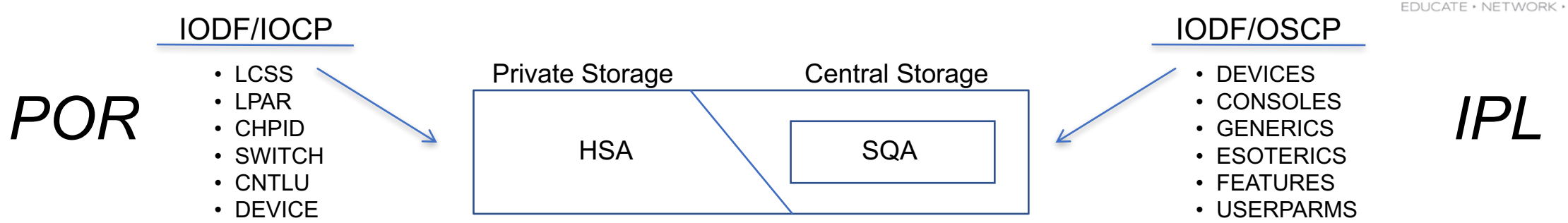
```
NIPCON DEVNUM(2100,2110,2116)
```

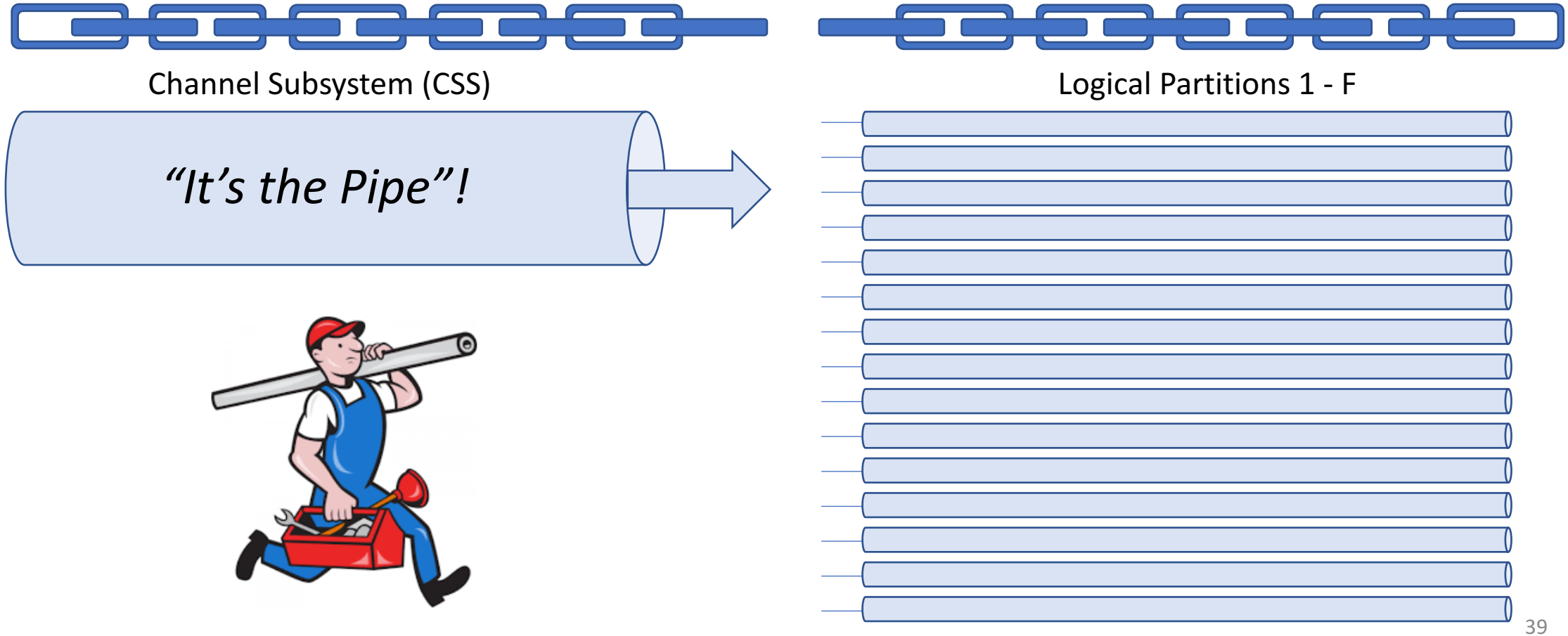
Specifies a list of device(s) to be used by MVS Operators as NIP consoles during the IPL.

- NIPCON and related devices are mandatory.
- All devices specified by NIPCON must be defined in the OSCP configuration.

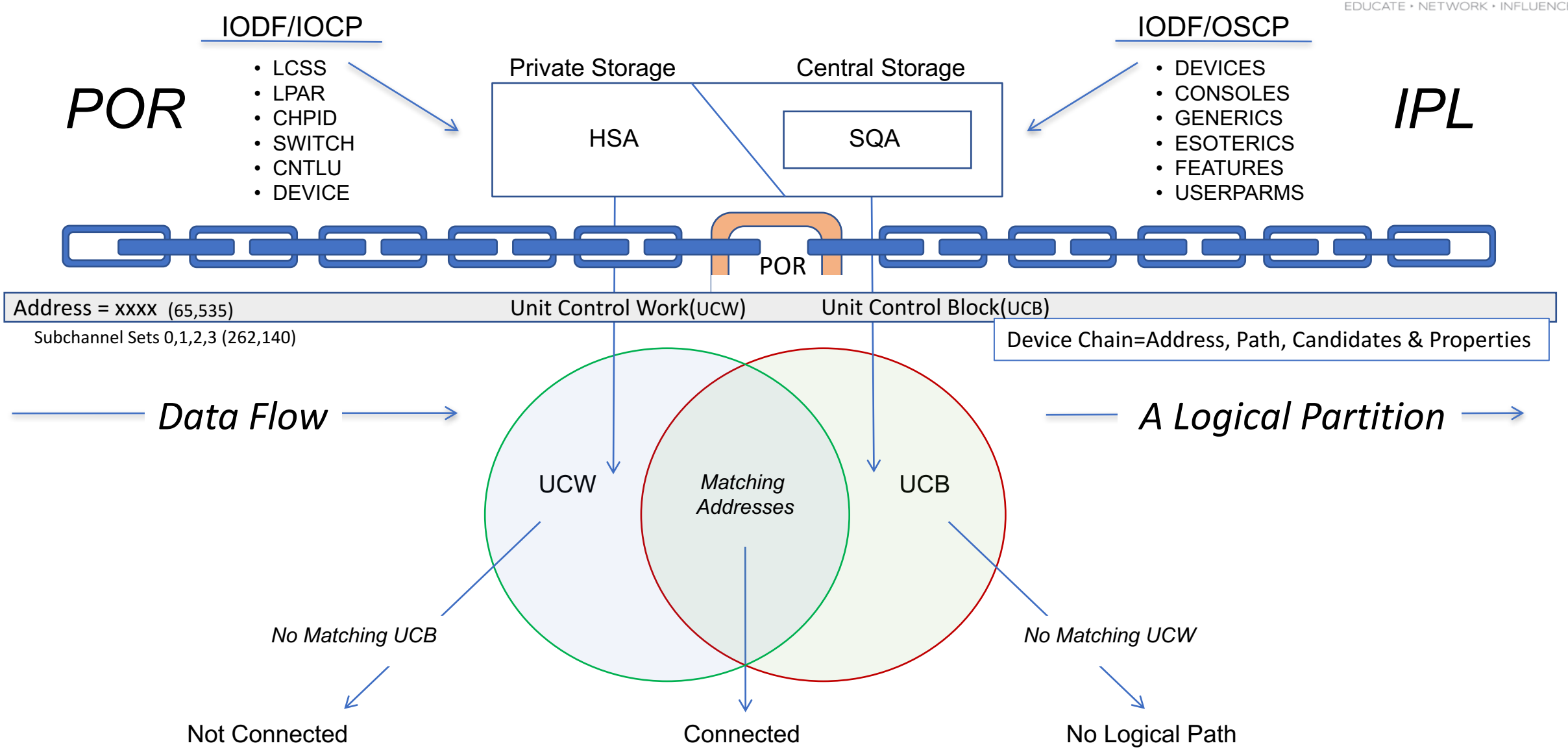
UCB
Unit Control Block

9 - Power-on-Reset and/or IPL – Almost done!





9 - Power-on-Reset and/or IPL – Linking UCWs to UCBs



UCW=Unit Control Work -> HSA=Hardware Storage Area
UCB=Unit Control Block -> SQA=System Queue Area

A – Defining a z/OS Image

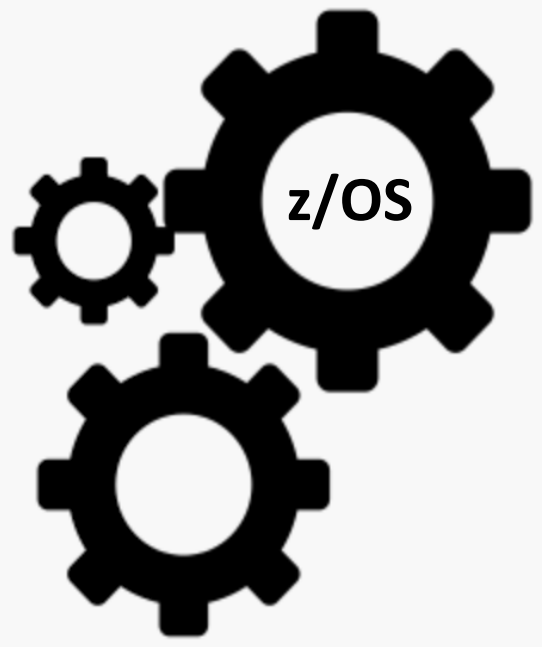
The Lessons Continue

- System Integrity
  Problem Vs. Supervisor State
- IPL Parameters
  IPLPARM - LOADxx
- z/OS Configuration
  PARMLIB – Parameters & Directors
- Start Parameters
  Auto Start Sub-Systems & Task
- Job Entry Parameters
  Yours, to Submit
- Data Security
  SAF, SMF, ESM
- System Modules
  SVC, LNK, LPA, EXIT
- Sub-System Modules
  JES, VTAM, TCP/IP, HCKR
- Vendor Modules
  IBM, CA, BMC, CompuWare
- Site Modules
  Yours, Locally Written

New 102 Words

```
APF    - Authorized Program Facility
ASID   - The Numeric Address Space Identifier
BCP    - The Base Control Program
DUCT   - Dispatchable Unit Control Table
IMSI   - Message Suppression Indicator
IRIM   - IPL Resource Initialization Modules
PPT    - Program Properties Table
PSW    - Program Status Word
SAF    - System Access Facility
SPKA   - Set Storage Protect Key
SVC    - Supervisor Call
TCB    - Task Control Block
```

Thank You – Evaluations Please



Let's Build a z Environment - 101

Session 23330

Tuesday, August 14 at 10:00-11:00 AM

Room 242

Presented by Paul R. Robichaux
NewEra Software, Inc.
prr@newera.com