Intro to z/OS: Part 2

Stephen Warren
Senior Technical Staff Member
Client Architect, IBM Worldwide Client Experience Center

November 2019
Session AG




Agenda

Part 1 – Previous session:

Why Z Matters

Hardware/LPAR

z/OS Components

Software Stack

App Dev, App Exec and Mgmt Envs

DASD

Data Sets / Allocation

TSO/E

ISPF

z/OS UNIX/ISHELL/OMVS/Remote

Address Spaces & Modes

Storage & DAT

Part 2:

Batch Processing/JES/JCL

Job Flow

SDSF

System Log

VTOC & Catalogs

PDS & PDSE

SMS

IPL

Sysplex/GDPS

Serialization

Managing Workloads

Batch Processing

Batch Processing

- z/OS is also **ideal for batch** jobs
  - **Workloads** that run in **background**
  - Little or **no human interaction**

Batch Processing

- Many customer **core applications**, such as payroll, are **performed** through **batch processing** (jobs)

Batch Processing

- Many customer **core applications**, such as payroll, are **performed** through **batch processing** (jobs)
- Jobs **run without** end user **interaction**
- Run **as resources permit**

Batch Processing

- Many customer **core applications**, such as payroll, are **performed** through **batch processing** (jobs)
- Jobs **run without** end user **interaction**
- Run **as resources permit**
- **JCL** (*Job Control Language*) is used to **control the operation** of each job

Batch Processing

```
 File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
------------------------------------------------------------------------------
VIEW        BIBOLET.ASSEMBLE.JCL(HELOWRLD) - 01.00          Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ******************************** Top of Data ********************************
000100 //BIBOLETC  JOB CLASS=J,
000200 //      MSGLEVEL=(1,1),MSGCLASS=H,NOTIFY=BIBOLET
000300 //*******************************************************************
000400 //*                  ASSEMBLE  STEP                                 *
000500 //*******************************************************************
000600 //ASM      EXEC PGM=ASMA90,REGION=0M,
000700 //     PARM=('LINECOUNT(111),USING(WARN(11)),XREF(SHORT),DECK')
001100 //SYSIN    DD  DSN=BIBOLET.ASSEMBLE.SOURCE(HELOWRLD),DISP=SHR
001200 //SYSPUNCH DD  DSN=BIBOLET.ASSEMBLE.OBJ(HELOWRLD),DISP=SHR
001300 //SYSLIN   DD  DUMMY
001400 //SYSPRINT DD  DSN=BIBOLET.ASSEMBLE.LISTCASM(HELOWRLD),DISP=SHR
001500 //SYSPRINT DD  DSN=&&PLSPRT,DISP=(MOD,PASS)
001600 //SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(5,5))
001700 //SYSLIB   DD  DSN=BIBOLET.ASSEMBLE.SOURCE,DISP=SHR
001800 //         DD  DSN=ZBLD.HBB77B0.MACLIB,DISP=SHR
001900 //         DD  DSN=SYS1.MACLIB,DISP=SHR
002000 //*******************************************************************
002100 //*             LINKAGE EDITOR STEP
002200 //*******************************************************************
```

Example of JCL

Batch Processing

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help

VIEW        BIBOLET.ASSEMBLE.JCL(HELOWRLD) - 01.00          Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
****** ******************************** Top of Data ********************************
000100 //BIBOLETC  JOB CLASS=J,
000200 //         MSGLEVEL=(1,1),MSGCLASS=H,NOTIFY=BIBOLET
000300 //*********************************************************
000400 //*                  ASSEMBLE  STEP
000500 //*********************************************************
000600 //ASM      EXEC PGM=
000700 //         PARM=('LINECOUNT(111),USING(WARN(2
001100 //SYSIN     DD  DSN=BIBOLET.ASSEMBLE.SOURCE(HELOWRL
001200 //SYSPUNCH  DD  DSN=BIBOLET.ASSEMBLE.OBJ(HELOWRLD),
001300 //SYSLIN    DD  DUMMY
001400 //SYSPRINT  DD  DSN=BIBOLET.ASSEMBLE.LISTCASM(HELOWRLD),DISP=SHR
001500 //SYSPRINT  DD  DSN=&&PLSPRT,DISP=(MOD,PASS)
001600 //SYSUT1    DD  UNIT=SYSDA,SPACE=(CYL,(5,5))
001700 //SYSLIB    DD  DSN=BIBOLET.ASSEMBLE.SOURCE,DISP=SHR
001800 //          DD  DSN=ZBLD.HBB77B0.MACLIB,DISP=SHR
001900 //          DD  DSN=SYS1.MACLIB,DISP=SHR
002000 //*********************************************************************
002100 //*                  LINKAGE EDITOR STEP                              *
002200 //*********************************************************************
```

JCL statements start with a **// in column 1**

Batch Processing

- An **initiator** (system program)
  - **Processes JCL**
  - **Creates environment** in an address space
  - **Runs the job** in that address space

Batch Processing

- **Management** of jobs & resources are **shared** between **z/OS and JES** (*Job Entry Subsystem*)

Batch Processing

- **Management** of jobs & resources are **shared** between **z/OS and JES** (*Job Entry Subsystem*)

- **JES**
  - **Receives** jobs into system
  - **Queue** jobs waiting **to be executed**

Batch Processing

- **Management** of jobs & resources **shared** between **z/OS and JES** (*Job Entry Subsystem*)

- **JES**
  - **Receives** jobs into system
  - **Queue** jobs waiting **to be executed**
  - Manages **priority**
  - **Schedules** for processing
  - Controls **output** processing

Batch Processing

- **Management** of jobs & resources **shared** between **z/OS and JES** (*Job Entry Subsystem*)

- **JES**
  - **Receives** jobs into system
  - **Queue** jobs waiting **to be executed**
  - Manages **priority**
  - **Schedules** for processing
  - Controls **output** processing
  - Uses **Spooling** (*Simultaneous Peripheral Operations OnLine*)
    - Process of **reading/writing data** on storage devices **concurrently with job execution**

Batch Processing

- **Management** of jobs & resources **shared** between **z/OS and JES** (*Job Entry Subsystem*)
- **JES**
  - **Receives** jobs into system
  - **Queue** jobs waiting **to be executed**
  - Manages **priority**
  - **Schedules** for processing
  - Controls **output** processing
  - Uses **Spooling** (*Simultaneous Peripheral Operations OnLine*)
    - Process of **reading/writing data** on storage devices **concurrently with job execution**
- **z/OS**
  - **Processes** the job

JES

Batch Processing – JES

- IBM **provides two** kinds of JES
- **JES2**
- **JES3**

Batch Processing – JES

- **Both provide JES functions** but in different ways
  - **Accept** jobs submitted in various ways
  - **Queue jobs** waiting to be executed
  - **Queue** jobs for an **initiator**

Batch Processing – JES

- **Both provide JES functions** but in different ways
- An overwhelmingly large percent of customers use **JES2**

Batch Processing – JES

- **Both provide JES functions** but in different ways
- An overwhelmingly large percent of customers use **JES2**
- IBM has made a recent statement of direction that **JES3** will not be supported many years from now.

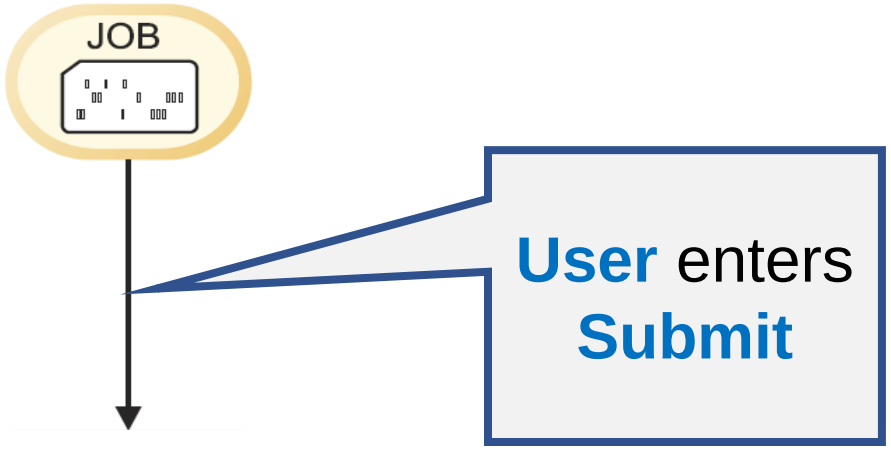
Batch Processing – JES

- **Both provide JES functions** but in different ways
- An overwhelmingly large percent of customers use **JES2**
- IBM has made a recent statement of direction that JES3 will not be supported many years from now.
  - IBM has been migrating many of the **JES3** functions onto **JES2.**
  - Some of the remaining **JES3** customers are now migrating from **JES3** to **JES2**

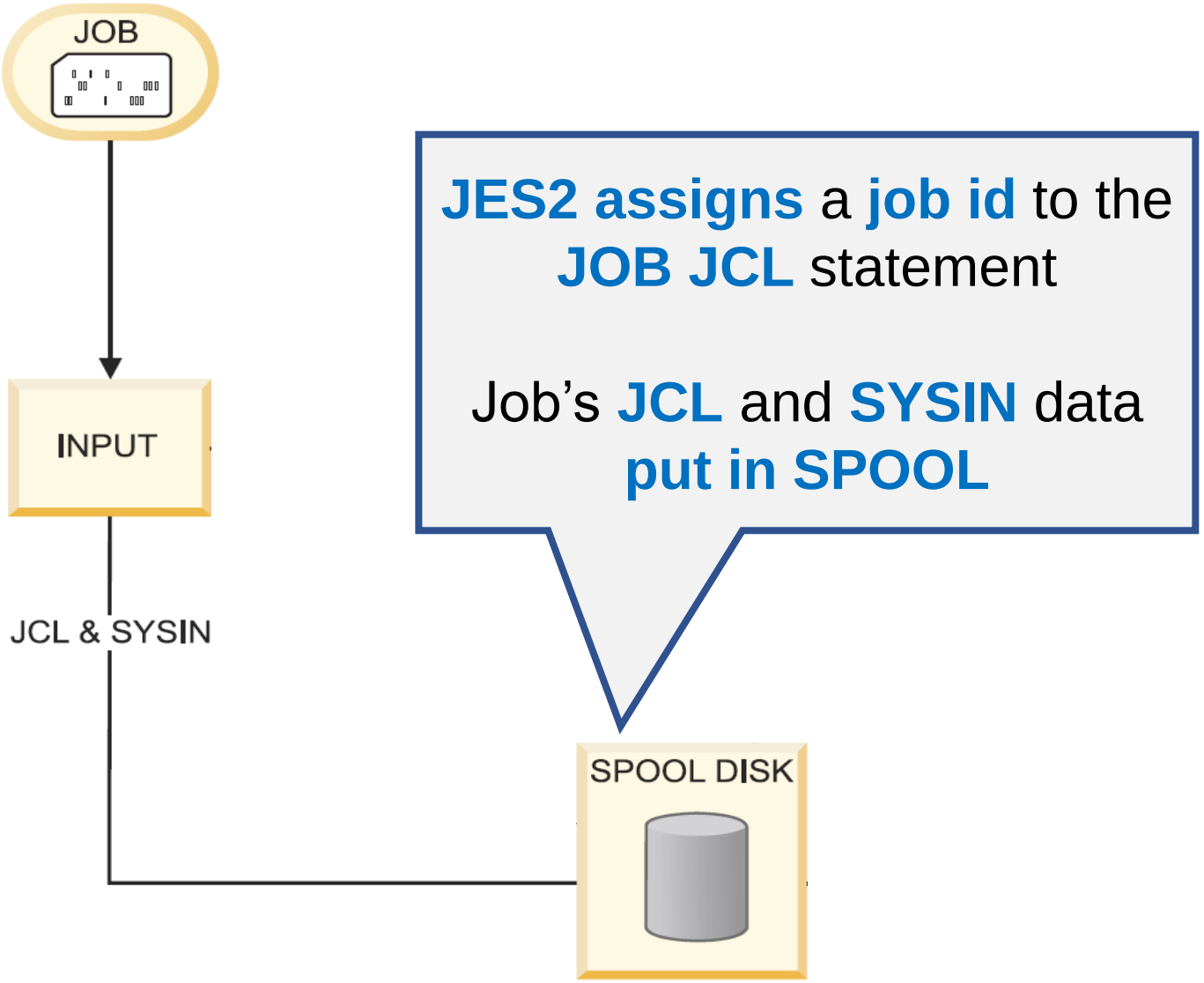
Job Flow

Job Flow

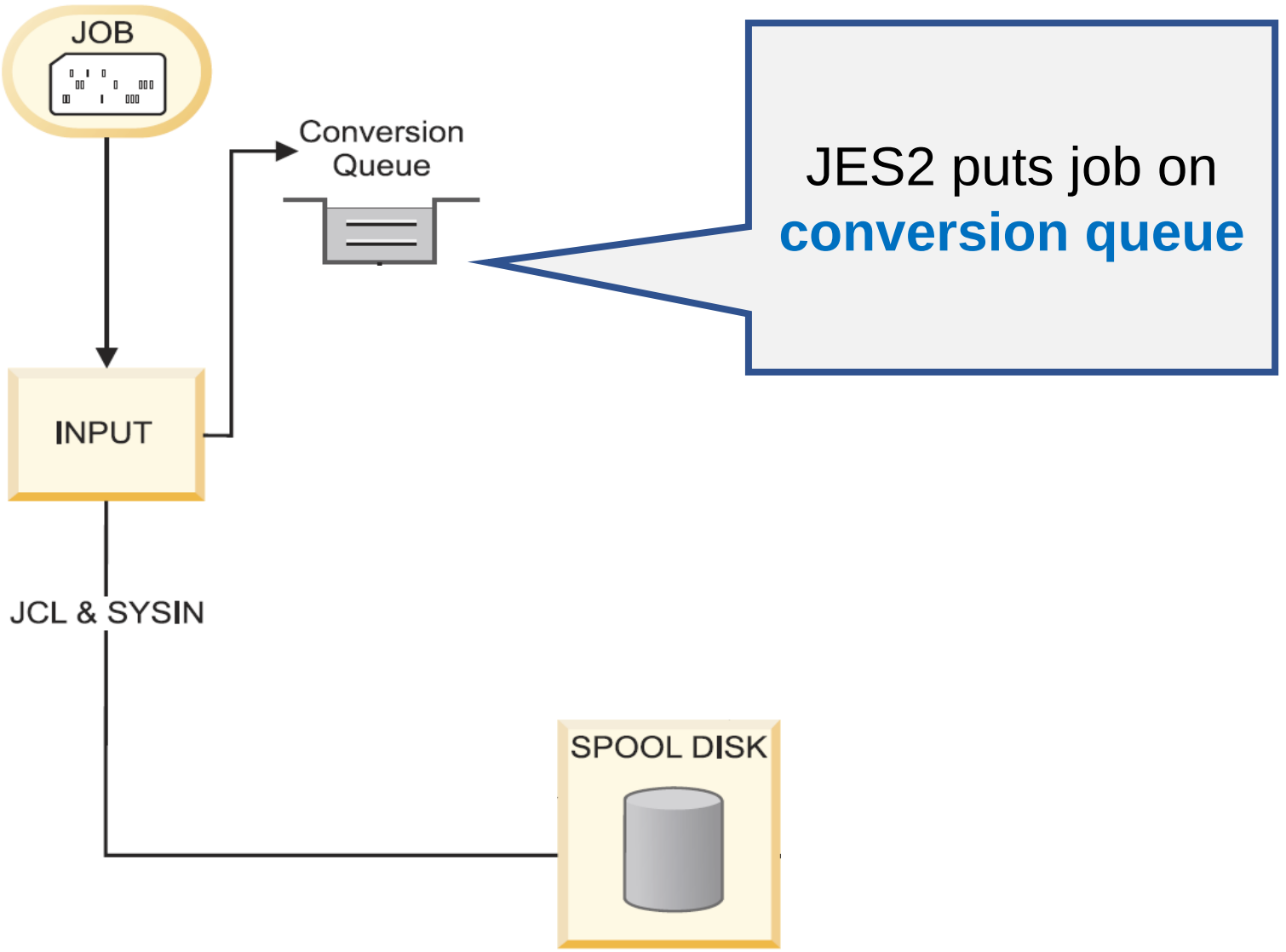
Job Flow



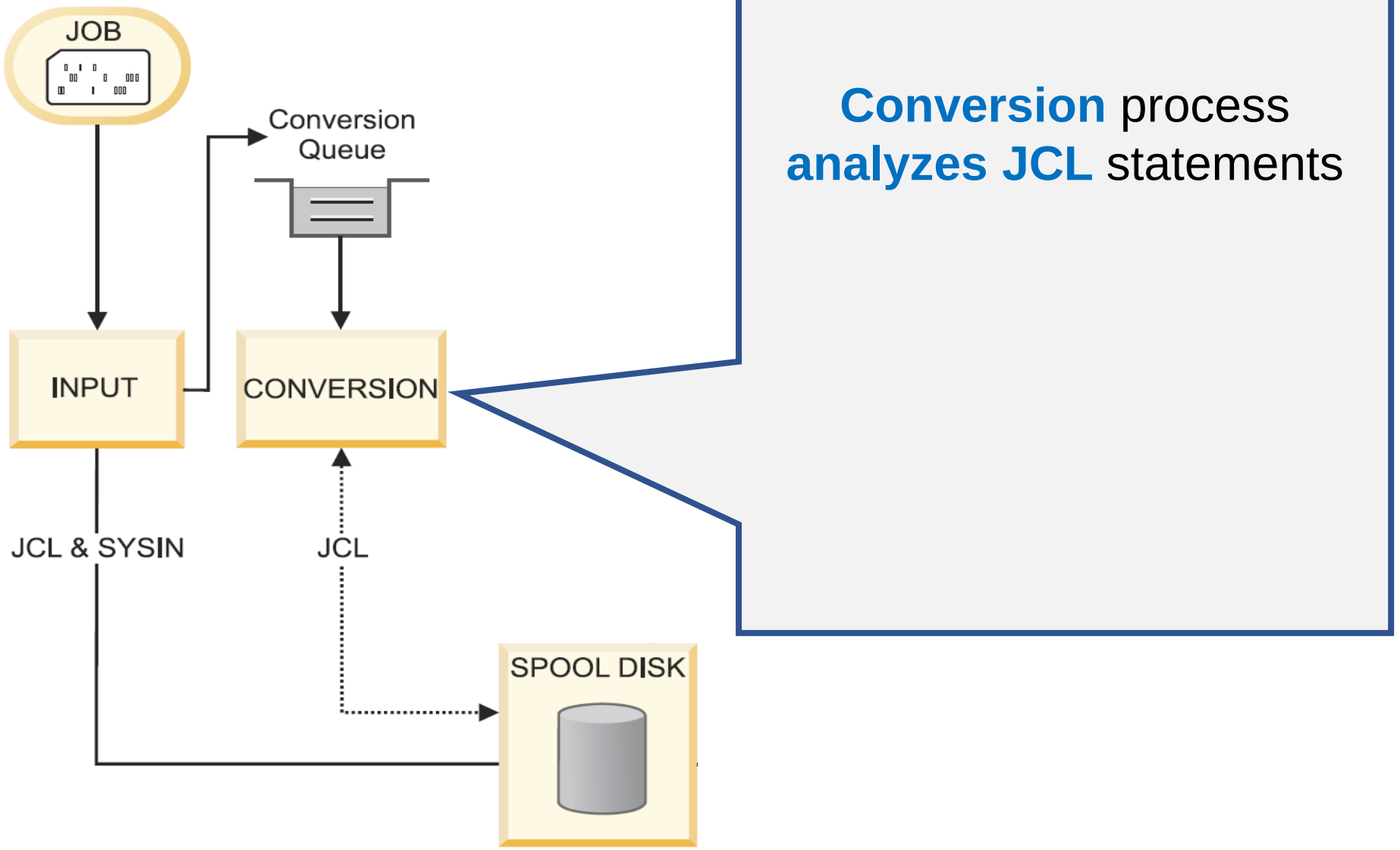
Job Flow


JOB
INPUT
JCL & SYSIN
SPOOL DISK
JES2 assigns a job id to the JOB JCL statement
Job's JCL and SYSIN data put in SPOOL

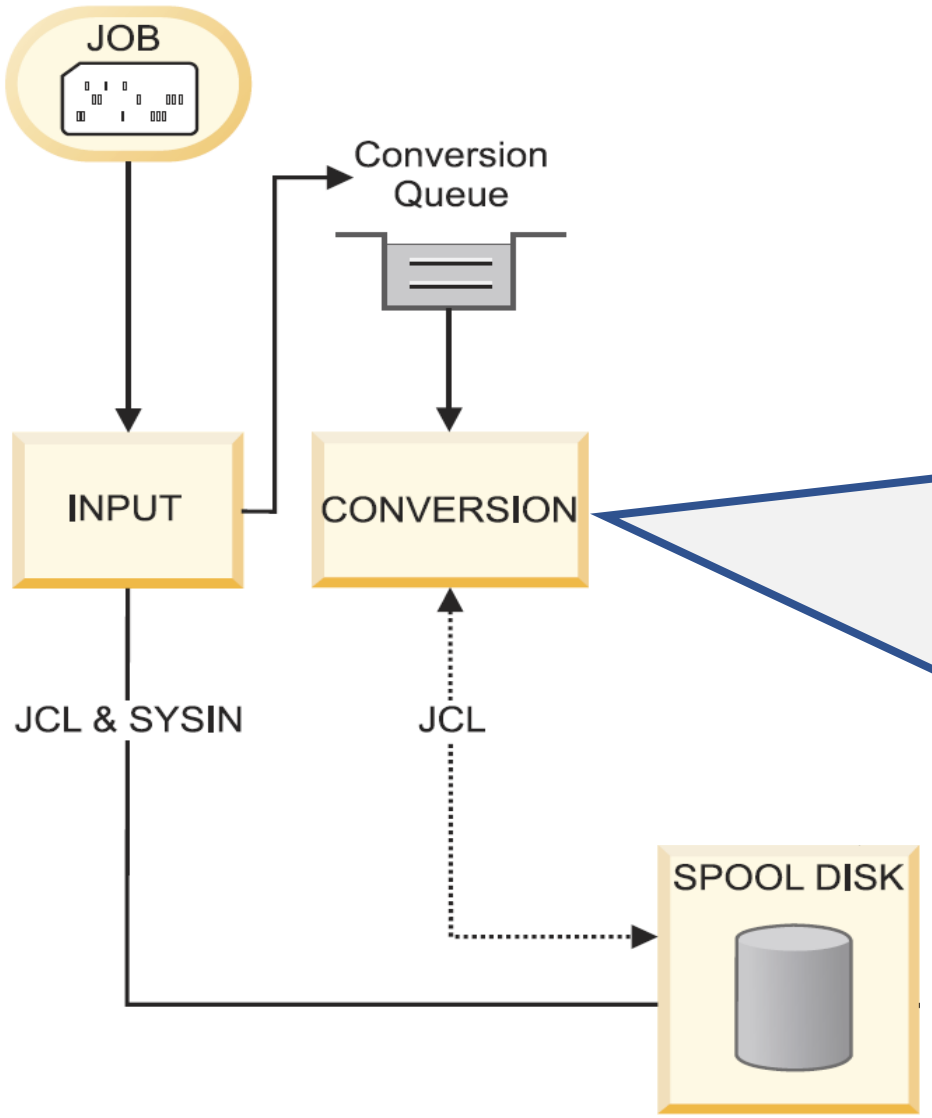
Job Flow


JOB
Conversion Queue
JES2 puts job on conversion queue
INPUT
JCL & SYSIN
SPOOL DISK

Job Flow


JOB
Conversion Queue
INPUT
CONVERSION
JCL & SYSIN
JCL
SPOOL DISK
Conversion process analyzes JCL statements

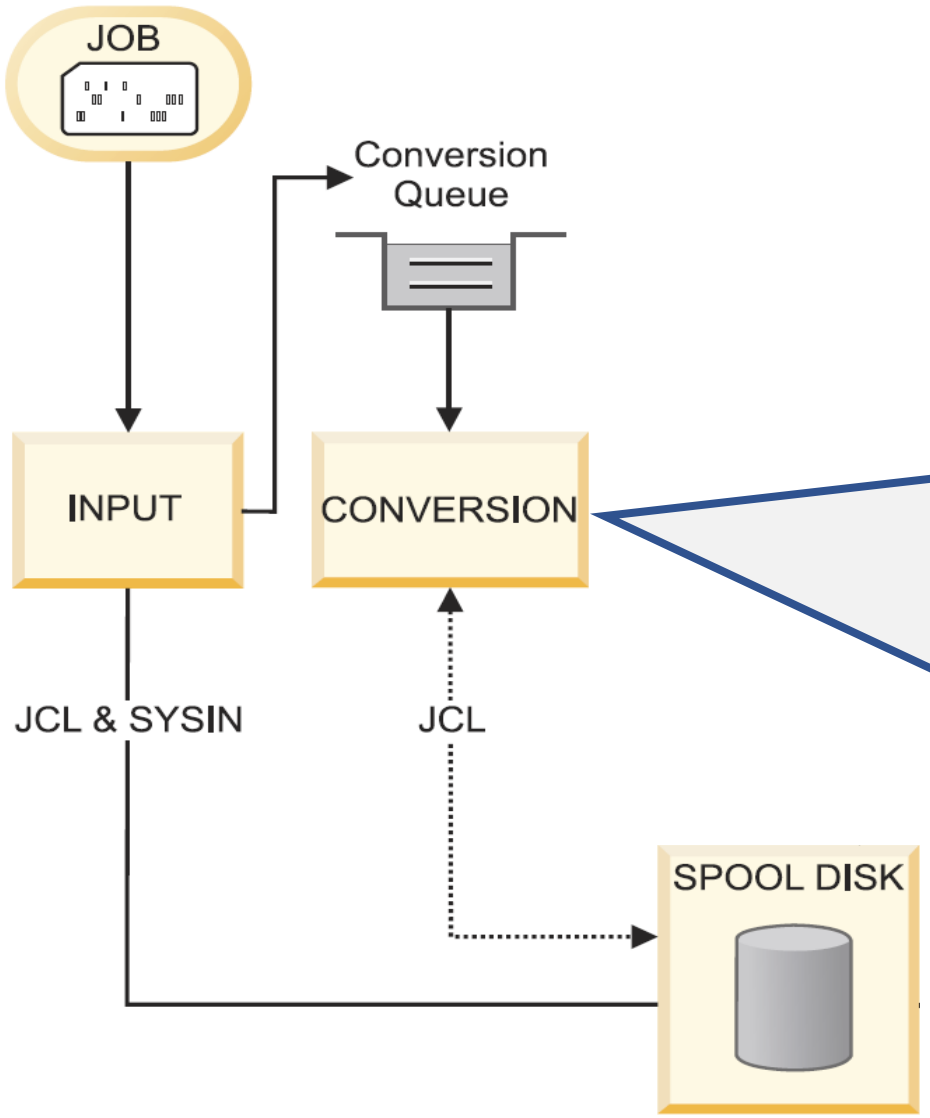
Job Flow



Conversion process **analyzes JCL** statements

JCL **converted** into **converter/interpreter** text

Job Flow

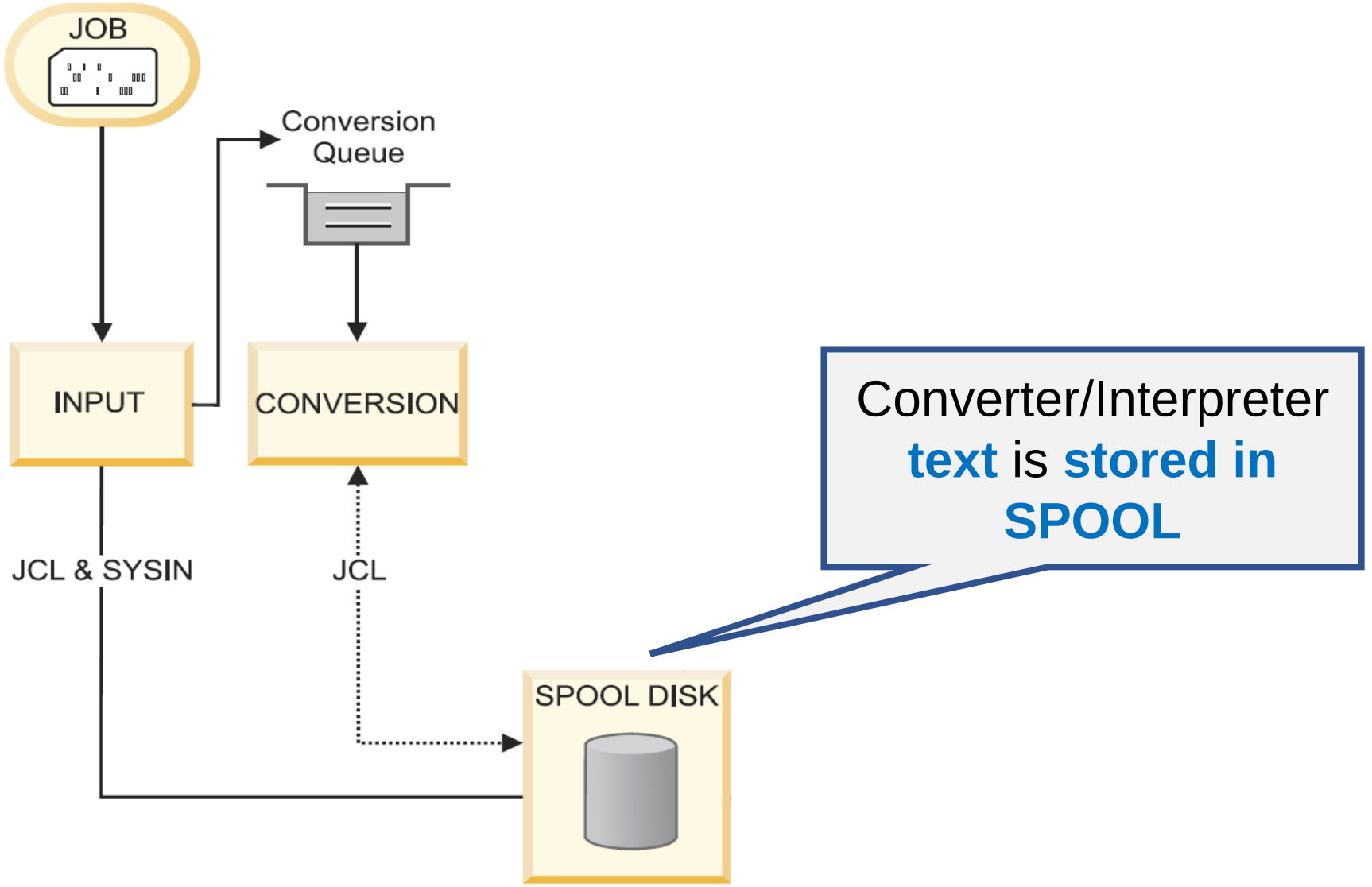


Conversion process **analyzes JCL** statements

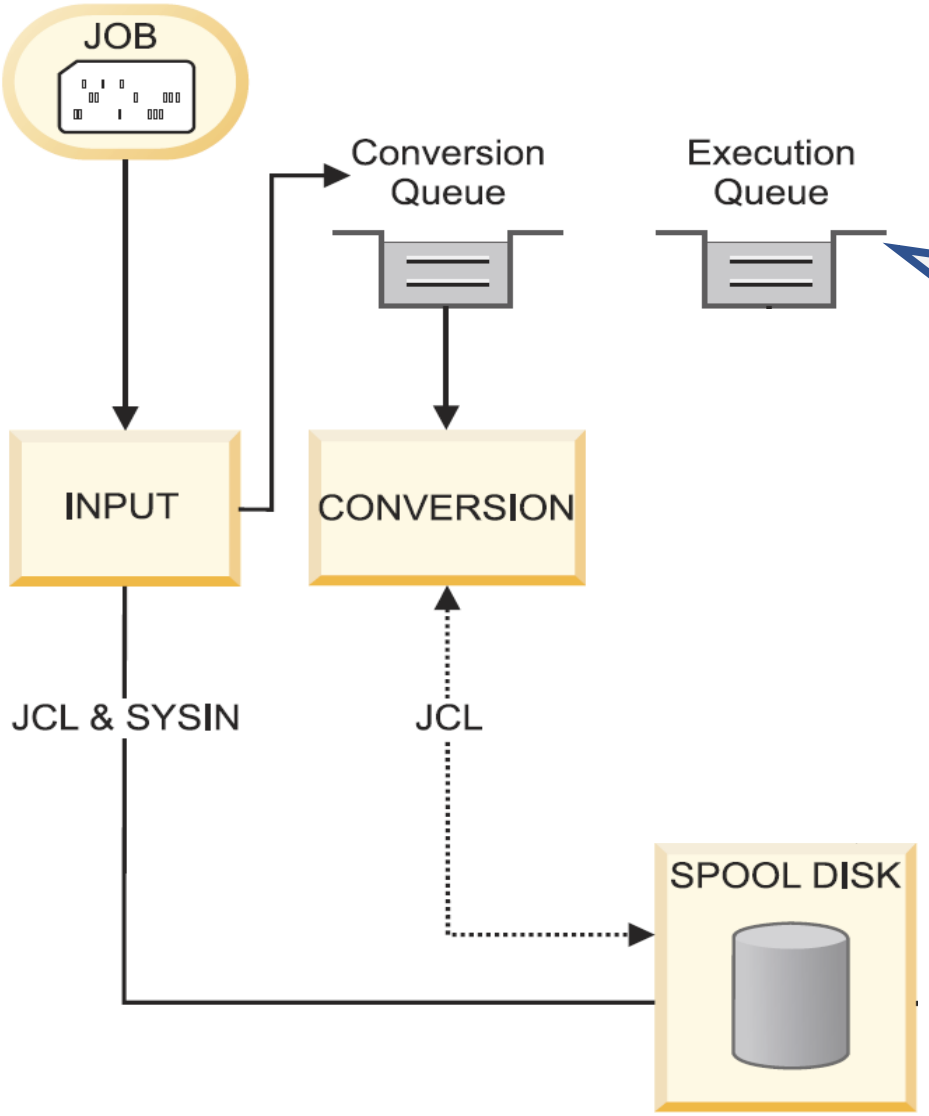
JCL **converted** into **converter/interpreter** text

Both **JES and Initiator understand** this **text**

Job Flow

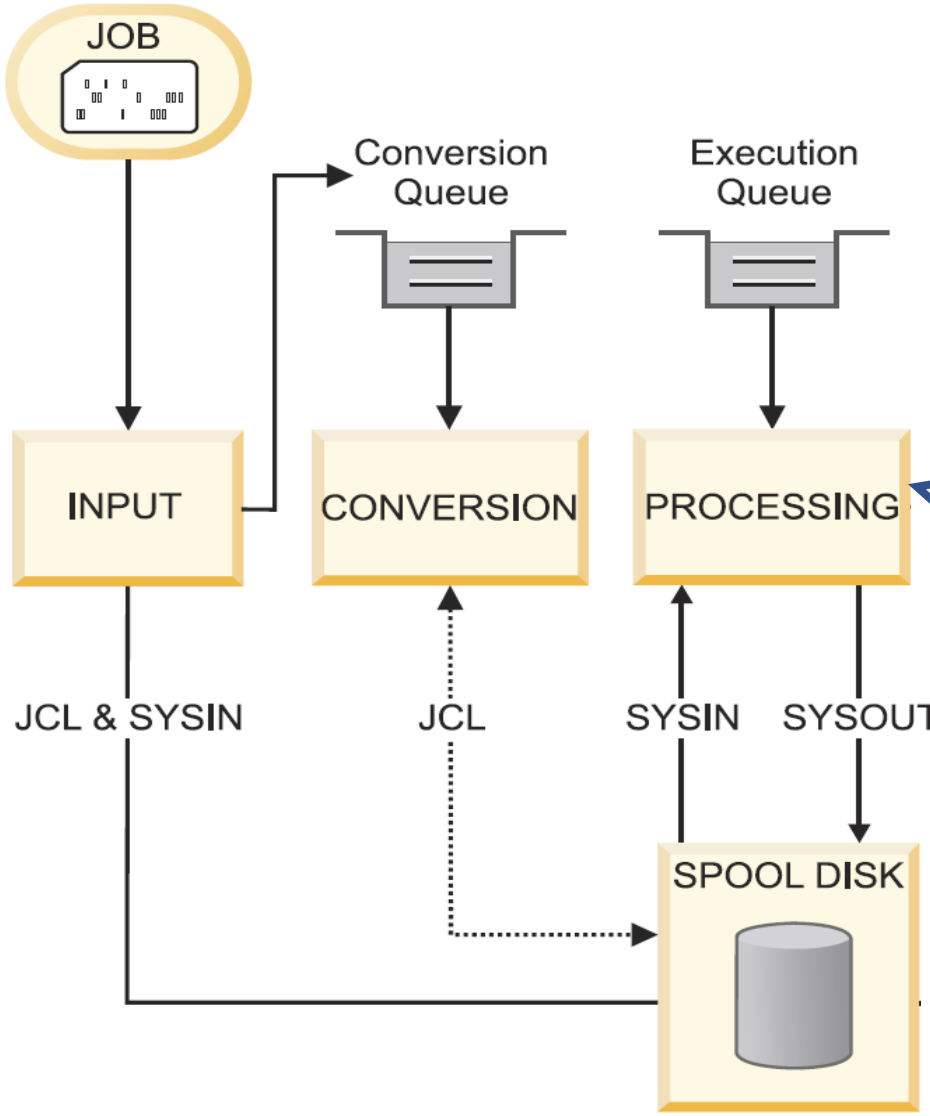

JOB
Conversion Queue
INPUT
CONVERSION
JCL & SYSIN
JCL
SPOOL DISK
Converter/Interpreter text is stored in SPOOL

Job Flow



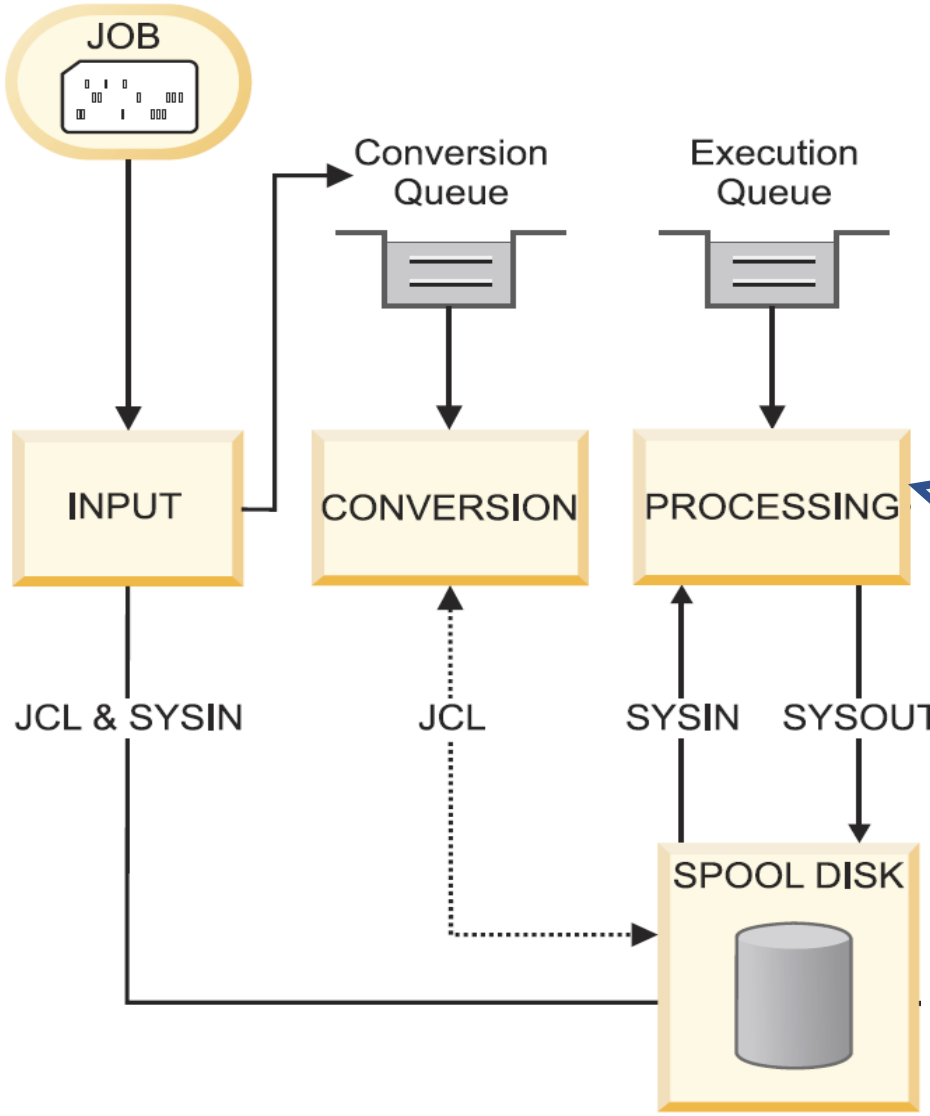
Initiator looking for work **asks JES2** to **put a job** on the **execution queue**

Job Flow



Initiator gets the converter/interpreter **text** from **SPOOL**

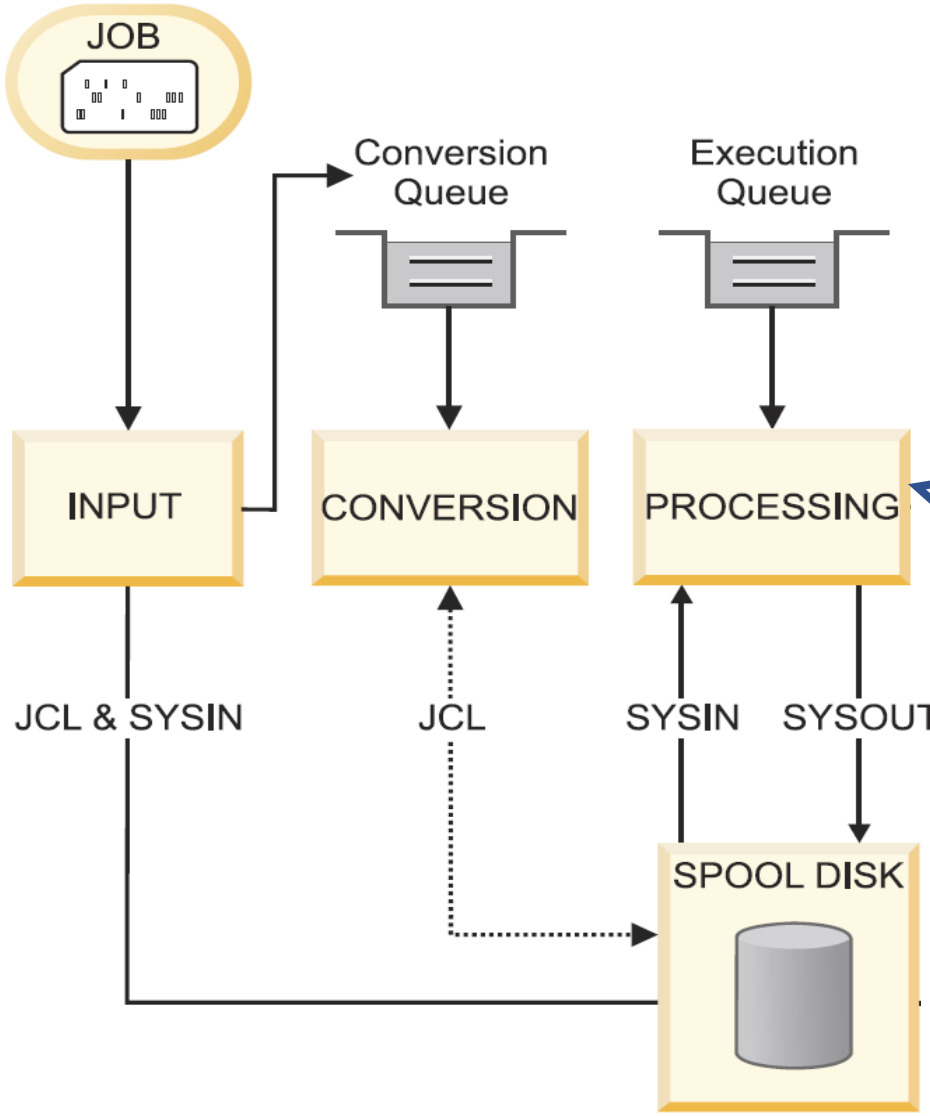
Job Flow



Initiator gets the converter/interpreter **text** from **SPOOL**

Allocates resources requested in the JCL

Job Flow

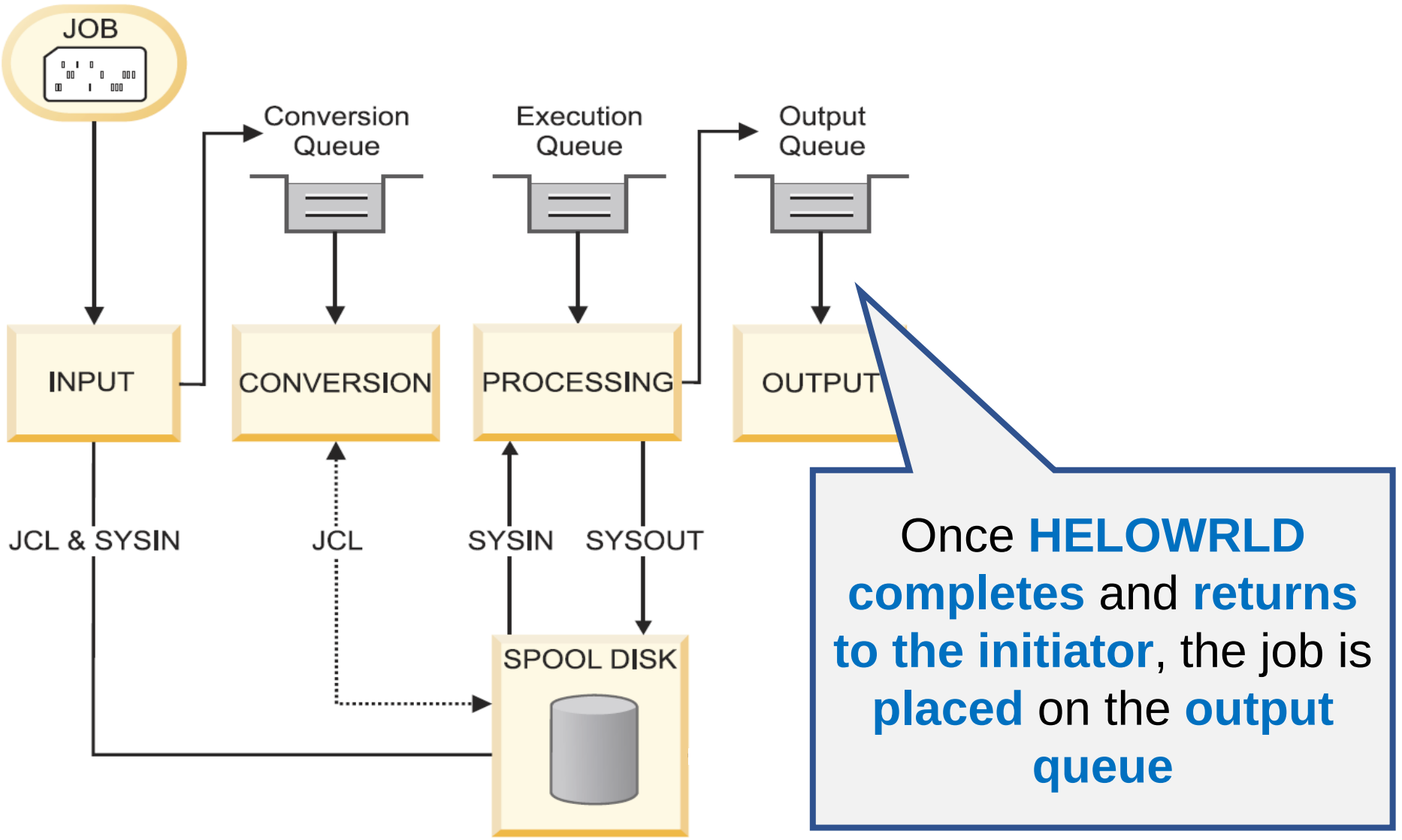


Initiator gets the converter/interpreter **text** from **SPOOL**

Allocates resources requested in the JCL

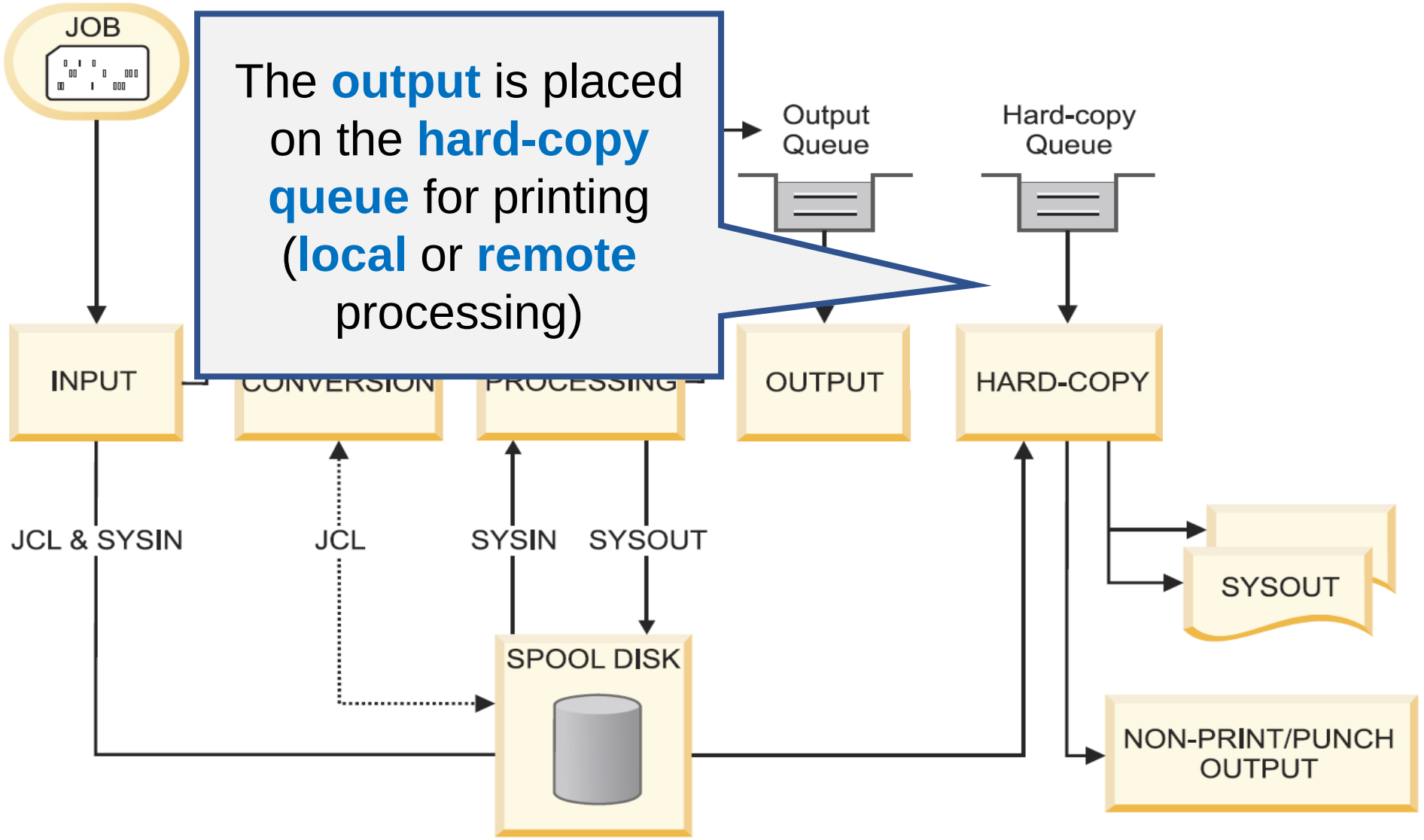
Passes control to the HELOWRLD program

Job Flow

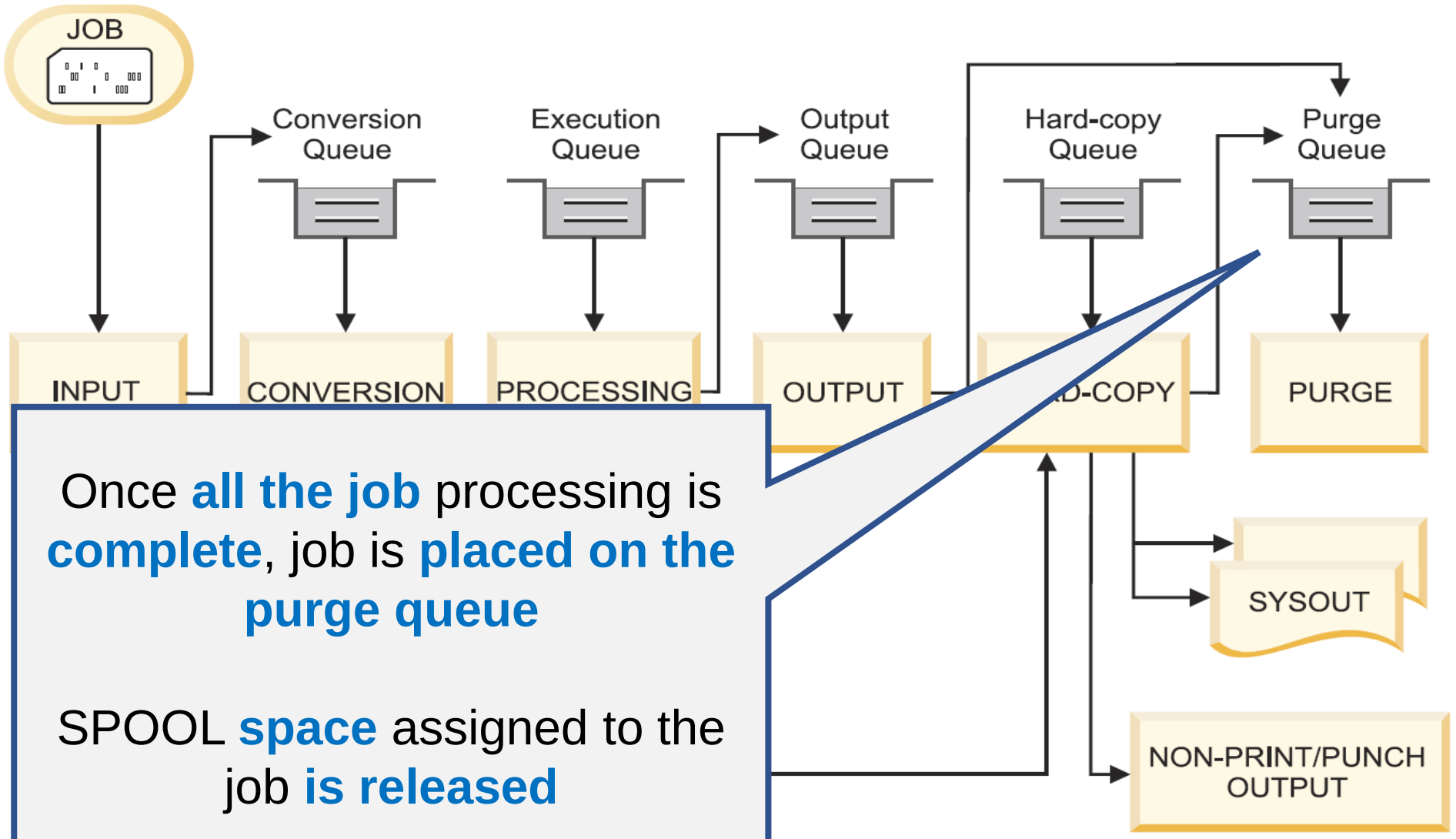


Once **HELOWRLD completes** and **returns to the initiator**, the job is **placed** on the **output queue**

Job Flow


JOB
The output is placed on the hard-copy queue for printing (local or remote processing)
Output Queue
Hard-copy Queue
INPUT
OUTPUT
HARD-COPY
JCL & SYSIN
JCL
SYSIN
SYSOUT
SYSOUT
SPOOL DISK
NON-PRINT/PUNCH OUTPUT

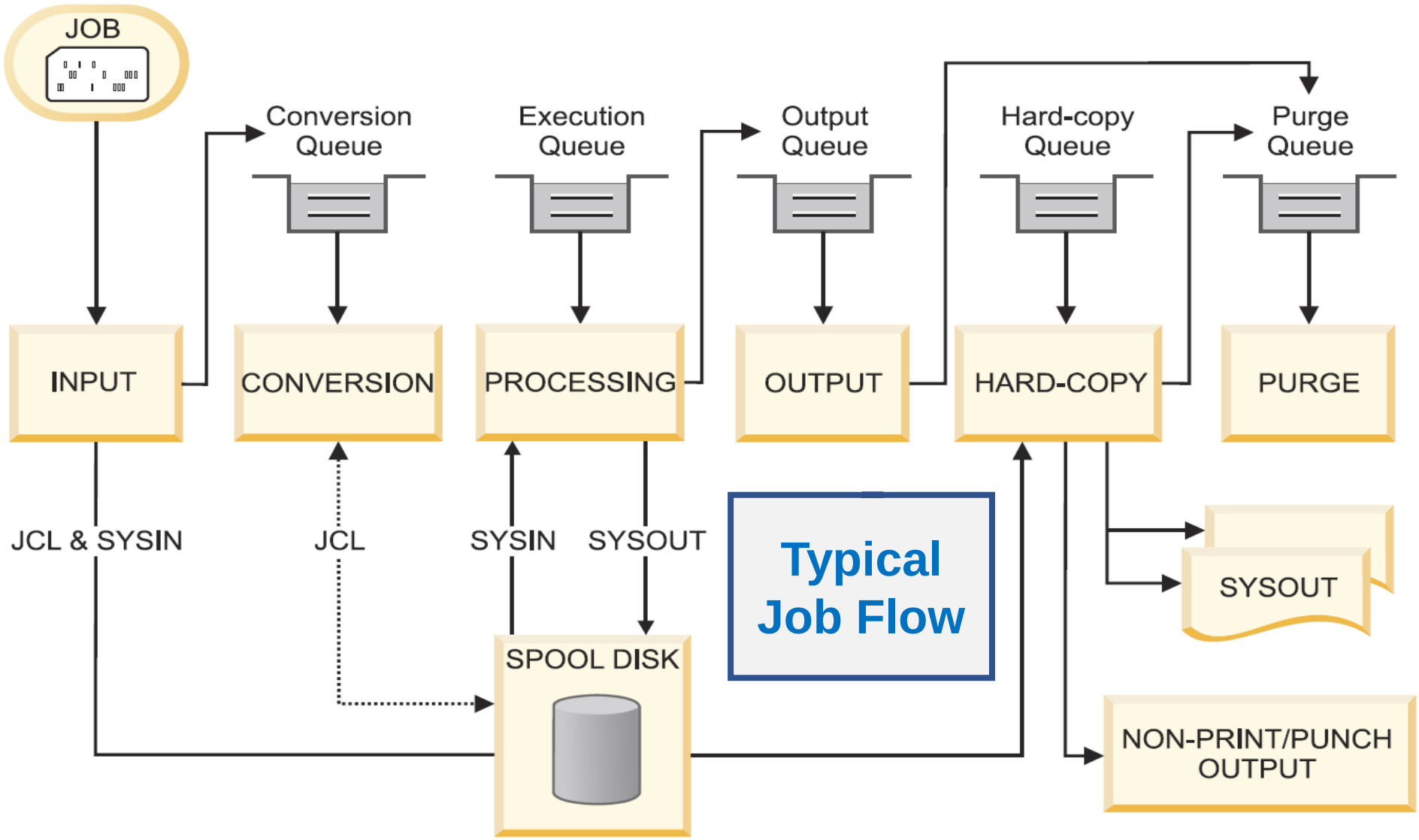
Job Flow



Once **all the job** processing is **complete**, job is **placed on the purge queue**

SPOOL **space** assigned to the job **is released**

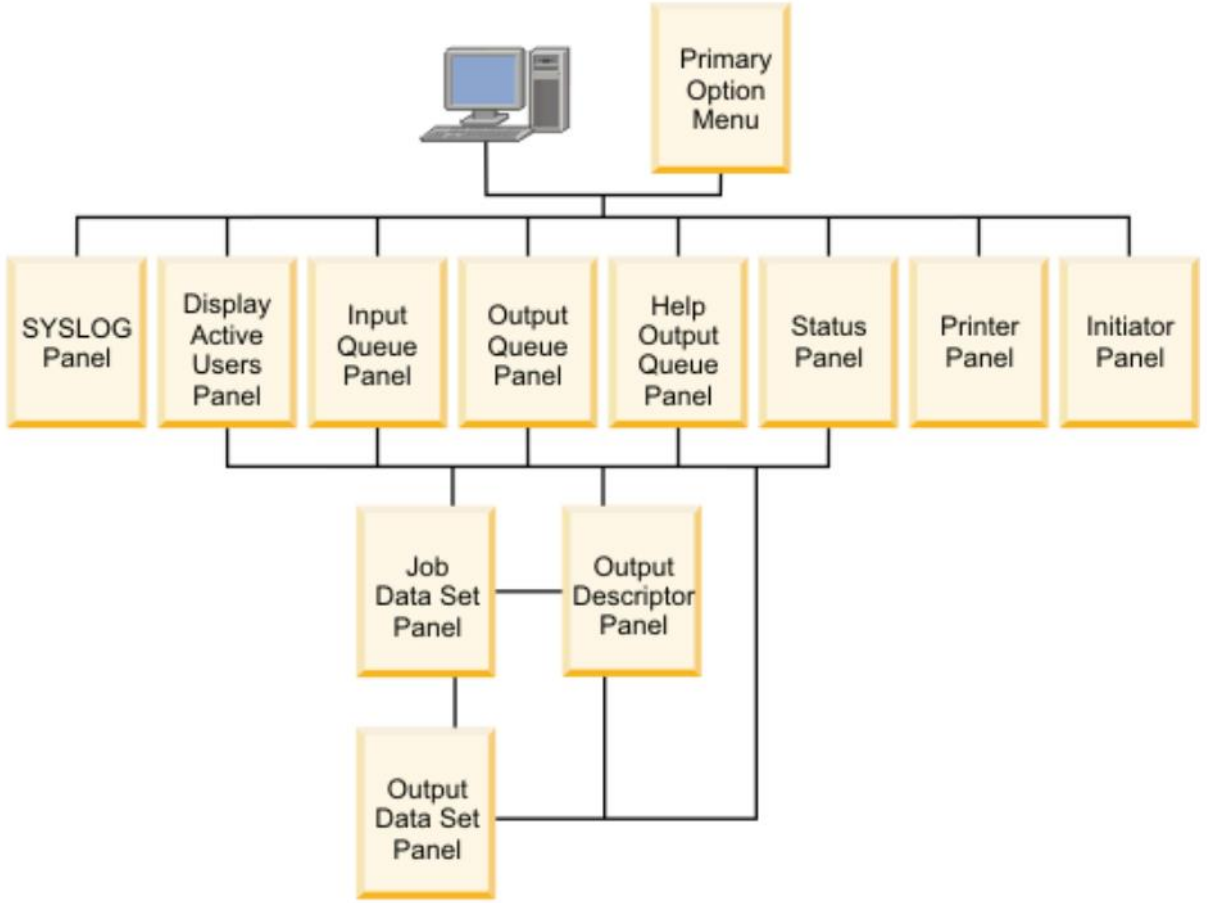
Job Flow


JOB
Conversion Queue
Execution Queue
Output Queue
Hard-copy Queue
Purge Queue
INPUT
CONVERSION
PROCESSING
OUTPUT
HARD-COPY
PURGE
JCL & SYSIN
JCL
SYSIN
SYSOUT
SPOOL DISK
Typical Job Flow
SYSOUT
NON-PRINT/PUNCH OUTPUT

SDSF

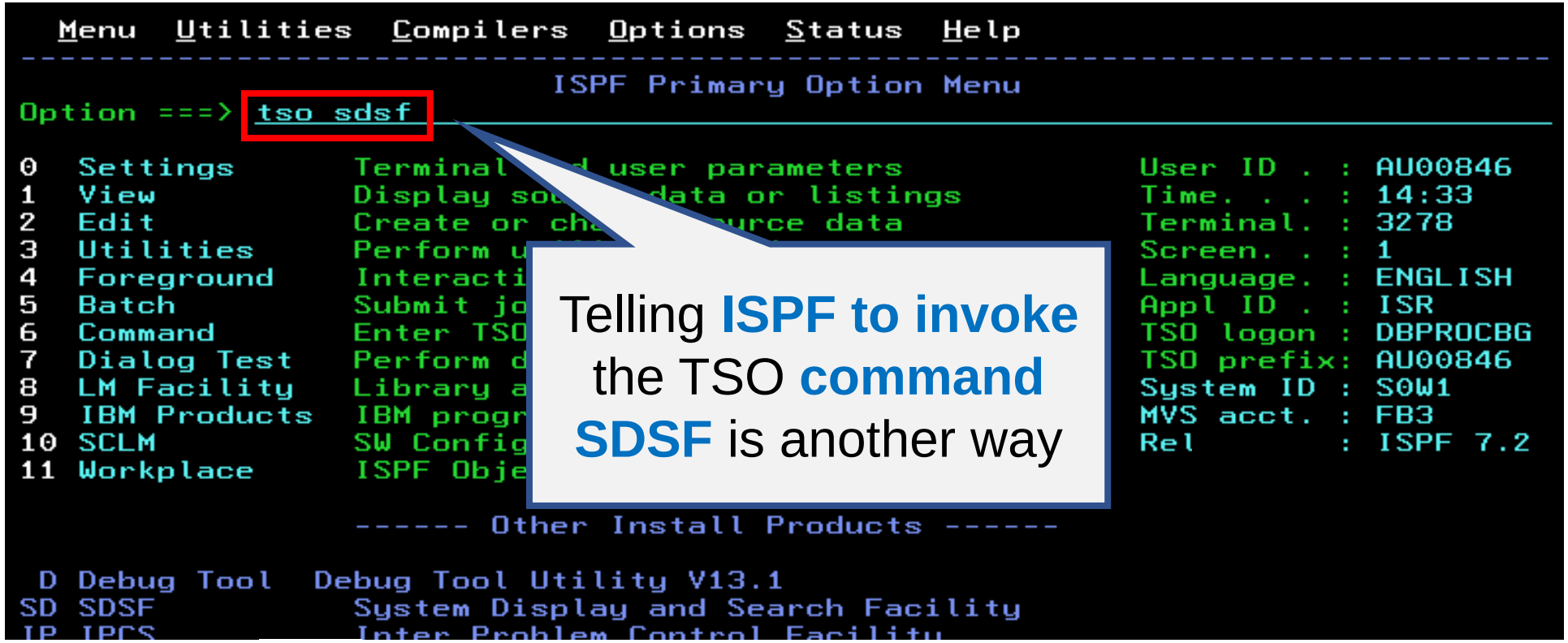
Batch Processing – SDSF

- **SDSF** runs as an ISPF-like application
  - Provides a **hierarchy** of online **panels**

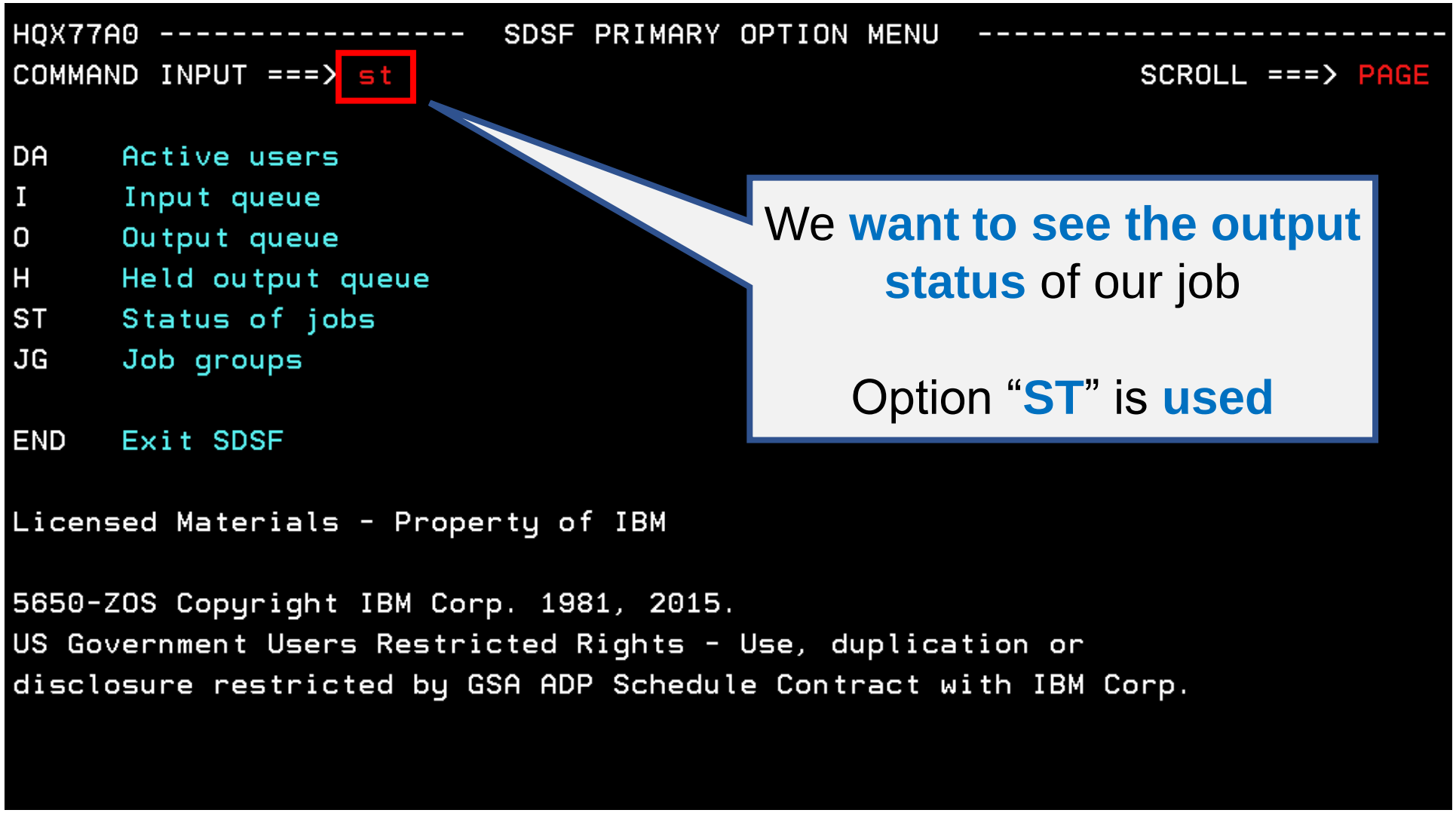


Batch Processing – SDSF

- SDSF can be invoked in different ways
  - Depends on the customer



Batch Processing – SDSF



We **want to see the output status** of our job

Option "**ST**" is **used**

Batch Processing – SDSF

```
SDSF STATUS DISPLAY ALL CLASSES                          LINE 1-2 (2)
COMMAND INPUT ===>                                         SCROLL ===> PAGE
NP    JOBNAME  JobID    Owner    Prty Queue      C  Pos  SAff  ASys Status
      BIBOLET  TSU01779 BIBOLET    15 EXECUTION           AOFT  AOFT
      BIBOLETC JOB01828 BIBOLET     1 PRINT      J 22200
```

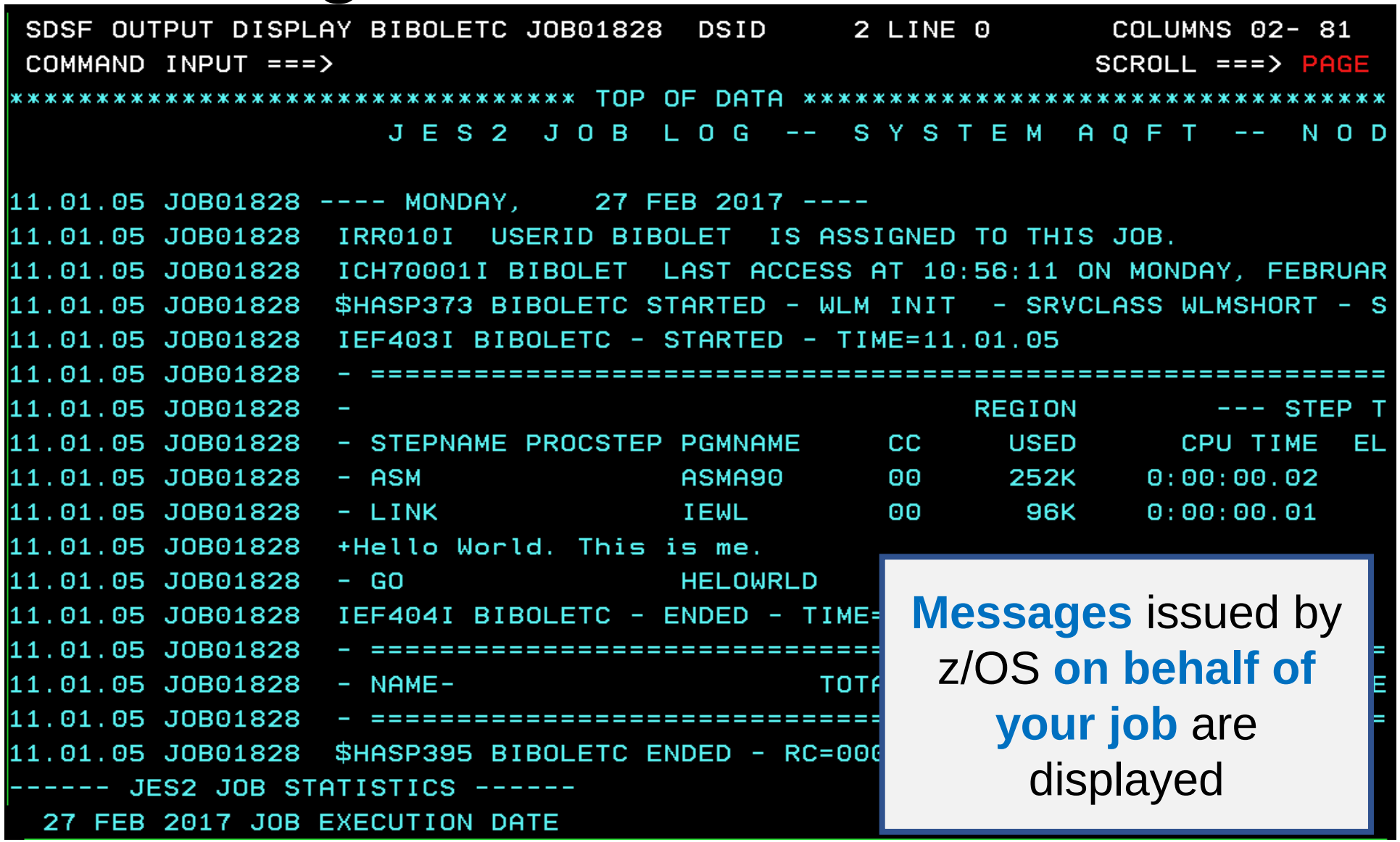
This is the job
we just ran

Batch Processing – SDSF

```
SDSF STATUS DISPLAY ALL CLASSES                          LINE 1-2 (2)
COMMAND INPUT ===>                                       SCROLL ===> PAGE
NP   JOBNAME  JobID    Owner    Prty Queue      C  Pos  SAff  ASys Status
     BIBOLET  TSU01779 BIBOLET    15 EXECUTION           AQFT  AQFT
s    BIBOLETC JOB01828 BIBOLET     1 PRINT      J 22200
```

To see the output, type "**S**" for **select**

Batch Processing – SDSF



Messages issued by z/OS **on behalf of your job** are displayed

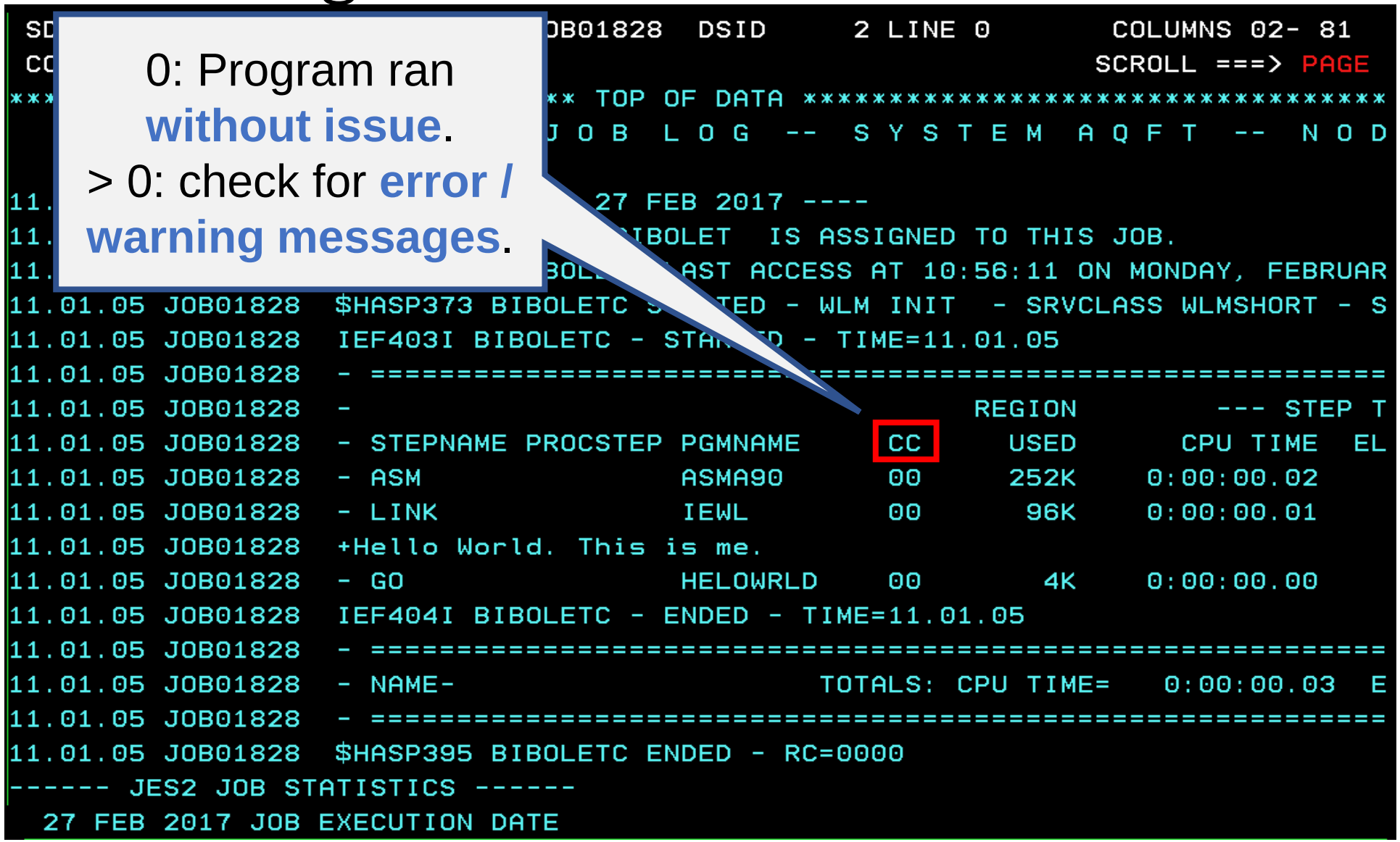
Batch Processing – SDSF

Condition Code:
How did the program run?

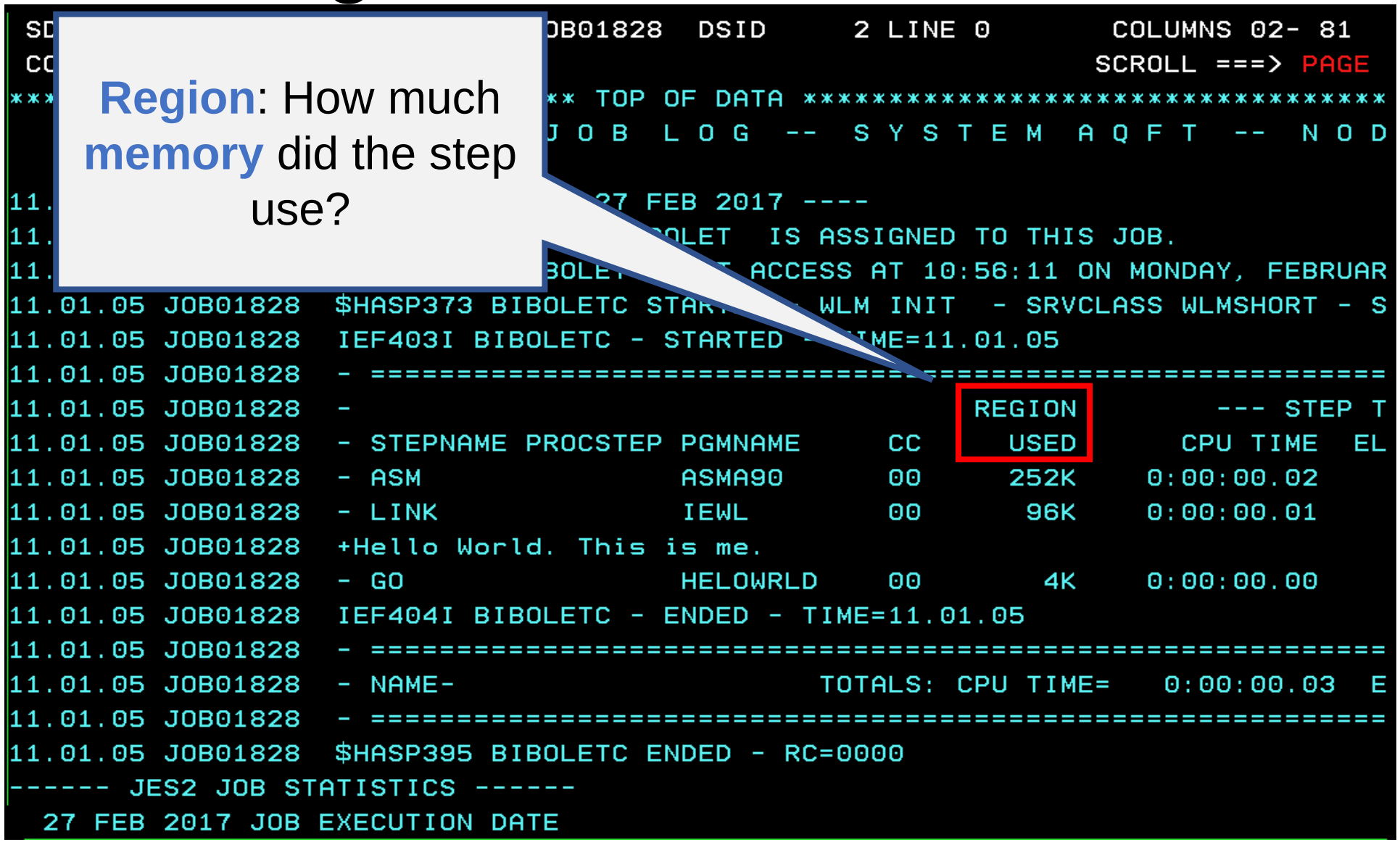
```
 SD                                OB01828  DSID     2 LINE 0       COLUMNS 02- 81
 CC                                                               SCROLL ===> PAGE
***                                ** TOP OF DATA ********************************
                                   J O B  L O G  --  S Y S T E M  A Q F T  --  N O D

11.                                    27 FEB 2017 ----
11.                                    IBOLET  IS ASSIGNED TO THIS JOB.
11.                                    AST ACCESS AT 10:56:11 ON MONDAY, FEBRUAR
11.01.05 JOB01828  $HASP373 BIBOLETC S   TED - WLM INIT  - SRVCLASS WLMSHORT - S
11.01.05 JOB01828  IEF403I BIBOLETC - STA   D - TIME=11.01.05
11.01.05 JOB01828  - ------------------------------------------------------------
11.01.05 JOB01828  -                                        REGION        --- STEP T
11.01.05 JOB01828  - STEPNAME PROCSTEP PGMNAME     CC      USED       CPU TIME  EL
11.01.05 JOB01828  - ASM               ASMA90      00      252K    0:00:00.02
11.01.05 JOB01828  - LINK              IEWL        00       96K    0:00:00.01
11.01.05 JOB01828  +Hello World. This is me.
11.01.05 JOB01828  - GO                HELOWRLD    00        4K    0:00:00.00
11.01.05 JOB01828  IEF404I BIBOLETC - ENDED - TIME=11.01.05
11.01.05 JOB01828  - ------------------------------------------------------------
11.01.05 JOB01828  - NAME-                        TOTALS: CPU TIME=   0:00:00.03  E
11.01.05 JOB01828  - ------------------------------------------------------------
11.01.05 JOB01828  $HASP395 BIBOLETC ENDED - RC=0000
------ JES2 JOB STATISTICS ------
  27 FEB 2017 JOB EXECUTION DATE
```

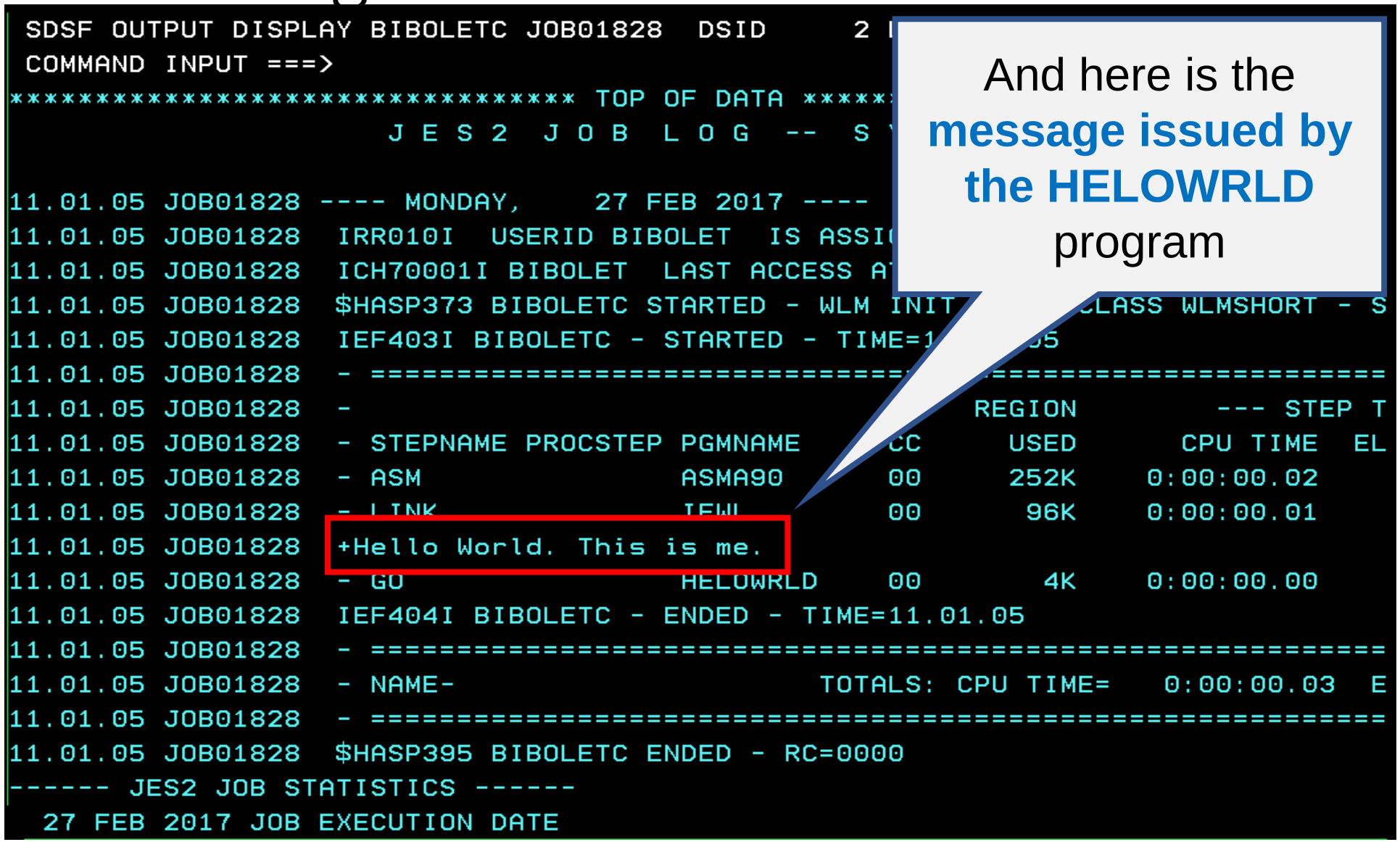
Batch Processing – SDSF



Batch Processing – SDSF



Batch Processing – SDSF



And here is the **message issued by the HELOWRLD** program

System Log

z/OS Concepts – SYSLOG

- What is the **System Log**?
  - Better known as **SYSLOG**
    - a.k.a. **Hardcopy Log** (at one time this was a **printer**)

z/OS Concepts – SYSLOG

- What is the **System Log**?
  - Better known as **SYSLOG**
    - a.k.a. **Hardcopy Log** (at one time this was a **printer**)
  - **Chronological** listing of **messages** about z/OS **system activity** and other major middleware software products

z/OS Concepts – SYSLOG

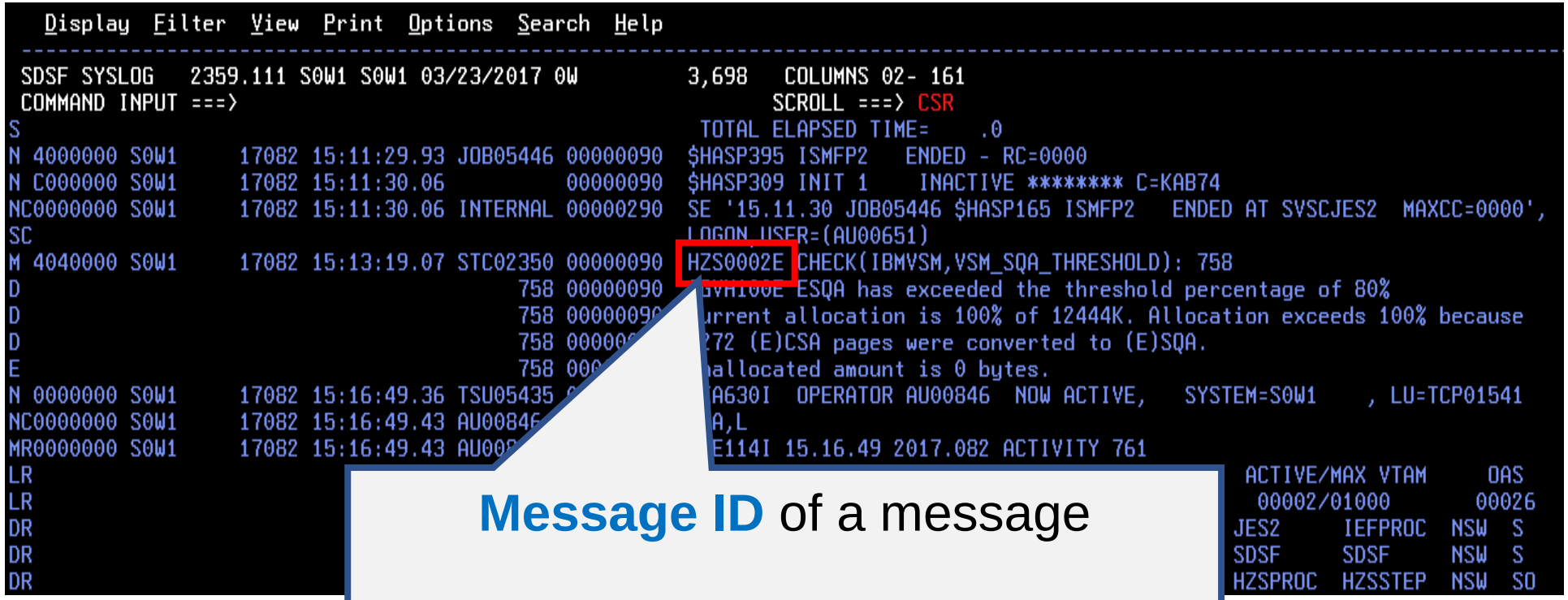
- What is the **System Log**?
  - Better known as **SYSLOG**
    - a.k.a. **Hardcopy Log** (at one time this was a **printer**)
  - **Chronological** listing of **messages** about z/OS **system activity** and other major middleware software products
  - Issued **system commands** and their **responses**

z/OS Concepts – SYSLOG

- What is the **System Log**?
  - Better known as **SYSLOG**
    - a.k.a. **Hardcopy Log** (at one time this was a **printer**)
  - **Chronological** listing of **messages** about z/OS **system activity** and other major middleware software products
  - Issued **system commands** and their **responses**

- When an **unexpected** system **problem** occurs, the SYSLOG is **the first place to look** to gather information about the problem

z/OS Concepts – SYSLOG



Message ID of a message

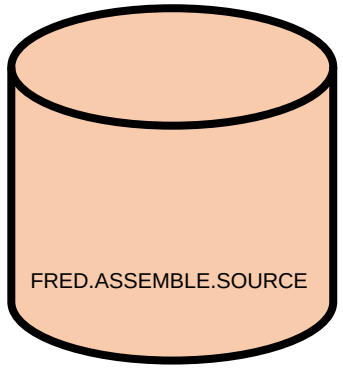
Every **message starts** with an ID (up to 10 characters)

Enables you to **look up message** in documentation

VTOC and Catalogs

IBM Speak – VTOC & Catalogs

Allocate space for data set
FRED.ASSEMBLE.SOURCE
on volume **VOL100**

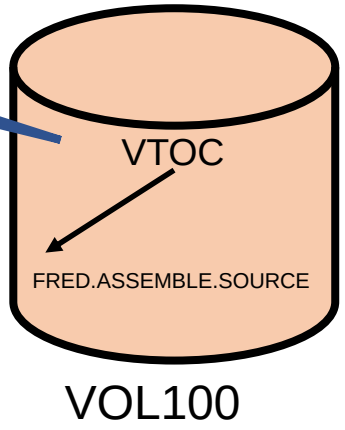


VOL100

VOLSER

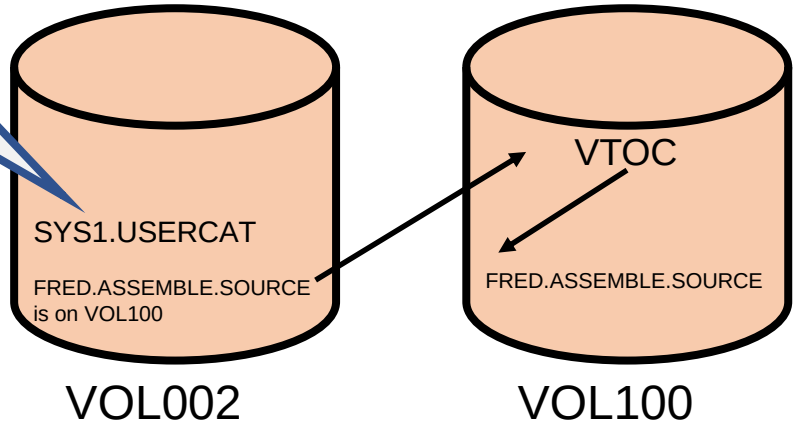
IBM Speak – VTOC & Catalogs

DASD volume has a *Volume Table of Contents* (**VTOC**) that **locates** a **data set** on the **volume**



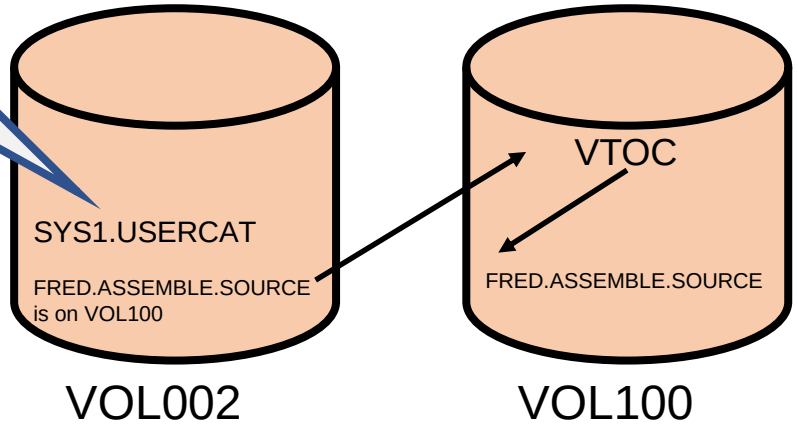
IBM Speak – VTOC & Catalogs

To **simplify find** the data set, the **name and location** information is **placed in a catalog**



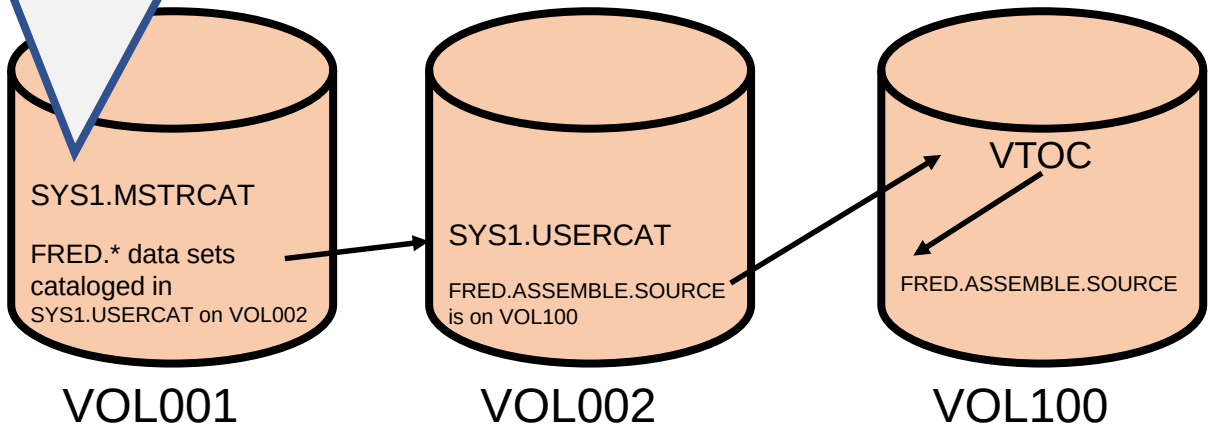
IBM Speak – VTOC & Catalogs

There can be **many catalogs** so **how do we locate** the correct one?



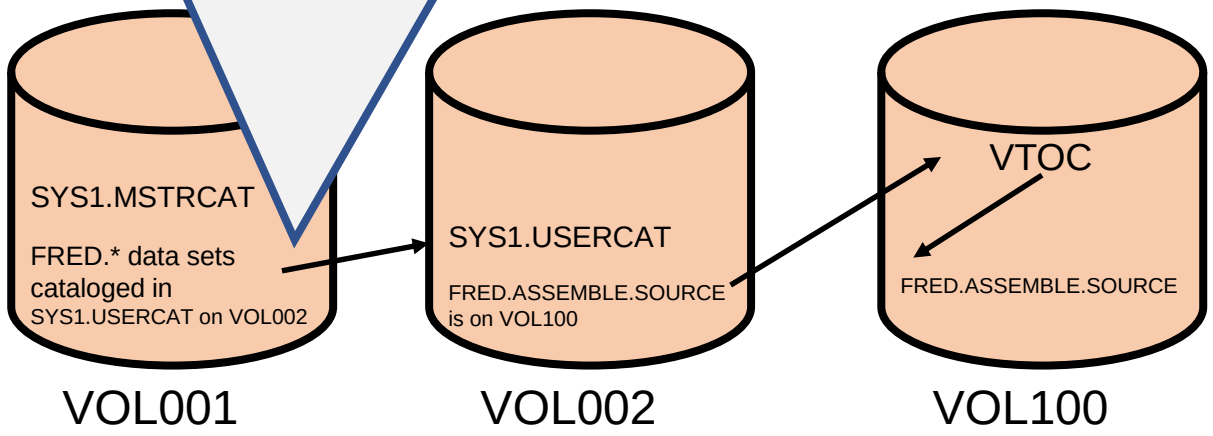
IBM Speak – VTOC & Catalogs

We look in the "**Master**" **catalog** (of which there is **only one**)



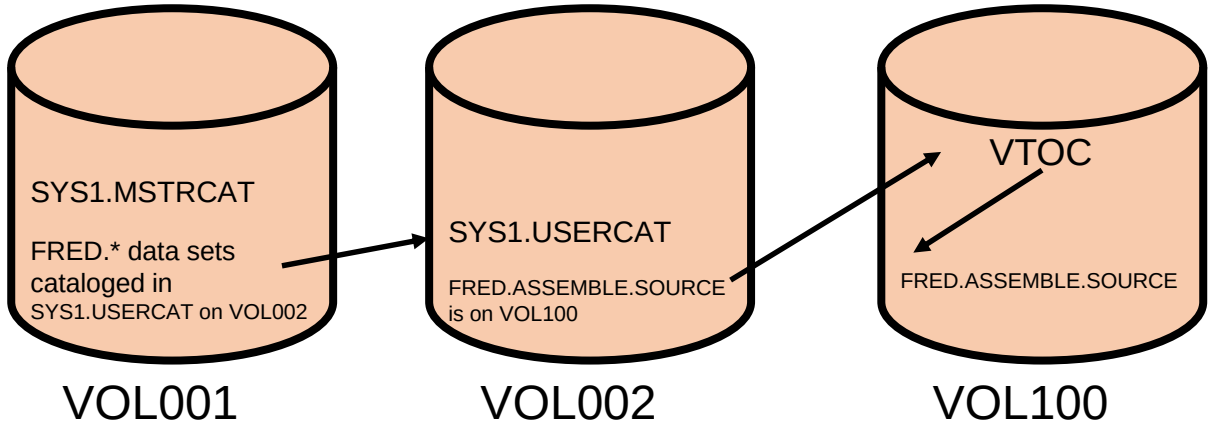
IBM Speak – VTOC & Catalogs

The **master** catalog contains an **HLQ of FRED** and indicates that **all of Fred's data sets** are **cataloged in the user catalog** named SYS1.USERCAT



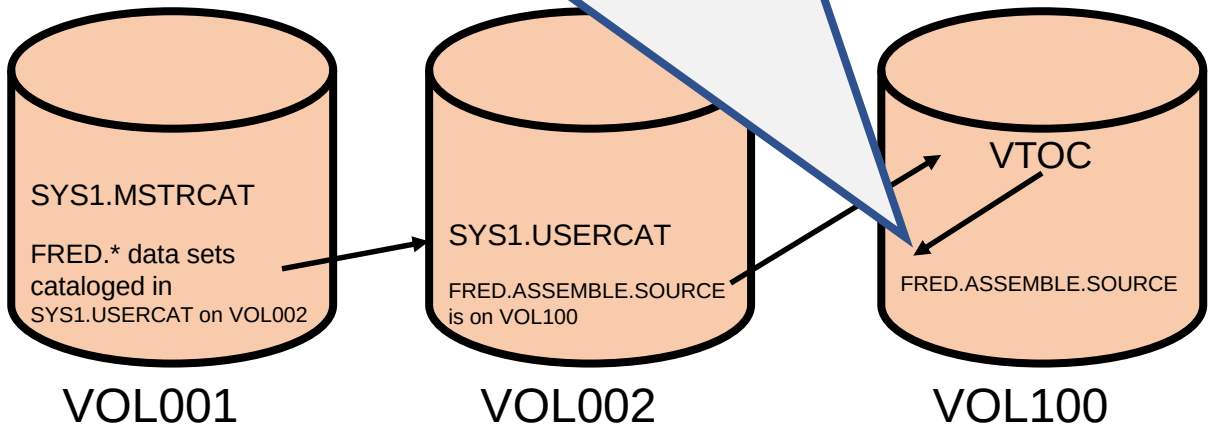
IBM Speak – VTOC & Catalogs

Now we can **tell z/OS to find** FRED.ASSEMBLE.SOURCE **without** giving **any more** location **information**



IBM Speak – VTOC & Catalogs

Accessing the **master** catalog then the **user** catalog then the **VTOC**, z/OS can easily find the **data set**



PDS and PDSE

PDS & PDSE

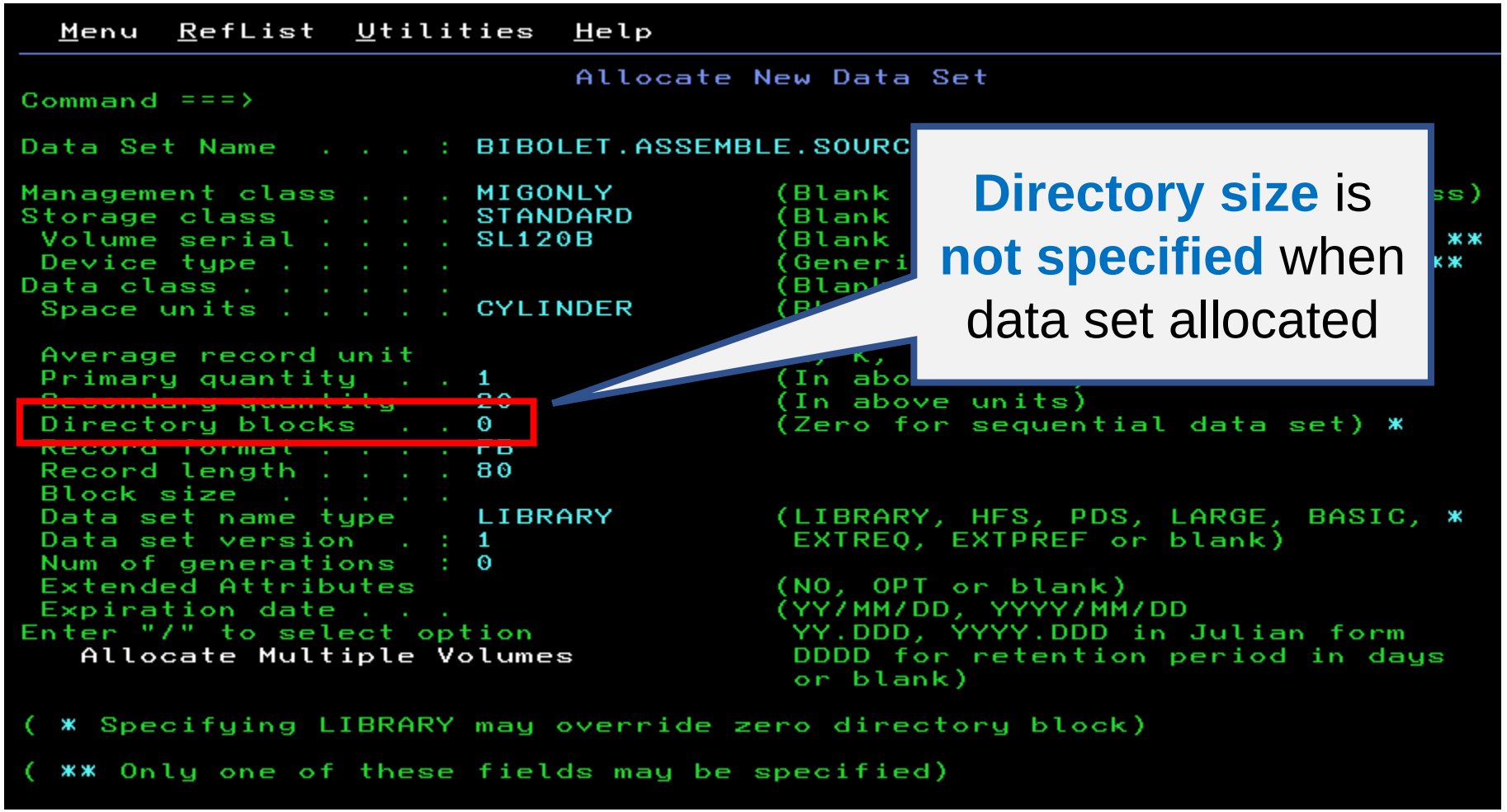
- PDSE data sets **can be used in place** of nearly all PDS data sets

PDS & PDSE

- **PDSE designed** to address running out of space issues
  - **Compression** is **not necessary**
  - PDSE **directory expands** to fit the members

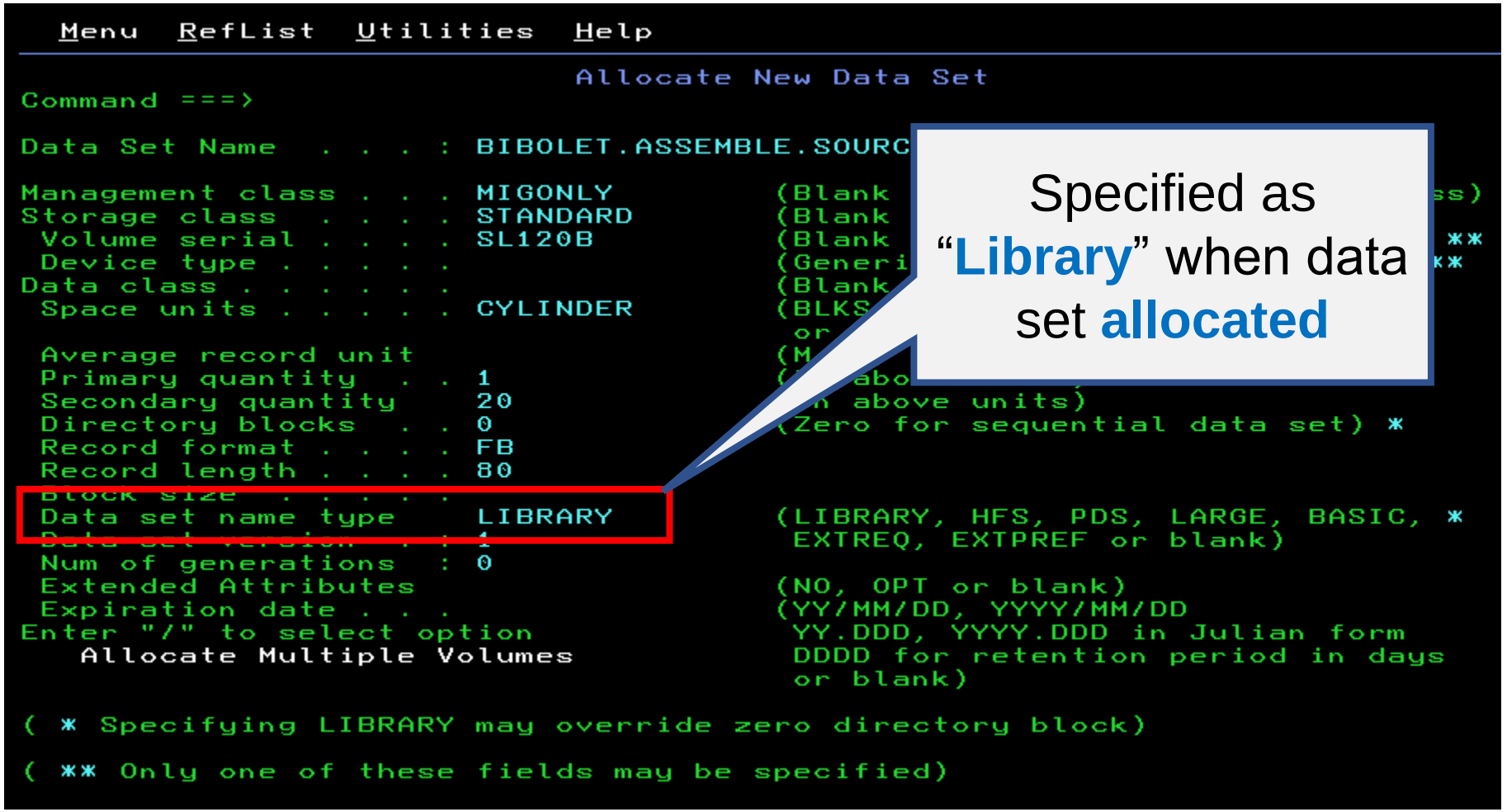
PDS & PDSE

- **PDSE designed** to address these problems



PDS & PDSE

- **PDSE designed** to address these problems



PDS & PDSE

- **PDSE designed** to address these problems
  - **Compression** is **not necessary**
  - PDSE **directory expa**

Recommendation:
Use **PDSE** (a.k.a. Library) data sets **whenever possible**

SMS

z/OS Concepts – SMS

- Data set management is **tedious** & **error prone**

z/OS Concepts – SMS

- Data set management is **tedious** & **error prone**
- IBM introduced **SMS** (*System-Managed-Storage*) to address this concern

z/OS Concepts – SMS

- SMS - **automated approach** to **managing** storage **resources**

z/OS Concepts – SMS

- SMS - **automated approach** to **managing** storage **resources**
- Uses **software** to manage
  - data **security**
  - data **placement**
  - **migration** – move to tape
  - **backup**
  - **recall** – move from tape to DASD
  - **recovery** – restore damaged data set
  - **deletion**
  - **encryption**
  - **compression**
  - **And …**

z/OS Concepts – SMS

- **SMS Goals**:
  - current **data is available** when needed

z/OS Concepts – SMS

- **SMS Goals**:
  - current **data is available** when needed
  - **space** is made **available** for **creating new** data and for **extending current** data

z/OS Concepts – SMS

- **SMS Goals**:
  - current **data is available** when needed
  - **space** is made **available** for **creating new** data and for **extending current** data
  - **obsolete data** is **removed** from storage

IPL and System Address Space Initialization

IBM Speak - IPL

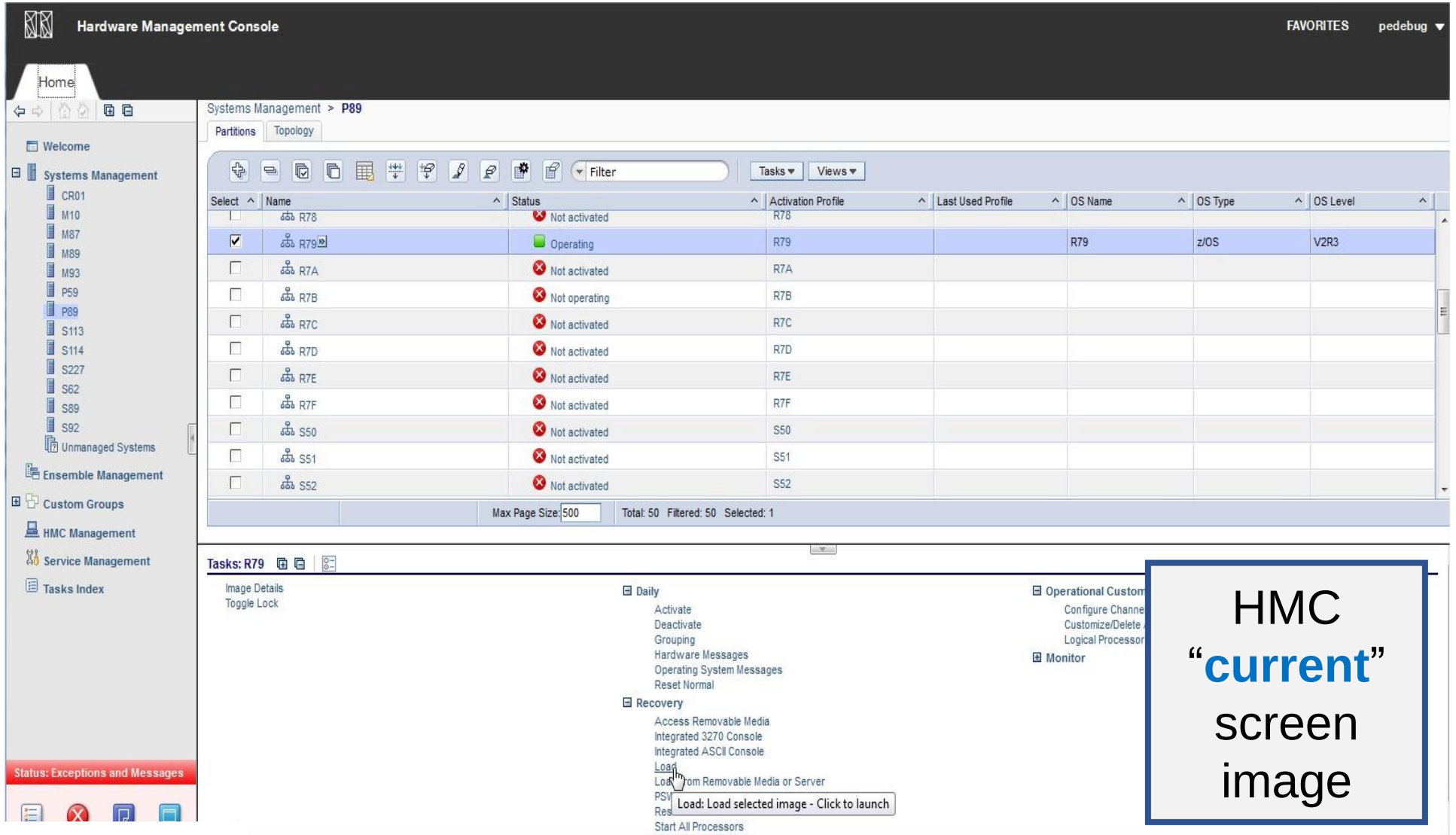
- **IPL** – *Initial Program Load*
  - Think "**Boot the System**"
  - Starting z/OS

IPL

In 2019,
activate the "**Load Profile**"

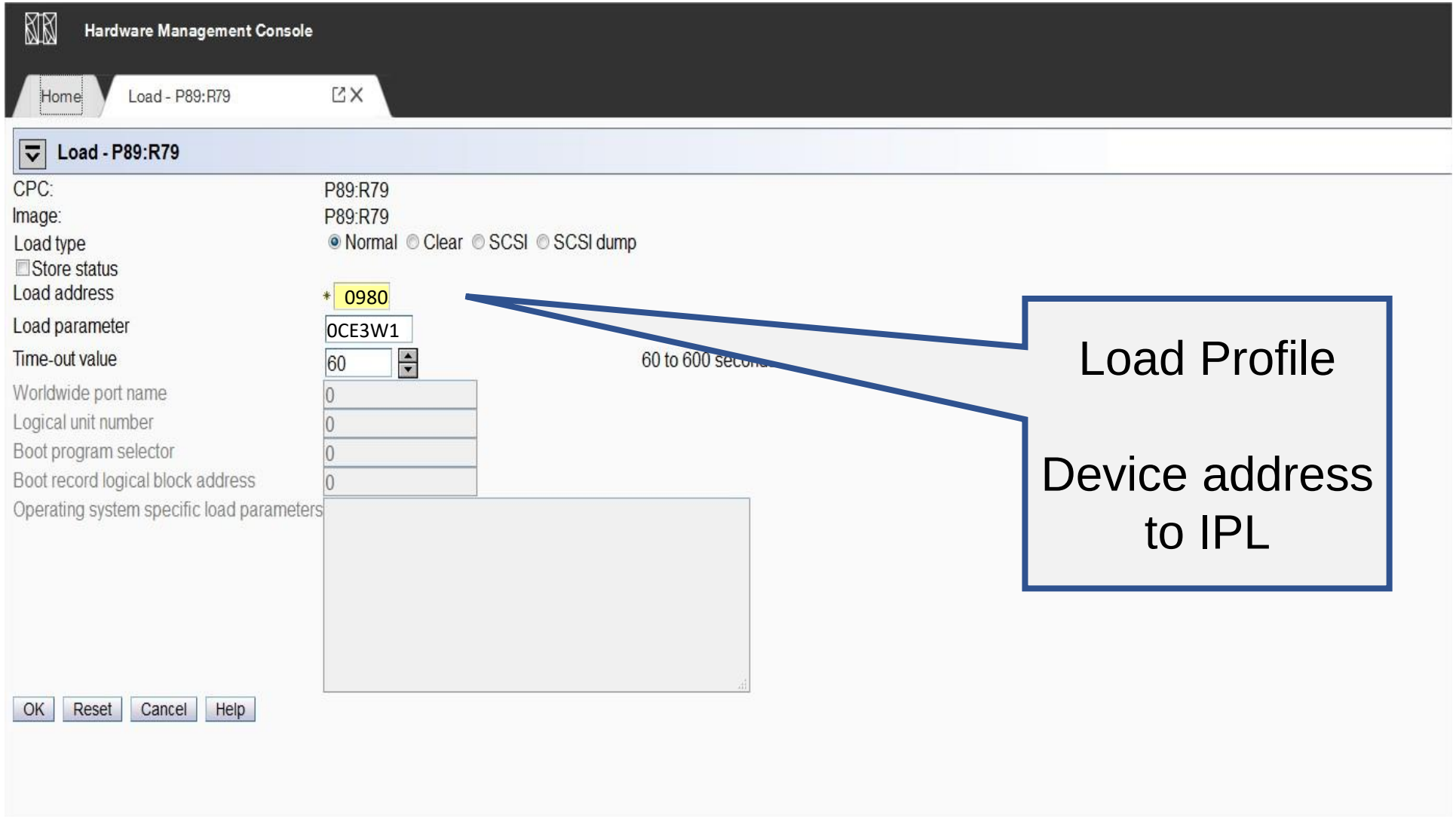
Profile **contains** DASD **device number to IPL**

IPL



HMC "**current**" screen image

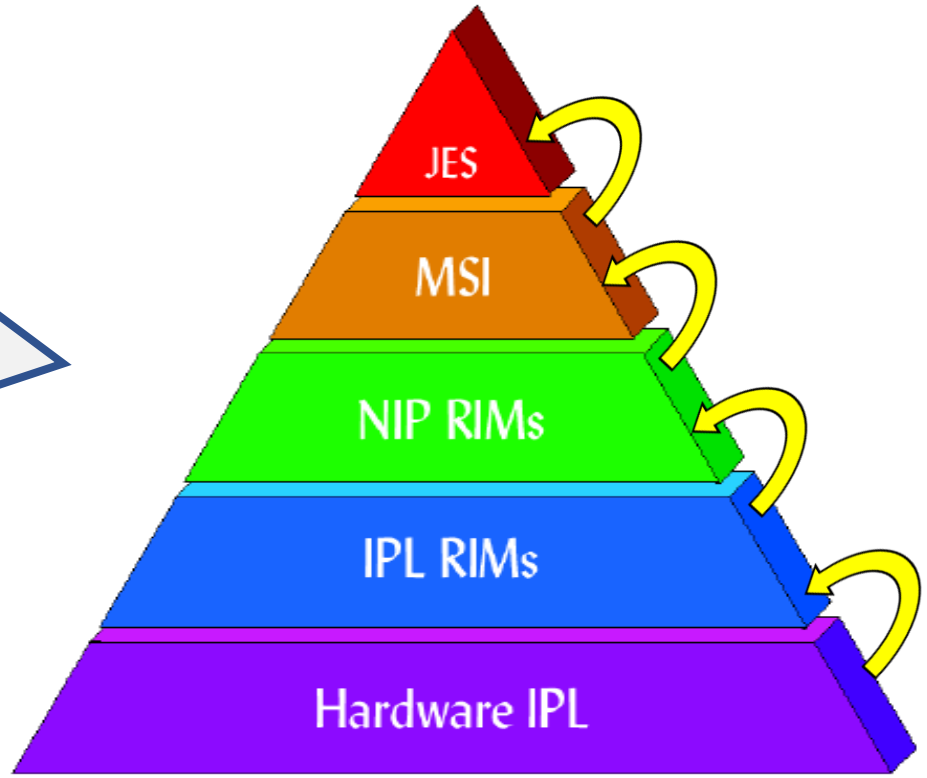
IPL


Hardware Management Console
Home
Load - P89:R79
Load - P89:R79
CPC:
P89:R79
Image:
P89:R79
Load type
Normal
Clear
SCSI
SCSI dump
Store status
Load address
0980
Load parameter
0CE3W1
Time-out value
60
60 to 600 seco
Worldwide port name
0
Logical unit number
0
Boot program selector
0
Boot record logical block address
0
Operating system specific load parameters
OK
Reset
Cancel
Help
Load Profile
Device address to IPL

IPL

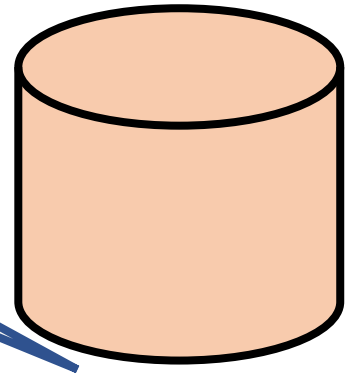
IPL processed **in phases**

Each phase **builds on the previous**



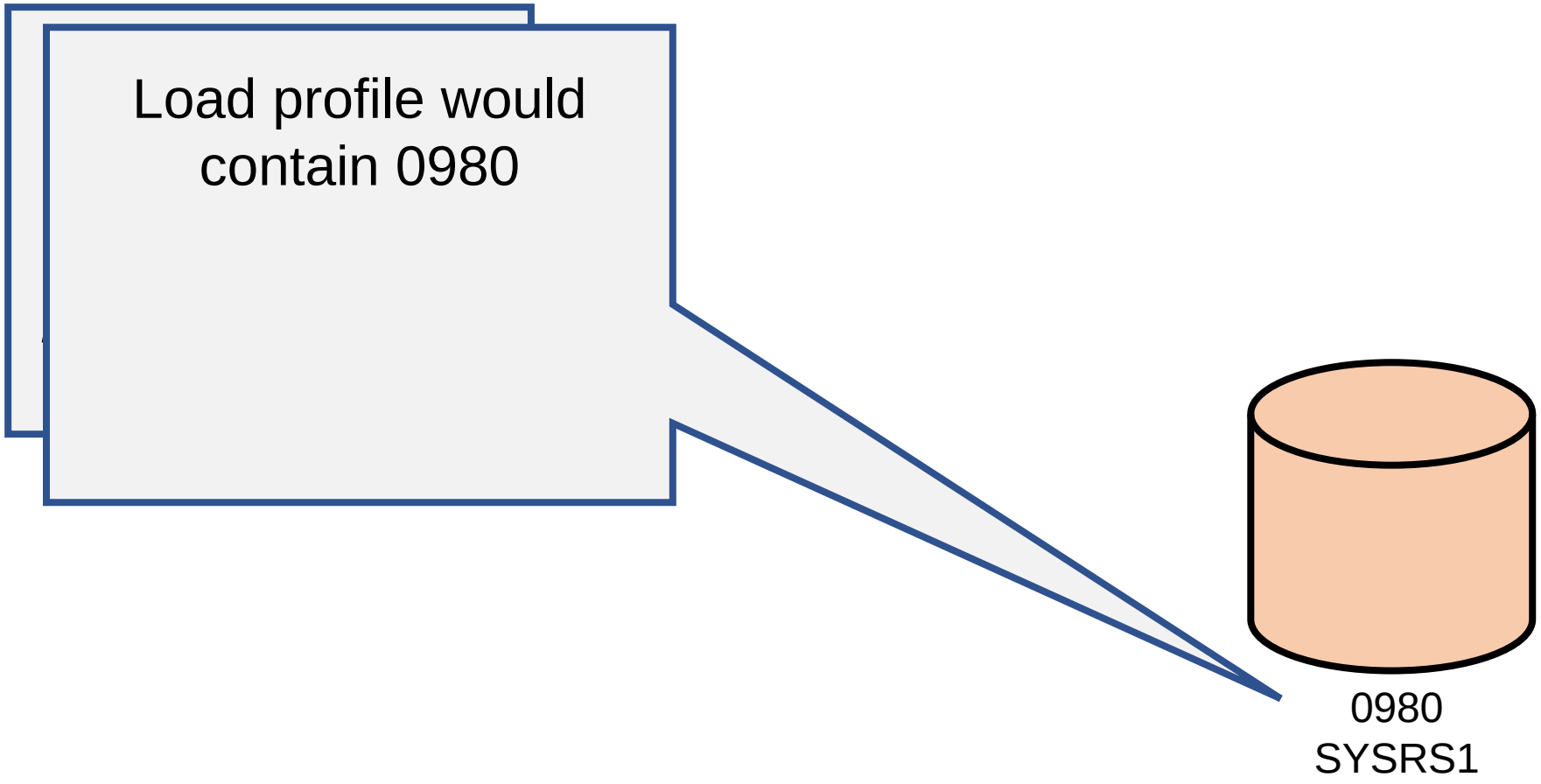
IPL

To “Boot” or “**IPL**” z/OS, the **device number** of the **SYSRES** (*system residence volume*) is specified

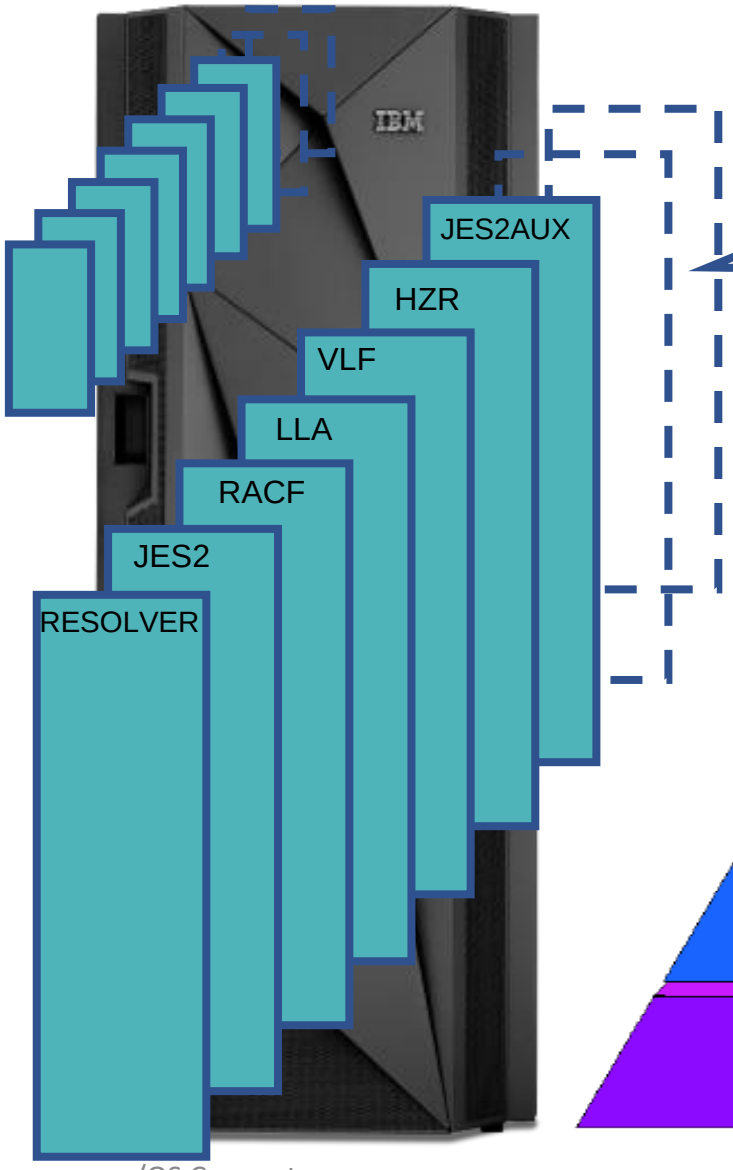


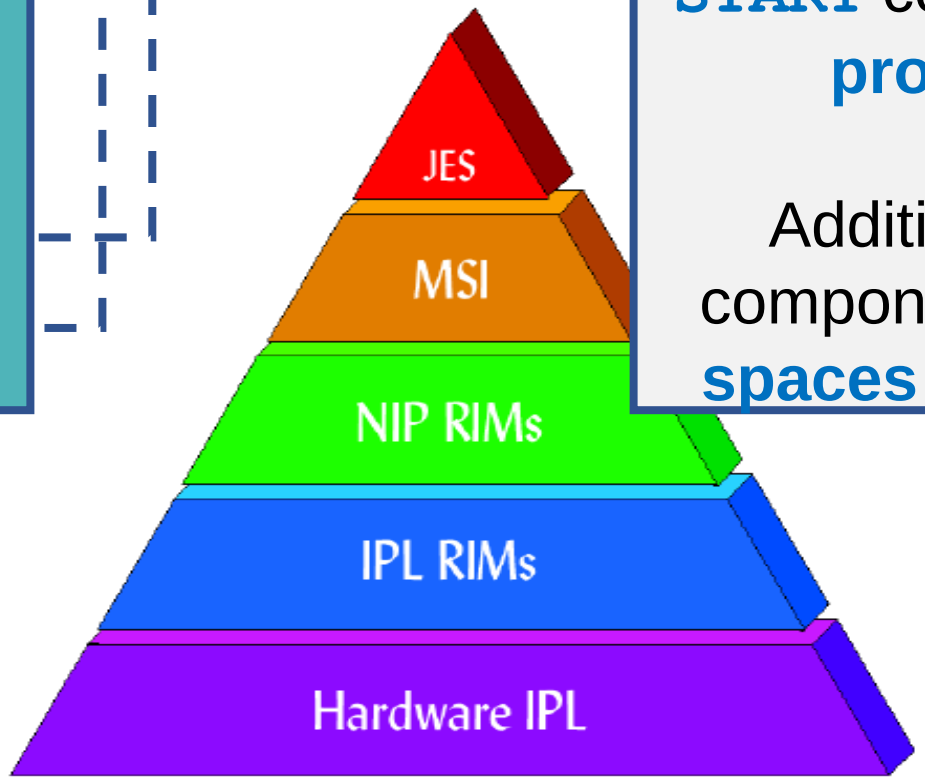
0980
SYSRS1

IPL



IPL





MSI – several z/OS component **address spaces created**

After MSI, queued **`START`** commands are **processed**

Additional z/OS component **address spaces are created**

IPL

Important address spaces started at Master Scheduler Initialization time (MSI):

- **SMF** *(System Management Facility)*

Collector of system and job-related information used for: billing users, reporting reliability, analyzing the configuration, evaluating activity, profiling system resource use, maintaining system security.

IPL

Important address spaces started at Master Scheduler Initialization time (MSI):

- **SMF** *(System Management Facility)*
- **System Logger**

A set of services that allow an application to **write, browse, and delete log data**. **Designed to merge data** from multiple instances of an application, including merging data from different systems across a sysplex.

IPL

Important address spaces started at Master Scheduler Initialization time (MSI):

- **SMF** *(System Management Facility)*
- **System Logger**
- **BCPii** *(Base Control Program internal interface)*

A set of services that allow special z/OS applications a way of programmatically **controlling the mainframe hardware configuration**

IPL

Important started address spaces

- **JES** (*Job Entry Subsystem* a.k.a. JES2 or JES3)

Receives **jobs**, schedules them for **processing** and handles their **output**

IPL

Important started address spaces

- **JES** (*Job Entry Subsystem* a.k.a. JES2 or JES3)
- **RACF®** (*Resource Access Control Facility* a.k.a. z/OS Security Server)

Security program used to **protect resources**

IPL

Important started address spaces

- **JES** (*Job Entry Subsystem* a.k.a. JES2 or JES3)
- **RACF®** (*Resource Access Control Facility* a.k.a. z/OS Security Server)
- **ICSF** *(Integrated Cryptographic Service Facility)*

Provides the z/OS **Cryptography** interfaces to applications: **Enciphering, Deciphering, Hashing,** and **Generating/verifying digital signatures.** Implements this via both crypto hardware and software.

IPL

Important started address spaces

- **JES** (*Job Entry Subsystem* a.k.a. JES2 or JES3)
- **RACF®** (*Resource Access Control Facility* a.k.a. z/OS Security Server)
- **ICSF** *(Integrated Cryptographic Service Facility)*
- **VTAM®** (*Virtual Telecommunications Access Method* a.k.a. z/OS Communications Server)

Implements **SNA** (*Systems Network Architecture*) with **API** (*Application Program Interface*) for **communicating** with devices and programs

IPL

Important started address spaces

- **JES** (*Job Entry Subsystem* a.k.a. JES2 or JES3)
- **RACF®** (*Resource Access Control Facility* a.k.a. z/OS Security Server)
- **ICSF** *(Integrated Cryptographic Service Facility)*
- **VTAM®** (*Virtual Telecommunications Access Method* a.k.a. z/OS Communications Server)
- **TCAS** (*Terminal Control Address Space*)

Communication **interface for TSO/E** (*Time Sharing Option/Extensions*)

IPL

Important started address spaces

- **JES** (*Job Entry Subsystem* a.k.a. JES2 or JES3)
- **RACF®** (*Resource Access Control Facility* a.k.a. z/OS Security Server)
- **ICSF** *(Integrated Cryptographic Service Facility)*
- **VTAM®** (*Virtual Telecommunications Ac*
- **TCAS** (*Terminal Control Addre Space)*
- **TCP/IP** (*Transmission Control Protocol/Internet Protocol* a.k.a. z/OS Communications Server: IP)

Networking **protocols** for computer **communications**

IPL

Important started address spaces

- **JES** (*Job Entry Subsystem* a.k.a. JES2 or JES3)
- **RACF®** (*Resource Access Control Facility* a.k.a. z/OS Security Server)
- **VTAM®** (Communications Server)
- **TCAS** (
- **TCP/IP** (z/OS Communications Server: IP)

Once **TCP/IP is initialized**, the **IPL** is **considered complete**

IPL

- Of course there is **still more work** to be done

- **Middleware** and **application products** like the following need to be **started**:
  - **SA z/OS** (*System Automation for z/OS*)
  - **DB2®** (*Data Base 2*)
  - **CICS® Transaction Server for z/OS** (*Customer Information Control System*)
  - **IMS™** (*Information Management System*)
  - **WAS** (*WebSphere® Application Server)*

Check your Knowledge

- What component provides cryptographic functions to applications?

Check your Knowledge

- What component provides cryptographic functions to applications?
  - **ICSF**

Check your Knowledge

- What component provides cryptographic functions to applications?
  - **ICSF**
- What is the component that is the collector of system and job-related information?

Check your Knowledge

- What component provides cryptographic functions to applications?
  - **ICSF**
- What is the component that is the collector of system and job-related information?
  - **SMF**

Sysplex

IBM Speak – Sysplex

- A **sysplex** is
  - A **collection** of z/OS **systems** that **cooperate**

IBM Speak – Parallel Sysplex

- A **Parallel Sysplex**
  - Is **a sysplex** that uses **multisystem data-sharing** technology

IBM Speak – Parallel Sysplex

- A **Parallel Sysplex**
  - Is **a sysplex** that uses **multisystem data-sharing** technology

  - Allows **direct, concurrent read/write** access to **shared data** from **all systems** without impacting **performance** or **data integrity**

IBM Speak – Parallel Sysplex

- A **Parallel Sysplex**
  - Is **a sysplex** that uses **multisystem data-sharing** technology
  - Allows **direct, concurrent read/write** access to **shared data** from **all systems** without impacting **performance** or **data integrity**
  - **Work** requests that are associated with a **single workload**
    - Can be **dynamically distributed** for parallel execution on systems
    - Based on **available** processor **capacity**

IBM Speak – Parallel Sysplex

- In many ways a **Parallel Sysplex appears** as a **single** large **system**

Parallel Sysplex

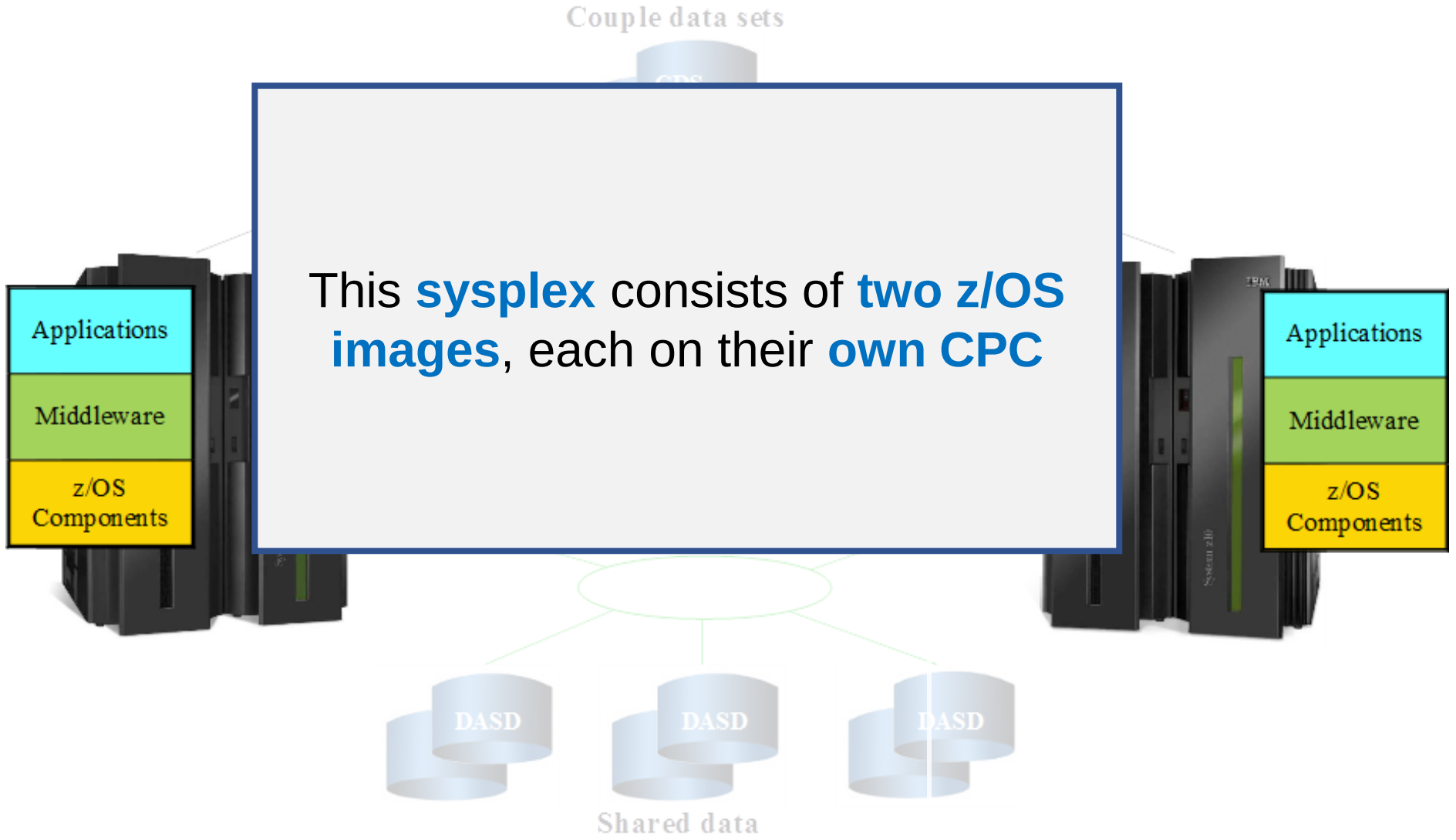
- In a **Parallel Sysplex**, **CPCs** (*central processing complexes*) are:
  - **Connected** through a **CF** (*coupling facility)*
    - A coupling facility **enables parallel processing**
    - And **improved data sharing** for authorized programs

Parallel Sysplex

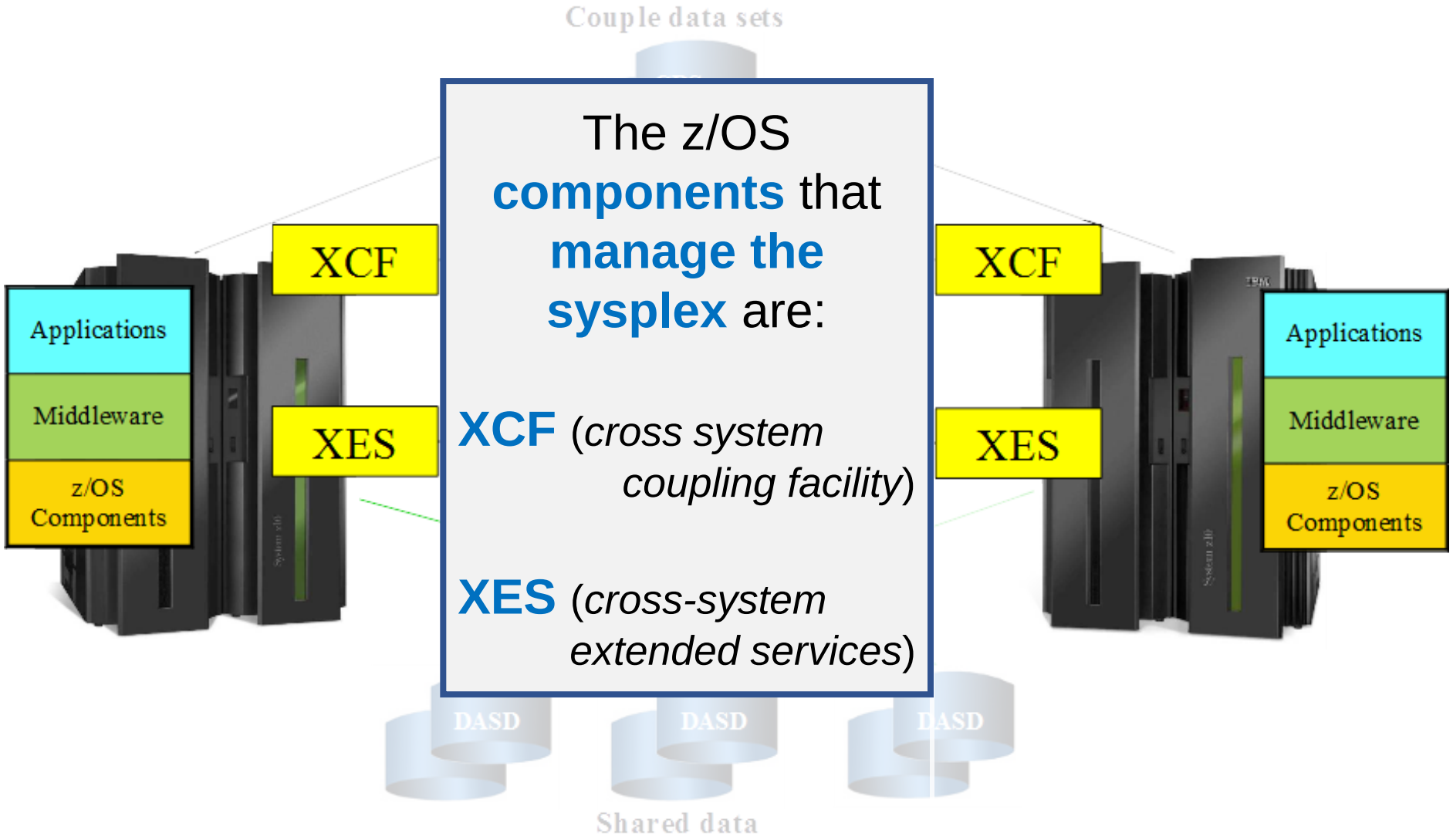
- In a **Parallel Sysplex**, **CPCs** (*central processing complexes*) are:
  - **Connected** through a **CF** (*coupling facility)*
    - A coupling facility **enables parallel processing**
    - And **improved data sharing** for authorized programs

- The **collection of z/OS** systems **cooperate** to:
  - Process **workloads**
  - Provide **higher availability**
  - Provide easier **systems management**
  - Provide **improved growth**

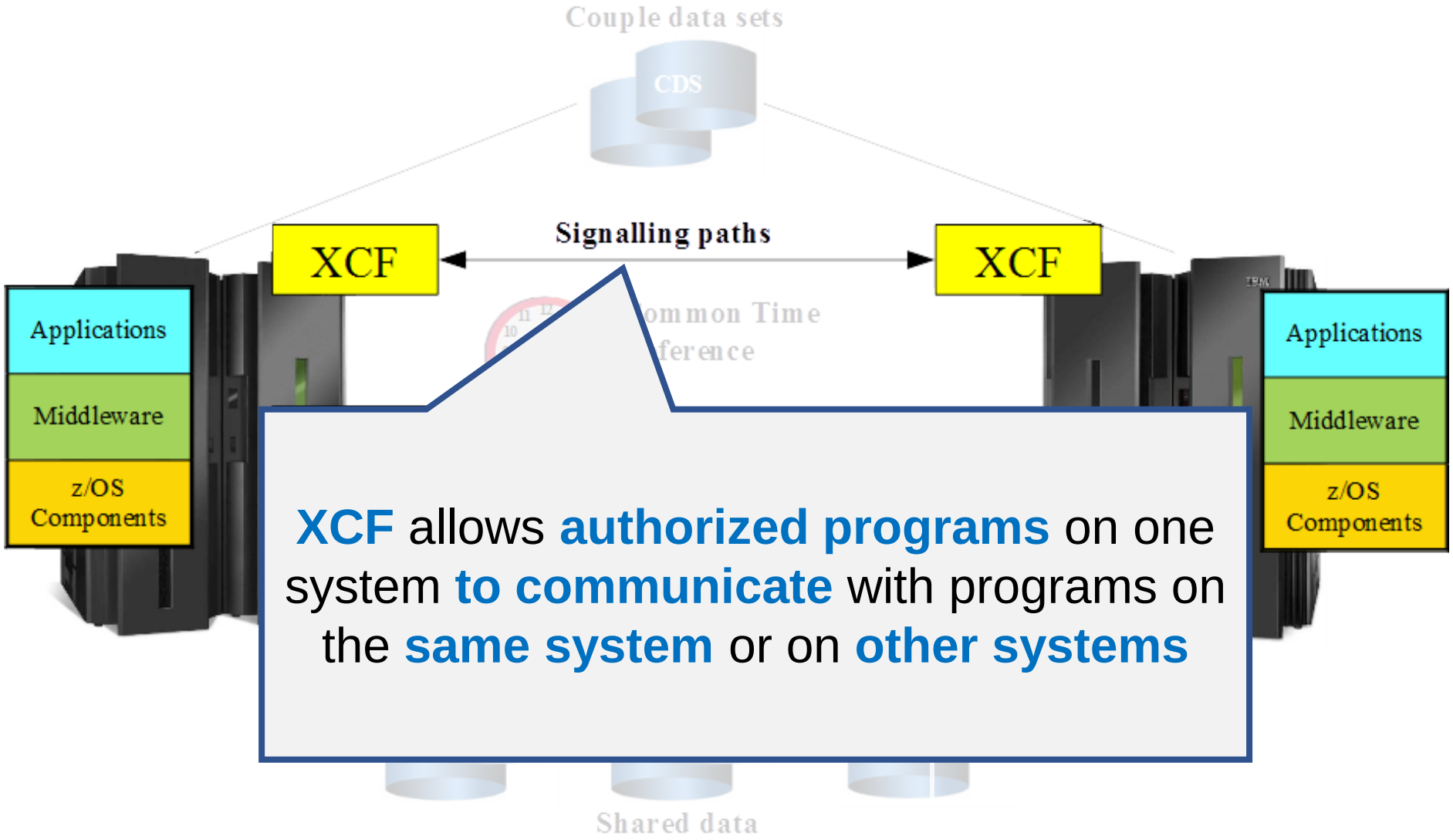
Parallel Sysplex



Parallel Sysplex

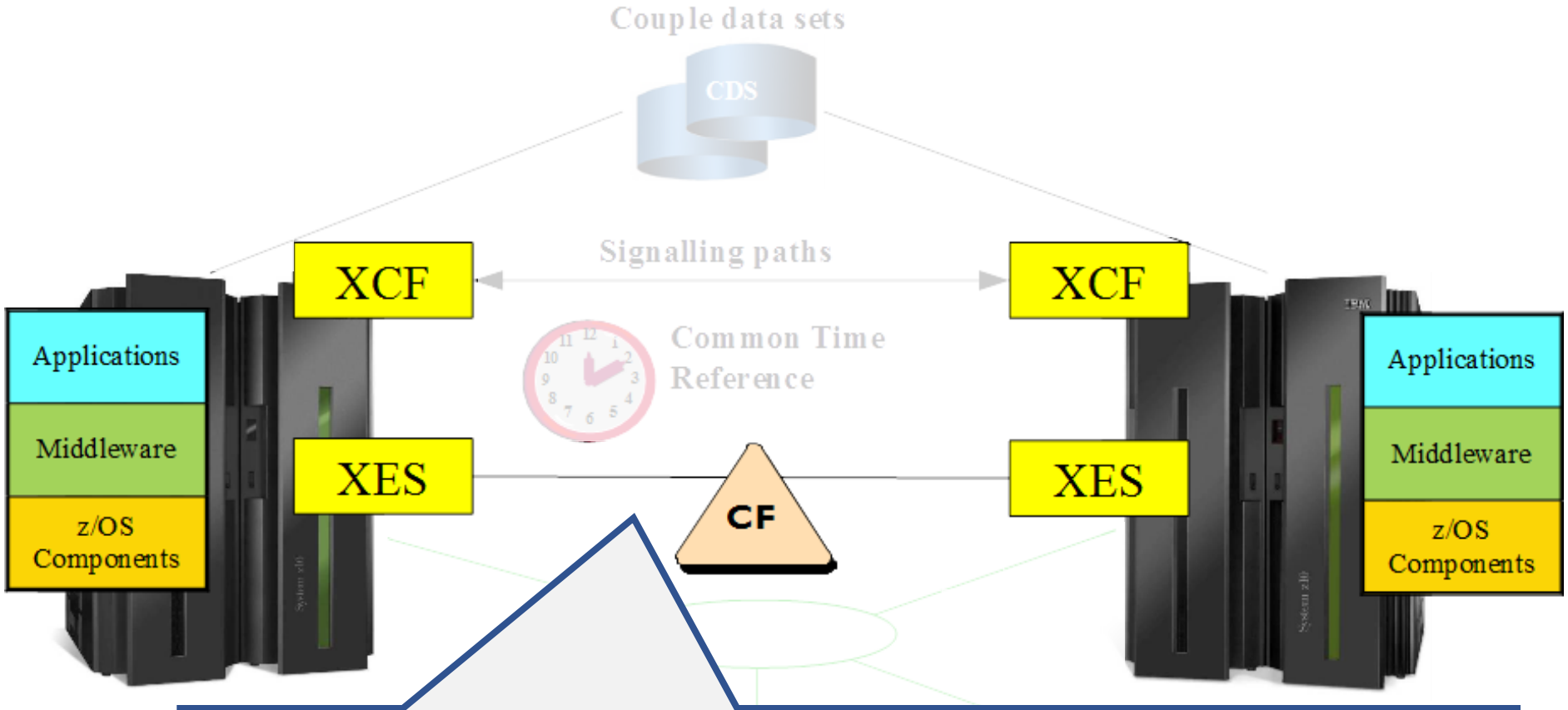


Parallel Sysplex



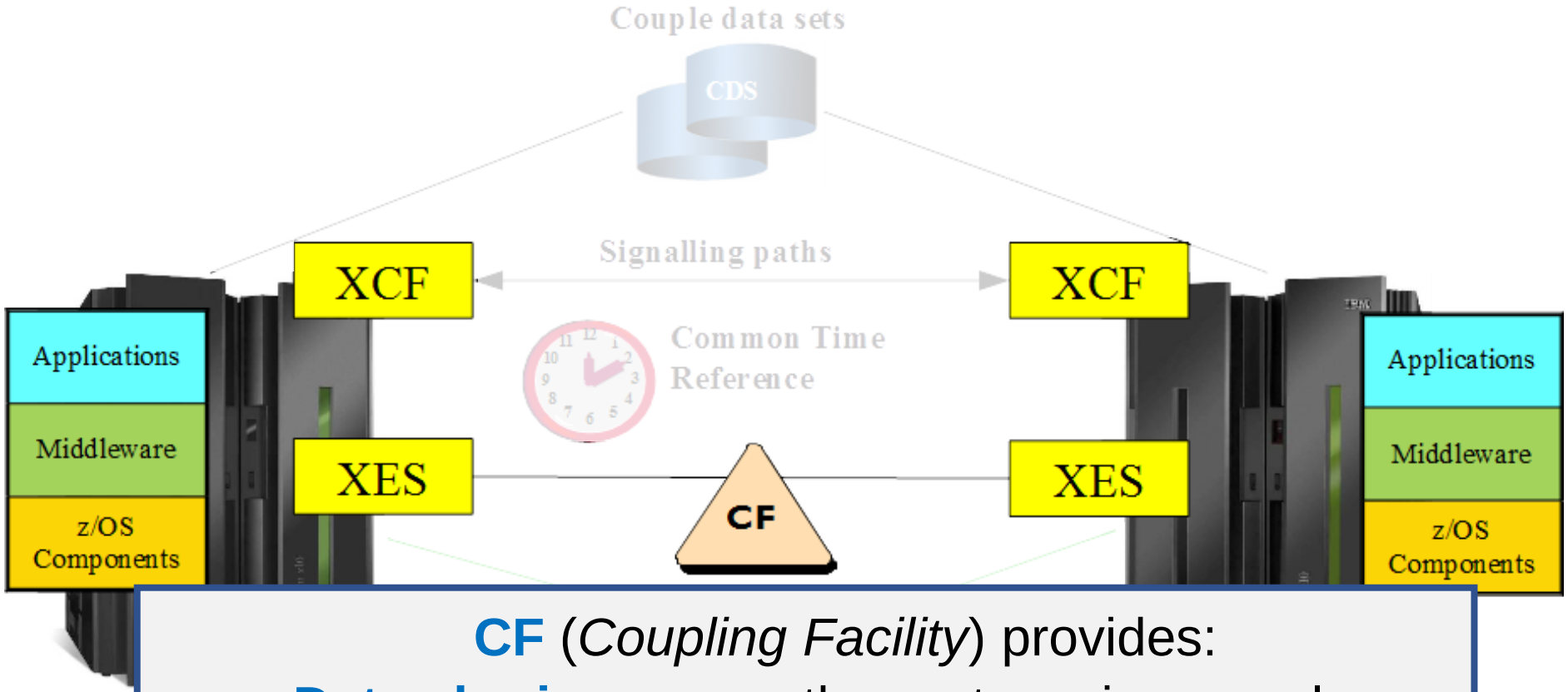
XCF allows **authorized programs** on one system **to communicate** with programs on the **same system** or on **other systems**

Parallel Sysplex



XES enables **applications** and **subsystems** to **take advantage** of the **coupling facility**

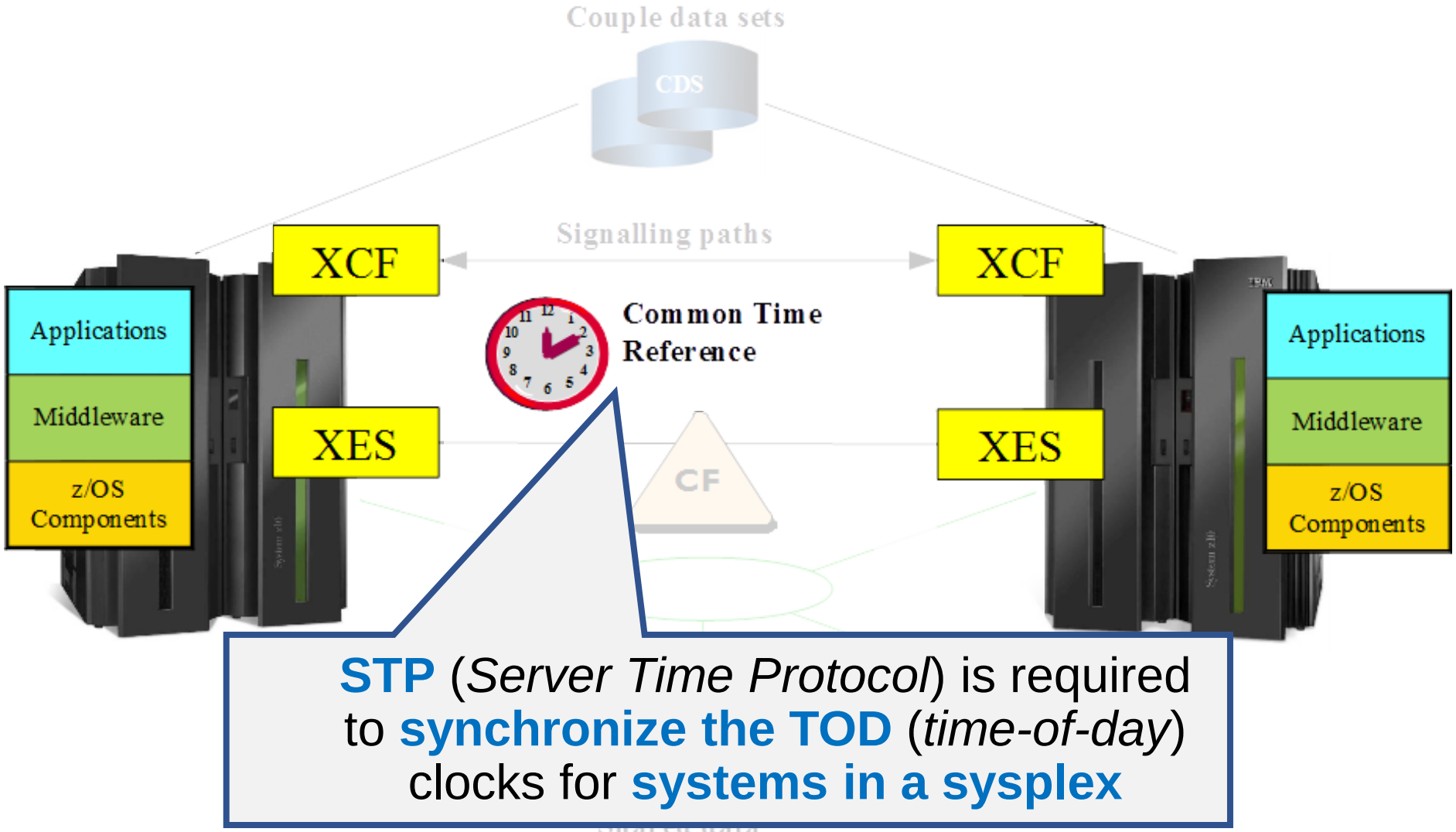
Parallel Sysplex



CF (*Coupling Facility*) provides:

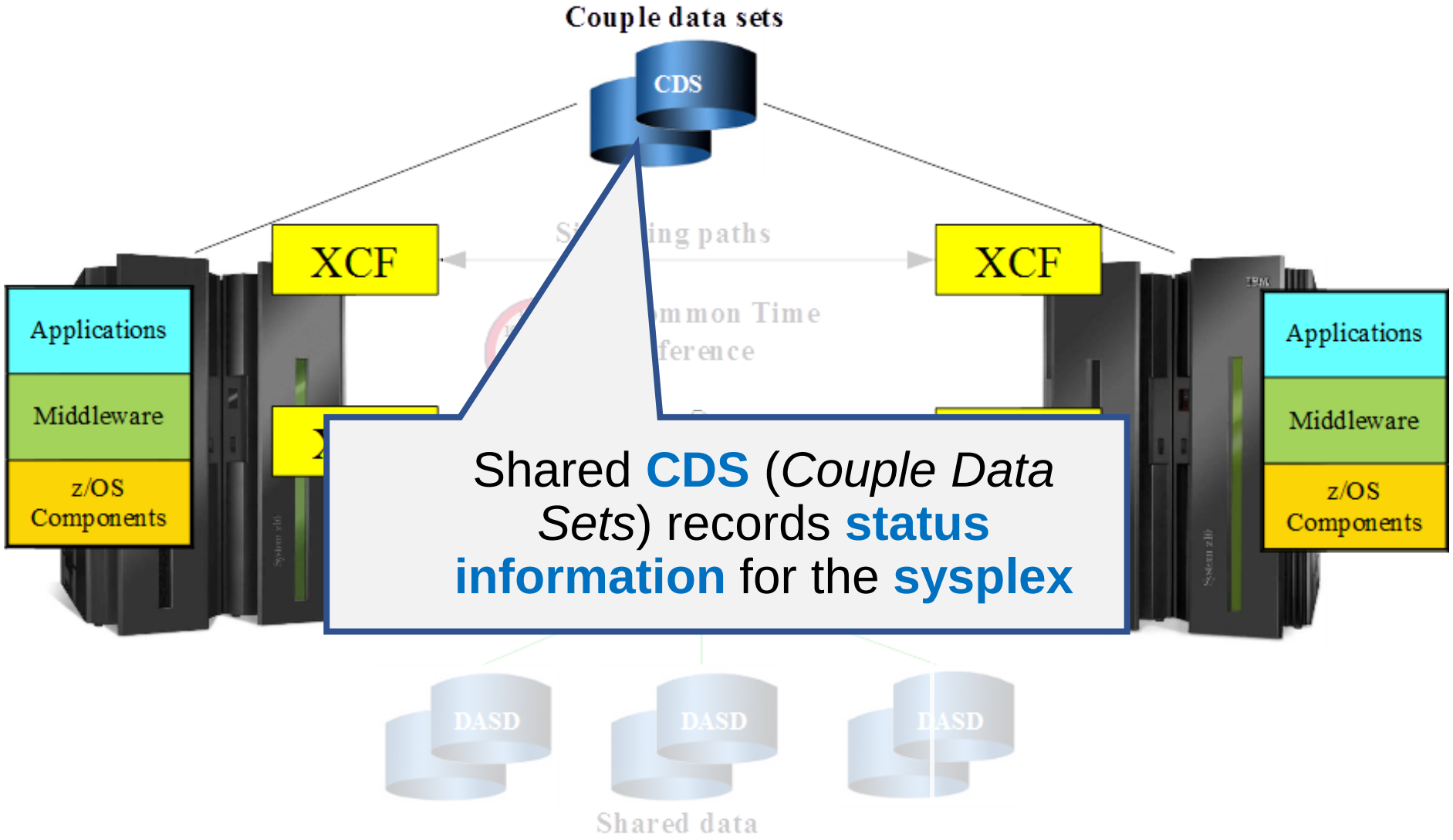
- **Data sharing** across the systems in a sysplex
- Maintaining **the integrity and consistency** of shared data
- Maintaining the **availability** of a sysplex

Parallel Sysplex

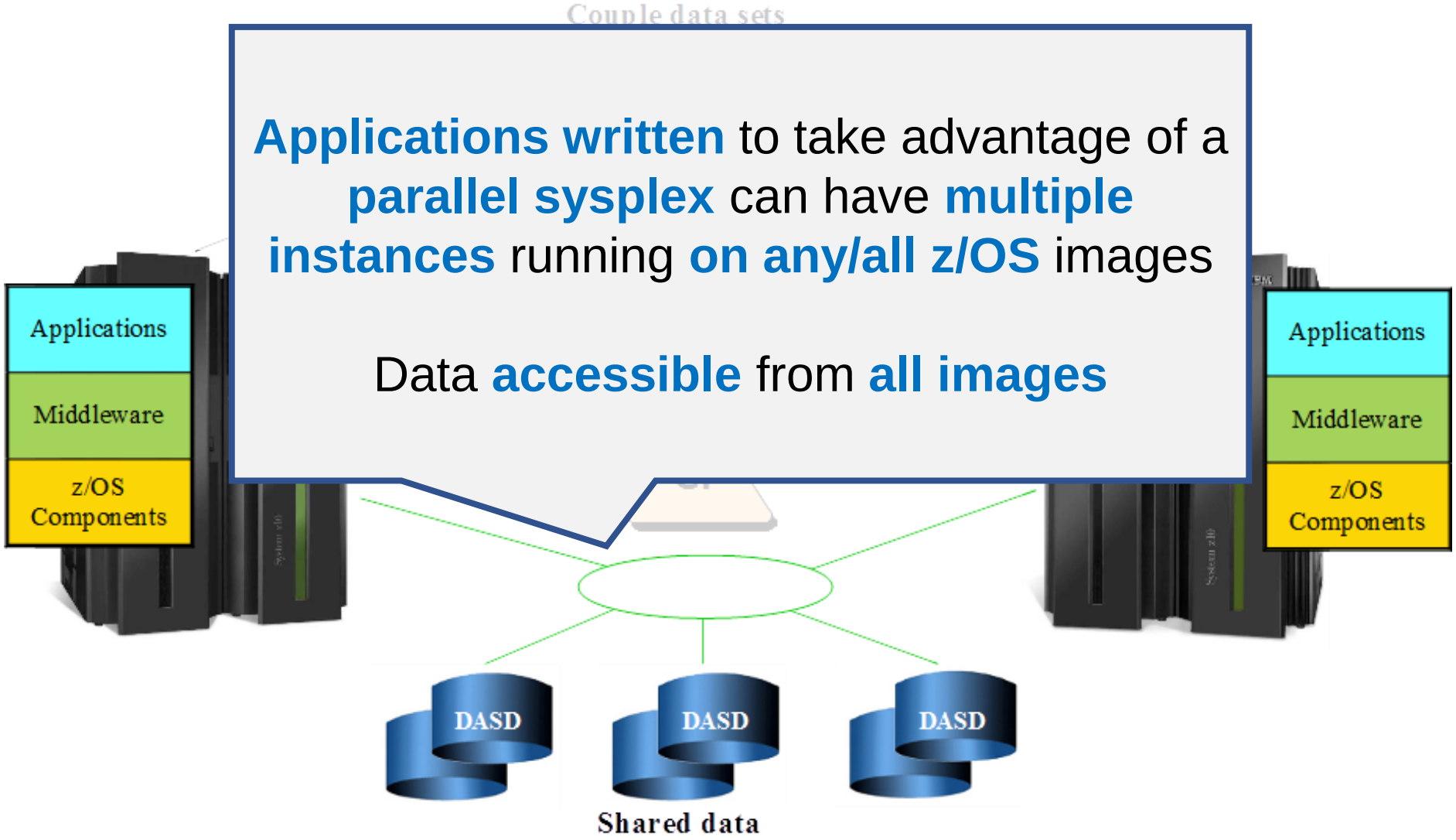


STP (*Server Time Protocol*) is required to **synchronize the TOD** (*time-of-day*) clocks for **systems in a sysplex**

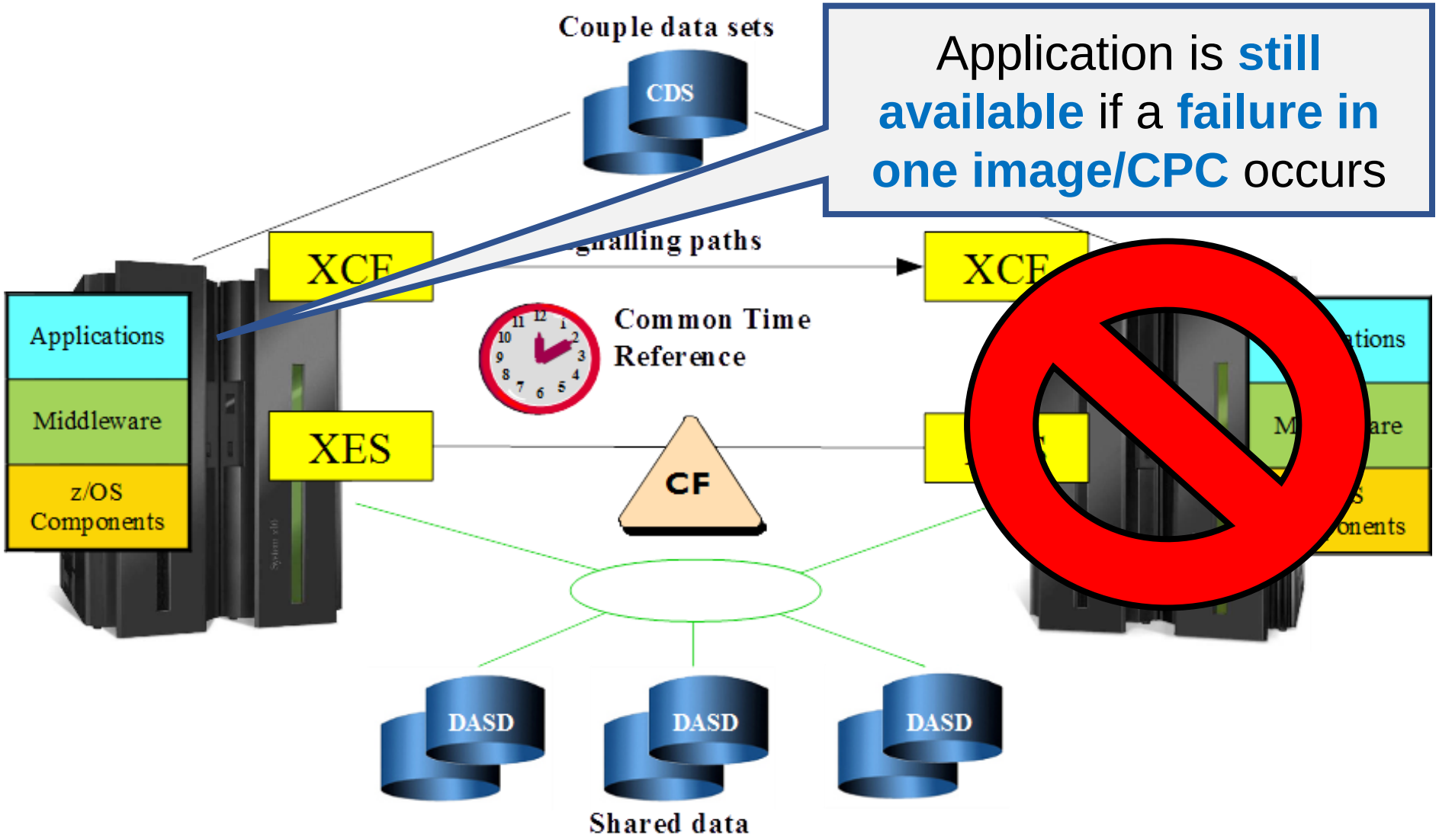
Parallel Sysplex



Parallel Sysplex



Parallel Sysplex



IBM Speak – GDPS

- **Distance** between systems in a parallel sysplex is **limited by connectivity**
  - To the **Coupling Facility**
  - To the **STP** (*Server Time Protocol*) (i.e., the clock)
  - To the shared **DASD**

100 km is the max
but not practical

IBM Speak – GDPS

- **Distance** between systems in a parallel sysplex is **limited by connectivity**
  - To the **Coupling Facility**
  - To the **STP** (*Server Time Protocol*) (i.e., the clock)
  - To the shared **DASD**

- **GDPS** (*Geographically Dispersed Parallel Sysplex*)
  - **Extension** of a parallel sysplex
  - Systems can be **located in different cities**
  - Goals of **continuous availability** and **disaster recovery**

Check Your Knowledge

Check your Knowledge

- What is the piece of hardware which allows data sharing between members in a sysplex?

Check your Knowledge

- What is the piece of hardware which allows data sharing between members in a sysplex?
  - **Coupling Facility (CF)**

Check your Knowledge

- What is the piece of hardware which allows data sharing between members in a sysplex?
  - **Coupling Facility (CF)**
- What synchronizes time-of-day clocks for systems in a sysplex?

Check your Knowledge

- What is the piece of hardware which allows data sharing between members in a sysplex?
  - **Coupling Facility (CF)**
- What synchronizes time-of-day clocks for systems in a sysplex?
  - **Server Time Protocol (STP)**

Serialization

z/OS Concepts – Serialization

- Customers using z/OS may runs **many concurrent programs** and **many threads of the same program simultaneously**.
- Think **credit card transactions!**

z/OS Concepts – Serialization

- Customers using z/OS may runs **many concurrent programs** and **many threads of the same program simultaneously**.
- **Think credit card transactions!**
- **What if these programs need to update resources at the same time?**

z/OS Concepts – Serialization

- Customers using z/OS may runs **many concurrent programs** and **many threads of the same program simultaneously**.
- **Think credit card transactions!**
- **What if these programs need to update resources at the same time?**
  - Data sets
  - Virtual resources
    - Lists
    - Queues
    - Data areas (control blocks)

z/OS Concepts – Serialization

- Customers using z/OS may runs **many concurrent programs** and **many threads of the same program simultaneously**.
- **Think credit card transactions!**
- **What if these programs need to update resources at the same time?**
  - Data sets
  - Virtual resources
    - Lists
    - Queues
    - Data areas (control blocks)
- **How do we ensure data integrity and fair access to these resources?**

z/OS Concepts – Serialization

- **Global Resource Serialization (GRS)** is the z/OS component designed to protect the integrity of resources in a multitasking, multi-host environment

z/OS Concepts – Serialization

- **Global Resource Serialization (GRS)** is the z/OS element designed to protect the integrity of resources in a multitasking, multi-host environment
- **Coordinates access to resources** used by more than one program

z/OS Concepts – Serialization

- **Global Resource Serialization (GRS)** is the z/OS element designed to protect the integrity of resources in a multitasking, multi-host environment
- **Coordinates access to resources** used by more than one program
- Uses **ENQs** and **Latches** to scope resources at various levels:
  - STEP – synchronize within a single address space
  - SYSTEM – synchronize single system apps
  - SYSTEMS – synchronize multisystem apps
  - SYSPLEX – synchronize across a sysplex
  - Custom (Latches) – synchronize within a multitasking, or multi-threaded application

z/OS Concepts – Serialization

- **Program obtains ENQ** or **Latch** before reading or updating protected resource

z/OS Concepts – Serialization

- Program obtains ENQ or Latch before updating protected resource
- Determines if it should be **exclusive (write access)** or **shared (read/only access)**

Managing Workloads

z/OS Concepts – Managing Workloads

- One of the strengths of the IBM Z platform and the z/OS operating system is the ability to run **multiple workloads** at the same time within one z/OS image or across multiple images.

z/OS Concepts – Managing Workloads

- One of the strengths of the IBM Z platform and the z/OS operating system is the ability to run multiple workloads at the same time within one z/OS image or across multiple images.
- z/OS needs to:
  - **Prioritize work**
  - **Use the installation resources as efficiently as possible**
  - **Maintain the highest possible throughput**
  - **Achieve the best possible system responsiveness.**

z/OS Concepts – Managing Workloads

- One of the strengths of the IBM Z platform and the z/OS operating system is the ability to run multiple workloads at the same time within one z/OS image or across multiple images.
- z/OS needs to:
  - Prioritize work
  - Use the installation resources as efficiently as possible
  - Maintain the highest possible throughput
  - Achieve the best possible system responsiveness.
- Dynamic workload management is accomplished through the **Workload Management (WLM)** component of the z/OS operating system

z/OS Concepts – Managing Workloads

- WLM allows a customer to define **performance goals** and assign a **business importance** to each goal.

z/OS Concepts – Managing Workloads

- WLM allows a customer to define performance goals and assign a business importance to each goal.
- Goals are defined in **business terms**, and the system decides how much resource, such as **CPU** and **storage**, should be given to the work to meet its goal.

z/OS Concepts – Managing Workloads

- WLM algorithms use the **service definition information** and **internal monitoring feedback** to check how well they are doing in meeting the goals.
  - Algorithms periodically adjust the allocation of resource as the workload level changes.

z/OS Concepts – Managing Workloads

- WLM algorithms use the **service definition information** and **internal monitoring feedback** to check how well they are doing in meeting the goals.
  - Algorithms periodically adjust the allocation of resource as the workload level changes.
- For each system, WLM manages the system resources.
  - **Coordinates and shares performance information across the sysplex.**
    - How well it manages one system is based on how well the other systems are also doing in meeting the goals. If there is contention for resources, it makes the appropriate trade-offs based on the importance of the work and how well the goals are being met.

z/OS Concepts – Managing Workloads

- WLM can **dynamically start and stop server address spaces** to process work from application environments.
  - On a single system or across the sysplex
  - Batch initiators can be managed
    - Can dynamically manage the number of batch initiators for one or more job classes to meet the performance goals of the work.

z/OS Concepts – Managing Workloads

- WLM can **dynamically start and stop server address spaces** to process work from application environments.
  - On a single system or across the sysplex
  - Batch initiators can be managed
    - Can dynamically manage the number of batch initiators for one or more job classes to meet the performance goals of the work.
- WLM also collects **real-time performance data** and **delay monitoring**.
  - Available for performance monitors and reporters for integration into detailed reports.

For More Information

Introduction to the New Mainframe: z/OS Basics
https://www.redbooks.ibm.com/redbooks/pdfs/sg246366.pdf

IBM z/OS basic skills education
https://www.ibm.com/support/knowledgecenter/en/zosbasics/com.ibm.zos.zbasics/lcmain.html

z/OS Introduction for IT professionals
ibm.biz/zOSclass

Please submit your session feedback!

- Do it online at http://conferences.gse.org.uk/2019/feedback/ag

- This session is AG

1. What is your conference registration number?

This is the three digit number on the bottom of your delegate badge

2. Was the length of this presention correct?

1 to 4 = "Too Short" 5 = "OK" 6-9 = "Too Long"

1 2 3 4 5 6 7 8 9

3. Did this presention meet your requirements?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

4. Was the session content what you expected?

1 to 4 = "No" 5 = "OK" 6-9 = "Yes"

1 2 3 4 5 6 7 8 9

IBM Systems Worldwide Client Experience Centers

IBM Systems Worldwide Client Experience Centers *maximize IBM Systems competitive advantage in the Cloud and Cognitive era by providing access to world class technical experts and infrastructure services to assist Clients with the transformation of their IT implementations..*

9 Worldwide Locations (* also Infrastructure Hubs):

Austin TX , *Poughkeepsie NY, Rochester MN, Tucson AZ, *Beijing CHINA, Boeblingen GERMANY, Guadalajara MEXICO,*Montpellier FRANCE, Tokyo JAPAN

Client Experience	***Architecture & Design***	***Infrastructure Solutions***	***Content***
Tailored, in-depth technology Innovation Exchange Events Relationship building Demonstrations Meetups Solution workshops Remote options	Advise clients, "Art of the Possible" Discovery & Design Workshops, Consulting, Showcases, Reference Architectures, Co-Creation of assets	Benchmarks, MVP & Proof of Technology "Test Drives" Demonstrations Infrastructure Services Certify ISV solutions Hosting Cloud Environment	Content Development IBM Redbooks Training Courses Video courses "Test Drives" Demonstrations
(Inbound & Outbound)	(Inbound & Outbound)	(Inbound to Centers)	

NEW: *Co-Creation Lab; CEC Cloud; RedHat Center of Competency*

For further information, please contact the Centers via email at:
ccenter@us.ibm.com

© 2019 IBM Corporation


z/OS


IBM