# Executing Programs That Require Special Runtime Environments



Dave Nicolette
Software Developer

@davenicolette neopragma.com

### Overview



- z/OS Runtime Environments
- TSO Batch
- USS Batch

# z/OS Runtime Environments

### Batch

Process a potentially-large quantity of input data in a single execution of a program

### Batch

Process a potentially-large quantity of input data in a single execution of a program

### TSO

Execute one program or command at a time, receiving notification of the result immediately

### Batch

Process a potentially-large quantity of input data in a single execution of a program

### TSO

Execute one program or command at a time, receiving notification of the result immediately

### USS

A UNIX-like command-line interactive environment

### Batch

Process a potentially-large quantity of input data in a single execution of a program

### TSO

Execute one program or command at a time, receiving notification of the result immediately

### USS

A UNIX-like command-line interactive environment

### CICS

A self-contained execution environment for interactive or OLTP applications

# JCL for TSO Batch

# IKJEFTxx Entry Points - Behaviors

Situation	IKJEFT01	IKJEFT1A, IKJEFT1B
Command completes with non-zero RC	TMP goes to next cmd	Return with RC in reg 15
System abend	Terminate, RC=12	S04C abend, RC in reg 15
User abend	Terminate, RC in reg 15	S04C abend, RC in reg 15
CLIST non-zero RC	TMP goes to next cmd	TMP goes to next cmd
Other non-zero RC	TMP goes to next cmd	Terminate, RC in reg 15
Error from program not given control directly by TMP	TMP goes to next cmd	TMP goes to next cmd

```
//STEP1 EXEC PGM=IKJEFT01, DYNAMNBR=30
//STEPLIB DD DSN=DSN810.SDSNLOAD, DISP=SHR
// DD DSN=CEE.SCEERUN, DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM(ssid)
RUN PROGRAM(progname) PLAN(planname)
    LIB('library-name')
END
//SYSPRINT DD SYSOUT=*
```

■ Run a DB2 application program using the TSO attachment facility (AF)

### Examples of Resource Managers

### DB<sub>2</sub>

Relational Database Management System

### IMS/DB

Hierarchical Database Management System

### **VSAM**

Virtual Sequential Access Method

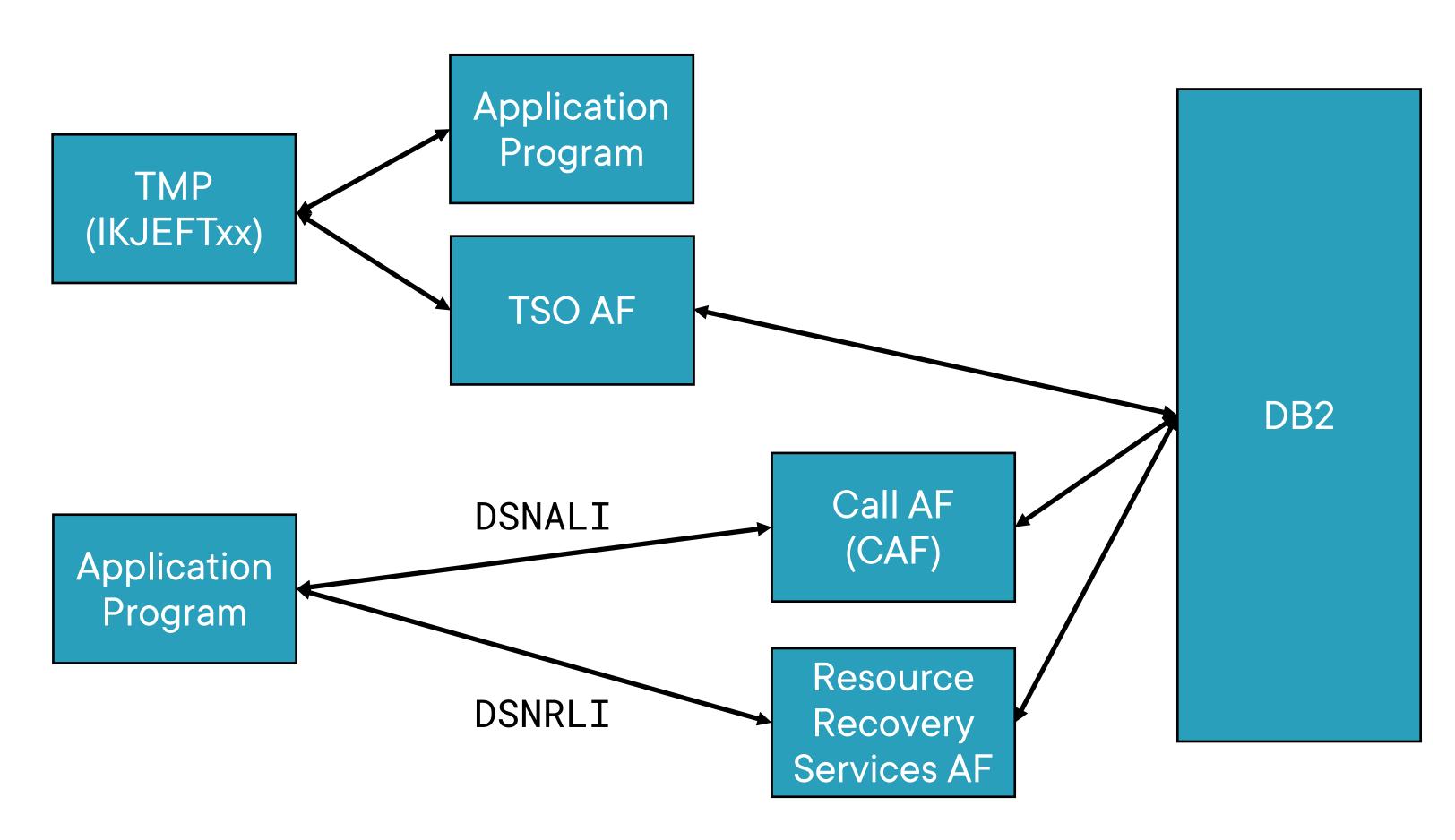
### MQ

Message Queueing System

### **QMF**

Query Management Facility (Analytics)

### DB2 Attachment Facilities



```
//STEP1 EXEC PGM=IKJEFT01, DYNAMNBR=30
//STEPLIB DD DSN=DSN810.SDSNLOAD, DISP=SHR
           DD DSN=CEE.SCEERUN, DISP=SHR
//
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
 DSN SYSTEM(ssid)
 RUN PROGRAM(progname) PLAN(planname)
     LIB('library-name')
 END
//SYSPRINT DD SYSOUT=*
//STEP1 EXEC PGM=progname
//STEPLIB DD DSN=my.library.dsn,DISP=SHR
          DD DSN=DSN810.SDSNEXIT, DISP=SHR
//
         DD DSN=DSN810.SDSNLOAD, DISP=SHR
          DD DSN=CEE.SCEERUN, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
```

■ Run a DB2 application program using the TSO attachment facility (AF)

■ Run a DB2 application program using the RRS attachment facility (AF)

# JCL for USS Batch

# USS Batch Utility - BPXBATCH

### **Entry Point BPXBATCH**

Forks process to run in a separate address space.

Shares user's profile.

Variables may be overwritten if profile is changed.

Cannot fork or exec child processes.

### Entry Point BPXBATSL

Spawns process to run in the same address space.

Has own profile.

Variables not affected by changes in profile.

Can fork or exec child processes.

# USS Batch Standard DD Statements

STDIN	USS standard input stream	
STDOUT	USS standard output stream	
STDERR	USS standard error stream	
STDENV	USS environment variables	
STDPARM	BPXBATCH parameters	

```
//STEP1 EXEC PGM=JVMLDM50, REGION=0M,
// PARM='javaprogramname'
//INPUT DD DSN=some.input.data.set,DISP=SHR
//OUTPUT DD DSN=some.output.data.set,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//STDENV DD *
/etc/profile
APP_HOME=/some/path/name
export JAVA_HOME=/usr/lpp/java/J6.0
export PATH=/bin:"$JAVA_HOME/bin":
LIBPATH="/lib:/usr/lib:$JAVA_HOME/bin:$JAVA_HOME/bin/cl
assic"
export LIBPATH="$LIBPATH":
CLASSPATH="$APP_HOME"
export CLASSPATH="$CLASSPATH":
```

■ Sample JCL to run a Java application using the
JZOS launcher

# Module and Course Summary

# JES, Initiators, Spooler

```
Class A Input Queue
                                                  Order
                                                             Initiator
//JOB7 JOB CLASS=A, PRTY=5
                              JOB1 Pri 15
                                                             JOB1
                              JOB2 Pri 11 Held
                              JOB7 Pri 5
                                                  3
                              JOB3 Pri 4
                                                             Initiator
                              JOB4 Pri 4
                                                             JOB7
                                                                              Spooler
                                                             Initiator
                              Class C Input Queue
                                                             JOB3
                                                  5
                              JOB5 Pri 15
//JOB9 JOB CLASS=C, PRTY=13
                                                  6
                              JOB9 Pri 13
                                                             Initiator
                              JOB6 Pri 5
 8
                                                             JOB4
                              JOB8 Pri 4
```

### Services - Subsystems

### Background

Subsystems run in the background

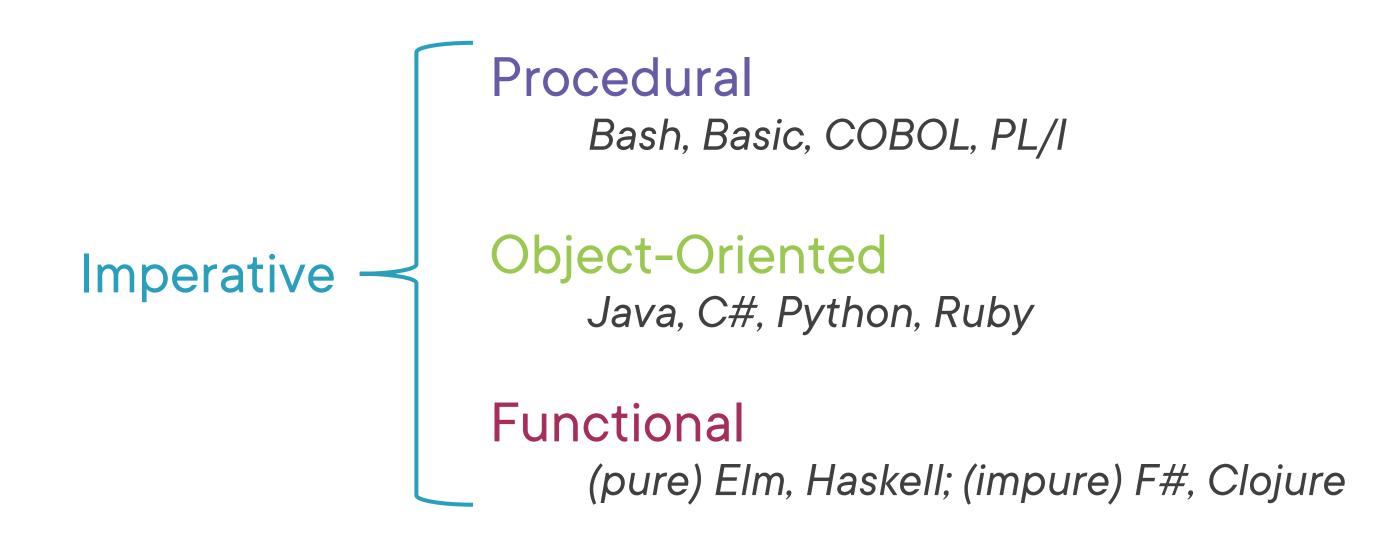
# Longrunning

Subsystems stay active waiting for requests

### Noninteractive

Subsystems are not attached to any user session

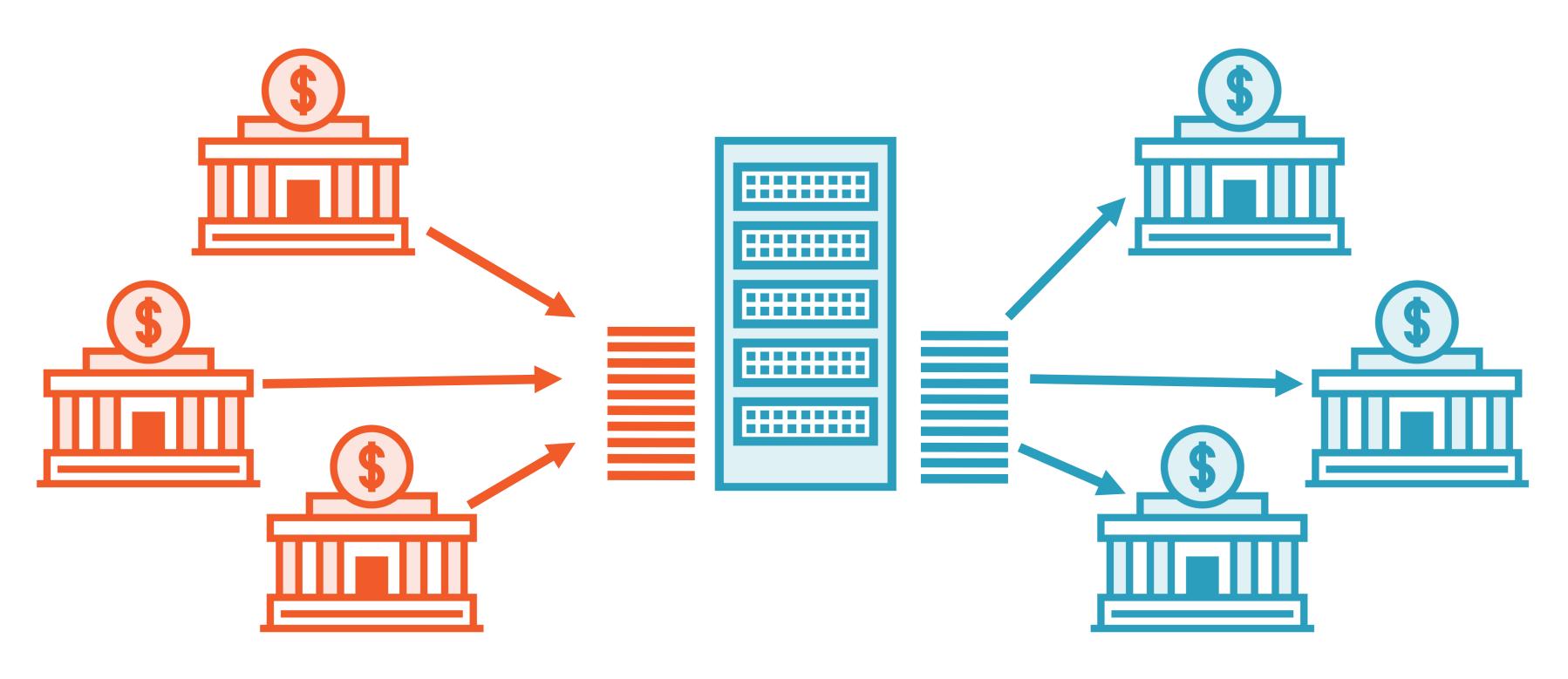
# Types of Languages (Simplified)



**Declarative** 

Prolog, SQL, JCL

# Automated Clearing House (ACH)



### Jobs Contain Steps

Job 1

Step 1

Step 2

Job 2

Step 1

Step 2

Step 3

Step 4

Step 5

Job 3

Step 1

Step 2

Step 3

Step 4

### COBOL

```
DD DSN=input.data.set,DISP=SHR
COBOL
 FILE-CONTROL.
     SELECT PEOPLE-TO-GREET
     ASSIGN TO 'INPUT'
     OPEN INPUT PEOPLE-TO-GREET
```

# Most Frequently-used Data Set Types

### **QSAM**

Queued Sequential Access Method

### GDG

Generation Data Group (GDG)

### **BPAM**

Basic Partitioned Access Method

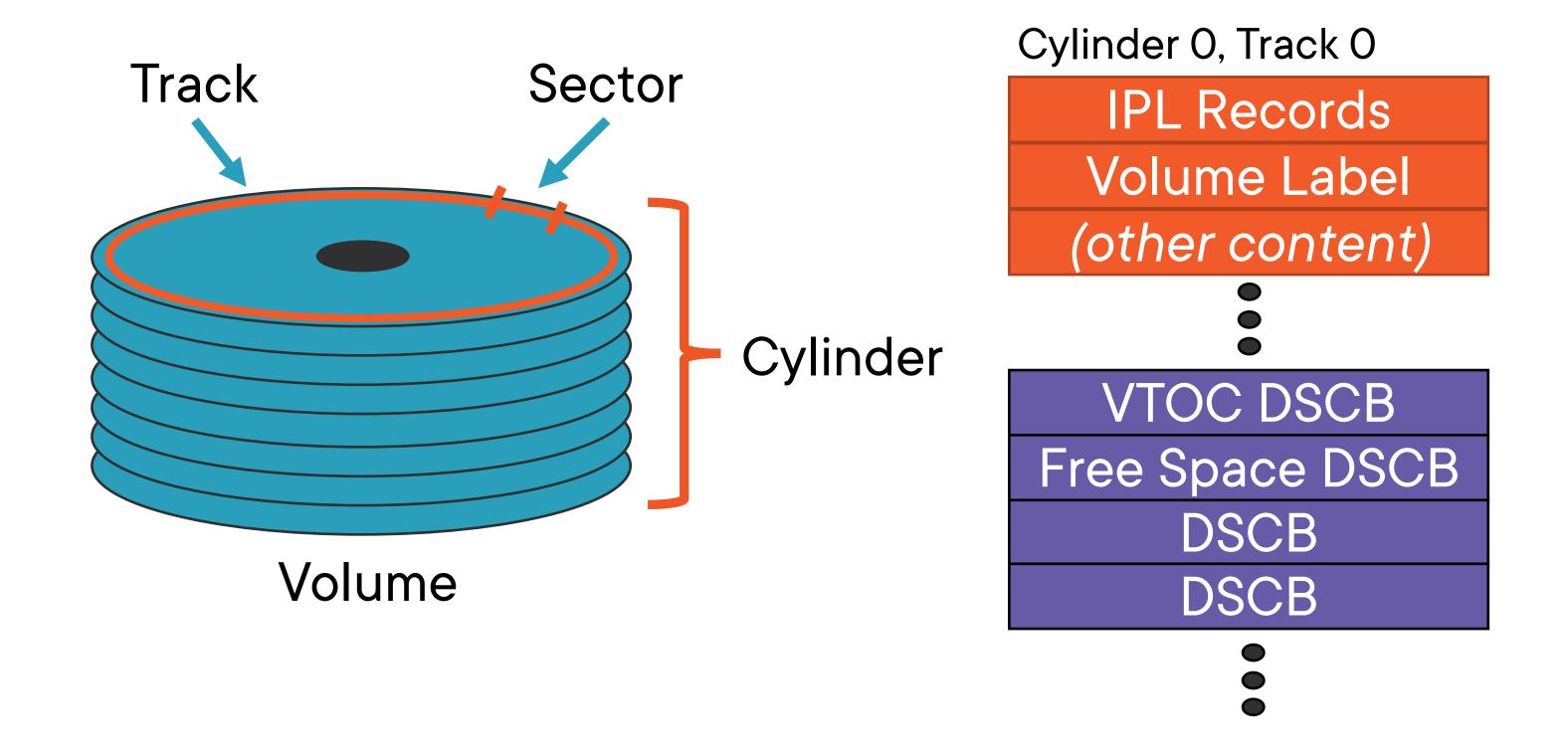
### **VSAM**

Virtual Sequential Access Method

### **HFS File**

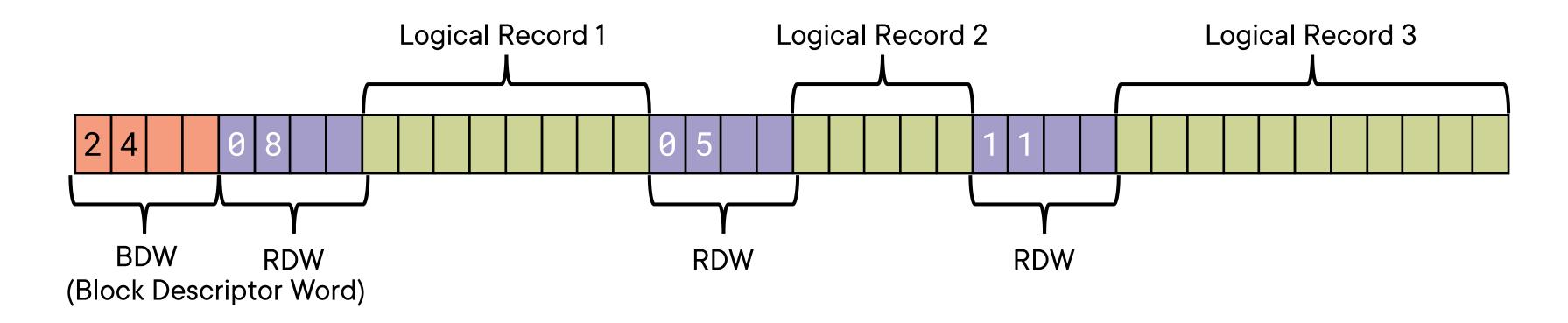
POSIX file (Unix System Services)

# DASD - Direct Access Storage Device



# Variable-length Records, Blocked

DD RECFM=VB, LRECL=8, BLKSIZE=20



Rule: Block size must be at least (average logical record length x number of logical records per block) + 4

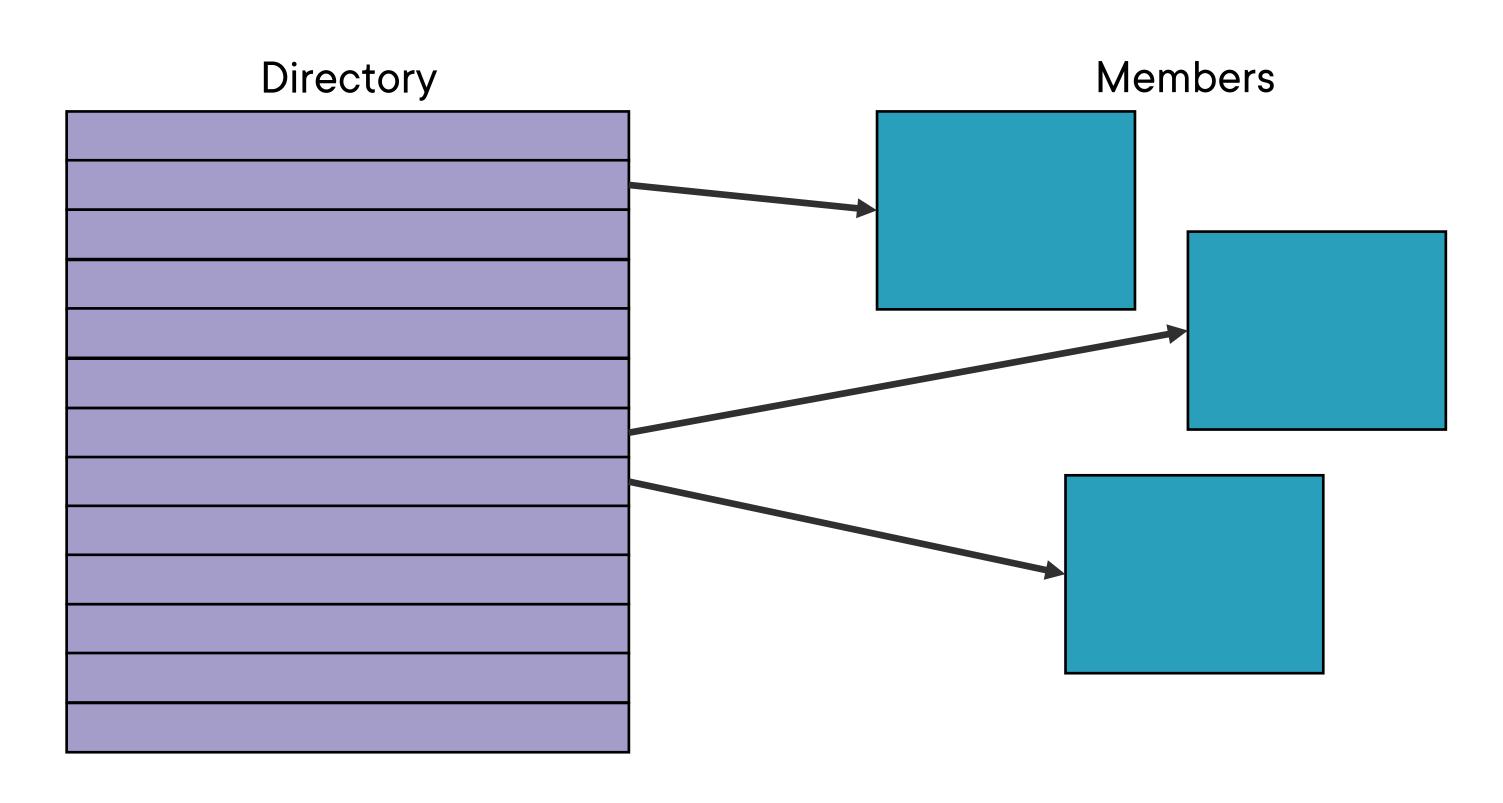
### GDG: Absolute Generation and Version

Relative	Generation	Number
NCIALIVE	<b>Generation</b>	NULLING

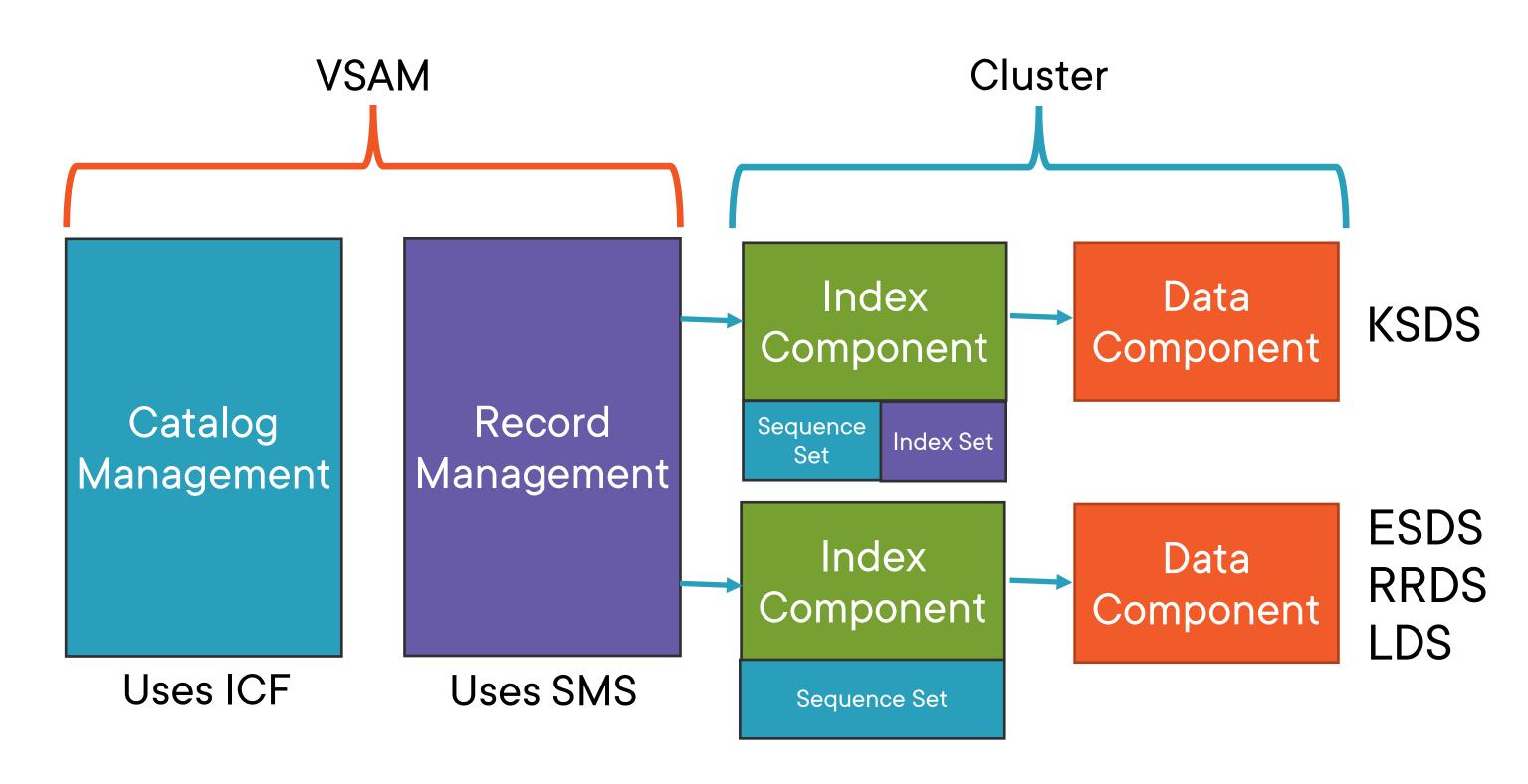
### Absolute Generation & Version Number

DATA.SET.NAME(0) DATA.SET.NAME(-1) DATA.SET.NAME(-2) DATA.SET.NAME(-3) DATA.SET.NAME(-4) DATA.SET.NAME(-4) DATA.SET.NAME(-5) DATA.SET.NAME(-6) DATA.SET.NAME(-7)	DATA.SET.NAME.G0820V00 DATA.SET.NAME.G0819V00 DATA.SET.NAME.G0818V00 DATA.SET.NAME.G0817V02 DATA.SET.NAME.G0816V00 DATA.SET.NAME.G0815V01 DATA.SET.NAME.G0814V00 DATA.SET.NAME.G0813V00
DATA.SET.NAME(-8)	DATA.SET.NAME.G0812V00

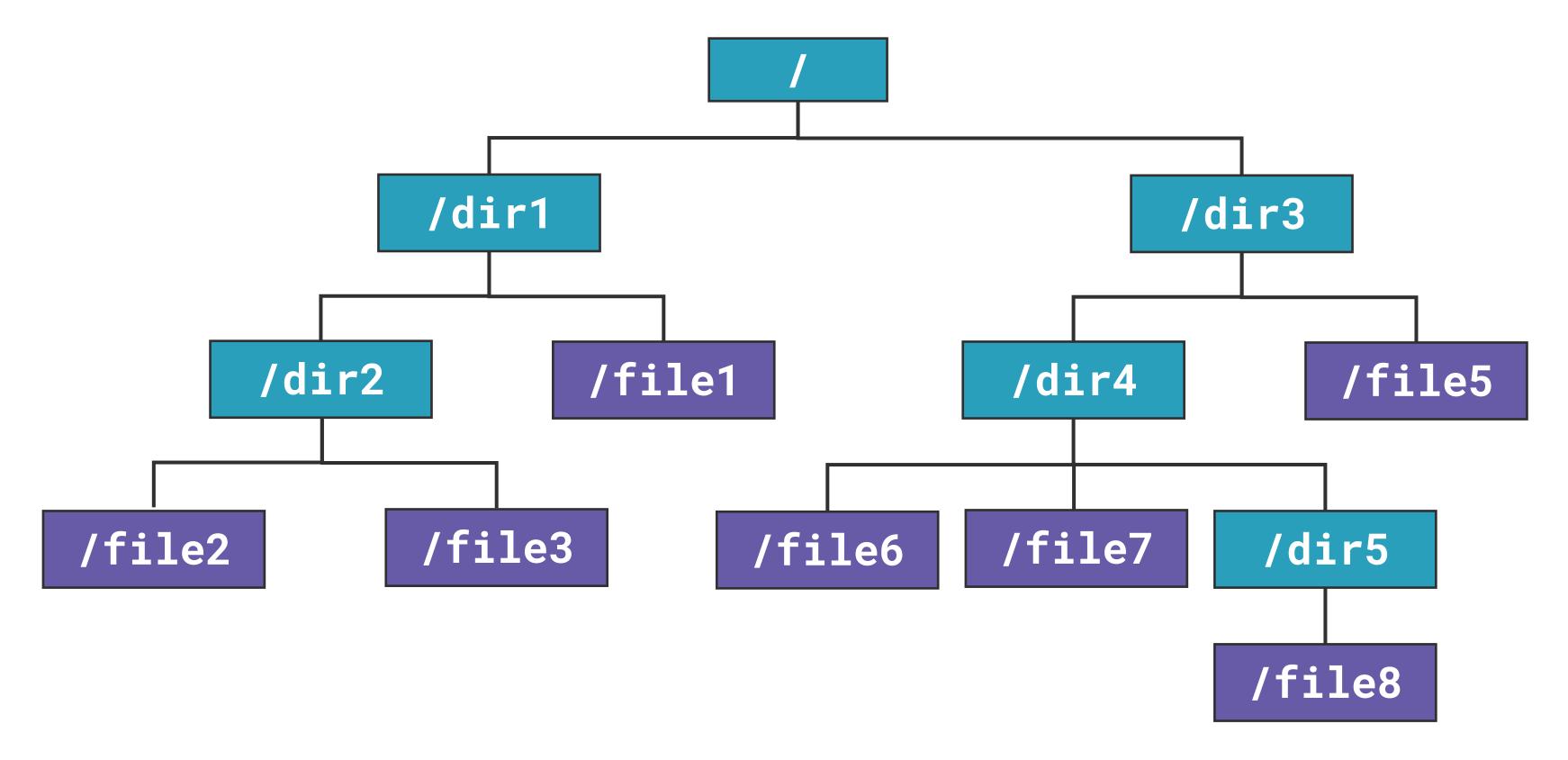
# PDS Directory and Members



# VSAM Components and Clusters



# Hierarchical File Systems on USS



```
//jname JOB ...
//STEP1 EXEC PGM=xxx...
// IF (STEP1.RC = 0) THEN
//ALLOK EXEC PGM=GOODSTF...
// ELSE
// IF (STEP1.RC < 5) THEN
//WARN EXEC PGM=WARNHAND...
// ELSE
// IF (STEP1.RC < 9) THEN
//ERR EXEC PGM=ERRHAND...
// ENDIF
// ENDIF
// ENDIF
// IF ABEND THEN
//CLEANUP EXEC PGM=ABHAND...
// ENDIF
```

**◄ JCL supports IF/THEN/ELSE/ENDIF structures**to control conditional step execution

# Next Steps

Operations

System Configuration and Tuning

Subsystems and Components

Application Development