IMS Version 12

IMS Catalog

Nancy Stein
IMS Advanced Technical Skills
ngstein@us.ibm.com

Information Management software

Disclaimer

Agenda

- IMS Catalog Overview
- IMS Catalog Enablement
- IMS Catalog Lifecycle
- IMS Catalog and ACBLIB Migrations
- Supplement - IMS Catalog Sharing

IMS Catalog Overview

Background

- Customer's need for IMS Metadata not being met
  - No trusted online source for IMS database, program and application metadata information
- Customers can't leverage newer IMS database functions
  - Large scale deployment of IMS Open Database is a challenge
  - JDBC metadata discovery APIs are insufficient
    - can't visualize and report on databases, generate SQL queries, query database contents, generate pureQuery applications, etc…
- Customers limited to database metadata generated by DLIModel Utility
  - Current Java metadata classes are …
    - offline and not guaranteed to be current
    - deployed everywhere
    - hard to maintain and keep in sync with realtime IMS resources
    - not trusted and inflexible
    - not easy to manage for scale for large Open Database solutions
    - lacking application information

Solution

- Provide a trusted, online source for IMS database, program and application metadata information →
  - IMS catalog
  - Store the IMS catalog in an IMS database
- Allow for better scalability of Open Database solutions by removing the local metadata requirement

Benefits

- Business Value
  - Offers a trusted and comprehensive view of IMS database metadata managed by IMS using standard interfaces
    - IMS Universal Drivers (SQL and DLI)
    - Traditional IMS DLI database queries
  - Opens up metadata discovery and exchange for IMS Open Database and the IMS Explorer
  - Permits future IMS integration with IBM tools
    - QMF
    - COGNOS
    - Optim Development Studio
    - Rational Asset Analyzer
    - InfoSphere Data Architect
  - Enables scalable and flexible IMS Open Database solutions
    - Applications no longer need to maintain local Java metadata
    - Applications can reference the IMS catalog

What is Metadata?

B

- Business metadata
  - Business rules, definitions, terminology, glossaries, algorithms and lineage using business language
  - Audience: Business users

T

- Technical metadata
  - Defines source and target systems
    - Table & Column / Segment & Field structures and attributes
    - Derivations and dependencies
  - Audience: Specific tool users, AD, BI, ETL, profiling, modeling

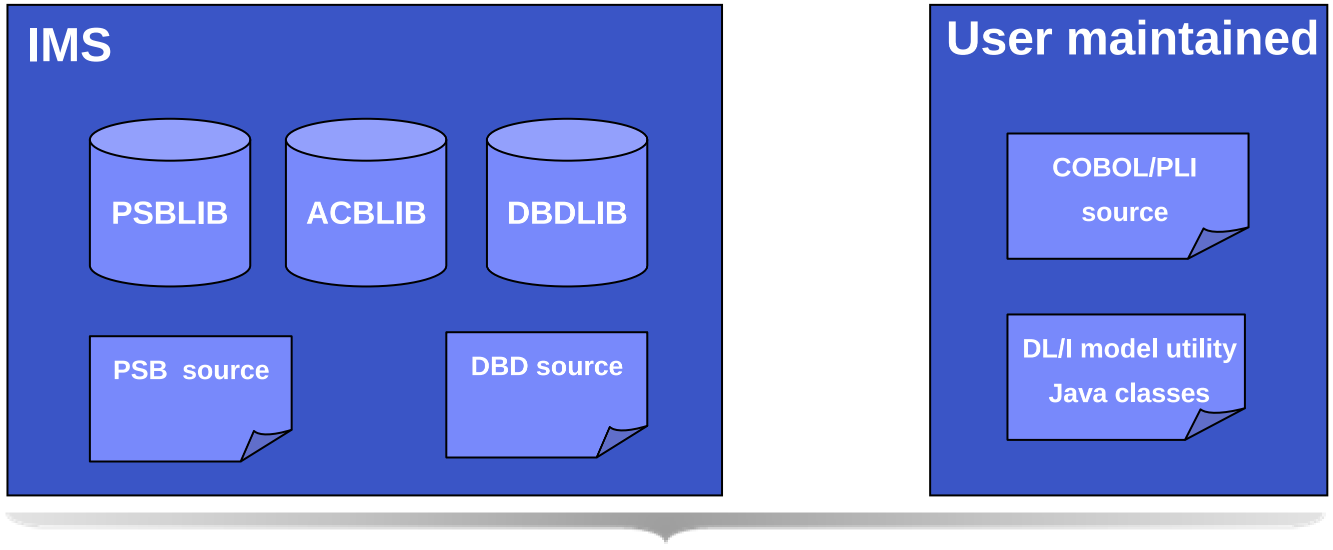
O

- Operational metadata
  - Information about application runtime
    - Frequency, record counts, component by component analysis and other statistics
  - Audience: Operations, management and business users

Literally, “*data about data*” that *describes* your company’s *information* from *multiple perspectives.*

Before the IMS Catalog

- Databases partially defined in the IMS DBD
  - Only key/searchable fields needed by applications
  - Remaining segment data is not defined
- Remaining database definition is within Applications
  - COBOL COPYBOOKs and PL/I INCLUDEs map all the segment data
  - Applications can have different mappings for one segment



IMS Metadata

With the IMS Catalog

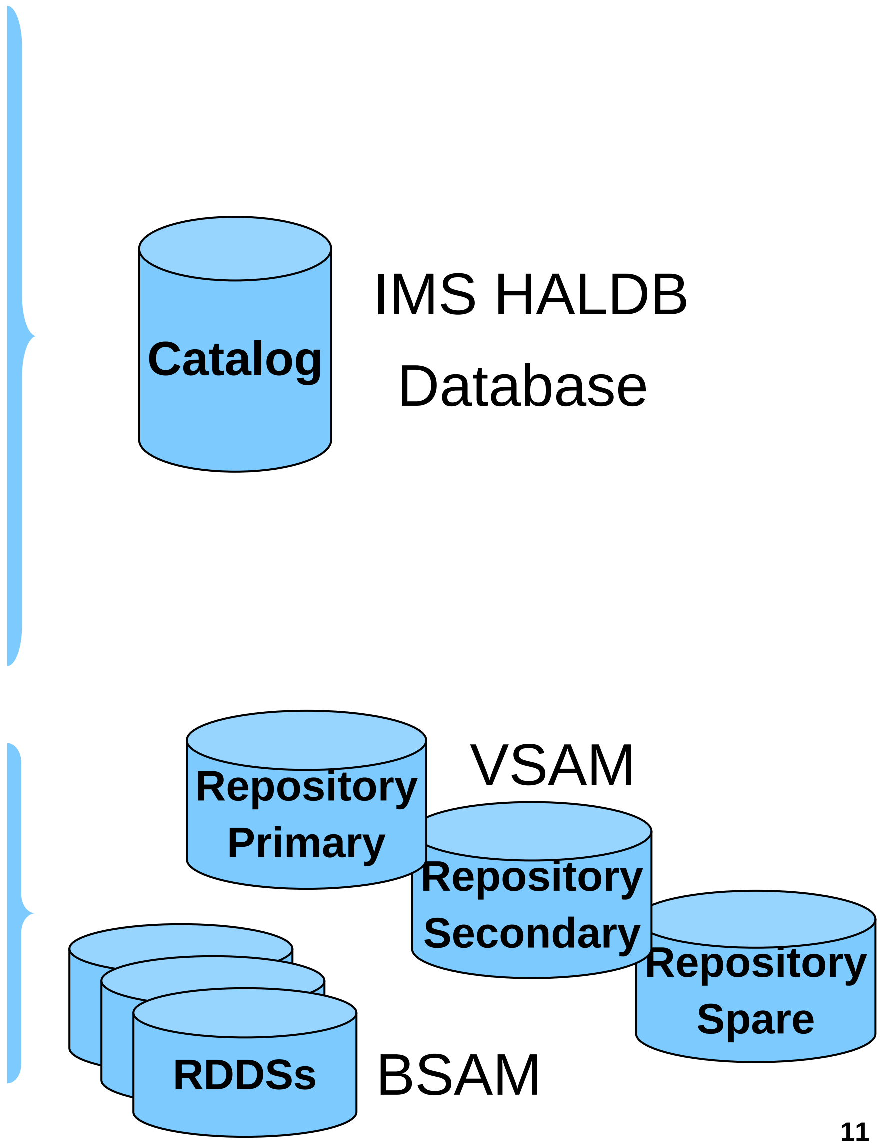
- Database and program resources defined to an IMS system and relevant application information can be stored as metadata in an IMS catalog
  - Databases, fields, segments, data types, mappings and more …
- The IMS catalog is updated when you create, alter or delete IMS resource or application information
- Updates to the IMS catalog are done only via integrated IMS processes

  - Catalog Populate Utility (initial catalog load and member update)

  - PSBGEN, DBDGEN and ACBGEN
- Catalog metadata can be used to:
  - Enhance understanding of the data
  - Improve consistency of the data
  - Improve impact analysis of data
  - Improve development productivity
  - Improve data governance
- Key component of the overall IMS strategy:
  - Simplification
  - Integration

Types of IMS Technical Metadata and Storage Method

- DB
  - PSB/DBD resources
    - Database structure definitions
    - Physical database definitions
    - Segment definitions
    - Field definitions
  - Application
    - Data types
    - Application defined fields
    - Encodings
    - Redefines
    - User defined types
    - Structures
- TM
  - MODBLKS resources
    - Program definitions
    - Transaction definitions
  - MFS FORMAT resources
  - Application
    - Input/output message definitions



The IMS Catalog Database

- Contains metadata related to an IMS system's databases & programs
  - DBDs and PSBs and Application info
- IMS PHIDAM/OSAM HALDB database
  - Defined with 4 DSGs (Data Set Groups)
- Has one Secondary Index
- Unique feature
  - DBRC use is optional for the IMS Catalog HALDB database
    - **ONLY** HALDB that isn't required to be defined to DBRC
  - IMS can manage allocation/creation of catalog database data sets
    - Uses parameters in the "CATALOG" section of DFSDFxxx PROCLIB member

The IMS Catalog Database

- IMS provides DBD and PSB source code for the Catalog database
- IMS provides object code for the Catalog DBDs and PSBs
- PHIDAM DBD reserved name is DFSCD000
- PSINDEX DBD reserved name is DFSCX000
  - Used to connect DBDs to PSBs that reference them
- PSBs provided to load, read and update the Catalog database
  - DFSCPL00 is used for initial load process
    - Used by the Catalog Populate Utility
  - DFSCP000 (COBOL/HLASM), DFSCP002 (PL/I), and DFSCP003 (PASCAL) are used for read access
  - DFSCP001 is used for update access
    - Used by ACBGEN and Catalog Populate Utility
- Default catalog PCB is DFSCAT00
- ACBGEN required for all catalog PSBs

The IMS Catalog Database

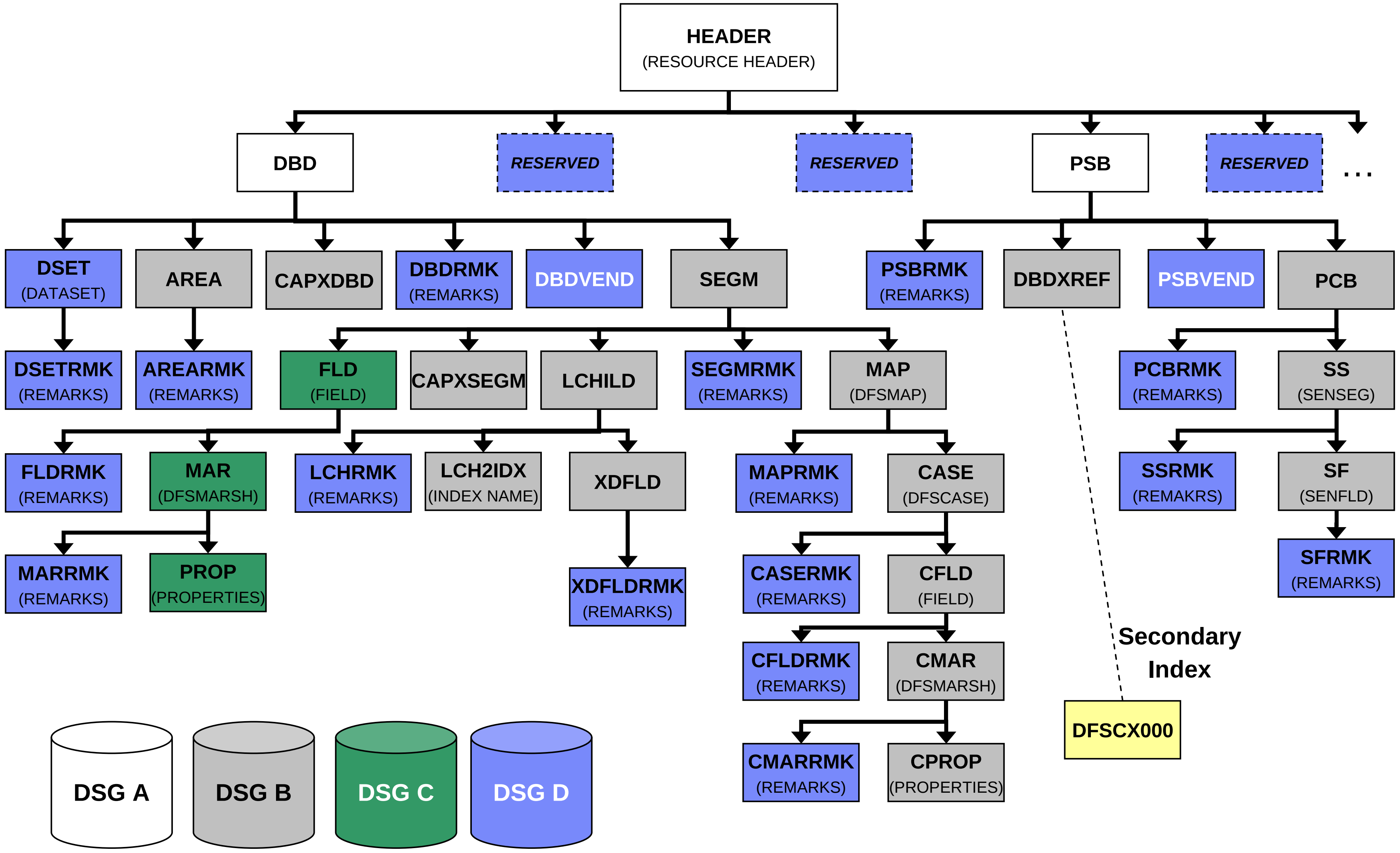
- Catalog database management is required
  - Review/adjust database buffer pool definitions
  - Perform routine management and maintenance on the Catalog database
    - Image Copy, Pointer Checker, Reorg, etc…
  - Catalog database will need to be REORG'd
    - If Catalog database is defined to the RECONs ➔
      - *HALDB OLR non-disruptive reorganization is supported*
    - If Catalog database is not defined to the RECONs ➔
      - *HALDB OLR can't be supported and a reorg utility must be employed*
- Support with existing backup and recovery procedures
  - Image copy, recovery, backout utilities etc…
  - If Catalog database is not defined to the RECONs ➔ recovery is limited
    - Same as non-registered, full function database recovery procedure

The IMS Catalog Database

- Root segment of the Catalog database is a generic resource header
  - Indicates the type of resource ➔ DBD or PSB
    - A dependent of the Root and it's Children are a complete DBD or PSB
      - Multiple versions of a specific resource are supported
        - Most resources are differentiated by their ACBGEN timestamp
        - Logical DBDs and GSAM DBs are differentiated by their DBDGEN timestamp
- Catalog database segments typically correspond to macro statements in the DBD and PSB source
- One segment at the first Child level under both the DBD and PSB segments is available for vendor/customer use
  - DBDVEND
  - PSBVEND

Let's look at the IMS Catalog database physical structure...

Physical Catalog Structure



IMS Catalog Database Restriction

WARNING:

Coexistence with previous IMS releases is not supported !

Access to the IMS Catalog Database from any IMS subsystem, program, utility or client executing or utilizing an IMS release earlier than IMS 12 is not supported !

- For Example:
  - Access to the Catalog database from an IMS 10 or IMS 11 online subsystem is not supported
  - Access to the Catalog database from the IMS Explorer via an IMS 10 or IMS 11 subsystem is not supported
  - Access to the Catalog database from an IMS 10 or IMS 11 DFSDDLT0 utility job is not supported

IMS Catalog Enablement

IMS Catalog Enablement

- Add catalog DBDs and PSBs to your DBDLIB, PSBLIB & ACBLIB
  - Copy DBD and PSB object code from SDFSRESL to your DBDLIB and PSBLIB

```
/CPYCMEM EXEC PGM=IEBCOPY
/SDFSRESL DD  DSN=SDFSRESL,DISP=SHR
/DBDLIB   DD  DSN=MYIMS.DBDLIB,DISP=OLD
/PSBLIB   DD  DSN=MYIMS.PSBLIB,DISP=OLD
/SYSIN    DD  *
      COPY   OUTDD=DBDLIB,INDD=((SDFSRESL,R)),LIST=YES
      SELECT MEMBER=(DFSCD000,DFSCX000)
      COPY   OUTDD=PSBLIB,INDD=((SDFSRESL,R)),LIST=YES
      SELECT MEMBER=(DFSCPL00,DFSCP000,DFSCP001,DFSCP002,DFSCP003)
```

  - ACBGEN the catalog DBD and PSB resources into your ACBLIB

```
//CATACB  EXEC PGM=DFSRRC00,PARM='UPB'
//STEPLIB  DD  DSN=SDFSRESL,DISP=SHR
//DFSRESLB DD  DSN=SDFSRESL,DISP=SHR
//IMS      DD  DSN=MYIMS.DBDLIB,DISP=SHR
//         DD  DSN=MYIMS.DBDLIB,DISP=SHR
//IMSACB   DD  DSN=IMS.ACBLIB,DISP=OLD
//SYSIN    DD  *
         BUILD PSB=(DFSCPL00)
         BUILD PSB=(DFSCP001)
         BUILD PSB=(DFSCP000)
```

N.B. The MODBLKS resources for the Catalog databases and programs do not need to be defined !

IMS Catalog Enablement

- Modify DFSDFxxx PROCLIB Member
  - New CATALOG section(s) for catalog related parameters
    - Single section format <SECTION=CATALOG>
    - Multiple section format <SECTION=CATALOG***imsid***>
      - Multiple IMS systems sharing one DFSDFxxx PROCLIB member
      - ***imsid*** suffix must be a four character IMS ID
  - CATALOG section parameters
    - CATALOG=**N** | Y
      - Catalog is disabled or enabled
      - If enabled, IMS automatically creates catalog DDIR & PDIRs at IMS startup
    - ALIAS=DFSC | xxxx (**no default value**)
      - Specifies any 1-4 alphanumeric value used as a Catalog database name prefix
      - Enables use of non-shared, Catalog databases within an IMSplex
        - Use in a data sharing environment where each IMS has its own Catalog database and all are registered in a single set of RECONs
        - At runtime, the alias Catalog database names are dynamically replaced with internal database names **DFSC**D000 and **DFSC**X000
      - For standalone IMS system – use ''**DFSC**'' which is the standard Catalog database name prefix ➔ **DFSC**D000 and **DFSC**X000

IMS Catalog Enablement

- DFSDFxxx PROCLIB Member
  - CATALOG section parameters (continued)
    - Information used by Catalog Populate Utility to automatically allocate the Catalog database data sets
      - DATACLAS
        - Optional data class for SMS managed data sets
      - MGMTCLAS
        - Optional management class for SMS managed data sets
      - STORCLAS
        - Required storage class for SMS managed data sets
      - IXVOLSER
        - Volume serial number for primary and secondary catalog indices
        - Required for non-SMS managed data sets
      - SPACEALLOC
        - Free space % (0 to 9999) added to the IMS-computed size of the primary & secondary allocations
      - SMSVOLCT
        - Number of volumes (1-20) created by the Catalog Populate utility for SMS-managed

IMS Catalog Enablement

- DFSDFxxx PROCLIB Member
  - CATALOG section parameters (continued)
    - RETENTION=(MAX=**2** | nnn) or RETENTION=(PERIOD=**0** | nnn)
      - Specifies retention schedule for metadata in the IMS catalog
      - By default IMS keeps only two copies of the DBD or PSB in catalog
      - MAX=**2** | nnn
        - Maximum number of versions of a DBD or PSB to be stored before they are replaced first-in first-out
      - PERIOD=**0** | nnn
        - Maximum number of days a version of a DBD or PSB is to be stored before it can be replaced
        - Metadata versions older than the specified retention period are not automatically deleted, but available for removal when new version of metadata is added
        - Default value of “0” disables this feature

IMS Catalog Enablement

- DFSDFxxx PROCLIB Member
  - Enabling the IMS catalog for IMS batch processing
    - Specify DFSDFxxx member on job EXEC parm
      - Requires JCL change to implement
    - User Exit Routine
      - Optional Catalog Definition user exit routine, DFS3CDX0
      - Alternative to specifying DFSDFxxx member through job JCL
      - Available if users cannot or choose not to modify job JCL

IMS Catalog Enablement

- Definition of the HALDB structure
  - Partitioning of the catalog is the users responsibility
    - Minimum of 1 partition is required
    - Last partition must be able to contain the highest-key PSB record
    - Catalog HALDB uses the high-key selection method
      - No use of Partition Selection Exit is allowed
  - Catalog Database Definition
    - For systems that use DBRC
      - Catalog database can be defined to the RECONs with the DBRC utility and commands
    - For systems that do not use DBRC
      - Catalog database must be defined to the Catalog Partition Definition data set using the Catalog Partition Definition Data Set utility, DFS3UCD0
    - If an ALIAS is used in the CATALOG sections of the DFSDFxxx member, each alias Catalog database must be defined

IMS Catalog Enablement

- Using DBRC DSPURX00 utility and commands to define the Catalog database to the RECONs

```
//DEFCAT EXEC PGM=DSPURX00
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSPRINT DD SYSOUT=*
//IMS DD DSN=IMS.DBDLIB,DISP=SHR
//SYSIN DD *
 INIT.DB DBD(DFSCD000) TYPHALDB SHARELVL(3)
 INIT.PART DBD(DFSCD000) PART(DFSCD01) –
    DSNPREFX(dsnprefix.DFSCD000) –
    BLOCKSZE(4096) –
    KEYSTRNG(X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF')
 INIT.DB DBD(DFSCX000) TYPHALDB SHARELVL(3)
 INIT.PART DBD(DFSCX000) PART(DFSCX01) –
    DSNPREFX(dsnprefix.DFSCX000) -
    KEYSTRNG(X'FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF') /*
```

  – If using a catalog alias prefix, replace ***DFSC*** in the database and partition names for the catalog and the catalog secondary index with the four character ALIAS name prefix

  – You might need to define multiple alias name databases to the RECONs

IMS Catalog Enablement

- Using the Catalog Partition Definition Data Set utility, DFS3UCD0, to define the Catalog database (for systems that do not use DBRC)

```
//S1 EXEC PGM=DFS3UCD0,REGION=0M
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSRESLB DD DSN=IMS.SDFSRESL,DISP=SHR
//DFSHDBSC DD DSN=...,DISP=
//SYSPRINT DD SYSOUT=*
//IMS DD DSN=IMS.DBDLIB,DISP=SHR
//SYSIN DD *
 HALDB=(NAME=DFSCD000)
 PART=(NAME=DFSCD000,PART=partitionname,
         DSNPREFX=dsnprefix,
         KEYSTRNG=keystring)
 HALDB=(NAME=DFSCX000)
 PART=(NAME=DFSCX000,PART=partitionname,
         DSNPREFX=dsnprefix,
         KEYSTHEX=FFFFFFFFFFFFFFFF) /*
```

Unregistered Catalog database

  – Catalog Partition Definition data set is populated with the information specified in the HALDB and PART control cards
    - RECON-like information for catalog database partition definition and structure
  – The name DFSCD000 in the HALDB and PART statements contains the default catalog prefix ***DFSC***. If your catalog uses an alias name prefix, substitute it in the JCL

IMS Catalog Enablement

- After Catalog database is defined in Catalog Partition Definition Data Set
  - Identify unregistered Catalog database names
    - UNREGCATLG parameter in the DATABASE section of the DFSDF*xxx* member

```
/******************************************************************/
/* Database Section */
/******************************************************************/
<SECTION=DATABASE>
UNREGCATLG=(DFSCD000,DFSCX000) /* Unregistered IMS catalog DB     */
/******************************************************************/
/*                                                                */
/******************************************************************/
```

  - If using an alias name prefix, replace ***DFSC*** in the UNREGCATLG database names with the four character alias name prefix
  - Limitations of using an unregistered Catalog database
    - NO IMS Data Sharing support
    - NO OLR support
    - NO partition definition change support
      - User must rebuild catalog partitions
    - Manual recovery required for unregistered Catalog databases

IMS Catalog Enablement

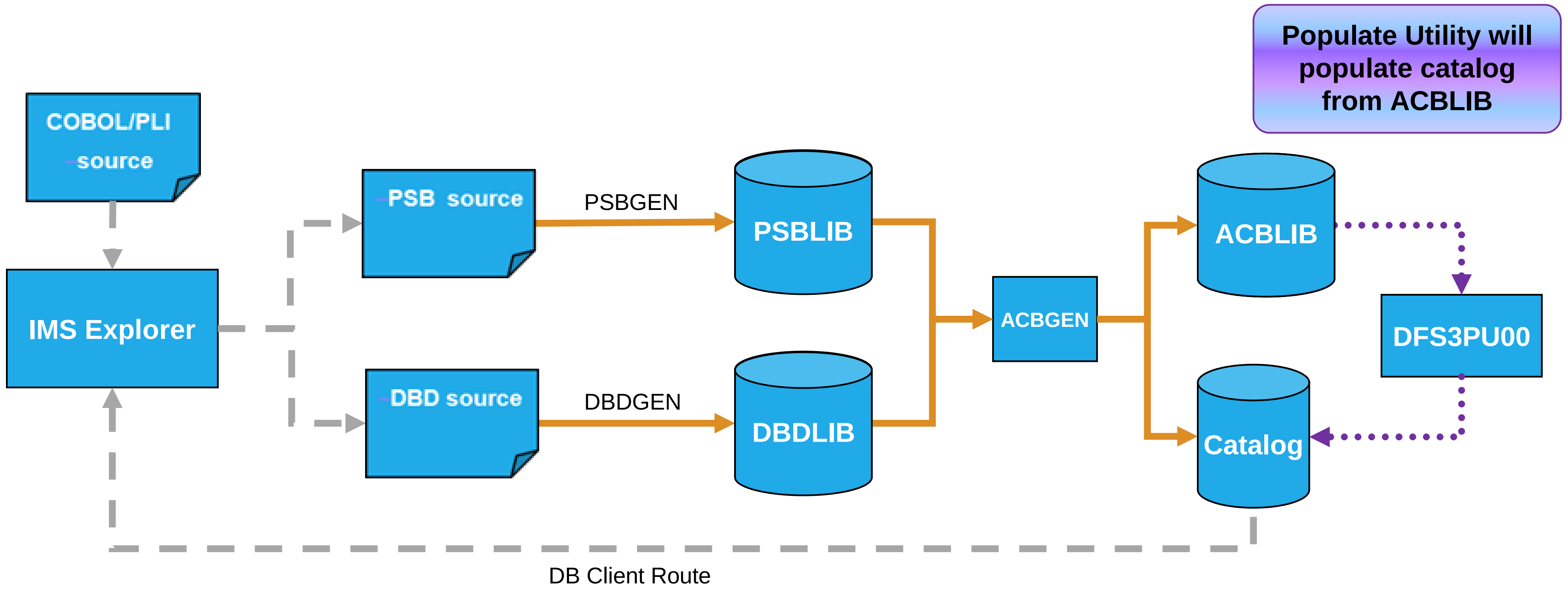
- After Catalog database is defined in Catalog Partition Definition Data Set
  - Create a new DFSMDA dynamic allocation member for the Catalog Partition Definition data set

```
//DYNALOC JOB
//*
//STEP EXEC IMSDALOC
//SYSIN DD *
   DFSMDA TYPE=INITIAL
   DFSMDA TYPE=CATDBDEF,DSNAME=dsn
   DFSMDA TYPE=FINAL
   END
/*
```

  - ***dsn*** is the name of the Catalog Partition Definition data set
    - Catalog Partition Definition data set was allocated in the DFS3UCD0 utility on the DFSHDBSC DD
  - Dynamically allocate Catalog Partition Definition data set in any IMS job step

IMS Catalog Lifecycle

IMS Catalog Lifecycle



- ACBGEN will populate ACBLIB and catalog in same UOW
  - Populates ACBLIB with *standard* ACB info and *extended* info
  - Populates the catalog with *extended* info
- Key points
  - Only way to update catalog is via the Populate Utility or ACBGEN process
  - Extended info is acquired via the IMS Explorer
  - Extended info stored in ACBLIB members for recoverability

IMS Catalog Creation

- Populate the IMS Catalog
  - Load the IMS Catalog using IMS Catalog Populate utility, DFS3PU00
    - Each ACB member is decoded, converted to catalog format, loaded into the catalog
  - Reads ACBLIB, DBDLIB and PSBLIB datasets as input
    - Data sets can be concatenated but only first occurrence of an ACB member is used
    - DBDLIB needed for Logical databases and GSAM databases
    - PSBLIB needed to determine which GSAM database go into the catalog
  - Catalog database DBD and PSB segments will have a version and contain a timestamp that matches the ACB member timestamp
    - Used to associate an ACB member with a catalog member
    - Timestamp exceptions
      - DBDGEN timestamp for Logical and GSAM DBs
      - PSBGEN timestamp for GSAM only PSBs

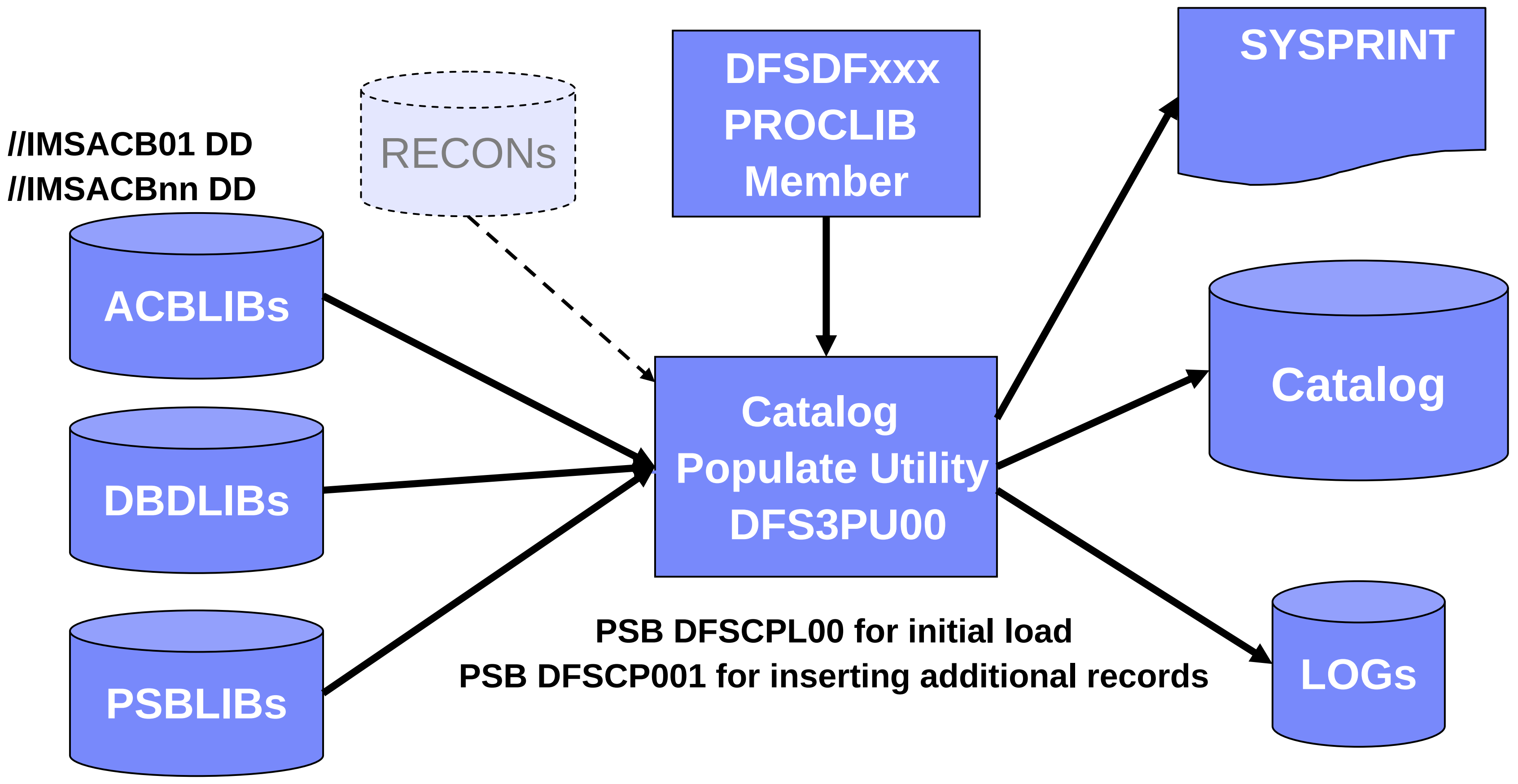
IMS Catalog Creation

- IMS Catalog Populate utility (DFS3PU00)
  - Can run as a typical IMS Batch or BMP job
    - Requires IMS logs for backout / recovery
    - Requires IRLM if catalog is shared and catalog active in an IMS subsystem
      - Business as usual for data sharing
    - BMP mode allows for updates to catalog in non-data sharing environment
    - Requires DBRC if catalog is defined in the RECON
    - If using Catalog Partition Definition Data Set
      - Users responsible to ensure online catalog access has ceased
      - Business as usual for non-registered full function DB
  - Can also be used to insert additional records to an existing Catalog database

IMS Catalog Creation

- Load the IMS Catalog using the new ACB Generation and Catalog Populate utility, DFS3UACB
  - Generates the ACB library members and loads the IMS catalog metadata in the same job step
  - Not recommended if ACBLIB is already valid
    - No need to recreate the ACBLIB, just populate the catalog

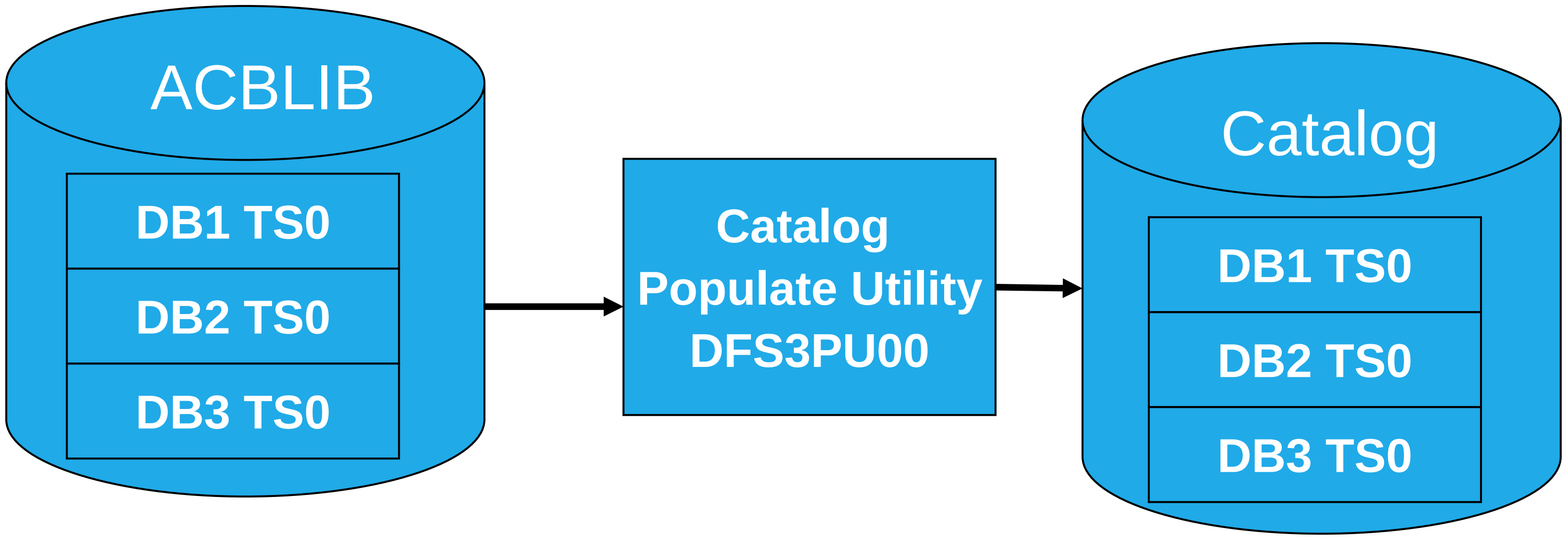
Catalog Populate Utility – DFS3PU00



PSB DFSCPL00 for initial load
PSB DFSCP001 for inserting additional records

Take an Image Copy of the Catalog database after the initial populate !

IMS Catalog Members After Populate Utility

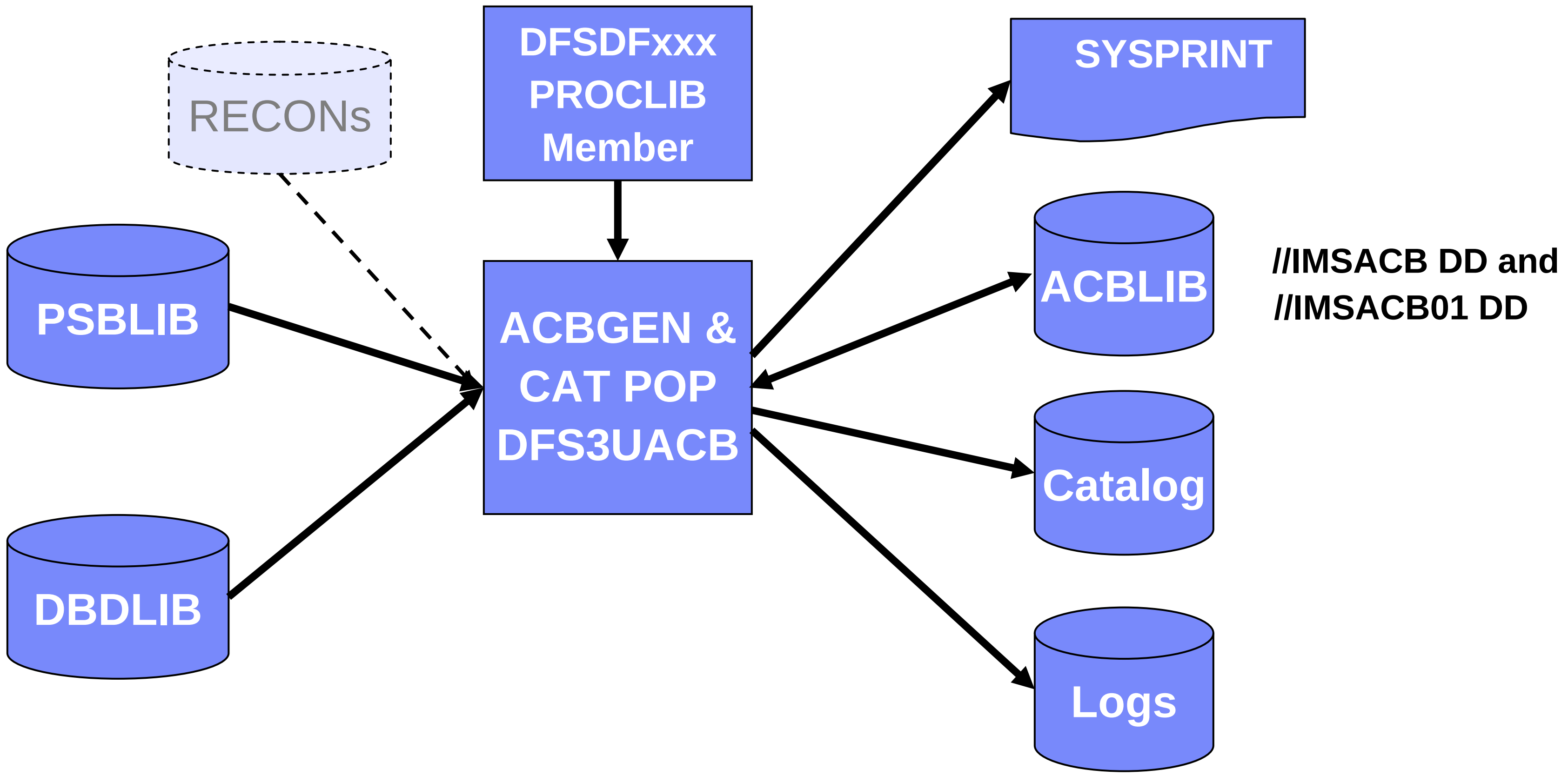


- There are 3 members in an ACBLIB
- Run the Catalog Populate Utility, DFS3PU00
- The catalog members will have a timestamp TS0
- This is the ACB member timestamp

Adding IMS Catalog Metadata

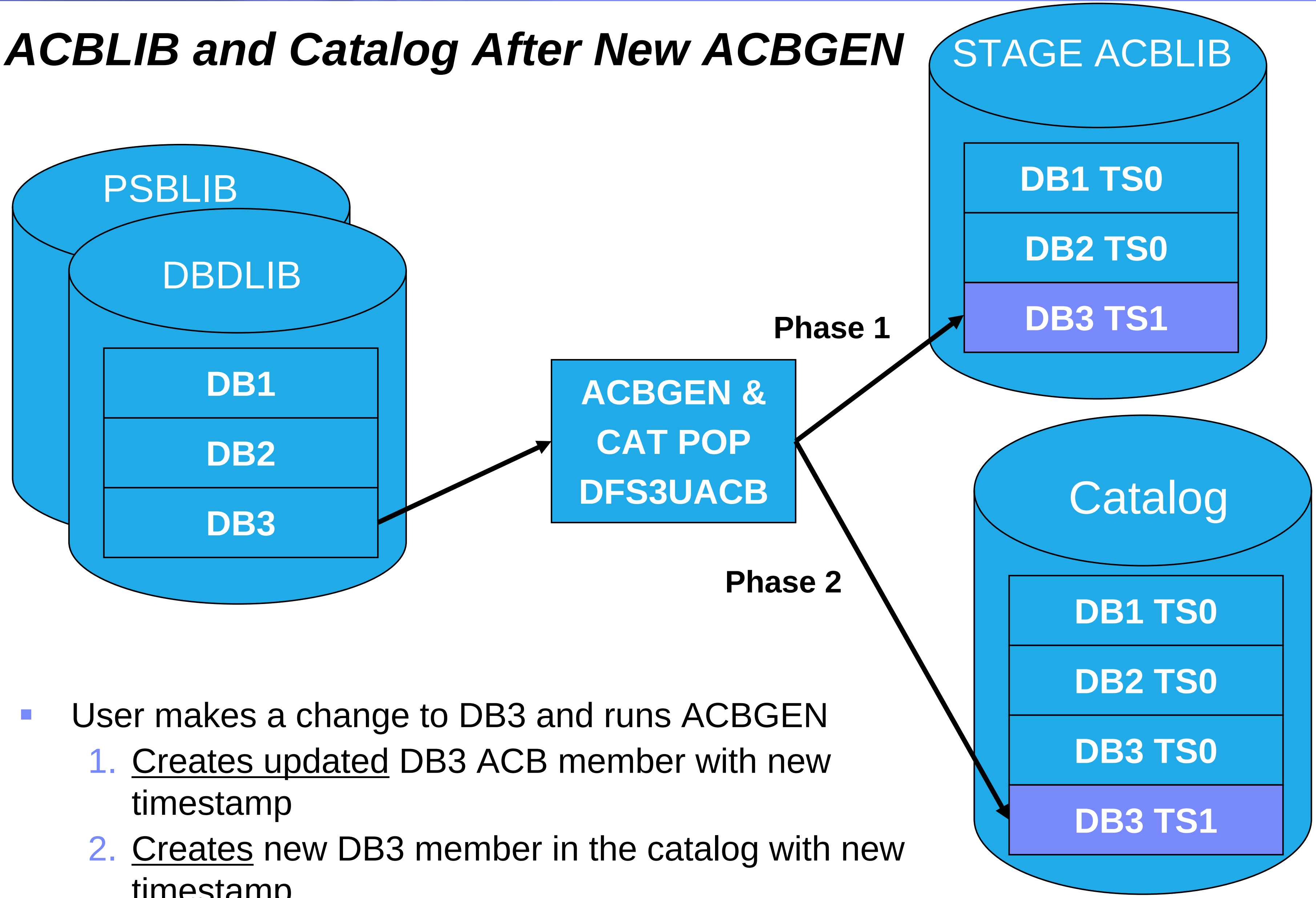
- New ACB Generation and Catalog Populate utility, DFS3UACB
  - Replaces existing ACBGEN Utility, DFSUACB0, if IMS catalog enabled
  - Generate ACBLIB member and create catalog metadata in a ***single job step***
    - Phase 1 - ACBGEN
      - DBDLIB and PSBLIB members used as input
      - Validation is unchanged
      - ACB member is written to ACBLIB with new ACBGEN timestamp
    - Phase 2 – IMS catalog update
      - Generated ACB is decoded, converted to catalog format, loaded into the catalog
        - DBD and PSB metadata created and inserted
        - Corresponding ACB member timestamp saved as timestamp in catalog DBD and PSB segments
        - ensures validity and consistency of ACBLIB and catalog
- New ACBGEN, DFS3UACB, and new Catalog Populate, DFS3PU00, utilities are the only updaters of the IMS catalog
  - IMS online and IMS batch regions will never update catalog data
  - IMS online and IMS batch regions will only retrieve data from the catalog

IMS Catalog Additions Via New ACBGEN & Populate Utility



- New integrated ACBGEN process includes update to the IMS catalog
- DFSDFxxx PROCLIB member has the catalog information
- DFSMDA member used to dynamically allocate the catalog datasets

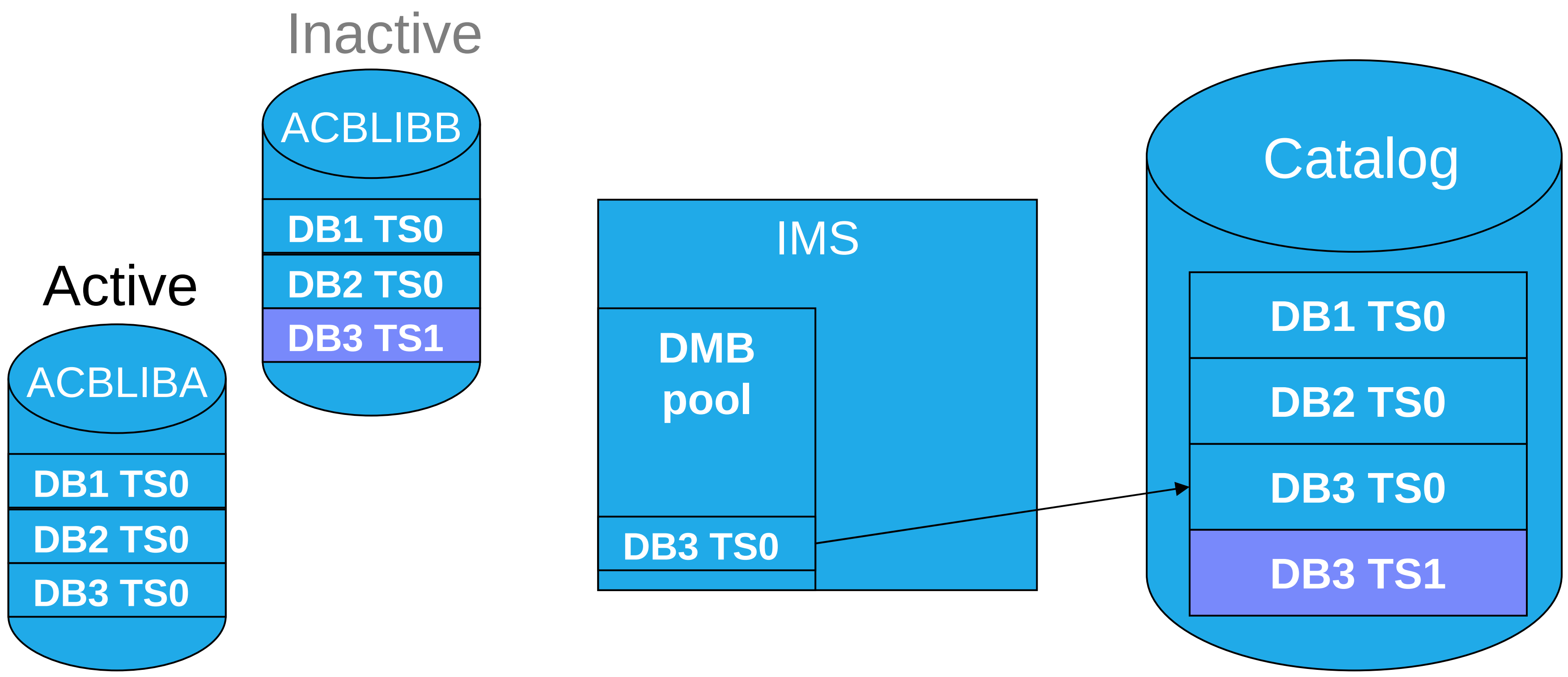
ACBLIB and Catalog After New ACBGEN



- User makes a change to DB3 and runs ACBGEN
  1. Creates updated DB3 ACB member with new timestamp
  2. Creates new DB3 member in the catalog with new timestamp

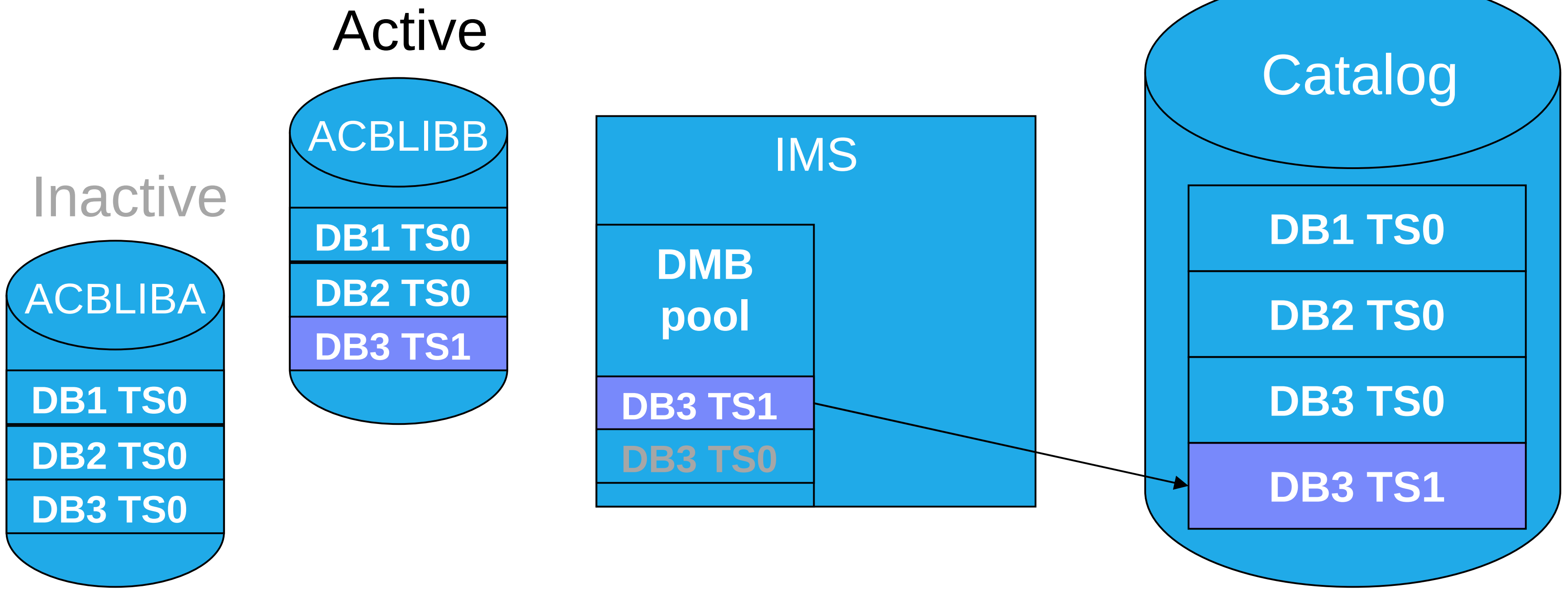
Catalog Member Timestamps

- Java application request is made to read DB3
  1. IMS determines active DB3 ACB member has timestamp TS0
  2. Internal DL/I call issued to retrieve member DB3 from IMS catalog
  3. IMS retrieves catalog member DB3 with timestamp TS0



Catalog Member Timestamps

- Initiate OLC to switch from ACBLIBA to ACBLIBB
  - Activates DB3 ACB with timestamp TS1
- Java application request is made to read DB3
  1. IMS determines active DB3 ACB member has timestamp TS1
  2. Internal DL/I call issued to retrieve member DB3 from IMS catalog
  3. IMS retrieves catalog member DB3 with timestamp TS1



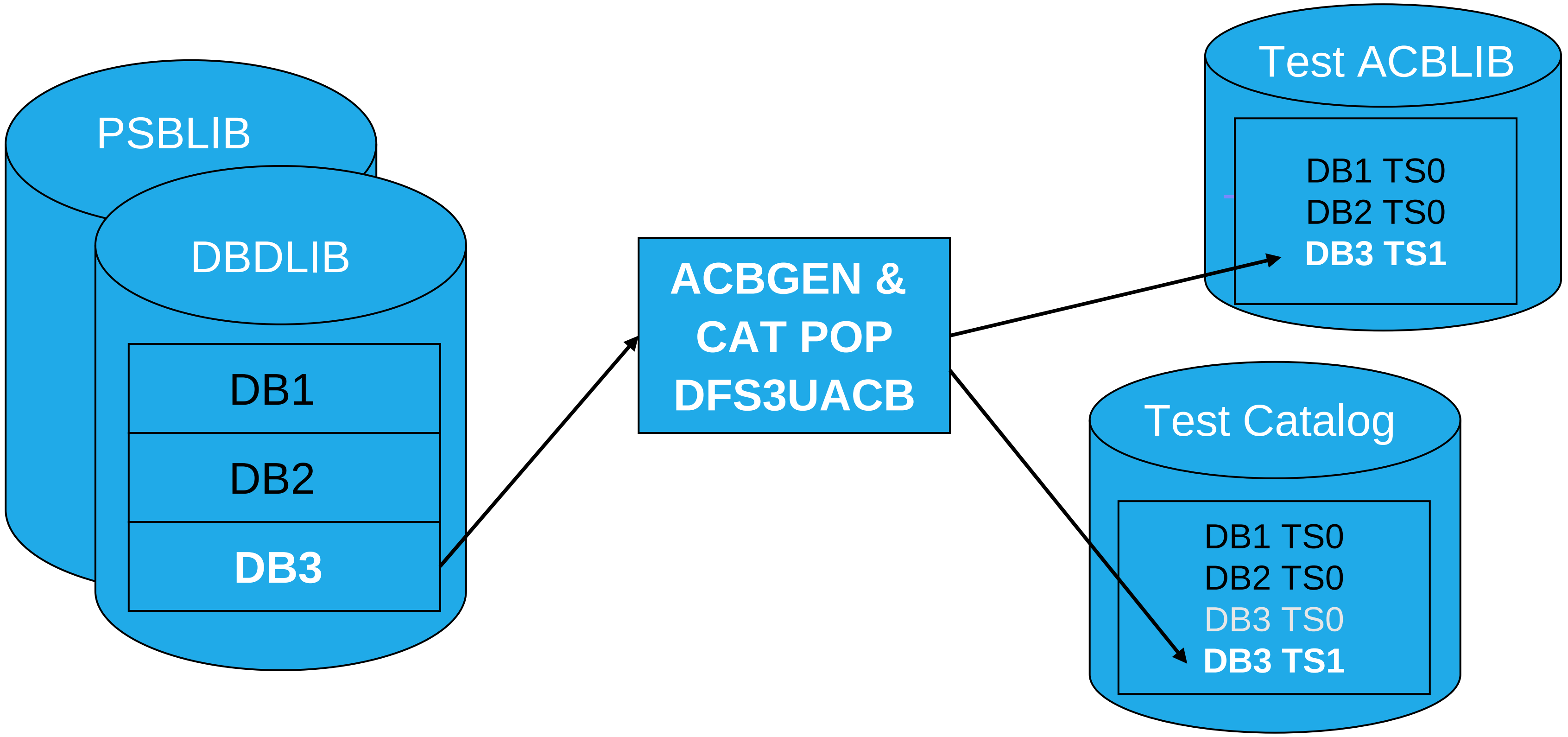
IMS Catalog and ACBLIB Migrations

Catalog and ACBLIB Migration from Test to Production

- Migration process for changed resources does not change significantly
- Use new ACB Generation and Catalog Populate utility, DFS3UACB
  - DFS3UACB replaces existing ACBGEN Utility DFSUACB0
- When copying ACB members from staging to inactive ACBLIB:
  - Use new IMS Catalog Copy Utility, DFS3CCU0
    - DFS3CCU0 replaces Online Change Copy utility DFSUOCU0
    - Ensures that the catalog and ACBLIB are kept in sync

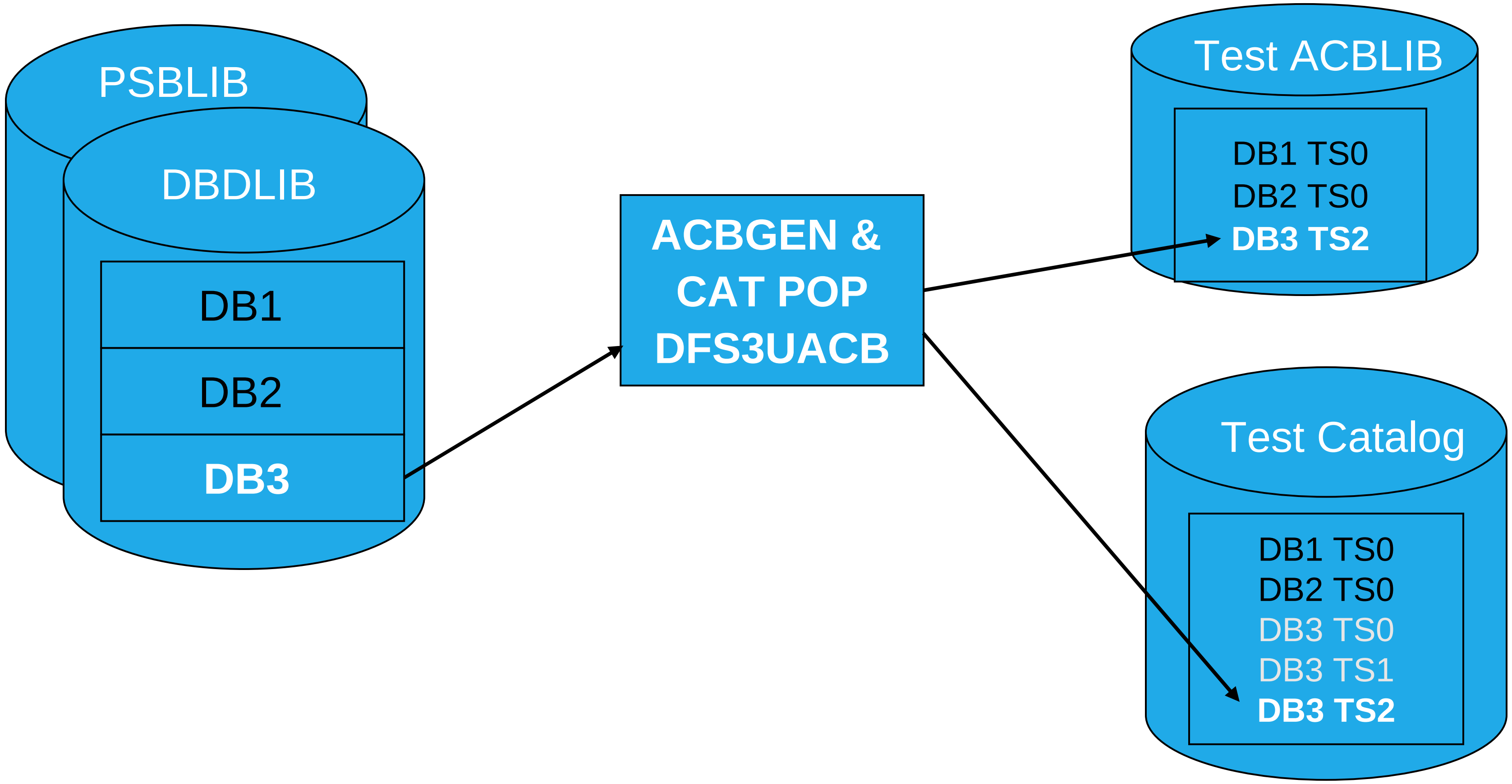
Testing Environment - Step I

- Developer or DBA makes a change to DB3 and runs ACBGEN
  1. Creates updated DB3 ACB member with new timestamp
  2. Creates new DB3 member in the catalog with new timestamp



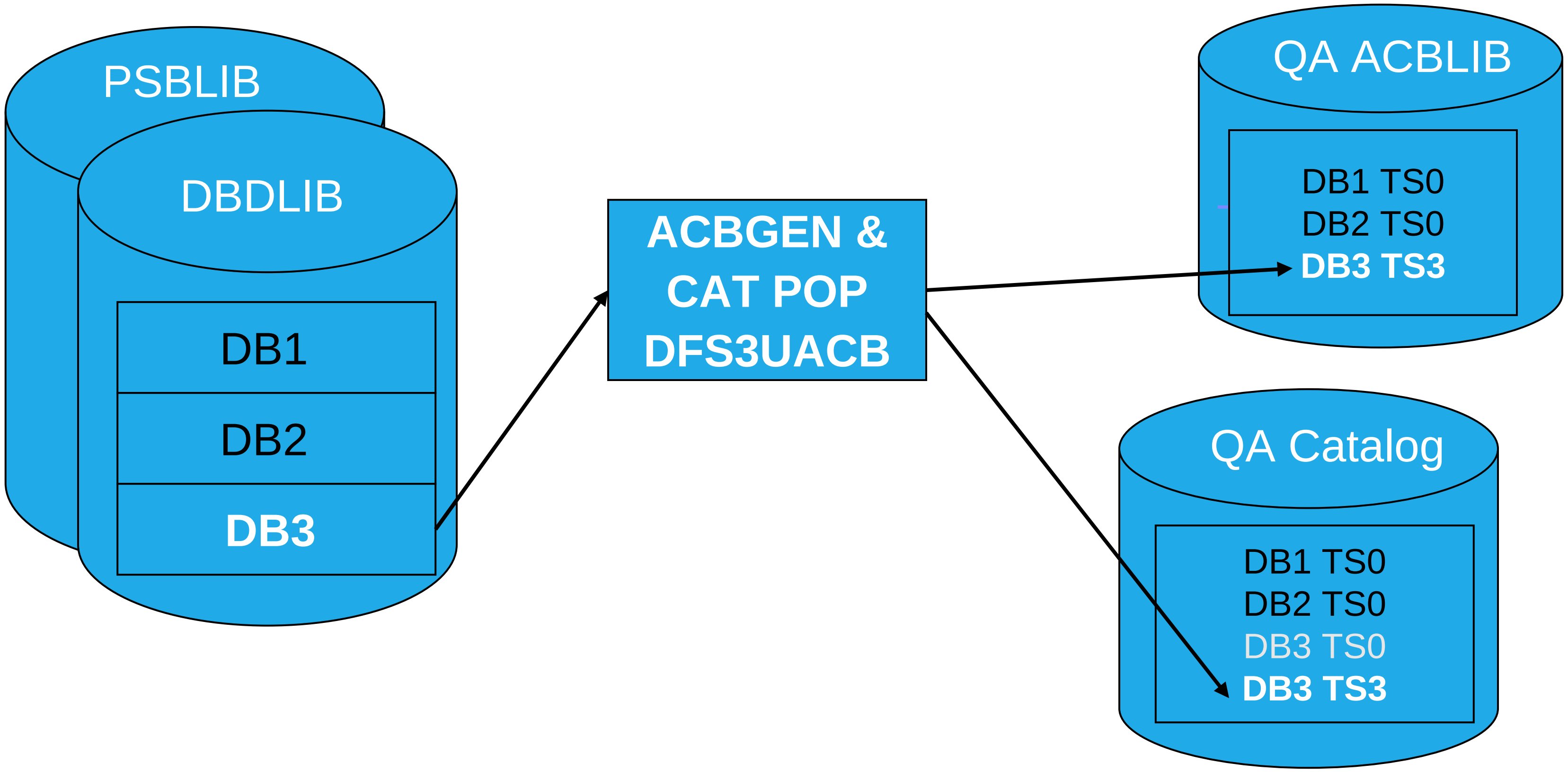
Testing Environment - Step II

- Developer or DBA makes another change to DB3 and runs ACBGEN again
  1. Creates updated DB3 ACB member with new timestamp
  2. Creates another new DB3 member in the catalog with new timestamp



QA Environment - Step III

- Developer testing is completed, ready for User testing in the QA system
  1. Run ACBGEN for DB3 into the QA ACBLIB and the QA catalog
  2. Creates updated DB3 ACB member with new timestamp
  3. Creates new DB3 member in the catalog with new timestamp



Catalog and ACBLIB Migration from QA to Production

- Ready to propagate corresponding ACBLIB members and IMS catalog members between QA and PROD IMS environments
  - Need to copy ACBLIB members and corresponding catalog metadata
    - New Catalog Copy Utility, DFS3CCU0 (recommended method)
    - Catalog Populate Utility, DFS3PU00
- Catalog Copy Utility, DFS3CCU0, can be used for …
  - Migration of resource changes from one IMS environment to another
  - IMS System cloning
  - Maintaining Disaster Recovery system
- Catalog Copy Utility, DFS3CCU0, is a two step process
  - Export step
    - creates an export data set based on catalog member from ACBLIB
  - Import step
    - uses export data set to write to both IMS catalog and ACBLIB
    - single step updates ACBLIB member and catalog metadata

Catalog Copy Utility - DFS3CCU0 Export Step

- Exports member(s) from an input IMS catalog
  - Runs as an IMS Batch or BMP job
- Export process
  - Unloads a single catalog member at a time
    - Cannot export different versions of the same catalog member
  - Uses ACBLIB to get timestamp for a catalog member export
  - If no ACBLIB provided → copy all the catalog members
  - If no ACBLIB member found → copy the catalog member anyway
  - Read catalog member looking for the ACBLIB timestamp match
  - Write catalog member to the export data set
- Input
  - ACBLIB and corresponding IMS catalog
- Output
  - Export data set containing unloaded catalog members
  - Similar format and size as an unload data set

Catalog Copy Utility – DFS3CCU0 Import Step

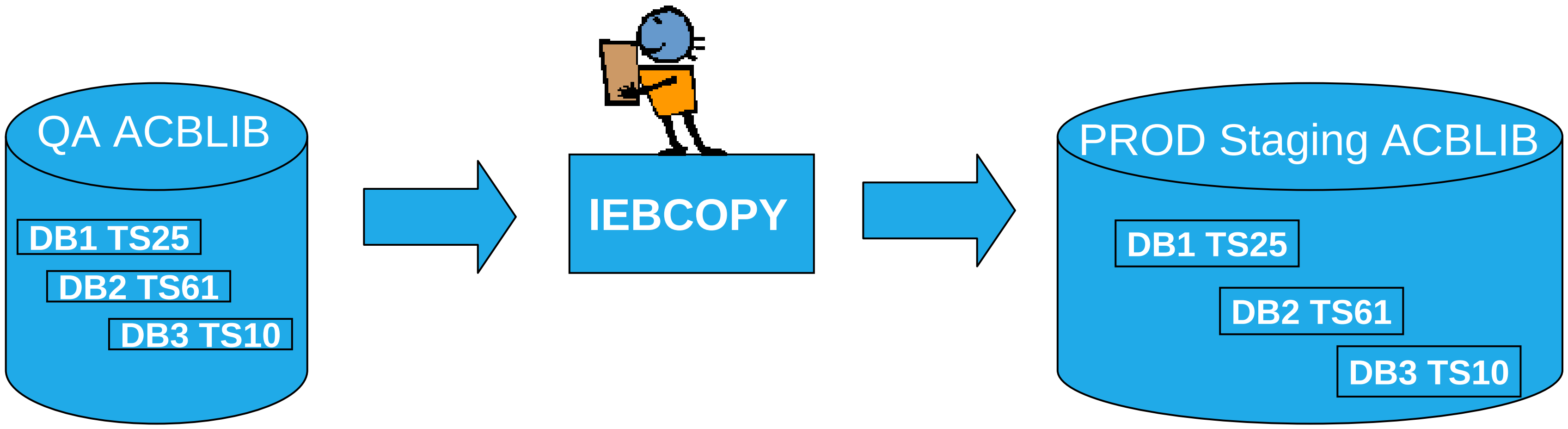
- Imports member(s) to an IMS catalog
  - Runs as an IMS Batch or BMP job
- Import process
  - Write catalog member from export dataset to the IMS catalog
  - Internal IEBCOPY of ACBLIB member that matches the catalog member timestamp
    - If no ACBLIB provided → skip the copy of all catalog members
    - If no matching ACBLIB member found → skip the catalog member copy
- Input
  - Input ACBLIB
  - Export data set containing unloaded catalog members
- Output
  - Output ACBLIB member and corresponding catalog member

Scenario: IMS Resource Migration

- Use the Catalog Copy Utility to migrate resource changes from one IMS environment to another
  - Example: Migrate IMS QA to IMS PROD environment
    - ACBGEN is done only in IMS QA environment
    - PROD Staging ACBLIB is clone of QA ACBLIB
      - same timestamps
    - ACBLIB members will determine which catalog members are copied to the production catalog
  - Process may vary slightly depending on method used for modifying IMS resources
    - OLC, GOLC, MOLC or IMS restart (if not using OLC)

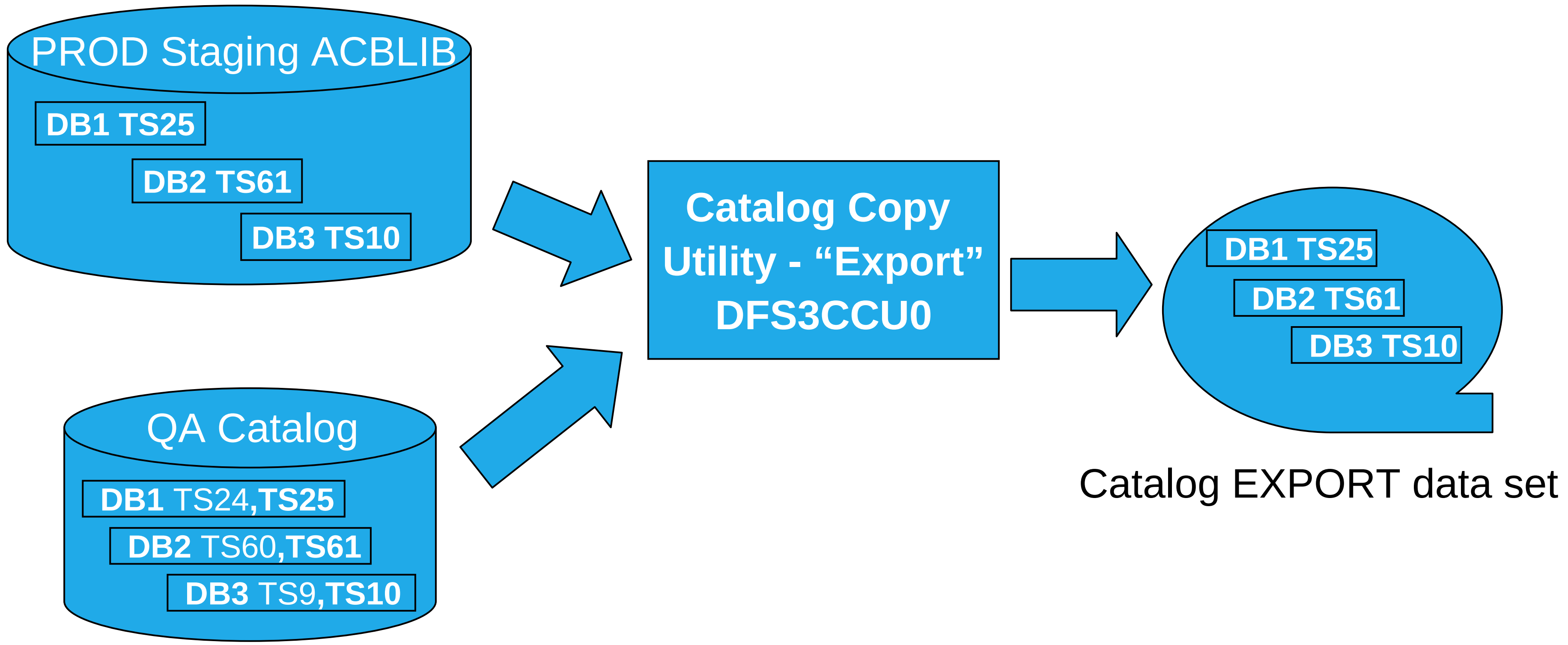
Migration to Production Environment - Step IV

- User testing in the IMS QA system is completed, ready to migrate changes to the IMS PROD system
  1. Copy members from QA ACBLIB to PROD Staging ACBLIB



Migration to the Production Environment - Step IV

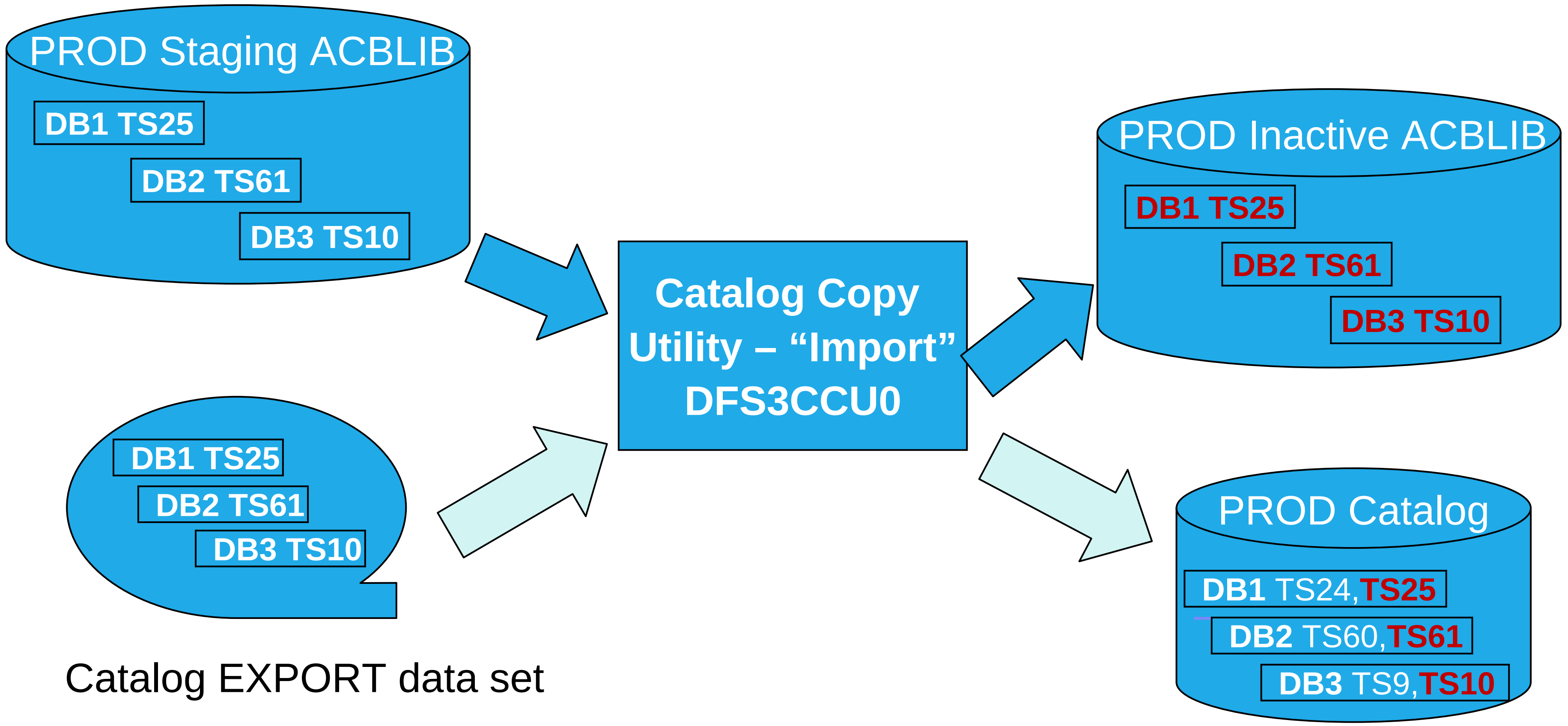
2. Export IMS QA catalog members that have a matching timestamp in PROD Staging ACBLIB



Catalog EXPORT data set

Migration to the Production Environment - Step IV

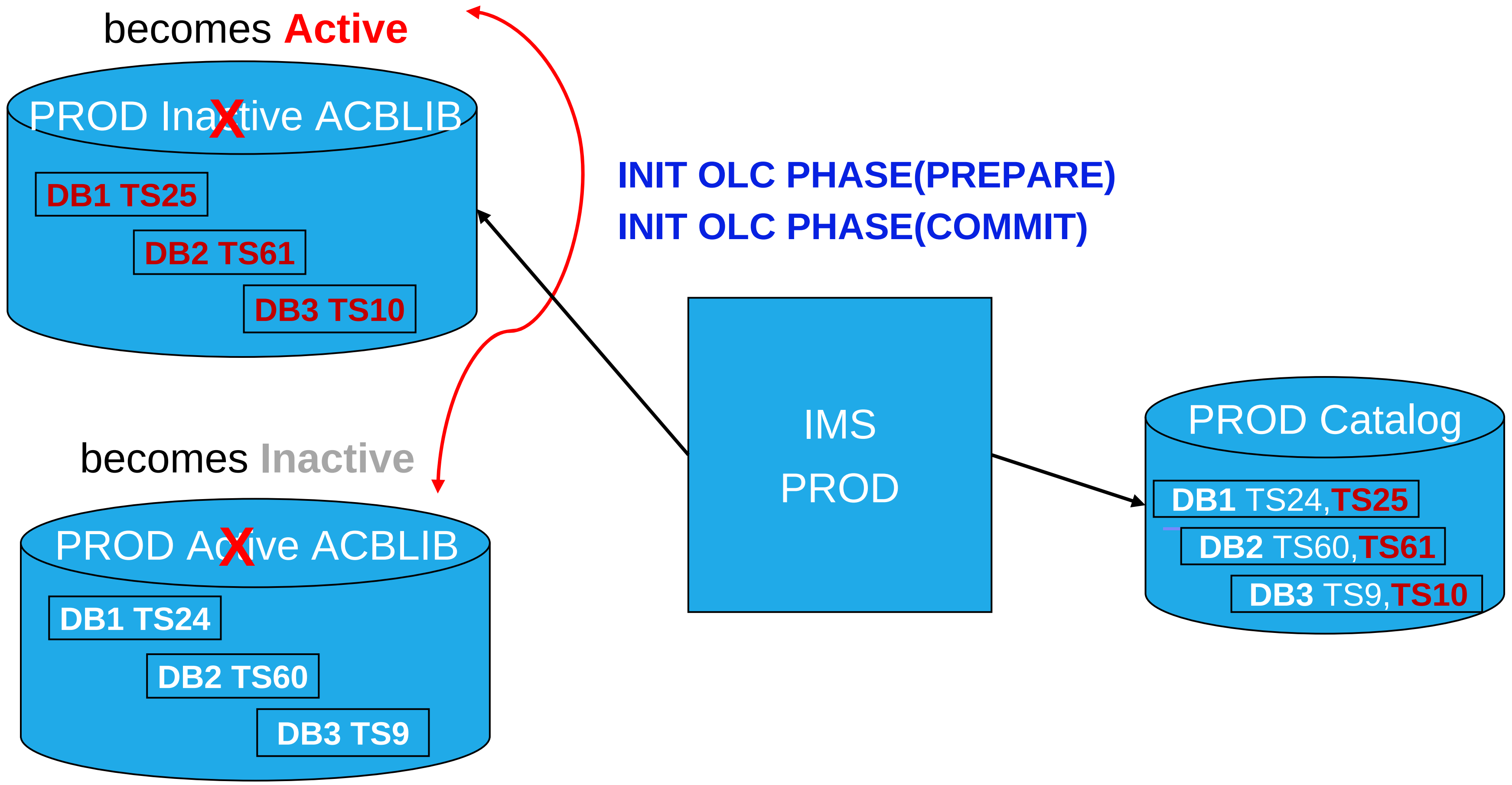
3. Import IMS catalog members and ACBLIB members into IMS PROD system
   - Catalog members from export data set migrated to IMS PROD catalog
   - ACBLIB members with timestamps matching catalog export data set members migrated from PROD Staging ACBLIB to PROD Inactive ACBLIB



Catalog EXPORT data set

Migration to Production Environment - Step IV

4. Operator issues OLC commands to bring the new ACB resources online
   - Inactive ACBLIB becomes active and active ACBLIB becomes inactive
   - Catalog has both active and inactive members



Supplement

IMSplex Catalog Sharing

IMS Catalog Sharing

- Catalog Database Data Sharing
  - One Catalog database can be defined and shared among IMS systems
  - Follows standard IMS data sharing protocols and procedures
  - Use standard processes for database management and monitoring
- Non-Shared Catalog database
  - A separate Catalog database can be defined for each IMS system
  - DFSDFxxx <SECTION=CATALOG> ALIAS parameter allows multiple Catalog databases to exist within same RECON
  - Catalog databases should not be shared if not registered with DBRC
  - Use if you need to run Catalog Populate utility as a BMP
    - Prevents a Catalog database outage when catalog updates needed

IMS Catalog Sharing

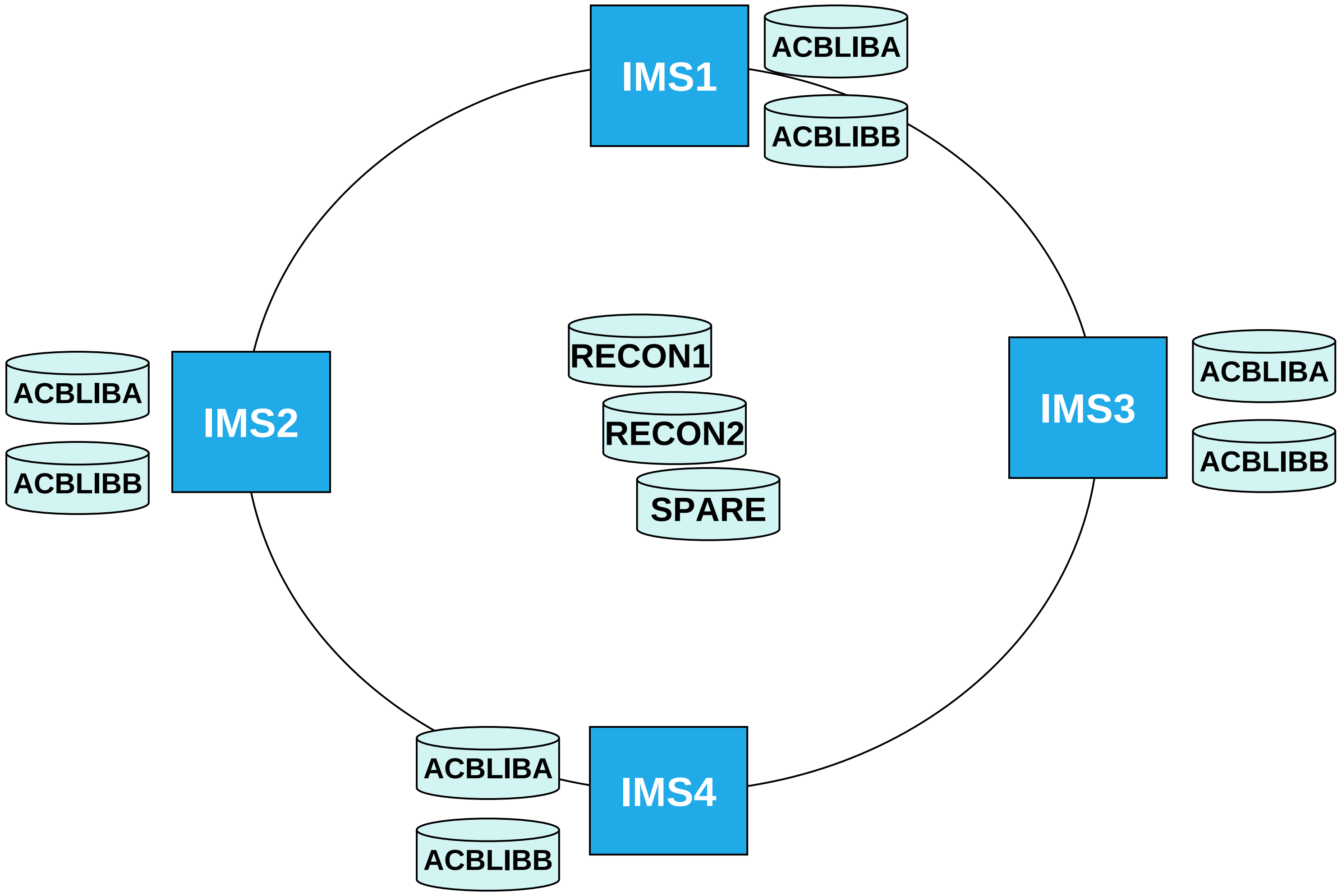
- DFSDFxxx PROCLIB Member
  - Can be shared or not shared among multiple IMS systems
  - New CATALOG section differentiates whether catalogs are shared or not shared
    - If only one definition for <SECTION=CATALOG>
      - One shared Catalog database
      - ALIAS=**DFSC** recommended
    - If multiple definitions for <SECTION=CATALOG***imsid***>
      - ***imsid*** differentiates that a Catalog database is not shared
      - Replace “***imsid***" in section heading with four byte IMS ID of the IMS system that uses this specific CATALOG

        <SECTION=CATALOG**IMS1**> or <SECTION=CATALOG**IMS2**>
      - Unique ALIAS=**xxxx** for each Catalog database
    - A standalone IMS system uses a DFSDFxxx member with one <SECTION=CATALOG> definition

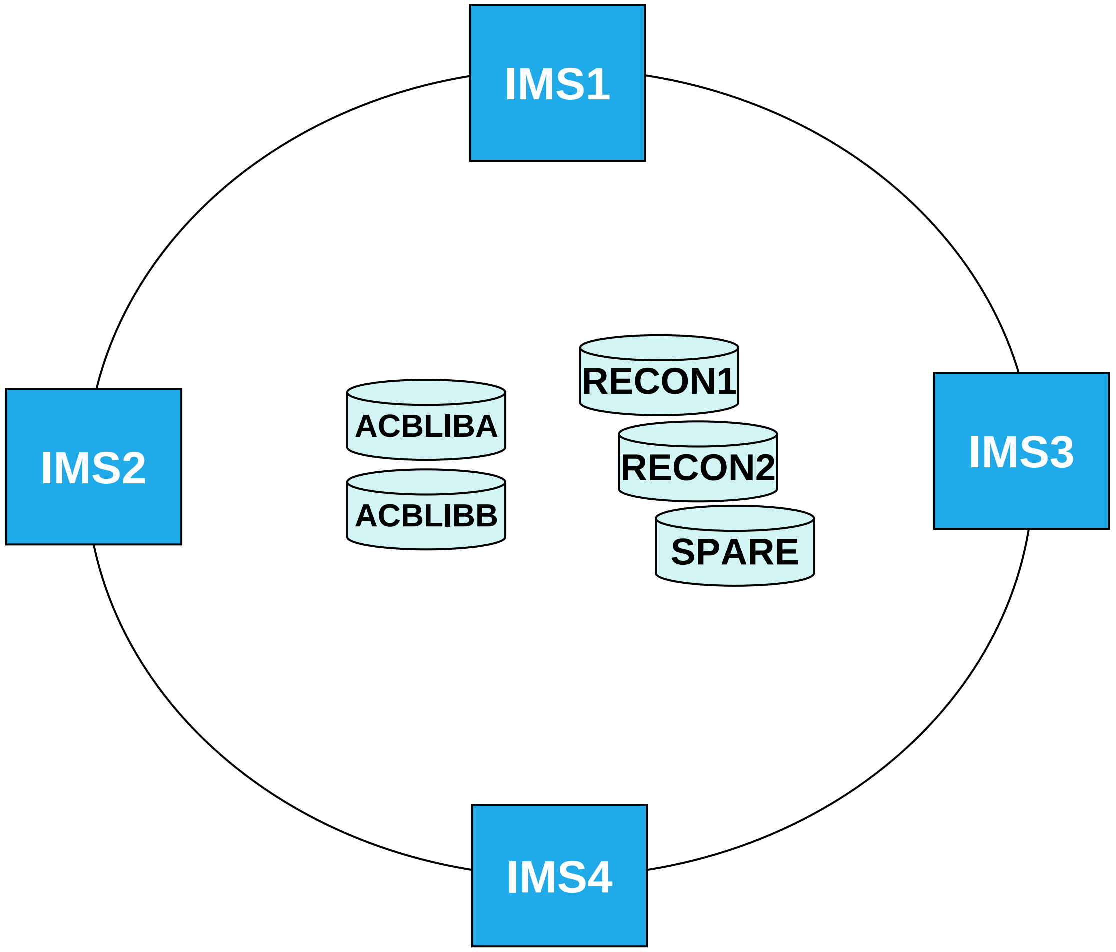
IMS Catalog Sharing

- Catalog ALIAS definition
  - Catalog database ALIAS names must be defined in the DBDLIB
    - Add ALIAS names to the Catalog database name list in your DBDLIB with the IMS Catalog ALIAS Names utility, DFS3ALI0
  - Alias name Catalog database must be defined in the RECON or the Catalog Partition Definition data set
    - Multiple Catalog databases may exist in the RECON
  - Alias name Catalog databases are not automatically maintained by IMS
    - User is responsible to keep Catalog databases in sync
    - Users with cloned ACBLIBs must have processes to keep ACBLIBs in sync during changes
      - Processes need to be modified to keep ACBLIBs and catalog in sync

Without an IMS Catalog
Multiple IMSes, each IMS has it’s own cloned ACBLIBs

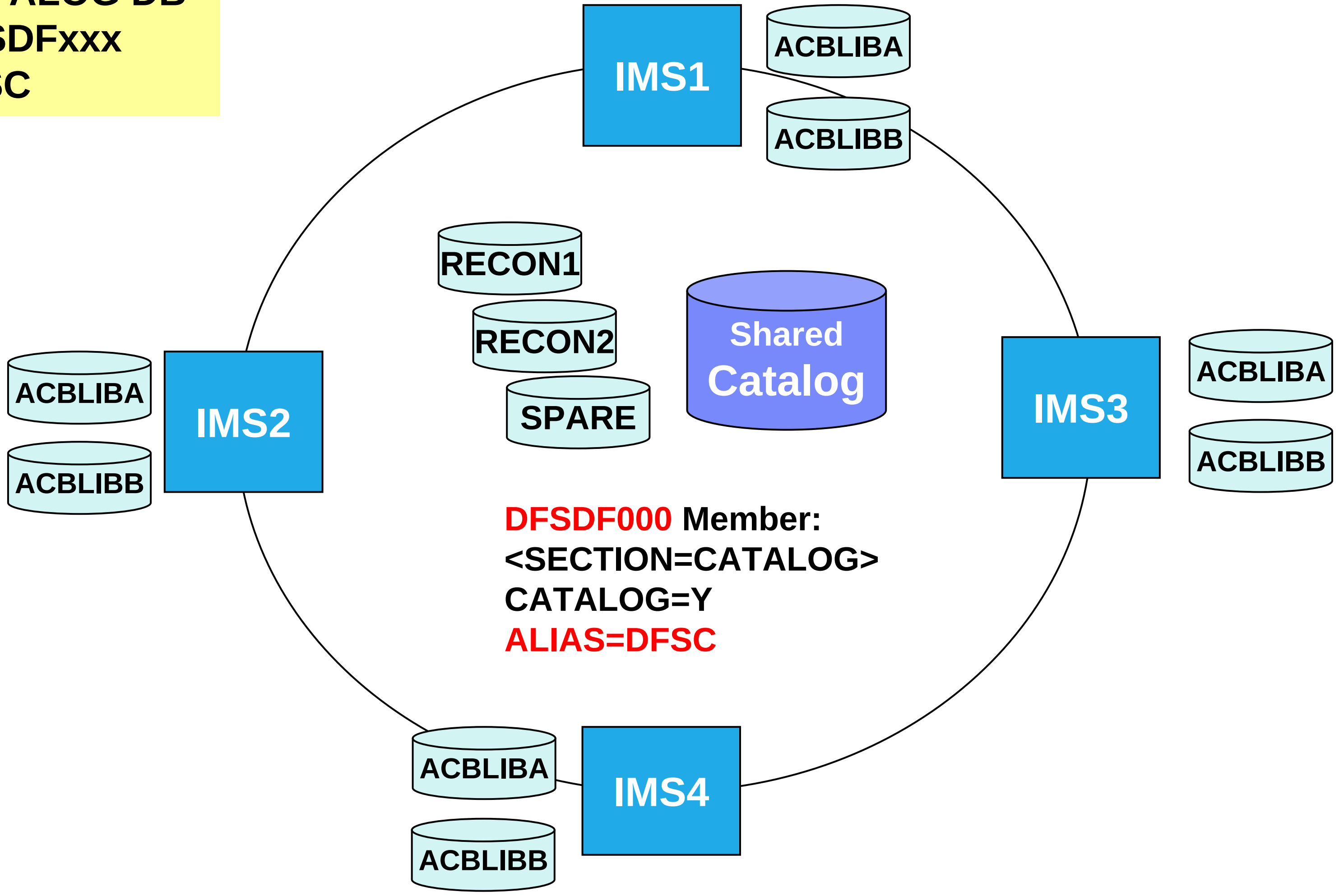


Without an IMS Catalog Multiple IMSes, shared ACBLIBs



Multiple IMSes, Cloned ACBLIBs, Shared Catalog

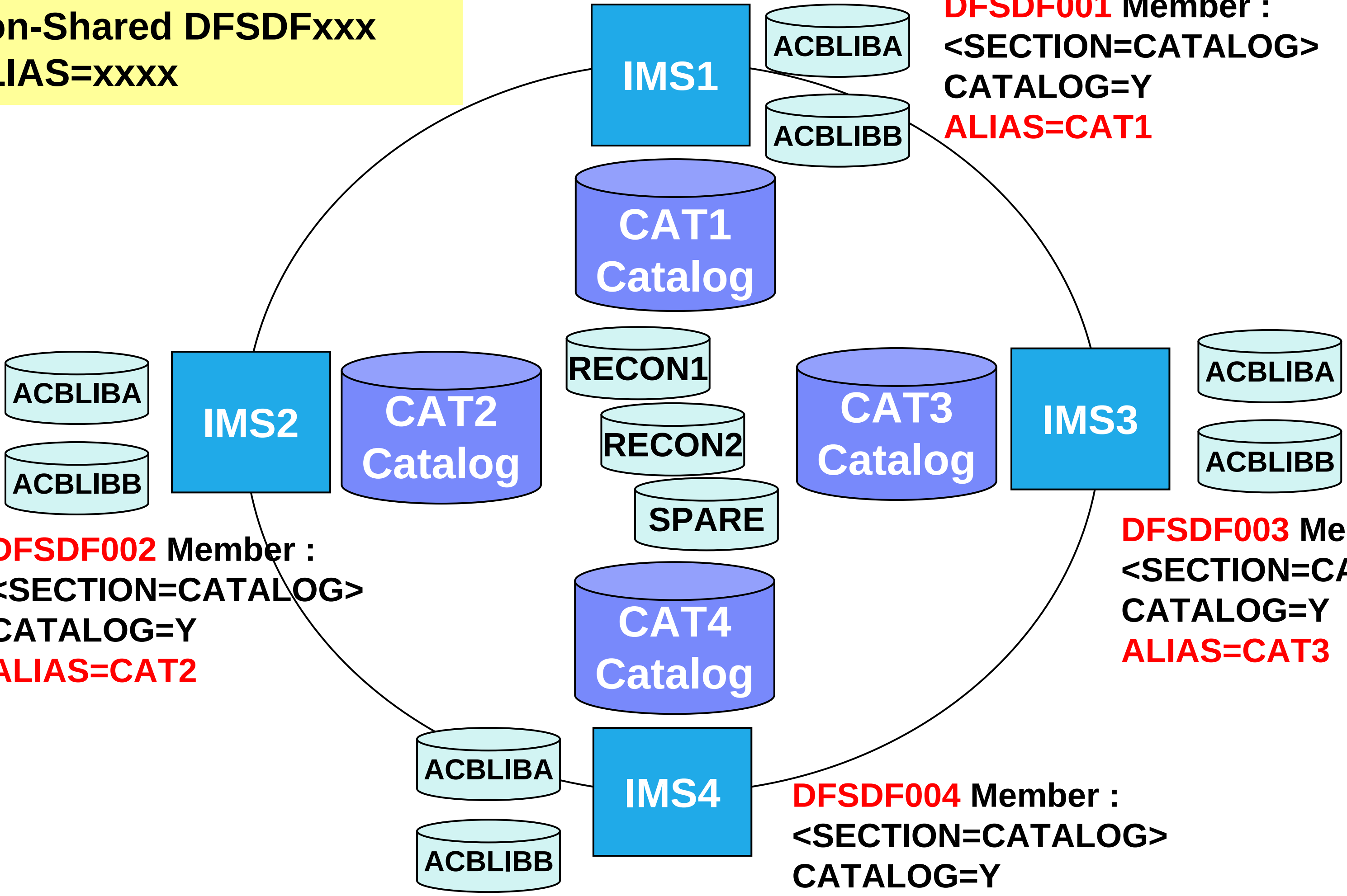
Shared CATALOG DB
Shared DFSDFxxx
ALIAS=DFSC



DFSDF000 Member:
<SECTION=CATALOG>
CATALOG=Y
ALIAS=DFSC

Multiple IMSes, Cloned ACBLIBs, each IMS has it’s own Catalog

Non-Shared CATALOG DB
Non-Shared DFSDFxxx
ALIAS=xxxx



DFSDF001 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=CAT1

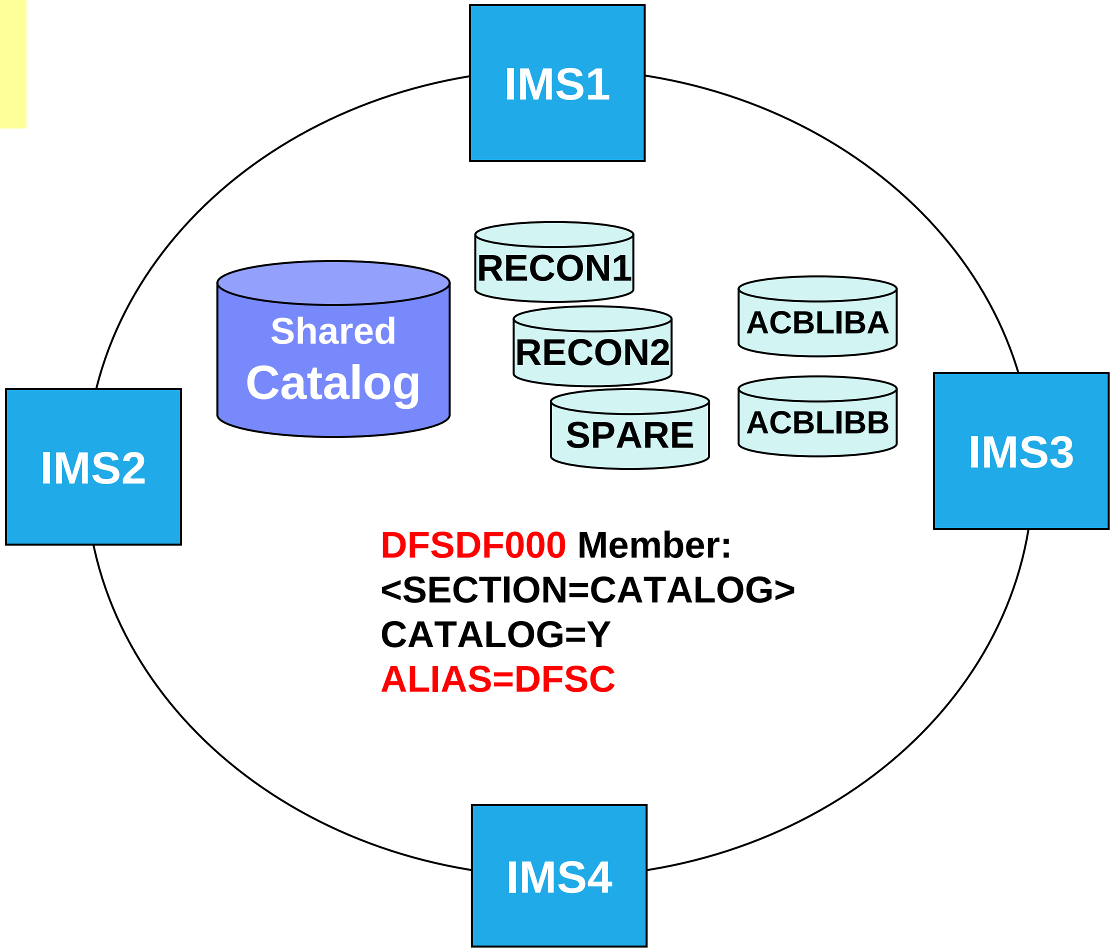
DFSDF002 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=CAT2

DFSDF003 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=CAT3

DFSDF004 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=CAT4

Multiple IMSes, Shared ACBLIBs, Shared Catalog

Shared CATALOG DB
Shared DFSDFxxx
ALIAS=DFSC



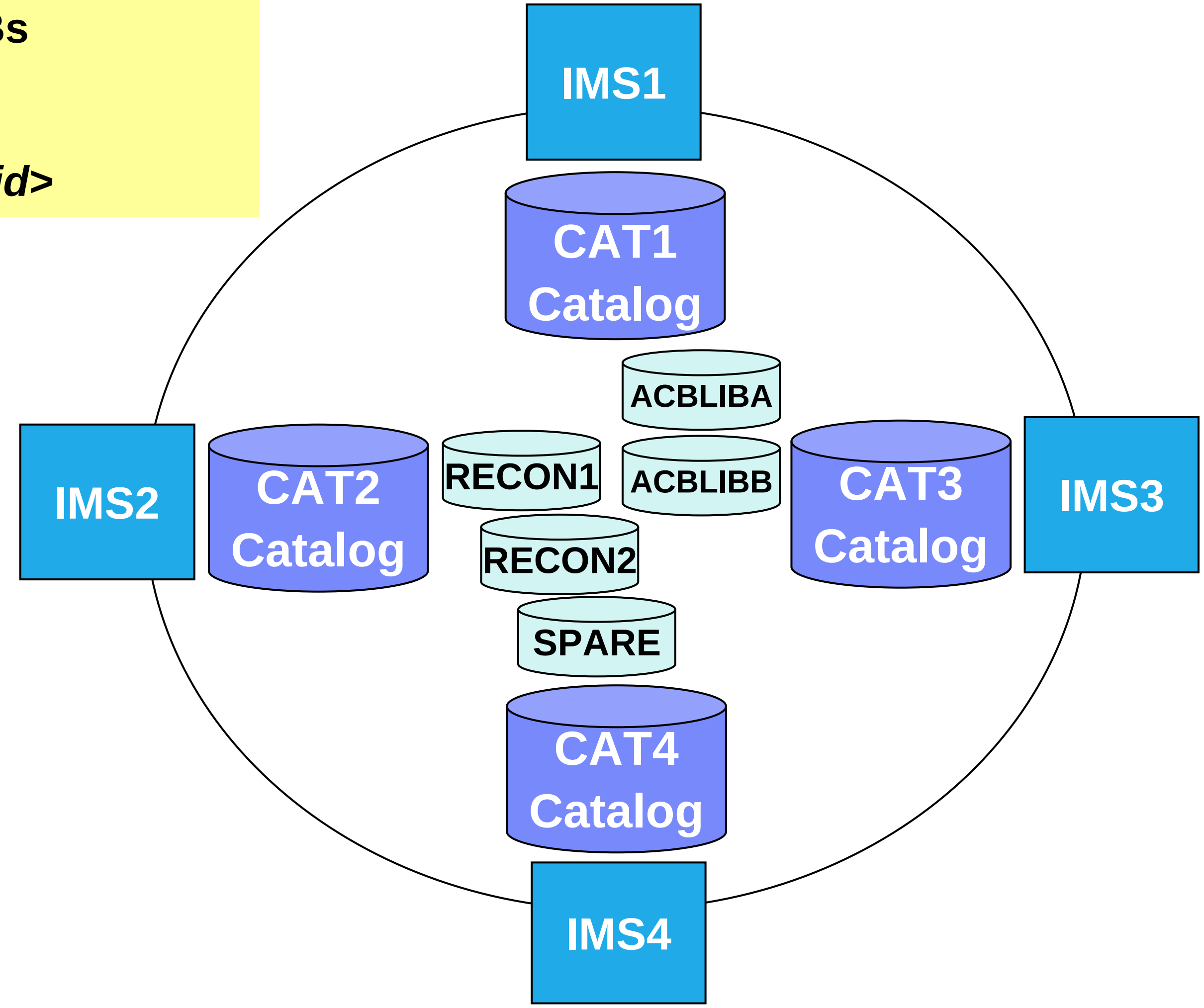
DFSDF000 Member:
<SECTION=CATALOG>
CATALOG=Y
ALIAS=DFSC

Multiple IMSes, Shared ACBLIBs, each IMS has it's own Catalog

Non-Shared CATALOG DBs
Shared DFSDFxxx
Separate
<SECTION=CATALOG*imsid*>
ALIAS=xxxx

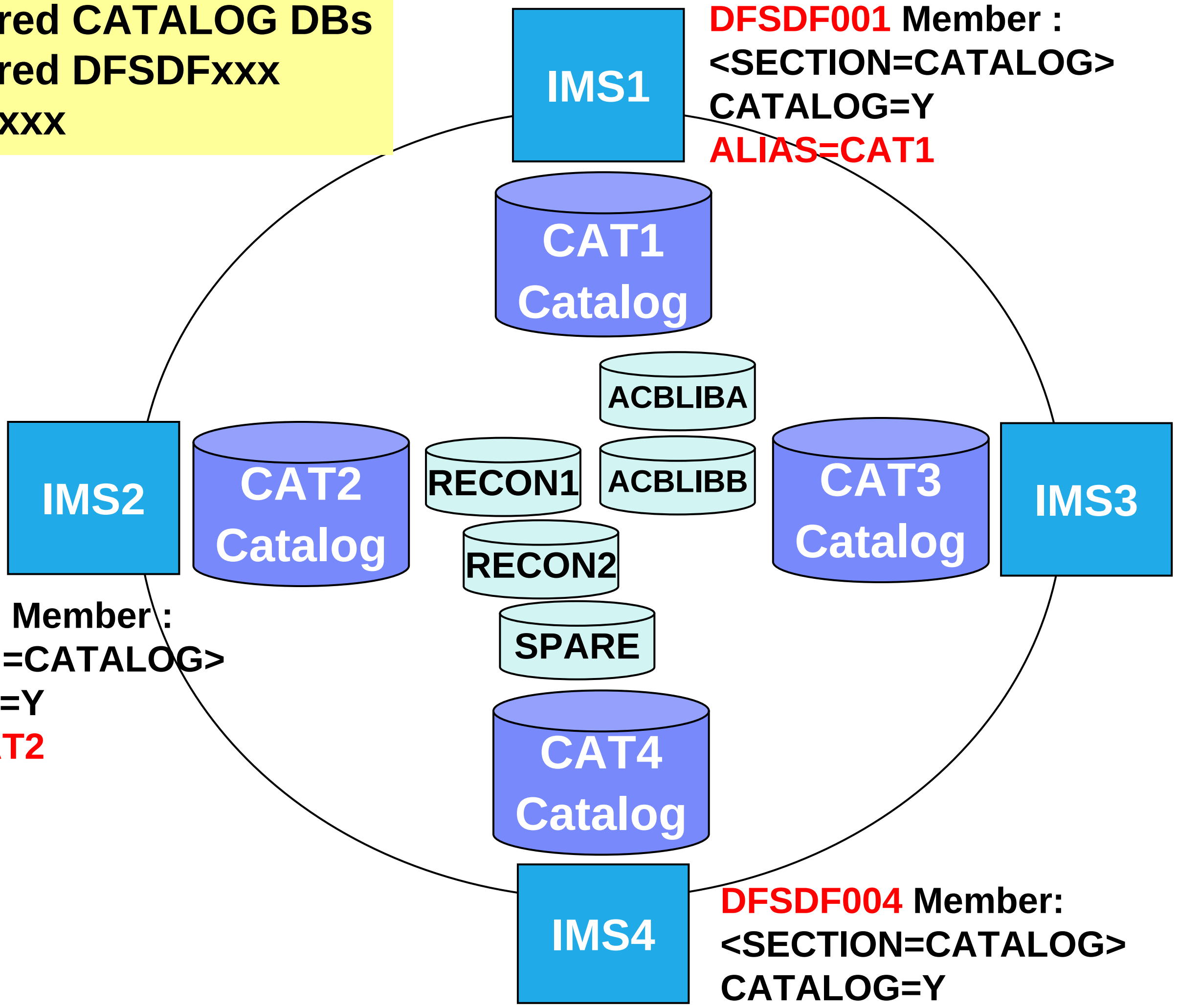
DFSDF000 Member:

```
<SECTION=CATALOGIMS1>
CATALOG=Y
ALIAS=CAT1
<SECTION=CATALOGIMS2>
CATALOG=Y
ALIAS=CAT2
<SECTION=CATALOGIMS3>
CATALOG=Y
ALIAS=CAT3
<SECTION=CATALOGIMS4>
CATALOG=Y
ALIAS=CAT4
```



Multiple IMSes, shared ACBLIBs, each IMS has it's own Catalog

Non-Shared CATALOG DBs
Non-Shared DFSDFxxx
ALIAS=xxxx



DFSDF001 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=CAT1

DFSDF002 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=CAT2

DFSDF003 Member :
<SECTION=CATALOG>
CATALOG=Y
ALIAS=CAT3

DFSDF004 Member:
<SECTION=CATALOG>
CATALOG=Y
ALIAS=CAT4

Thank You for Joining Us today!

Go to [www.ibm.com/software/systemz/events/calendar] to:

- Replay this teleconference
- Replay previously broadcast teleconferences
- Register for upcoming events