

Kyle Charlet

STSM, IMS SOA and Modernization
charletk@us.ibm.com



IMS Catalog



Overview of IMS Catalog

- The IMS catalog will contain information about IMS program resources, database resources, and relevant application metadata that IMS controls:
 - All program- and database-related information defined to the IMS database system including databases, fields, segments, data types, and more
 - Changes made to any of these resources when you create, alter, or delete any IMS resource information will be reflected in the catalog
- The IMS catalog is a key component of the IMS growth strategy:
 - Simplification
 - Integration
 - Dynamic database
 - Versioning

Metadata principles

- Simply defined, metadata is **data** about **data**.
 - IMS example: metadata about an IMS database segment might include information about data types, application-defined fields, user-defined types
- Metadata is not a means to an end
 - Users use tools; tools manage and interact with metadata
- Metadata should
 - Enhance understanding
 - Improve consistency
 - Improve impact analysis
 - Improve productivity
 - Improve governance
- Metadata needs to be linked
 - For impact analysis
 - For understanding data lineage
- Metadata needs to include physical implementation, logical design intent and business semantics

What is metadata?

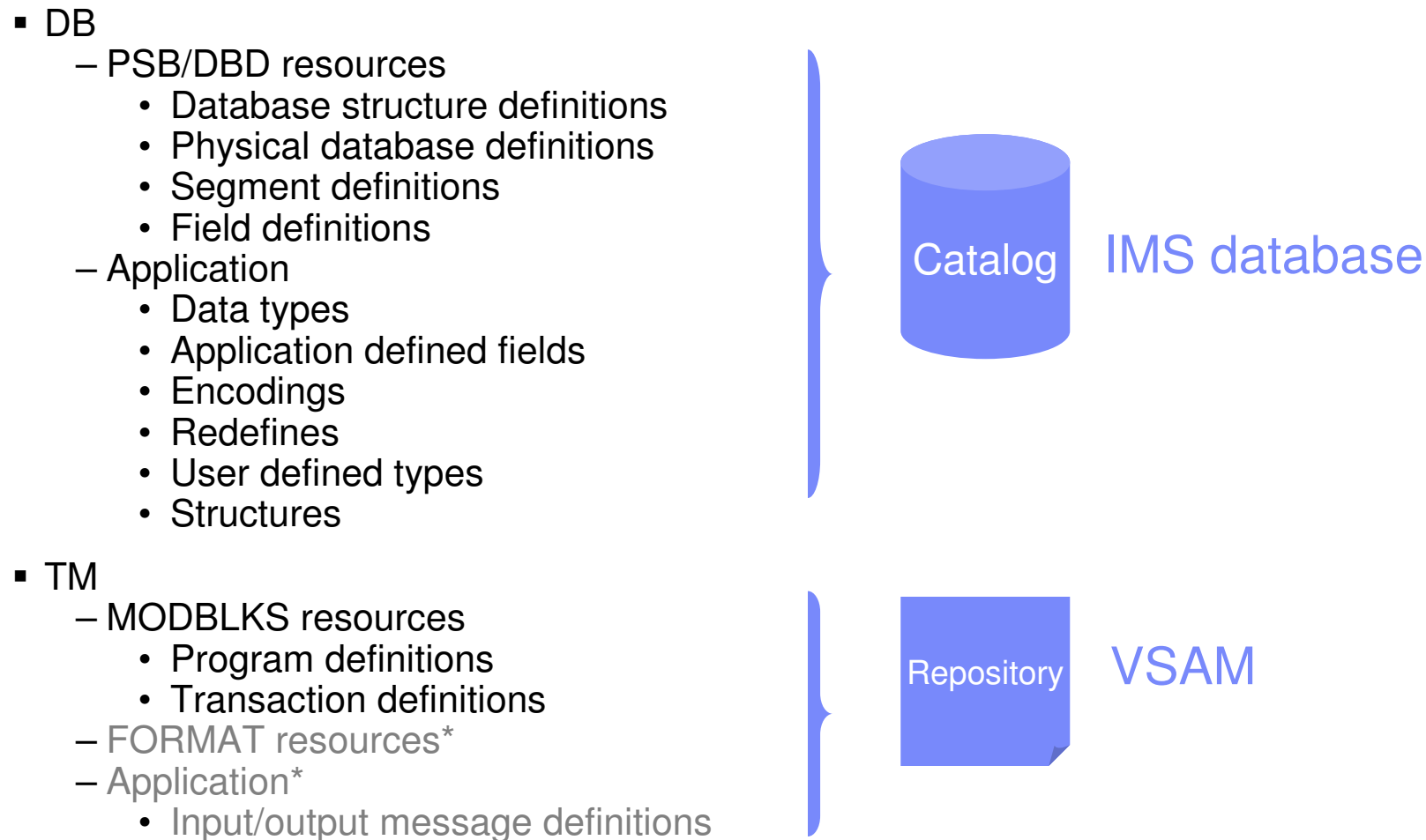
- B** ■ Business metadata
 - Business rules, definitions, terminology, glossaries, algorithms and lineage using business language
 - Audience: Business users

- T** ■ Technical metadata
 - Defines source and target systems
 - Table and field structures and attributes
 - Derivations and dependencies
 - Audience: Specific tool users, AD, BI, ETL, profiling, modeling

- O** ■ Operational metadata
 - Information about application runs
 - 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 both a *business and a technical perspective*

Types of technical metadata and storage medium



* It is our intention to store this metadata in the repository

IMS Catalog

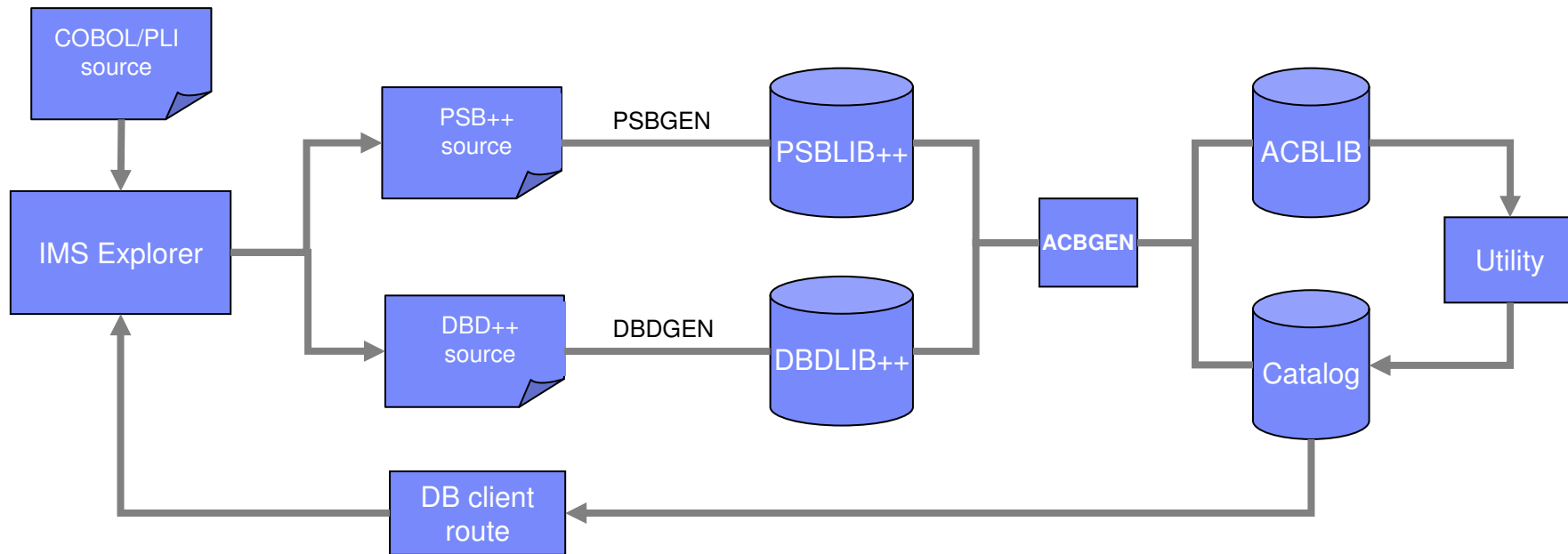
- Trusted information
- Comprehensive view of IMS database metadata (including application metadata) managed by IMS with standard access patterns (JDBC/SQL)
- Offers metadata discovery and exchange via IMS Open Database and the IMS Explorer for Application Development
- Scalable Open Database solution – large scale deployment into virtualized production and test environments
- Enables broad IMS integration into the IBM and non-IBM portfolio of tools (Optim Development Studio, Rational Asset Analyzer, InfoSphere Data Architect, etc)

Comprehensive view of database metadata

- Metadata is defined in a variety of locations
 - PSB/DBD source
 - Not trusted, proprietary, not complete
 - PSBLIB/DBDLIB
 - Not trusted, proprietary, not complete
 - ACBLIB
 - Trusted
 - Proprietary, not complete
 - Application
 - Applications control the data
 - COBOL copybooks, for example, further refine database segments and assign meaningful data types

In aggregate, application and ACBLIB information provide a *comprehensive view of IMS database metadata*. The IMS catalog will house this metadata.

Catalog overview – trusted information

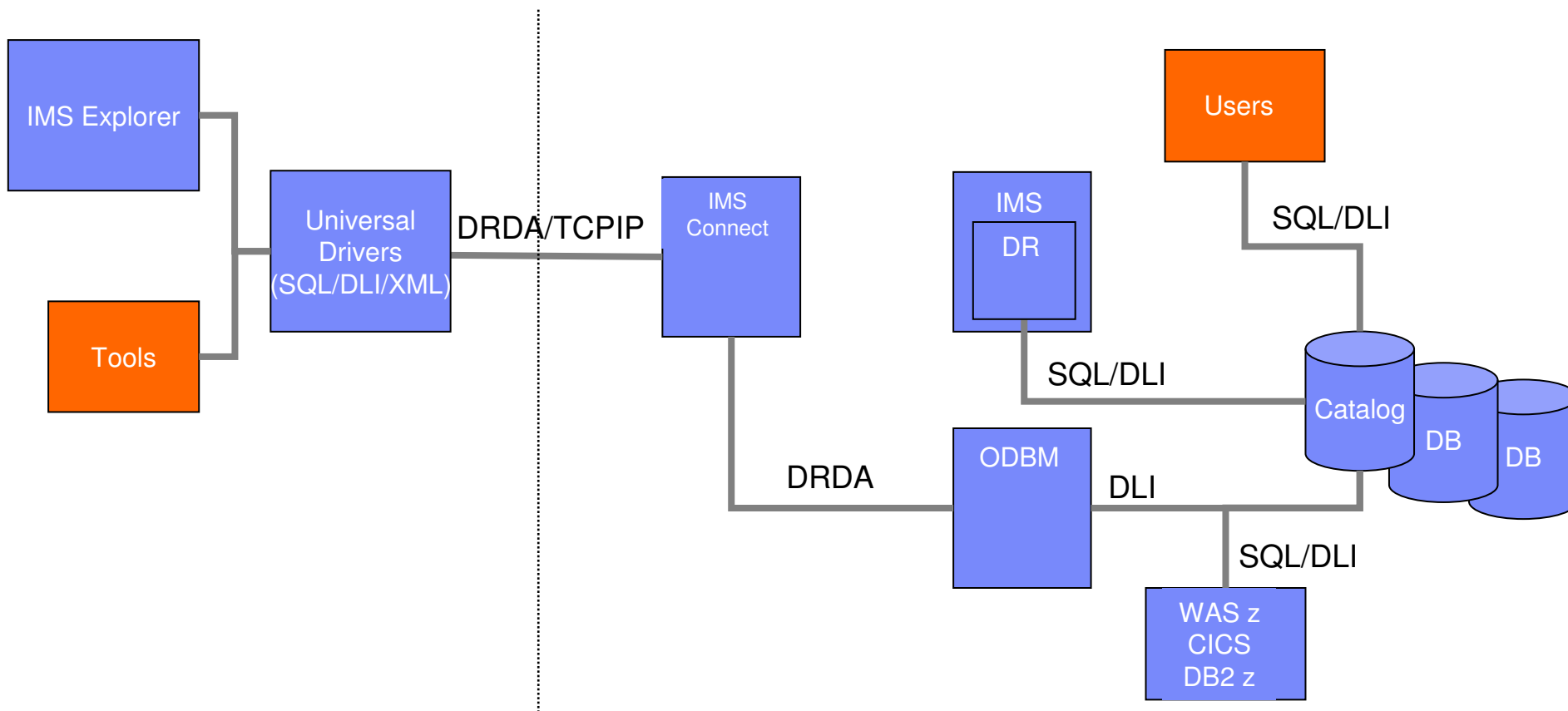


- ACBGEN will populate ACBLIB and catalog in the *same* UOW
- Key points
 - Only way to update catalog is via the GEN process
 - Extended info is acquired via the IMS Explorer

Catalog access and interface

- Open systems
 - Universal drivers
 - SQL and DLI interfaces directly to the catalog
 - XML – render catalog information as XML instance document valid to published IMS metadata schema
- z/OS
 - Universal drivers
 - SQL and DLI interfaces directly to the catalog
 - XML
 - Traditional IMS languages
 - DLI access directly to the catalog
 - Batch access supported

Catalog runtime access



Installation

- IMS provides PSBLIB and DBDLIB members for the catalog
 - User to run ACBGEN
 - IMS internally handles the rest of the initialization process
 - MODBLKS creation (PDIR, DDIR)
 - Loading of DMBs and PSBs into resident pools
- IMS provides utilities that will
 - Create the catalog database
 - Load the catalog from a user ACBLIB
- IMS provides an option that does not require DBRC for the catalog
 - Many customers have expressed that the DBRC requirement for HALDB databases puts undesired burden on test system infrastructure

Migration

- Standard database migration for the catalog
 - No additional migration considerations necessary
- Catalog schema evolution
 - Catalog database will implement the “IMS best practices” for allowing the evolution of the catalog schema

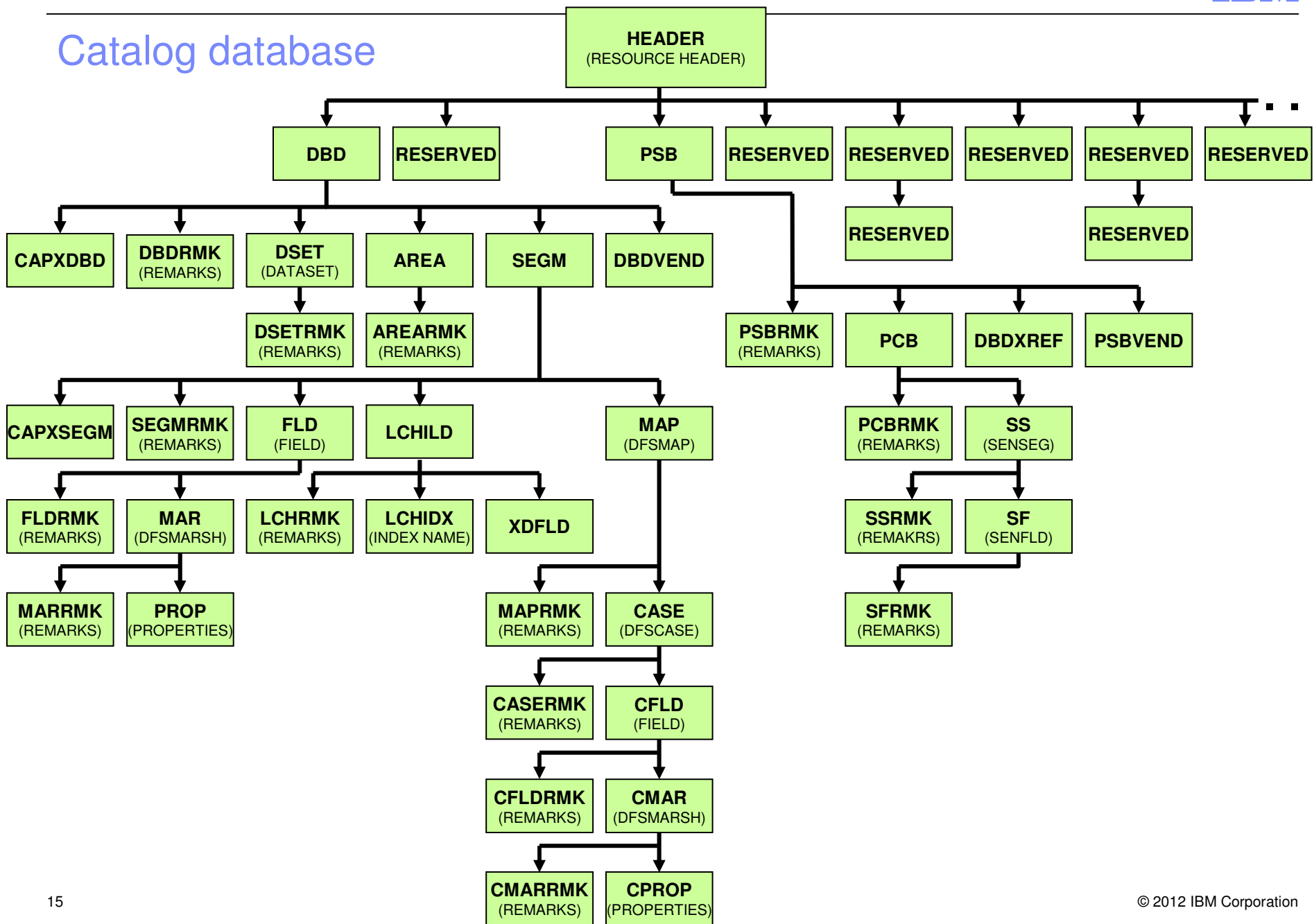
Management

- Catalog supports all standard utilities for backup and recovery
- Catalog supports online reorg (it is a PHIDAM database)
- As part of initial catalog load process IMS will determine the size of the catalog datasets
 - User can allocate or defer to IMS to allocate on their behalf
- DBRC is optional

Coexistence

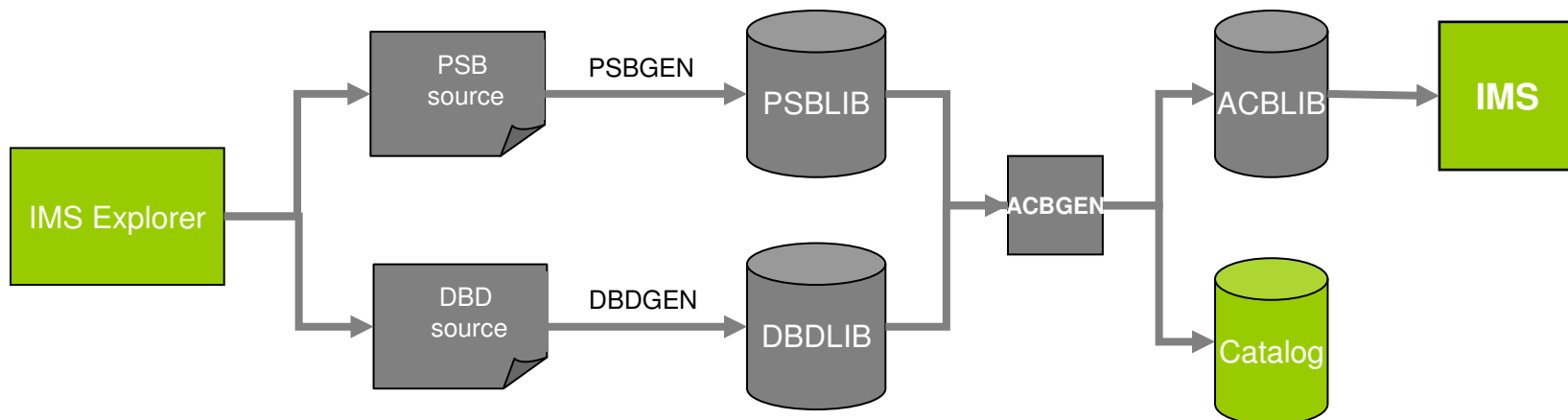
- Two models
 - A single catalog can be data shared among multiple IMS systems
 - Single catalog per IMS system
- ACBLIB and catalog will remain in sync with one another
 - Managed by IMS
- Future
 - IMS will be configured to load (cold start) from either catalog or ACBLIB

Catalog database

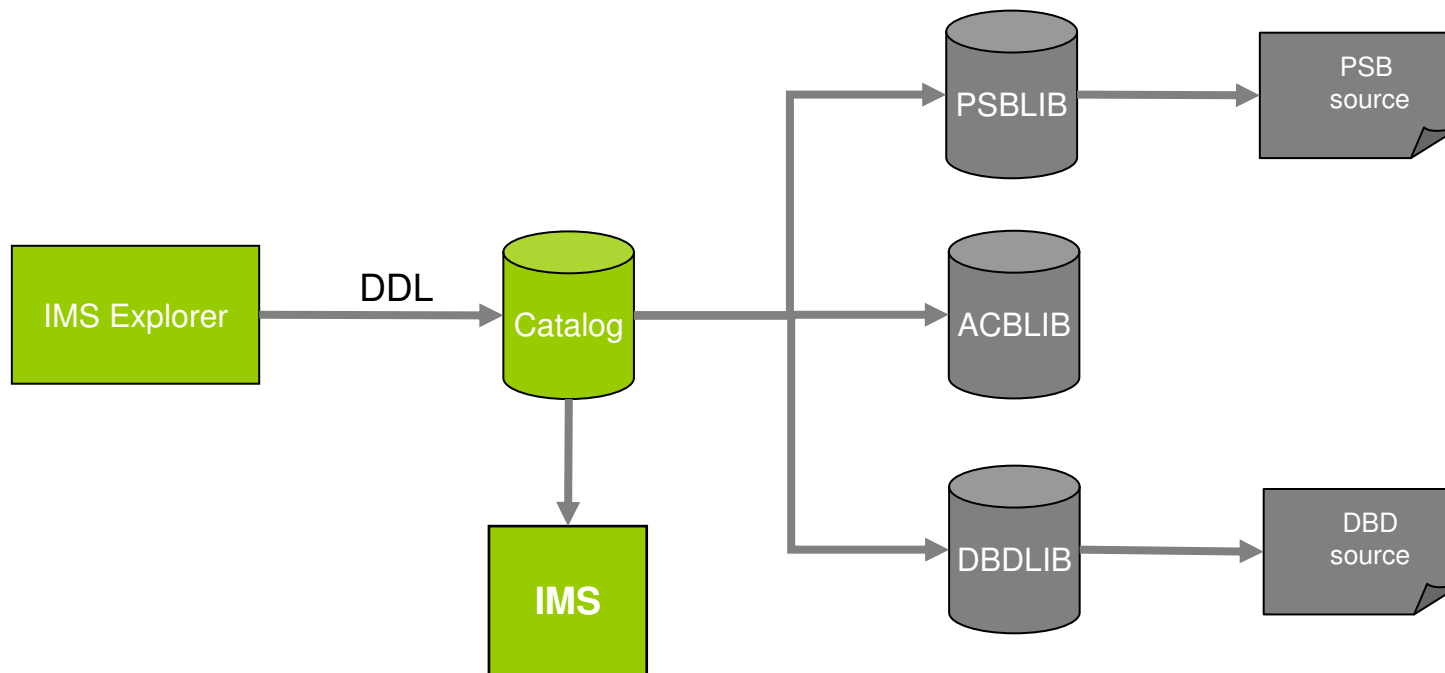


IMS 12 catalog

- Trusted IMS metadata information
- Comprehensive view of IMS database metadata (including application metadata) managed by IMS with standard access patterns (JDBC/SQL)
- Offers metadata discovery and exchange via IMS Open Database and the IMS Explorer for Application Development
- Scalable Open Database solution – large scale deployment into virtualized production and test environments
- Enables broad IMS integration into the IBM and non-IBM portfolio of tools (Optim Development Studio, Rational Asset Analyzer, InfoSphere Data Architect, etc)



IMS catalog – intended support



- IMS DB changes *start* with catalog
 - IMS loads resource information from catalog
 - ACBLIB/PSBLIB/DBDLIB updates will be the by-product of catalog updates
 - Tools that use these libraries can continue to operate, but should migrate to catalog
 - PSB and DBD source can still be optionally generated from PSBLIB and DBDLIB

Dynamic database - Data Definition Language

- SQL incorporates DDL to modify the schema of a database
- Authoring DDL is straight-forward with sophisticated tooling support in the industry
- SQL/DDDL can be used to update/add metadata in the catalog **without** the need of a GEN
 - Directly update the catalog
- IMS can be notified of such an update and load the new definitions
- It is our intention to offer this type of dynamic definition for IMS

```

CREATE TABLE CYCCOUNT (
  PARTROOT_PARTKEY CHAR(17) NOT NULL,
  STOKSTAT_STOCKEY CHAR(16) NOT NULL,
  CYCLKEY CHAR(2) NOT NULL,
  TOTALSTOCK BIGINT NOT NULL,
  PHYSICALCOUNT BIGINT NOT NULL
);

CREATE TABLE STANINFO (
  PARTROOT_PARTKEY CHAR(17) NOT NULL,
  STANKEY CHAR(2) NOT NULL,
  MAKEDEPT CHAR(2) NOT NULL,
  PROCCODE CHAR(2) NOT NULL,
  FILL_0 CHAR(16) NOT NULL,
  FILL_1 CHAR(24) NOT NULL,
  FILL_2 CHAR(2) NOT NULL,
  COMMCODE CHAR(4) NOT NULL,
  MAKECOST CHAR(2) NOT NULL,
  PLANNUM CHAR(2) NOT NULL,
  INVCODE CHAR(1) NOT NULL
);

CREATE TABLE BACKORDR (
  PARTROOT_PARTKEY CHAR(17) NOT NULL,
  STOKSTAT_STOCKEY CHAR(16) NOT NULL,
  FILL_0 CHAR(50) NOT NULL,
  ORDERQTY DECIMAL(7, 7) NOT NULL,
  BACKKEY CHAR(10) NOT NULL,
  WORKORDER CHAR(8) NOT NULL
);

CREATE TABLE PARTROOT (
  PART CHAR(15) NOT NULL,
  FILL_0 CHAR(9) NOT NULL,
  FILL_1 CHAR(2) NOT NULL,
  PARTDESK CHAR(20) NOT NULL,
  PARTKEY CHAR(17) NOT NULL
);

CREATE TABLE STOKSTAT (
  PARTROOT_PARTKEY CHAR(17) NOT NULL,
  UNPLREQMTS INTEGER NOT NULL,
  AREA1 CHAR(1) NOT NULL,
  DEPT CHAR(2) NOT NULL,
  UNITPRICE INTEGER NOT NULL,

```