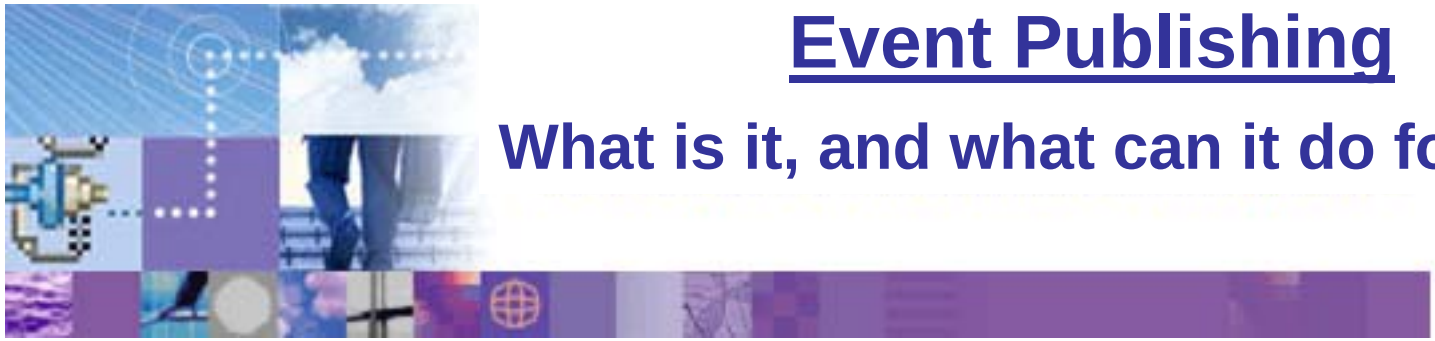
IBM Software Group



Event Publishing

What is it, and what can it do for you

WebSphere software

ON DEMAND BUSINESS

Paul Cadarette
WebSphere Information Integrator Classic Federation Lead Architect
WebSphere Information Integrator Event Publisher Lead Architect
pcadaret@us.ibm.com

Agenda

- **Replication, Synchronization, and Event Monitoring**
- **Replication**
  - Problem Description
- **Solutions for Replication**
  - Recommendations
  - Step by Step
- **Synchronization**
  - Problem Description
- **Solutions For Synchronization**
  - Recommendations
- **Event Monitoring**
  - Problem Description
- **Solutions For Event Monitoring**
  - Recommendations
- **Event Publishing Summary**
  - What
  - Why
  - How
- **Wrap-up**

Replication, Synchronization, and Event Monitoring

- **A description of the 3 most common problems that can be solved with Event Publishing**
- **Pushing, Pulling, and Latency.**
- **A discussion of alternative solutions for each problem**
- **We shall show how Event Publisher products can be used to solve these problems**
- **The Event Publishing Products that we can use are:**
  - WebSphere Information Integrator Event Publisher for Z/OS
    - IMS
    - VSAM
    - IDMS
    - ...
  - DB2 Event Publisher
  - DB2 Queue Replication
    - Packaged product for End to End Replication
  - CICS Business Event Publisher (BEP)
    - CICS Transaction Events
    - CICS Data Events

Replication

- **Replication involves the synchronization of two like databases**
  - Minimal transformations
  - Mirrors
  - Front-End and Back-End (Web Catalogs)
- **These databases are usually on different platforms**
  - Z/OS
  - Wintel
  - AIX
  - Solaris
  - HP
  - Linux (Z/OS, Wintel, etc.)
  - …
- **These databases can be implemented on different Data Base Management Systems (DBMS)**
  - DB2 for Z/OS
  - DB2 distributed
  - Oracle
  - SQL Server
  - IMS
  - IDMS
  - …

Solutions for Replication

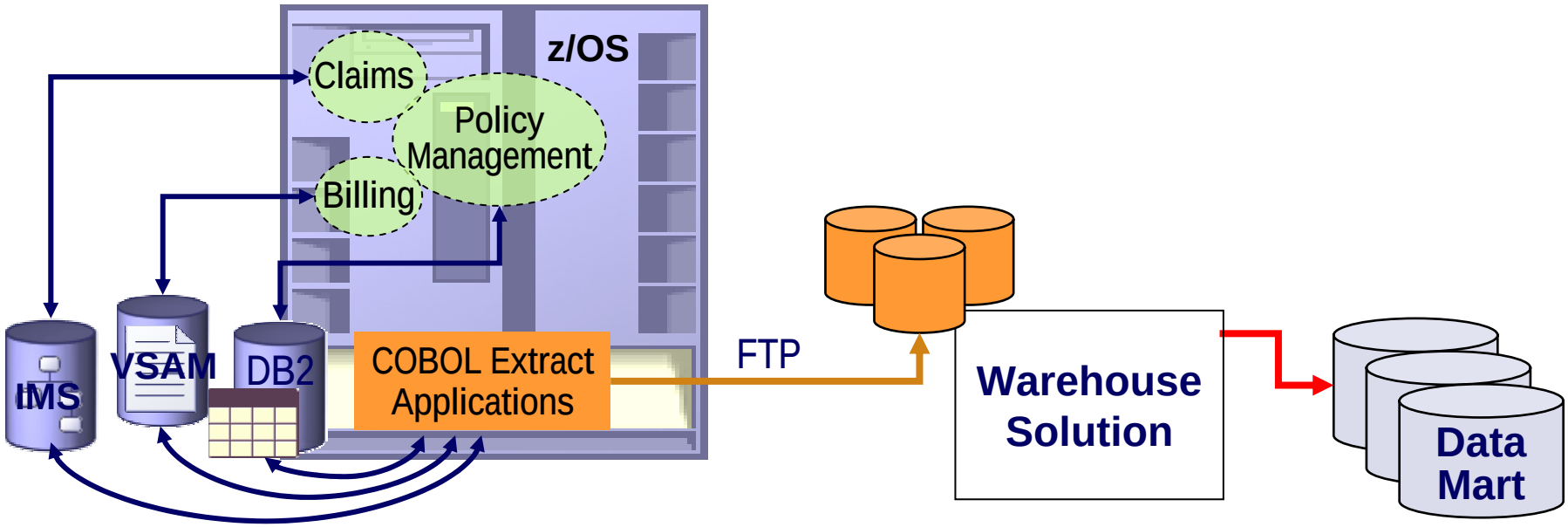
- **Two Phase Commit**
  - Tightly Coupled
  - Zero Latency
  - Cross System Locks
  - DBMS Impact
  - Application impact
- **Staged Replication (ETL)**
  - Very Loosely Coupled
  - High Latency - hours, days
  - Multiple batch jobs
  - Can be automated
  - Stale Data
- **Event Publishing**
  - Loosely Coupled
  - Low Latency - seconds, minutes
  - Dynamic, Adaptive
  - Asynchronous

Packaged Replication vs. Event Publishing

- **DB2 Q Replication is the ideal choice for replication problems**
  - End to end solution
- **Some databases are not yet supported as sources and/or targets**
  - IDMS
  - IMS
  - Adabas
  - ….
- **Event Publisher provides basics for Replication**
  - Capture
  - Normalization
  - Delivery
  - Supports both Tools and User Applications
    - Data Stage
    - WB II Event Broker
    - ...
  - Can be used for Synchronization and Event Monitoring
  - Can coexist with Replication product

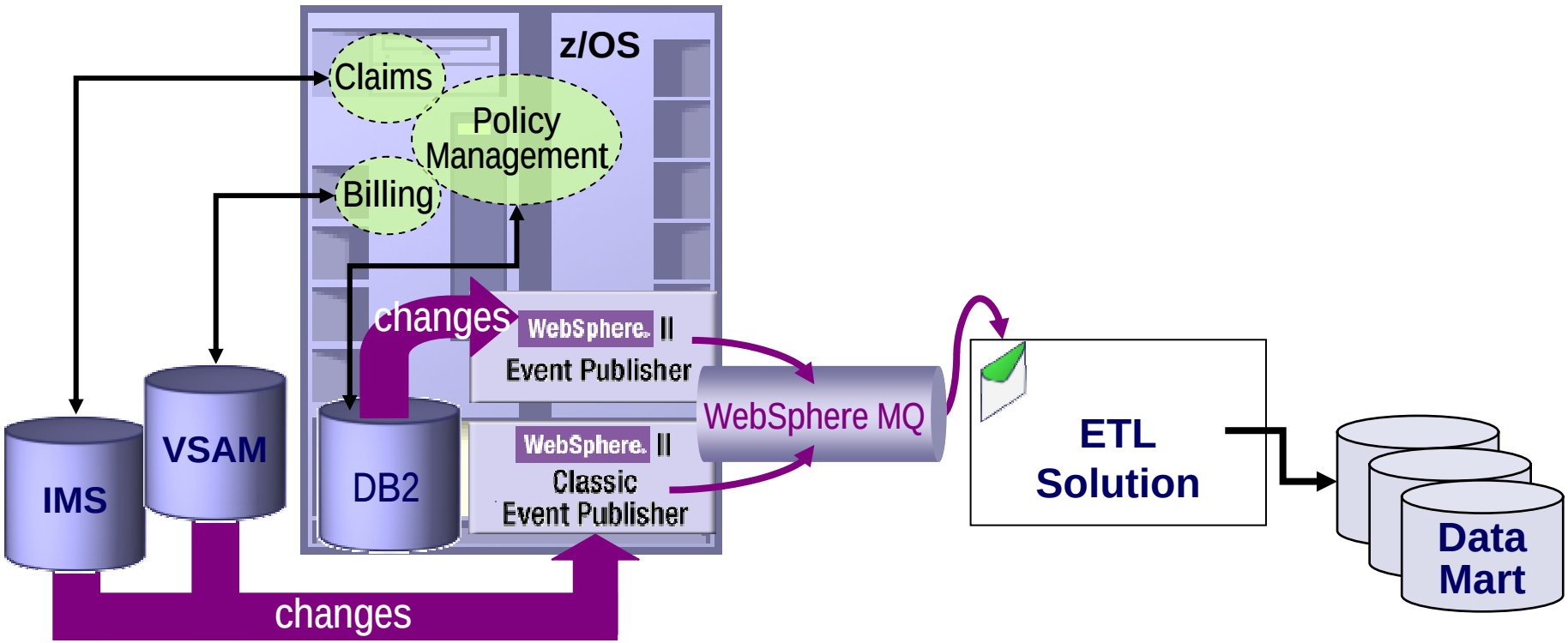
Replication Scenario using traditional ETL

- **ETL environments can't keep up with the data**
  - Shrinking batch windows demand ever larger "pipes" – no time for errors
  - Full data pulls are too large
- **Difficult to find only "the changes"**
  - Legacy data stores may not have date/time stamps
    - Wasted machine cycles searching legacy data stores
    - Wasted man-hours building legacy application hooks



Replication Scenario using Event Publishing

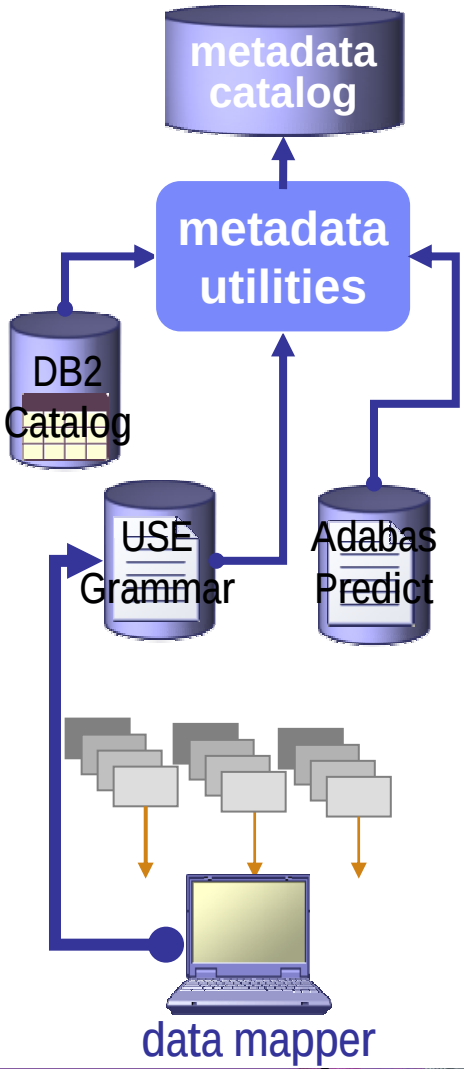
- **Dynamic, changed-data feed**
  - Maximize data currency while minimizing & stabilizing bandwidth utilization
- **Reliable and recoverable**
  - Recoverability built-in and WebSphere assures high performance delivery



Replication using Event Publishing – Step by Step

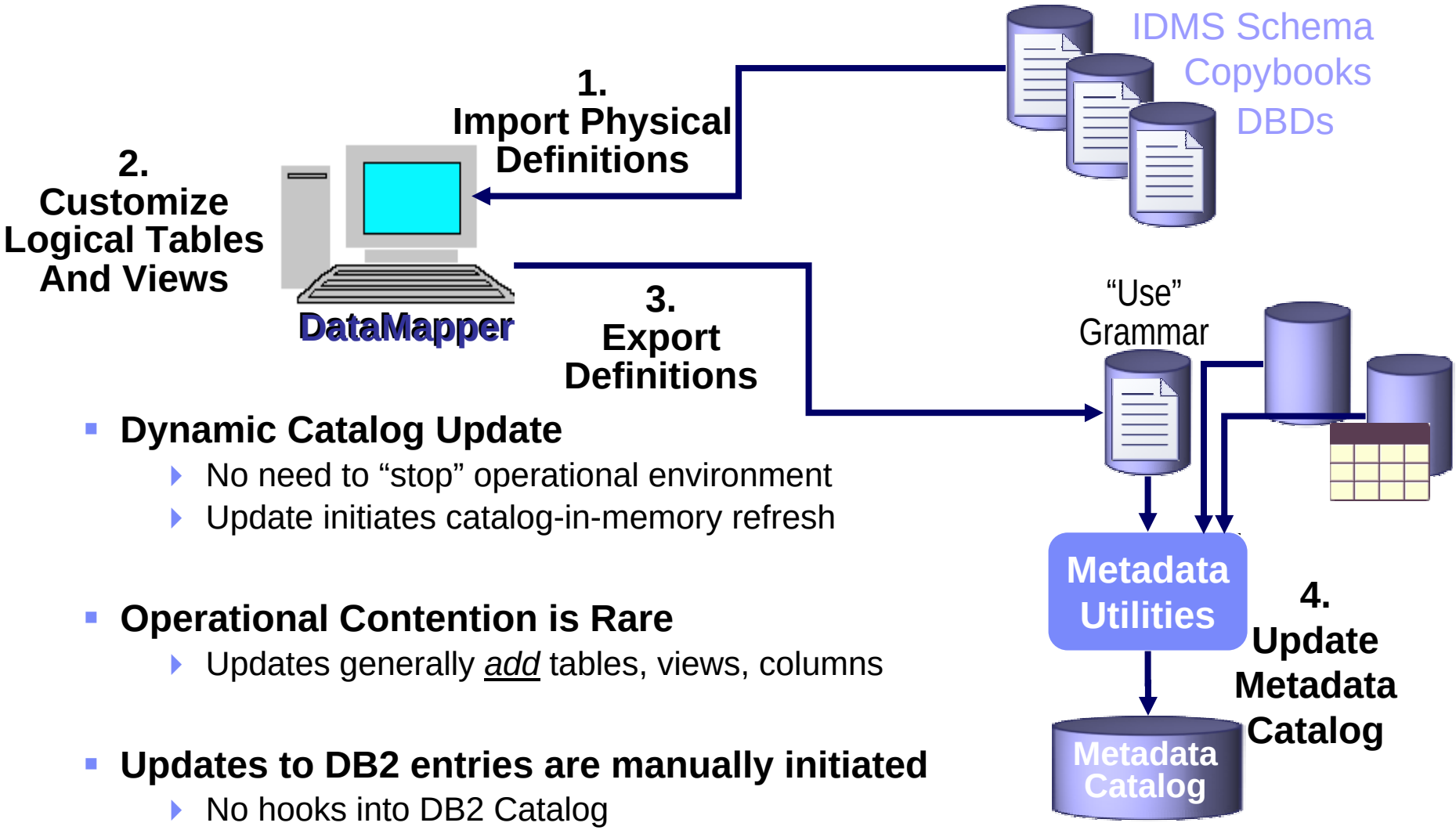
- **Event Publisher for Z/OS example**
  - Discover
  - Map
  - Populate
- **Apply side choices**
- **Sample Java Queue Reader**
  - Distributed reading of Z/OS queues
- **Mid Tier Product Enablement**
  - Data Stage
  - WBI Event Broker
  - 3rd Party

Common Metadata Mapping



- **Metadata defines business-oriented relational mappings**
  - Import existing copybooks, IDMS schemas, IMS DBDs, etc.
  - Generate logical relational *reference* table definitions
  - GUI to customize logical tables to business requirements
- **Simulated RDBMS catalog and more**
  - RDBMS-like catalog support: systables, syscolumns, etc.
  - Query-able tables for non-relational metadata
- **Some metadata-driven features**
  - Automatic translation of legacy data types
  - Handles legacy constructs like recurring data and redefines
  - Complex tables can span segments, records, etc.
  - Metadata-driven filtering using WHERE clauses
  - Enhances security via schema mapping, views, & DB2-like security
- **Metadata Utilities**
  - Create and update metadata catalog entries
  - Verify metadata against physical (e.g. VSAM index checks)
- **Data mapper**
  - Metadata customization and visual administration

Metadata Management Workflow

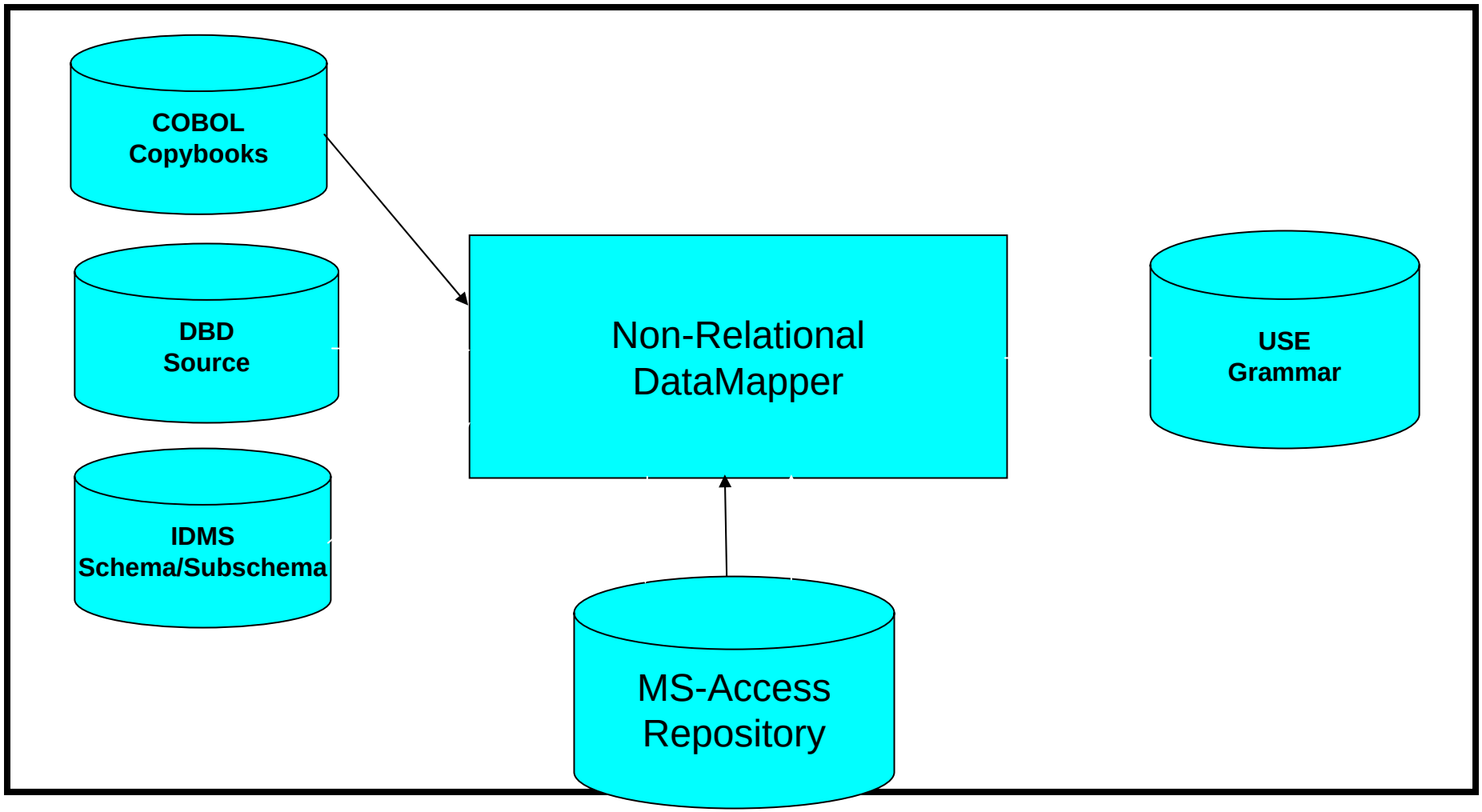


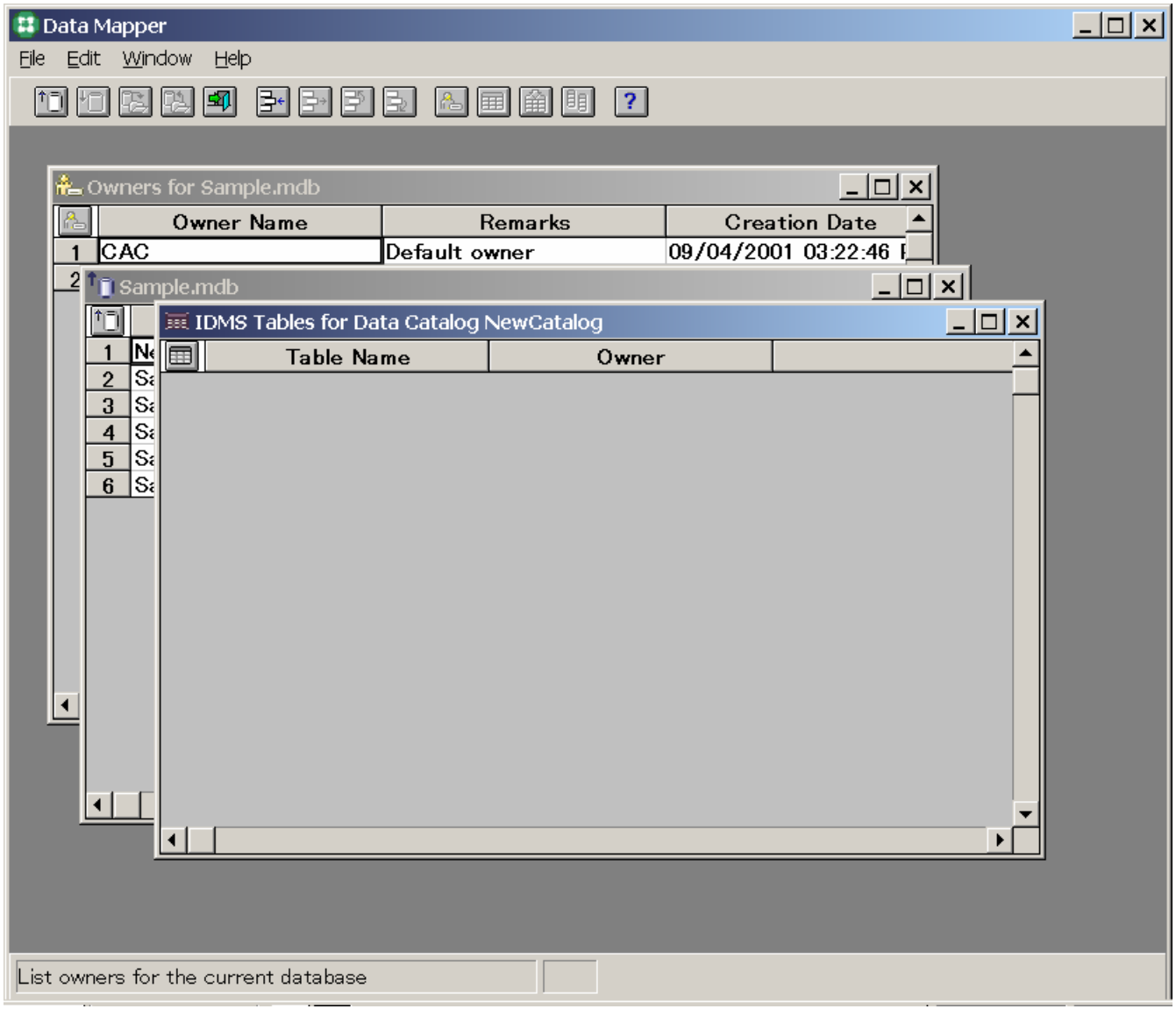
- **Dynamic Catalog Update**
  - No need to "stop" operational environment
  - Update initiates catalog-in-memory refresh
- **Operational Contention is Rare**
  - Updates generally *add* tables, views, columns
- **Updates to DB2 entries are manually initiated**
  - No hooks into DB2 Catalog

General Information

- **The Data Mapper is the primary tool for creating Logical Tables:**
  - A Logical Table is a relational description of a non-relational database or file.
  - A Logical Table can also be thought of as a virtual table. They are materialized on the fly from the underlying database or file system.
  - Logical Tables are generally prefixed with their DBMS-type (e.g., an IMS Logical Table).
  - A DBMS is also referred to as a data source (e.g., an IDMS) which should not be confused with a CLI, JDBC or ODBC data source which can be used to access any type of Logical Table.
  - Logical Tables have attributes that are associated with all type of Logical Tables as well as DBMS-specific attributes and behaviors.
- **The Data Mapper is a Windows application**
- **The Data Mapper relies heavily on the use of a Toolbar**

Tool Overview




Data Mapper
File Edit Window Help
Owners for Sample.mdb
Owner Name
Remarks
Creation Date
1 CAC
Default owner
09/04/2001 03:22:46
Sample.mdb
IDMS Tables for Data Catalog NewCatalog
Table Name
Owner
List owners for the current database

Mapping Process Overview

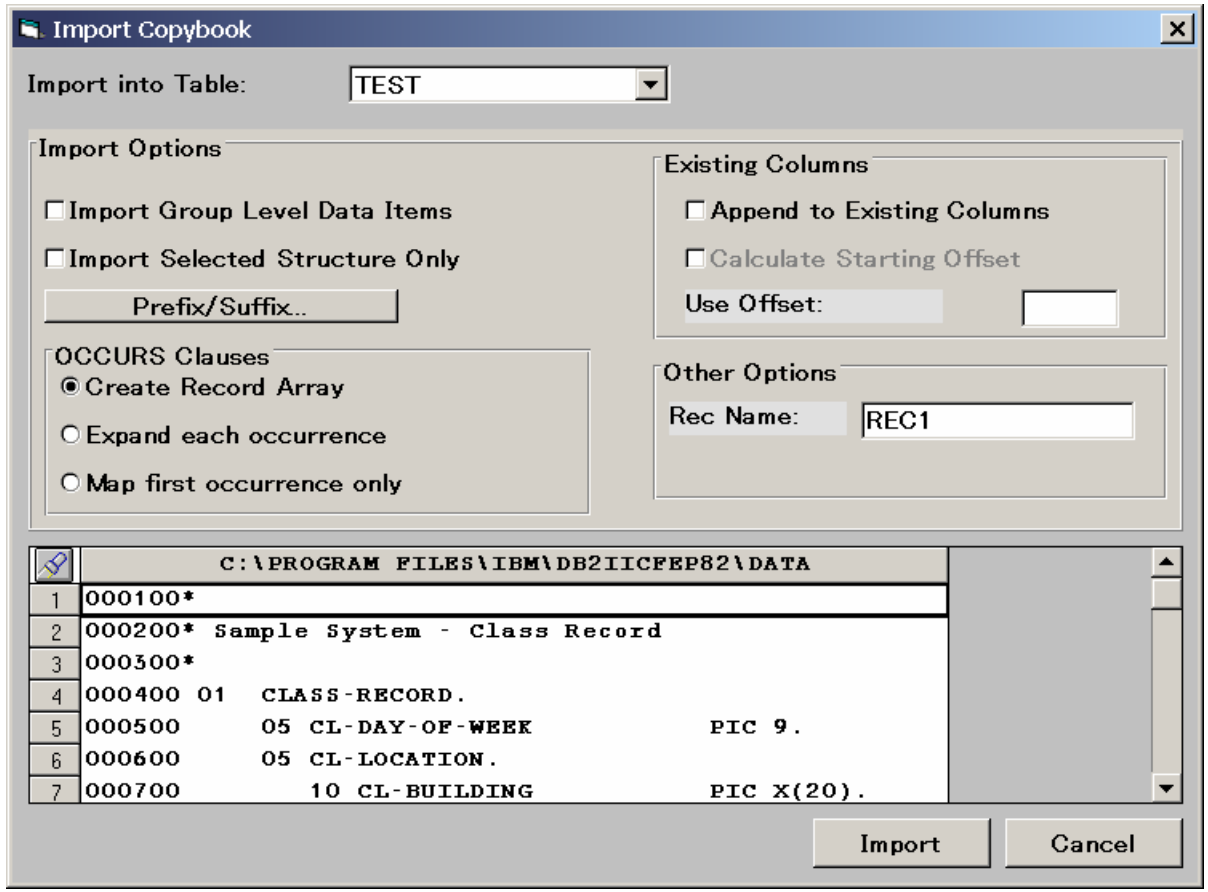
- **Discovery and collection**
- **Mapping – using the Data Mapper to create Logical Tables**
- **Generating the USE grammar and transferring it to Z/OS.**
- **Loading the System Catalogs**

Mapping Process
Discovery and Collection

- **Identify Target Database/File**
- **Identify Source Defintion(s)**
  - IMS - Logical/Physical DBD(s)
  - IDMS – Schemas/Subschemas
- **Identify COBOL Copybooks**
- **Find out where the source lives so that it can be brought down to the Workstation where the DataMapper is installed.**
- **Discuss with the DBA what data is available and the keys/indexes that are available to access the data.**
- **Discuss with the business user/client developers what information is required and how it needs to be presented.**
- **Generally will want to create some initial discovery mappings and issue queries to determine what is really in the database, general performance aspects and data "quirks"**
- **Create new Logical Tables to meet individual business needs – a Logical Table should generally represent the data needed for a particular query or class of queries.**

Mapping Process
Using the DataMapper

- **Launch Data Mapper**
- **Select/Create Repository**
- **Select/Create Data Catalog**
- **Optionally, define Owner (s)**
- **Load Source Definitions**
  - Use Built-in FTP capabilities to download or perform manually
- **Create Table**
- **Import COBOL copy book (s) to create the Tables Columns**
  - Use built-in FTP capabilities to download or perform manually
- **Review/tailor Column Definitions**
- **Define Index(es)**
- **Identify Keys**
- **Generate USE Grammar**

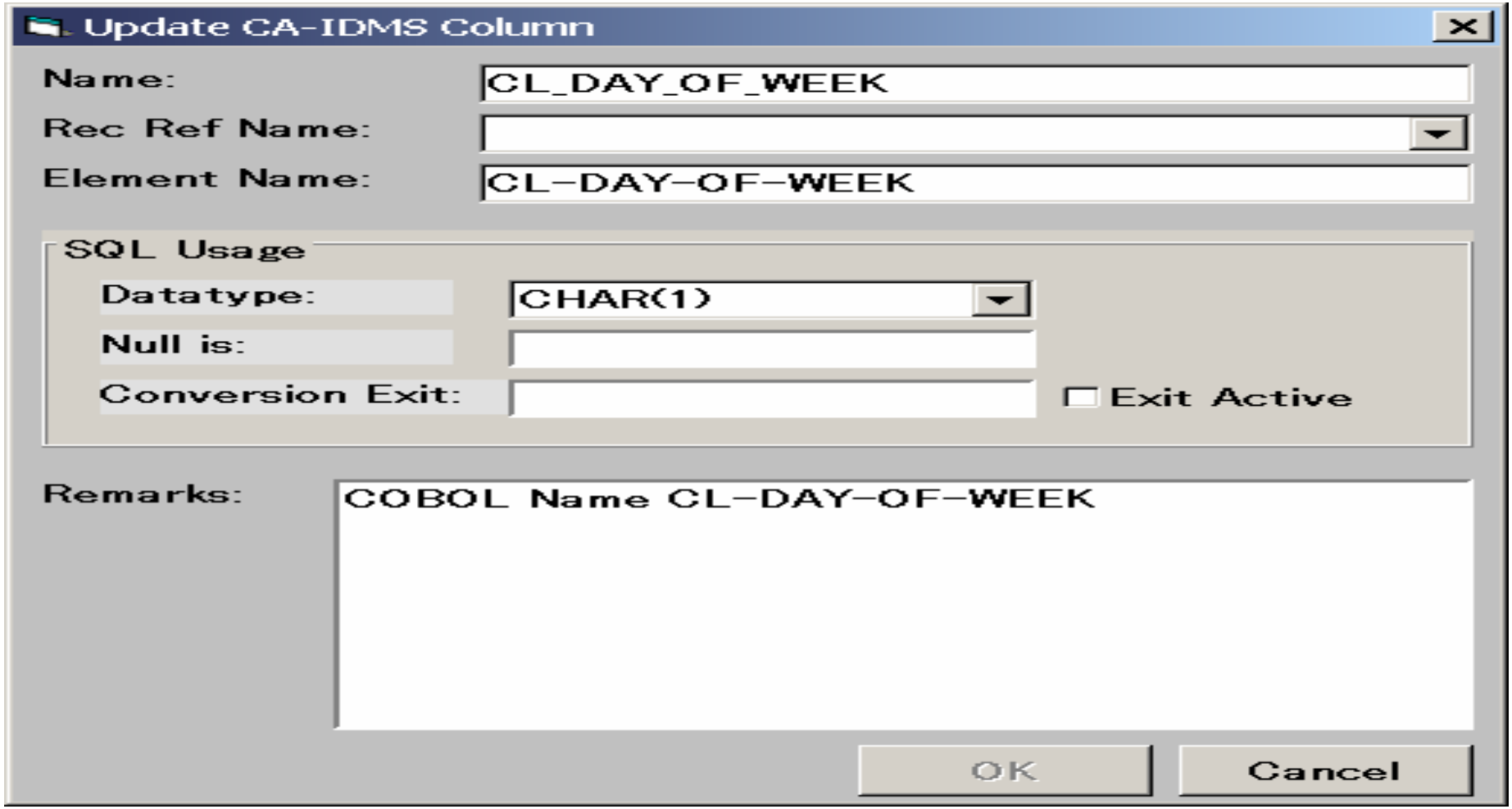

Import Copybook
Import into Table:
TEST
Import Options
Import Group Level Data Items
Import Selected Structure Only
Prefix/Suffix...
OCCURS Clauses
Create Record Array
Expand each occurrence
Map first occurrence only
Existing Columns
Append to Existing Columns
Calculate Starting Offset
Use Offset:
Other Options
Rec Name:
REC1
C:\PROGRAM FILES\IBM\DB2IICFEP82\DATA
000100*
000200* Sample System - Class Record
000300*
000400 01 CLASS-RECORD.
000500 05 CL-DAY-OF-WEEK PIC 9.
000600 05 CL-LOCATION.
000700 10 CL-BUILDING PIC X(20).
Import
Cancel

Importing COBOL Copybooks Guidelines

- **Review the contents of the copybook**
- **Look for a complex object:**
  - Redefinitions
  - OCCURS clauses
  - Multiple OCCURS clauses
- **Generally, want to create separate tables for each:**
  - Redefinition
  - OCCURS group – with “key” and non-repeating fields
- **Do not use default import settings for a complex object**
- **Consider using a reference table when you encounter a complex object**

Columns Overview

- **Columns are automatically created based on the data items contained in the COBOL copybook.**
- **SQL data type are assigned to each Column based on the PICTURE clause associated with each data item.**
- **Relative offset mapping start at zero:**
  - Record/Segment
  - Record Array
- **Remarks are not stored in System Catalog.**
- **The Create/Update Column dialog box is DBMS-specific but contains common elements/functions.**
- **Common functions:**
  - SQL data type support
  - Native data type support
  - NULL specifications
- **Techniques for dealing with unsupported data types.**


Update CA-IDMS Column
Name:
CL_DAY_OF_WEEK
Rec Ref Name:
Element Name:
CL-DAY-OF-WEEK
SQL Usage
Datatype:
CHAR(1)
Null is:
Conversion Exit:
Exit Active
Remarks:
COBOL Name CL-DAY-OF-WEEK
OK
Cancel

Columns
SQL Data Types

- **Commonly used data types:**
  - CHAR
  - DECIMAL
  - SMALLINT
  - INTEGER
- **Exotic data types:**
  - FLOAT
  - VARCHAR
  - LONG VARCHAR
  - GRAPHIC
  - VARGRAPHIC
  - LONG VARGRAPHIC

Columns
SQL Data Types

Name	COBOL PICTURE CLAUSE
Character	PIC X(n).
Packed Decimal	PIC S9(n)V9(n) COMP-3.
Unsigned Packed Decimal	PIC 9(n)V9(n) COMP-3.
Zoned Decimal	PIC S9(n).
Unsigned Zoned Decimal	PIC 9(n).
Half Word	PIC S9(4) COMP.
Unsigned Half Word	PIC 9(4) COMP.
Full Word	PIC S9(8) COMP.
Unsigned Full Word	PIX 9(8) COMP.
Double Word	COMP-2.
Variable Length Character	STUCTURE. LENGTH PIC S9(4) COMP. DATA PIC X(n).

Mapping Process
Generating USE Grammar

- **Select Data Catalog Window**
- **Select Generate**
  - File->Generate USE Statements…
- **Identify file name and**
  - Save to disk or,
  - Use built-in FTP support to transmit file to Z/OS
- **Review generated grammar**

Mapping Processing
Loading the System Catalogs

- **Once the USE grammar is on Z/OS you run the Meta Data Utility to load the Logical Table definitions into the System Catalogs.**
- **The Meta Data Utility:**
  - Performs final Table and Column validation
  - Obtains additional DBMS-specific information
  - Populates the System Catalog

Sample Java Queue Reader

```
 qMgr = new MQQueueManager(qManager);
int openOptions = MQC.MQOO_INPUT_EXCLUSIVE |
                  MQC.MQOO_BROWSE |
                  MQC.MQOO_FAIL_IF_QUIESCING ;
     /*** Open the queue ***/
     MQQueue local_queue =  qMgr.accessQueue(queue,
    openOptions);
     MQMessage retrievedMessage = new MQMessage();
     MQGetMessageOptions gmo = new MQGetMessageOptions();
     retrievedMessage.characterSet = ccsid;
     local_queue.get(retrievedMessage, gmo);
     /* get message*/
     int len = retrievedMessage.getDataLength();
     String msgt = retrievedMessage.readString(len);
    /*process message*/
     System.out.println(msgt);
    /*wrap up*/
     local_queue.close();
qMgr.disconnect();
```

Synchronization

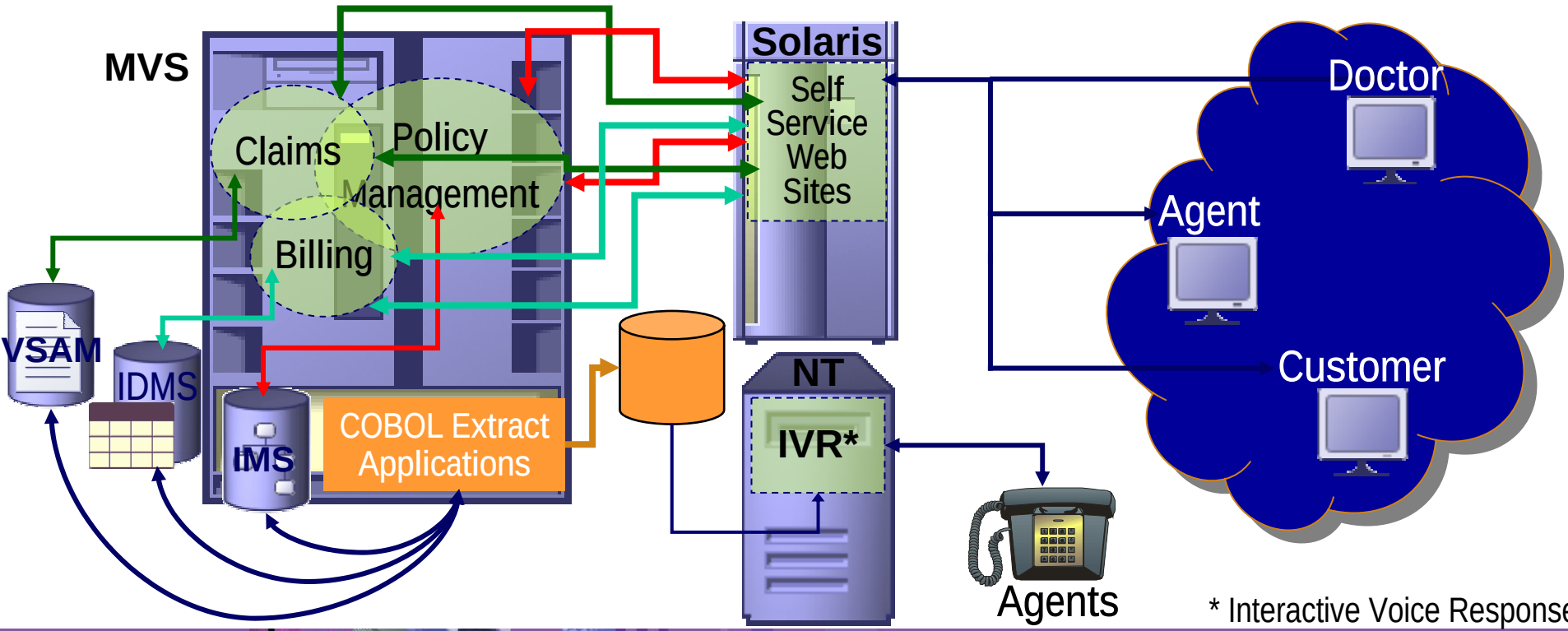
- **Synchronization involves the, well, synchronization of two, or more, unlike Databases**
  - Many transformations
  - One to One, One to Many, Many to One
  - Multiple products (Cleansing, Transformational, Transactional, ….)
  - Data Marts
  - BI
- **These databases are usually on different platforms**
  - Z/OS
  - Wintel
  - AIX
  - Solaris
  - HP
  - Linux (Z/OS, Wintel, etc.)
  - …
- **These databases can be implemented on different Data Base Management Systems (DBMS)**
  - DB2
  - Oracle
  - SAP
  - IMS, IDMS, …
  - CICS
  - …

Solutions for Synchronization

- **Two Phase Commit**
  - Only for sub-components
- **Staged Replication**
  - Only for sub-components
- **Home Grown**
  - Closely Tailored
  - Non-Adaptive
- **Federation**
  - Aggregate and Pull
  - Polling
  - Synchronous
- **Event Publishing**
  - Aggregate and Push
  - Loosely Coupled and Asynchronous
  - Allows for ‘triggered’ operation

Integrated Voice Response (IVR) System Example

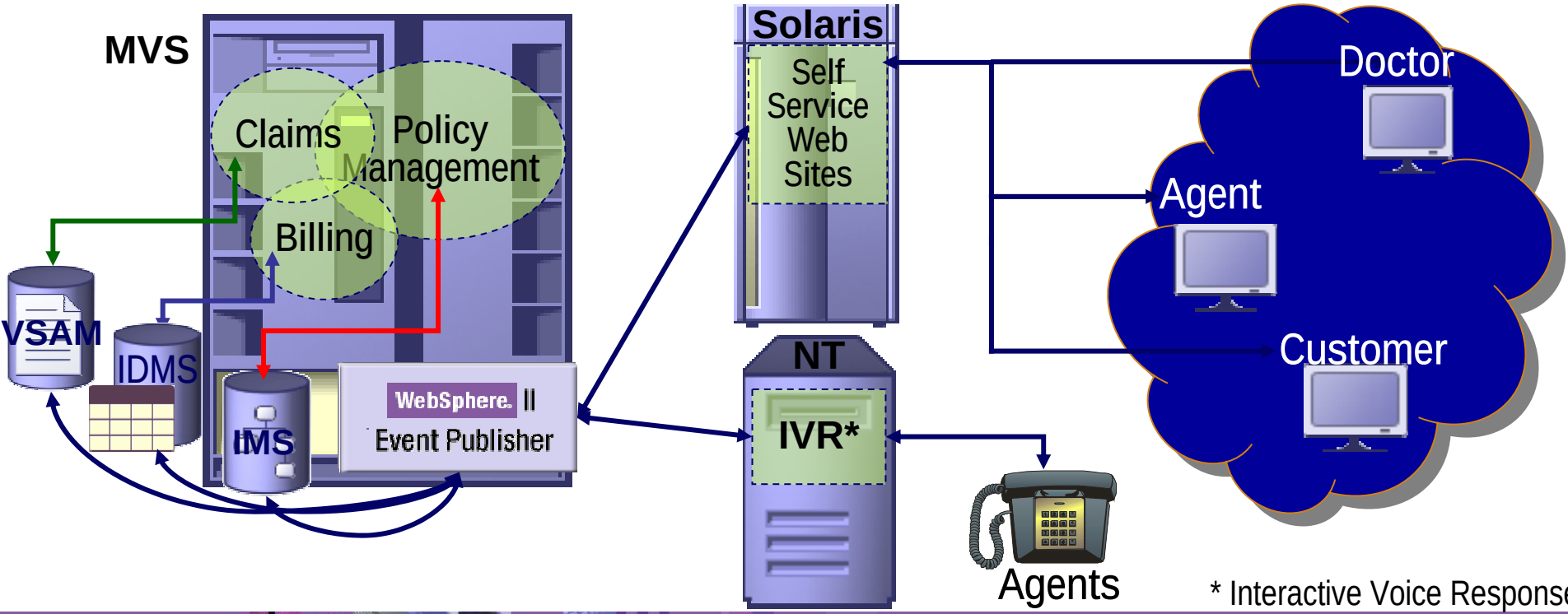
- Solution a: copy data to non-mainframe environments
  - Estimated cost $2M
  - Data refreshed every 30 hours or so
- Solution" b: integrate the transactions
  - Estimated cost 10,000 man-hours per application



IBM solution -- empower self-service

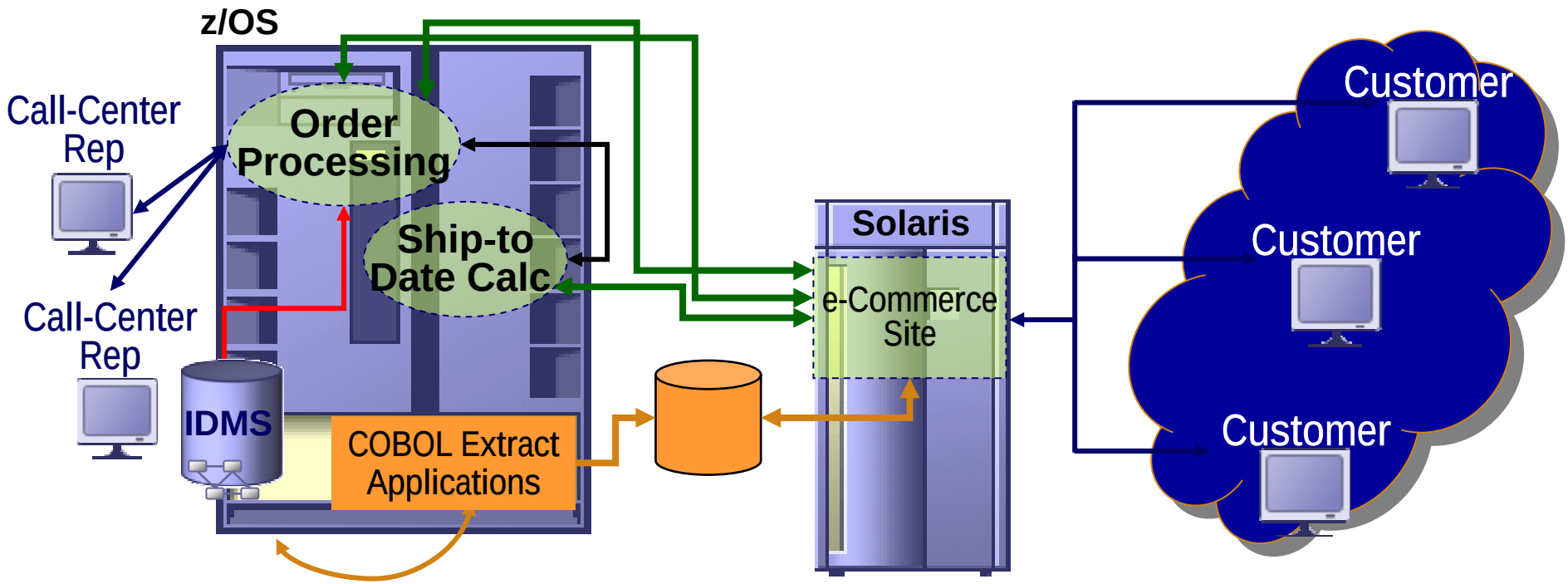
Provide up-to-the-minute policy, claims and accounting information

- Connect interactive voice response (IVR) system to IMS, VSAM & IDMS
  - $250K versus $2M
- Connect operational data with self-service Web sites
  - 200 man-hours versus 10,000



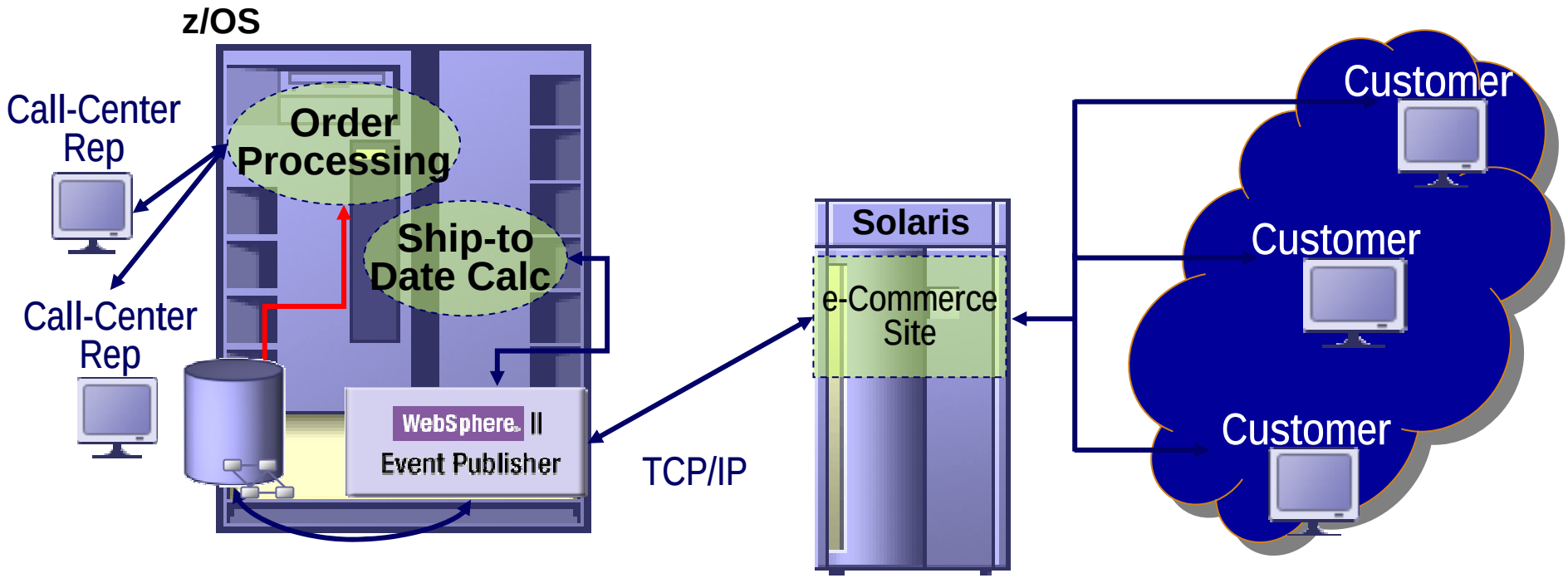
Order Processing Example

- Solution a: copy data to non-mainframe environments
  - Out-of-date inventory = potential “out-of-stock” sales
  - Dissatisfied buyers stop shopping here
- Solution” b: integrate the CICS transactions
  - Java Web developers have no mainframe skills
  - e-Commerce is “shopping” NOT “order processing”



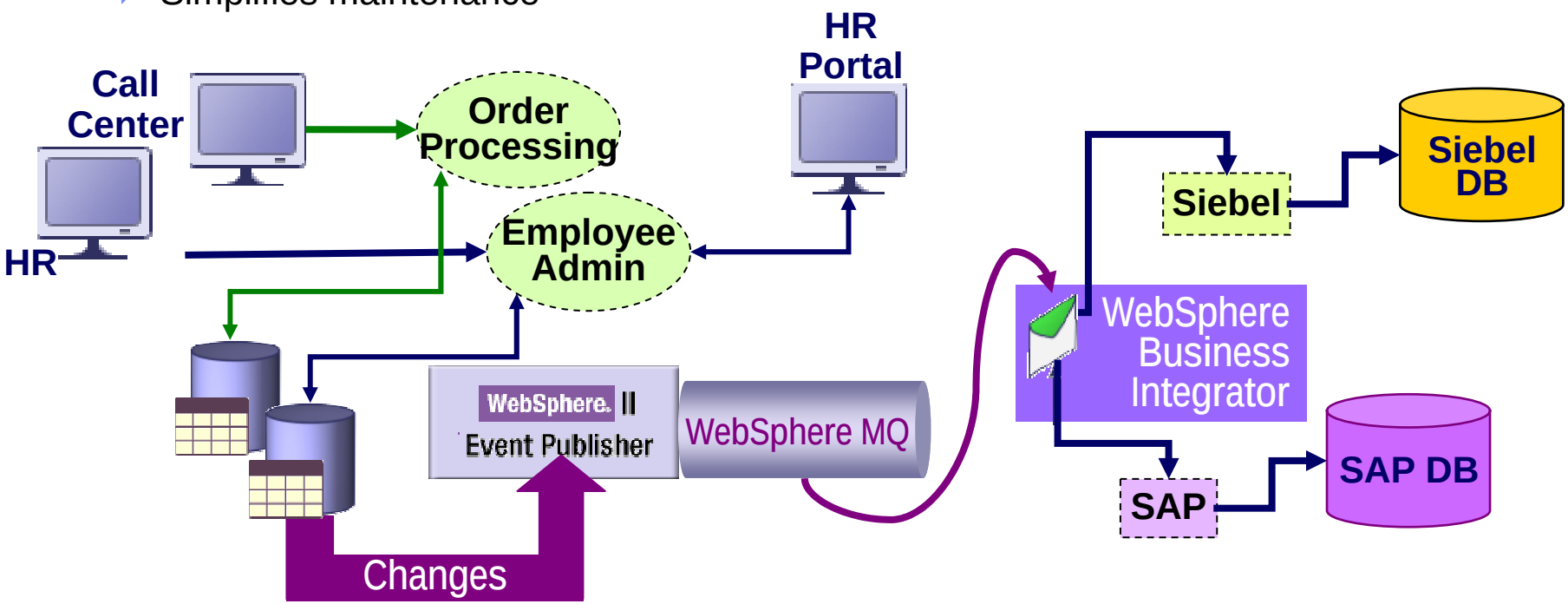
IBM Solution - Single-source mission critical data

- Seamlessly share order processing data and algorithms
  - WebSphere e-commerce applications share critical data, e.g. inventory, pricing…
  - Leverage “common” procedures such as ship-to-date calc when appropriate
  - WebSphere Studio development independent of mainframe skills
  - High performance profile addresses Internet scalability demands
    For example: 52 million transactions daily
    100 milliseconds/transaction for mainframe data access



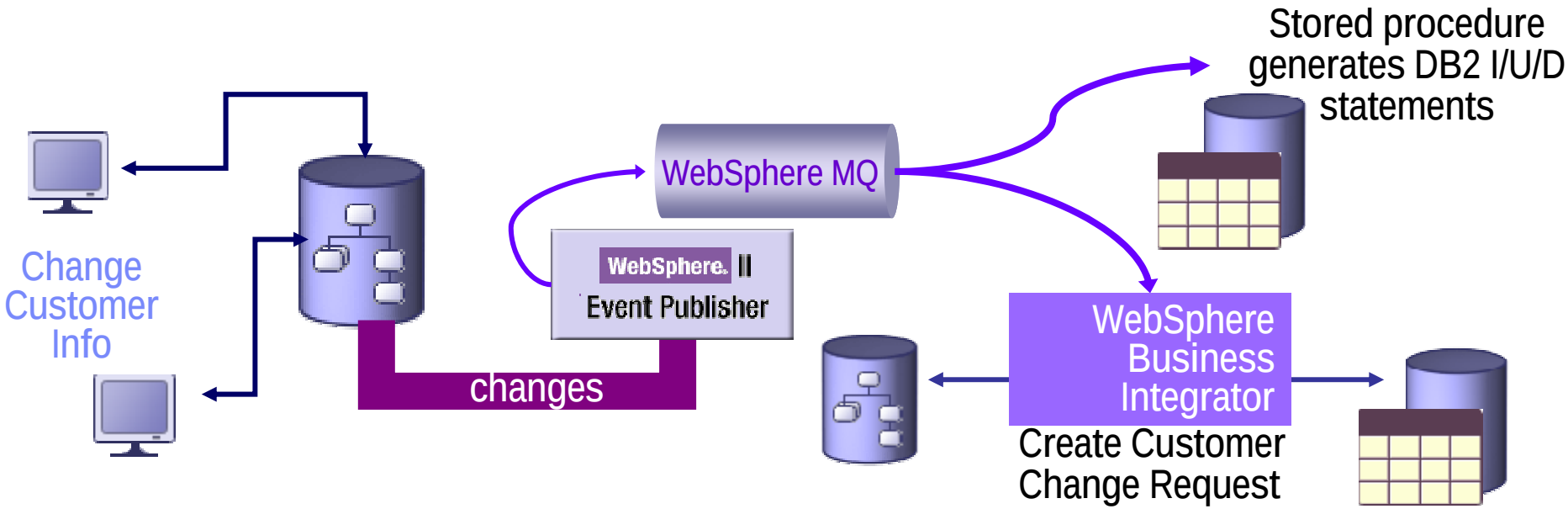
Synchronization example with CRM and SAP

- Near real-time cross-silo data synchronization
  e.g. New order data is automatically pushed to a CRM application
  e.g. VSAM employee data updates are pushed to SAP payroll
  - Loosely coupled integration
  - Minimizes development effort
  - Simplifies maintenance



Customer Profile Management (CRM) example

- **Basic customer profile management**
  - WebSphere "listener application" picks up changes and initiates update
    - Directly to database
    - Using a database stored procedure
- **Complex customer profile management**
  - Push customer updates to WebSphere Business Integrator Event Broker
  - Appropriate updates or transactions initiated on other systems



Event Monitoring

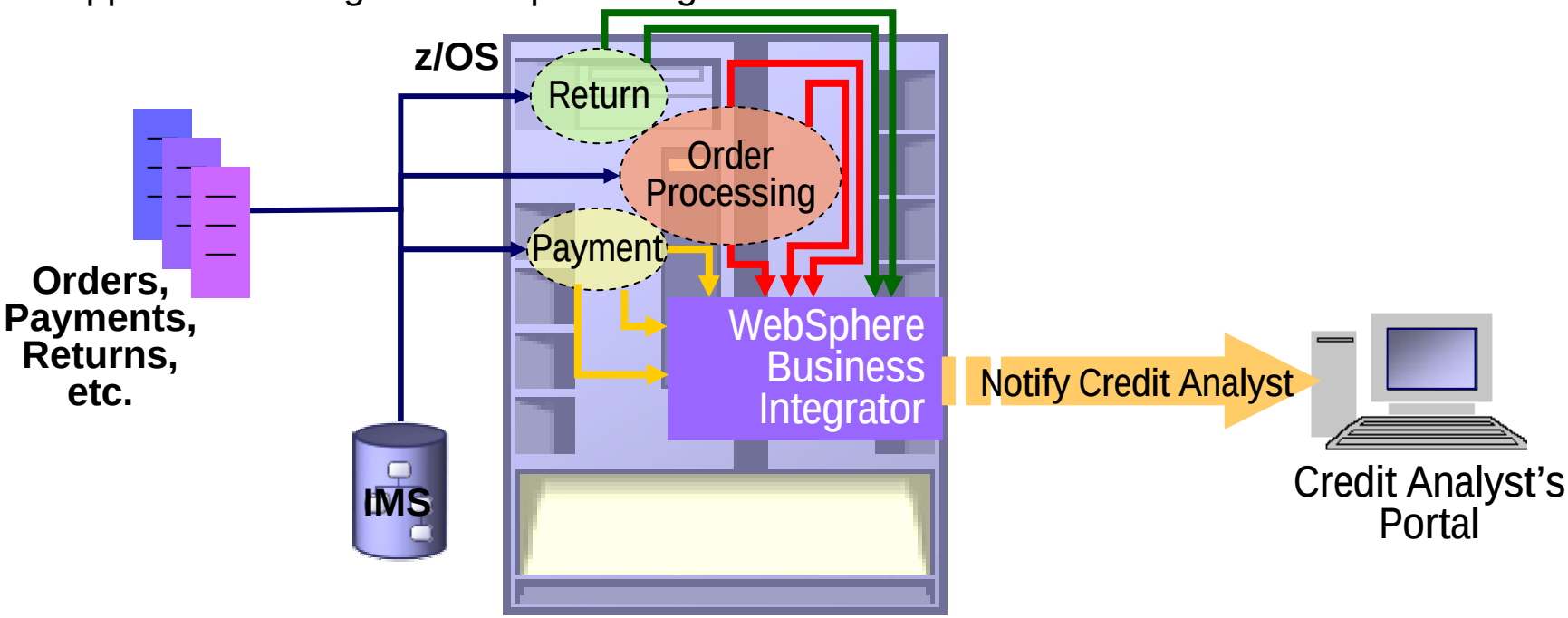
- **Event Monitoring involves the processing of discrete events**
  - Events can be application, or data related
  - Data events can be as small as a change to a single column, or data element
  - Thresholds and Water Marks
  - Counters and Aggregates
  - Rules and Actions
- **Monitored databases and transactions can be on different platforms**
  - Z/OS
  - Wintel
  - AIX
  - Solaris
  - HP
  - Linux (Z/OS, Wintel, etc.)
  - …
- **These events can originate from different Data Base Management Systems (DBMS) and Transaction Managers**
  - DB2
  - Oracle
  - IMS, IDMS, …
  - CICS
  - …

Solutions for Event Monitoring

- **Data Base specific Triggers**
  - Transactional impact
- **Exits**
  - Home Grown
  - Inflexible
- **EAI**
  - Centralized ‘Hub’ approach
  - Message based
- **Event Publishing**
  - De-Centralized ‘Source” Architecture
  - Data based
  - Transaction based

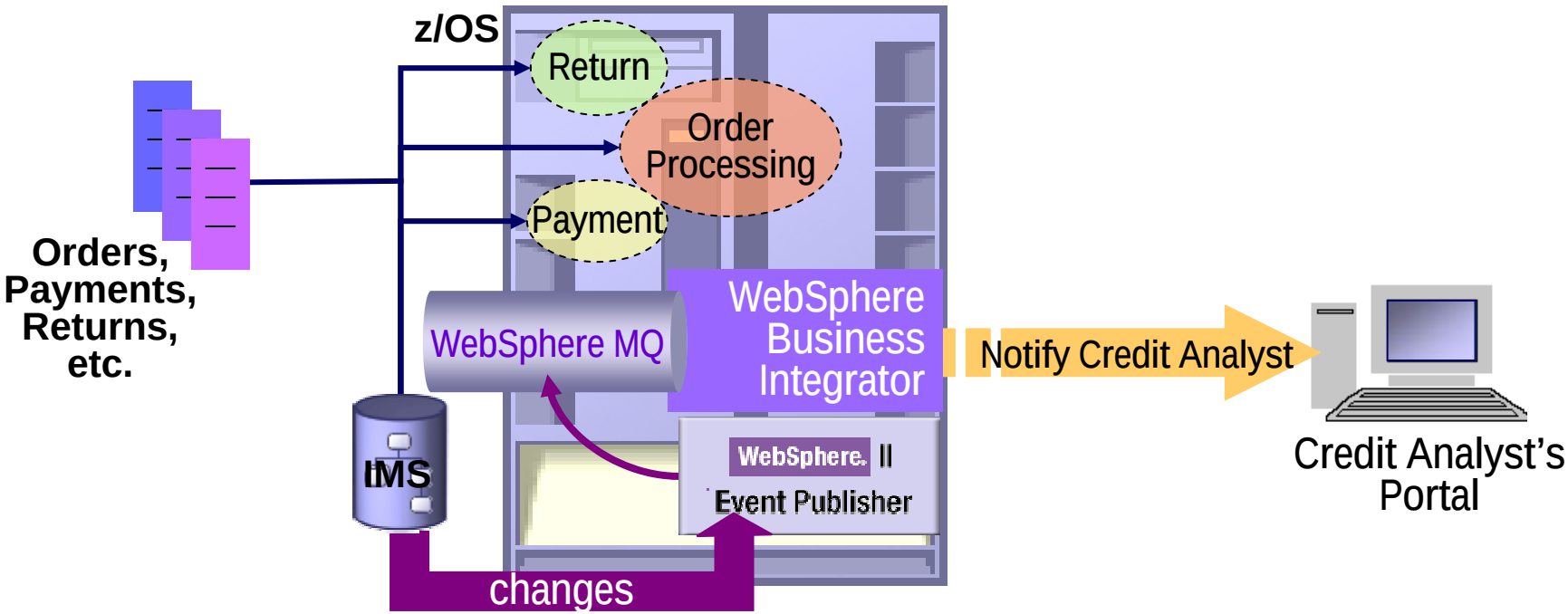
Event Monitoring using “traditional” EAI integration

- **Business integration using application hooks**
  - **Complex**
    One hook for each process involved
    Many processes can impact the same data
  - **Maintenance intensive**
    Application changes can impact integration



Event Monitoring using Event Publishing

- **Event Notification using data events**
  - **Simplified implementation**
    Centralized data used to drive event notification
  - **Simplified maintenance**
    Loosely-coupled – application changes rarely impact integration



WebSphere App Server

Stock Price Update

Cell Phone #
Stock Symbol
High/Low Price

WebSphere App Server

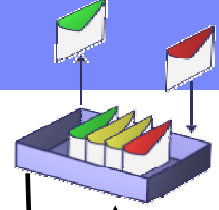
Stock Monitor

Price Change

$>high or $<low

yes

no



WebSphere MQ for z/OS

Text Msg:
IBM Stock
Hit $200!

XML Message

SQL Request

DB2 Information Integrator
Classic Federation

Query Processor

VSAM Connector

DB2 Information Integrator
Event Publisher

Correlation Service

Distribution Service

VSAM Change Capture Agent

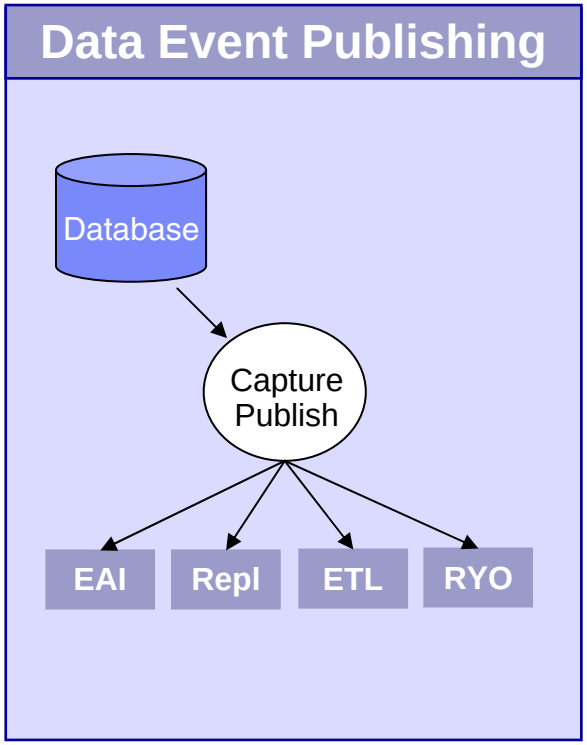
Store/Retrieve
VSAM Data

VSAM
Stock Records
•Symbol
•Price

Data Changes

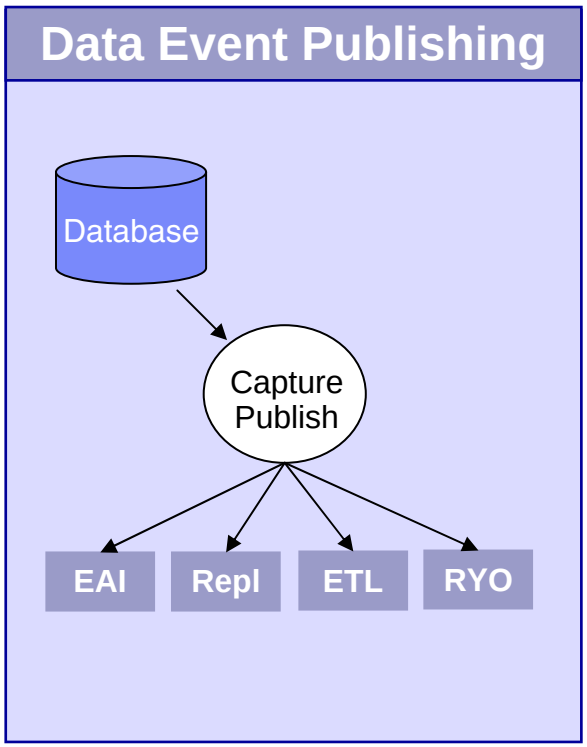
MVS Log

WHAT: Data Event Publishing



- **Message-based publishing based on event capture from a database**
- **Add-on to EAI, ETL, or Replication**

WHY: Data Event Publishing



- **Application to Application Messaging**
  - Drive downstream applications or APIs based on transactional data events
  - Reduce application development and maintenance
  - Reduce performance impact to source applications
  - Reduce availability impact to source applications
- **Meet Auditing Requirements**
  - Capture and store information regarding what changes were made to critical business data and by whom
- **Event Notification**
  - Stream changed data information to Web interfaces
  - Stream only particular events of interest (filter data)
- **Warehouse / Business Intelligence**
  - Integrate captured changed data with an ETL tool
  - Perform very complex transformations
  - Use a specific transaction format to update target

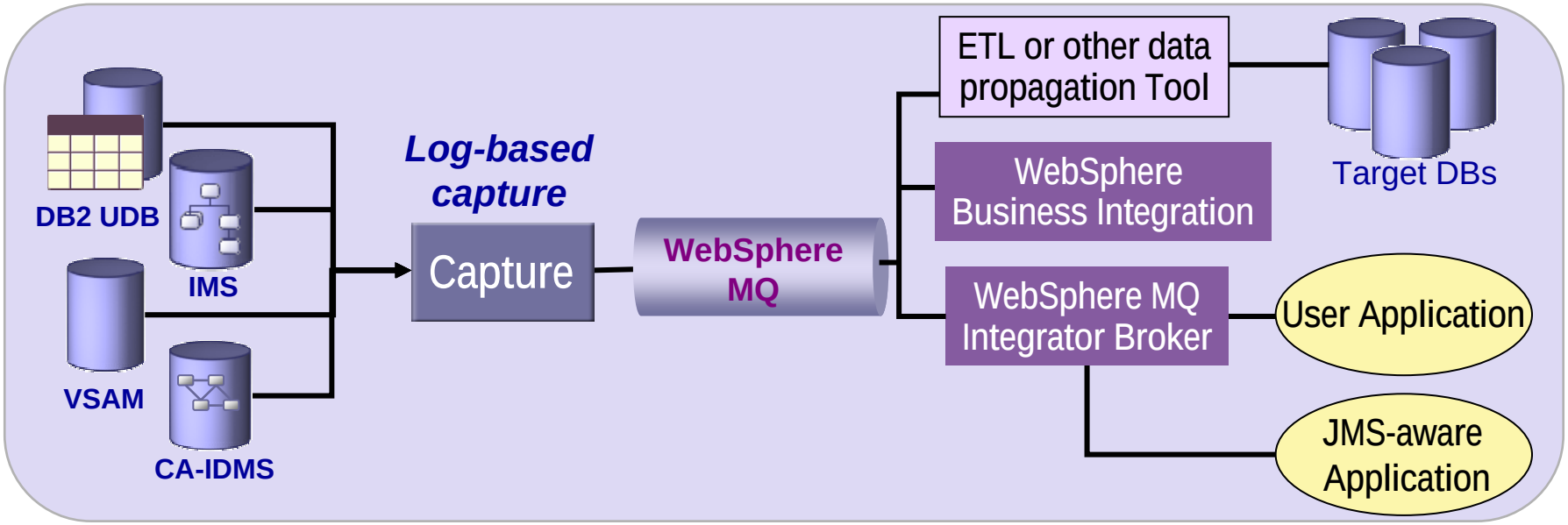
WHY: Data event publishing … facilitates business integration

Function

- **Capture data events in real time**
- **Publish these data events:**
  - to a message queue for widespread delivery
  - in XML format for widespread use

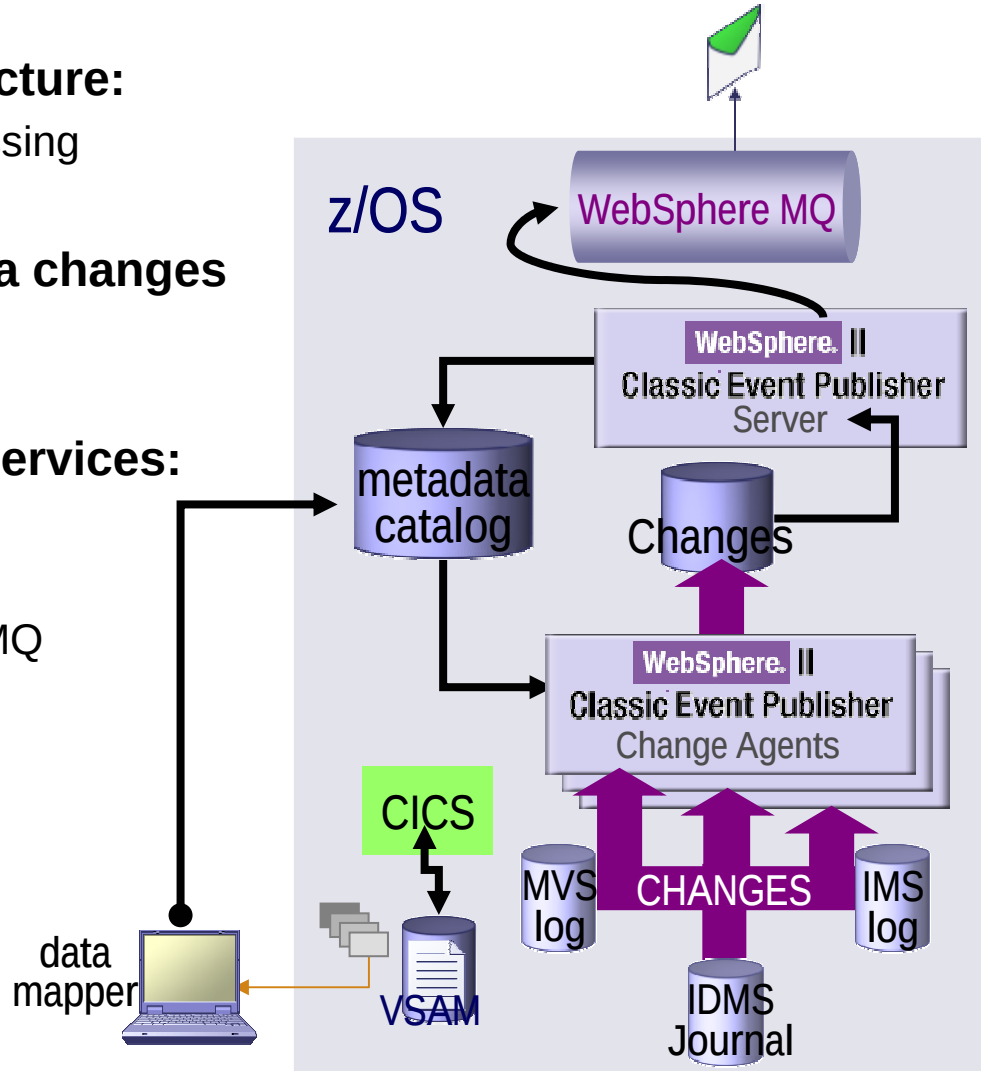
Usage

- **Application to application messaging**
- **Event streaming**
- **Change-only data distribution**

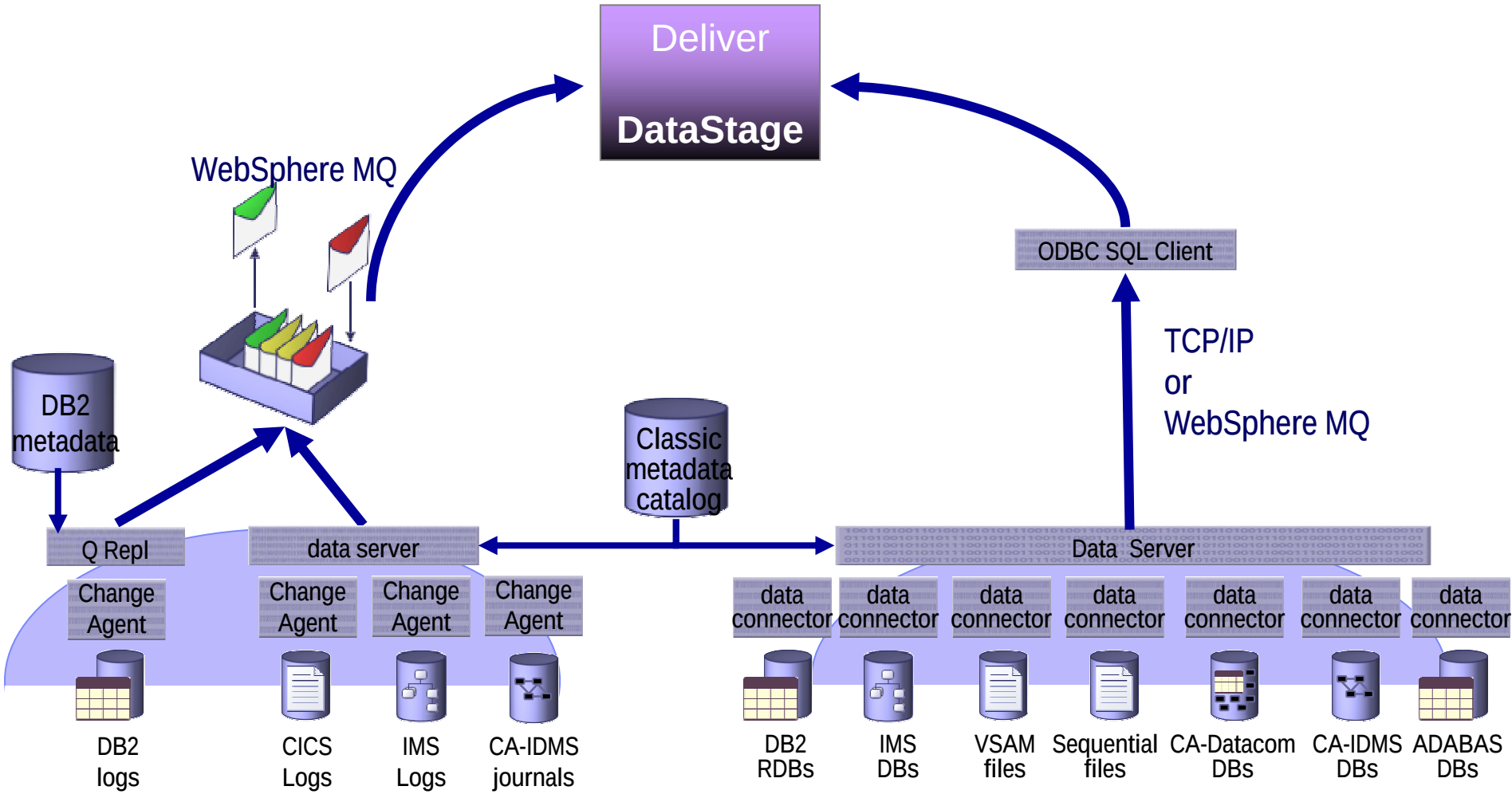


HOW: "Classic" Implementations

- **Shares Classic Federation infrastructure:**
  - Metadata management & catalog processing
  - Server infrastructure
- **Change capture agents monitor data changes**
  - Active via log exits or log files
  - Recovery via log files
- **Server's correlation & distribution services:**
  - Reformat data into relational model
  - Are transaction aware
  - Publish XML messages to WebSphere MQ
  - Handle recovery



EII and ETL


Deliver
DataStage
WebSphere MQ
ODBC SQL Client
TCP/IP
or
WebSphere MQ
DB2
metadata
Classic
metadata
catalog
Q Repl
data server
Data Server
Change
Agent
Change
Agent
Change
Agent
Change
Agent
data
connector
data
connector
data
connector
data
connector
data
connector
data
connector
data
connector
DB2
logs
CICS
Logs
IMS
Logs
CA-IDMS
journals
DB2
RDBs
IMS
DBs
VSAM
files
Sequential
files
CA-Datacom
DBs
CA-IDMS
DBs
ADABAS
DBs