



Workshop: How to populate a Data Warehouse with VSAM data from z/VSE

WAVV 2013

Wilhelm Mild
IT Architect
IBM Lab Germany
mildw@de.ibm.com

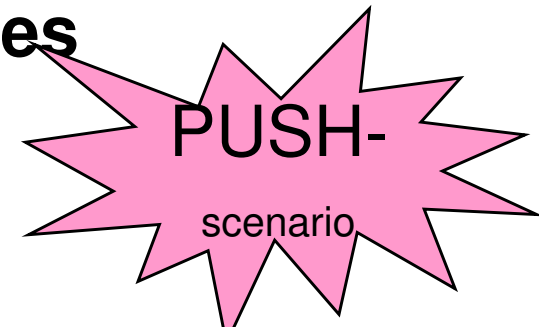
Ingo Franzki
Connectors Specialist
IBM Lab Germany
ifranzki@de.ibm.com

Solutions with VSAM PUSH scenario using VSAM Redirector functions

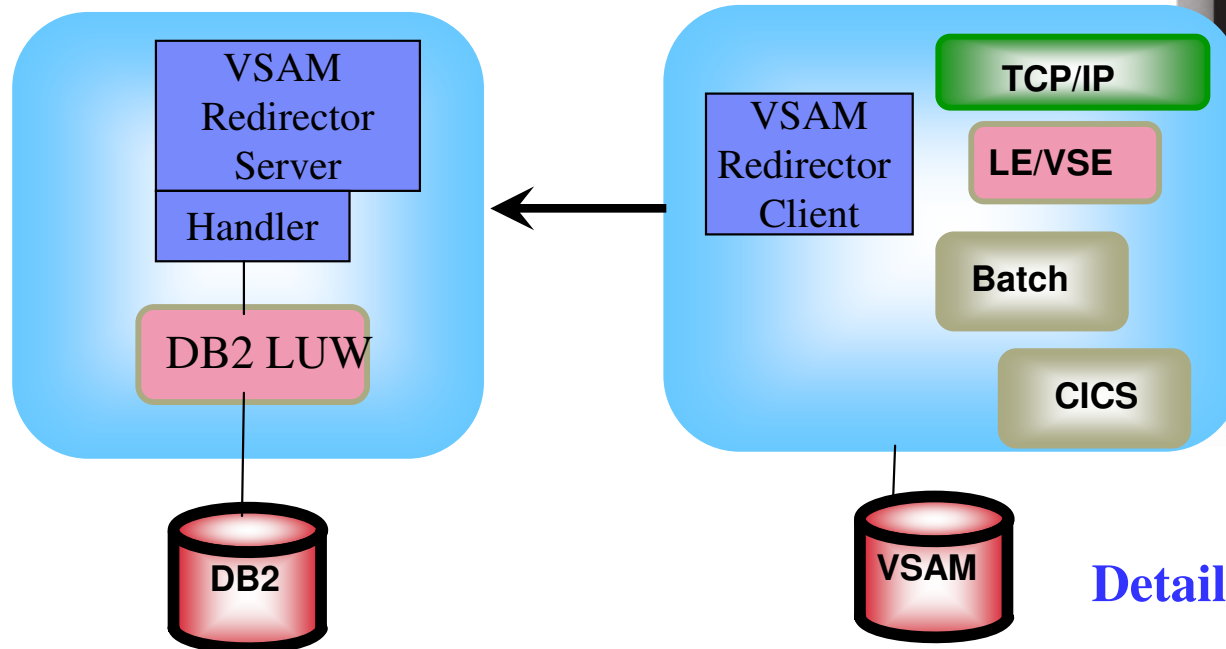
- Push VSAM data to relational databases with VSAM applications
- Synchronize VSAM data with a DB2 data Warehouse
- Collect changes in VSAM files with Redirector Capture
- Enable MQ Series for VSAM applications without application change
- Data Warehouse with DB2 Linux on System z

PUSH scenario: VSE/VSAM applications, access remote relational databases

- (1) Real time VSAM access to relational databases
 - a) synchronization (two phase commit of VSAM and DB2)
 - b) Real time access to DB2 (no VSAM access anymore)
- (2) VSE local data collection for VSAM
 - a) Capture Exit and Incremental FTP, processing
 - b) MQ Exit and MQ Series solutions



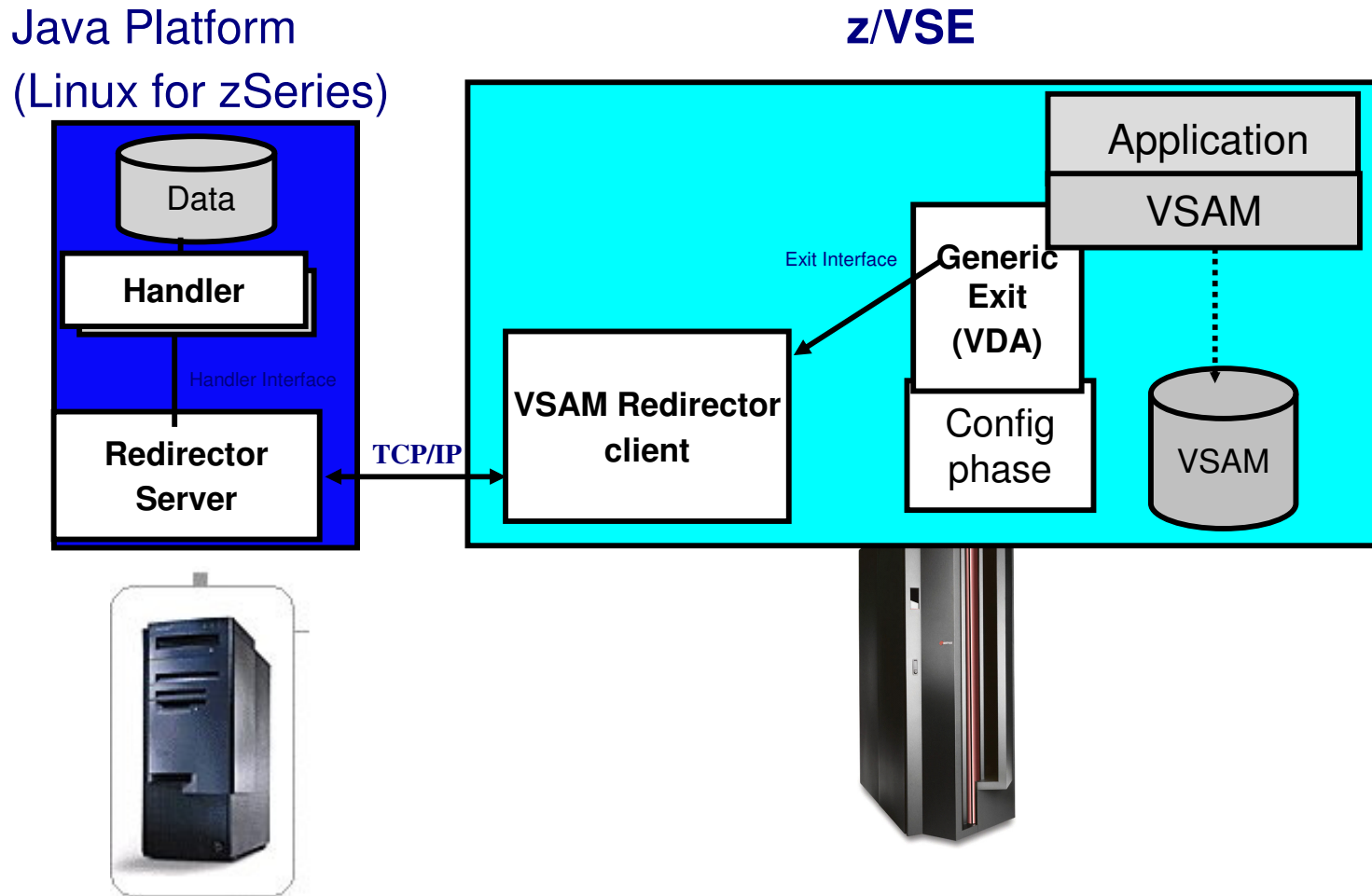
z/VSE Server



Details in Redirector session

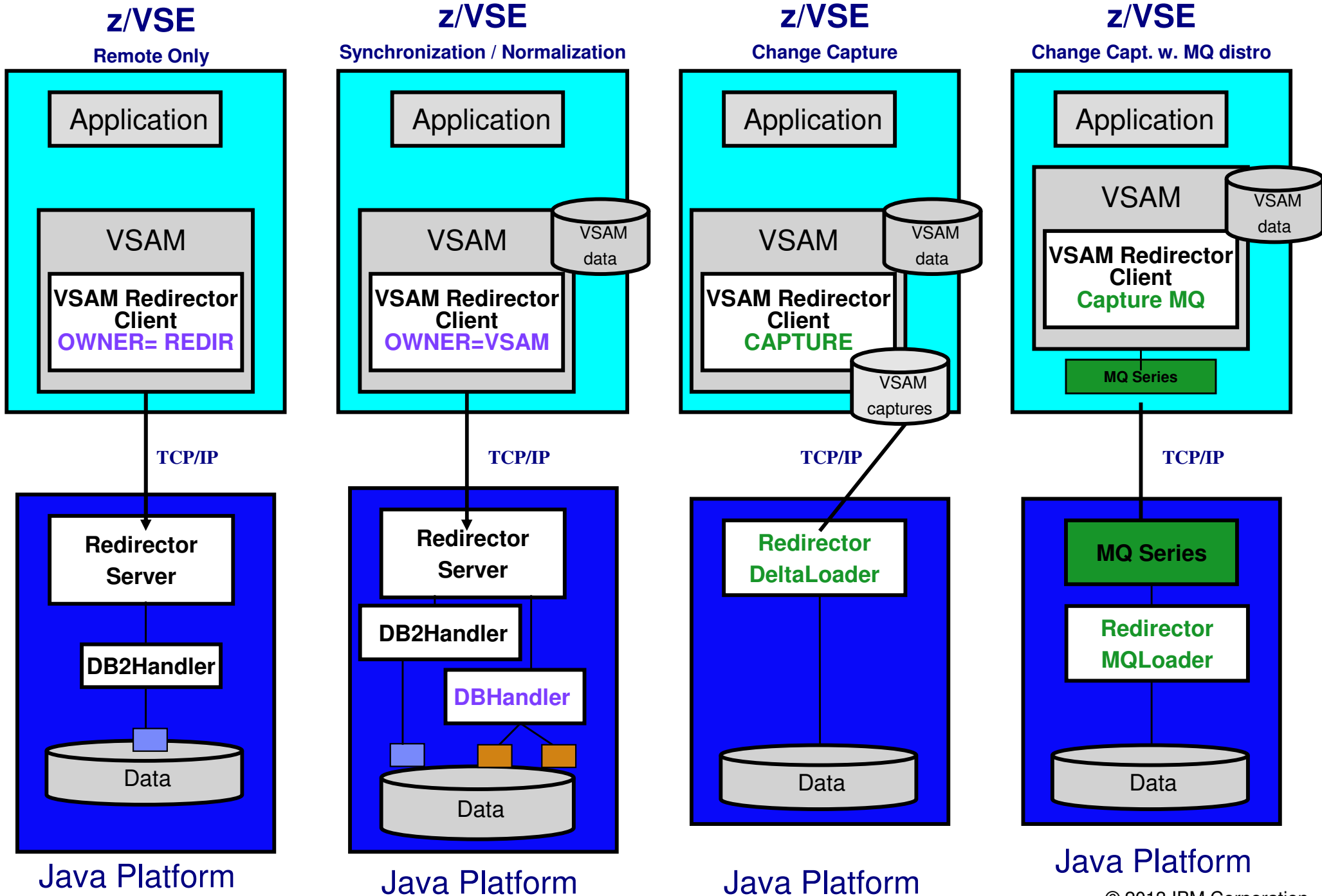
(1) Remote processing

a) Synchronization of VSAM with a database



No changes required in applications (CICS, batch).

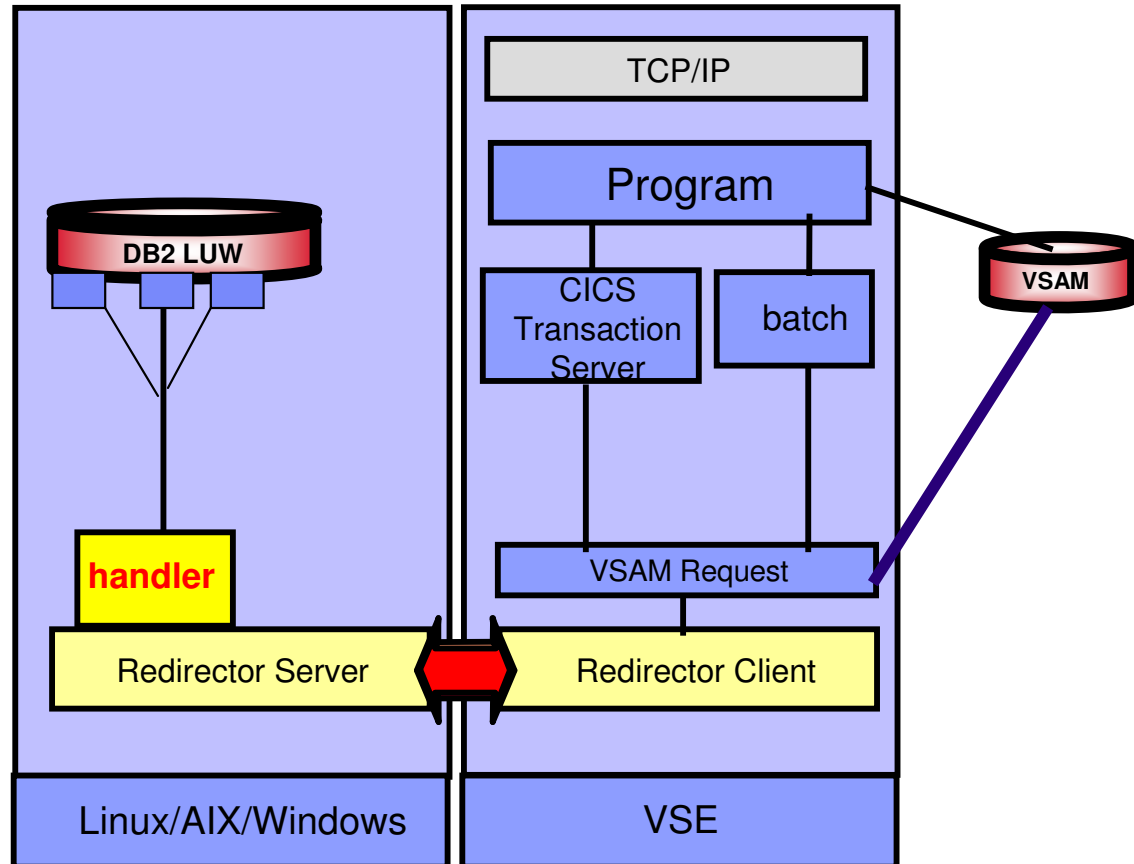
Redirector Solution Overview



Data synchronization – Normalization

VSE/VSAM Redirector can store VSAM data normalized

- ▶ No changes to the existing VSE applications
- ▶ **The new Redirector Handler in z/VSE can store 'VSAM' data in multiple DB Tables**



- ▶ Applications on VSE should be able to access DB2 data on Linux
- ▶ Population of DB2 LUW on Linux with VSAM using VSAM Redirector. (VSAM Redirector is part of VSE)

VSAM Redirector – Normalization - Handler

- **Two different Handler with the New VSAM Redirector**

- **Old: DB2Handler**

- Is still packaged with z/VSE
- supports data access consolidation (OWNER=REDIRECTOR) as well as data synchronization (OWNER=VSAM)
- supports DB2, Oracle, MS-SQL, ...
- was enhanced with new data formats (packed, zoned, date, ...)

- **New: DBHandler**

- enables data Normalization
- supports data synchronization only (OWNER=VSAM)
 - supports Record-Types
 - supports lists with fixed and variable length
- supports new data formats (Packed, Zoned, Datum, ...)
 - supports DB2, Oracle, MS-SQL, ...

VSAM Redirector – Normalization - Handler

- New Redirector handler in z/VSE 4.1
- Handler to Normalize VSAM data
 - store one VSAM record in multiple tables
 - based on VSAM indicator fields
 - administrator decision
 - for synchronization only (owner = VSAM) - READS are done from VSAM
 - relation between tables need to be unique
 - definitions via GUI (mapping configuration)
 - SQL loader provided for database load
 - RedirLoader - fast initial LOAD of a database from VSAM
 - MQLoader - MQ trigger application
 - DeltaLoader – Processing of the Delta file – insert into the database

VSAM Redirector - Normalization – data types

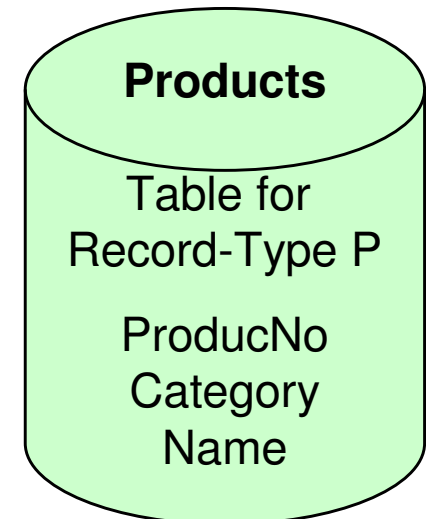
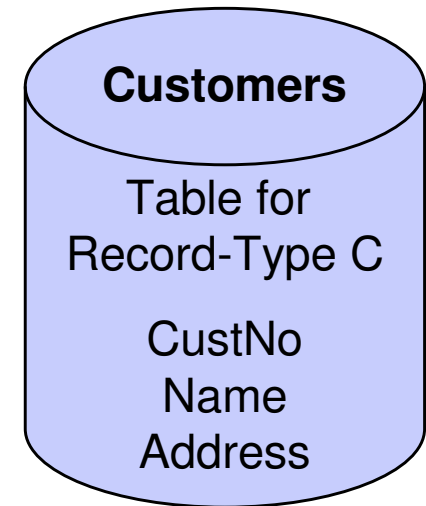
- New extendable Concept: **Converters**
 - One Java-class per data type
 - Open interface
 - New data types can be extended easily
- Data types:
 - STRING
 - BINARY
 - BIT
 - Numbers (INTEGER, PACKED, ZONED, FLOAT, FIXEDTEXT, FLOATTEXT), supports Implied decimal positions
 - DATETIME, TOD
 - HEXCHAR
- Various Options
 - Settings (i.e. date format, number of decimals, ...)
 - Error handling: ONERROR= TERMINATE, TO-NULL, TO_ZERO
 - Text handling: TRIM, PAD, BLANK-TO-NULL, CODEPAGE

VSAM Redirector - Normalization – Record-Types

COBOL Copybook:

```

01 RECORD-3
  03 RECORD-TYPE          PIC X(1)
  03 RECORD-FORMAT-C
    05 CUSTOMER-NO       PIC X(7)
    05 CUSTOMER-NAME     PIC X(25)
    05 CUSTOMER-ADDRESS  PIC X(45)
  03 RECORD-FORMAT-P REDEFINES RECORD-FORMAT-C
    05 PRODUCT-NO        PIC X(7)
    05 PRODUCT-CATEGORY  PIC X(15)
    05 PRODUCT-NAME      PIC X(15).
  
```



Depending on the value of Record-Type field, the data will be store in different database tables

- Type = C → Customers-Table
- Type = P → Products-Table

The association takes place at runtime for each individual record.

VSAM Redirector - Normalization – Lists

COBOL Copybook:

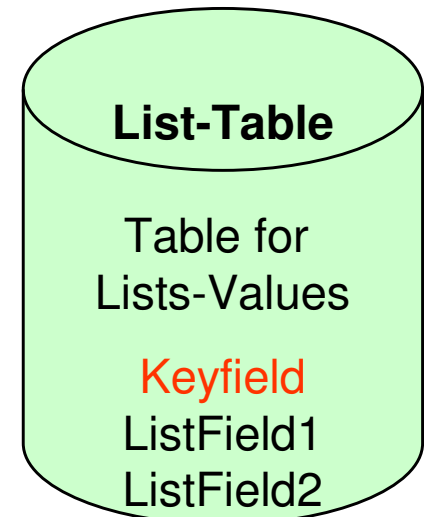
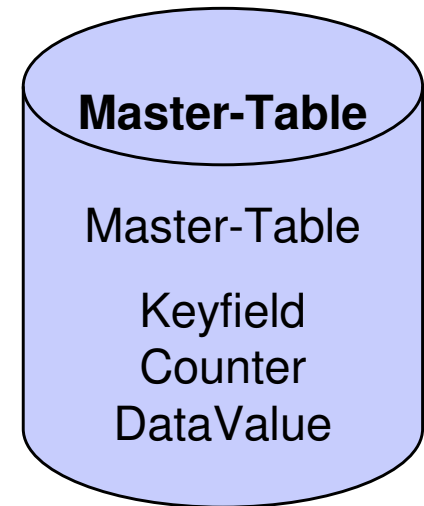
```

01 RECORD-2.
   03 KEYFIELD          PIC X(8).
   03 COUNTER          PIC 9(5) COMP-3.
   03 VARIABLE-LIST OCCURS 1 TO 5
                        DEPENDING ON COUNTER.
       05 LISTFIELD-1    PIC 9(9).
       05 LISTFIELD-2    PIC X(5).
   03 DATAVALUE        PIC X(10).
  
```

Depending on the value of the field Counter, there will be inserted 1 to 5 rows into the List-Table.

The relation to the Master-Table is defined through the foreign key Keyfield in the List-Table

Normalization takes place at runtime for each individual record.



Mapping of **fixed-length lists** with (old) DB2Handler

With DB2Handler: possible, but suboptimal

- All fields are in the same database table

COBOL copybook

```

01 RECORD-1.
   03 KEYFIELD PIC X(8).
   03 DATAFIELD PIC X(10).
   03 FIXED-LIST OCCURS 3.
       05 LISTFIELD-A PIC 9(9).
       05 LISTFIELD-B PIC X(5).
    
```



Database table RECORD 1

- KEYFIELD
- DATAFIELD
- LISTFIELD_A_1
- LISTFIELD_B_1
- LISTFIELD_A_2
- LISTFIELD_B_2
- LISTFIELD_A_3
- LISTFIELD_B_3

Field	KEYFIELD	DATAFIELD	LISTFIELD_A_1	LISTFIELD_B_1	LISTFIELD_A_2	LISTFIELD_B_2	LISTFIELD_A_3	LISTFIELD_B_3
Data	KEY12345	DataValue	1	va11	2	va12	3	va13

Mapping of **fixed-length lists** with DBHandler

With new DBHandler: optimal with *normalization*

COBOL copybook

```

01 RECORD-1.
   03 KEYFIELD PIC X(8).
   03 DATAFIELD PIC X(10).
   03 FIXED-LIST OCCURS 3.
       05 LISTFIELD-A PIC 9(9).
       05 LISTFIELD-B PIC X(5).
    
```

Database table RECORD_1

- KEYFIELD
- DATAFIELD

Database table FIXED_LIST

- KEYFIELD (foreign key relation)
- LISTFIELD-A
- LISTFIELD-B

RECORD_1

Field	KEYFIELD	DATAFIELD
Data	KEY12345	DataValue

FIXED_LIST

Field	KEYFIELD	LISTFIELD-A	LISTFIELD-B
Data	KEY12345	1	va11
Data	KEY12345	2	va12
Data	KEY12345	3	va13

Same keyfield value allows retrieval of all list entries from FIXED_LIST that belong to RECORD_1.

Mapping of variable-length lists

- Not possible with DB2Handler
- Normalized with DBHandler

COBOL copybook

01 RECORD-2.

03 KEYFIELD PIC X(8).

03 COUNTER PIC 9(5) COMP-3.

**03 VARIABLE-LIST OCCURS 1 TO 5
DEPENDING ON COUNTER.**

05 LISTFIELD-1 PIC 9(9).

05 LISTFIELD-2 PIC X(5).

03 DATAVALUE PIC X(10).



Database table RECORD 2

- KEYFIELD
- COUNTER
- DATAVALUE

Database table VARIABLE LIST

- KEYFIELD (foreign key relation)
- LISTFIELD-A
- LISTFIELD-B

Mapping of variable-length lists (sample cont.)

Database table RECORD 2

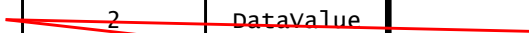
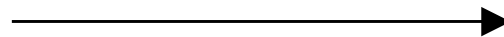
- KEYFIELD
- COUNTER
- DATAVALUE

Field	KEYFIELD	COUNTER	DATAVALUE
Data	KEY12345	2	DataValue
Data	KEY98765	4	DataValue

Database table VARIABLE LIST

- KEYFIELD (foreign key relation)
- LISTFIELD-A
- LISTFIELD-B

Field	KEYFIELD	LISTFIELD-A	LISTFIELD-B
Data	KEY12345	1	va11
Data	KEY12345	2	va12
Data	KEY98765	9	va19
Data	KEY98765	8	va18
Data	KEY98765	7	va17
Data	KEY98765	6	va16



Mapping of record-types

- Not possible with DB2Handler
- **Normalized with DBHandler**

COBOL copybook

01 RECORD-3.

03 RECORD-TYPE PIC X(1).

03 RECORD-FORMAT-C.

05 CUSTOMER-NO PIC X(7).

05 CUSTOMER-NAME PIC X(25).

05 CUSTOMER-ADDRESS PIC X(45).

03 RECORD-FORMAT-P REDEFINES

RECORD-FORMAT-C.

05 PRODUCT-NO PIC X(7).

05 PRODUCT-CATEGORY PIC X(15).

05 PRODUCT-NAME PIC X(15).

No RECORD_3 table, relation of record type value to target table is stored in the configuration:

Type ,C' -> Target table RECORD_FORMAT_C
Type ,P' -> Target table RECORD_FORMAT_P

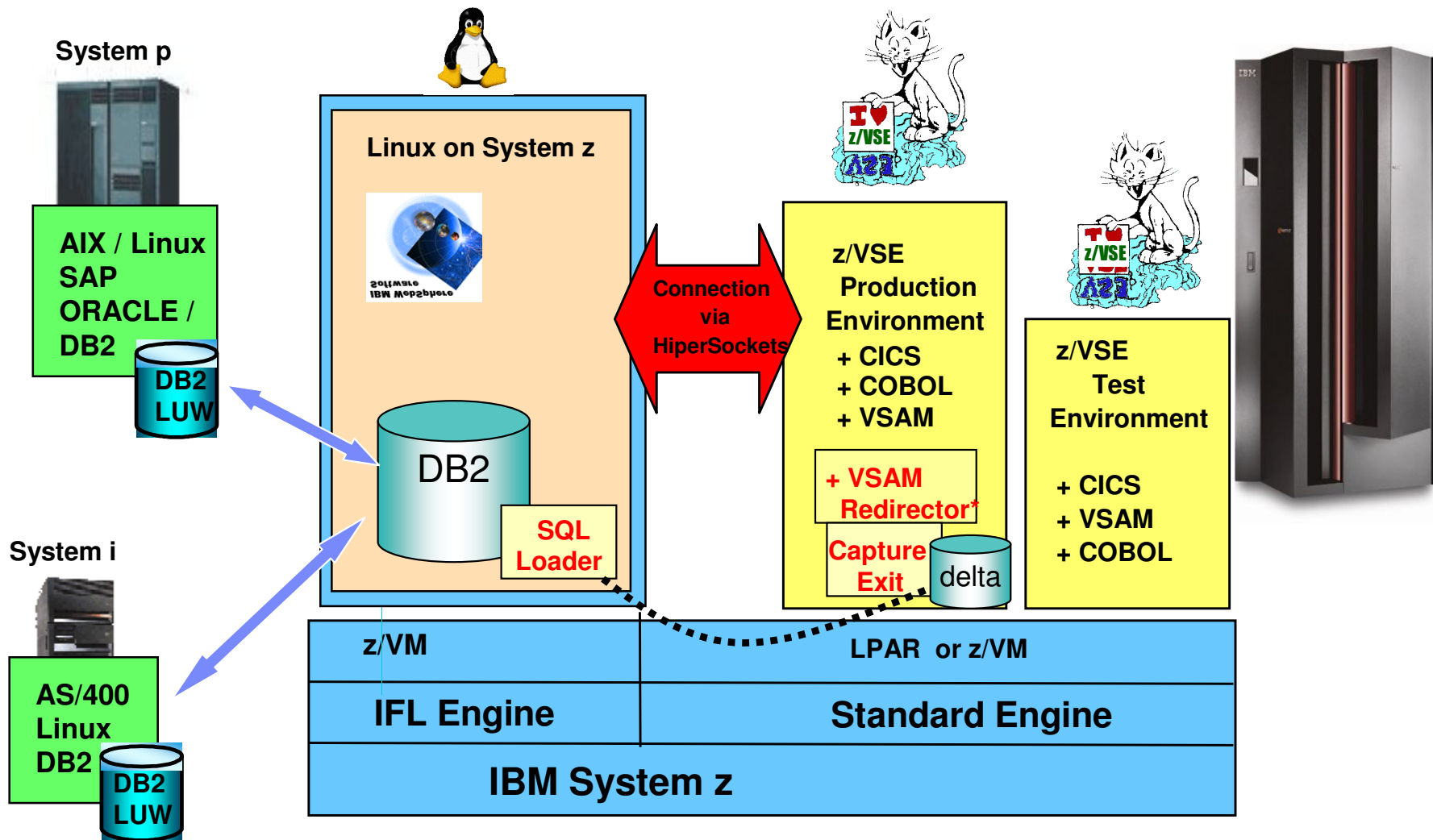
Database table RECORD FORMAT C

- CUSTOMER_NO
- CUSTOMER_NAME
- CUSTOMER_ADDRESS

Database table RECORD FORMAT P

- PRODUCT_NO
- PRODUCT_CATEGORY
- PRODUCT_NAME

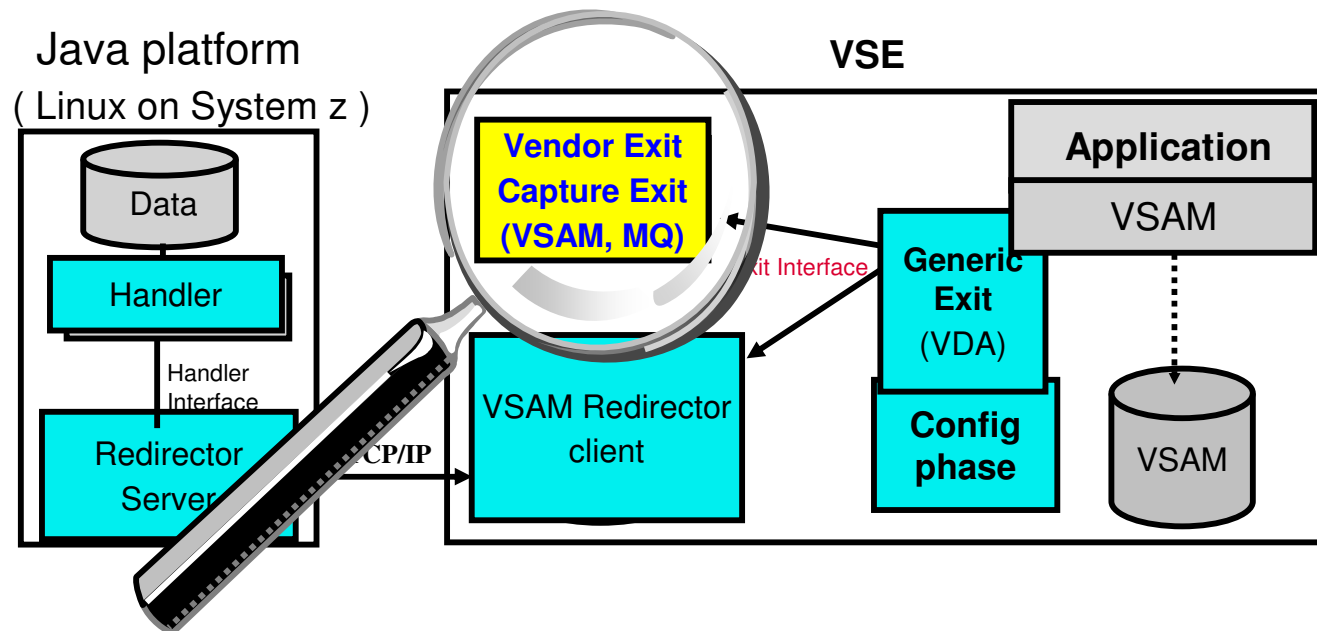
(2) VSAM-capture and apply with DB2 LUW on Linux on System z



(*) VSAM Redirector – Common data store solution – with DB2 on Linux on zSeries Solutions without changes to VSAM programs

VSAM Data collection / transformation / journaling on VSE

Capture Exit



CAPTURE – wit Decision Exit as filter

► Vendor Exit

- user (vendor) written phase for data collection/transformation
- has to comply with the documented **Exit Interface**

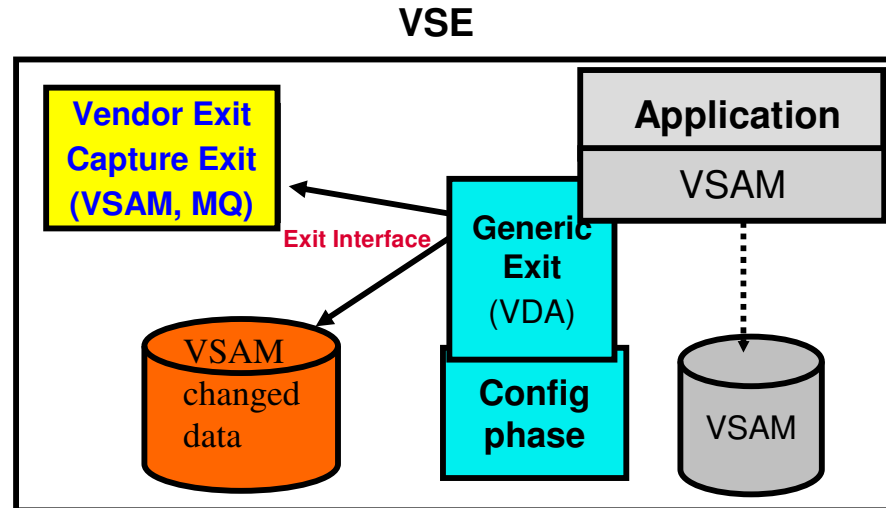
► Capture Exit

- an exit delivered by IBM for **capturing changed VSAM** data
- an exit delivered by IBM for **generating MQ** messages

Note: No chaining of Vendor Exit with VSAM Redirector client supported

VSAM Data collection / transformation / journaling on VSE

Capture Exit



CAPTURE – wit Decision Exit as filter

➤ Vendor Exit

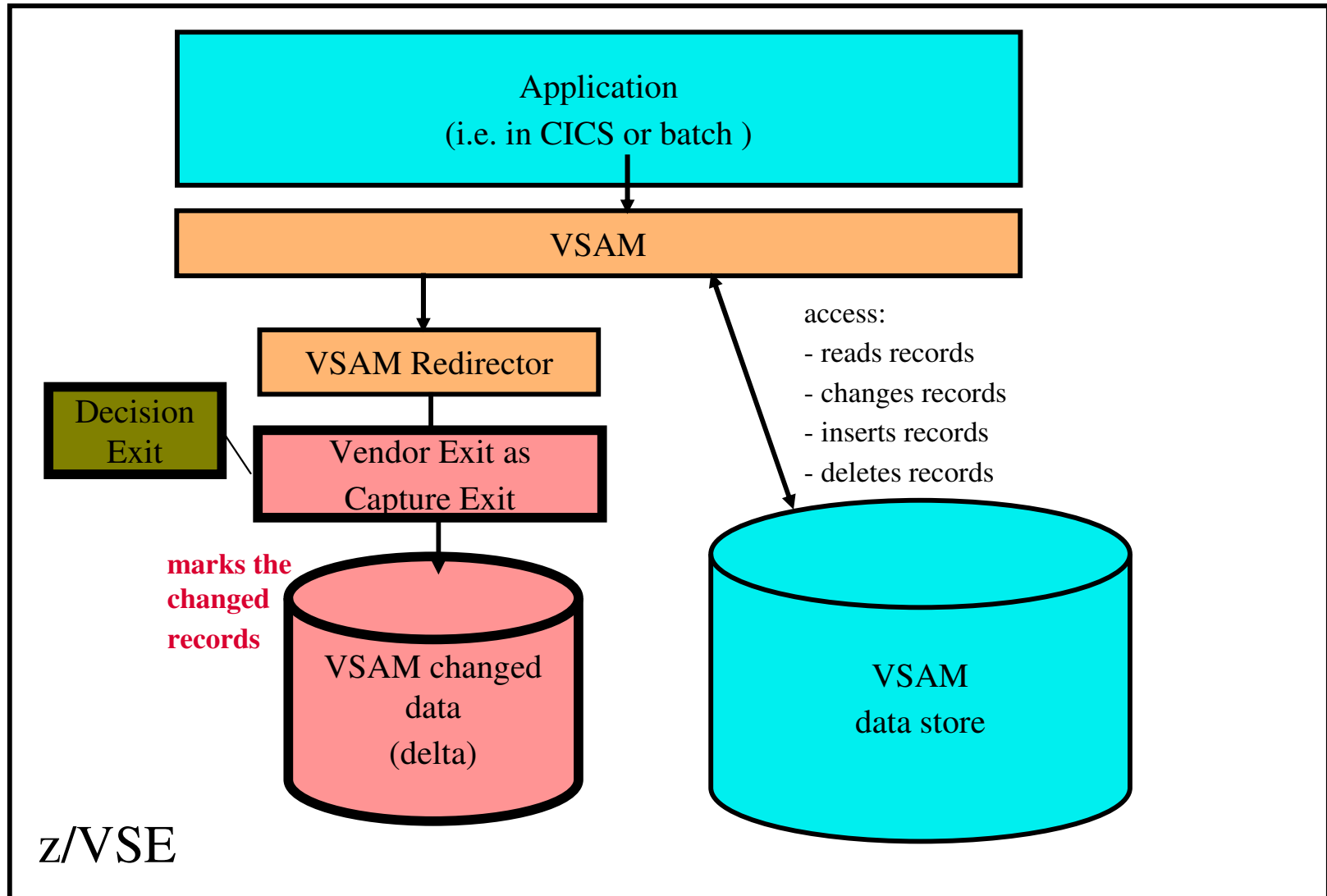
- user (vendor) written phase for data collection/transformation
- has to comply with the documented **Exit Interface**

➤ Capture Exit

- an exit delivered by IBM for **capturing changed VSAM** data
- an exit delivered by IBM for **generating MQ** messages

Note: No chaining of Vendor Exit with VSAM Redirector client supported

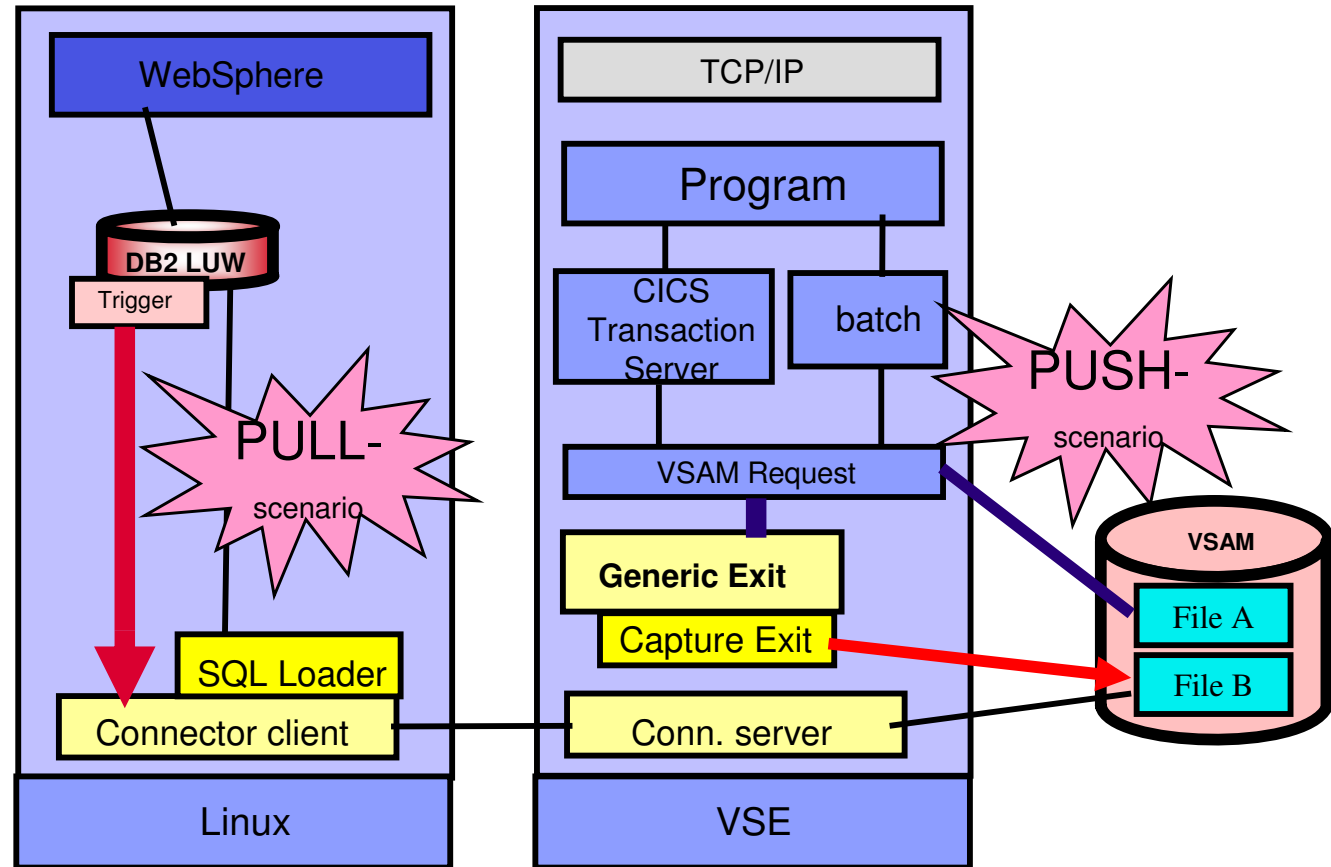
Redirector Capture Architectural View



Solution for incremental synchronization

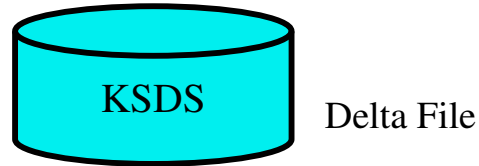
Bidirectional updates with VSE connectors

- ▶ With VSAM Capture – the performance of the VSE production system protected
- ▶ The changes are processed asynchronously and not influencing the production system



- ▶ Collect the changed records in a separate VSAM file
 - ▶ Possibility of cleansing
- ▶ Process them – with the VSE Connectors

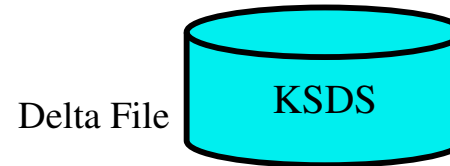
Journaling



Record 1	inserted
Record 2	inserted
Record 3	inserted
Record 2	updated
Record 1	deleted
Record 3	updated
Record 4	inserted
Record 1	inserted
Record 2	updated
Record 4	updated
Record 4	deleted

or

cumulative



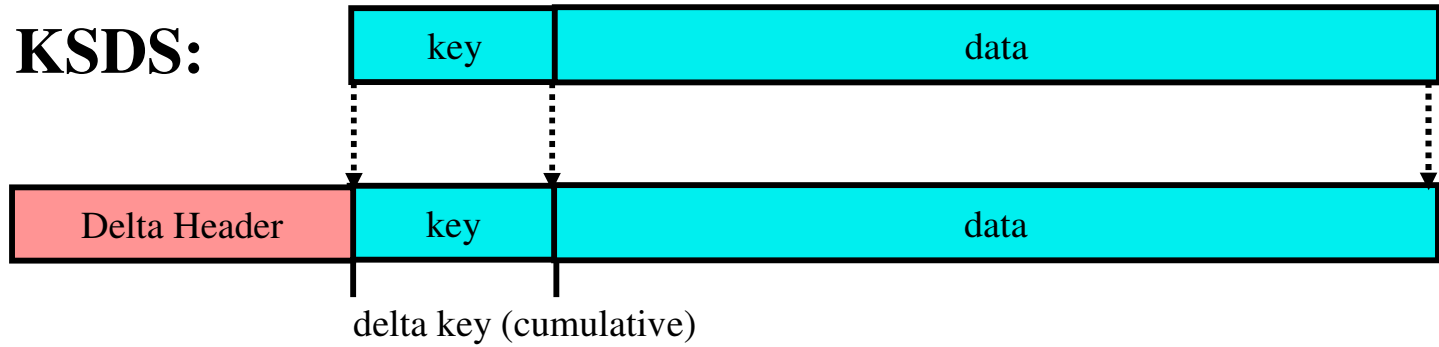
key

Record 1	inserted
Record 2	updated
Record 3	updated
Record 4	deleted

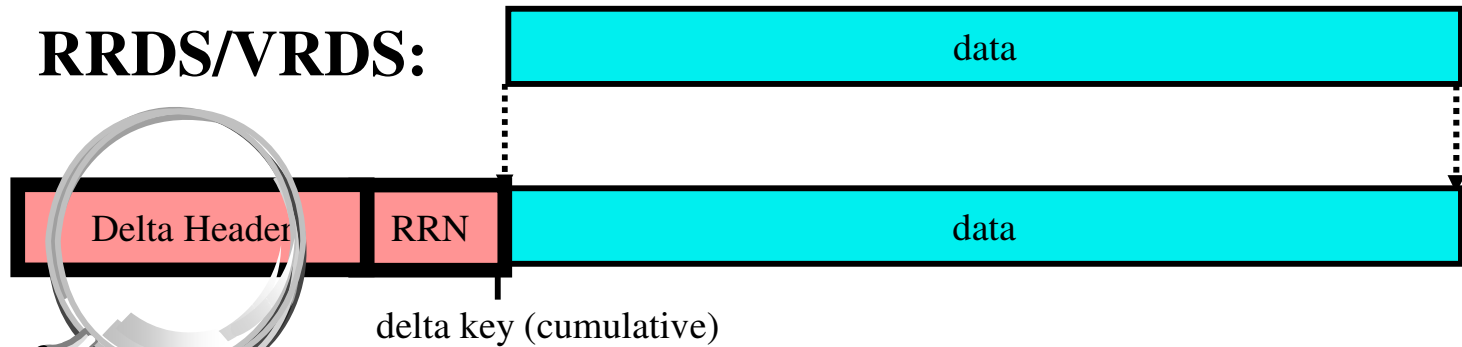
☞ The last version only of a changed VSAM record is stored into the delta file

Delta Record

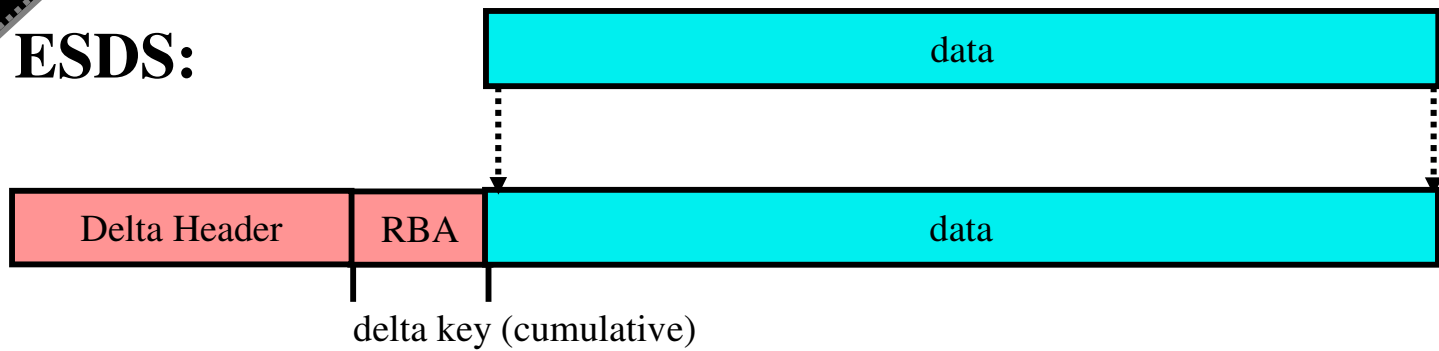
KSDS:



RRDS/VRDS:



ESDS:



Delta Header



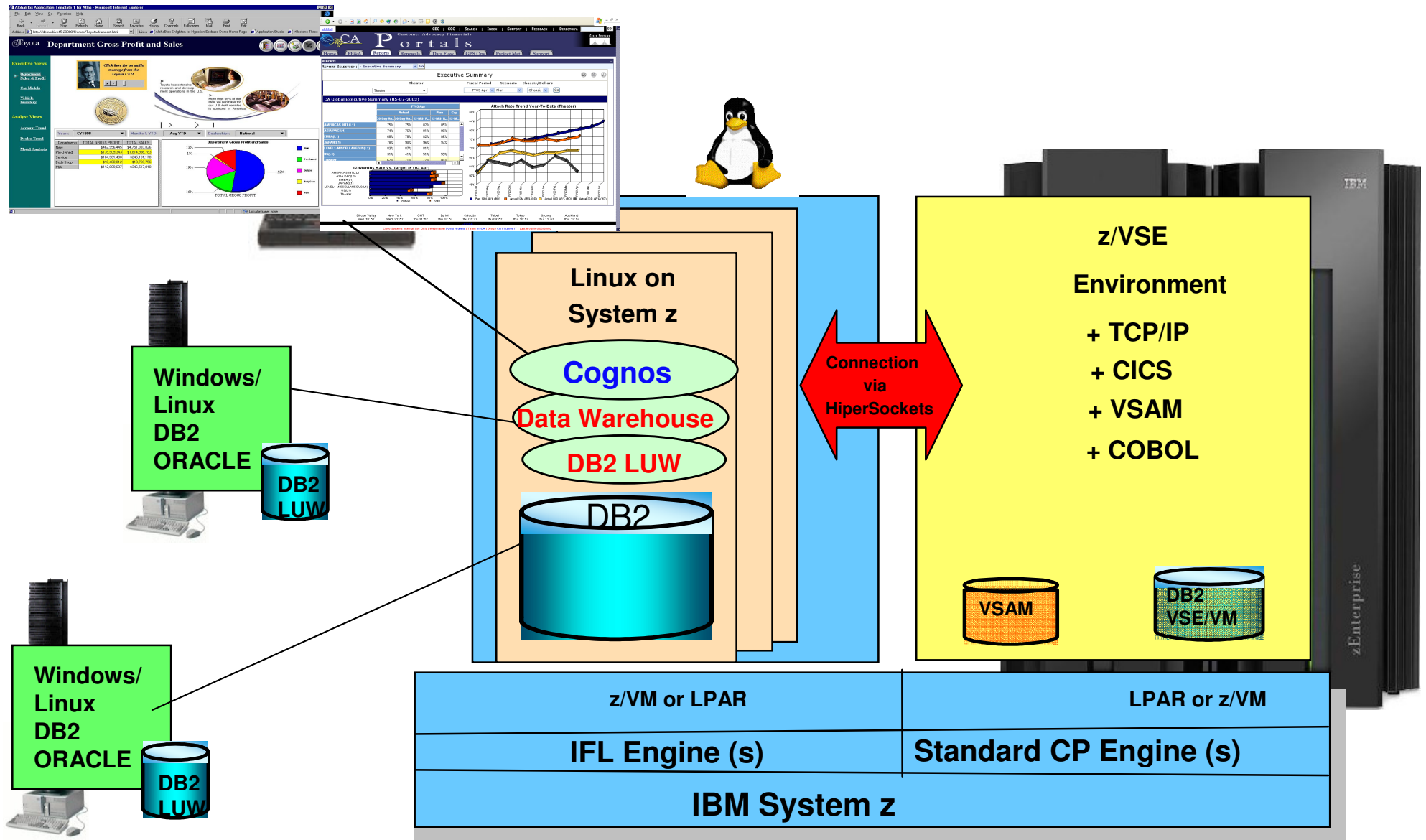
Offset	Parameters	Length	Description
0	TODCLOCK	8	Time of change
8	JobName	8	Job name
16	PHASEName	8	Phase name
24	Origin	8	String from Config or file Label
32	PartID	2	Partition ID (i.e. F2)
34	OpCode	1	I=Insert, D=Delete, U=Update
35	Flags	1	X'01'=RRN/RBA follows
36	RRN/RBA	4	RRN/RBA (RRDS/VRDS/ESDS)

Contains information about:

- **when** change took place (TODCLOCK)
- **who** did the change (Job/Phase/Partition)
- **request type** of change (Insert/Delete/Update)
- **which record** was affected (key/RRN/RBA)

Solution: Data Warehouse and BI with Linux on System z

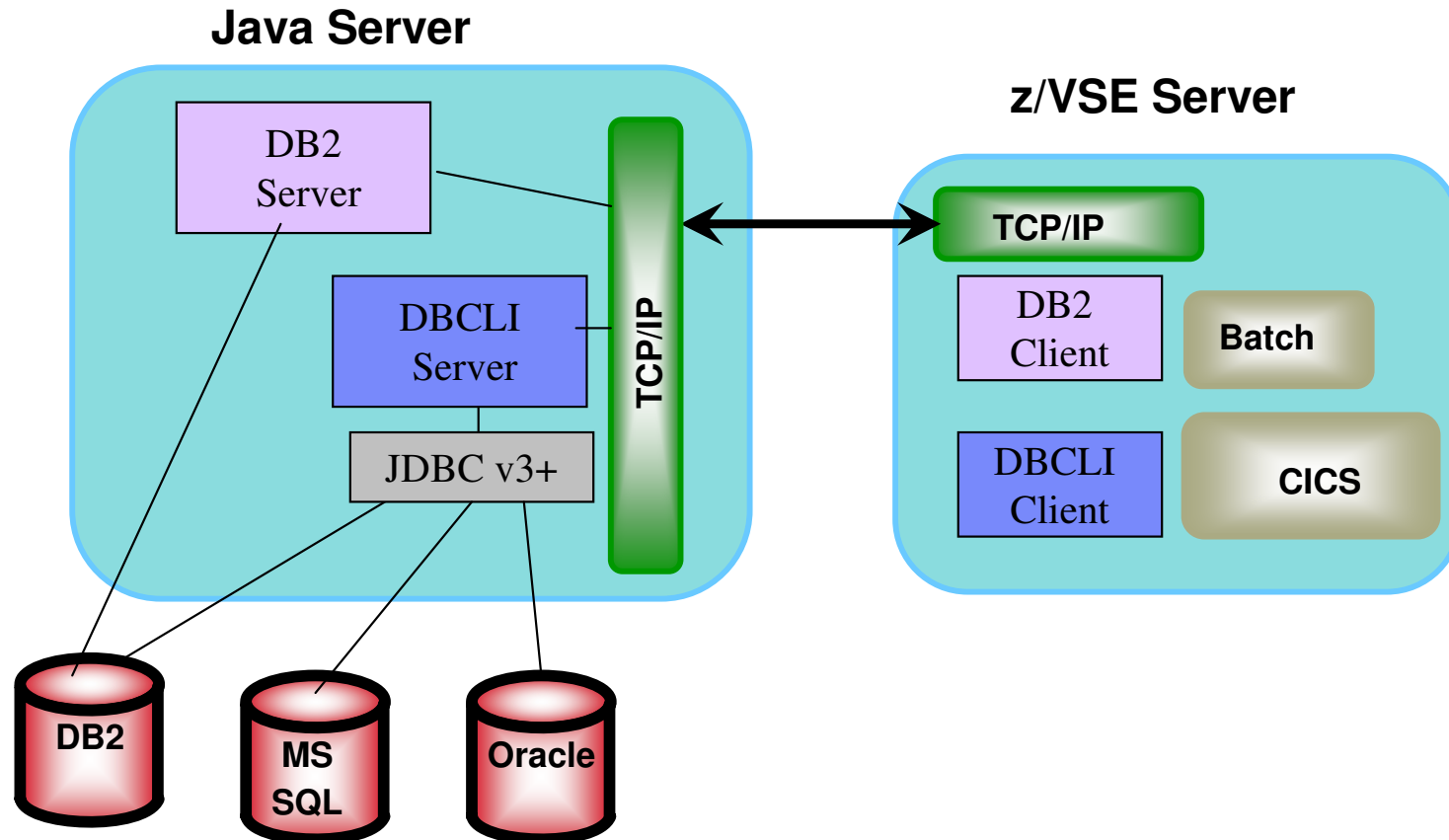
Consolidate, Integrate, Evaluate - use DB2 LUW and VSAM Redirector



Applications on z/VSE access 'any' remote relational databases

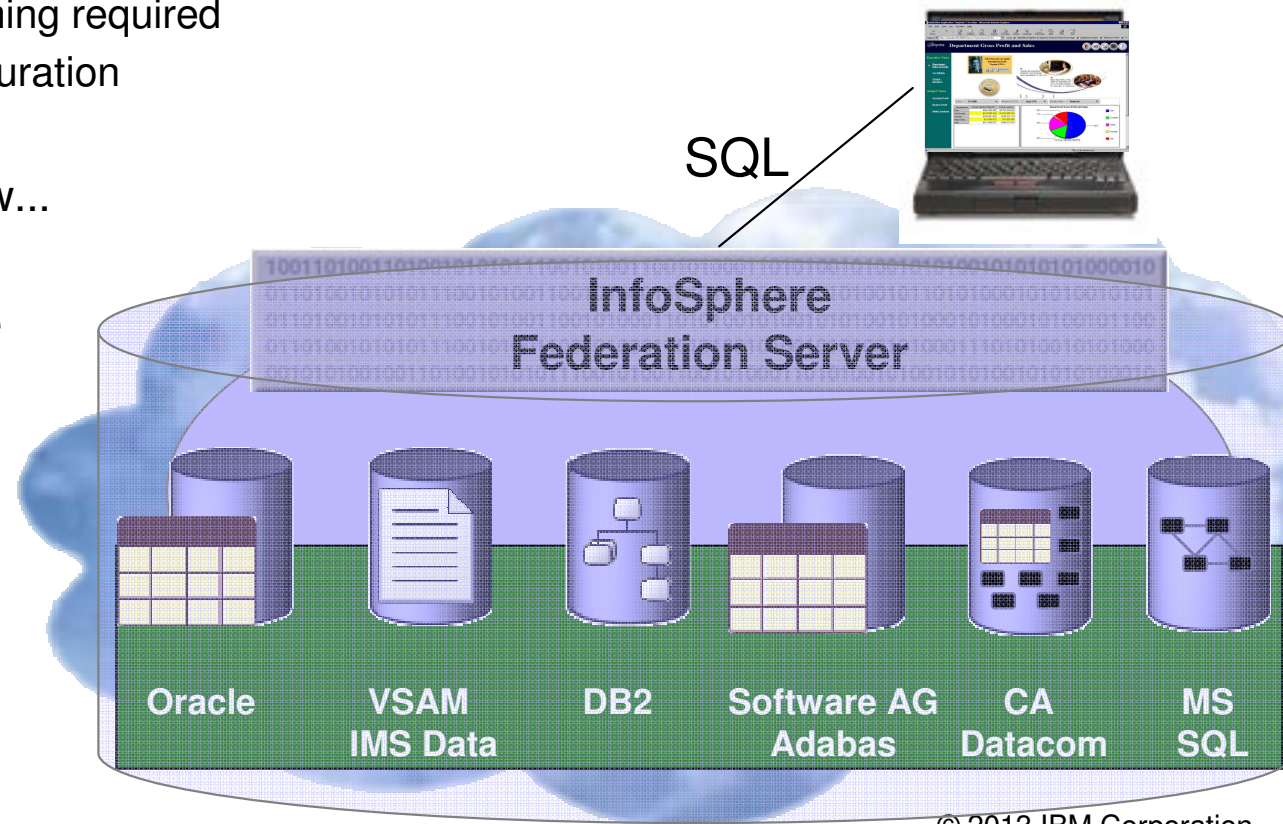


- ▶ Real time access to Relational databases
 - ▶ two different ways from batch and CICS
 - ▶ Access based on z/VSE DBCLI interface **AND / OR** DB2 Client



InfoSphere Federation Server on Linux on System z

- Integrating at the data layer – Federation of data
 - Read from and write to federated mainframe data sources using SQL
 - Standards-based access via JDBC, ODBC, or Call Level Interface
 - Including for mainframe VSAM data and flat files
 - Multithreaded with native drivers for scalable performance
 - Metadata-driven means...
 - No mainframe programming required
 - Fast installation & configuration
 - Ease of maintenance
 - Works with existing and new...
 - Mainframe infrastructure
 - Application infrastructure
 - Toolsets



Conclusion: Populate a Data Warehouse with VSAM data

- Define the VSAM files and VSAM field structures you need
- Real time synchronization is recommended for CICS workload
- Mass data change in VSAM should use capture/apply function
 - Capture VSAM changes in a delta file
 - Apply the changes through the schedule of DeltaLoader or MQ Series mechanism
- Data Warehouse changes could be replicated back into VSAM
 - using triggers in the database for corresponding columns

- z/VSE
- About z/VSE
- How to buy
- News & announcements
- Events
- Solutions**
- Products & components
- Documentation
- Service & support
- Downloads
- Education
- Partners
- FAQ
- Contact z/VSE

- Related links
- Linux on IBM System z
 - z/OS
 - z/VM
 - IBM Storage



IBM Systems > Mainframe servers > Operating systems >

z/VSE

z/VSE is designed to help provide robust, cost-effective solutions for customers with a wide range of capacity needs, in most industries, worldwide. z/VSE is built on a heritage of ongoing refinement and innovation that spans four decades. It brings the value of innovative IBM System z and IBM System Storage technology to z/VSE clients.

- Learn more
- [About z/VSE](#)
 - [News](#)
 - [z/VSE Status](#)
 - [History of z/VSE](#)

z/VSE V5.1.1 is available

Built on a heritage of ongoing refinement and innovation that spans more than four decades




IBM z/VSE V5.1 - Additional enhancements

In addition to function delivered at general availability of IBM z/VSE V5.1, you get supplemental enhancements:

- **Support for IBM CICS Explorer - "The new face of CICS Transaction Server for VSE/ESA V1.1"**
CICS Explorer V1.1 capabilities can now be used with CICS TS. The CICS Explorer is the new systems management framework for CICS TS. It provides read-only capabilities to display CICS resources. Please see the [CICS web page](#) for additional information and updates.
- **The Fast Path to Linux on System z function (Linux Fast Path) in a logical partition (LPAR) environment**
Linux Fast Path was introduced with z/VSE V4.3 for use in a z/VM guest environment. Now LPAR support is added, which is intended to extend the connectivity options for z/VSE clients. Linux Fast Path in an LPAR environment requires [IBM zEnterprise](#) technology with the HyperSockets Completion Queue function.
- **64-bit Input/Output (I/O) processing for applications**
64-bit virtual addressing for applications was introduced at general availability of z/VSE V5.1. z/VSE V5.1 enhancements add 64-bit I/O processing for applications. With 64-bit I/O processing, clients have the flexibility to also use 64-bit virtual storage for I/O buffers and thus benefit from increased processor storage available with the latest IBM System z servers.
- **A z/VSE database connector**

We're here to help



Easy ways to get the answers you need.

E-mail us

Stay informed

Get the latest news about z/VSE through [Twitter](#)

Mark your calendar



WAVV 2013
April 7-10, 2013
Covington, KY, USA

→ [Enroll now!](#)

→ [Other z/VSE events](#)

Announcing

