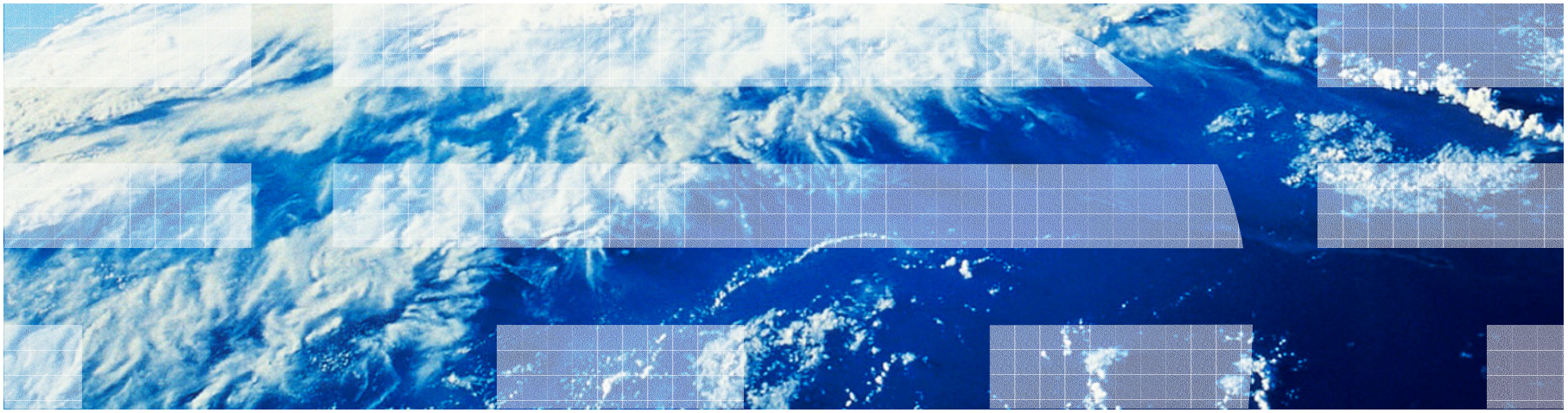
Synchronizing your VSAM Data with a Database Environment

Ingo Franzki & Wilhelm Mild, IBM



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market.

Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

*, AS/400®, e business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

FTP data transfer of VSE/VSAM data today

(1) disadvantages of FTP

a) NO real-time access/synchronization

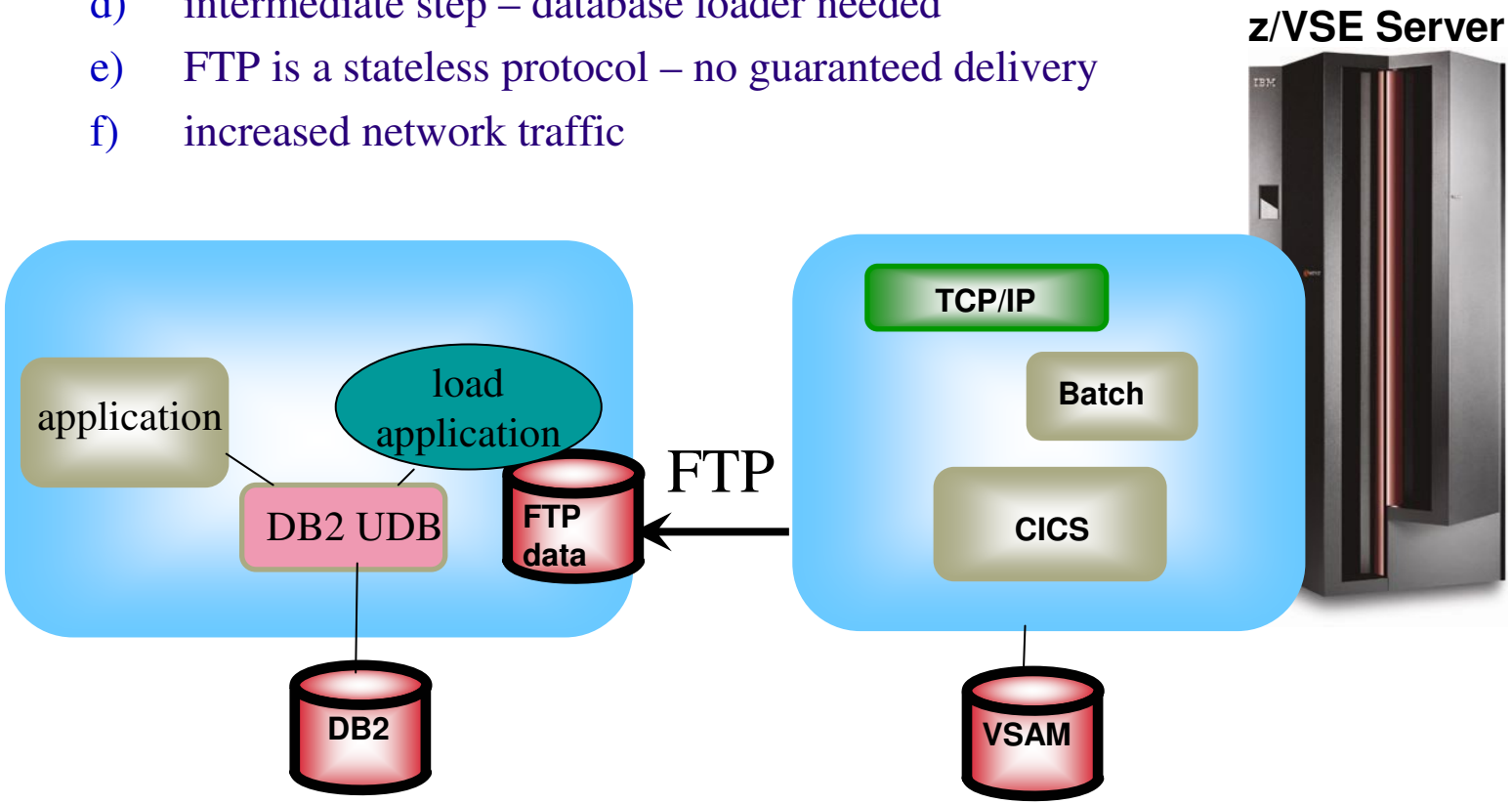
b) data immediate out of sync, till next FTP

c) always the whole file is transferred

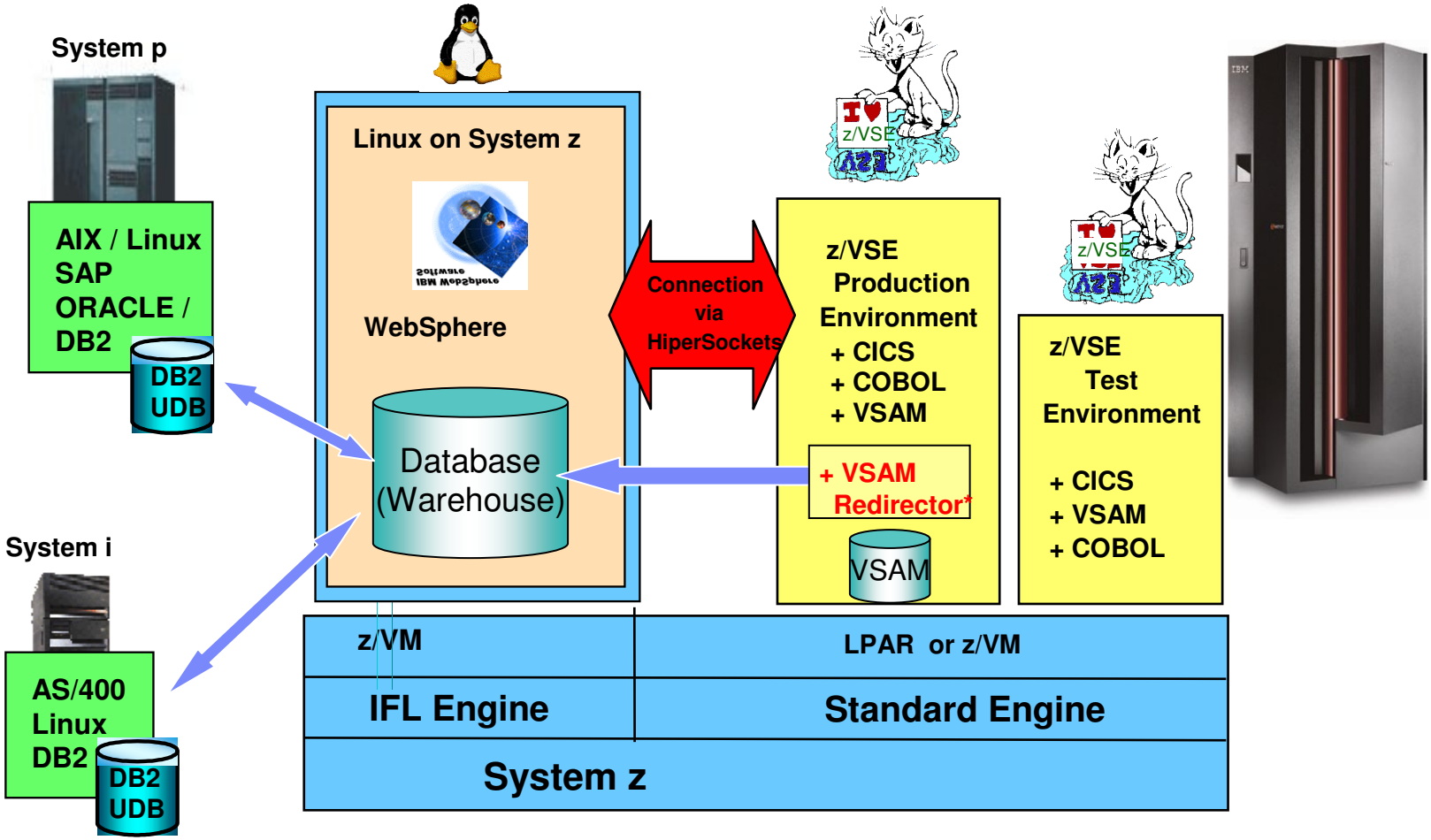
d) intermediate step – database loader needed

e) FTP is a stateless protocol – no guaranteed delivery

f) increased network traffic

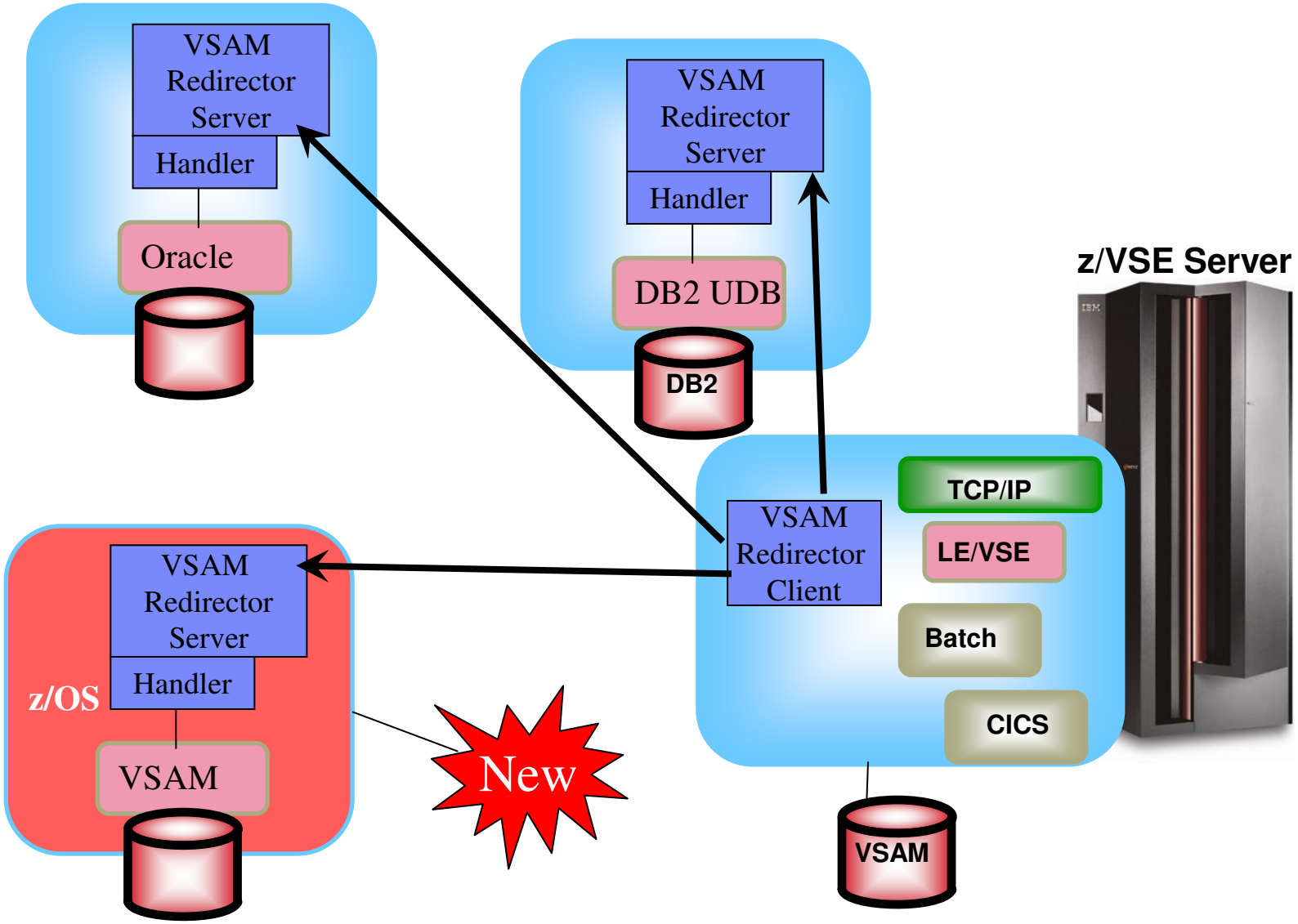


VSAM Programs with DB2 on Linux on System z



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

VSE/VSAM applications, access remote relational databases



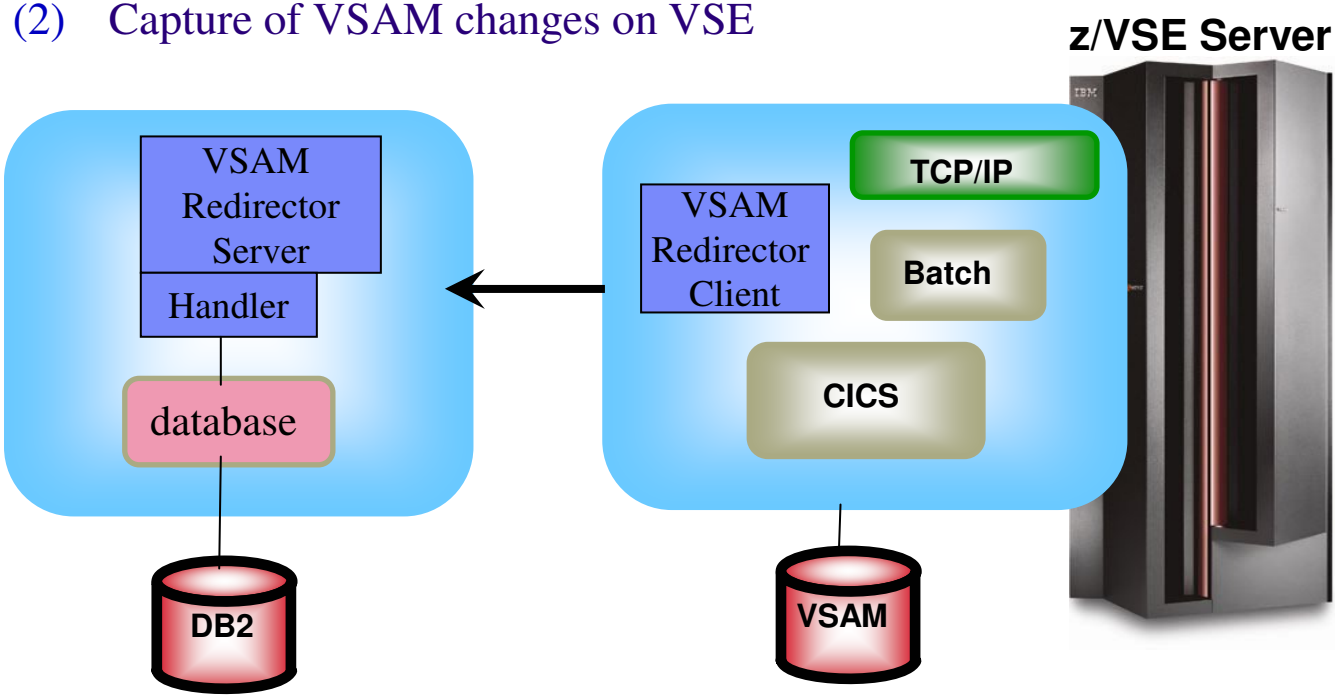
VSE/VSAM applications (without any change), access remote relational databases

(1) Real time access VSAM to DB2

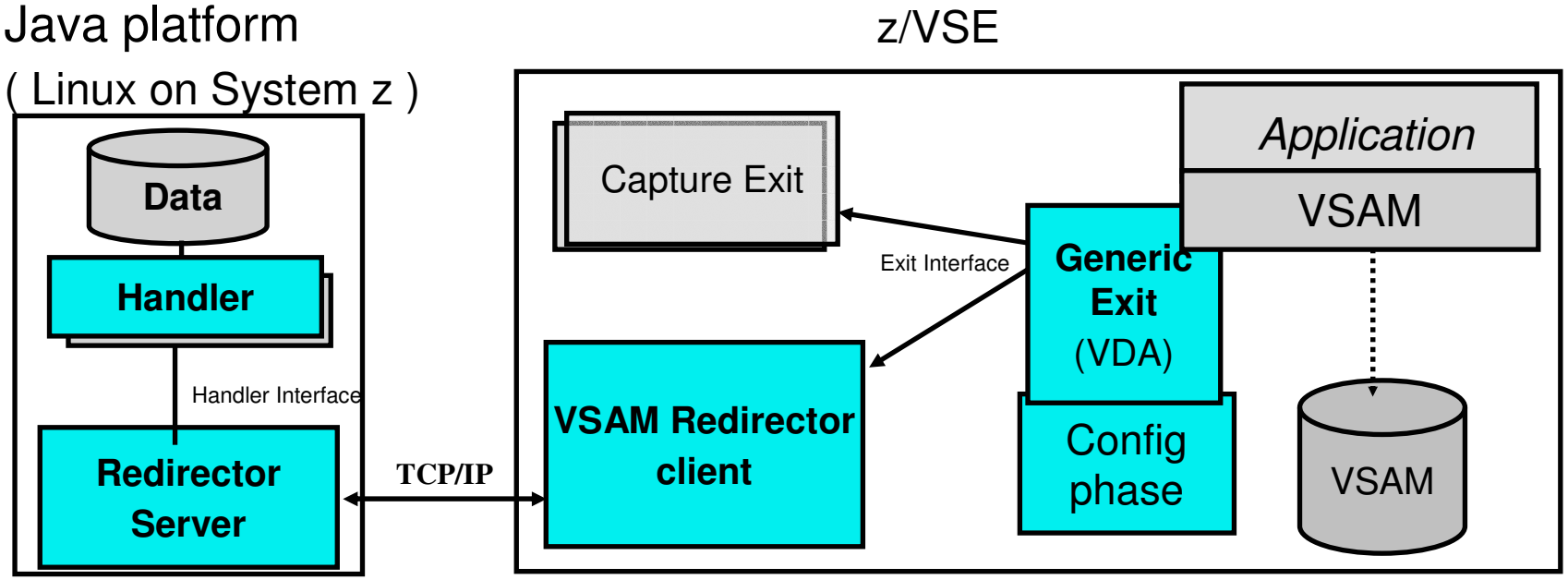
a) synchronization (two phase commit of VSAM and DB2)

b) Real time push of VSAM data to DB2

(2) Capture of VSAM changes on VSE



VSE/VSAM Redirector - functional view



- **Redirector Components:**

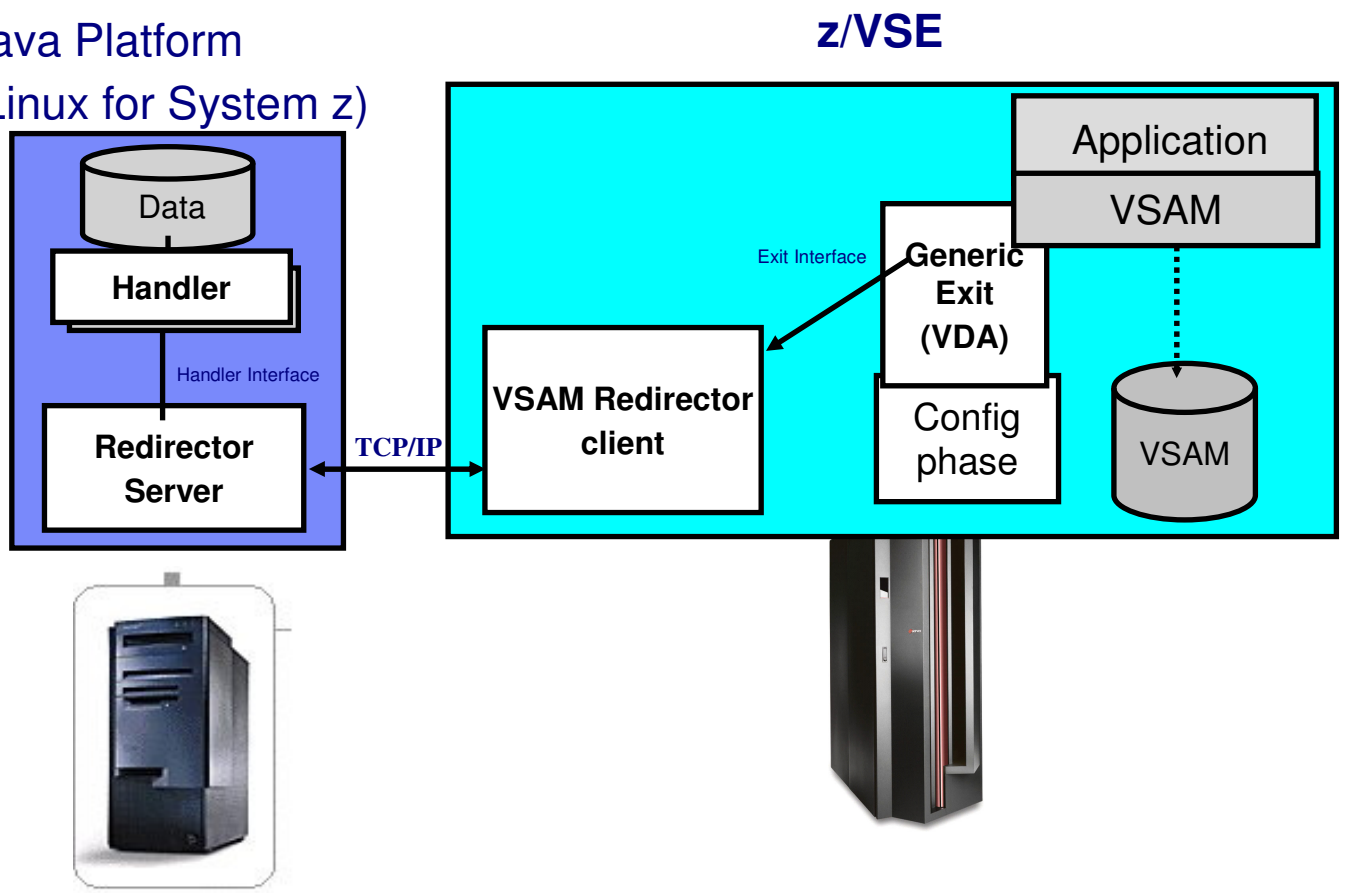
- **Generic Exit is based on VSAM Data Access Exit (VDA)**
- **Config phase – contains the redirection properties**
- **Redirector client (SVA phase)**
- **Redirector server – manages the connections (Java component)**
- **Handler – takes care of data processing (Java component)**

(1) Remote processing

a) Remote only processing – NO VSAM access anymore

- Requests for redirected VSAM files will be handled by VSAM Redirector client and send to the remote system.
- OWNER = REDIRECTOR

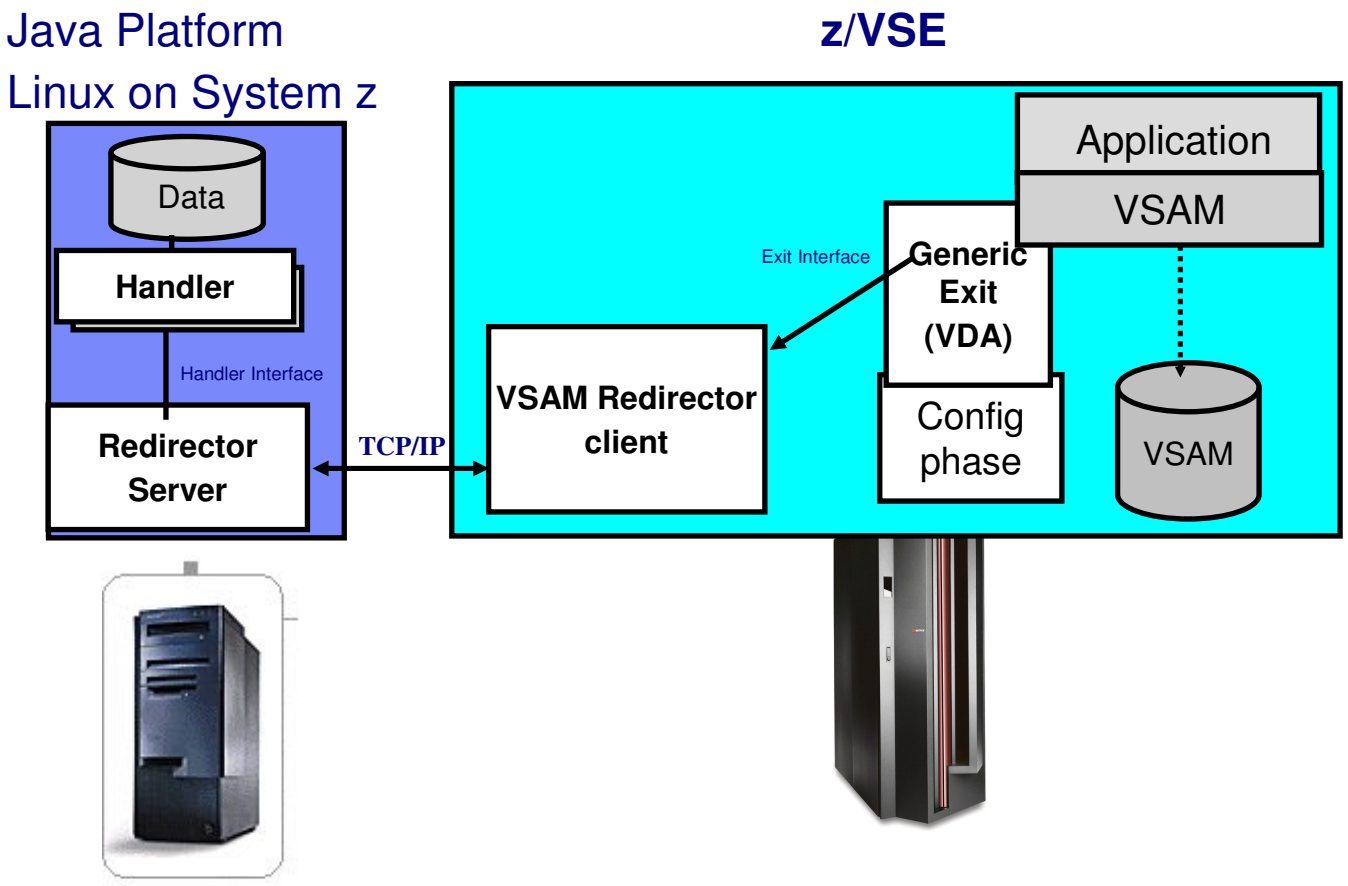
- Dataflow for a read/write request:
  - Generic exit is involved
  - VSAM Redirector Client is called
  - Redirector client sends request to Redirector Server
  - Handler processes data
  - Return Code if any is translated to VSAM error
- All reads and writes are done from/to remote



No changes required in applications (CICS, batch).

(1) Remote processing

a) Synchronization of VSAM with a database



No changes required in applications (CICS, batch).

(1) Remote processing

a) Synchronization of VSAM with a database

- Requests for redirected VSAM files will be handled by VSAM Redirector client and send to the remote system.
- OWNER = VSAM

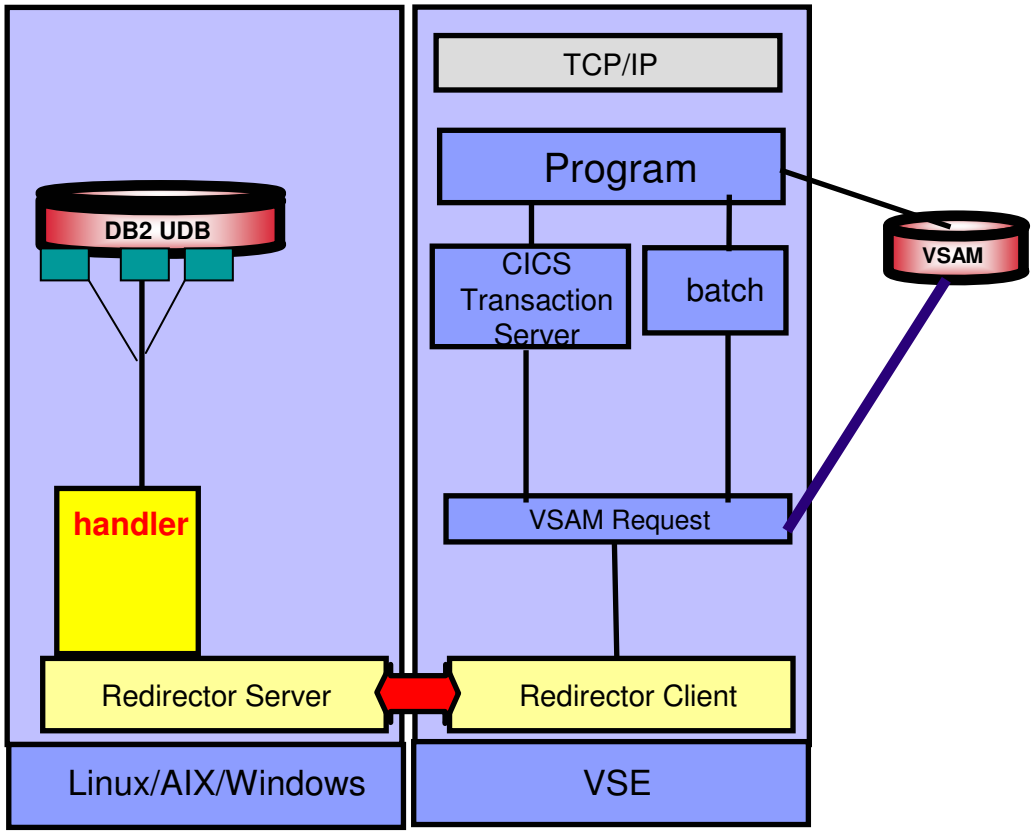
- Dataflow for a read/write request:
  - READs will be performed from VSAM only
  - Generic exit calls VSAM Redirector Client
  - Redirector client sends request to Redirector Server
  - (a) Handler processes the request in the database
  - Return Code is send back to Redirector client
    - In case of an error it will be translated to a VSAM error
  - (b) If return code is favorable the request is processed in VSAM
  - (c-1) If the VSAM request is processed correct – the database will get a COMMIT request through the Redirector Client-Server-Handler
  - (c-2) If the VSAM request ends in error – the database will get a ROLLBACK request

- This is the two phase commit mechanism to keep VSAM and DB2 in sync

Data synchronization – Normalization

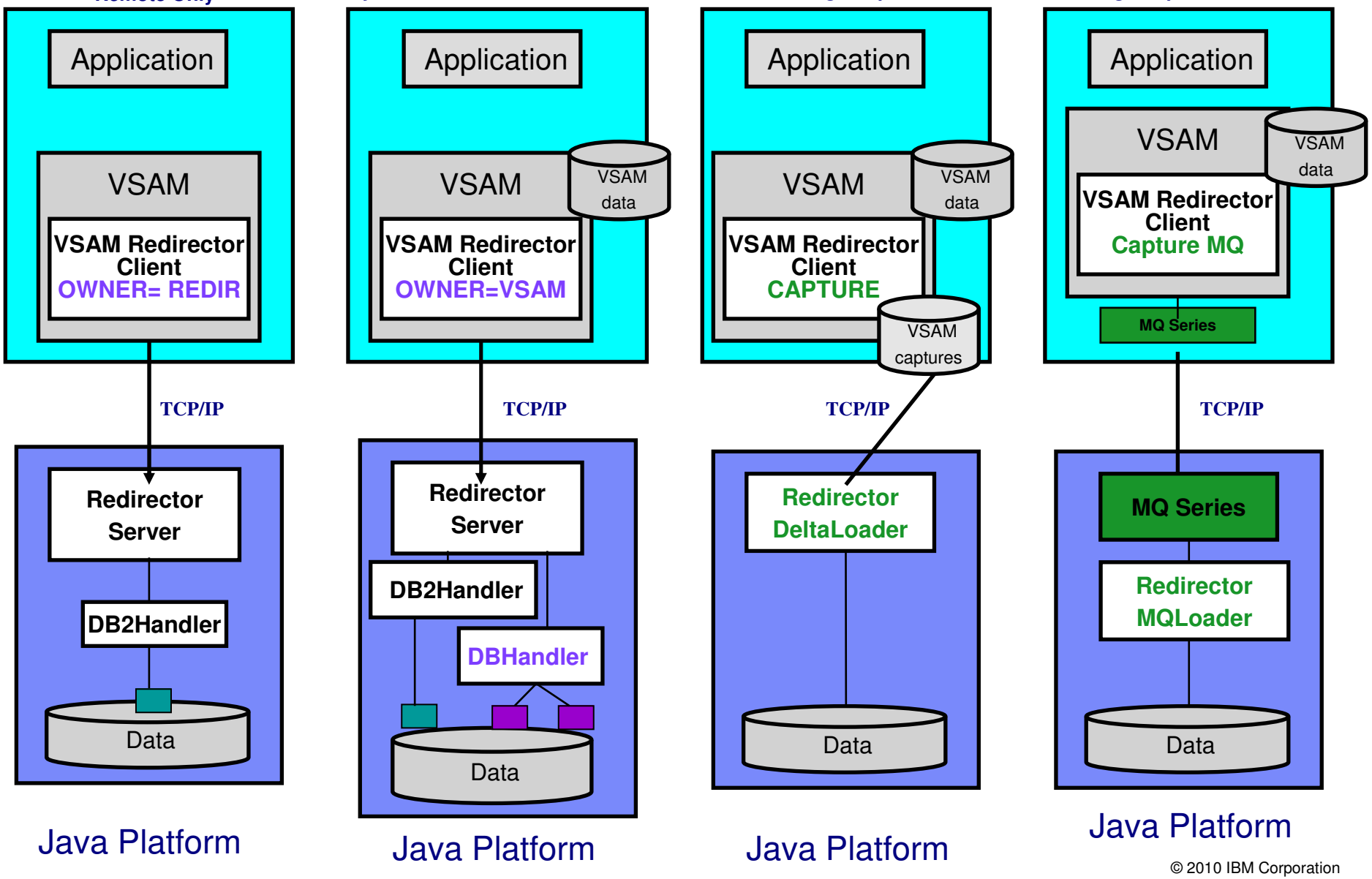
VSE/VSAM Redirector can store VSAM data normalized

- No changes to the existing VSE applications
- **The new Redirector Handler in z/VSE 4.1 can store 'VSAM' data in multiple DB Tables .**
- **Redir Loader utility provided for initial data transfer**



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

Redirector Solution Overview



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 new data formats (Packed, Zoned, Datum, ...)

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

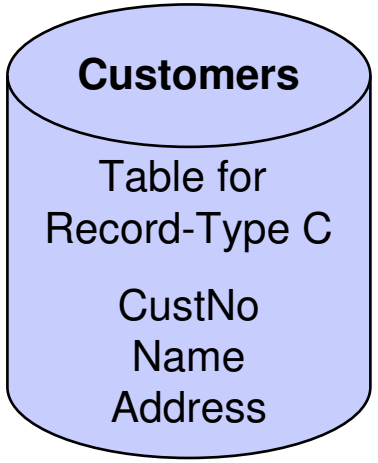
VSAM Redirector - Normalization – data types

- 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:

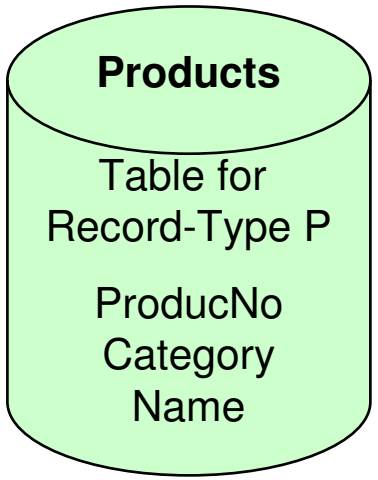
```
01 RECORD-3
  03 RECORD-TYPE           PIC X(1)
  03 RECORD-FORMAT-C
    05 CUSTOMER-NO                 PIC X(7)
    05 CUSTOMER-NAME       PIC X(25)
    05 CUSTOMER-ADDRESS    PIC X(45)
  03 RECORD-FORMAT-P REDEFINES RECORD-FORMAT-C
    05 PRODUCT-NO                  PIC X(7)
    05 PRODUCT-CATEGORY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

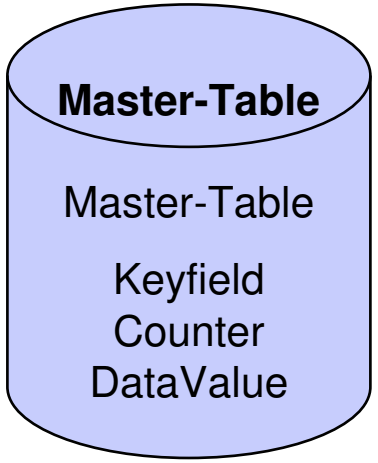
The association takes place at runtime for each individual record.

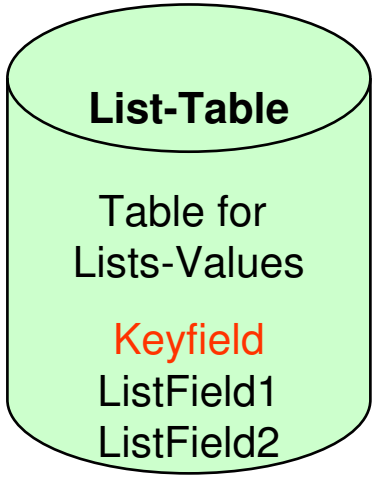


VSAM Redirector - Normalization – Lists

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	Val1	2	Val2	3	Val3

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	Val1
Data	KEY12345	2	Val2
Data	KEY12345	3	Val3

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

Database table VARIABLE_LIST

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

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

Field	KEYFIELD	LISTFIELD-A	LISTFIELD-B
Data	KEY12345	1	Val1
Data	KEY12345	2	Val2
Data	KEY98765	9	Val9
Data	KEY98765	8	Val8
Data	KEY98765	7	Val7
Data	KEY98765	6	Val6

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

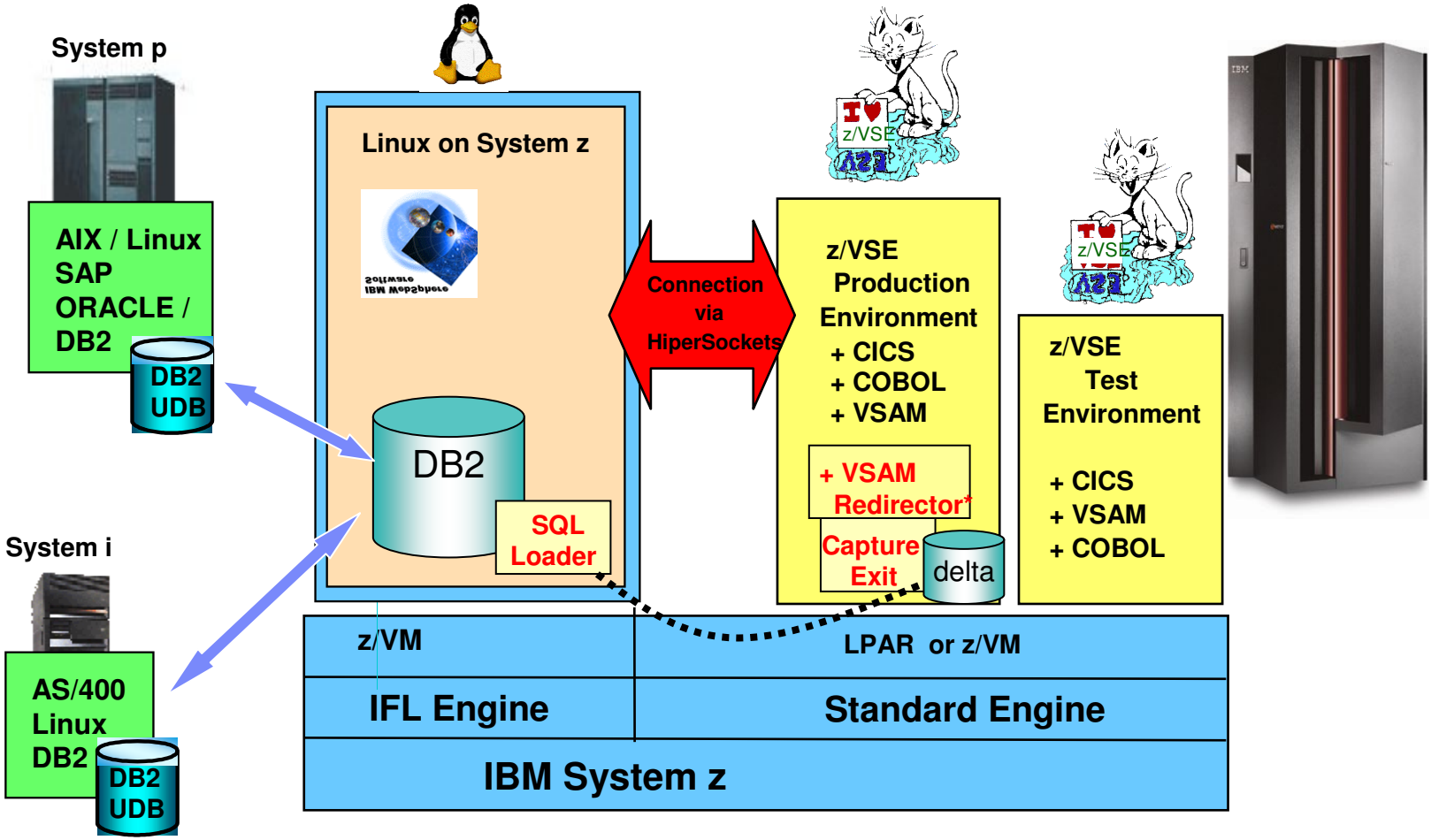
Database table RECORD_FORMAT_C

- CUSTOMER_NO
- CUSTOMER_NAME
- CUSTOMER_ADDRESS

Database table RECORD_FORMAT_P

- PRODUCT_NO
- PRODUCT_CATEGORY
- PRODUCT_NAME

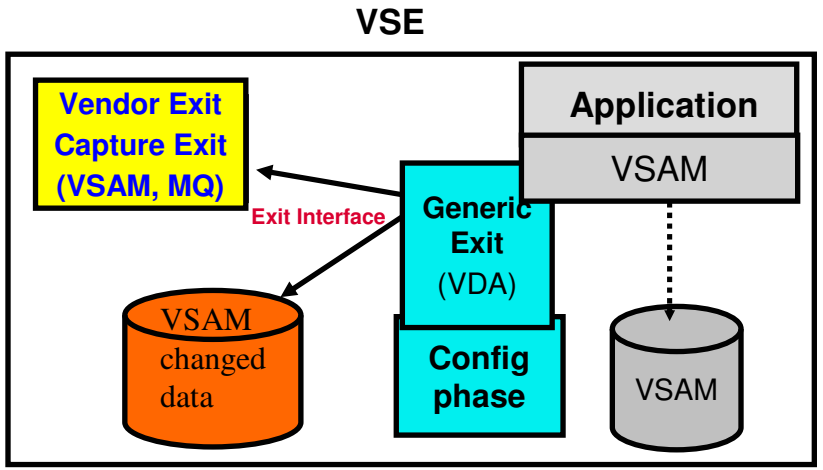
VSAM Programs with DB2 UDB on Linux on System z



(*) 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

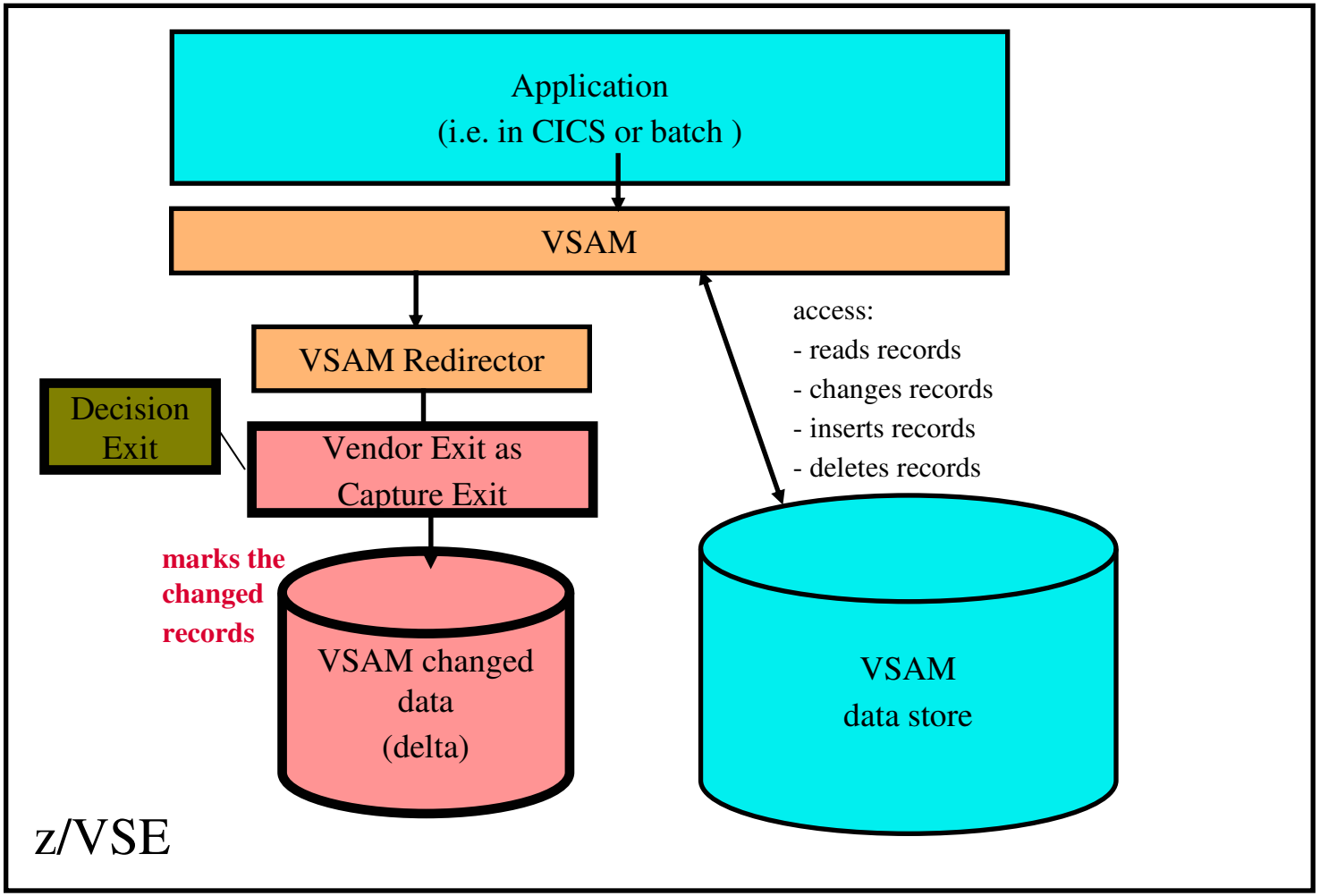


CAPTURE – wit Decision Exit as filter

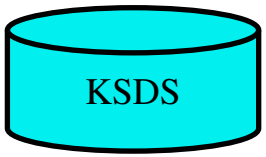
- **Vendor Exit**
  - **user (vendor) written phase for data collection/transformation**
  - **has to comply with the documented Exit Interface**
- **Capture Exit**
  - **an exit delivered by IBM for capturing changed VSAM data**
  - **an exit delivered by IBM for generating MQ messages**

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

Redirector Capture
Architectural View



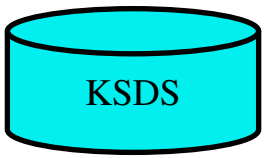
Journaling or cumulative



Delta File

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

Delta File

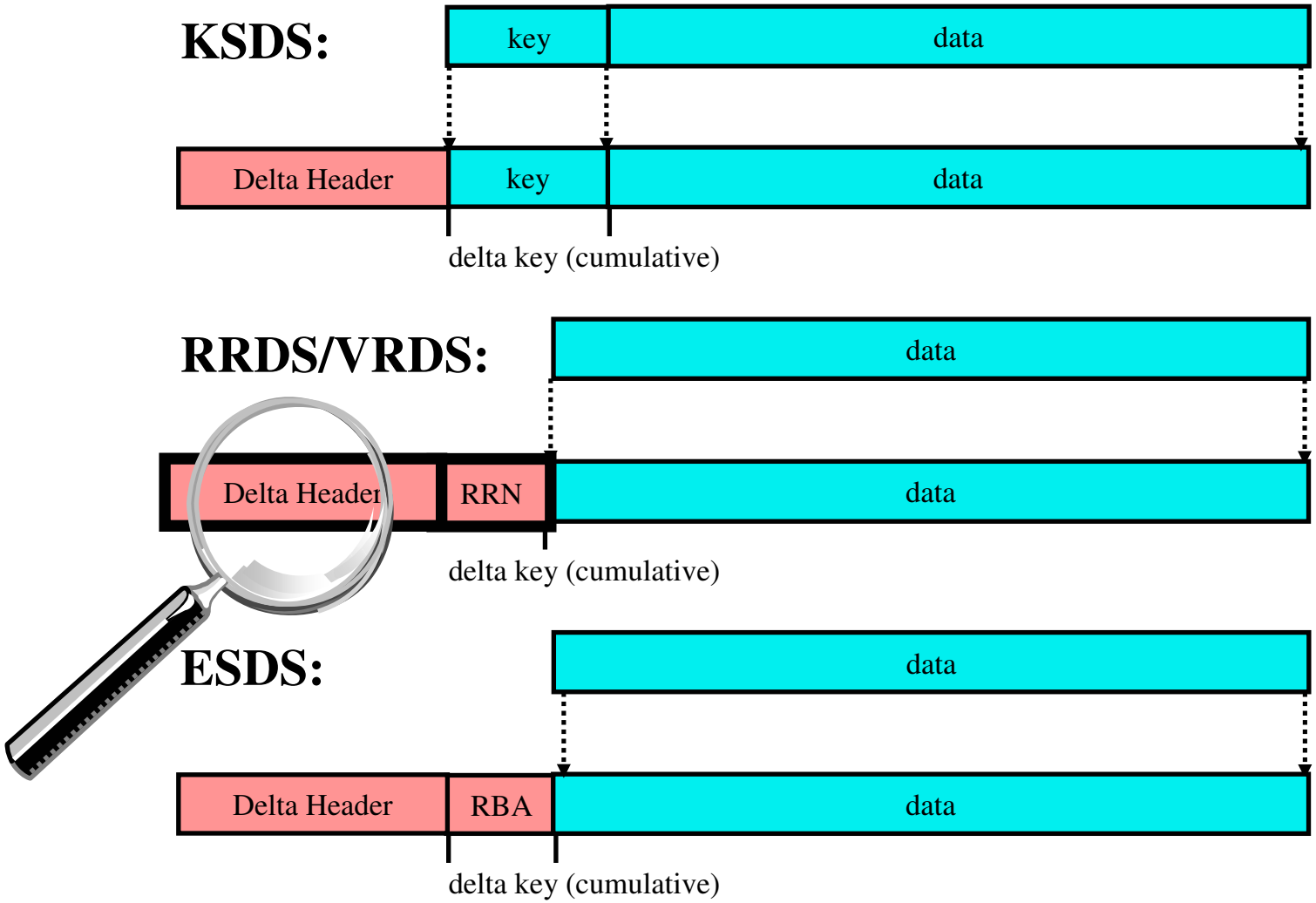


key

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



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)

Solutions

http://www.ibm.com/systems/z/os/zvse/

Additional Information

- z/VSE Home Page
  http://www.ibm.com/zvse

- z/VSE solutions
  http://www-1.ibm.com/systems/z/os/zvse/solutions

- e-business Connectors User's Guide SC33-6719
  http://www-1.ibm.com/systems/z/os/zvse/documentation/#conn

Redbooks

- e-business Solutions for VSE/ESA SG24-5662

- e-business Connectivity for VSE/ESA SG24-5950

- CICS Transaction Server for VSE/ESA
  CICS Web Support *SG24-5997-00*

- WebSphere Connectivity Handbook SG24-7042

We appreciate your comments at : zvse@de.ibm.com