z/VSE Connectors
Best practices and use cases
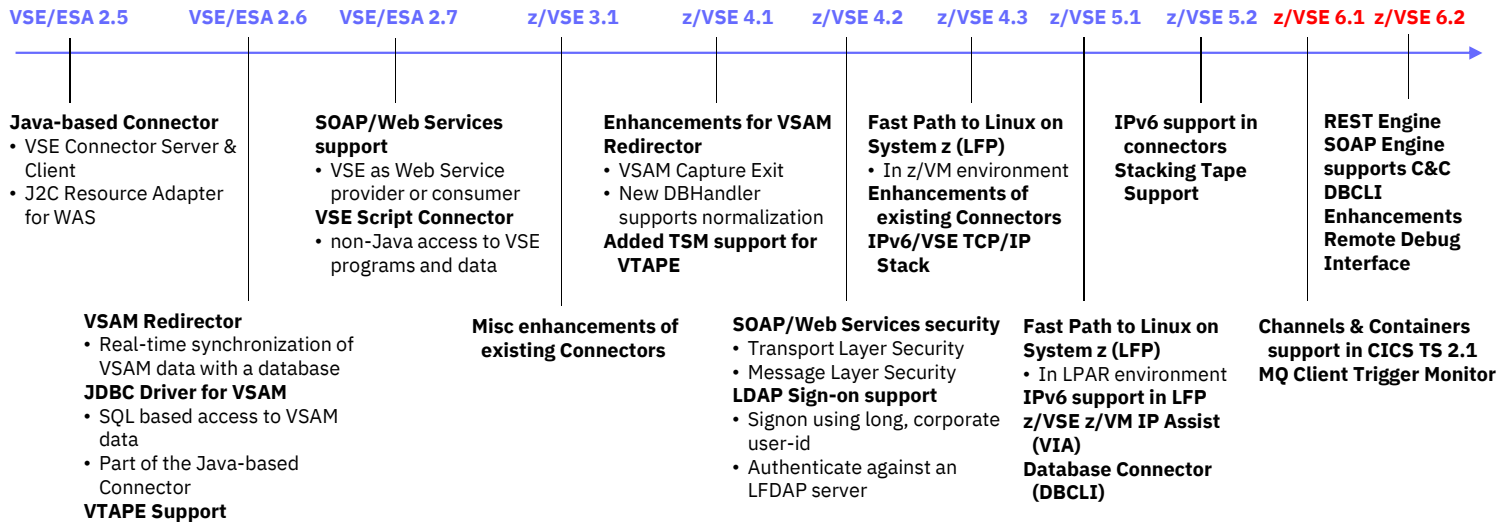
Ingo Franzki
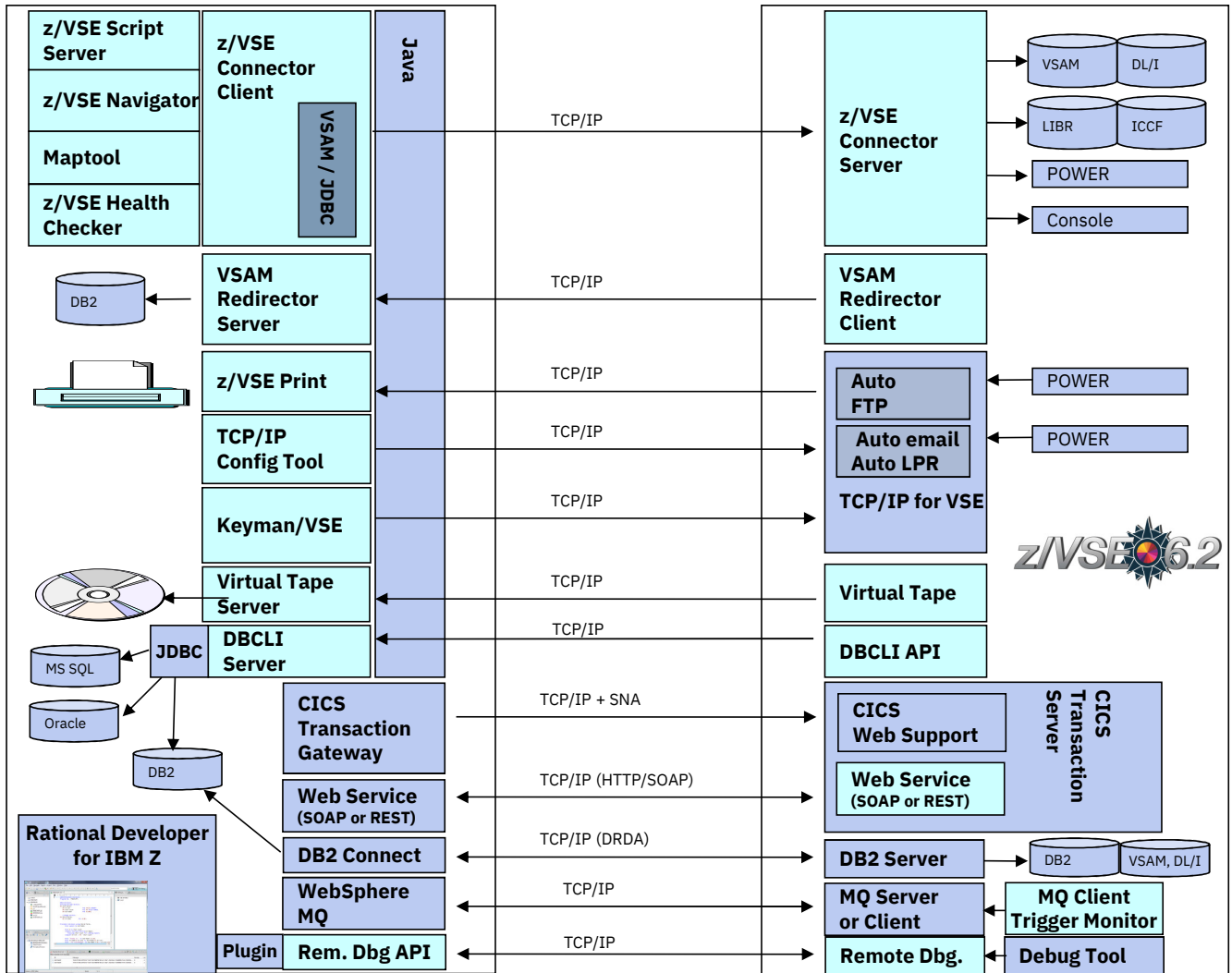**ifranzki@de.ibm.com**

z/VSE Connectors - Introduction

— **The z/VSE Connectors started as a single function in VSE/ESA 2.5**

— **Since then, it evolved over time**

— **Its now more a technology than just a function**
  - consisting of many different features and functions
  - supporting various connector solutions

Continuously supporting
the z/VSE PIE Strategy
Protect
Integrate
Extend

VSE/ESA 2.5	VSE/ESA 2.6	VSE/ESA 2.7	z/VSE 3.1	z/VSE 4.1	z/VSE 4.2	z/VSE 4.3	z/VSE 5.1	z/VSE 5.2	z/VSE 6.1	z/VSE 6.2

Java-based Connector
- VSE Connector Server & Client
- J2C Resource Adapter for WAS

SOAP/Web Services support
- VSE as Web Service provider or consumer

VSE Script Connector
- non-Java access to VSE programs and data

Enhancements for VSAM Redirector
- VSAM Capture Exit
- New DBHandler supports normalization

Added TSM support for VTAPE

Fast Path to Linux on System z (LFP)
- In z/VM environment

Enhancements of existing Connectors IPv6/VSE TCP/IP Stack

IPv6 support in connectors Stacking Tape Support

REST Engine SOAP Engine supports C&C DBCLI Enhancements Remote Debug Interface

VSAM Redirector
- Real-time synchronization of VSAM data with a database

JDBC Driver for VSAM
- SQL based access to VSAM data
- Part of the Java-based Connector

VTAPE Support

Misc enhancements of existing Connectors

SOAP/Web Services security
- Transport Layer Security
- Message Layer Security

LDAP Sign-on support
- Signon using long, corporate user-id
- Authenticate against an LFDAP server

Fast Path to Linux on System z (LFP)
- In LPAR environment

IPv6 support in LFP z/VSE z/VM IP Assist (VIA)

Database Connector (DBCLI)

Channels & Containers support in CICS TS 2.1
MQ Client Trigger Monitor

z/VSE Script Server

z/VSE Navigator

Maptool

z/VSE Health Checker

z/VSE Connector Client

VSAM / JDBC

Java

TCP/IP

z/VSE Connector Server

VSAM   DL/I

LIBR   ICCF

POWER

Console

DB2

VSAM Redirector Server

TCP/IP

VSAM Redirector Client

z/VSE Print

TCP/IP

Auto FTP

POWER

TCP/IP Config Tool

TCP/IP

Auto email Auto LPR

POWER

Keyman/VSE

TCP/IP

TCP/IP for VSE

z/VSE 6.2

Virtual Tape Server

TCP/IP

Virtual Tape

MS SQL

JDBC

DBCLI Server

TCP/IP

DBCLI API

Oracle

DB2

CICS Transaction Gateway

TCP/IP + SNA

CICS Web Support

CICS Transaction Server

Web Service (SOAP or REST)

TCP/IP (HTTP/SOAP)

Web Service (SOAP or REST)

Rational Developer for IBM Z

DB2 Connect

TCP/IP (DRDA)

DB2 Server

DB2   VSAM, DL/I

WebSphere MQ

TCP/IP

MQ Server or Client

MQ Client Trigger Monitor

Plugin   Rem. Dbg API

TCP/IP

Remote Dbg.

Debug Tool

z/VSE Connector & Tools - Downloads

— **Connector components**
  - Delivered as part of the z/VSE system as WBOOKs in PRD2.PROD
  - Also available on the z/VSE Homepage
    https://www.ibm.com/it-infrastructure/z/zvse-downloads

— **Many 'as-is' tools are provided for download**

— **Information about the Connector Components can be found here:**
    https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ws_connection_possibilities_use_.html

Scenario 1: Web- / Mobile-enabling of Applications

→ **Web-enable z/VSE Applications**

→ **Mobile-enable z/VSE Applications**

→ **Provide RESTful APIs for z/VSE Applications (microservices)**

→ **Modernize User Interface for applications**

Frontend

Middle-Tier (e.g. Linux on IBM Z)

Backend (e.g. z/VSE)

Extranets

Intranets

Internet

Edge servers
(Firewall, Load balancing, etc.)

Portal & Web Presentation Servers

Web Application Server

Mobile Application Server

Directory & Security Services

Connectors

Transaction Servers
(e.g. CICS TS)

Batch Processing
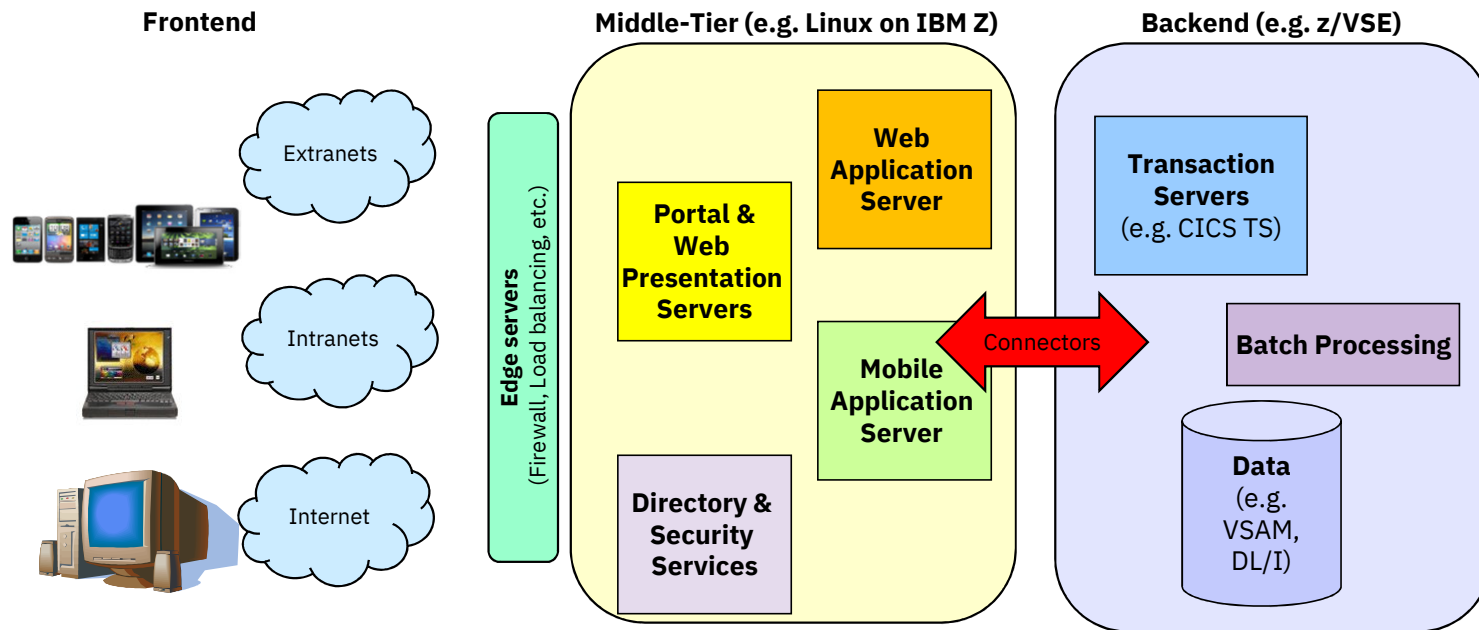
Data
(e.g. VSAM, DL/I)

Scenario 2: Central Database

→ **Use a central database for all business related applications**

→ **Allow z/VSE applications to work with central database**

→ **Add analytics and business intelligence**
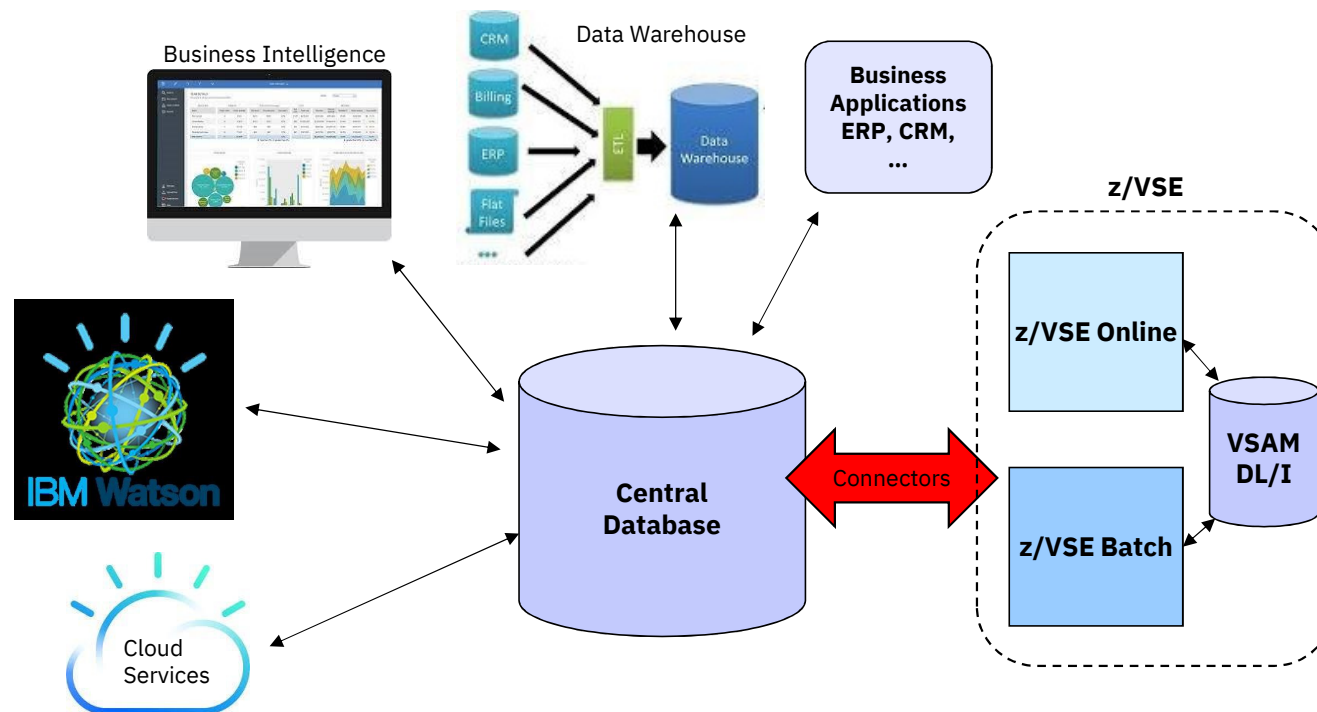
Scenario 3: Modern z/VSE Administration, Operation, Automation

→ **Use graphical Tools for z/VSE administration**

→ **Add cross-system automation**

CICS Explorer

VSE Navigator

z/VSE

| z/VSE Batch | Transaction Servers (e.g. CICS TS) |

VSE Health Checker

Automation cross system

| VSE Connector Server | TCP/IP |

Keyman/VSE

Data (e.g. VSAM, DL/I)
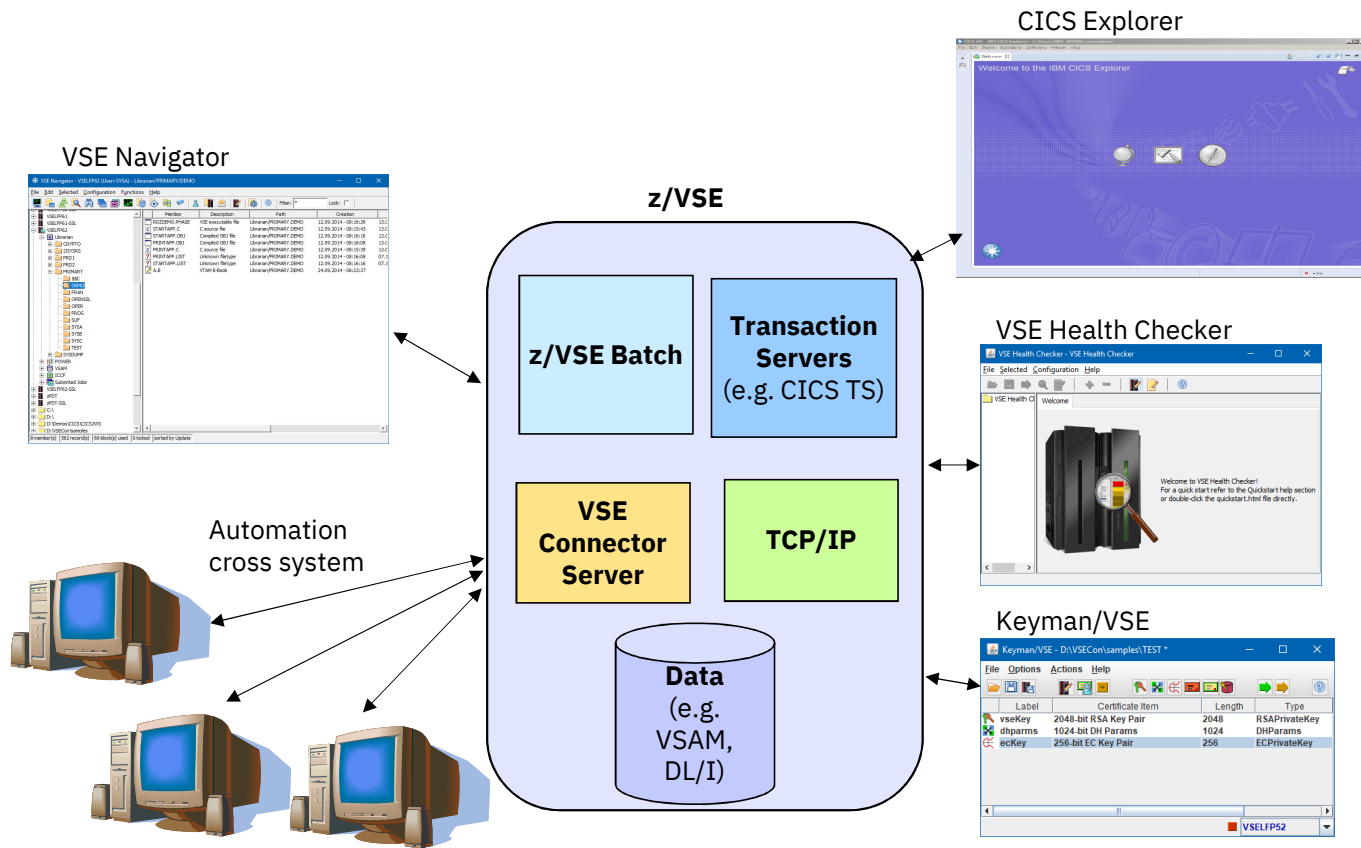
Java-based Connector

— **Remote access to z/VSE data and programs from a Java program**

- Real time access to VSAM, DL/1, LIBR, POWER, Console, Jobs, ...
- From standalone Java programs
- From web/mobile applications (servlets, JSPs, Mobile Apps etc.)
- Deployable as J2C Resource Adapter or JDBC Data Source into
  J2EE web applications servers, such as:
  - WebSphere Application Server
  - IBM Mobile First Server

Scenario 1: Web-Mobile-enablement

Scenario 3: Cross system Automation

VSAM Maptool

VSE Navigator   Java platform

Standalone Java application	z/VSE Connector Client	Java
Web application	WAS	
Mobile App	Mobile First	VSAM / JDBC

TCP/IP

z/VSE

z/VSE Connector Server   Plugin   Field mapping

VSAM   DL/I

LIBR   ICCF

POWER

Console
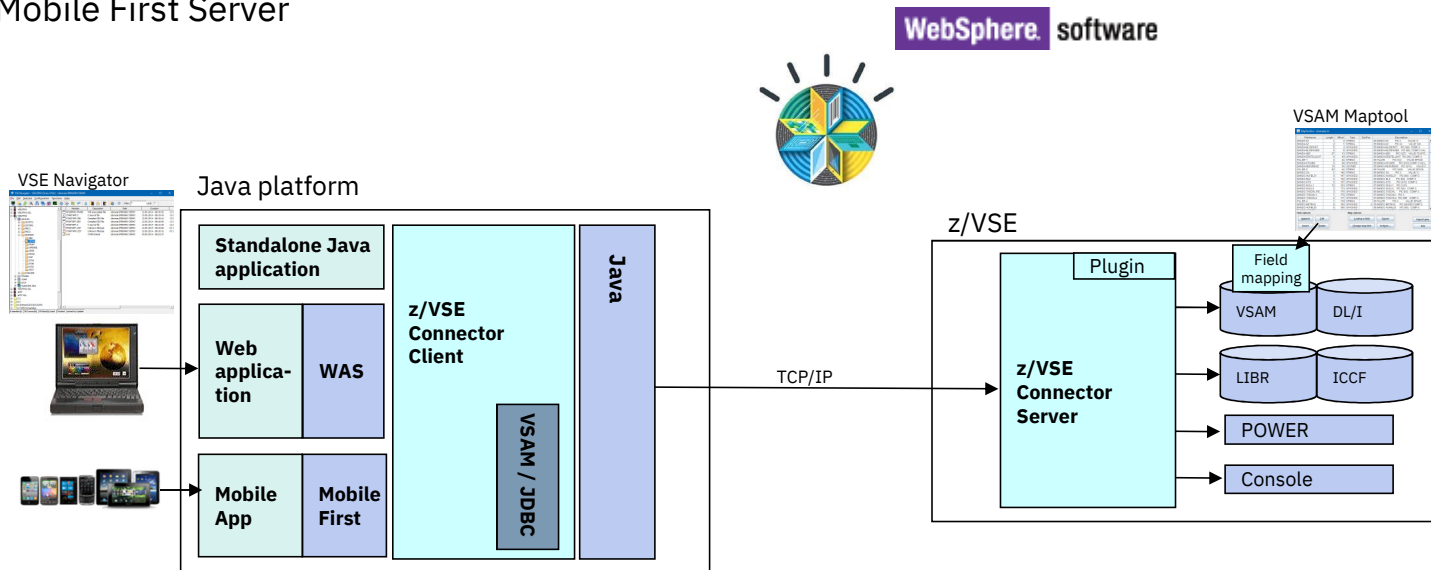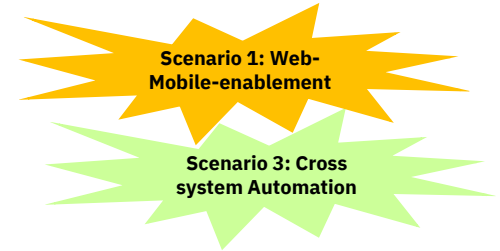
VSAM Redirector

— **Synchronization of VSAM data with a database**

— **Real-time:** VSAM Redirector Client/Server
  - **Data synchronization (OWNER=VSAM):**
    - any INSERT, UPDATE or DELETE request is immediately replicated into the database
    - Read requests go against the VSAM dataset on z/VSE
  - **Data migration (OWNER=REDIRECTOR):**
    - All VSAM requests are sent to the database.
    - No access to the VSAM dataset anymore (except OPEN / CLOSE).

— **Near real-time:** VSAM Capture Exit
  - Data changes are collected in a delta file or MQ queue
  - Delta file is downloaded and applied to database from time to time

Scenario 2: Central Database
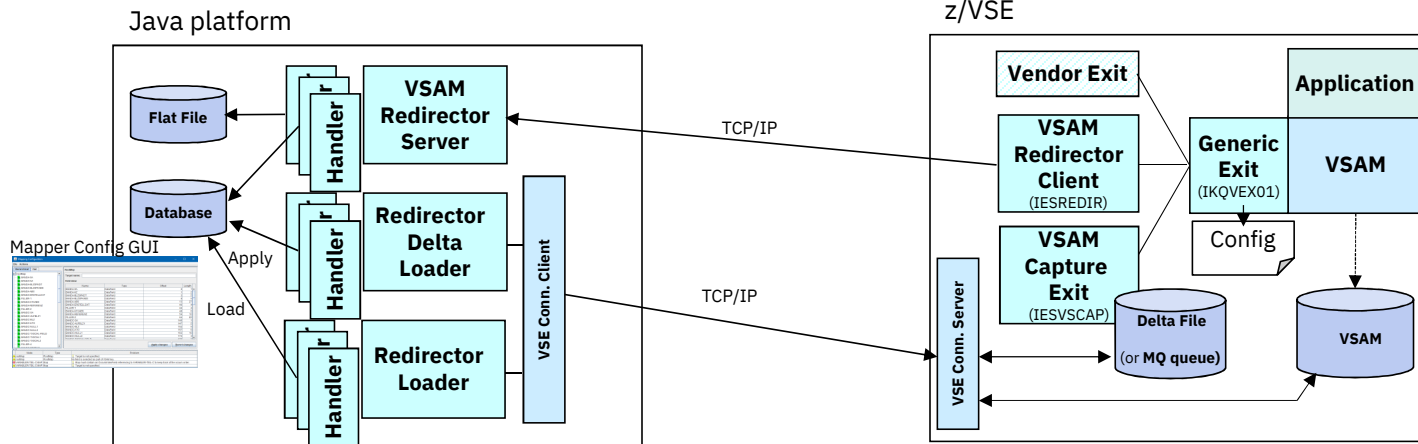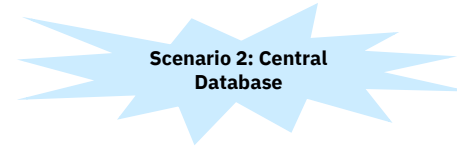
VSAM Redirector - modes of operation

Data Migration
— EXIT=IESREDIR, OWNER=REDIR

Application
VSAM
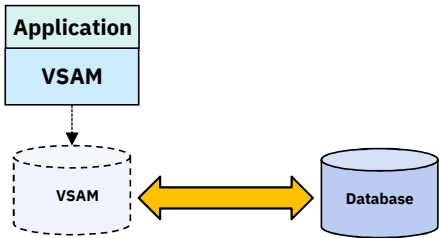
VSAM ⟷ Database

— Real time access to data in database

— All VSAM requests are redirected to the database

— High performance impact

Data Synchronization
— EXIT=IESREDIR, OWNER=VSAM

Application
VSAM
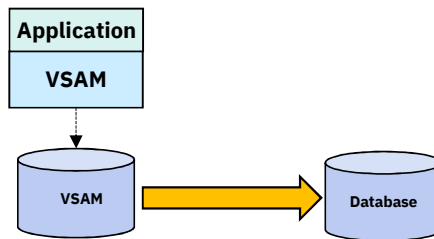
VSAM → Database

— Real time data replication (one way)

— Only updating VSAM requests (update, insert, delete) are redirected to database

— Medium performance impact

Capture & Apply
— EXIT=IESVSCAP

Application
VSAM

VSAM / Delta File (or MQ queue) → Database
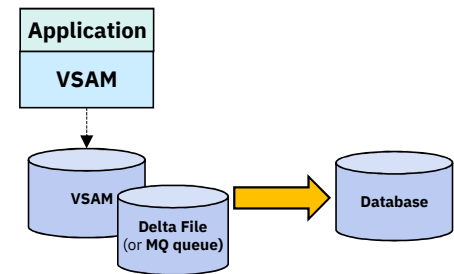
— Near real time data replication (every n minute, etc.)

— Changes to VSAM data are captured and collected, and then applied to the database asynchronously

— Low performance impact

z/VSE Database Call Level Interface (DBCLI)

Scenario 2: Central Database
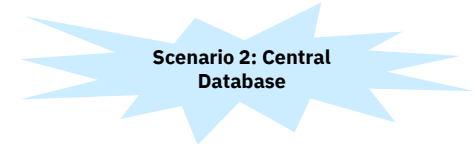
— **Allows z/VSE applications to access a relational database on any suitable database server**
  - IBM DB2, IBM Informix, Oracle, MS SQL Server, MySQL, etc.
    →The database product must provide a JDBC driver that supports JDBC V3.0 or later

— **Utilize advanced database functions and use SQL statements provided by modern database products**
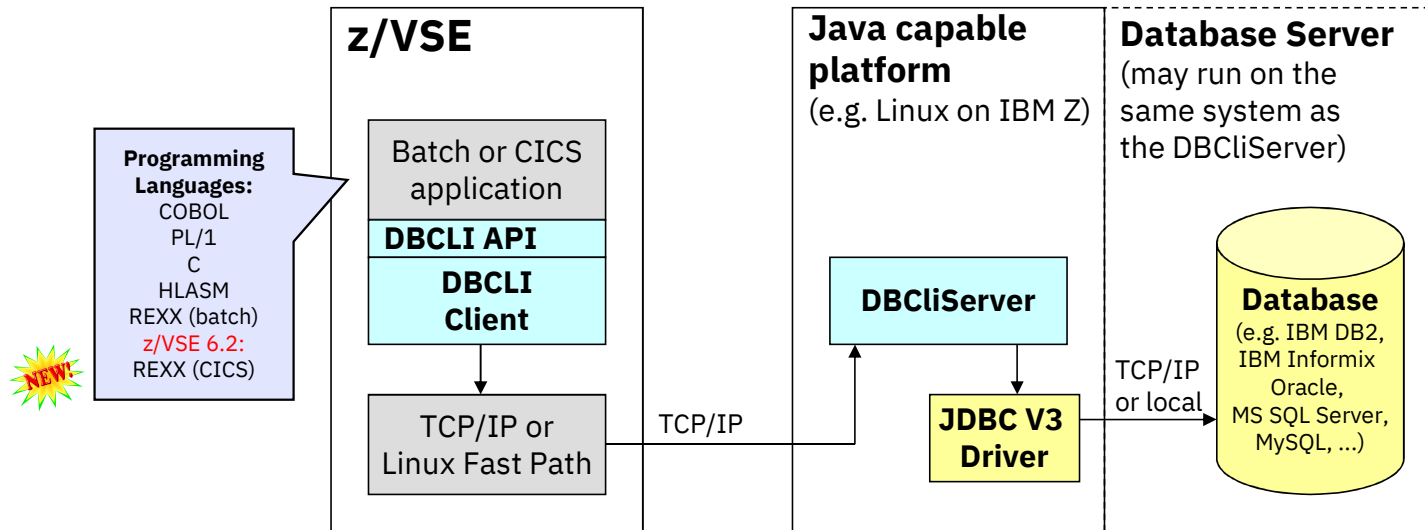
z/VSE applications accessing Databases

z/VSE

CICS or batch Application

DBCLI

DBCLI ✔ → MS SQL Server

DBCLI & DRDA ✔ → DB2 LUW

And others ...

with DB2 Federation

DBCLI ✔ → Oracle

DBCLI ✔ → MySQL

12
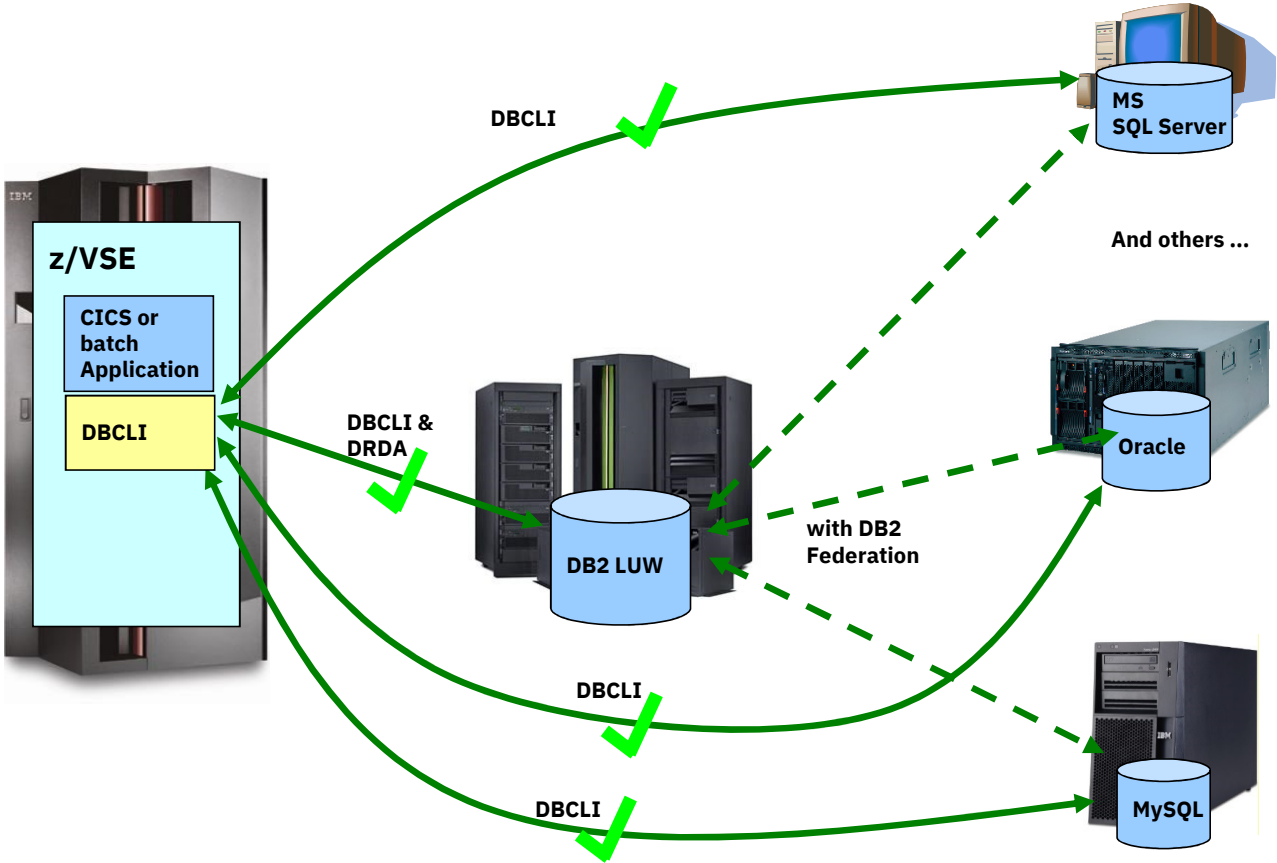
z/VSE 6.2: Interactive Query Tool for DBCLI

NEW!

z/VSE 6.2: Batch Query Tool for DBCLI

NEW!

```
* $$ JOB JNM=RUNDBCLI,DISP=D,CLASS=4
* $$ LST DISP=D,CLASS=Q,PRI=3
// JOB RUNDBCLI
// LIBDEF
*,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.TCPIPC)
// EXEC IESDBCLB,PARM='SYMBOLS=YES ECHO=ON'

CONNECT SERVER=my.database.server.com DBNAME=SAMPLE
        USER=db2user PASSWORD=password;

SELECT EMPNO,FIRSTNME,LASTNAME,SALARY,BONUS
       FROM EMPLOYEE;

DISCONNECT;
/*
/&
```

```
// JOB RUNDBCLI
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.TCPIPC)
// EXEC IESDBCLB,PARM='SYMBOLS=YES ECHO=ON'
1S54I  PHASE IESDBCLB IS TO BE FETCHED FROM PRD1.BASE
DBCLI BATCH QUERY TOOL

CONNECT SERVER= my.database.server.com DBNAME=SAMPLE
        USER=db2user PASSWORD=(PASSWORD SUPPRESSED);
INFO: CONNECTED TO SERVER 'my.database.server.com' DBNAME 'SAMPLE'.
INFO: DATABASE PRODUCT 'DB2/NT' VERSION 'SQL09075'
INFO: LAST RC=0

SELECT EMPNO,FIRSTNME,LASTNAME,SALARY,BONUS FROM EMPLOYEE;
INFO: STATEMENT HAS BEEN EXECUTED, IT PRODUCED THE FOLLOWING RESULT
SET:

EMPNO    FIRSTNME     LASTNAME         SALARY      BONUS
CHAR(6) VARCHAR(12)  VARCHAR(15)      DECIMAL(9,2) DECIMAL(9,2)
------- ------------ ---------------- ------------ ------------
000010  CHRISTINE    HAAS                152750.00     1000.00
000020  MICHAEL      THOMPSON             94250.00      800.00
000030  SALLY        KWAN                 98250.00      800.00
...

INFO: ROWCOUNT: 42
INFO: LAST RC=0

DISCONNECT;
INFO: DISCONNECT SUCCESSFULL.
INFO: LAST RC=0

1S55I  LAST RETURN CODE WAS 0000
EOJ RUN      MAX.RETURN CODE=0000
```

z/VSE Script Connector
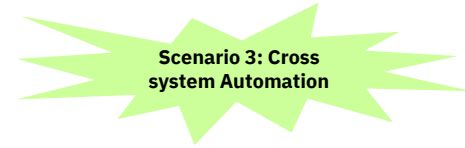
— **z/VSE Script Connector can be used to:**
- Access z/VSE resources from distributed non-Java applications
- Call/Execute processes on distributed systems from z/VSE applications or Jobs
- Cross-System automation

Scenario 3: Cross system Automation

z/VSE

VSE Script Client (Batch or CICS)

Distributed System (Java)

VSE Script Server

VSE Conn. Client

Script Repository

Script

External Process (running on distributed system)

TCPIP

call/execute

z/VSE

VSE Connector Server

LIBR   POWER   VSAM

TCPIP

MS Office OpenOffice (Non-Java App)

VSE Script Client (Windows DLL, .Net/C# Class)

TCPIP

execute script

VSE Script Client (Java)

CICS Connectivity

— **CICS Web Support is the base of CICS connectivity**
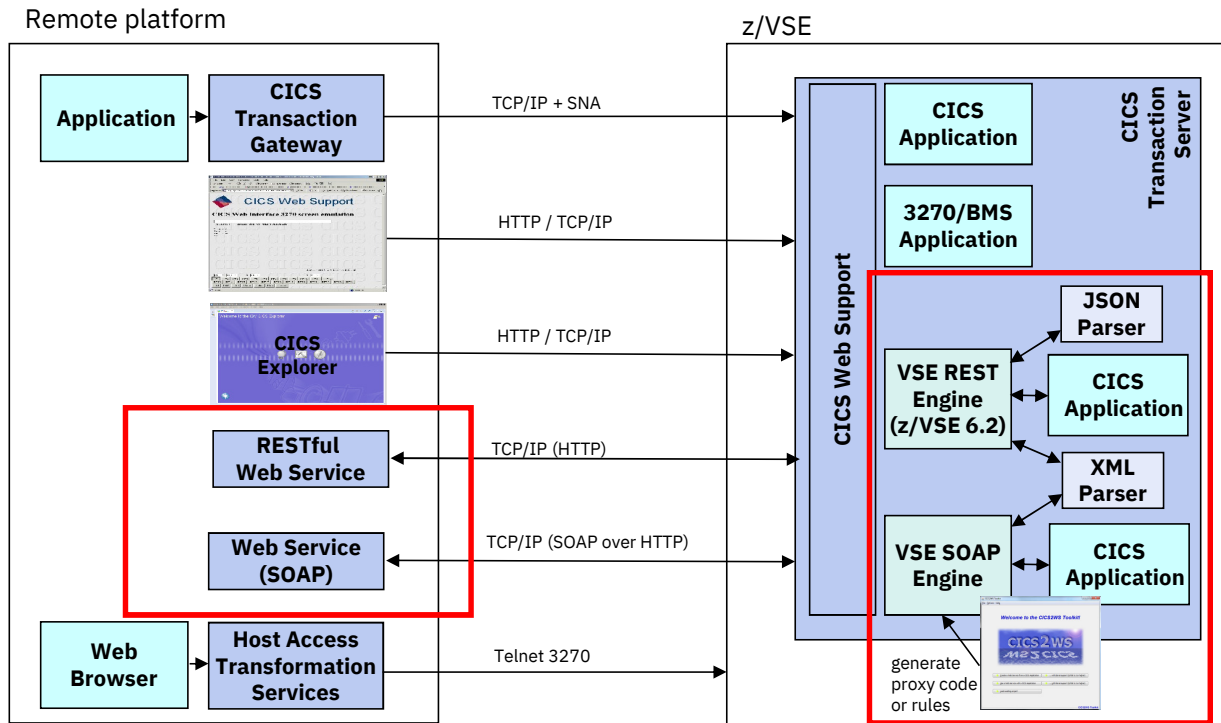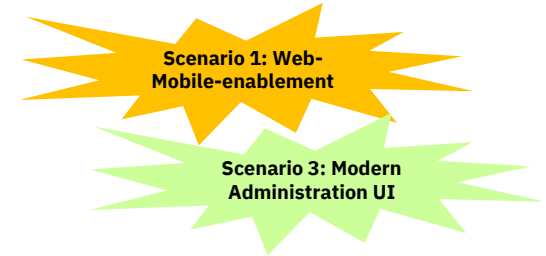
SOAP / Web Services support

— **Web Service-enable z/VSE CICS TS applications**
  - **Provide existing CICS applications as Web Service** to the outside world
    - z/VSE as the SOAP server
  - **Use/call external Web Services** from within z/VSE CICS applications
    - z/VSE as the SOAP client
  - CICS2WS Tool is used to generate proxy code or mapping rules

— **z/VSE 6.2:**
  - z/VSE SOAP Engine now supports
    Channels & Containers

NEW!

z/VSE

Remote platform

SOAP
Web Service

TCP/IP (SOAP over HTTP)

CICS Web Support

CICS
Transaction
Server

XML
Parser

VSE SOAP
Engine

CICS
Application

Welcome to the CICS2WS Toolkit

cics2ws
m@3cics

generate
proxy code
or rules

17
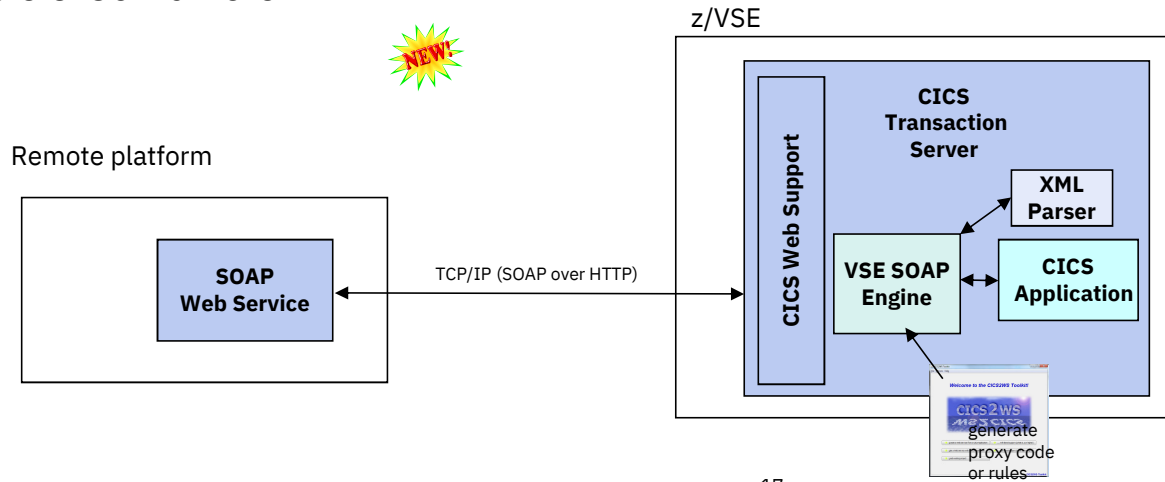
z/VSE 6.2: RESTful Web Services support
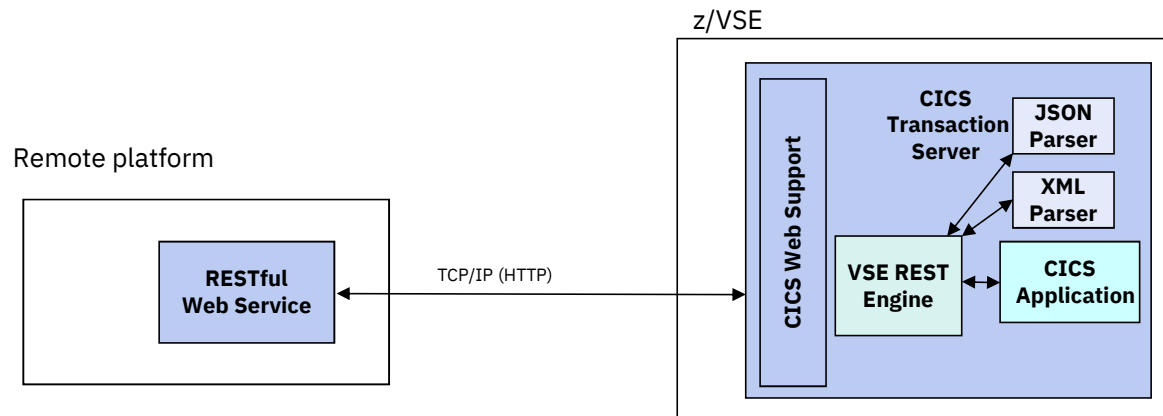
— **Use REST (Representational State Transfer) with CICS applications**

- **Provide existing CICS applications as RESTful Web Service to the outside world**
  - z/VSE as the REST server
  - Provide an easy to use RESTful API to services for z/VSE services
- **Use/call external RESTFul Web Services from within z/VSE CICS applications**
  - z/VSE as the REST client
  - Use external RESTful APIs within z/VSE applications
- **Payload can be:**
  - JSON (JavaScript Object Notation)
  - XML
  - Plain text, Binary, Form fields, Multipart

z/VSE

Remote platform

RESTful Web Service

TCP/IP (HTTP)

CICS Web Support

CICS Transaction Server

JSON Parser

XML Parser

VSE REST Engine

CICS Application

What is REST (Representational State Transfer)?

— Representational State Transfer (REST) is a **software architecture style consisting of guidelines and best practices** for creating web services

— REST has gained widespread acceptance across the web as a **simpler alternative to SOAP** and WSDL-based web services

— RESTful systems typically communicate over the **Hypertext Transfer Protocol** (**HTTP**)
  • with the same HTTP verbs (GET, POST, PUT, DELETE, and so on) used by web browsers

— The **payload** (message) transported by RESTful web services can be of various types (content types)
  • Commonly used is **JSON** as well as **XML**, but it can also be plain text, or even binary data

What is REST (Representational State Transfer)?

— A RESTful web service typically operates on a certain **'object'** on a server
  - The object is typically addressed through the URI (part of the URL)
    - http://host:port/resource-uri

— Actions on such resources are typically denoted by the HTTP request types:
  - **GET**   would typically **read** the resource
  - **PUT**   would typically **update/replace** the resource
  - **POST**   would typically **create** the resource
  - **DELETE**   would typically **delete** the resource

— Additional parameters can be supplied via the URL query string
  - http://host:port/resource-uri?query-string

What is REST (Representational State Transfer)?

— RESTful web services are typically **stateless**
  - Each request from any client contains all the information necessary to service the request
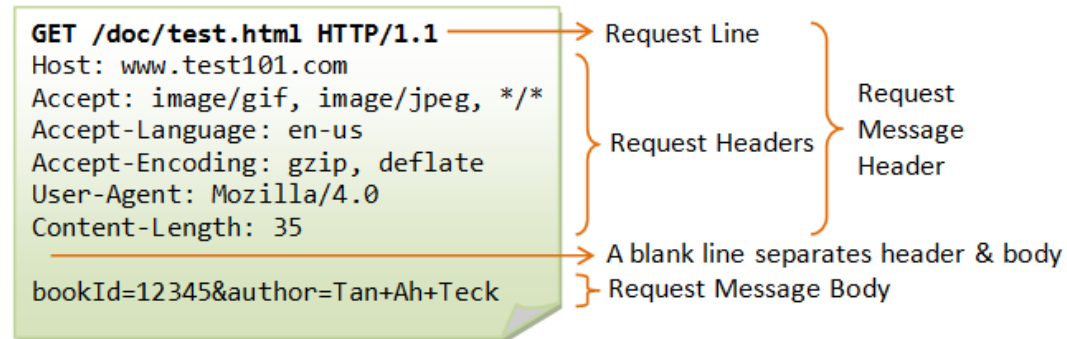 - The session state is therefore held in the client

— RESTful web services may use **HTTP specific features**
  - **HTTP headers** – to transport additional attributes
  - **Cookies** – to manage state information between requests

→ As denoted by the term 'typically' in above descriptions, there is no hard requirement for any of the described properties
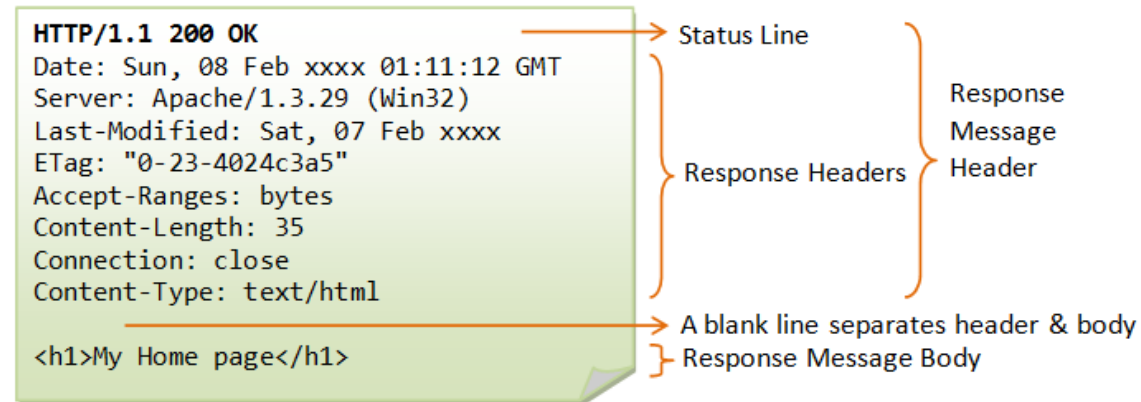
HTTP basics

Request:

```
GET /doc/test.html HTTP/1.1          ────→  Request Line
Host: www.test101.com
Accept: image/gif, image/jpeg, */*                          Request
Accept-Language: en-us               ──  Request Headers     Message
Accept-Encoding: gzip, deflate                               Header
User-Agent: Mozilla/4.0
Content-Length: 35
                                     ────→  A blank line separates header & body
bookId=12345&author=Tan+Ah+Teck      ──  Request Message Body
```

Response:

```
HTTP/1.1 200 OK                      ────→  Status Line
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)                               Response
Last-Modified: Sat, 07 Feb xxxx                             Message
ETag: "0-23-4024c3a5"                ──  Response Headers    Header
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html
                                     ────→  A blank line separates header & body
<h1>My Home page</h1>                ──  Response Message Body
```

Taken from: https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html

Example: A RESTful web service

Request a list of books:

```
Request:
GET /api/v1/books

Response:
{
  meta: {  },
  data: [{
    id: 24,
    title: 'Behavior-Driven Development',
    author: 'Viktor Farcic'
  }, {
    id: 25,
    title: 'Continuous Integration',
    author: 'Viktor Farcic'
  }]
}
```

Create a book:

```
Request:
POST /api/v1/books/id/24
{
  id: 24,
  title: 'Behavior-Driven Development',
  author: 'Viktor Farcic'
}

Response:
Status: 201 Created
{
  meta: {  },
  data: {
    uri: /api/v1/books/id/24
  }
}
```

Request a single book:

```
Request:
GET /api/v1/books/id/24

Response:
{
  meta: {  },
  data: {
    id: 24,
    title: 'Behavior-Driven Development',
    author: 'Viktor Farcic'
  }
}
```

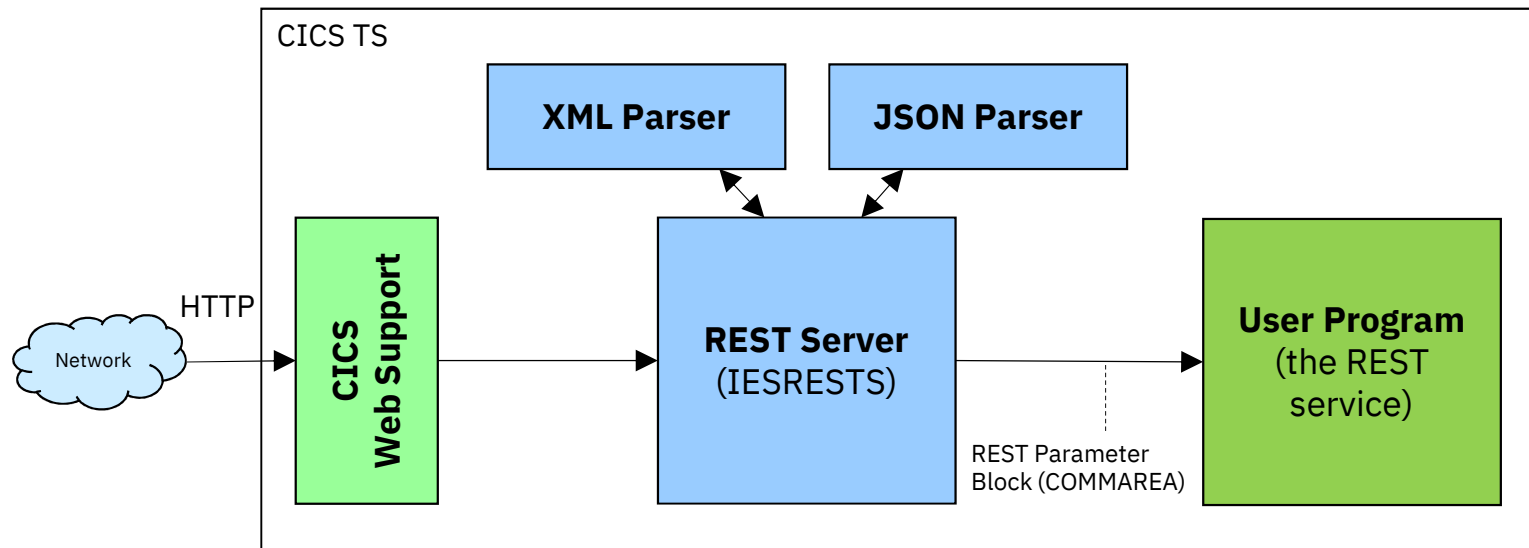**Delete a book:**

```
Request:
DELETE /api/v1/books
{
  id: 24,
}

Response:
Status: 202 Accepted
{
  meta: {  },
  data: {  }
}
```

Example taken from: https://technologyconversations.com/2014/08/12/rest-api-with-json/

z/VSE 6.2: z/VSE as a REST Server

CICS TS

XML Parser   **JSON Parser**

HTTP

Network → **CICS Web Support** → **REST Server (IESRESTS)** → **User Program** (the REST service)
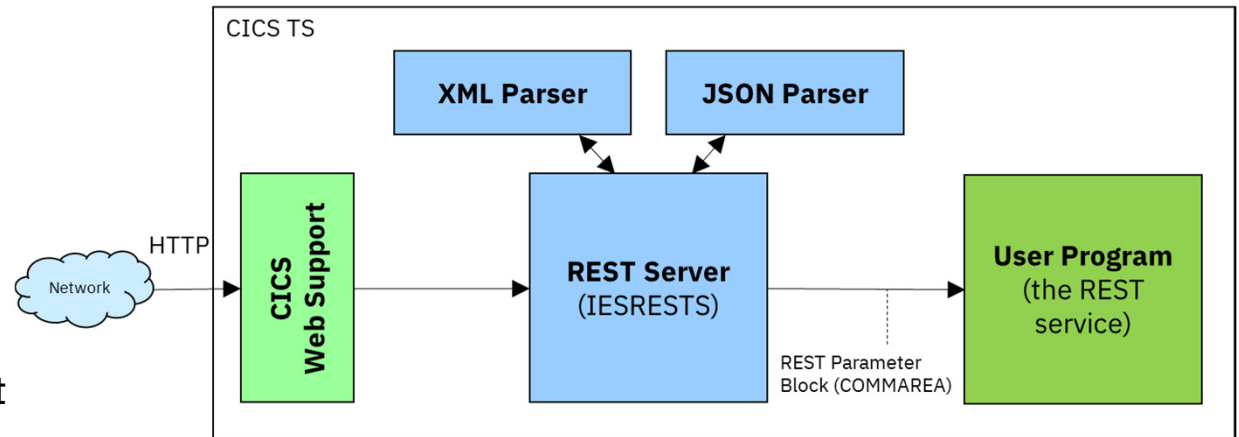
REST Parameter Block (COMMAREA)

Description of the REST Parameter Block:
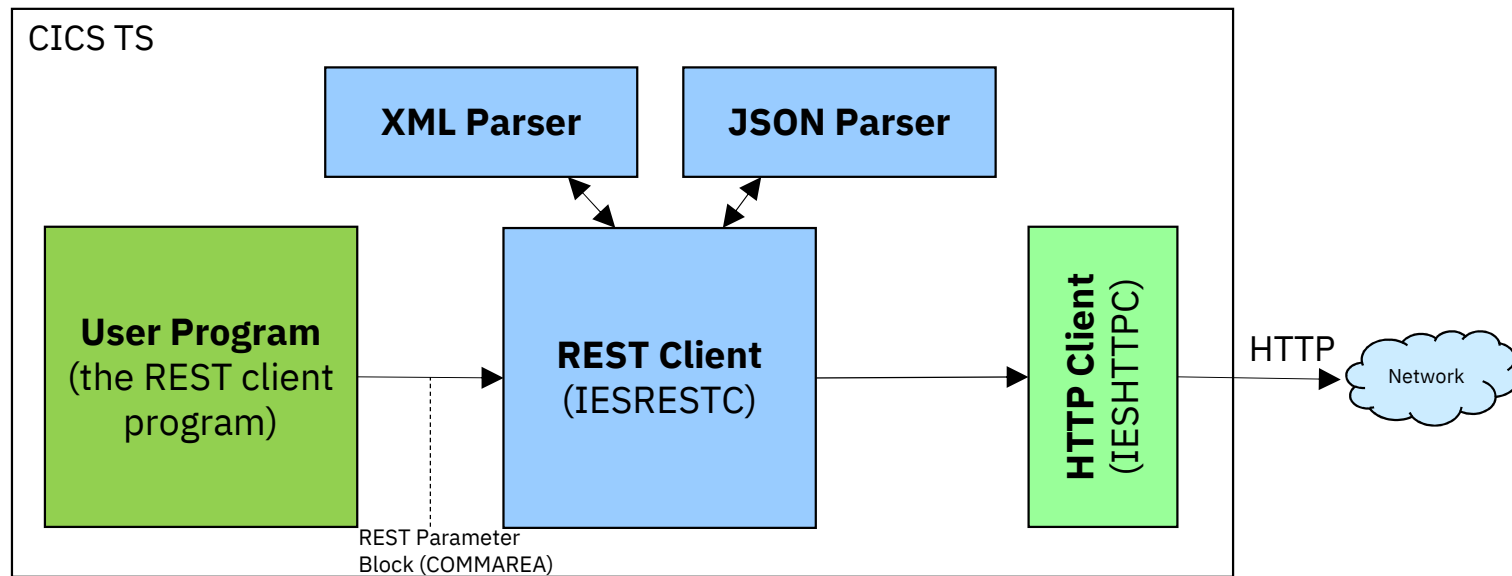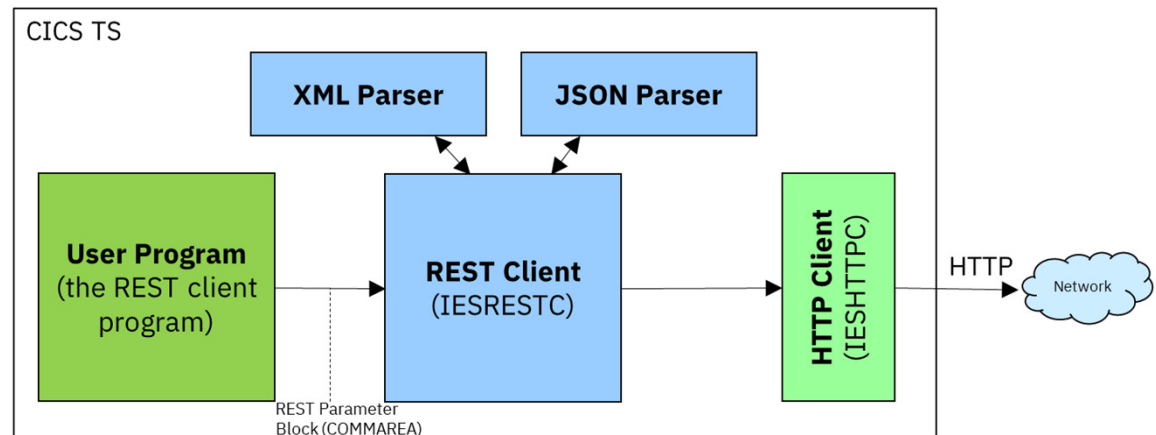https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ws_how_rest_control_blocks_are_used.html

z/VSE 6.2: z/VSE as a REST Server

— **The z/VSE REST-Engine…**
- Receives the request (via CICS Web Support)
- Extracts information from the request:
  - User program to call from the URL:
    http://host:port/cics/CWBA/IESRESTS/user-program/resource-uri[?query-string]
  - URL parameters from the query string (if any)
  - HTTP headers
  - Cookies (if any)
  - Request data (if any)
- Calls the user program
- Constructs the response:
  - HTTP status code
  - HTTP headers
  - Set-Cookies requests (if any)
  - Response data (if any)
- Sends the response back to the client

z/VSE 6.2: z/VSE as a REST Client

Description of the REST Parameter Block:
https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ws_how_rest_control_blocks_are_used.html

z/VSE 6.2: z/VSE as a REST Client

— **The z/VSE REST-Engine...**

- Gets called from the user program
- Constructs the request
  - ○ Splits the URL into host and port and resource-uri
    http://host:port/resource-uri[?query-string]
  - ○ Adds URL parameters to the query string (if any)
  - ○ HTTP headers
  - ○ Cookies (if any)
  - ○ Request data (if any)
- Sends the request to the server
- Receives the response from the server
- Extracts information from the response:
  - ○ HTTP status code
  - ○ HTTP headers
  - ○ Set-Cookies requests (if any)
  - ○ Response data (if any)
- Returns back to the user program

CICS TS

XML Parser   JSON Parser

User Program
(the REST client
program)

REST Client
(IESRESTC)

HTTP Client
(IESHTTPC)

HTTP

Network

REST Parameter
Block (COMMAREA)

The REST parameter block

— ## Contains information about the request

- Request type (GET, PUT, POST, …)
- URL
- Content-Type
- Data-type (XML, JSON, plain text, binary)
- URL parameters from query string (http://…..?a=b&c=d)
- Form fields
- HTTP headers
- Cookies
- Authentication information
- Response status code

— ## Copybooks in PRD1.BASE:

- IESRESTH.H   LE/C
- IESRESTL.C   COBOL
- IESJSONP.P   PL/1
- IESRESTA.A   HLASM

```
* COMMAREA layout used by the REST Engine to call the user program
* (VSE as REST server) or to get called by the user program
* (VSE as REST client):
*
 01 REST-COMMAREA.
    02 REST-VERSION             PIC 9(9) BINARY.
    02 REST-EBCDIC-CODEPAGE     PIC X(16).
    02 REST-FLAGS               PIC 9(9) BINARY.
    02 REST-RETCODE             PIC 9(9) BINARY.
    02 REST-PRIVATE             USAGE IS POINTER.
*  Request specific fields:
    02 REST-REQ-ACTION          PIC 9(9) BINARY.
    02 REST-REQ-URL             PIC X(2048).
    02 REST-REQ-CONTENT-TYPE    PIC X(128).
    02 REST-REQ-DATA-TYPE       PIC 9(9) BINARY.
    02 REST-REQ-DATA-PTR        USAGE IS POINTER.
    02 REST-REQ-DATA-LENGTH     PIC 9(9) BINARY.
    02 REST-REQ-URL-PARAMS-TSQ  PIC X(8).
    02 REST-REQ-FORM-FIELDS-TSQ PIC X(8).
    02 REST-REQ-HTTP-HEADERS-TSQ PIC X(8).
    02 REST-REQ-COOKIES-TSQ     PIC X(8).
    02 REST-REQ-AUTH-TYPE       PIC 9(9) BINARY.
    02 REST-REQ-AUTH-USER       PIC X(64).
    02 REST-REQ-AUTH-PASSWORD   PIC X(64).
    02 REST-REQ-ACCEPT          PIC X(128).
*  Response specific fields:
    02 REST-RESP-HTTP-STATUS-CODE PIC 9(9) BINARY.
    02 REST-RESP-HTTP-STATUS-TEXT PIC X(128).
    02 REST-RESP-CONTENT-TYPE   PIC X(128).
    02 REST-RESP-DATA-TYPE      PIC 9(9) BINARY.
    02 REST-RESP-DATA-PTR       USAGE IS POINTER.
    02 REST-RESP-DATA-LENGTH    PIC 9(9) BINARY.
    02 REST-RESP-FORM-FIELDS-TSQ PIC X(8).
    02 REST-RESP-HTTP-HEADERS-TSQ PIC X(8).
    02 REST-RESP-COOKIES-TSQ    PIC X(8).
    02 REST-RESP-LOCATION       PIC X(2048).
```

Handling XML and JSON data

The z/VSE REST Engine automatically translates request and response data

— **XML:**   Content-Type: text/xml or application/xml
  - XML data is parsed by the XML parser
  - An XML tree in memory is passed to the user program

— **JSON:**   Content-Type: text/json or application/json
  - JSON data is parsed by the JSON parser
  - A JSON tree in memory is passed to the user program

— **URL encoded:**   Content-Type: application/x-www-form-urlencoded
  - Form field data is parsed and passed via TS queue entries

— **Plain text:**   Content-Type: text/* (other than xml or json)
  - ASCII-EBCDIC converted

— **Binary data:**   anything else

— **Multipart data:**
  - Each part is converted individually based on its content type

JSON data

Example:

```
{"menu": {
 "id": "file",
 "popup": {
  "menuitem": [
   {"value": "Open", "onclick": "OpenDoc()"},
   {"value": "Close", "onclick": "CloseDoc()"}
  ]
 }
}}
```

The JSON control blocks are defined in copybooks in PRD1.BASE:
- IESJSONH.H   LE/C
- IESJSONC.C   COBOL
- IESJSONP.P   PL/1
- IESJSONA.A   HLASM

name=	valueType=OBJECT
parent next	jsonNodeValue

name=menu	valueType=OBJECT
parent next	jsonNodeValue

name=id	valueType=STRING
parent next	stringValue=file

name=popup	valueType=OBJECT
parent next	jsonNodeValue

name=menuitem	valueType=ARRAY
parent next	jsonNodeValue

name=	valueType=OBJECT
parent next	jsonNodeValue

name=value	valueType=STRING
parent next	stringValue=Open

name=onClick	valueType=STRING
parent next	stringValue=OpenDoc()

name=	valueType=OBJECT
parent next	jsonNodeValue

name=value	valueType=STRING
parent next	stringValue=Close

name=onClick	valueType=STRING
parent next	stringValue=CloseDoc()

XML data

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<VSAMmap type="temporary" status="test">
<!-- VSAM map layout -->
<mapname>mapname12</mapname>
<field>
 <name>str</name>
 <type>STRING</type>
</field>
<field>
 <name>sign</name>
 <type>SIGNED</type>
</field>
</VSAMmap>
```

The XML control blocks are defined in copybooks in PRD1.BASE:
- IESXMLAH.H   LE/C
- IESXMLCB.C   COBOL
- IESXMLPL.P   PL/1
- IESXMLAS.A   HLASM

type=PI	data=?xml version="1.0" ?
parent sibling child	

type=START	data=VSAMmap
parent sibling child atts	

	name=type	value=temporary
nextatt		

	name=status	value=test
nextatt		

type=COMMENT	data=VSAM map layout
parent sibling child	

type=START	data=mapname
parent sibling child	

type=DATA	data=mapname12
parent sibling child	

type=START	data=field
parent sibling child	

type=START	data=name
parent sibling child	

type=DATA	data=str
parent sibling child	

type=START	data=type
parent sibling child	

type=DATA	data=STRING
parent sibling child	

LDAP Sign-on Support

z/VSE

CICS Applications and Batch Jobs

LDAP Sign-on

User mapping

Authenticate →

Identity Management

LDAP Directory

Manages Identities
- User-IDs
- Passwords
- Groups
- Access rights
- Policies

Sign-on

Authenticate

Authenticate

Add, Change, Delete, ...

User   User   User   User   Administrator

z/VSE SNMP Monitoring Agent support
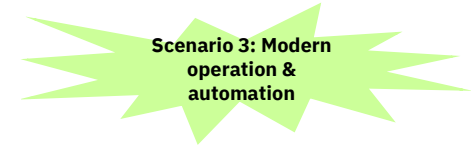
— **z/VSE Monitoring Agent enables customers to monitor z/VSE systems using standard monitoring interfaces (SNMP V1)**
  - Available since z/VSE V4.3
  - It also includes an open interface, which enables customers or vendors to use own programs (plugins) to collect additional data

— **Data collected by the IBM provided plugins contains**
  - Information about the environment (e.g. Processor, LPAR and z/VM information)
  - Number of partitions (static, dynamic, total, maximum)
  - Partition priorities
  - Number of CPUs (active, stopped, quiced)
  - Paging (page ins, page outs)
  - Performance counters overall and per CPU
  - CPU address and status
  - CPU time, NP time, spin time, allbound time
  - Number of SVCs and dispatcher cycles

— Plugin interface allows Vendors to provide additional monitoring data
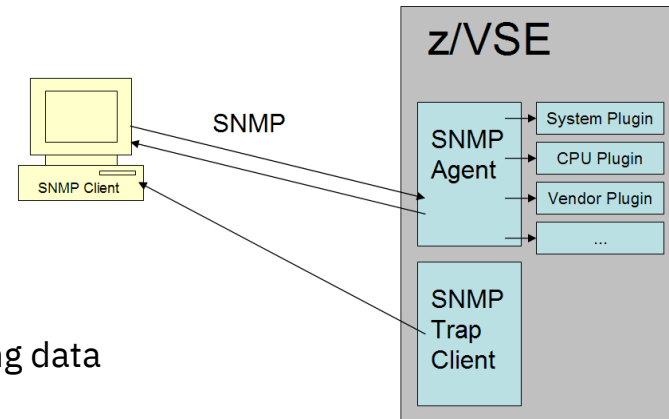
Rational Developer for IBM Z (RDz)

with z/VSE Development & Debug Plug-in for RDz – from QGroup

Scenario 3: Modern application development

Workstation

IBM Rational Developer for IBM Z

- SQL/CICS Parser
- HLASM Parser
- JCL Parser
- PL1 Parser
- COBOL Parser
- LPEX Editor
- Local COBOL/PL1 Compiler

z/VSE Development Plugin (Q-Group)

z/VSE Connector Client

z/VSE Debug Plugin (Q-Group)

Remote Debug API

Eclipse

z/VSE

VSE Connector Server

- VSAM
- ICCF
- LIBR
- POWER
- Console

- HLASM
- PL1 Compiler
- COBOL Compiler

TCP/IP

3270 Frontend

Remote Debug Interface

VSE Debug Tool

Debugged Program (Batch or CICS)

TCP/IP

VSE - RDZDEMO/PRINTAPP.cbl - IBM Rational Developer for System z

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access   ║ VSE ║ Debug ║ Java ║ Java EE ║ Database Development ║ CVS Repository Exploring

z/OS Projects

- RDZDEMO
  - .vseproperties
  - BuildOutput
  - BuildOutputRemote
  - jcl
  - PRINTAPP.cbl
  - RDZDEMO.link
  - run.jcl
  - run.jcl.out
  - rundebug.jcl
  - STARTAPP.cbl

*PRINTAPP.cbl

```
----+-*A-1-B--+----2----+----3----+----4----+----5----+----6----+----7--|-+--
       Identification Division.
       Program-ID.    PRINTAPP.

       Data Division.
       Working-Storage Section.
       01 Work-Parms.
          05 In-Len             PIC S9(4) BINARY.
          05 Char-count         Pic 99 Value ZEROS.
          05 Out-Name           PIC X(100).

          Linkage Section.
       01 Recvd-Parms.
          05 In-name            Pic x(30).


       Procedure Division using Recvd-Parms.
          Move spaces to Out-Name.

          Move 0 to Char-count3                    Remote Compile shows error
                                                        in source editor
          Inspect Function Reverse(In-Name)
              Tallying Char-count For Leading Spaces
          Compute In-Len = 30 - Char-count
```

VSE SYSTEM
- zPDT
- VSEDEMO
  - Librarian
  - Power
  - VSAM
  - ICCF
  - Console
- Local

O...   Pr...

PROGRAM: PRINTAPP
- Identification Division.
- Data Division.
- Procedure Division

Remote Error List   Problems   Tasks   VSE Console   Progress   History

Filter matched 2 of 2 messages

ID	Message	Sev...	Line	Location
i IGYOS4027	THE SYSTEM OPTION "SYM" IS INTERPRETED AS "MAP" AND MAY OVERRIDE THE COMPI...	0	-1	PRINTAPP
⊗ IGYPS2121	"CHAR-COUNT3" WAS NOT DEFINED AS A DATA-NAME. THE STATEMENT WAS DISCAR...	2	19	PRINTAPP

Compiler Messages after Remote Compile

"CHAR-COUNT3" WAS NOT DEFINED AS A DATA-NAME. THE STATEMENT WAS DISCARDED.

Summary

The z/VSE Connectors
provide lots of possibilities
to connect z/VSE
with the rest of the world.

40

Questions?

Notices and disclaimers

Notices and disclaimers continued

— Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products about this publication and cannot confirm the accuracy of 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. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a purpose.**

— The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

— IBM, the IBM logo, ibm.com and [names of other referenced IBM products and services used in the presentation] are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml