

Using the latest and greatest z/VSE Connectors features for Mobile, Analytics and asynchronous processing

Ingo Franzki, 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.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

- Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at http://www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT").
- No other workload processing is authorized for execution on an SE.
- IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

Agenda

- **Overview about z/VSE Connectors**

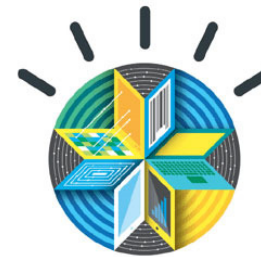
- **Including z/VSE data into Big Data analysis tools**

- Put z/VSE data into Apache Hadoop and run Big Data analysis tasks against it



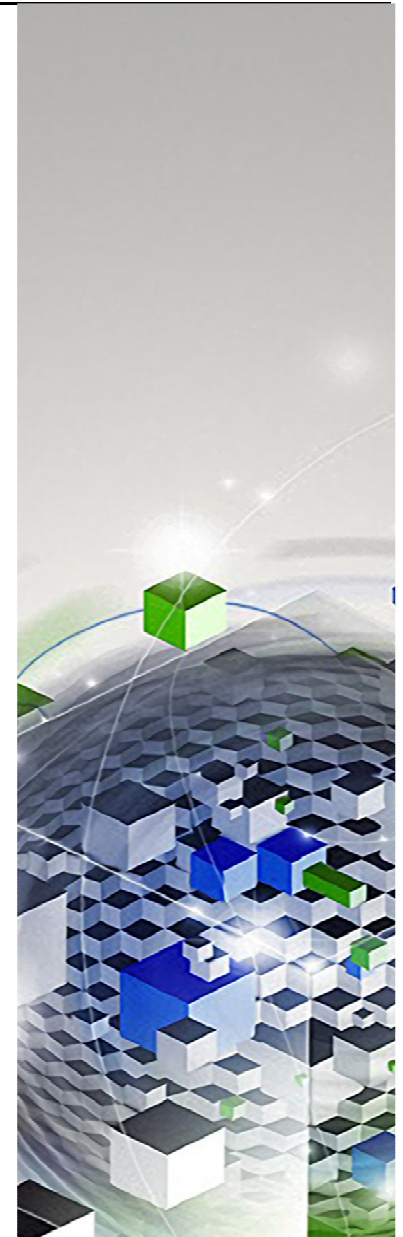
- **Mobile apps access your z/VSE resources, applications or data**

- Make your z/VSE applications and data accessible from mobile applications



- **Integrate z/VSE into asynchronous processes using WebSphere MQ**

- Use the new MQ Client Trigger Monitor to integrate your z/VSE applications into a WebSphere MQ environment for synchronous and asynchronous processing



Agenda

- **Overview about z/VSE Connectors**

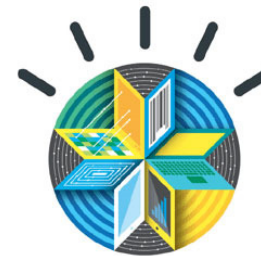
- **Including z/VSE data into Big Data analysis tools**

- Put z/VSE data into Apache Hadoop and run Big Data analysis tasks against it



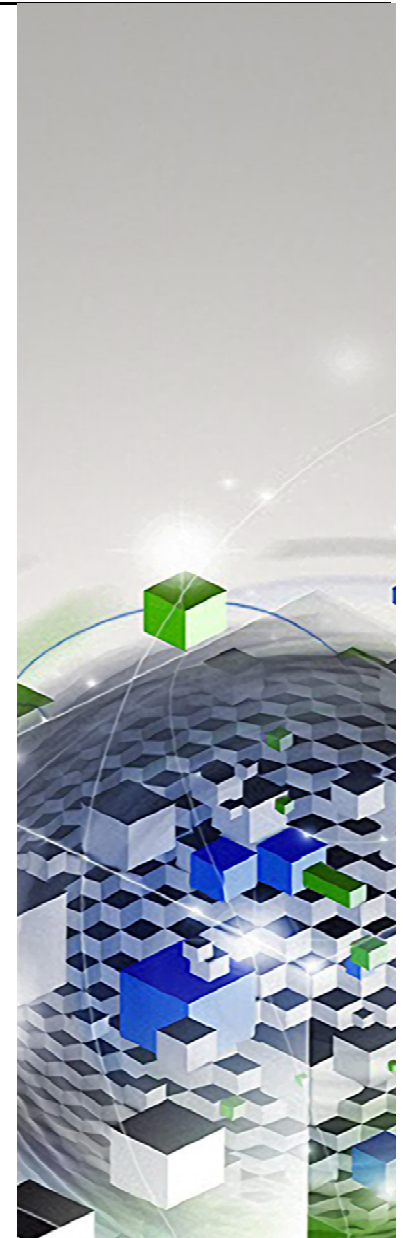
- **Mobile apps access your z/VSE resources, applications or data**

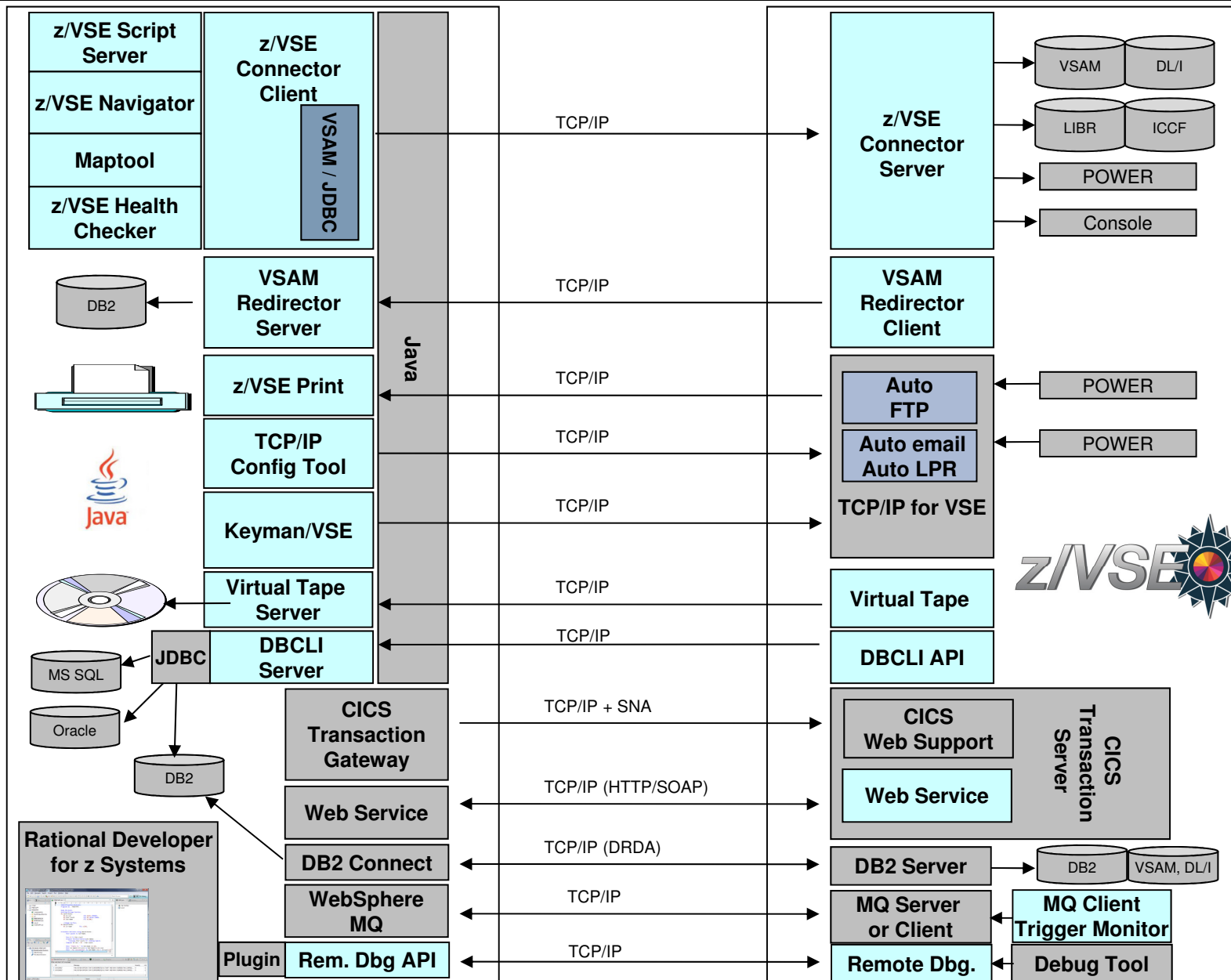
- Make your z/VSE applications and data accessible from mobile applications



- **Integrate z/VSE into asynchronous processes using WebSphere MQ**

- Use the new MQ Client Trigger Monitor to integrate your z/VSE applications into a WebSphere MQ environment for synchronous and asynchronous processing



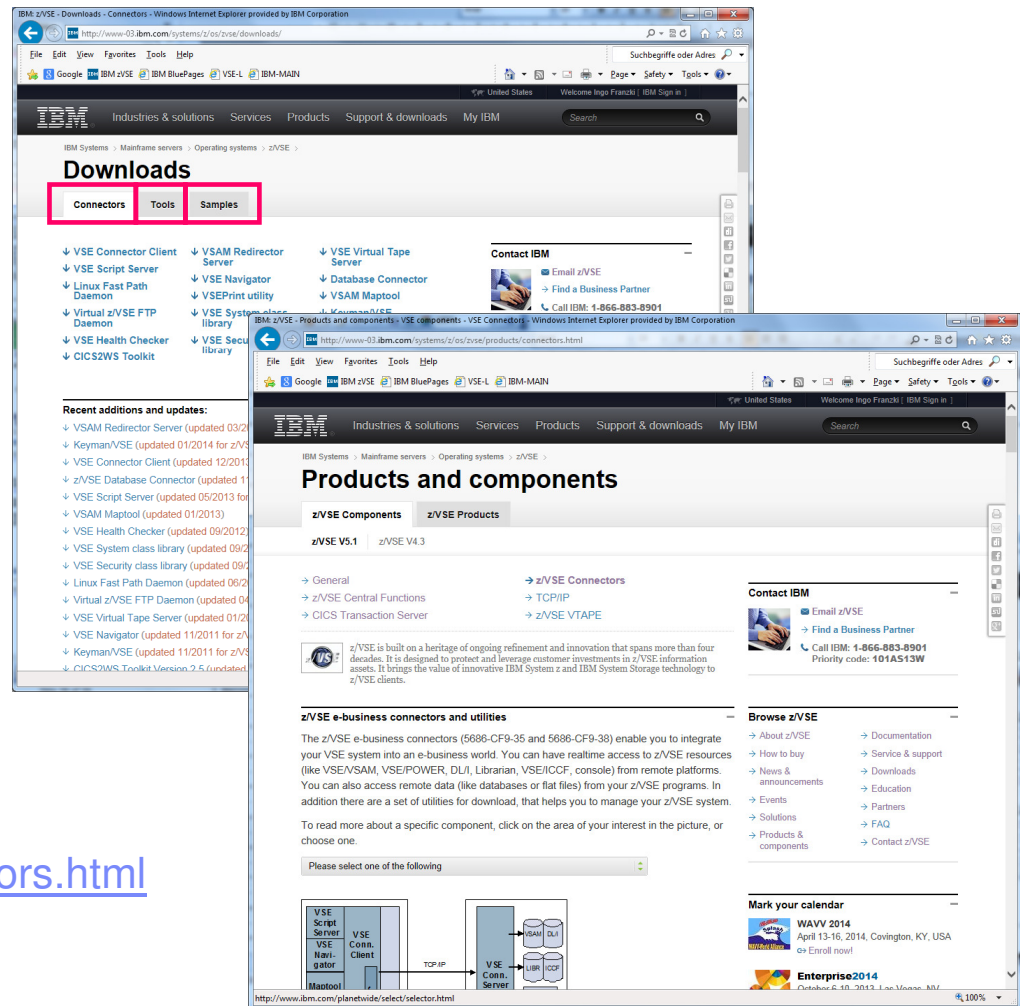


z/VSE Connector & Tools - Downloads

- **Connector components**
 - Delivered as part of the z/VSE system as WBOOKs in PRD2.PROD
 - Also available on the z/VSE Homepage
<http://ibm.com/zvse/downloads>

- **Many ‘as-is’ tools are provided for download**

- **Information about the Connector Components can be found here:**
<http://ibm.com/zvse/products/connectors.html>



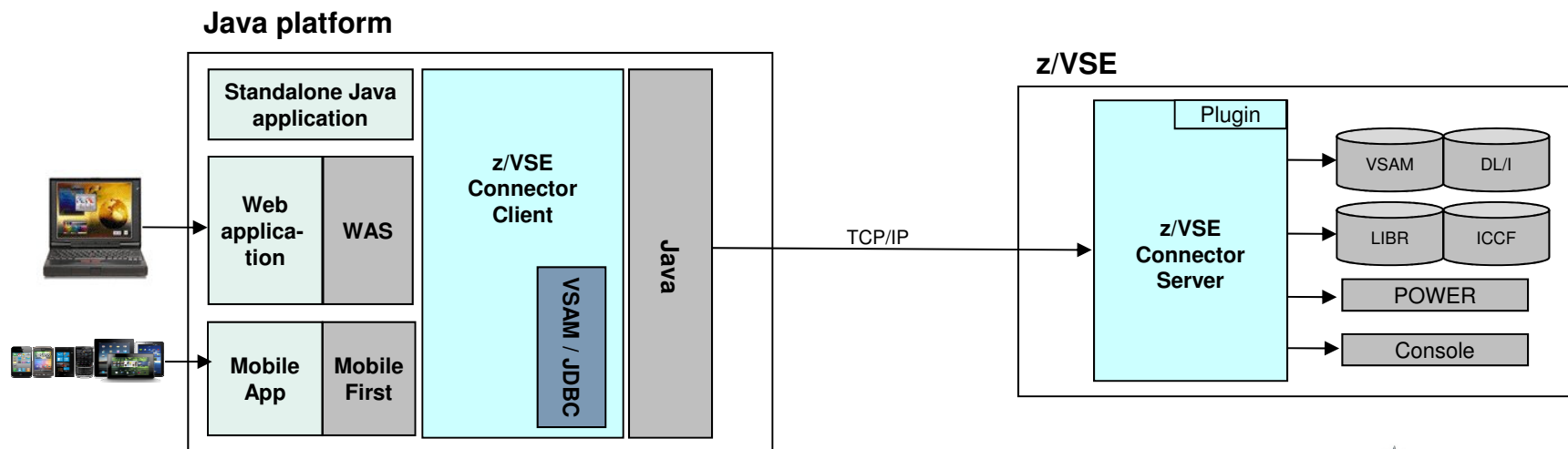
Java-based Connector

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
...or from web 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
- Extendable via Plugins



WebSphere software



VSAM Redirector



Synchronization of VSAM data with a database

- **Real-time: VSAM Redirector Client/Server**

- **OWNER=VSAM:**

- any INSERT, UPDATE or DELETE request is immediately replicated into the database
 - Read requests go against the VSAM dataset on z/VSE

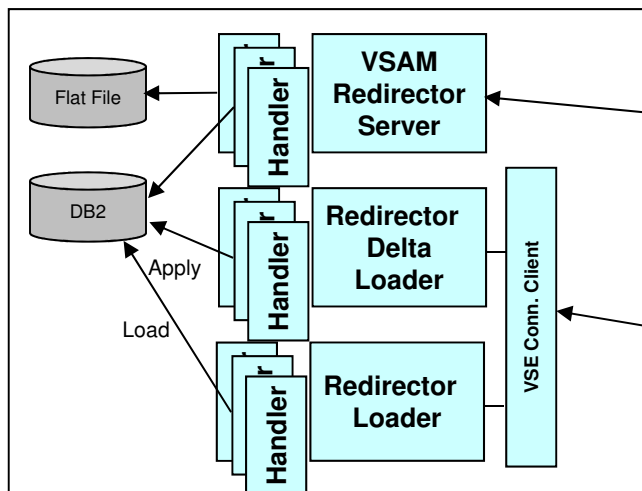
- **OWNER=REDIRECTOR:**

- All VSAM requests are sent to the database.
 - No access to the VSAM dataset anymore (except OPEN / CLOSE).

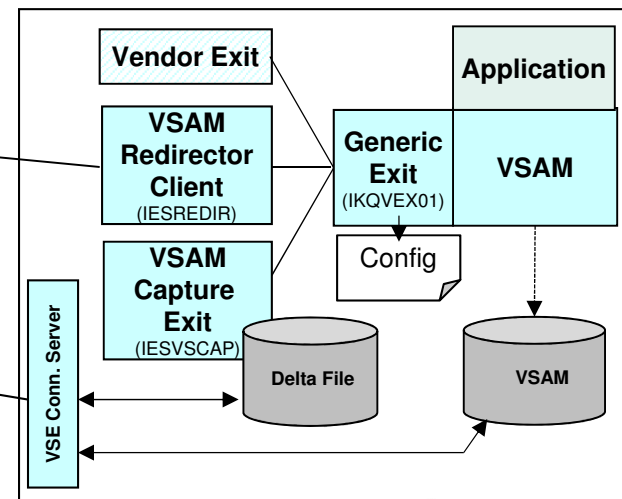
- **Near real-time: VSAM Capture Exit**

- Data changes are collected in a delta file or MQ queue
 - Delta file is downloaded and applied to database from time to time

Java platform



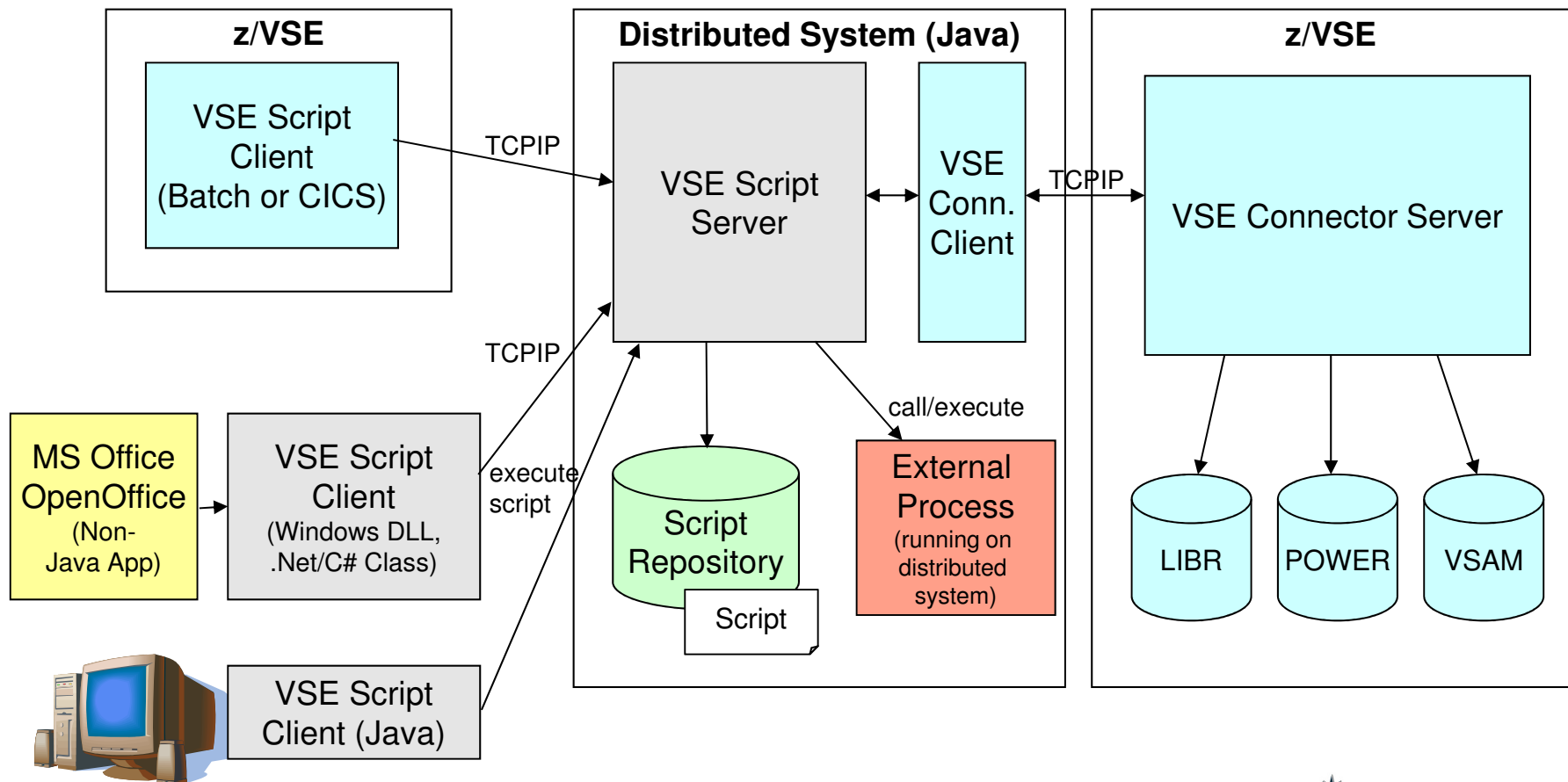
z/VSE



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



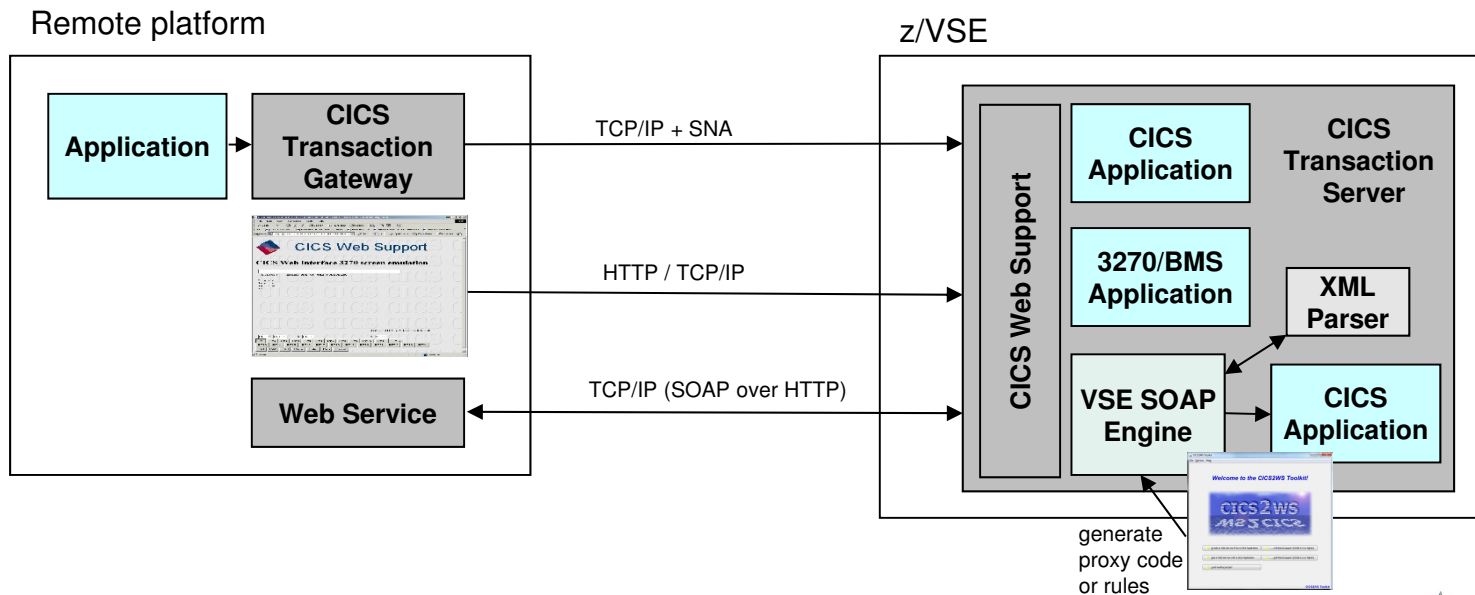
SOAP / Web Services support

Web Service-enable z/VSE CICS 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

CICS Web Support (CWS) is the base for all of this

- Including support for CICS Transaction Gateway and 3270-bridge



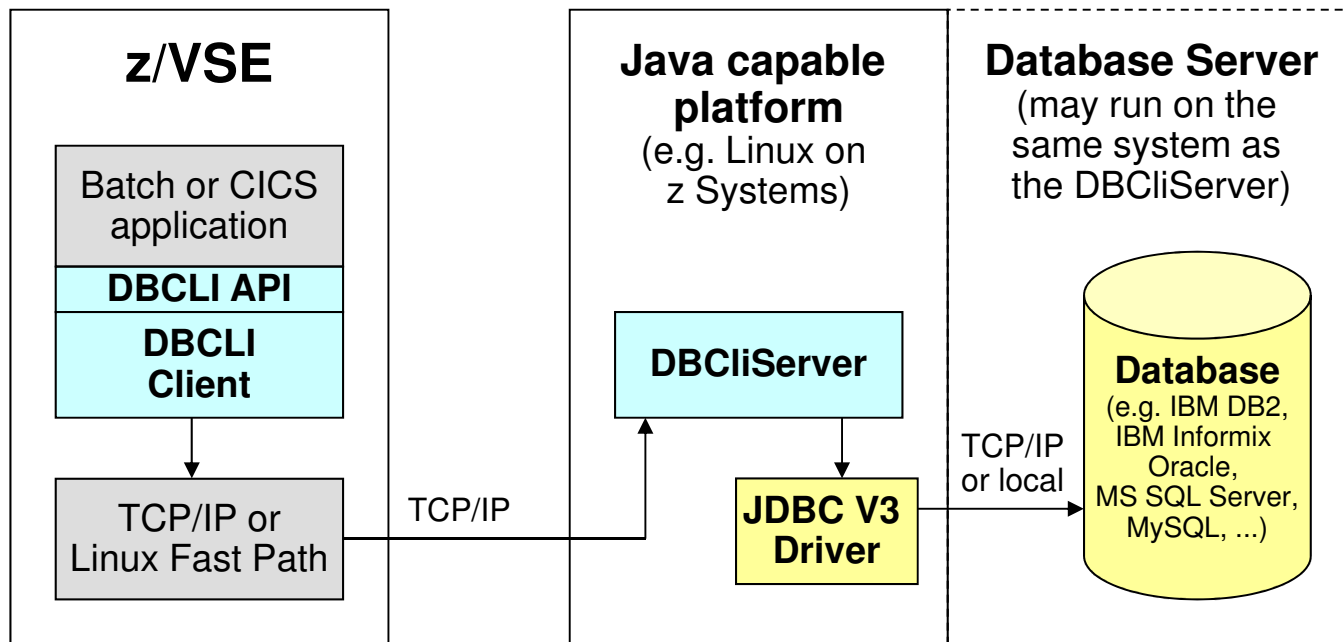
z/VSE Database Call Level Interface (DBCLI)

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**



Agenda

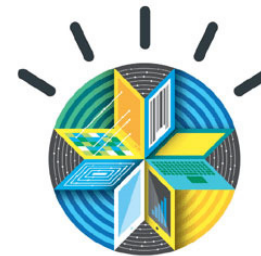
- Overview about z/VSE Connectors
- Including z/VSE data into Big Data analysis tools

- Put z/VSE data into Apache Hadoop and run Big Data analysis tasks against it



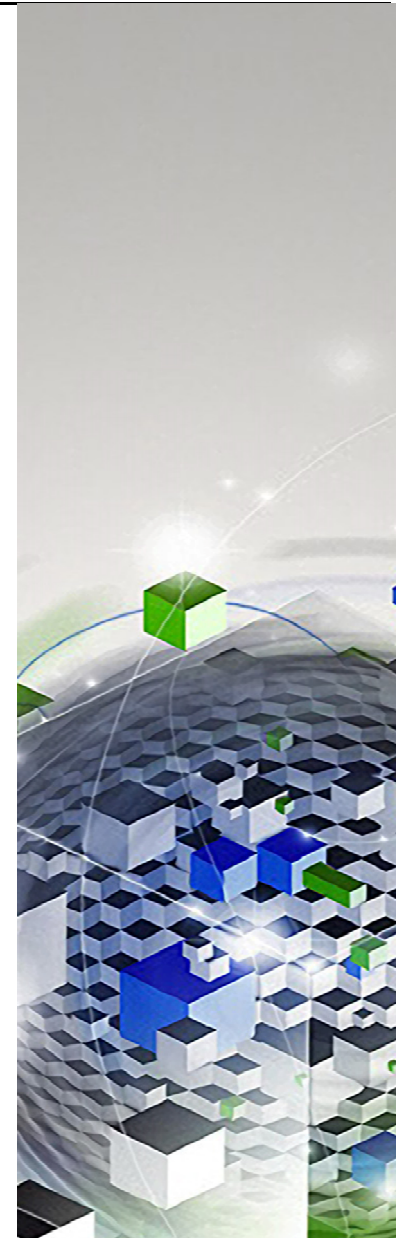
- Mobile apps access your z/VSE resources, applications or data

- Make your z/VSE applications and data accessible from mobile applications



- Integrate z/VSE into asynchronous processes using WebSphere MQ

- Use the new MQ Client Trigger Monitor to integrate your z/VSE applications into a WebSphere MQ environment for synchronous and asynchronous processing



What is Hadoop ?



<http://hadoop.apache.org/>

Apache Hadoop is an open source software project

- Enables distributed processing of large data sets across clusters of commodity servers.
- It is designed to scale up from a single server to thousands of machines, with very high degree of fault tolerance.
- Rather than relying on high-end hardware, the resiliency of these clusters comes from the software's ability to detect and handle failures at the application layer.

The project includes these modules:

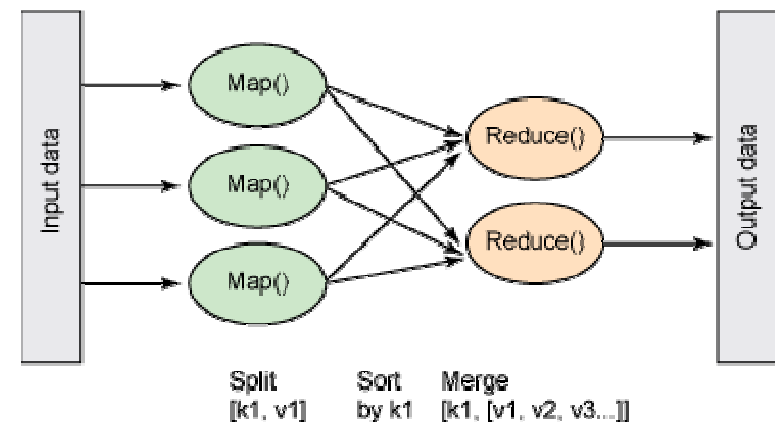
- **Hadoop Common:** The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS):** A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN:** A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce:** A YARN-based system for parallel processing of large data sets.

What is MapReduce?



MapReduce is the heart of Hadoop

- It is this programming paradigm that allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster.
- The term **MapReduce** actually refers to two separate and distinct tasks that Hadoop programs perform.
- The **map job**
 - takes a set of data and converts it into another set of data
 - where individual elements are broken down into tuples (key/value pairs).
- The **reduce job**
 - takes the output from a map as input
 - combines those data tuples into a smaller set of tuples.
- As the sequence of the name MapReduce implies, the reduce job is always performed after the map job.



MapReduce Example



Assume you have 5 files

- Each file contains two columns (**a key and a value** in Hadoop terms) that represent
 - a **city**
 - the corresponding **temperature recorded** in that city

➤ We want to find the **maximum temperature for each city across all of the data files**

- **Using the MapReduce framework, we can break this down into 5 map tasks**

- Each mapper works on one of the 5 files
- The mapper task goes through the data and returns the maximum temperature for each city.

- **The results produced from one mapper task for the data above looks like this:**

```
(Toronto, 20) (Whitby, 25) (New York, 22) (Rome, 33)
```

Toronto	20
Whitby	25
New York	22
Rome	32
Toronto	4
Rome	33
New York	18

MapReduce Example



- **Let's assume the other four mapper tasks produced the following intermediate results:**

```
(Toronto, 18) (Whitby, 27) (New York, 32) (Rome, 37) (Toronto, 32)
(Whitby, 20) (New York, 33) (Rome, 38) (Toronto, 22) (Whitby, 19)
(New York, 20) (Rome, 31) (Toronto, 31) (Whitby, 22) (New York, 19)
(Rome, 30)
```

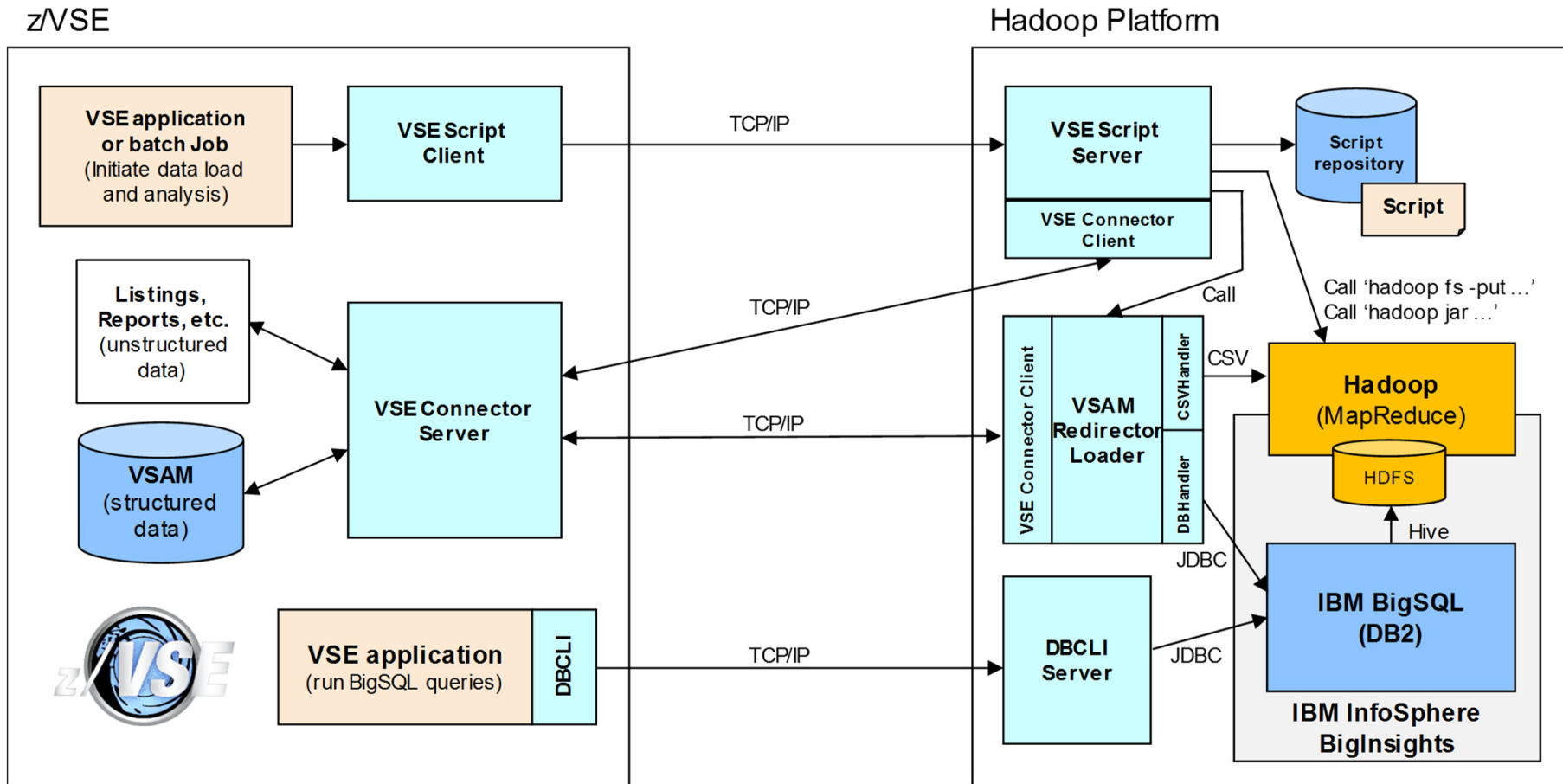
- **All five of these output streams are fed into the reduce tasks**

- Sorted by City
- Which combine the input results
- Output a single value for each city

- **This produces the final result set as follows:**

```
(Toronto, 32) (Whitby, 27) (New York, 33) (Rome, 38)
```

Using the z/VSE Connectors to analyze VSE data with Hadoop



Whitepaper: Big Data and Hadoop with z/VSE
ftp://public.dhe.ibm.com/eserver/zseries/zos/vse/pdf3/BigData_with_zVSE.pdf

Agenda

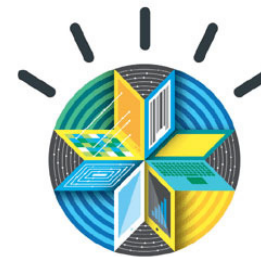
- Overview about z/VSE Connectors
- Including z/VSE data into Big Data analysis tools

- Put z/VSE data into Apache Hadoop and run Big Data analysis tasks against it



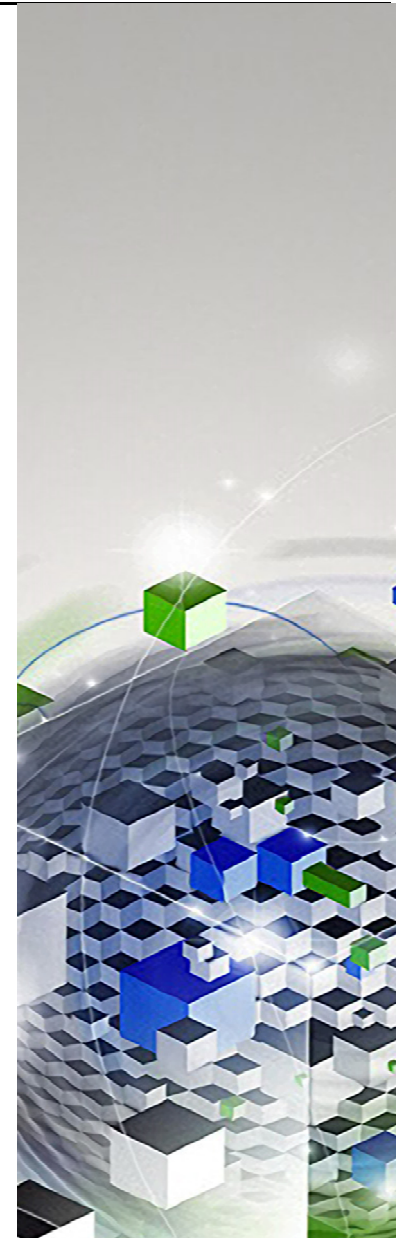
- **Mobile apps access your z/VSE resources, applications or data**

- Make your z/VSE applications and data accessible from mobile applications



- Integrate z/VSE into asynchronous processes using WebSphere MQ

- Use the new MQ Client Trigger Monitor to integrate your z/VSE applications into a WebSphere MQ environment for synchronous and asynchronous processing



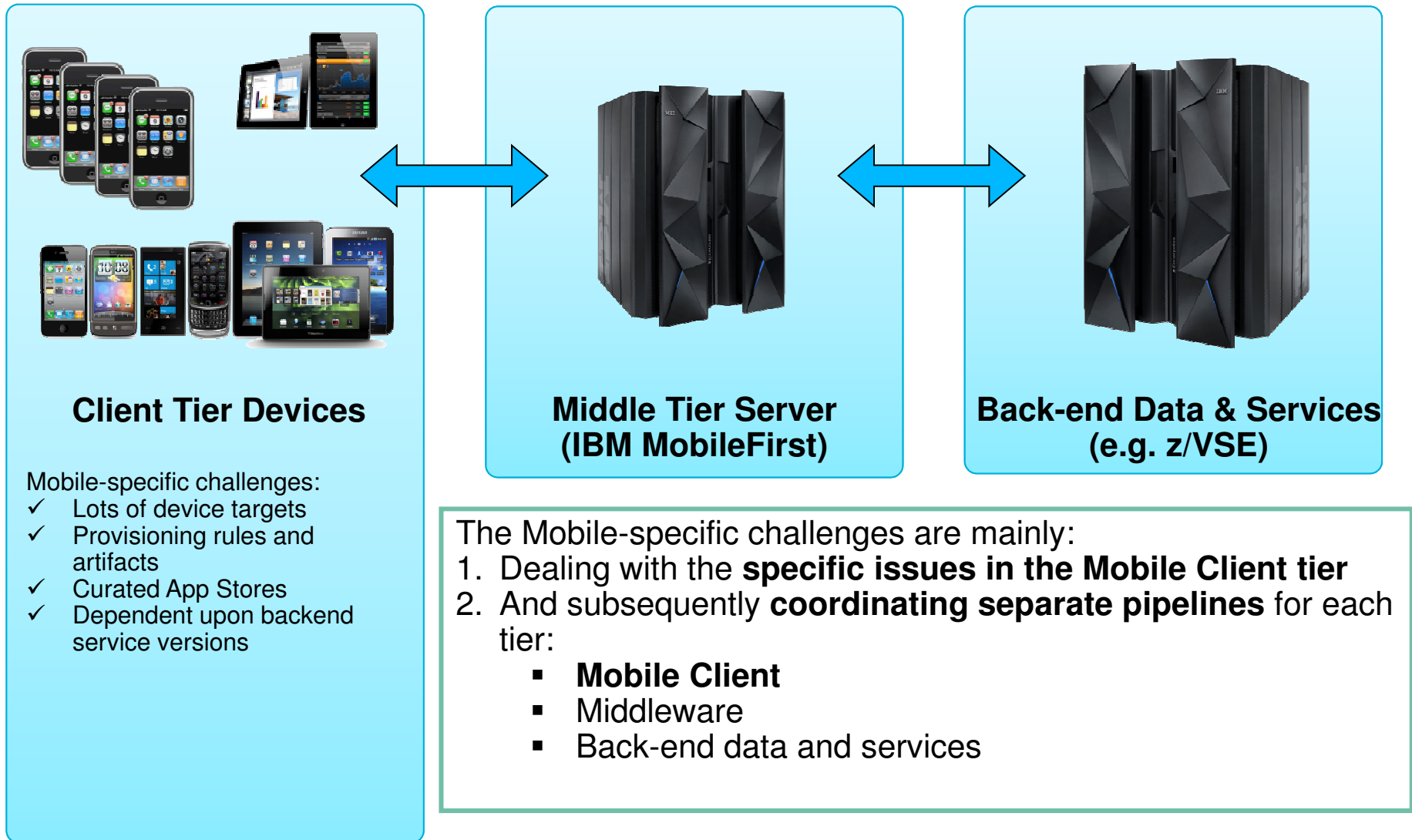
The world is going mobile

The mainframe...

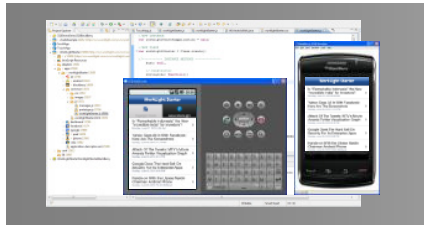
- Home to business critical applications and data
- How do we bridge the gap?



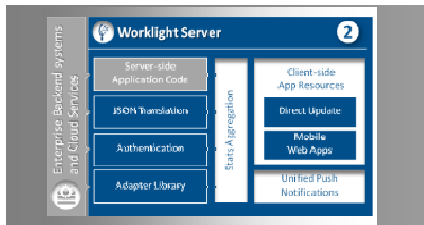
Multi-tier Mobile Apps



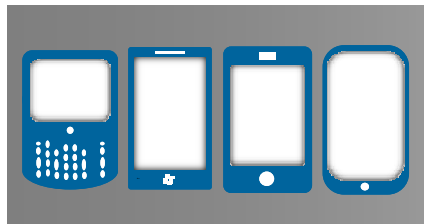
IBM MobileFirst Platform Foundation Overview



IBM MobileFirst Studio
The most complete, extensible environment with maximum code reuse and per-device optimization



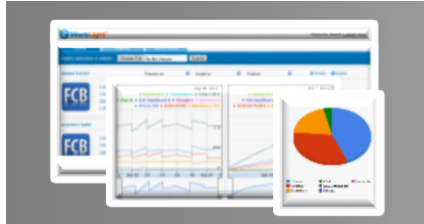
IBM MobileFirst Server
Unified notifications, runtime skins, version management, security, integration and delivery



IBM MobileFirst Device Runtime Components
Extensive libraries and client APIs that expose and interface with native device functionality



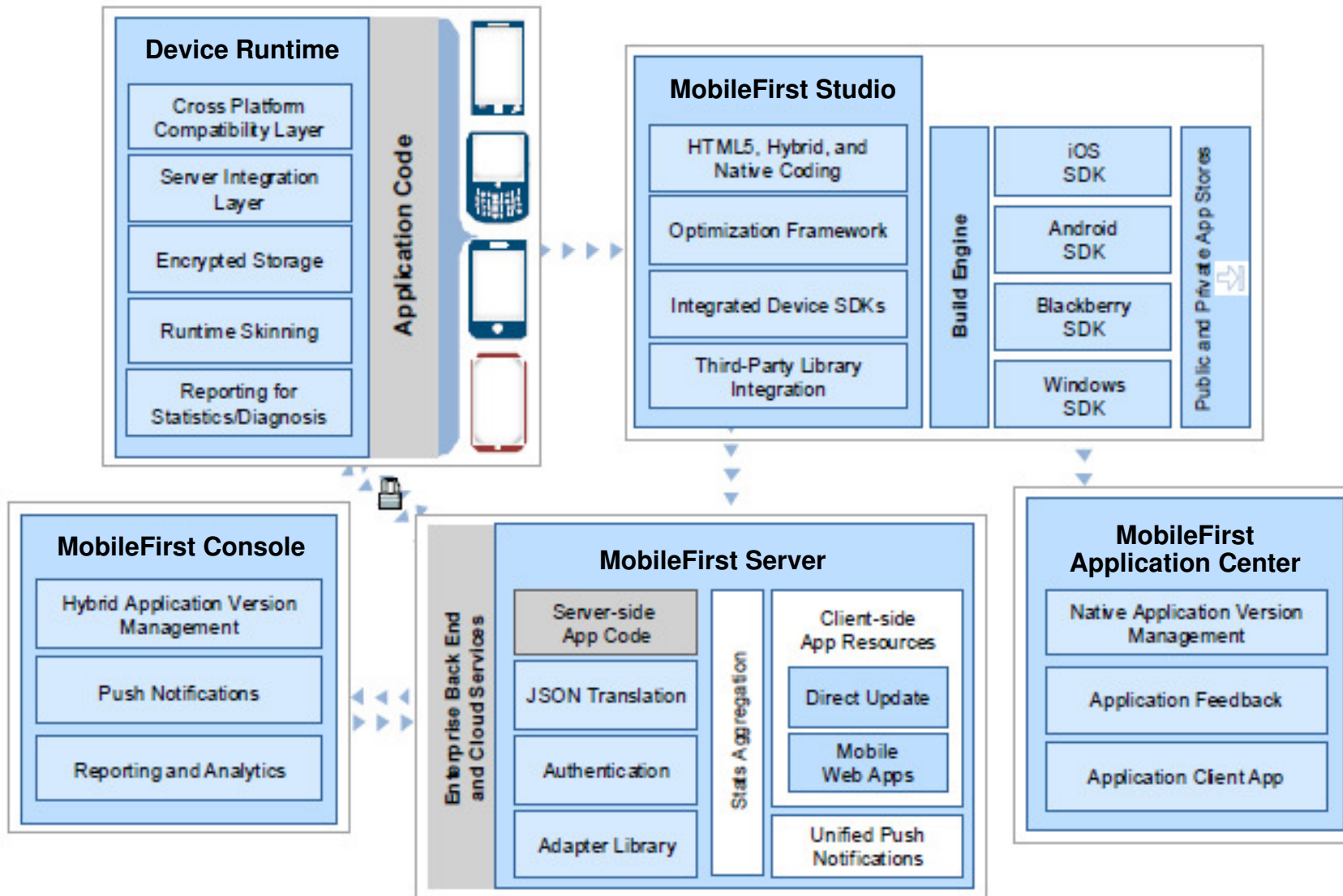
IBM MobileFirst Application Center
The MobileFirst Application Center can function as an enterprise application storage to deploy mobile applications across mobile platforms.



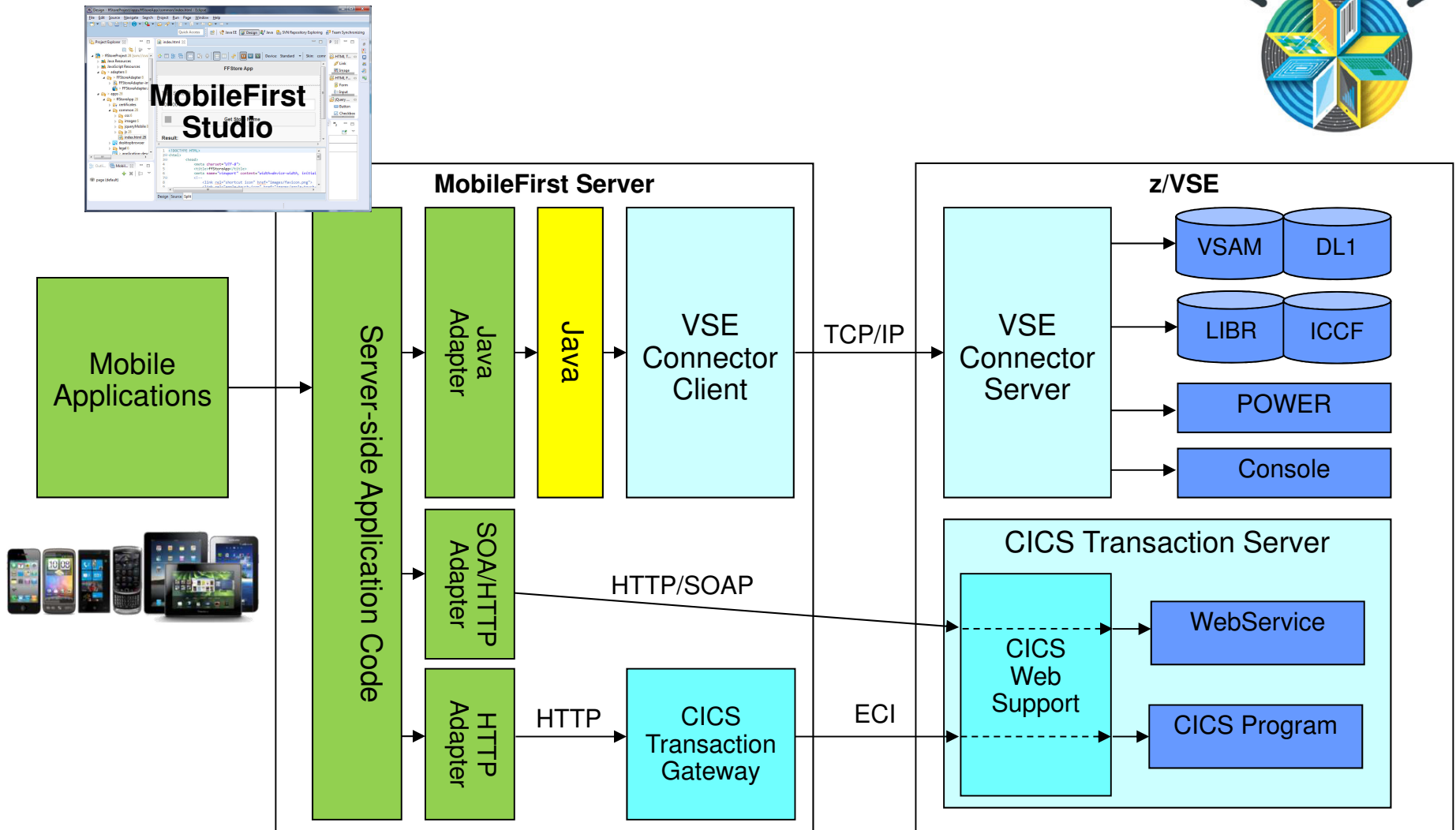
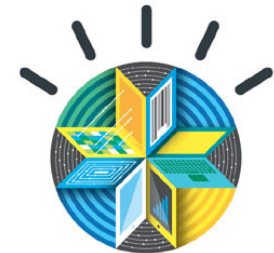
IBM MobileFirst Console
A web-based console for real-time analytics and control of your mobile apps and infrastructure



IBM MobileFirst Platform Foundation Components



Mobile applications accessing z/VSE data and programs



Whitepaper: Getting started with mobile development for z/VSE

<http://public.dhe.ibm.com/eserver/zseries/zos/vse/pdf3/GettingStartedWithMobileDevelopmentForVSE.pdf>

Agenda

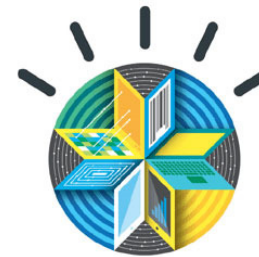
- Overview about z/VSE Connectors
- Including z/VSE data into Big Data analysis tools

- Put z/VSE data into Apache Hadoop and run Big Data analysis tasks against it



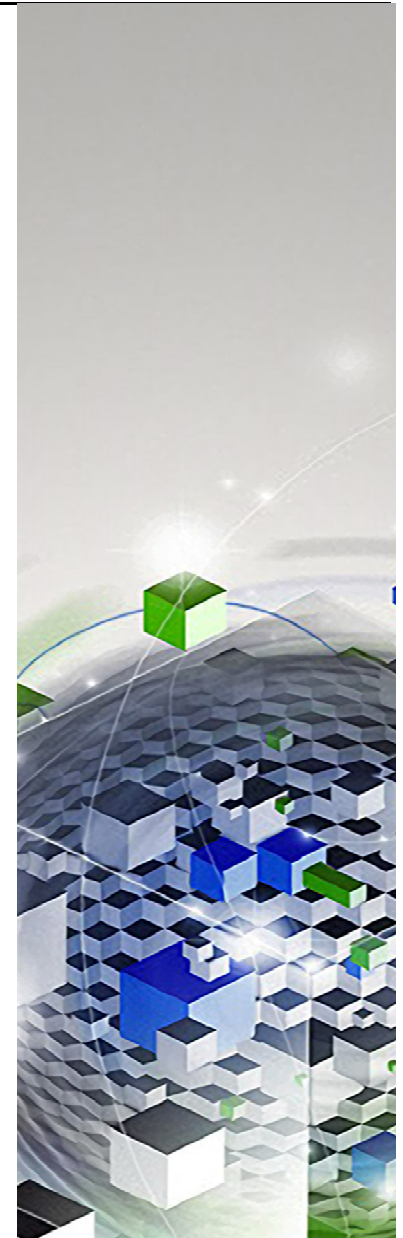
- Mobile apps access your z/VSE resources, applications or data

- Make your z/VSE applications and data accessible from mobile applications



- Integrate z/VSE into asynchronous processes using WebSphere MQ

- Use the new MQ Client Trigger Monitor to integrate your z/VSE applications into a WebSphere MQ environment for synchronous and asynchronous processing



WebSphere MQ for z/VSE V3.0 (5655-U97)

The WebSphere MQ for z/VSE V3.0 (aka MQ Server) is no longer available on z/VSE:

- Withdrawn from marketing (EoM) on September 8, 2014
- Withdrawn from service (EoS) effective September 30, 2015



The EoM announcement letter 914-104 contains the following notes:

- While IBM WebSphere MQ for z/VSE V3.0 is being withdrawn from marketing **with no direct replacement**, a **WebSphere MQ Client for VSE** will **continue to be available** to enable connectivity over WebSphere MQ from VSE environments to MQ deployments on other systems.
- One alternative option for users of WebSphere MQ for z/VSE V3.0 is to use the **WebSphere MQ Client for VSE** available as SupportPac MQC5 http://www.ibm.com/support/docview.wss?rs=171&uid=swg24010051&loc=en_US&cs=utf-8&lang=en
- Following the End of Support announcement for WebSphere MQ for VSE V3.0, there are **no immediate plans to withdraw support for this SupportPac** within the same timeframe.

WebSphere MQ Client for VSE

From the README.TXT file contained in the MQC5 package:

- A WebSphere MQ client is part of the product that can be installed on its own, on a separate machine from the base product and server.
- You can run a WebSphere MQ (MQSeries) application on a WebSphere MQ client and it can interact, by means of a communications protocol, with one or more WebSphere MQ servers and connect to their queue managers.
- The servers to which the client connects may or may not be part of a cluster.
- WebSphere MQ Family Platforms provides a list of which platforms are supported as WebSphere MQ Clients and/or Servers.

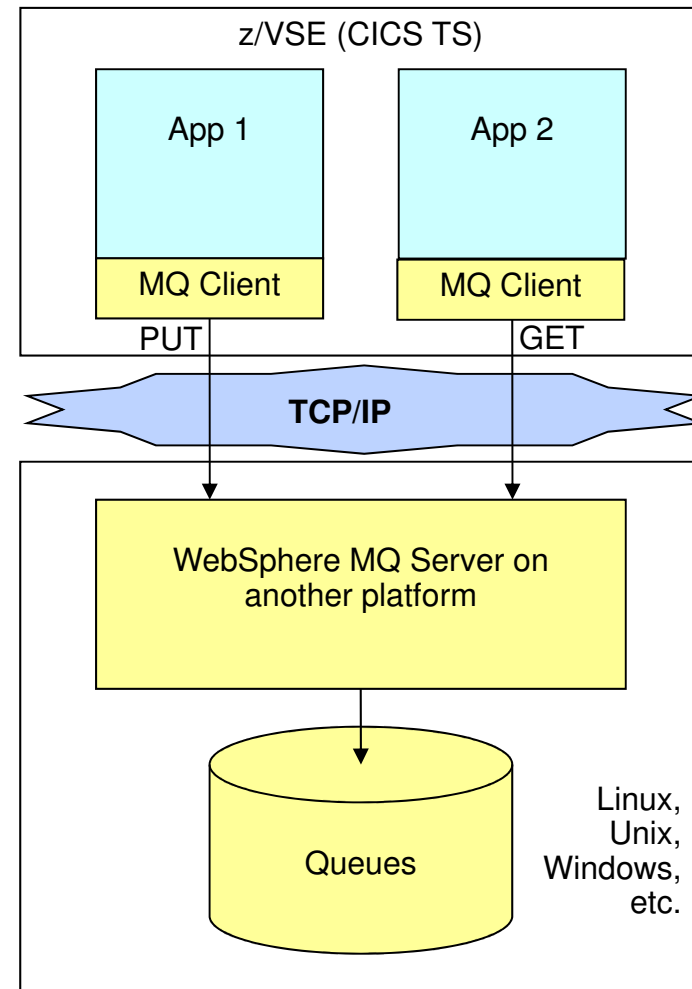
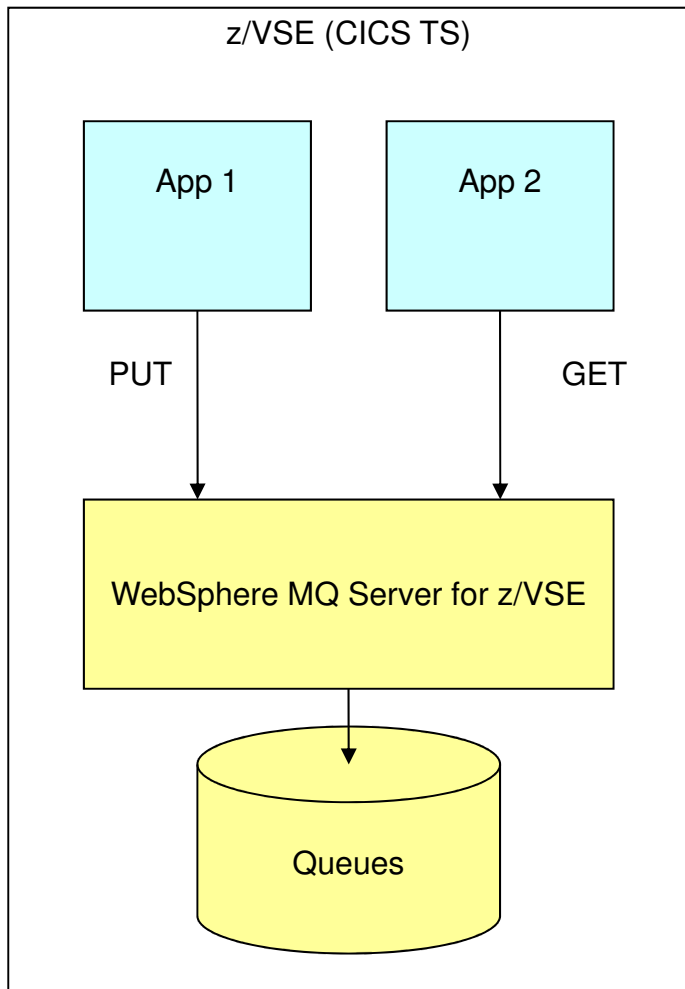
Support:

- Category 3 WebSphere MQ SupportPacs are supplied under the standard terms and conditions provided by the International Program License Agreement (IPLA) and thus, carry program defect service for WebSphere MQ customers.
- If you encounter what you believe to be a defect with the SupportPac, [and you have a current license for a WebSphere MQ server product](#), you may request Program Services by reporting the problem [via the same defect reporting channel](#) you employ for the [WebSphere MQ server product\(s\)](#) on which you are using the SupportPac.
- No proof of entitlement is required to use this SupportPac.

WebSphere MQ Server

vs.

WebSphere MQ Client



WebSphere MQ Client for VSE - Installation

Download file [MQC5.zip](#) from

http://www.ibm.com/support/docview.wss?rs=171&uid=swg24010051&loc=en_US&cs=utf-8&lang=en and unzip it.

Unzip the MQC5.zip file to get the following files:

- MQICLIB.JCL catalogs MQ Client objects and phases.
- MQICINC.JCL catalogs copybooks and include files
- MQICSMP.JCL catalogs example client application sources
- MQICBRG.JCL catalogs components of the MQ Client for VSE Bridge
- README.TXT please read this file carefully.
- LICENSES*.TXT

Upload the .JCL files to the z/VSE reader (or to z/VM) in BINARY and LRECL 80.

- These files catalog PHASEs, OBJs, include files, copy books and examples into a library of your choice.
- There is a PAUSE statement in the jobs that allow you to specify the destination sublibrary via a SETPARM statement.
- The default sublibrary is PRD2.MQICVSE. MQICLIB.JCL should be run as first, since it creates the destination sublibrary if it does not already exist.

WebSphere MQ Client for VSE - Installation (cont'd)

LIBDEF considerations:

- You should add the MQ Client installation sublibrary to the PHASE LIBDEF chain of your applications as well as CICS.
- When compiling MQ Client applications, you need to have the MQ Client installation sublibrary in the SOURCE LIBDEF in your compile jobs.
- You should have only the MQ Client sublibrary, but **not** the MQ Server sublibrary included.
 - This ensures that the include files and copybooks from the MQ client are used.
 - The declarations in the MQ Client include files and copybooks should be compatible with those from the MQ server.
 - The MQ client include files and copybooks contain some additional definitions needed for the MQ client only (e.g. structures for MQCONN call).

CICS Definitions:

- You must define the MQ Client program “MQICVSEP” to CICS, unless you have program autoinstall active:

```
DEFINE PROGRAM(MQICVSEP) GROUP(MQM) LANGUAGE(C)
EXECKEY(CICS) DATALOCATION(ANY)
```

- If you plan to use the MQ Client for VSE Bridge, then you should also perform CICS CSD file updates via member MQCICSD.Z.

Using the WebSphere MQ Client for VSE

- When a z/VSE application is using MQ Server on z/VSE, it does not need to know with which MQ Server it is working, since there is only one (local) MQ Server.
- When using **MQ Client**, then the application **must tell the MQ Client with which MQ Server it wants to work with**, and where the MQ Server is located.

There are 2 options how an application specifies the MQ Server it wants to work with:

1. Using **MQCONN** with **MQSERVER** environment variable:

- While the MQCONN call itself remains unchanged, the application must **set the MQSERVER environment variable prior to calling MQCONN**.
- Setting environment variables can be done via the **setenv()** call when using LE/C.
- COBOL and PL/1 applications must use the **MQSETENV** API call, which is provided by the MQ Client for VSE.
- Environment variables can also be set in the LE runtime default options (see CEEUOPT, CEEDOPT and CEECOPT).

2. Using **MQCONNX**:

- This requires changing the MQCONN call to an MQCONNX call and setup the MQCNO and MQCD data structures accordingly.

Re-link of the applications against MQ Client library

You must **re-link** the applications against the **MQ Client library (MQICVSE.OBJ)**.

- When using the MQ Client, you must **explicitly include the MQ Client object deck MQICVSE.OBJ** to your application.
- MQICVSE contains the entries of all MQI calls (e.g. MQGET, MQCONN, etc).
- Make sure that the MQ Client sublibrary is in the LIBDEF of your compile and link jobs.

In addition, applications using the MQ Client must be **prelinked** using the **LE prelinker**.

- When using the MQ Client, you must prelink all your applications, even when they are implemented in COBOL or PL/1.

```
// OPTION CATAL
  PHASE YOURPROG, *
  INCLUDE YOURPROG
  INCLUDE MQICVSE
/*
// EXEC EDCPRLK, SIZE=EDCPRLK, PARM='UPCASE MAP `
/*
// EXEC LNKEDT, SIZE=256K
/*
```


Additional application changes might be required

Application changes to handle **MQ Client specific errors**

- Because MQ client applications use a remote MQ Server, additional error situations can occur.
- Each MQ Client MQI call requires network communication through the MQ Client Channel over TCP/IP.
- Network connections may break or get interrupted for various reasons (firewalls idle timeouts, etc).

Application changes in regards of **data translation (ASCII/EBCDIC)**

- When using MQ Client, the applications directly work with an MQ Server that is located outside of z/VSE, and thus is mostly ASCII-based.
- Messages are not automatically converted, but the MQ Client application can request to convert the message with the MQGET call by using option **MQGMO-CONVERT**.
- This performs message conversion for known message formats.
- To specify the character set used by the MQ Client application use the environment variable **MQCCSID**.
- **Note:** When using an MQ Client, codepage conversion (ASCII/EBCDIC) is performed by the MQ Server.

MQ Server features not supported by MQ Client

Publish and Subscribe (Pub/Sub) is not supported

- WebSphere MQ Server for VSE supports Publish and Subscribe since APARs PM85374 and PM86869.
 - MQSUB Register subscription
 - MQSUBRQ Subscription request

→ These MQI calls are not available with the MQ Client on z/VSE.

Message property API calls are not supported

- WebSphere MQ Server for VSE supports Message Property API calls since APARs PM85374 and PM86869.
 - MQBUFMH Convert buffer into message handle
 - MQCRTMH Create message handle
 - MQDLTMH Delete message handle
 - MQDLTMP Delete message property
 - MQINQMP Inquire message property
 - MQMHBUF Convert message handle into buffer
 - MQSETMP Set message property

→ These MQI calls are not available with the MQ Client on z/VSE.

MQ Server features not supported by MQ Client (cont'd)

The WebSphere MQ Client for VSE does not support **SSL enabled channels**

- While the MQ Server for VSE supports SSL enabled channels, the MQ Client does not support SSL.
- Thus, all data is transported 'in clear' over the Server Channel to the MQ Server.

DB2 Q Capture requires an MQ Server environment on z/VSE

- The DB2 DataPropagator Q Capture Supplement for DB2 on VSE & VM requires an MQ Server environment on z/VSE to run the Q Capture program.
- Thus, the DB2 DataPropagator Q Capture feature can not be used in a MQ client only environment.

MQ CICS Bridge is not available with the MQ Client

- The WebSphere MQ for VSE Server function MQ CICS Bridge is not available with the MQ Client.

Triggering

Triggering is a must-have feature when asynchronous message passing is used.

- It allows to **start (trigger) an application** when one or more **messages arrive on a queue**.
- Per queue, you can define which trigger program is to be triggered.

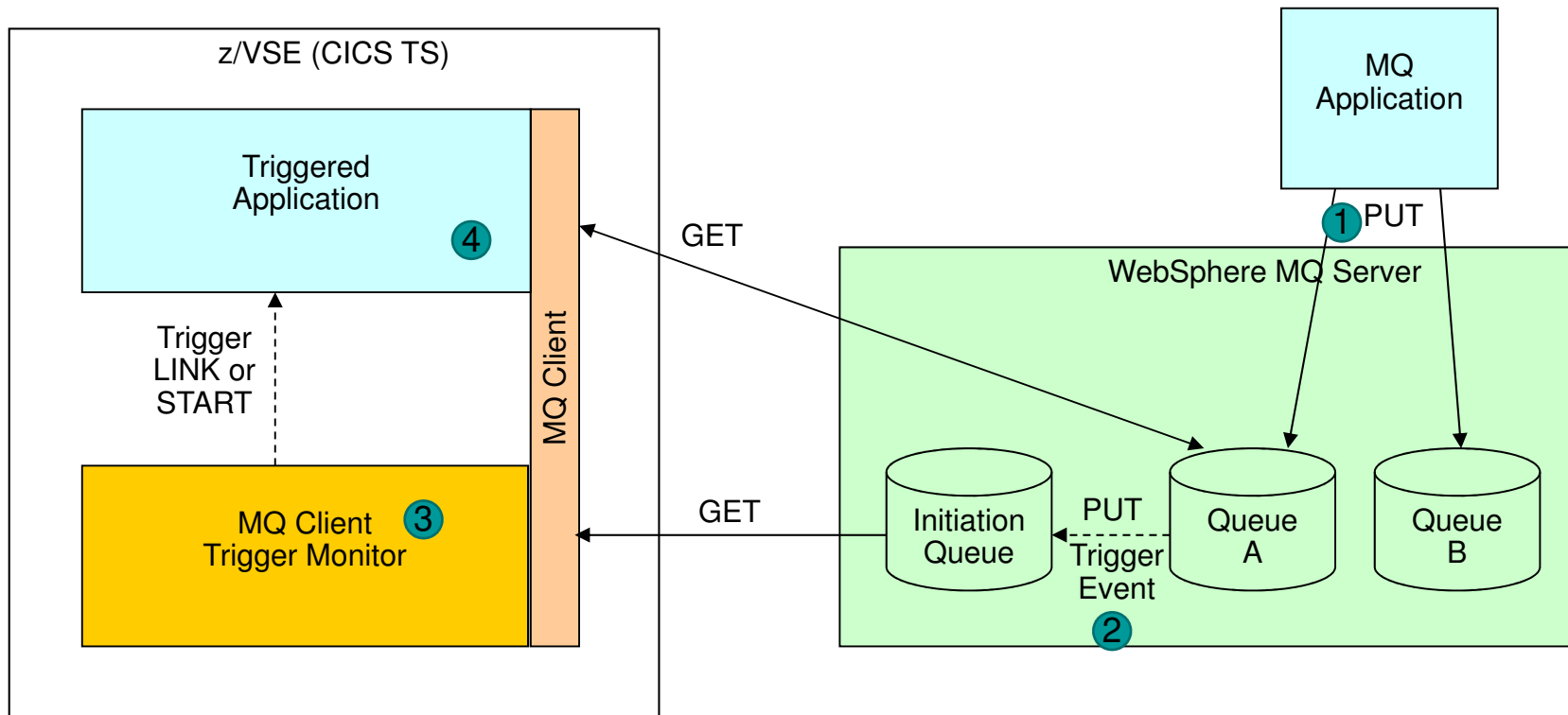
When invoked (triggered), the trigger program gets information about the trigger event it was invoked for.

- Besides others, the trigger event contains information about the queue that caused the trigger.
- The triggered program then performs MQGET operations to get the messages from the queue and processes them.

There is a nice description of how triggering works in the following blog entry “Triggering for beginners”:

https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/triggering_for_beginners?lang=en

The MQ Client Trigger Monitor



1. An MQ application puts a message onto queue A on a remote MQ server
2. Triggering is activated for Queue A, thus a trigger event is put onto the trigger initiation queue.
3. The MQ Client Trigger Monitor watches for trigger events on the initiation queue, gets the event and invokes (triggers) the associated z/VSE program
4. The triggered z/VSE application get the message associated to the trigger event from queue A and processes it

The MQ Client Trigger Monitor

The z/VSE MQ Client Trigger Monitor is available as follows:

- **z/VSE 6.1:** integrated
- **z/VSE 5.2:** APAR PI42615 / PTF UI28409
- **z/VSE 5.1:** APAR PI42612 / PTF UI28408

The user interface to the z/VSE MQ Client Trigger Monitor is provided by the **MQTM transaction**

- You can use MQTM from a terminal
- Or from the z/VSE console (via MSG F2,DATA=MQTM ...)

There can be multiple MQ Client Trigger Monitor instances active on a CICS TS system.

- Each instance is monitoring exactly one Initiation-Queue on a specific MQ Server and Queue manager.
- Instances can be started and stopped individually through the MQTM transaction.
- There is also an auto-start feature that will start pre-configured instances during CICS start-up (PLTSI).
- During CICS shutdown (PLTSD) all active instances will be stopped gracefully.

Starting and stopping Trigger Monitor instances

Start a Trigger Monitor instance:

```
MQTM START MQSERVER=server-channel-name/TCP/ip-or-hostname
           [,QMGR=queue-manager-name]
           [,INITQ=initiation-queue-name]
           [,RECONTIME=n]
           [,RECONTRIES=n]
           [,WAITTIME=n]
```

Stop a Trigger Monitor instance:

```
MQTM STOP MQSERVER=server-channel-name/TCP/ip-or-hostname
           [,QMGR=queue-manager-name]
           [,INITQ=initiation-queue-name]
```

Start all preconfigured Trigger Monitor instances defined in IESMQCTL.Z

```
MQTM STARTALL
```

Stop all currently active Trigger Monitor instances

```
MQTM STOPALL
```

List all currently active Trigger Monitor instances

```
MQTM SHOW
```

Trigger programs

The MQ Client Trigger Monitor invokes trigger programs the exact same way as WebSphere MQ for z/VSE V3.0.

- Below description is a repeat of what is documented in the WebSphere MQ for z/VSE V3.0 manuals.

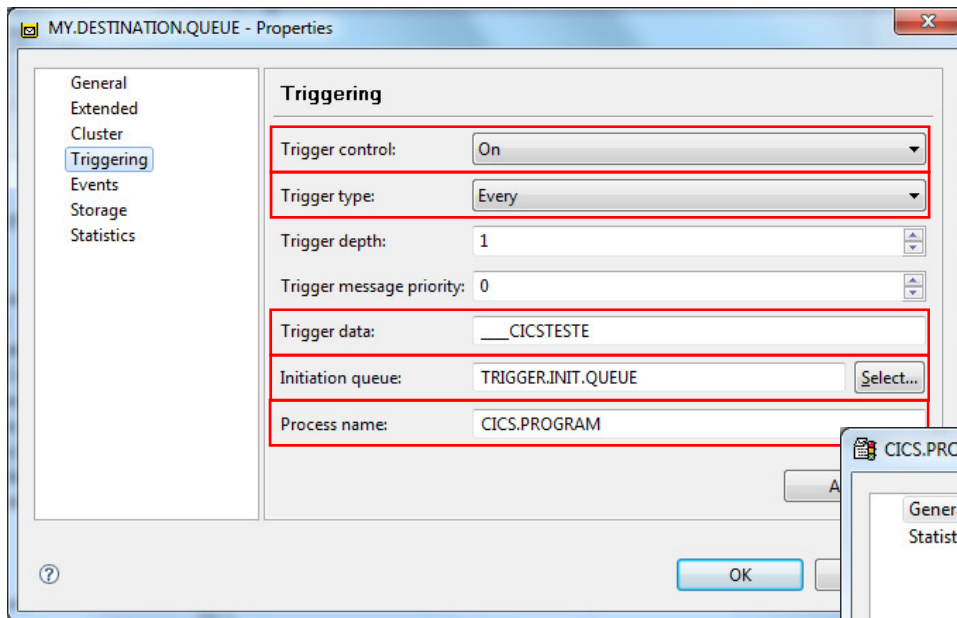
The program or transaction to be invoked as trigger is determined using the “Trigger Data” field of the Trigger Event message.

- The “Trigger Data” field is configured on the queue where the trigger is defined for.
- For triggers running on VSE, the “Trigger Data” is a 13 character field, consisting of
 - 4 characters transaction name (or blanks)
 - 8 characters program name (or blanks)
 - 1 character trigger-event flag

The trigger program is either started via EXEC CICS LINK when the program name is specified, or via EXEC CICS START starting the transaction, when the transaction name is specified.

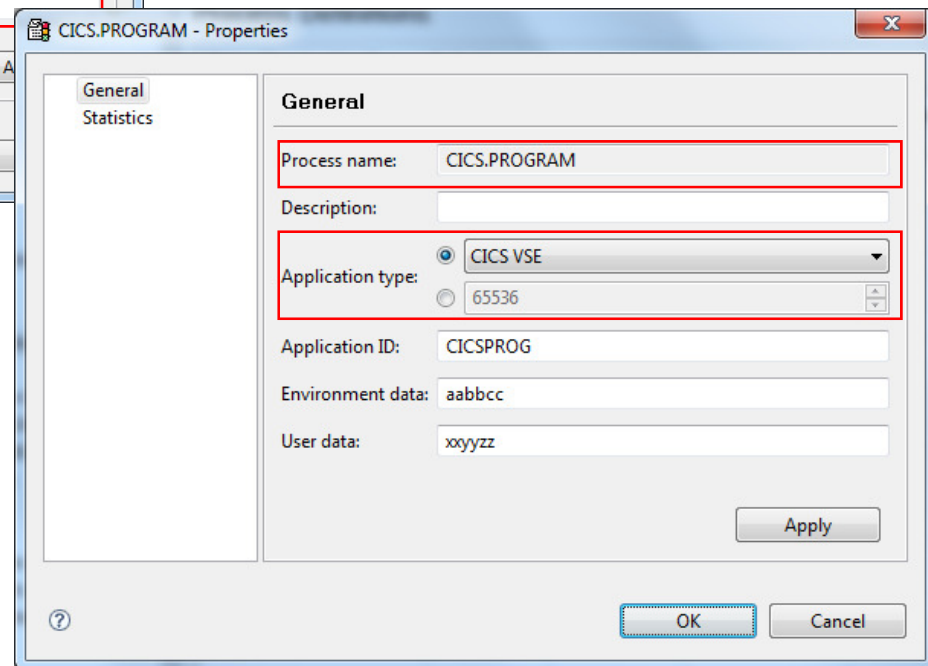
- The trigger even message (MQTM structure) is passed as COMMAREA or can be retrieved via EXEC CICS RETRIEVE, dependent on the start mode.

Setup triggering on an WebSphere MQ Server

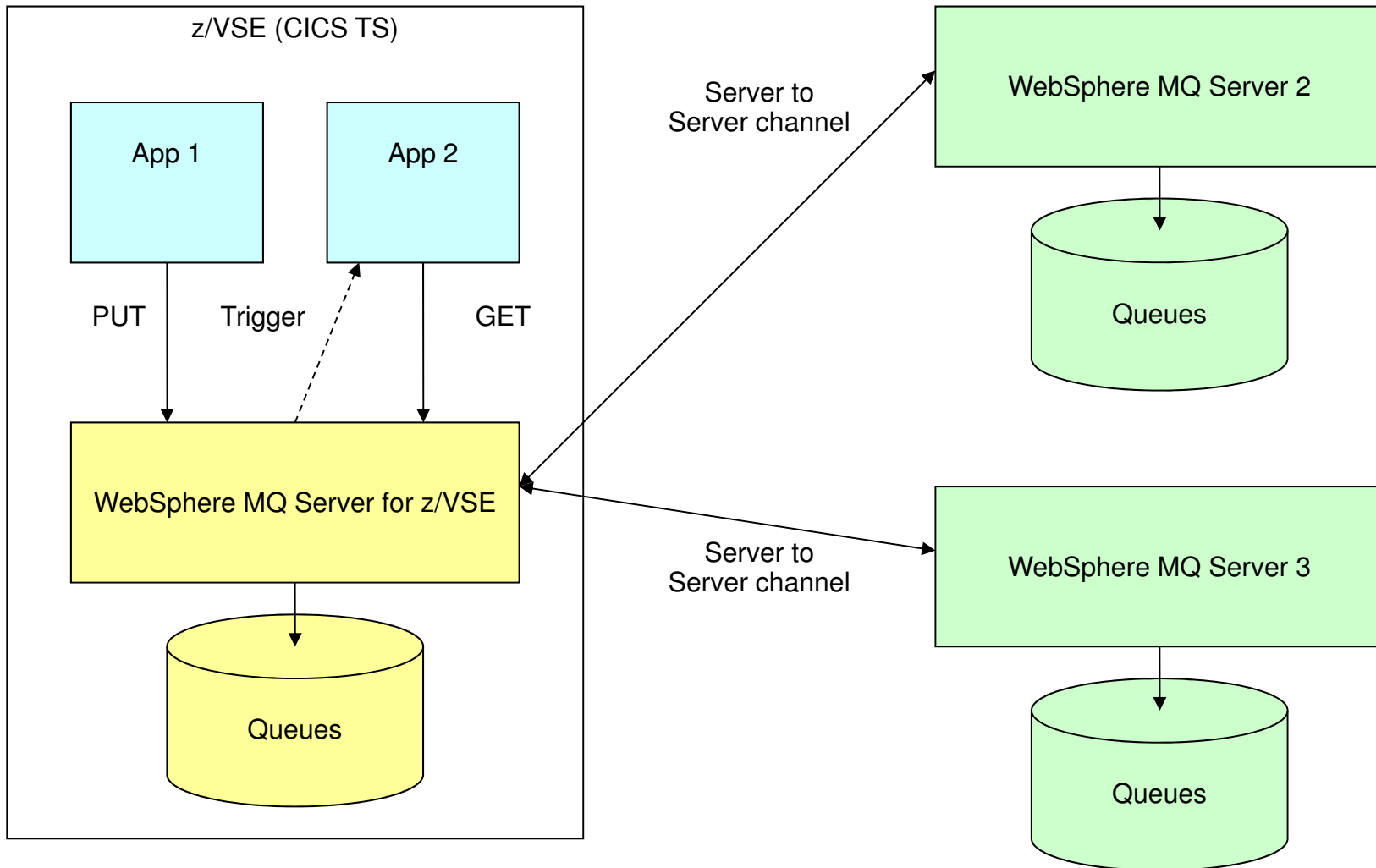


- This trigger configuration triggers on EVERY new message
- “Trigger Data” specifies an 8 character program name (‘CICSTEST’) and 1 character flag ‘E’ for EVERY.
- Process is ‘CICS.PROGRAM’

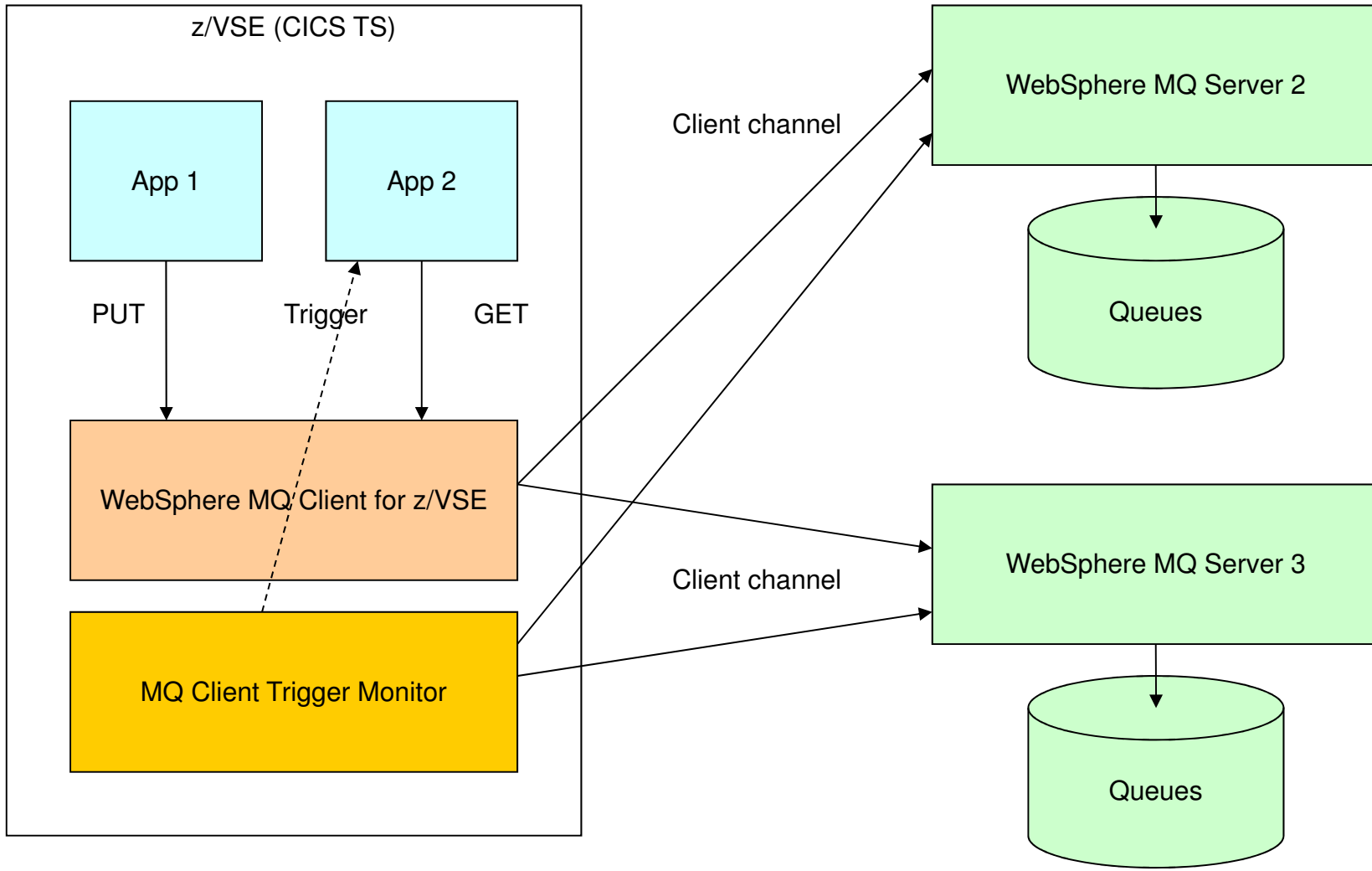
- Process definition ‘CICS.PROGRAM’ with an “Application type” of ‘CICS VSE’.
- The remaining fields are not used, but are passed to the triggered program within the MQTM structure.



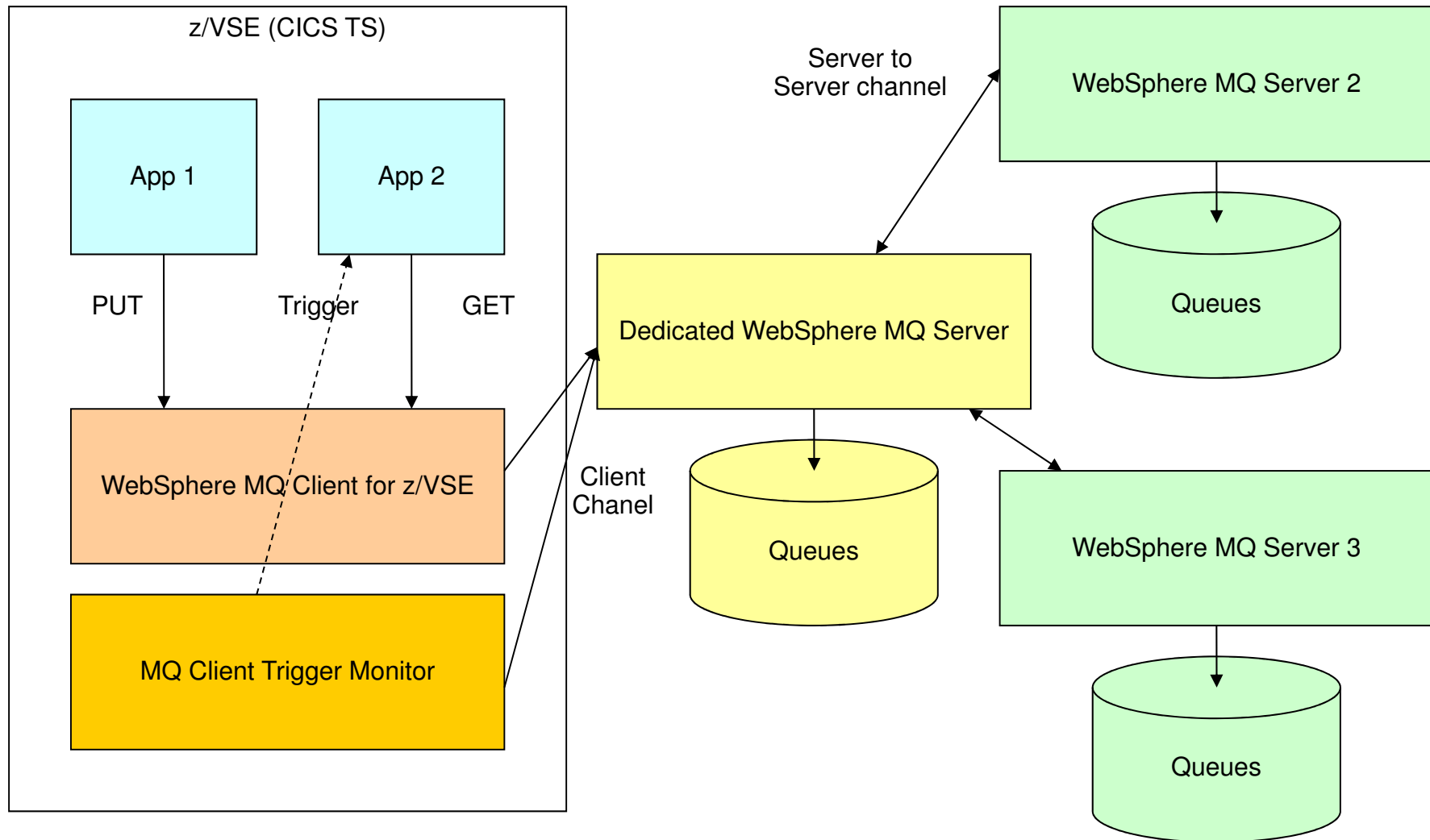
Typical WebSphere MQ Server scenario

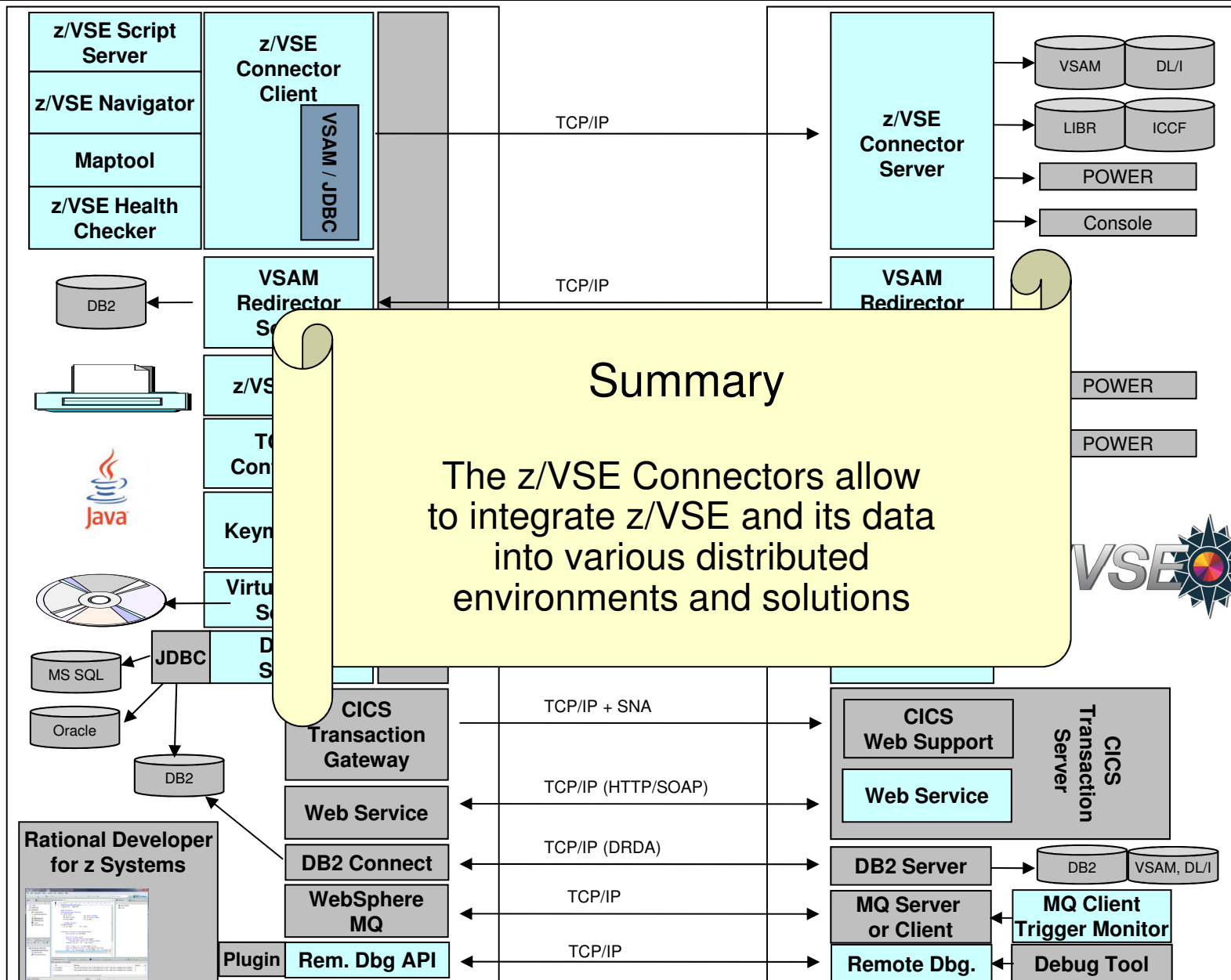


Option 1: Replace MQ Server with MQ Client directly



Option 2: Add a dedicated MQ Server for VSE applications





Questions ?



THANK YOU