



WebSphere Liberty Profile Administration and Management

With focus on IBM z/OS Connect
and IBM MQ Console and REST support

Mitch Johnson
mitchj@us.ibm.com
Washington Systems Center

mitchj@us.ibm.com



Notes and Disclaimers

- Additional information included in this presentation was distilled from experience implementing security using RACF with z/OS products like CICS, IMS, Db2, MQ, etc. as well as Java runtimes environments like WebSphere Application Server and WebSphere Application Server Liberty (commonly called Liberty).
- There will be additional information on slides that will be designated as Tech/Tips. These contain information that at perhaps at least interesting and hopefully, useful to the reader.
- A z/OS , or a Java , or a Liberty , or a z/OS Connect OpenAPI2 , or a z/OS Connect OpenAPI3 , or a CICS  or a MQ  icon will appear on slides where the information is specific to these products. Don't hesitate to ask questions as to why the icon does or does not appear on certain slides.
- **IBM z/OS Connect (OpenAPI 2)** refers the z/OS Connect EE product prior to service level V3.0.55. **IBM z/OS Connect (OpenAPI 3)** refers to the additional functions and features added with service level V3.0.55. Important - servers configured for OpenAPI 2 can will continue to operate as is with service level V3.0.55 and later.
- The examples, tips, etc. present in this material are based on firsthand experiences.

Agenda

- Review the OMVS and Liberty basics
 - The OMVS, Java and Liberty execution environments
 - Creating Liberty servers
 - Managing the server XML configuration
- Liberty Security
 - Security provided by the Angel process
 - Liberty authentication and authorization using basic, digital certificates and third-party tokens
- Security when accessing z/OS subsystems
- Security when accessing non-z/OS subsystems
- Useful Liberty features and MVS commands
- Managing and Monitoring Liberty servers
 - WLM configurations
 - SMF options
 - Monitoring OMVS processes
 - Connection pooling options
 - Above the bar storage
 - High availability options
- Where do I look when things go wrong?
 - Problem determination techniques
 - Understand the anatomy of messages
- Appendix - Sample administrative JCL

**Let's start by reviewing some of the basic
configuration details and options for
Java on z/OS, OMVS and Liberty**



Verifying that the Java and OMVS environments are properly configured*

Basic system configuration settings which have more than once, have caused issues

- Prevent out-of-memory or other storage issues:
 - Verify the Java runtime is not being limited by system parameters, e.g., *MAXASSIZE* (2 147 483 647), *MAXTHREADS*, etc., for details see *BPXPRM setting* at URL https://www.ibm.com/docs/en/sdk-java-technology/8?topic=SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/j9_configure_zos_bpxprm.html
 - Check the value of *ASSIZEMAX* in the OMVS segments of the identities involved and ensure it is adequate, see *MAXASSIZE* above.
 - Exclude OMVS from any IEFUSI exit, SUBSYS(OMVS,NOEXITS) in PARMLIB member *SMFRPMxx*.
 - Verify the JCL MEMLIMIT parameter value is within reason for your system.

CWWKB0125I: This server requested a REGION size of 0KB. The below-the-line storage limit is 8MB and the above-the-line storage limit is 1527MB.

CWWKB0126I: MEMLIMIT=1000. MEMLIMIT CONFIGURATION SOURCE=JCL.

- Start an OMVS shell session using the identity under which a server is running and verify that the required version of Java is fully **operational** and **current** by entering command **java -version**, you see should results like this:
- Verify that RACF identities associated with started tasks have OMVS segments with UIDs and GIDs and valid HOME directories and that the identities can invoke Java commands.

Tech-Tip - Check the Java version information using JCL

```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=JOHNSON,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1) ,USER=LIBSERV  
//*****  
//* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
//*****  
//* STEP JAVA - INVOKE THE java -version COMMAND  
//*****  
//JAVA EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//STDENV DD DUMMY  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
export JAVA_HOME=&JAVAHOME; +  
$JAVA_HOME/bin/java -version
```

```
java version "1.8.0_301"  
Java(TM) SE Runtime Environment (build 8.0.6.35 - pmz6480sr6fp35-20210714_01(SR6 FP35))  
IBM J9 VM (build 2.9, JRE 1.8.0 z/OS s390x-64-Bit Compressed References 20210622_7763 (JIT enabled, AOT  
enabled)  
OpenJ9 - b1f3adb  
OMR - c2f4a18  
IBM - c24a144)  
JCL - 20210625_01 based on Oracle jdk8u301-b09
```

Tech-Tip: Using the SAF SURROGAT resources for administration

RACF Surrogate access allows a designated administrative identity the ability to invoke commands and perform functions as if they were running under the identity that will be used for the z/OS Connect server started task. This may be useful because identities associated with started task are normally restricted and cannot be used for accessing TSO or OMVS shells,

Use the following examples as guides and create the surrogate resources and permit access. In these examples, ***LIBSERV*** represents the RACF identity under which the z/OS Connect server will be running and ***adminGrp*** represent the administrative RACF administrative group.

Define a SURROGAT profile for the server's SAF identity

RDEFINE SURROGAT BPX.SRV.*LIBSERV*

Define a SURROGAT submit profile to allow job submission as the server's SAF identity

RDEFINE SURROGAT *LIBSERV*.SUBMIT

Permit a member of the administrative group to act as a surrogate of the Liberty task identity

PERMIT BPX.SRV.*LIBSERV* CLASS(SURROGAT) ID(*adminGrp*) ACC(READ)

PERMIT *LIBSERV*.SUBMIT CLASS(SURROGAT) ID(*adminGrp*) ACC(READ)

Refresh the SURROGAT in storage profiles

SETROPTS RACLIST(SURROGAT) REFRESH

Now any identity in group *adminGrp* can submit JCL with the *USER=LIBSERV* parameter on the job card or use the OMVS switch user command (*su -s LIBSERV*) to execute OMVS scripts or commands as LIBSERV.

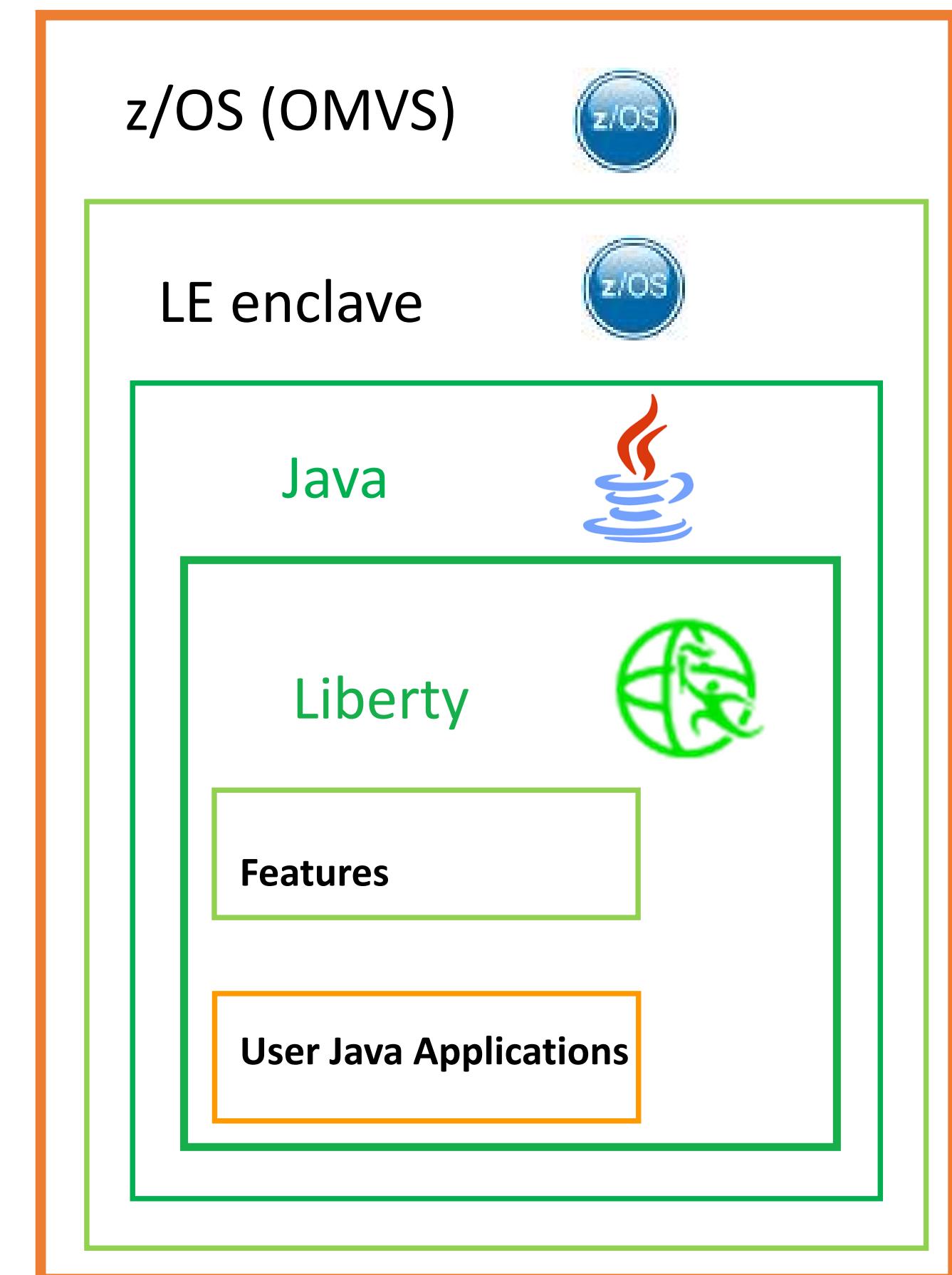


It helps to think of a Liberty server as consisting as a layers of software products

- Liberty servers are started by a command (either a script or binary executable) that initializes an OMVS process. A process which has been tailored for Java applications.
- Liberty runs in a Language Environment (LE) enclave configured to support OMVS and Java processes.
- Liberty is a Java application and on z/OS. There are Liberty features that provide access to z/OS services like SAF, WLM, RRS, SMF, JCL, etc. to other Java applications.
- The Liberty Java application provides an execution environment for multiple concurrent Java applications.

Knowing and understanding the different layers and their relationships is important regarding:

- Understanding which layer a configuration options, e.g., environment variables, Java directives, etc., applies.
- Monitoring and understanding the health of the server
- Performing problem determination and performance tuning





The *WLP_USER_DIR* environment variable

- The *WLP_USER_DIR* environment variable provides a directory location where:
 - A server's configuration files can be located
 - A default directory for writing logs and other files
 - A default directory where a server can locate application artifacts
- The same value for *WLP_USER_DIR* must be used for starting a server that was used when the server was created.
- There can be multiple “*WLP_USER_DIR*” directories on an LPAR
- Each server (*serverName*) will have a unique subdirectory in the location specified by *WLP_USER_DIR*.

```
 ${WLP_USER_DIR}
   /servers
     /serverName1
     /serverName2
     /serverName3
   |--shared/
     |--apps/
     |--config/
     |--resources/
```

- The location of the *serverName* directory is based on the concatenation of the value of the *WLP_USER_DIR* environment variable with the constant *servers*
- The *serverName* directory structure and its initial contents are created when the server is created
- *serverName* can be a mount point with a dedicated file system mounted at this mount point (see above). This can be used to isolate servers to dedicated file systems.
- The number, size and output location of messages.log and trace files in the *logs* directory can be controlled with the Liberty *<logging>* configuration element or the output location controlled by using the *com.ibm.ws.logging.log.directory* Java directive as a JVM options override, more on this later.
- #These directories maintain state information and it is a good practice is to add the --clean parameter to the server startup JCL, e.g., PARMS='*serverName* --clean', especially after service is applied.
- By default, the ‘owner’ of these directories and files is the identity that creates them.



Tech-Tip: Use JCL to create the *WLP_USER_DIR* directory and its subdirectories

Permission bit are set properly by taking advantage of RACF **SURROGAT** or **UNIXPRIV** resources

Example of using **UNIXPRIV** privileges

```
/MYSERVER JOB CLASS=A,REGION=0M,MSGCLASS=H,NOTIFY=&SYSUID
//*****SET SYMBOLS
//*****EXPORT EXPORT SYMLIST=(*)
// SET WLPUSER='/var/wlp'
// SET USER='LIBGRP'
// SET GROUP='LIBSERV'
//*****Step CREATE - Use the mkdir command to create direcotries
//*****CREATE EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export WLP_USER_DIR=&WLPUSER; +
mkdir -p &WLPUSER/shared/apps; +
mkdir -p &WLPUSER/shared/config; +
mkdir -p &WLPUSER/shared/resources; +
chown -R &USER:&GROUP $WLP_USER_DIR/shared
```

Example of using **SURROGAT** privileges

```
/MYSERVER JOB CLASS=A,REGION=0M,MSGCLASS=H,NOTIFY=&SYSUID,USER=LIBSERV
//*****SET SYMBOLS
//*****EXPORT EXPORT SYMLIST=(*)
// SET WLPUSER='/var/wlp'
//*****Step CREATE - Use the mkdir command to create direcotries
//*****CREATE EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export WLP_USER_DIR=&WLPUSER; +
mkdir -p &WLPUSER/shared/apps; +
mkdir -p &WLPUSER/shared/config; +
mkdir -p &WLPUSER/shared/resources
```

N.B. Multiple OMVS commands can be entered as input to *BPXBATCH* utility. Individual OMVS commands are terminated with a semi-colon. The semi-colon should be followed by at least one space. A plus symbol is used to indicate continuation. It is a good practice to not start a command in column 1.

Tech/Tip: Also consider using the SAF UNIXPRIV/FACILITY resources

An alternative to using a surrogate access is to permit the identity under which the customization will be done to enhanced Unix privileges. Specifically, permitting the identity to Unix privileges SUPERUSER.FILESYS, SUPERUSER.FILESYS.CHANGEPERMS and SUPERUSER.FILESYS.CHOWN.

- *Permit an administrative identity to write to any local directory or file*
PERMIT SUPERUSER.FILESYS CLASS(UNIXPRIV)
 ID(adminUser) ACC(CONTROL)
- *Permit an administrative identity to change permission bit of any local directory or file*
PERMIT SUPERUSER.FILESYS.CHANGEPERMS CLASS(UNIXPRIV)
 ID(adminUser) ACC(READ)
- *Permit an administrative identity to change the ownership of any directory or file*
PERMIT SUPERUSER.FILESYS.CHOWN CLASS(UNIXPRIV)
 ID(adminUser) ACC(READ)
- *Permit an administrative identity switch to root (su -s root) or the Enable superuser mode(SU) Setup option in ISHELL*
PERMIT BPX.SUPERUSER CLASS(FACILITY) ID(adminUser) ACC(READ)
- *Refresh the UNIXPRIV and/or FACILITY instorage profiles*
SETROPTS RACLIST(UNIXPRIV,FACILITY) REFRESH

https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.bpxb200/usspriv.htm

Use the power of these commands provide carefully and only when necessary



Basic Liberty servers are created by invoking a Liberty command *server create*

Invoking the command creates the server's required directory structure and populates key configuration files.

To create a basic Liberty server, use the Liberty *server create* command, as in `server create serverName`

- Where `serverName` is any value you wish, such as `wlpopsrv` or `wlpOpenIDAuthServer` and this value will be the name of the server instance. The default value for the serve name is `defaultServer`
- Environment variable `WLP_USER_DIR` must be set to determine the location of the configuration directory and files created by this command. The constant `servers` is appended to the value of this variable, e.g., `{$WLP_USER_DIR}/servers` and the server's name is appended to this root directory and full directory path is the location where the server's configuration files, and default directories are created, e.g., `{$WLP_USER_DIR}/servers/serverName`. The `WLP_USER_NAME` variables is required when starting a server and must be the same value used when the server was created. The default value for a Liberty server is `../wlp/usr`.

N.B. The name of the server does not have to be same as the started task name, as shown in this example (note in this example how the value for `WLP_USER_DIR` is provided by the `PATH` attribute of the `WLPUDIR` DD statement):

```
//WLPOPID PROC PARMS='wlpOpenIDAuthServer'  
///*  
// SET INSTDIR='/usr/lpp/liberty_zos/22.0.0.9'  
// SET USERDIR='/var/wlp'  
///*  
//STEP1 EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,  
// PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv &PARMS'  
//WLPUDIR DD PATH='&USERDIR.'  
//STDOUT DD SYSOUT=*  
//STDERR DD SYSOUT=*  
//MSGLOG DD SYSOUT=*  
//STDENV DD PATH='/etc/system.env', PATHOPTS=(ORDONLY)
```

Tech-Tip: Let's do a quick review of permissions bits

Owner	Group	Other																																				
<table><thead><tr><th>Bit</th><th>Read</th><th>Write</th><th>Execute</th></tr></thead><tbody><tr><td></td><td>1</td><td>1</td><td>1</td></tr><tr><td>Base-2 Value</td><td>[4]</td><td>[2]</td><td>[1]</td></tr></tbody></table> <p>7 The owner has READ, WRITE and EXECUTE</p>  <p>The owner of the file or directory</p> <p>chmod -R * u+rwx zceesrv1</p>	Bit	Read	Write	Execute		1	1	1	Base-2 Value	[4]	[2]	[1]	<table><thead><tr><th>Bit</th><th>Read</th><th>Write</th><th>Execute</th></tr></thead><tbody><tr><td></td><td>1</td><td>0</td><td>1</td></tr><tr><td>Base-2 Value</td><td>[4]</td><td>[2]</td><td>[1]</td></tr></tbody></table> <p>5 The group has READ and EXECUTE, but not WRITE</p>  <p>IDs that are part of the group for the file or directory</p> <p>chmod g+rwx server.xml</p>	Bit	Read	Write	Execute		1	0	1	Base-2 Value	[4]	[2]	[1]	<table><thead><tr><th>Bit</th><th>Read</th><th>Write</th><th>Execute</th></tr></thead><tbody><tr><td></td><td>0</td><td>0</td><td>0</td></tr><tr><td>Base-2 Value</td><td>[4]</td><td>[2]</td><td>[1]</td></tr></tbody></table> <p>0 Others have no access</p>  <p>IDs that are not the owner and not part of the group; that is, other</p> <p>chmod -R * o+rx resources</p> <p>chmod -R * o-w resources/security</p>	Bit	Read	Write	Execute		0	0	0	Base-2 Value	[4]	[2]	[1]
Bit	Read	Write	Execute																																			
	1	1	1																																			
Base-2 Value	[4]	[2]	[1]																																			
Bit	Read	Write	Execute																																			
	1	0	1																																			
Base-2 Value	[4]	[2]	[1]																																			
Bit	Read	Write	Execute																																			
	0	0	0																																			
Base-2 Value	[4]	[2]	[1]																																			
* indicates recursion																																						



Details of a Liberty servers' configuration directories and files



```
export WLP_USER_DIR=/var/zosconnect
```

```
$WLP_USER_DIR
|--servers/
|   |--server1_name1/
|   |   |--bootstrap.properties
|   |   |--jvm.options
|   |   |--server.env
|   |   |--server.xml
|   |   |--apps/
|   |   |--dropins/
|   |   |--configDropins/
|   |       |--defaults/
|   |       |--overrides/
|   |--logs/
|       |--messages.log
|   |--resources/
|       |--security/
|--server2_name/
|   ...
|--server3_name/
|   ...
|--shared/
|   |--apps/
|   |--config/
|   |--resources/
```

Permission	Bits	Owner	Group
750	LIBSERV	LIBGRP	
750	LIBSERV	LIBGRP	
750	LIBSERV	LIBGRP	
700	LIBSERV	LIBGRP	
700	LIBSERV	LIBGRP	
640	LIBSERV	LIBGRP	
640	LIBSERV	LIBGRP	
750	LIBSERV	LIBGRP	
666	LIBSERV	LIBGRP	
750	LIBSERV	LIBGRP	

The create command will create the directories and files under the <WLP_USER_DIR> and assign ownership based on the ID and Group that created the server

There are some questions that need answers with this in a production setting:

- If you have multiple people with a need to change configuration files, do you share the password of LIBSERV?
No. In fact, LIBSERV should be a PROTECTED identity with no password in the first place. Better to take advantage SAF SURROGAT so permitted users can switch to the owning ID so they can make changes.
- If you have multiple people with a need to read or update configuration files, do you simply connect them to LIBGRP?
No. The owner group may be granted access to other resources (on z/OS SAF profiles notably: SERVER) and you do not want others inheriting that. Better to make the configuration group be something different from the owner group and grant READ/WRITE through that group.
- The *shared* directory structure is shared among all servers started that have a common value for the *WLP_USER_DIR* environment variable. Each server can access common server configuration files using the *shared.config.dir* environment variable or access web applications using the *shared.app.dir* environment variable.



Tech-Tip: Use JCL to create a Liberty server by invoking the *server* command using IKJEFT01

```
//*****  
//* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
// SET WLPPATH='/usr/lpp/liberty_zos/22.0.0.9'  
// SET SERVER='myServer'  
// SET TEMPLATE='defaultServer'  
// SET WLPUSER='/var/wlp'  
// SET USER='LIBSERV'  
// SET GROUP='LIBGRP'  
//*****  
//* Step CREATE - Use the server command to create a server  
//*****  
//CREATE EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
export JAVA_HOME=&JAVAHOME; +  
export WLX_USER_DIR=&WLPUSER; +  
&WLPPATH/bin/server create &SERVER +  
--template=&TEMPLATE; +  
chown -R &USER:&GROUP $WLP_USER_DIR/servers/&SERVER
```

- Set and export JCL symbols variables

- Export the JCL symbols passed to OMVS environment
- Invoke the *server* command to create the Liberty server
- Use the chown command to set the correct owner/group settings



Creating a Liberty server involves creating the server's required OMVS directory structure and an initial server configuration file, e.g., *server.xml*

The server's directory structure will be located at `${WLP_USER_DIR} /servers /serverName` and the *server.xml* file is key configuration file for any Liberty server.

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.3</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host
attribute to the following element, e.g. host="*" -->
    <httpEndpoint id="defaultHttpEndpoint"
                  httpPort="9080"
                  httpsPort="9443" />

    <!-- Automatically expand WAR files and EAR files -->
    <applicationManager autoExpand="true"/>

</server>
```

An administrator adds or removes features (security, and other required functions) as needed in the *featureManager* configuration elements.

Configures connectivity in *httpEndpoint* configuration elements

Adds other configuration elements as needed.

A sample initial *server.xml* configuration file



And MQ Console/REST servers are created by invoking the MQ *crtmqweb* command

```
//*****  
///* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
// SET MQPATH='/usr/lpp/mqm/V9R3M4/web'  
// SET WLPUSER='/var/mqm/V9R3M4'  
// SET USER='MQSERV'  
// SET GROUP='MQGRP'  
//*****  
///* Step CRTMQWEB - Use the crtmqweb command  
//*****  
//CRTMQWEB EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
export JAVA_HOME=&JAVAHOME; +  
export WLP_USER_DIR=&WLPUSER; +  
&MQPATH/bin/crtmqweb &WLPUSER -p MQ; +  
chown -R &USER:&GROUP $WLP_USER_DIR/servers/&SERVER
```

- Set JCL symbols

- Export the JCL symbols passed to the OMVS environment
- Invoke the *crtmqweb* command to create the MQ Console and REST server. The default value for a MQ Console server is */var/mqm/web/installation1*
- Use the chown command to set the correct owner/group settings

While z/OS Connect servers are created using the z/OS Connect `zosconnect` command



To create a z/OS Connect server, use the `zosconnect` command using one of these templates, as in:

`zosconnect create serverName --template=templateName`

Where *templateName* can be:

- **`zosconnect:apiRequester`** for an OpenAPI2 z/OS Connect API requester enabled server
- **`zosconnect:default`** template for base OpenAPI2 z/OS Connect servers
- **`zosconnect:openApi3`** template for base OpenAPI3 z/OS Connect native servers
- **`zosconnect:openApi3Requester`** template for base OpenAPI3 z/OS Connect native API requester servers

- Where *serverName* is any value you wish, such as `zceesrvr` or `zCEEServer`, and this value will be the name of the server instance. The templates can be found in directory `/usr/lpp/IBM/zosconnect/v3r0/runtime/templates/servers`.
- Environment variable **`WLP_USER_DIR`** will be used to set the location of the configuration directory and files created by this command, default location is `/var/zosconnect/servers` where `/var/zosconnect` is default value for `WLP_USER_DIR` for z/OS Connect.
- The `zosconnect:openApi3` template installs feature **`zosconnect:zosConnect-3.0`**. z/OS Connect service provider features, e.g., `zosconnect:cics-1.0`, `zosconnect:mqService-1.0`, `zosconnect:dbService-1.0` and `imsmobile:imsmobile-2.0` have dependencies on feature **`zosconnect:zosConnect-2.0`** and are not compatible with feature **`zosconnect:zosConnect-3.0`**.
- The `zosconnect:openApi3Requester` template installs feature **`zosconnect:oasRequester-1.0`**.
- See URL <https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=reference-configuration-elements> for a current list of OpenAPI2 configuration elements and URL <https://www.ibm.com/docs/en/zos-connect/zos-connect/3.0?topic=reference-configuration-elements> for the current list of OpenAPI3 configuration elements.



Tech-Tip: Use JCL to make the creation and configuration of servers repeatable and portable

And take advantage of RACF SURROGAT and UNIXPRIV resources

Example of using **SURROGAT** privileges

```
//ZCEESRVR JOB CLASS=A,REGION=0M,NOTIFY=&SYSUID,USER=LIBSERV
//*****
//* SET SYMBOLS
//*****
//EXPORT EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
// SET ZEEPATH='/usr/lpp/IBM/zosconnect/v3r0'
// SET SERVER='zceesrvr'
// SET TEMPLATE='zosconnect:default'
// SET WLPUSER='/var/ats/zosconnect'
// SET USER='ATSSERV'
// SET GROUP='ATSGRP'
//*****
//* Step ZCEESRVR - Use the zosconnect command to create a server
//*****
//ZCEESRVR EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export JAVA_HOME=&JAVAHOME; +
export WLP_USER_DIR=&WLPUSER; +
&ZEEPATH/bin/zosconnect create &SERVER +
--template=&TEMPLATE; +
```

Example of using **UNIXPRIV** privileges

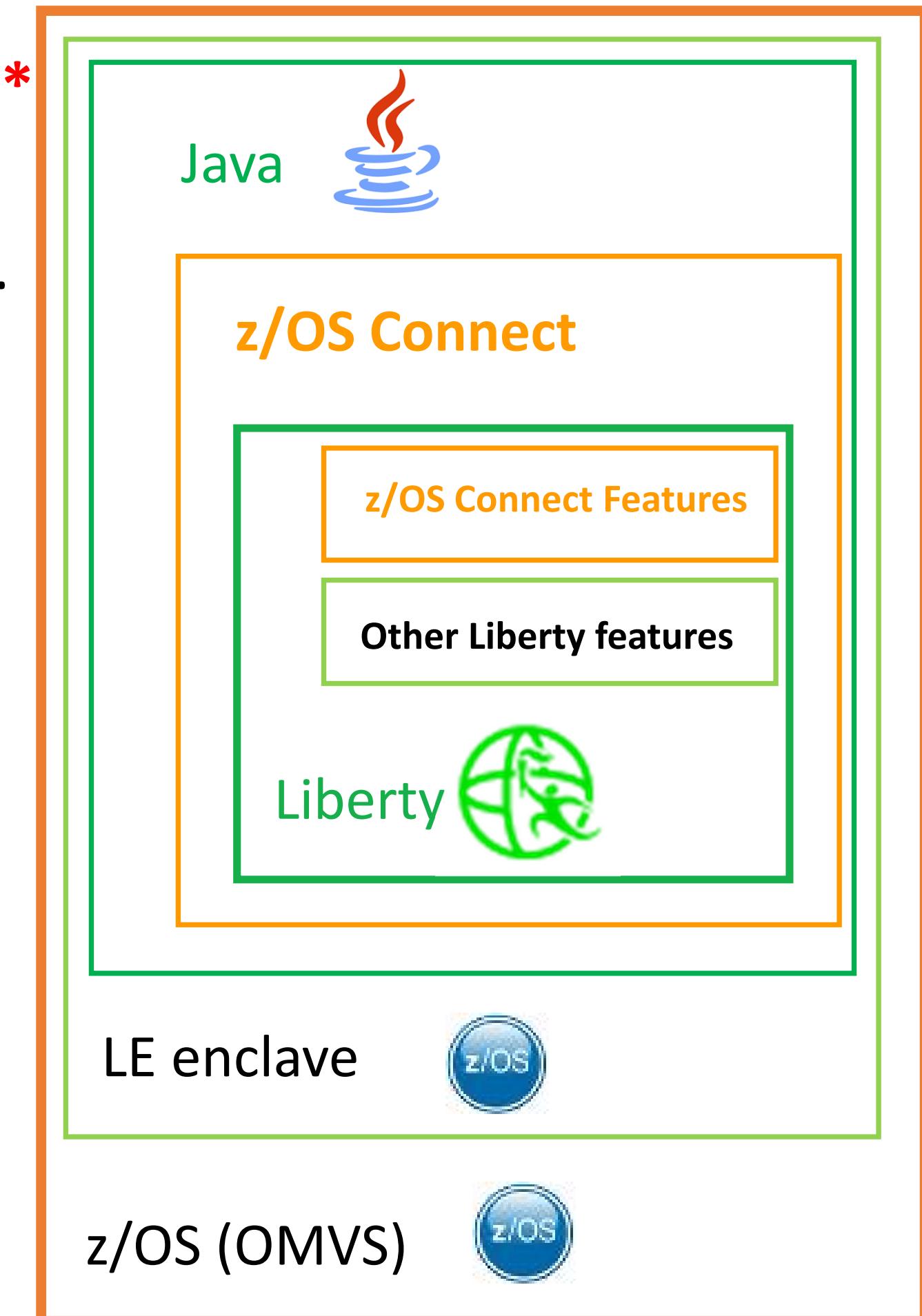
```
//ZCEESRVR JOB CLASS=A,REGION=0M,NOTIFY=&SYSUID
//*****
//* SET SYMBOLS
//*****
//EXPORT EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
// SET ZEEPATH='/usr/lpp/IBM/zosconnect/v3r0'
// SET SERVER='openApi3'
// SET TEMPLATE='zosconnect:openApi3'
// SET WLPUSER='/var/ats/zosconnect'
// SET USER='ATSSERV'
// SET GROUP='ATSGRP'
//*****
//* Step ZCEEAPI3 - Use the zosconnect command to create a server
//*****
//ZCEEAPI3 EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
export JAVA_HOME=&JAVAHOME; +
export WLP_USER_DIR=&WLPUSER; +
&ZEEPATH/bin/zosconnect create &SERVER +
--template=&TEMPLATE; +
chown -R &USER:&GROUP $WLP_USER_DIR/servers/&SERVER
```



Adding z/OS Connect to a Liberty server adds an additional software layer

z/OS Connect is both a Java application and a set of Liberty features

- The z/OS Connect Java application provides code that spawns a Liberty process *
- The z/OS Connect Liberty features provides a REST interface to user application artifacts that access to common z/OS subsystems, e.g., CICS, Db2, IMS, MQ, etc. and external REST endpoints.



* z/OS Connect starts a Liberty process using a system programming interface (SPI). See the Note regarding environment variables and *jvm.options* and *server.env* files at URL <https://www.ibm.com/docs/en/was-liberty/zos?topic=liberty-embedding-server-in-your-applications> regarding restrictions in this environment.



Differences between z/OS Connect OpenAPI2 and OpenAPI3 server.xml files

```
default template - OpenAPI 2 server.xml configuration file
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:zosConnect-2.0</feature>
        <feature>zosconnect:zosConnectCommands-1.0</feature>
        <feature>apiDiscovery-1.0</feature> *
    </featureManager>

    <!-- To access this server from a remote client add a host attribute
    <httpEndpoint id="defaultHttpEndpoint"
        host="*"
        httpPort="9080"
        httpsPort="9443" />
    <!-- add cors to allow cross origin access, e.g. when using swagger UI
    to fetch swagger doc from zOS Connect Enterprise Edition -->
    <cors id="defaultCORSConfig"
    - - - - - 24 Line(s) not Displayed

    <!-- config requires updateTrigger="mbean" for REFRESH command support
-->
<config updateTrigger="mbean" monitorInterval="500"/>

    <zosconnect_zosConnectManager setUTF8ResponseEncoding="true"/>

    <!-- zosConnect APIs -->
    <zosconnect_zosConnectAPIs updateTrigger="disabled" pollingRate="5s"
        <!-- zosConnect Services -->
    <zosconnect_services updateTrigger="disabled" pollingRate="5s"/>

    <!-- applicationMonitor is not applicable for z/OS Connect EE servers --
->
    <applicationMonitor updateTrigger="disabled" dropinsEnabled="false"/>

</server>
```

* Include these features if not already present.

```
openApi3 template - OpenAPI 3 server.xml configuration file
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:zosConnect-3.0</feature>
        <feature>openapi-3.0</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute
    <httpEndpoint id="defaultHttpEndpoint"
        host="*"
        httpPort="9080"
        httpsPort="9443" />
    - - - - - 12 Line(s) not Displayed
    <!-- config requires updateTrigger="mbean" for REFRESH command support
    config updateTrigger="mbean"/>

    <!-- applicationMonitor requires updateTrigger="mbean" for REFRESH command
    support -->
    <applicationMonitor updateTrigger="mbean" dropinsEnabled="false"/>

    <!-- Automatic expansion of WAR files is required for z/OS Connect native
    servers running the zosConnect-3.0 feature -->
    <applicationManager autoExpand="true" />

    <!-- APIs are deployed as WAR files and a webApplication element must be
    used to specify the location of the API WAR and optionally the name of the API
    -->
    <webApplication id="My API" location="${server.config.dir}/apps/api.war"
        name="MyAPI"/>

</server>
```

Note there are no *zosconnect* or *cors* configuration elements present with Open API 3.



Contrast a Liberty JCL procedure versus a z/OS Connect JCL procedure

```
//ZCEESRVR PROC PARMs='serverName'  
///*  
// SET ZCONHOME='/usr/lpp/IBM/zosconnect/v3r0'  
// SET INSTDIR='/usr/lpp/liberty_zos/21.0.0.9'  
///*  
//ZCON      EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=8G,  
// PARM='PGM &ZCONHOME./bin/zosconnect run &PARMS. --clean'  
// PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv &PARMS'  
//STDOUT    DD   SYSOUT=*  
//STDERR    DD   SYSOUT=*  
//STDIN     DD   DUMMY  
//MSGLOG   DD   SYSOUT=*  
//STDENV   DD   *  
BPX_SHAREAS=YES  
CEE_RUNOPTS=HEAPPOOLS (ON) ,HEAPPOOLS64 (ON)  
JAVA_HOME=/usr/lpp/java/J8.0_64  
WLP_USER_DIR=/var/zosconnect  
JVM_OPTIONS=-Dcom.ibm.ws.zos.core.angelName=zCEEAngel -Xmx512m  
OPENJ9_JAVA_OPTIONS=-Xoptionsfile=/var/zcee/properties/myServer.property
```

OMVS
LE
JAVA
LIBERTY
z/OS Connect



Tech-Tip: Displaying Liberty messages on the console and/or STDERR spool

server.xml

```
<zosLogging wtoMessage=
  "BAQR0657E,BAQR0658E,BAQR0660E,BAQR0686E,BAQR0687E"
  hardCopyMessage=
  "BAQR0657E,BAQR0658E,BAQR0660E,BAQR0686E,BAQR0687E"/>
```

MVS Console

```
18.12.02 STC00137 +BAQR0686E: Program CSCVINC is not available in the CICS region with
  811           connection ID cscvinc; service cscvincService failed.
18.12.02 STC00137 +BAQR0686E: Program CSCVINC is not available in the CICS region with
  812           connection ID cscvinc; service cscvincService failed.
19.07.12 STC00137 +BAQR0657E: Transaction abend MIJO occurred in CICS while using
  745           connection cscvinc and service cscvincService.
```

STDERR

```
ÝERROR  .. BAQR0686E: Program CSCVINC is not available in the CICS region with connection cscvinc and service cscvincService.
ÝERROR  .. BAQR0686E: Program CSCVINC is not available in the CICS region with connection cscvinc and service cscvincService.
ÝERROR  .. BAQR0657E: Transaction abend MIJO occurred in CICS while using CICS connection cscvinc and service cscvincService.
```



Liberty JCL parameters and server XML environment variables

- **serverName** – This JCL parameter is used to provides the name of the server and used to set the value of several environment variables.
- **WLP_USER_DIR** – This environment variable is used when a server is created to determine where the server's working directories will be created and where the initial *server.xml* file will be created. This variable is also used by the runtime environment to locate the server's existing working directories and the *server.xml* file. Also, the WLP_USER_DIR is used to set the shared variables.
- **JAVA_HOME** – The OMVS directory where the Java executables (*/bin* directory) can be located.
- **JVM_OPTIONS** – A z/OS Connect environment variables that provides Java options and/or system properties. The contents of **JVM_OPTIONS** is added to the *java* command line in the *zosconnect* startup script.
- **IBM_JAVA_OPTIONS** – An IBM JAVA environment variable (deprecated and eventually will be replaced by environment variable *OPENJ9_JAVA_OPTIONS*). Environment variable *IBM_JAVA_OPTIONS* variable can be used to provide Java options and/or system properties.
- **OPENJ9_JAVA_OPTIONS** – An OpenJ9 environment variable (eventually will replace the deprecated environment variable *IBM_JAVA_OPTIONS*). Environment variable *OPENJ9_JAVA_OPTIONS* variable can be used to provide Java options and/or system properties.

Note: Any Java option or system property using **JVM_OPTIONS** supersedes the same Java non-standard options or system property when provided by **IBM_JAVA_OPTIONS** or **OPENJ9_JAVA_OPTIONS**

The following environment variables are automatically set in a Liberty server and can be used as variables in the server XML configuration files.

- **server.config.dir** – whose value will automatically be set to the value of variable *WLP_USER_DIR* concatenated with the name of the server, e.g.
/var/zosconnect/servers/serverName
- **shared.config.dir** – whose value will automatically be set to the value of variable *WLP_USER_DIR* concatenated with /shared/config, e.g.
/var/zosconnect/shared/config
- **shared.app.dir** – whose value will automatically be set to the value of variable *WLP_USER_DIR* concatenated with /shared/apps, e.g.
/var/zosconnect/shared/apps
- **wlp.server.name** - whose value will automatically be set to the value of the server as provided in the PARMS value provided in the JCL procedure.



Tech-Tip: The `zconsetup` script must be invoked once per LPAR per product install path

The `zconsetup` script creates a symbolic link from the z/OS Connect's WLP `..v3r0/wlp/etc` directory in the product path (normally R/O) to a local R/W directory (creating a default configuration and local extension directories). This provides a means for administrator to configure the image.

```
JOHNSON:/usr/lpp/IBM/zosconnect/v3r0/wlp/etc: ls -al
total 32
drwxrwxr-x  2 OMVSKERN 0          8192 Jun 24 10:24 .
drwxrwxr-x 10 OMVSKERN 0          8192 Jun 24 10:24 ..
lrwxrwxrwx  1 990023 0          31 Jul 27 2020 extensions -> /var/zosconnect/v3r0/extensions
```

```
/var/zosconnect
  /servers
  /v3r0
    /extensions
      imsmonkey.properties
      zosconnect.properties
      filemanager.properties
      omegamon.properties
```

```
com.ibm.websphere.productId=com.ibm.ims.mobile
com.ibm.websphere.productInstall=imsmobile/wlp-ext relative path

com.ibm.websphere.productId=com.ibm.zosconnect
com.ibm.websphere.productInstall=runtime

com.ibm.websphere.productId=omegamonRequestMonitor-2.0
com.ibm.websphere.productInstall=/var/zosconnect/omegamon absolute path
```

- This directory structure and contents is created by invoking the `zconsetup` script and **must be created on each LPAR** on which z/OS Connect will execute. This is how the z/OS Connect Liberty server locates service provider executables. Note: the `com.ibm.websphere.productInstall` directive value that is **relative** to directory `/usr/lpp/IBM/zosconnect/v3r0`.
- Not creating this link will cause messages `CWWKF0001E: A feature definition could not be found for zosconnect:....` or `CWWKE0054E: Unable to open /usr/lpp/IBM/zosconnect/v3r0/wlp/etc/extensions/zosconnect.properties`

Tech-Tip: Verify the `zconsetup` script has been executed. My recommendation is to execute this script in the SMP/E target environment, otherwise it will be lost when service is applied and propagated to other images.



Tech-Tip: Executing the z/OS Connect zconsetup script using JCL

```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=JOHNSON,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)  
//*****  
//* Set symbols  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
// SET ZCEEPATH='/usr/lpp/IBM/zosconnect/v3r0'  
//*****  
//* Step ZCSETUP - Invoke the zconsetup script  
//*****  
//ZCSETUP EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
  export JAVA_HOME=&JAVAHOME; +  
  &ZCEEPATH/bin/zconsetup install
```



A z/OS Connect Liberty server's configuration directories and files



ID=**LIBSERV**
Group=**LIBGRP**

```
export WLP_USER_DIR=/var/zosconnect
$WLP_USER_DIR
  |--servers/
    |  |--server1_name1/
    |    |  |--bootstrap.properties
    |    |  |--jvm.options
    |    |  |--server.env
    |    |  |--server.xml
    |    |  |--apps/
    |    |  |--dropins/
    |    |  |--configDropins/
    |    |    |--defaults/
    |    |    |--overrides/
    |    |  |--logs/
    |    |    |--messages.log
    |    |  |--resources/
    |    |    |--security/
    |    |      |--zosconnect/
    |    |        |--apis/
    |    |        |--apiRequesters/
    |    |        |--rules/
    |    |        |--services/
    |    |--server2_name/
    |    . .
    |    |--server3_name/
    |    . .
  |--shared/
    |--apps/
    |--config/
    |--resources/
```

- OpenAPI3 application artifacts

- OpenAPI2 application artifacts



Tech-Tip: Using permission bits to control access



ID=**LIBSERV**
Group=**LIBGRP**

```
export JAVA_HOME=<path_to_64_bit_Java>
export WLP_USER_DIR=/var/zosconnect
./server create zceesrvr
```

/var/zosconnect	751	LIBSERV	LIBGRP
/servers	751	LIBSERV	LIBGRP
/zceesrv1	751	LIBSERV	LIBGRP
/apps	761	LIBSERV	LIBGRP
/configDropins	761	LIBSERV	LIBGRP
/overrides	761	LIBSERV	LIBGRP
/logs	771	LIBSERV	LIBGRP
/messages.log	644	LIBSERV	LIBGRP
/resources	751	LIBSERV	ADMGRP
/security	777	LIBSERV	LIBGRP
/zosconnect	751	LIBSERV	ADMGRP
/apis	761	LIBSERV	ADMGRP
/apiRequesters	761	LIBSERV	ADMGRP
/rules	761	LIBSERV	ADMGRP
/services	761	LIBSERV	ADMGRP
server.xml	460	LIBSERV	ADMGRP
/shared	750	LIBSERV	LIBGRP
/apps	750	LIBSERV	LIBGRP
/config	750	LIBSERV	LIBGRP

~~Often you may be tempted to use command chmod -R 777 *~~

Sample of OMVS commands to manage permission bits

```
export WLP_USER_DIR=/var/zosconnect
cd $WLP_USER_DIR
chmod o+x -R servers
chmod o+x servers/zceesrvr/resources
chmod -R o+x servers/zceesrvr/resources/*
chmod g+r -R servers
chmod g+r servers/zceesrvr/resources
chmod -R g+r servers/zceesrvr/resources/*
chmod g+w server.xml
```

Warning: Access for Owner (u), Group(g), Others(o) depend on user ID (UID) and group ID (GID) as stored with the directory or file, not the actual SAF identity or group.
This has implications when moving entire filesystems from one LPAR to another using utilities like ADRDSSU.

Tech Tip: Use multiple mount points and dedicated ZFS file systems

- Create mount points in the “administrative” directory for shared r/w directories
- Avoid creating directories and files in the root file system.
- Use a common or shared mount point
 - Use /var mount point for local read/write file systems
 - Use /global for sharing a mount point across multiple LPARs
- Use ZFS filesystems and use AGGRGROW to allow R/W ZFS filesystems to automatically go into extents (>16).

SYS1.PARMLIB (BPXPRM##)

```

MOUNT FILESYSTEM('OMVS.ZCEEVAR.ZFS')
  MOUNTPOINT('/var/zosconnect')
  TYPE(ZFS) MODE(READ)
MOUNT FILESYSTEM('OMVS.ZCEE.SERVERS.ZFS')
  MOUNTPOINT('/var/zosconnect/servers')
  TYPE(ZFS) PARM('AGGRGROW') MODE(RDWR)
MOUNT FILESYSTEM('OMVS.ZCEE.GROUP1.ZFS')
  MOUNTPOINT('/var/zosconnect/group1')
  TYPE(ZFS) PARM('AGGRGROW') MODE(RDWR)
MOUNT FILESYSTEM('OMVS.ZCEE.GROUP2.ZFS')
  MOUNTPOINT('/var/zosconnect/group2')
  TYPE(ZFS) PARM('AGGRGROW') MODE(RDWR)
MOUNT FILESYSTEM('OMVS.ZCEE.GROUP.ZFS')
  MOUNTPOINT('/var/zosconnect/group3')
  TYPE(ZFS) PARM('AGGRGROW') MODE(RDWR)

```

- Create a dedicated filesystem for the root z/OS Connect /var directory, e.g., /var/zosconnect/v3r0/extensions. This directory structure can not be changed. This provides portability for migrations and system upgrades. Note: MODE(READ) will apply to /var/zosconnect/servers.

- Create a dedicated filesystem for each set or groups of servers. These filesystems will contain the server configuration directories for 1 or more servers.
- Each server's WLP_USER_DIR environment variable will be set to the mount point, e.g., *WLP_USER_DIR=/var/zosconnect/group1* when the server is created and in the server's startup JCL.

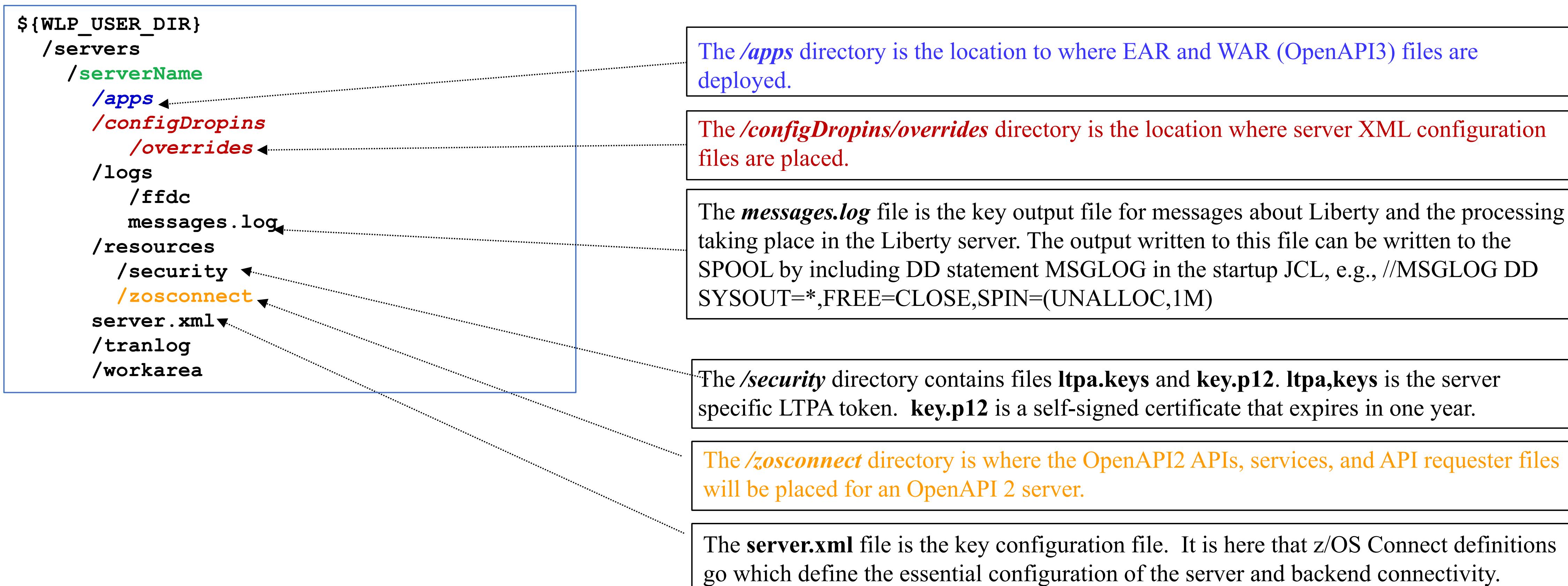
df -P | grep /var/zosconnect

Filesystem	512-blocks	Used	Available	Capacity	Mounted on
OMVS.ZCEEVAR.ZFS	69120	68658	462	100%	/var/zosconnect
OMVS.ZCEE.SERVERS.ZFS	159120	76455	82665	48%	/var/zosconnect/servers
OMVS.ZCEE.GROUP1.ZFS	135360	1506	133854	2%	/var/zosconnect/group1
OMVS.ZCEE.GROUP2.ZFS	4059360	2591284	1468076	64%	/var/zosconnect/group2
OMVS.ZCEE.GROUP3.ZFS	135360	17858	117502	14%	/var/zosconnect/group3



More details of the contents of a server's directory structure

A server's configuration structure looks like this (N.B. OpenAPI 2 and OpenAPI 3 servers do not coexist as shown here):



The `WLP_USER_DIR` environment variables sets the value of the root directory of the server's configuration files and directories, e.g.,
`WLP_USER_DIR=/var/zosconnect`

Use “include” files to extend and manage a server’s configuration

- Setup a server.xml using ‘include’ statements and allow other administrator to manage those included files, but not the server.xml itself.
- Control what configuration can be overridden in included files using the ‘onConflict’ option provided with the include element (see Ignore, Replace, Merge).

https://www.ibm.com/support/knowledgecenter/en/SSAW57_liberty/com.ibm.websphere.wlp.nd.multiplatform.doc/ae/cwlp_config_include.html

server.xml (owned by ID ADMIN1)

```
<include location="${server.config.dir}/includes/db2.xml onConflict="IGNORE"/>
<include location="${server.config.dir}/includes/cics.xml onConflict="IGNORE"/>
<include location="${server.config.dir}/includes/imsDb.xml onConflict="IGNORE"/>
<featureManager>
  <feature>zosconnect:zosConnect-2.0</feature>
  <feature>zosconnect:zosConnectCommands-1.0</feature>
  <feature>apiDiscovery-1.0</feature>
<featureManager>
```

db2.xml (owned and managed by a DBA)

```
<server description="Db2 REST">
  <zosconnect_zosConnectServiceRestClientConnection id="Db2Conn" host="wg31.washington.ibm.com" port="2446" basicAuthRef="dsn2Auth" />
  <zosconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth" applName="DSN2APPL"/>
</server>
```

cics.xml (owned and managed by a CICS administrator)

```
<server description="CICS">
  <featureManager> <feature>zosconnect:cicsService-1.0</feature> </featureManager>
  <zosconnect_cicsIpicConnection id="catalog" host="wg31" port="1491"/>
  <zosconnect_cicsIpicConnection id="cscvinc" host="wg31" port="1493"/>
</server>
```

imsDB.xml (owned and managed by a IMS administrator)

```
<server description="IMS DATABASE">
  <featureManager> <feature>zosconnect:dbService-1.0</feature> </featureManager>
  <connectionFactory id="DFSTIVPAComm" > <properties.imsudbJLocal databaseName="DFSTIVPA" datastoreName="IVP1" driverType="4" portNumber="5555" datastoreServer="wg31" user="USER1" password="USER1" flattenTables="True"/> </connectionFactory>
</server>
```

Nesting of an include file within a include file is possible



Tech-Tip: Review configuration conflicts

```
ÝAUDIT  " CWWKG0102I: Found conflicting settings for cscvincAPI instance of zosconnect_endpointConnection configuration.  
Property port has conflicting values:  
  Value 9443 is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value 9443 is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value 9463 is set in file:/var/zosconnect/servers/myServer/includes/oauth.xml.  
Property port will be set to 9463.  
Property host has conflicting values:  
  Value https://dvipa.washington.ibm.com is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value https://dvipa.washington.ibm.com is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value https://mpz3.washington.ibm.com is set in file:/var/zosconnect/servers/myServer/includes/oauth.xml.  
Property host will be set to https://mpz3.washington.ibm.com.  
Property authenticationConfigRef has conflicting values:  
  Value mySAFAuth is set in file:/var/zosconnect/servers/myServer/includes/apiRequesterHTTPS.xml.  
  Value myoAuthConfig is set in file:/var/zosconnect/servers/myServer/includes/oauth.xml.  
Property authenticationConfigRef will be set to myoAuthConfig.
```

onConflict="MERGE" Conflicting elements will be merged, and the last value encountered will be used.

onConflict="REPLACE" When elements conflict, the element in the included file will be ignored

onConflict="IGNORE" Conflicting elements in the included file are ignored.

Tech-Tip: Understand the difference between a singleton XML configuration entry versus entries where multiple entries are allowed. During the XML parsing of the configuration files, multiple occurrences of singleton XML configuration entry are combined as they are encountered whereas multiple entries are kept separate based on the value of their id.

Use a *bootstrap.properties* file to help customize a server's XML configuration#



zceesrv1's bootstrap.properties

```
httpPort=9080
httpsPort=9443
ipicPort=1491
host=*
cicsHost=wg31.washington.ibm.com
network=ZOSCONN1
applid=ZOSCONN1
com.ibm.ws.zos.core.angelName=namedAngel
bootstrap.include=/var/liberty/common-bootstrap.properties
```

zceesrv2's bootstrap.properties

```
httpPort=9090
httpsPort=9453
ipicPort=1492
host=wg31.washington.ibm.com
cicsHost=wg31.washington.ibm.com
network=ZOSCONN2
applid=ZOSCONN2
com.ibm.ws.zos.core.angelName=namedAngel
bootstrap.include=/var/liberty/common-bootstrap.properties
```

server.xml

```
<!-- To access this server from a remote client, add a host attribute to the following
element, e.g. host="*" -->
<httpEndpoint id="defaultHttpEndpoint"
               host="${host}"
               httpPort ="${httpPort}"
               httpsPort ="${httpsPort}" />
```

ipicIDProp.xml

```
<zosconnect_cicsIpicConnection id="catalog"
                                host ="${cicsHost}" port ="${ipicPort}"
                                zosConnectNetworkid ="${network}"    zosConnectApplid ="${applid}"/>

<zosconnect_cicsIpicConnection id="cscvinc"
                                host ="${cicsHost}" port ="${ipicPort}"
                                zosConnectNetworkid ="${network}"    zosConnectApplid ="${applid}"/>

<zosconnect_cicsIpicConnection id="miniloan"
                                host ="${cicsHost}" port ="${ipicPort}"
                                zosConnectNetworkid ="${network}"    zosConnectApplid ="${applid}"/>
```

**N.B. Java directives
can also be provided.**

**N.B. Boot strap
properties can be
included from a
common boot strap
properties file**



Tech-Tip: A suggestion for managing the server.xml configuration file

Default server.xml configuration file

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
    <!-- Enable features -->
    <featureManager>
    - - - - -
        </featureManager>

    <!-- To access this server from a remote client add a host attribute
        <httpEndpoint id="defaultHttpEndpoint"
            host="*"
            httpPort="9080"
            httpsPort="9443" />
    <!-- add cors to allow cross origin access, e.g. when using swagger
        <cors id="defaultCORSConfig"
            domain="/"
            allowedOrigins="*"
            allowedMethods="GET, POST, PUT, DELETE, OPTIONS"
            allowedHeaders="Origin, Content-Type, Authorization, Cache-Control"
            allowCredentials="true"
            maxAge="3600"/>
    - - - - - 30 Line(s) not Displayed
</server>
```

Modified server.xml configuration file

```
<server description="zCEE Server">
    <include location="${server.config.dir}/includes/safSecurity.xml"/>
    <include location="${server.config.dir}/includes/ipicIDProp.xml"/>
    <include location="${server.config.dir}/includes/keyring.xml"/>
    <include location="${server.config.dir}/includes/groupAccess.xml"/>
    <include location="${server.config.dir}/includes/shared.xml"/>
    <include location="${server.config.dir}/includes/apiRequesterHTTPS.xml"/>
    <include location="${server.config.dir}/includes/imsDatabase.xml"/>
    <!-- Enable features -->
    <featureManager>
    - - - - -
        </featureManager>
    <!-- To access this server from a remote client add a host attribute
        <httpEndpoint id="defaultHttpEndpoint"
            host="${host}"
            httpPort="${httpPort}"
            httpsPort="${httpsPort}" />
    - - - - - 36 Line(s) not Displayed
</server>
```

The simplifies administration by :

- Using a *bootstrap.properties* file to customize the ports in the *server.xml* file.
- Using “include” statements to make further changes such as adding additional features and additional XML configuration elements.
- Review <https://www.ibm.com/docs/en/was-liberty/nd?topic=liberty-configuration-element-merging-rules> to understand merging rules.
- Consider providing configuration elements by placing server XML files in the .../configDropins/original subdirectory.



Two ways to provide STDENV input in a JCL Procedure

Use the STDENV DD statement to scale servers and share configuration properties horizontally

```
//ZCON      EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=8G,  
//          PARM='PGM &ZCONHOME./bin/zosconnect run &PARMS. --clean'  
//STDERR    DD SYSOUT=*,FREE=CLOSE,SPIN=(UNALLOC,1M)  
//STDOUT    DD SYSOUT=*  
//STDIN     DD DUMMY  
//STDENV    DD PATH='/var/zcee/properties/&PARMS..property',  
//                  PATHOPTS=ORDONLY  
//          or  
//STDENV    DD DISP=SHR,DSN=JOHNSON.ZCEE.STDENV(COMMON)  
//          DD DISP=SHR,DSN=JOHNSON.ZCEE.ZCEESRVR
```

Either one OMVS property file or multiple PDS members.

The last occurrence environment variable encountered determines the value of the environment variable.

Member COMMON
_BPX_SHAREAS=YES
_CEE_RUNOPTS=HEAPPOOLS (ON) ,HEAPPOOLS64 (ON)
JAVA_HOME=/usr/lpp/java/J8.0_64
ZCON_ENV_DEBUG=TRUE
WLP_USER_DIR=/var/zosconnect

Which value used for a Java option or property depends on which environment variable is used to specify the option or property.

Member ZCEESRVR
OPENJ9_JAVA_OPTIONS=-Dcom.ibm.ws.zos.core.angelName=ZCEEANGL
JVM_OPTIONS=-Xoptionsfile=/var/zcee/properties/javaHCD.property -Xmx512m -verbose:sizes
JAVA_HOME=/u/johnson/java/J8.0_64
WLP_USER_DIR=/var/ats/zosconnect

Using //STDENV DD * is discouraged because of the 80 character record limit.



Tech/Tip: Liberty Java Directives for controlling output

com.ibm.ws.logging.console.format (consoleFormat) - The required format for the console. Valid values are basic or json format.

com.ibm.ws.logging.console.log.level (consoleLogLevel) - This filter controls the granularity of messages that go to the console. The valid values are INFO, AUDIT, WARNING, ERROR, and OFF. By default, the console log level is set to AUDIT.

com.ibm.ws.logging.hideMessage (hideMessage) - Use this attribute to configure the messages that you want to hide from the *console.log* and *message.log* files. If the messages are configured to be hidden, then they are redirected to the *trace.log* file.

com.ibm.ws.logging.log.directory (logDirectory) - Use this attribute to set a directory for all log files, excluding the *console.log* file, but including FFDC. The default log location path is *WLP_OUTPUT_DIR/serverName/logs*

com.ibm.ws.logging.max.file.size (maxFileSize) - The maximum size (in MB) that a log file can reach before it is rolled. The Liberty runtime does only size-based log rolling. To disable this attribute, set the value to 0. The maximum file size is approximate. By default, the value is 20.

com.ibm.ws.logging.max.files (maxFiles) - If a maximum file size exists, this setting is used to determine how many of each of the log files are kept. This setting also applies to the number of exception logs that summarize exceptions that occurred on any day. So, if this number is 10, you might have 10 message logs, 10 trace logs, and 10 exception summaries in the *ffdc* directory. The default value is 2.

com.ibm.ws.logging.message.format (messageFormat) - The required format for the *messages.log* file. Valid values are basic or json format. By default, *messageFormat* is set to the environment variable *WLP_LOGGING_MESSAGE_FORMAT* (if set) or basic.

com.ibm.ws.logging.trace.file.name (traceFileName) - The *trace.log* file is only created if additional or detailed trace is enabled. *stdout* is recognized as a special value; and causes trace to be directed to the original standard out stream.

bootstrap.properties example:

```
com.ibm.ws.logging.message.file.name=basqstrtMessages.log  
com.ibm.ws.logging.log.directory=/u/common/logs
```

N.B. *consoleFormat*, *logDirectory*, etc. can be specified in the *<logging/>* Liberty configuration element. Note the recommendation for the attributes in red is for them to be provided in Java directives.

Tech-Tip: Consider using symbolic links as an administrative shortcut

- Create an “administration” subdirectory, e.g., *LibertyConfigs* in directory */var*
- Then create a symbolic link in the “administration” directory to each Liberty server’s configuration directory and other frequently accessed directories.

```
ls -al /var/LibertyConfigs
drwxrwxrwx 4 JOHNSON SYS1          8192 Aug 16 12:23 .
drwxrwxrwt 25 OMVSKERN SYS1        8192 Aug 16 11:56 ..
lrwxrwxrwx 1 JOHNSON SYS1          57 Aug 16 12:22 CSCWLP -> /var/wlp/cics/CICS53Z/CSCWLP/wlp/usr/servers/defaultServer
lrwxrwxrwx 1 JOHNSON SYS1          57 Aug 16 12:22 CICSWLP -> /var/wlp/cics/CICS53Z/CICSWLP/wlp/usr/servers/cicswlp
drwxrwxrwx 2 JOHNSON SYS1        8192 Aug 16 15:30 hcd
lrwxrwxrwx 1 JOHNSON SYS1          27 Jun 10 15:55 includes -> /global/zosconnect/includes
lrwxrwxrwx 1 JOHNSON SYS1          28 Aug 16 10:12 mqweb -> /var/mqm/mqweb/servers/mqweb
lrwxrwxrwx 1 JOHNSON SYS1          32 Jun  4 12:49 myServer -> /var/zosconnect/servers/myServer
drwxr-xr-x 2 JOHNSON SYS1        8192 Aug 16 13:14 properties
lrwxrwxrwx 1 JOHNSON SYS1          18 Aug 17 12:47 shared -> /var/shared/zosconnect/resources/zosconnect
lrwxrwxrwx 1 JOHNSON SYS1          24 May 13 2020 walop3a -> /var/wlp/servers/walop3a
lrwxrwxrwx 1 JOHNSON SYS1          24 May 13 2020 walrp3a -> /var/wlp/servers/walrp3a
lrwxrwxrwx 1 JOHNSON SYS1          31 May 14 2020 wazs34a -> /var/zosconnect/servers/wazs34a
lrwxrwxrwx 1 JOHNSON SYS1          24 Aug 16 10:32 wlphats -> /var/wlp/servers/wlphats
lrwxrwxrwx 1 JOHNSON SYS1          36 Aug 16 10:31 zceearpir -> /var/ats/zosconnect/servers/zceearpir
lrwxrwxrwx 1 JOHNSON SYS1          39 Aug 16 10:18 zceecics -> /var/cicsts/zosconnect/servers/zceecics
lrwxrwxrwx 1 JOHNSON SYS1          35 Aug 16 10:31 zceedvm -> /var/ats/zosconnect/servers/zceedvm
lrwxrwxrwx 1 JOHNSON SYS1          32 Jun 10 15:54 zceepid -> /var/zosconnect/servers/zceepid
lrwxrwxrwx 1 JOHNSON SYS1          36 Aug 16 10:14 zceesrvr -> /var/ats/zosconnect/servers/zceesrvr
lrwxrwxrwx 1 JOHNSON SYS1          44 Aug 16 11:57 zosmfServer -> /var/zosmf/configuration/servers/zosmfServer
```

There are various Liberty servers shown here. There are CICS Liberty servers, z/OS Connect servers, a MQ Web Console Liberty server, a zOSMF Liberty server, a HATS Liberty server and a few standard Liberty servers for Java applications.

One administration directory to manage them all!

Tech-Tip: Be aware of the default for update polling

```
<!-- Application Monitoring      -->
<applicationMonitor updateTrigger="polled" dropinsEnabled="true"/>

<!-- config requires updateTrigger="mbean" for REFRESH command support -->
<config updateTrigger="polled" monitorInterval="500ms"/>           Change to mbean

<!-- zosConnect APIs -->
<zosconnect_zosConnectAPIs pollingRate="5s" updateTrigger="disabled"     />

<!-- zosConnect API requesters -->
<zosconnect_apiRequesters updateTrigger="disabled" pollingRate="5s"/>

<!-- zosConnect Services -->
<zosconnect_services pollingRate="5s" updateTrigger="disabled"/>

<!-- zosConnect policies -->
<zosconnect_policy pollingRate="1m" updateTrigger="disabled"/>

<!-- zosConnect data transformer -->
<zosconnect_zosConnectDataXform pollingrate="2s" updateTrigger="polled"/>

<!--A security certificate repository -->
<keystore pollingrate="500ms" updateTrigger="mbean"/>
```



Sharing XML configuration files between servers using \${shared.config.dir}

Add an “includes” subdirectories {shared.config.dir} with a symbolic links to a common location. This common directory can be accessed from multiple servers on a single or from multiple LPARs. Additions and updates to the “include” files are then made in one single administrative directory.

Symbolic links from servers shared configuration \${shared.config.dir} to common directory

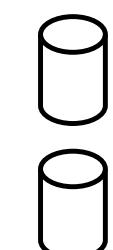
Symbolic links to a shared local LPAR directory

```
ln -s /var/shared/zosconnect/includes /var/zosconnect/shared/config  
ln -s /var/shared/zosconnect/includes /var/ats/zosconnect/shared/config  
ln -s /var/shared/zosconnect/includes /var/wsc/zosconnect/shared/config
```

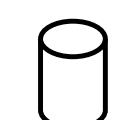
Symbolic links to a shared Sysplex directory *

```
ln -s /global/zosconnect/includes /var/zosconnect/servers/shared/config  
ln -s /global/zosconnect/includes /var/ats/zosconnect/shared/config  
ln -s /global/zosconnect/includes /var/wsc/zosconnect/shared/config
```

```
<include location="${shared.config.dir}/safSecurity.xml"/>  
<include location="${shared.config.dir}/ipicIDProp.xml"/>  
<include location="${shared.config.dir}/keyringOutboundMutual.xml"/>  
<include location="${shared.config.dir}/groupAccess.xml"/>  
<include location="${shared.config.dir}/shared.xml"/>  
<include location="${shared.config.dir}/db2.xml"/>  
<include location="${shared.config.dir}/oauth.xml"/>
```



/var/shared/zosconnect/includes



/global/zosconnect/includes

Contents of the common “includes” directory

basicSecurity.xml
db2.xml
db2TLS.xml
groupAccess.xml
ipic.xml
ipicIDProp.xml
keyringInbound.xml
keystore.xml
keyringMutual.xml
keyringOutboundMutual.xml
safSecurity.xml

The server.xml file contains these “include” statements and the server XML is portable regardless of the underlying filesystem infrastructure.

For example, changing *basicSecurity.xml* to *safSecurity.xml* and refreshing the configuration changes security from basic to SAF

F ZCEESRVR ,REFRESH,CONFIG



A practical example-PTF V3.0.35 included a CORS update

July 2020

V3.0.35 (APAR PH26291)
Server code update

Enhancements

- The text of messages BAQR0417W and BAQR0418W has been updated. For more information, see z/OS Connect EE [Runtime Messages](#).

Fixes

- PH21761 A CICS region reports **SOS DFHSM0133 WBSEBUF** when z/OS Connect EE requester is in use.
- PH25345 Passing user credentials in the request body to the authentication server to obtain a JWT causes a NPE in z/OS Connect EE.
- PH21819 z/OS Connect EE sets some CORS headers automatically.

Attention

When this fix is applied, additional CORS configuration is required in `server.xml` to enable connections from the z/OS Connect EE API toolkit and JavaScript clients. For more information, see [Configuring Cross-Origin Resource Sharing on a z/OS Connect Enterprise Edition Server](#)

`cors.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="CORS entries">

    <!-- add cors to allow cross origin access, e.g. when using swagger doc from zOS Connect Enterprise Edition -->
    <cors id="defaultCORSConfig"
        domain="/"
        allowedOrigins="*"
        allowedMethods="GET, POST, PUT, DELETE, OPTIONS"
        allowedHeaders="Origin, Content-Type, Authorization, Cache-Control, Expires, Pragma"
        allowCredentials="true"
        maxAge="3600"/>

</server>
```

`server.xml`

```
<include location="${server.config.dir}/cors.xml"/>
```



Sharing XML configuration files – using ‘variables’ files

“variables” files whose names are based on the name of the server, e.g., \${wlp.server.name}

myServer.xml

```
<variable name= "unauthenticatedUser" value= "WSGUEST" />
<variable name="profilePrefix" value= "BBGZDFLT" />
```

zceopid.xml

```
<variable name= "unauthenticatedUser" value="ZCGUEST" />
<variable name="profilePrefix" value="EMJZDFLT" />
```

server.xml

```
<server description="new server">
<include location="${server.config.dir}/includes/safSecurity.xml"/>
<include location="${server.config.dir}/includes/${wlp.server.name}.xml"/>

    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:zosConnect-2.0</feature>
        <feature>zosconnect:zosConnectCommands-1.0</feature>
    </featureManager>
```

safSecurity.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="SAF security">

    <!-- Enable features -->
    <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>zosSecurity-1.0</feature>
    </featureManager>

    <webAppSecurity allowFailOverToBasicAuth="true" />
    <safRegistry id="saf" />
    <safAuthorization racRouteLog="ASIS" />
    <safCredentials unauthenticatedUser="${unauthenticatedUser}"
        profilePrefix="${profilePrefix}" />
</server>
```



Using the Liberty server's configuration drop-in's directories

Located in the same directory as the *server.xml* configuration file.

- Configuration files in the */default* directory provides defaults for configuration elements not present in *server.xml*
- Configuration files in the */overrides* directory adds to or replaces the configuration elements found in *server.xml*

```
 ${WLP_USER_DIR}
  /servers
    /serverName
      /apps
      /configDropins
        /default
        /overrides
    /logs
      /ffdc
      messages.log
    /resources
      /security
  server.xml
  /tranlog
  /workarea
```

```
commonFeatures.xml
<server description="Common Server Features">

  <!-- Enable features -->
  <featureManager>
    <feature>adminCenter-1.0</feature>
    <feature>restConnector-2.0</feature>
  </featureManager>

  <remoteFileAccess>
    <readDir>/var/zcee/includes</readDir>
    <readDir>/global/zosconnect/includes</readDir>
    <writeDir>${server.config.dir}</writeDir>
  </remoteFileAccess>

</server>
```

```
safSecurity.xml
<?xml version="1.0" encoding="UTF-8"?>
<server description="SAF security">

  <!-- Enable features -->
  <featureManager>
    <feature>appSecurity-2.0</feature>
    <feature>zosSecurity-1.0</feature>
  </featureManager>

  <webAppSecurity allowFailOverToBasicAuth="true" />
  <safRegistry id="saf" />
  <safAuthorization racRouteLog="ASIS" />
  <safCredentials unauthenticatedUser="${unauthenticatedUser}"
    profilePrefix="${profilePrefix}" />
</server>
```

Another directory that must be manually created.



Consider simplifying administration by combining include files and by using server variables

Default server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
<include location="${server.config.dir}/includes/${wlp.server.name}.xml"/>

    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:zosConnect-3.0</feature>
        <feature>openapi-3.0</feature>
    </featureManager>

    <!-- To access this server from a remote client add a host attribute
    to the following element, e.g. host="*"
    <httpEndpoint id="defaultHttpEndpoint"
        host="*"
        httpPort="9080"
        httpsPort="9443" />
```

`${server.config.dir}/includes/${wlp.server.name}.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
<variable name="httpPort" value="9081"/>
<variable name="httpsPort" value="9445"/>
<variable name="hostName" value="*"/>
<variable name="CICS_HOST" value="wg31.washington.ibm.com"/>
<variable name="CICS_PORT" value="1491"/>
<variable name="DB2_HOST" value="wg31.washington.ibm.com"/>
<variable name="DB2_PORT" value="2446"/>
<variable name="DB2_USERNAME" value="USER2"/>
<variable name="DB2_PASSWORD" value="USER2"/>
<include location="${shared.config.dir}/safSecurity.xml"/>
<include location="${shared.config.dir}/httpEndpoint.xml"/>
<include location="${shared.config.dir}/db2.xml"/>
<include location="${shared.config.dir}/cics.xml"/>
<include location="${shared.config.dir}/keystore.xml"/>
</server>
```

```
 ${server.config.dir}/includes/httpEndpoint.xml"/>
<server description="basic security">
    <httpEndpoint id="defaultHttpEndpoint"
        host="${hostName}"
        httpPort="${httpPort}"
        httpsPort="${httpsPort}" />
</server>
```

```
 ${server.config.dir}/includes/db2.xml"/>
<?xml version="1.0" encoding="UTF-8"?>
<server description="Default server">
    <featureManager>
        <feature>zosconnect:db2-1.0</feature>
    </featureManager>
    <zosconnect_credential user="${DB2_USERNAME}"
        password="${DB2_PASSWORD}" id="commonCredentials" />
    <zosconnect_db2Connection id="db2Conn" host="${DB2_HOST}"
        port="${DB2_PORT}" credentialRef="commonCredentials" />
</server>
```

`${server.config.dir}/includes/cics.xml"`

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="CICS IPIC connections">
    <!-- Enable features -->
    <featureManager>
        <feature>zosconnect:cics-1.0</feature>
    </featureManager>
    <zosconnect_cicsIpicConnection id="cicsConn" host="${CICS_HOST}"
        port="${CICS_PORT}" />
</server>
```



For example, the MQ Console Liberty configuration uses a R/O common configuration file

The server.xml only contain 'include' commands.

server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
This file defines the configuration for the mqweb server. Please do not
make changes to this file. Any changes made by users should be in the
referenced mqwebuser.xml file.
-->
<server>
    <include location="${wlp.install.dir}/mq/etc/mqweb.xml"/>
    <include location="mqwebuser.xml"/>
</server>
```

'Include' file *mqweb.xml* is provided by MQ and usually a read only file. It sets default values for variables and the required Liberty configuration elements.

'Include' file *mqwebuser.xml* is a read/write file and provides overrides for the default values of variables and any other Liberty configuration elements.



Sharing XML configuration files – using ‘variables’ files

```
/usr/lpp/mqm/V3R3M4/web/mqm/etc/mqweb.xml

<?xml version="1.0" encoding="UTF-8"?>
<server>

<featureManager>
    <feature>jaxrs-2.1</feature>
    <feature>ssl-1.0</feature>
    <feature>jndi-1.0</feature>
    <feature>concurrent-1.0</feature>
    <feature>websocket-1.0</feature>
    <feature>applicationMonitorMQ-1.0</feature>
</featureManager>

<!--
Configurable properties. These can be overridden by setting appropriate
-->
<variable name="httpHost" value="localhost"/>
<variable name="httpPort" value="-1"/>
<variable name="httpsPort" value="9443"/>
<variable name="mqRestRequestTimeout" value="30"/>
<variable name="mqConsoleAutostart" value="true"/>
<variable name="mqRestAutostart" value="true"/>
<variable name="mqRestGatewayQmgr" value="" />
<variable name="mqRestGatewayEnabled" value="true"/>
<variable name="traceSpec" value="*=info"/>
<variable name="maxTraceFileSize" value="20"/>
<variable name="maxTraceFiles" value="2"/>
<variable name="maxMsgTraceFileSize" value="200"/>
<variable name="maxMsgTraceFiles" value="5"/>
<variable name="ltpaExpiration" value="120"/>
<variable name="ltpaCookieName" value="LtpaToken2_${httpsPort}"/>
<variable name="secureLtpa" value="true"/>
<variable name="mqRestCsrfValidation" value="true"/>
<variable name="mqRestCorsAllowedOrigins" value="" />
<variable name="mqRestCorsMaxAgeInSeconds" value="0"/>
<variable name="mqRestMftCoordinationQmgr" value="" />
<variable name="mqRestMftCommandQmgr" value="" />
<variable name="mqRestMftEnabled" value="false"/>
<variable name="mqRestMftReconnectTimeoutInMinutes" value="30"/>
<variable name="mqRestMessagingEnabled" value="true"/>
<variable name="mqRestMessagingMaxPoolSize" value="20"/>
```

mqwebuser.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server>
    <featureManager>
        <feature>appSecurity-2.0</feature>
        <feature>zosSecurity-1.0</feature>
        <feature>basicAuthenticationMQ-1.0</feature>
        <feature>apiDiscovery-1.0</feature>
    </featureManager>

    <enterpriseApplication id="com.ibm.mq.console"/>
    <enterpriseApplication id="com.ibm.mq.rest"/>

    <safAuthorization racRouteLog="ASIS"/>
    <safRegistry id="saf"/>
    <safAuthorization id="saf"
        reportAuthorizationCheckDetails="true" />
    <safCredentials profilePrefix="MQWEB" unauthenticatedUser="WSGUEST"/>

    <webAppSecurity allowFailOverToBasicAuth="true" />

    <variable name="httpsPort" value="1419"/>
    <variable name="httpHost" value="-1"/>
    <variable name="mqRestMessagingEnabled" value="true"/>

    <b><config monitorInterval="5s" updateTrigger="mbean"/></b>

</server>
```

Tech-Tip: Use Symbolic links to simplify commands used in OMVS and JCL

Performing commands:

```
ln -s /global/zosconnect/includes /var/zcee/includes
ln -s /var/zosconnect/servers/zceesrv1 /var/zcee/zceesrv1
ln -s /var/zosconnect/servers/zceesrv2 /var/zcee/zceesrv2
```

Will change these OMVS commands from:

```
ln -s /global/zosconnect/includes /var/zosconnect/servers/zceesrv1/includes
ln -s /global/zosconnect/includes /var/zosconnect/servers/zceesrv2/includes
```

To simpler (and shorter) OMVS commands:

```
ln -s /var/zcee/includes /var/zcee/zceesrv1/includes
ln -s /var/zcee/includes /var/zcee/zceesrv2/includes
```

Directory Shortcuts

- Create a shortcut from the shared administrative *include* directory to the Sysplex or LPAR shared directory
- Create shortcuts from the server's administrative directories to each server's configuration directory.

N.B. These are symbolic links to symbolic links.

ln -s oldname newname

These symbolic links can be used as JCL symbols

```
//EXPORT EXPORT SYMLIST=(*)
// SET SERVER= 'zceesrv1'
// SET SHARED='/var/zcee/includes'
// SET WLPUSER='var/zosconnect'
//ZCEELN EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH +
ln -s &SHARED /var/zcee/&SERVER/includes
instead of entering the full directory names as in
ln -s /global/zosconnect/includes +
&WLPUSER/servers/&SERVER/includes
```

And added as exports to /u/home/.profile or /etc/profile files

```
export serverName=zceesrv1
export shared=/var/zcee/includes
export WLP_USER_DIR=/var/zosconnect
```

```
//ZCEELN EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//SYSTSIN DD *
BPXBATCH SH +
ln -s $shared /var/zcee/$serverName/includes
instead of entering the full directory names as in
ln -s /global/zosconnect/includes +
$WLPUSER/servers/$serverName/includes
```



Combine these techniques to standardized the creation and customization of new servers

```
//*****  
///* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
// SET ZCEEPATH='/usr/lpp/IBM/zosconnect/v3r0'  
// SET SERVER='zceesrvr'  
// SET TEMPLATE='zosconnect:default'  
// SET WLPUSER='/var/ats/zosconnect'  
// SET USER='ATSSERV'  
// SET GROUP='ATSGRP'  
//*****  
///* Step ZCEESRVR - Use the zosconnect command to create a server  
//*****  
//ZCEESRVR EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
export JAVA_HOME=&JAVAHOME; +  
export WLP_USER_DIR=&WLPUSER; +  
&ZCEEPATH/bin/zosconnect create &SERVER +  
--template=&TEMPLATE; +  
ln -s $WLP_USER_DIR/servers/&SERVER /var/liberty/&SERVER; +  
cp /var/liberty/properties/bootstrap.properties +  
/var/liberty/&SERVER; +  
cp /var/liberty/properties/server.xml +  
/var/liberty/&SERVER; +  
chown -R &USER:&GROUP $WLP_USER_DIR/servers/&SERVER
```

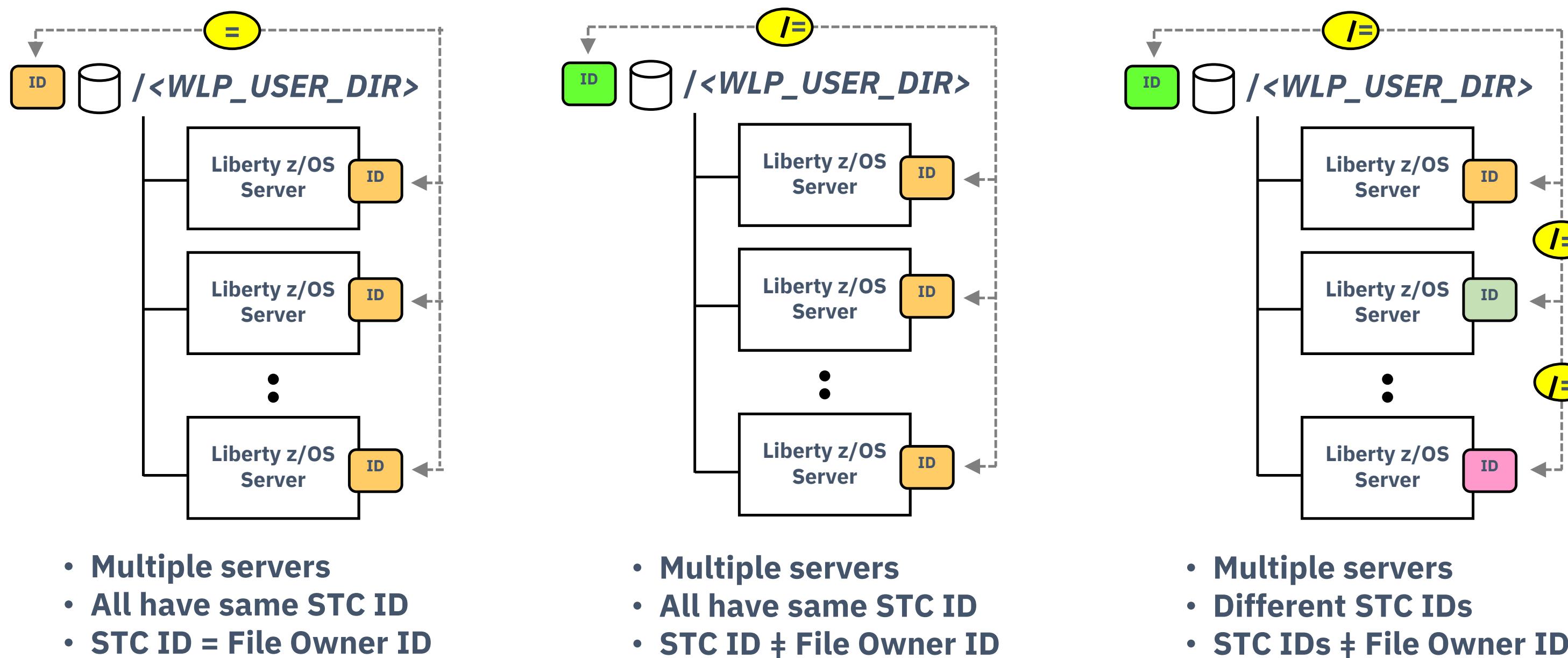
- Export environment variables
- Create the server and its configuration directories, etc.
- **Create a symbolic link from the server's configuration directory to common administration directory**
- **Copy a default bootstrap.properties and server.xml files into the server's configuration directory**
- **Change owner and group of the server's configuration directories and files to the SAF identity under which the server will run**

Liberty Security

Overview

z/OS Security – Range of options – Started Task IDs

On z/OS, the best practice for Liberty servers in production is that they run as ‘Started Tasks’ (STCs).

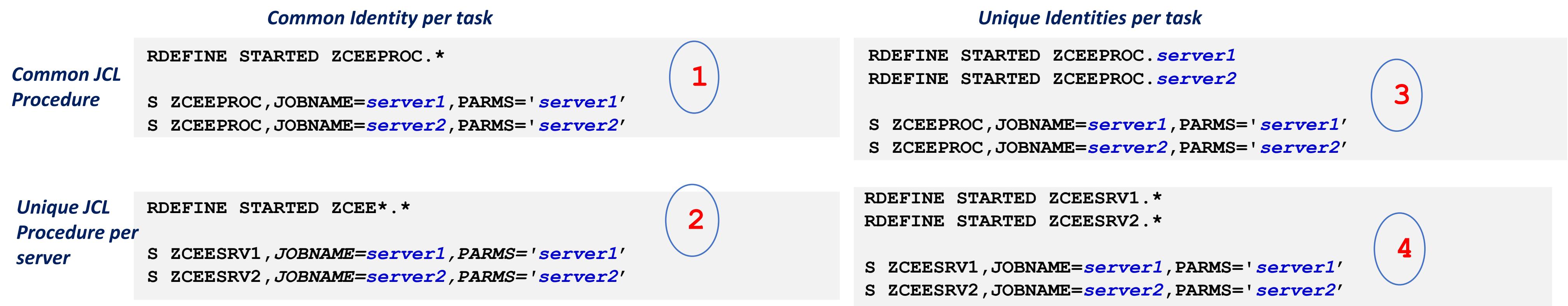


Should all servers sharing WLP_USER_DIR share the same STC ID?
 It is a matter of the degree of identity isolation that is required

z/OS Security: Assigning ID to started tasks: SAF STARTED class

The first question here is whether you wish to have a common started task ID that is shared among servers, or if you wish each server to have a unique ID

Then the second question is whether servers under a WLP_USER_DIR will share a common JCL start proc, or use unique start procs for each server



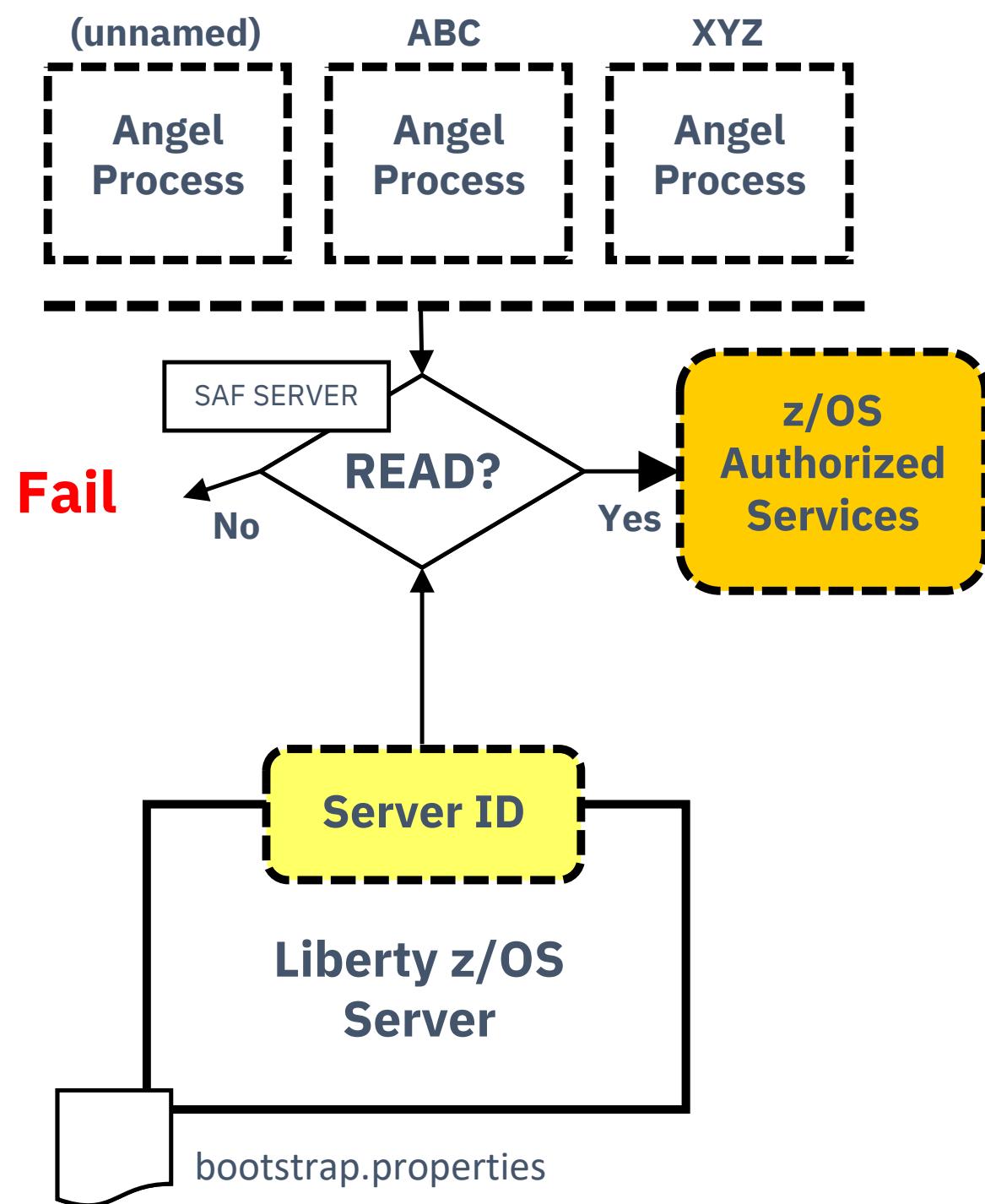
Note: Using unique JCL procedure eliminates the need to specify PARMS on the start commands

1. The same identity is used for all servers using a RACF STARTED class resource where the JCL procedure is discrete, and the job name is generic.
2. The same identity is used for all servers using a RACF STARTED class resource where both the JCL procedure and job names are generic
3. Different identities are used for each server using a RACF STARTED class resource where both the JCL procedure and job name are discrete.
4. Different identities are used for each server using a RACF STARTED class resource where the JCL procedure is discrete and job name is generic.

It's possible to use a combination of the above, even under the same WLP_USER_DIR. So there's no "one best answer" here. What's best is what's best for you.



z/OS Security: The Angel process – what is this about?



```
com.ibm.ws.zos.core.angelRequired=true  
com.ibm.ws.zos.core.angelName=<name>  
com.ibm.ws.zos.core.angelRequiredServices=SAFCRED,ZOSWLM,TXRRS,ZOSDUMP
```

The Angel Process is a started task that is used to protect access to z/OS privileged or authorized services. This is done with SAF SERVER profiles.

- Authorized services include: WOLA, SAF, WLM, RRS, DUMP
- The ability to start multiple Angel processes on an LPAR was introduced in 16.0.0.4. This is called "Named Angels". It provides a way to separate Angel usage between Liberty servers:
 - An Angel process can be started with a NAME='<name>' parameter (or it can be started as a "default" without a name). The name may be up to 54 characters.
 - Liberty servers can be pointed at a specific Angel with a bootstrap property

Best practice:

- You may create separate named Angels for isolation of Test and Production, but do not take this practice too far. A few Angels, yes; dozens, no.
- Establish automation routines to start the Angels at IPL
- Grant SAF GROUP access to the SERVER profiles, then connect server IDs as needed

List of current Liberty Features

https://www.ibm.com/support/knowledgecenter/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/rwlp_feat.html

z/OS Security: SAF SERVER profiles related to the Angel



Best practice:

- Establish all the SERVER profiles ahead of time. Existence of profile does not grant access; READ access does.
- Determine what access a server needs and grant only that; check "is available" messages in messages.log to verify

Tech/Tip: The SAFLOG parameter was added in a recent Liberty service. If this parameter is set to Y, additional security related messages will be written to the JES messages and console if a Liberty server does not have authorization to use an angel-controlled privileged function. See URL

https://www.ibm.com/support/knowledgecenter/SS7K4U_liberty/com.ibm.websphere.wlp.zseries.doc/ae/rwlp_newinrelease.html

Liberty 21.0.6 add a new property to identify required services, com.ibm.ws.zos.core.angelRequiredServices, for more details see URL

<https://www.ibm.com/docs/en/was-liberty/zos?topic=overview-process-types-zos>

z/OS Authorized service security: Angel Required Services

To use z/OS authorized services, you must have a Liberty Angel process and grant access for your Liberty server's SAF identity to use these services.

- LOCALCOM - Required to use *WebSphere Optimized Local Adapters* (WOLA).
- PRODMGR - Required to use IFAUSAGE services for SMF reporting.
- SAFCRED - Required to use SAF authorized user registry services and SAF authorization services.
- TXRRS - Required by the IBM® MQ resource adapter when the connection to IBM MQ is made in BINDINGS mode
- WOLA - Required to use *WebSphere Optimized Local Adapters* (WOLA).
- ZOSAIO - Required to use AsyncIO on z/OS. **It is a good practice to enable this service**
- ZOSDUMP - Only required if asked to obtain an SVC dump by IBM service. It provides access to SVCDUMP services.
- ZOSWLM - Required to use WLM services. For more information, see [Measuring API workloads with WLM](https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=considerations-measuring-api-workloads-wlm) at URL <https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=considerations-measuring-api-workloads-wlm>

- When a Liberty server connects to an angel process during server startup, it checks that the server's identity has access to the z/OS authorized services. By default, access checks are performed for all authorized services.
- You can restrict the Liberty server to check and use only the authorized services it requires, which then makes other authorized services unavailable by using property, **com.ibm.ws.zos.core.angelRequiredServices**
- The value for this property, **com.ibm.ws.zos.core.angelRequiredServices**, must be a comma-separated list of valid angel process services, as described above. This property must be specified with the **com.ibm.ws.zos.core.angelRequired** property set to **true**. Only these services, when properly specified, are the ones used by the server.

Lack of access to the angel process itself or any of these listed required services will cause a server startup failure.

Tech/Tip: Sample RACF Commands for SERVER resources

```
RDEFINE SERVER BBG.ANGEL.angelName UACC(NONE) OWNER(SYS1)
PERMIT BBG.ANGEL.angelName CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.SAFCRED UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.SAFCRED CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSWLM UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSWLM CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.TXRRS UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.TXRRS CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSDUMP UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSDUMP CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.SECPFX.BBGZDFLT UACC(NONE)
PERMIT BBG.SECPFX.BBGZDFLT CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.WOLA UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.WOLA CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.LOCALCOM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.LOCALCOM CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSCFM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSCFM CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSCFM.WOLA UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSCFM.WOLA CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.PRODMGR UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.PRODMGR CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSAIO UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSAIO CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
SETROPTS RACLIST(SERVER) REFRESH
```

Example: Use the *bootstrap.properties* file to specify the required z/OS privileges



zceesrv1's bootstrap.properties

```
httpPort=9080
httpsPort=9443
ipicPort=1491
host=*
cicsHost=wg31.washington.ibm.com
network=ZOSCONN1
applid=ZOSCONN1
com.ibm.ws.zos.core.angelName=namedAngel
com.ibm.ws.zos.core.angelRequired=true
com.ibm.ws.zos.core.angelRequiredServices=SAFCRED,ZOSWLM,PRODMGR,ZOSAIO,TXRRS,LOCALCOM
```

zceesrv2's bootstrap.properties

```
httpPort=9090
httpsPort=9453
ipicPort=1492
host=wg31.washington.ibm.com
cicsHost=wg31.washington.ibm.com
network=ZOSCONN2
applid=ZOSCONN2
com.ibm.ws.zos.core.angelName=namedAngel
com.ibm.ws.zos.core.angelRequired=true
com.ibm.ws.zos.core.angelRequiredServices=SAFCRED,ZOSWLM,PRODMGR,ZOSAIO,TXRRS,LOCALCOM
```

Tech-Tip: SAF APPL and EJBRole Resources

Connect z/OS Connect users to a common group

CONNECT (FRED,USER1,JOHNSON) GROUP(ZCEEUSR)

Define a APPL profile for the server's SAF profilePrefix and permit access

RDEFINE APPL BBGZDFLT UACC(NONE) OWNER(SYS1)

PERMIT BBGZDFLT CLASS(APPL) ACCESS(READ) ID(WSGUEST#, ZCEEUSR)

SETROPTS RACLIST(APPL) REFRESH

Example of defining an EJBROLE profile for the server's SAF profilePrefix and permit access

RDEFINE EJBROLE BBGZDFLT.zos.connect.access.roles.zosConnectAccess OWNER(SYS1) UACC(NONE)

PERMIT BBGZDFLT.zos.connect.access.roles.zosConnectAccess +

CLASS(EJBROLE) ID(ZCEEUSR) ACCESS(READ)

SETROPTS RACLIST(EJBROLE) REFRESH

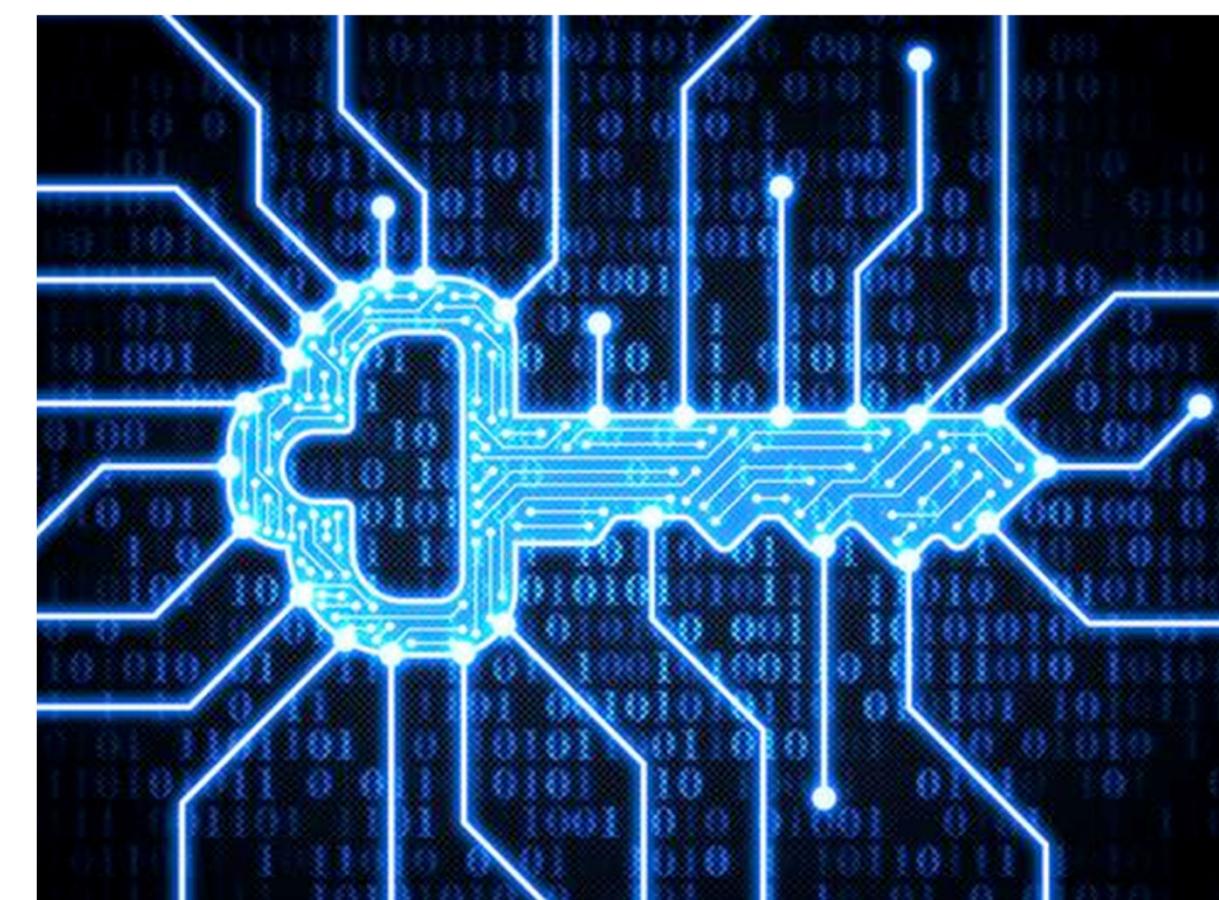
```
<safCredentials unauthenticatedUser="WSGUEST" profilePrefix="BBGZDFLT"/>
<safAuthorization racRouteLog=ASIS reportAuthorizationCheckDetails="true"/>
```

- # https://www.ibm.com/support/knowledgecenter/SS7K4U_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp_config_security_saf.html
<https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=registry-saf-unauthenticated-user-id>

General security terms or considerations

Security involves

- Identifying who or what is requesting access (**Authentication**)
 - Basic Authentication
 - Mutual Authentication using Transport Layer Security (TLS), formerly known as SSL
 - Third Party Tokens
- Ensuring that the message has not been altered in transit (**Data Integrity**) and ensuring the confidentiality of the message in transit (**Encryption**)
 - TLS (encrypting messages and using a digital signature)
- Controlling access (**Authorization**)
 - Is the authenticated identity authorized to access to z/OS Connect
 - Is the authenticated identity authorized to access a specific API, Services, etc.

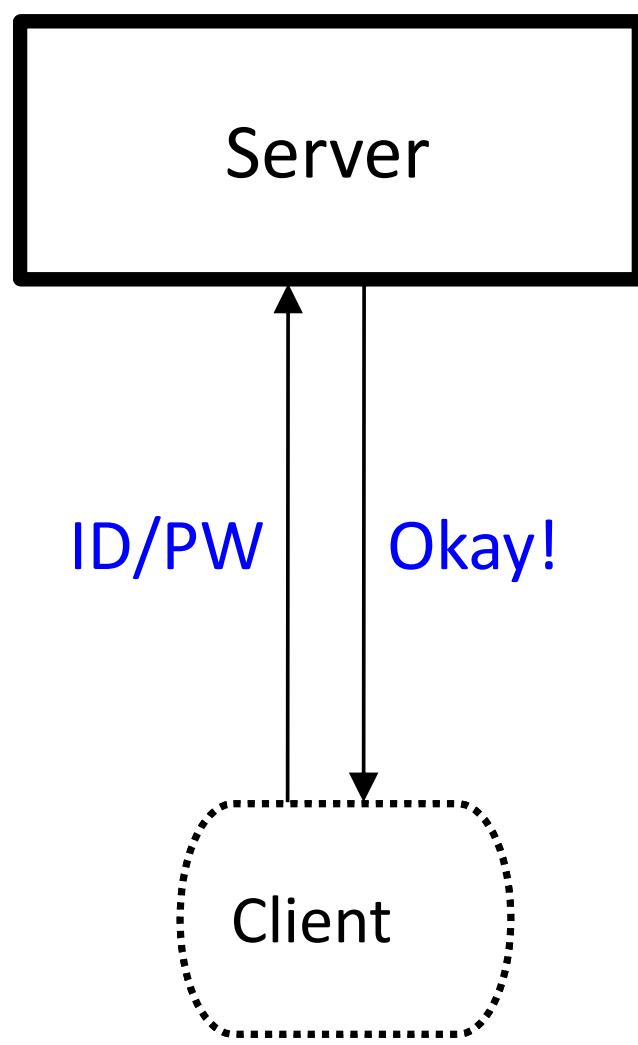




Liberty Authentication Options

Several different ways this can be accomplished:

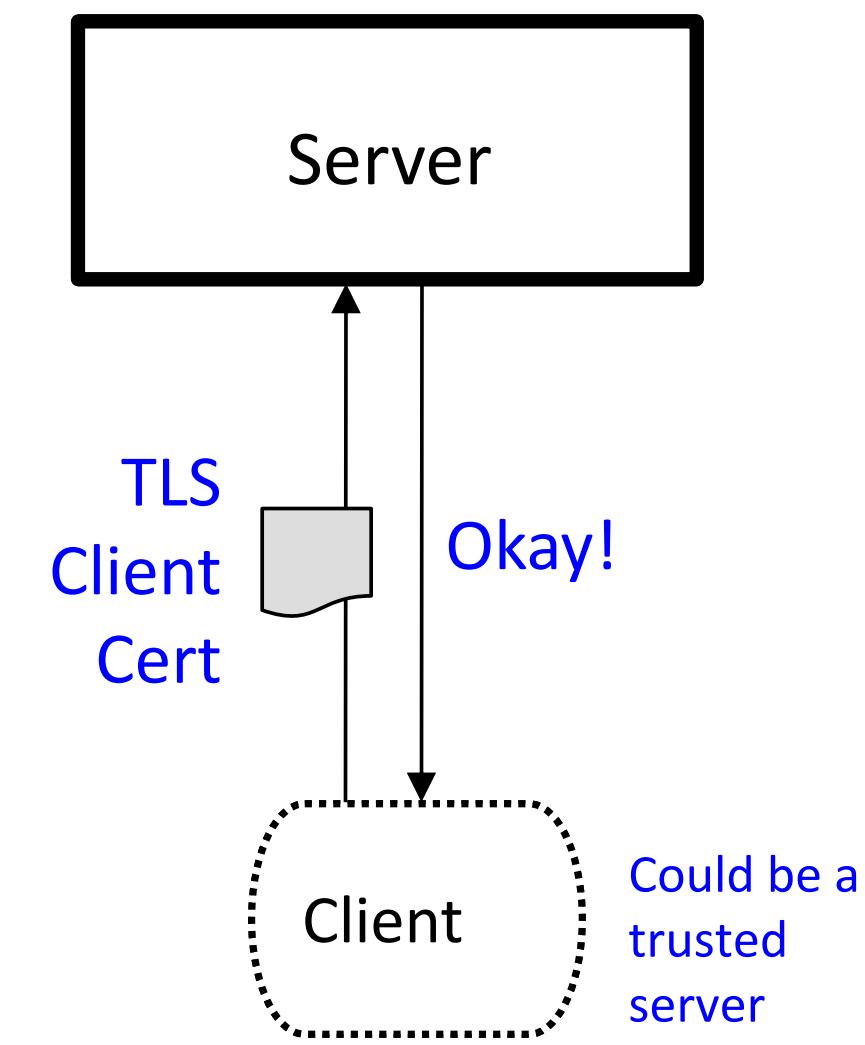
Basic Authentication



Client supplies ID/PW or ID/PassTicket

- Server checks registry:**
- Basic (server.xml)
 - SAF

Client Certificate

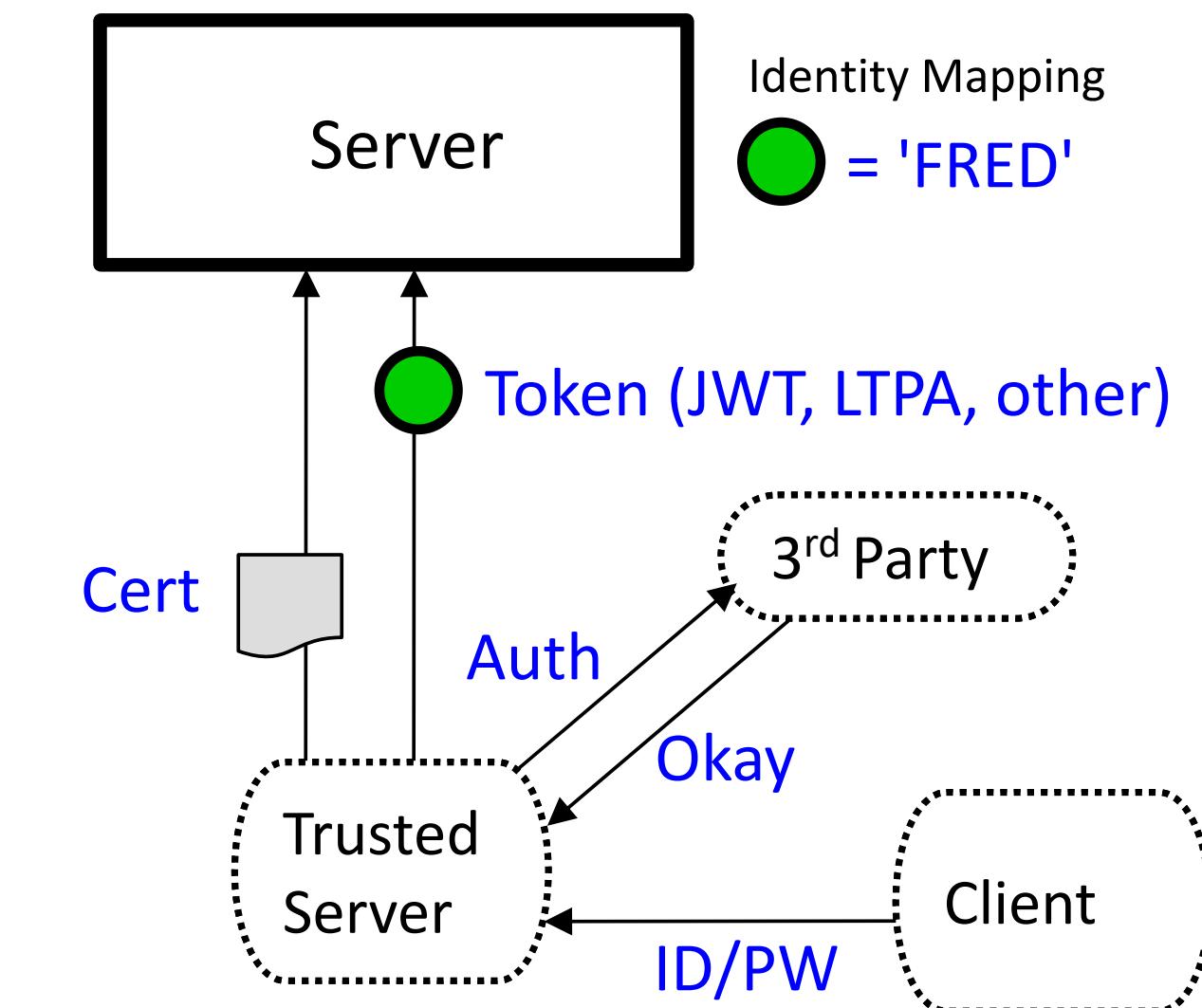


Client supplies client personal certificate

Server validates client personal certificate and maps it to an identity

- Registry options:**
- SAF

Third Party Authentication



Client authenticates to 3rd party sever

Client receives a trusted 3rd party token

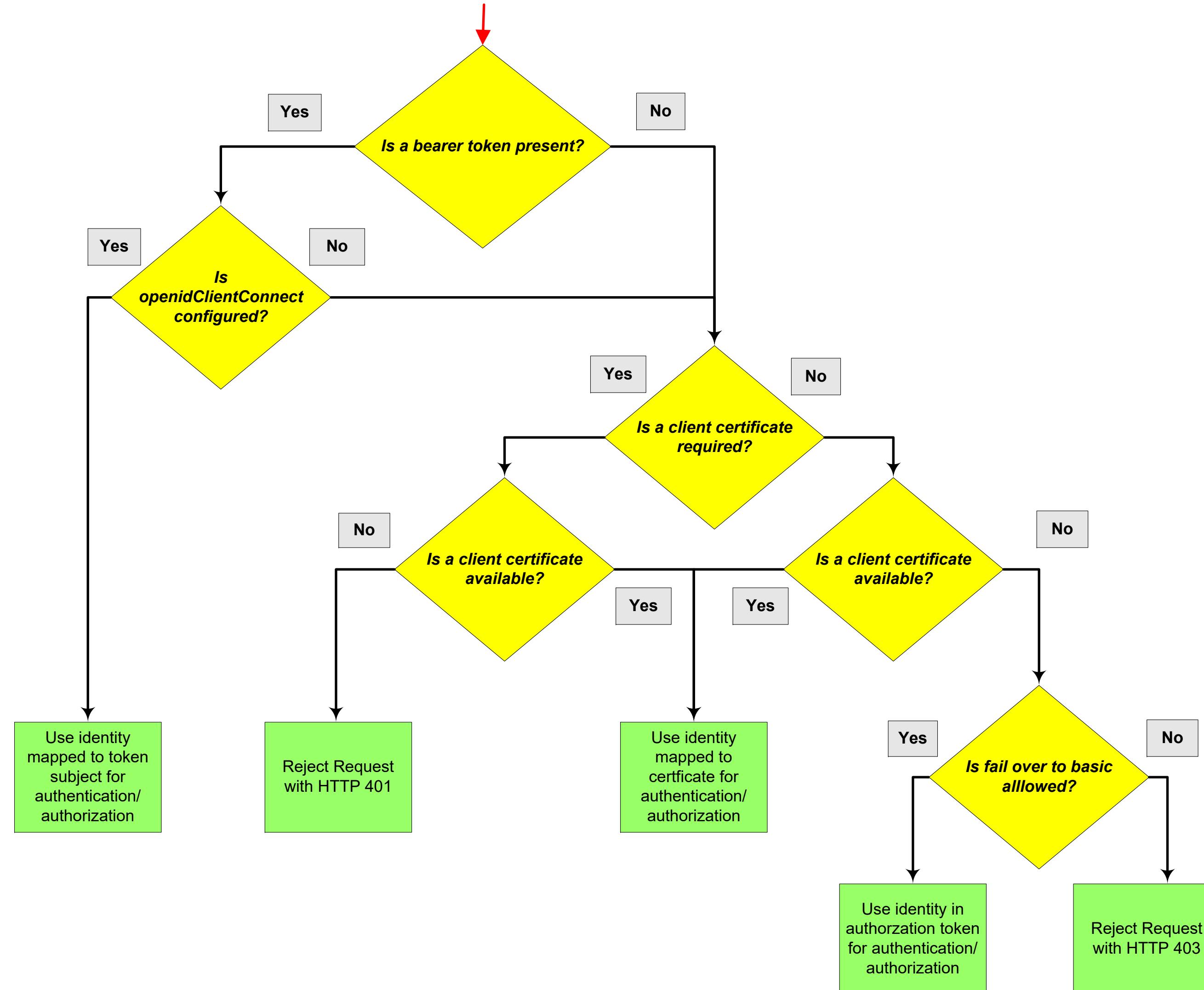
Token flows to server and is mapped to an identity

Registry options:

- We may not need to know these details.



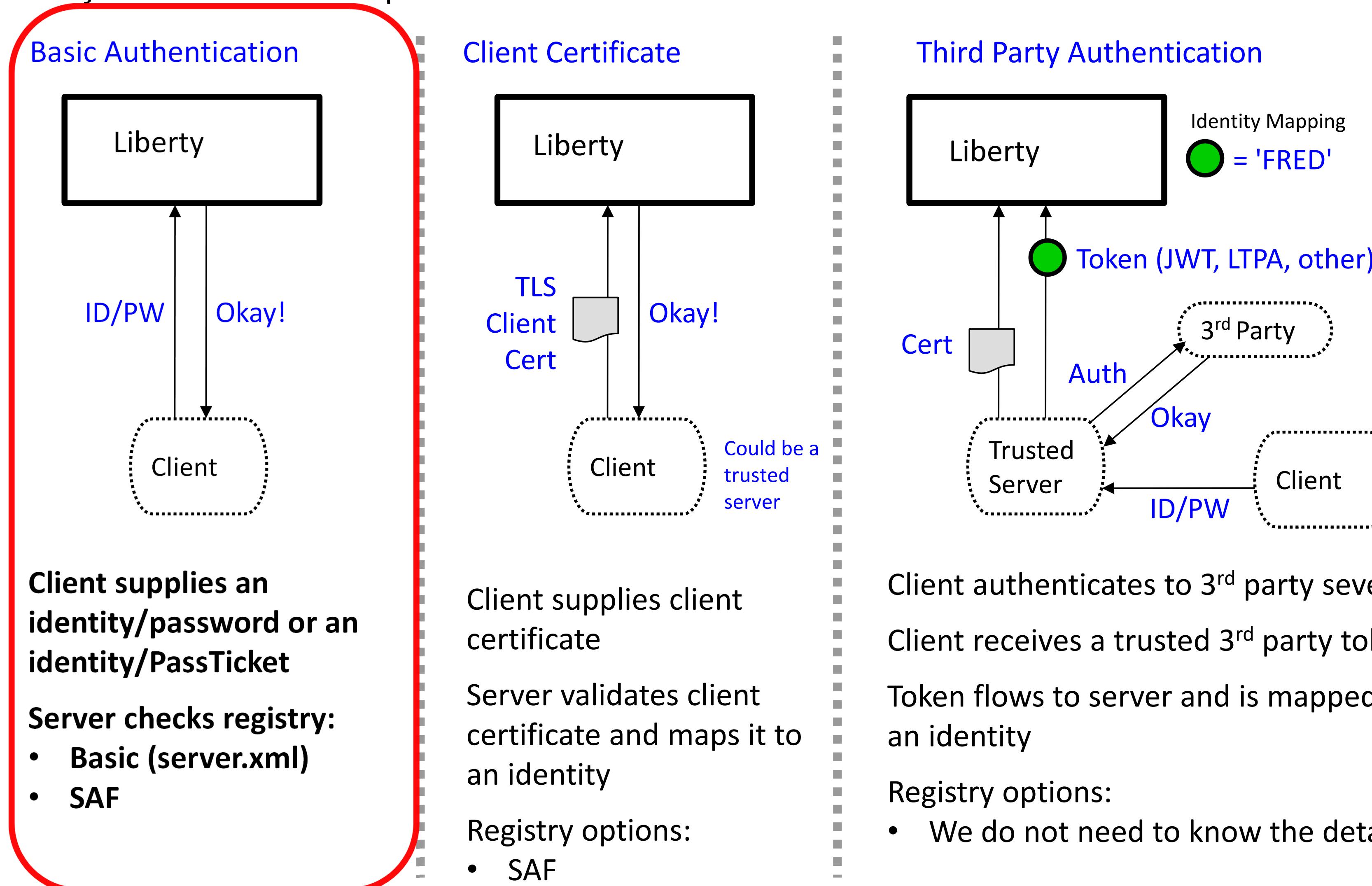
Authentication credential precedence order for determining authorization identity





Authentication - Basic Authentication

Several different ways this can be accomplished:





Basic authentication – Where the client provides an identity and password

- ❑ server XML security configuration:

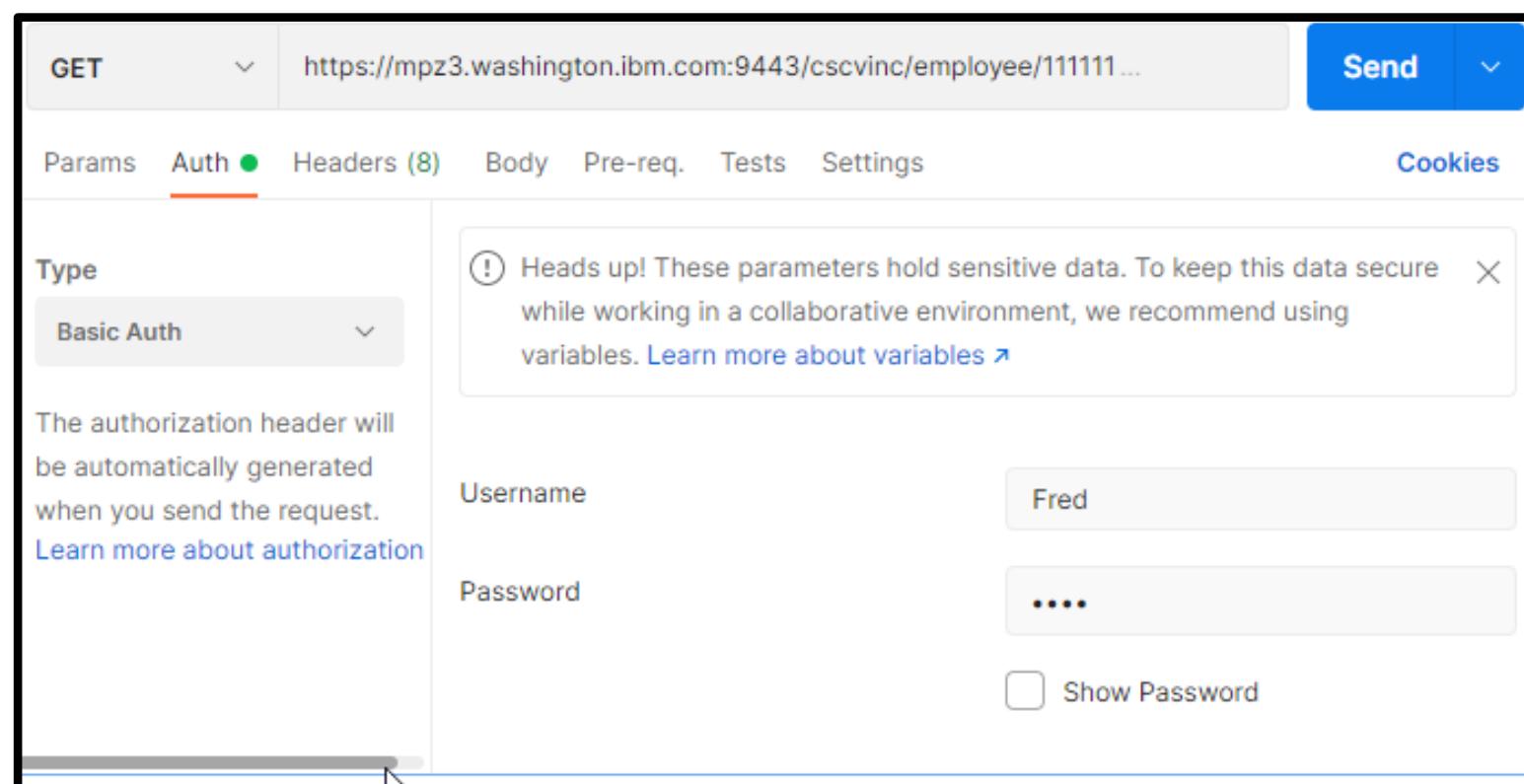
```
<featureManager>
  <feature>appSecurity-2.0</feature>
  <feature>zosSecurity-1.0</feature>
</featureManager>

<webAppSecurity allowFailOverToBasicAuth="true" />

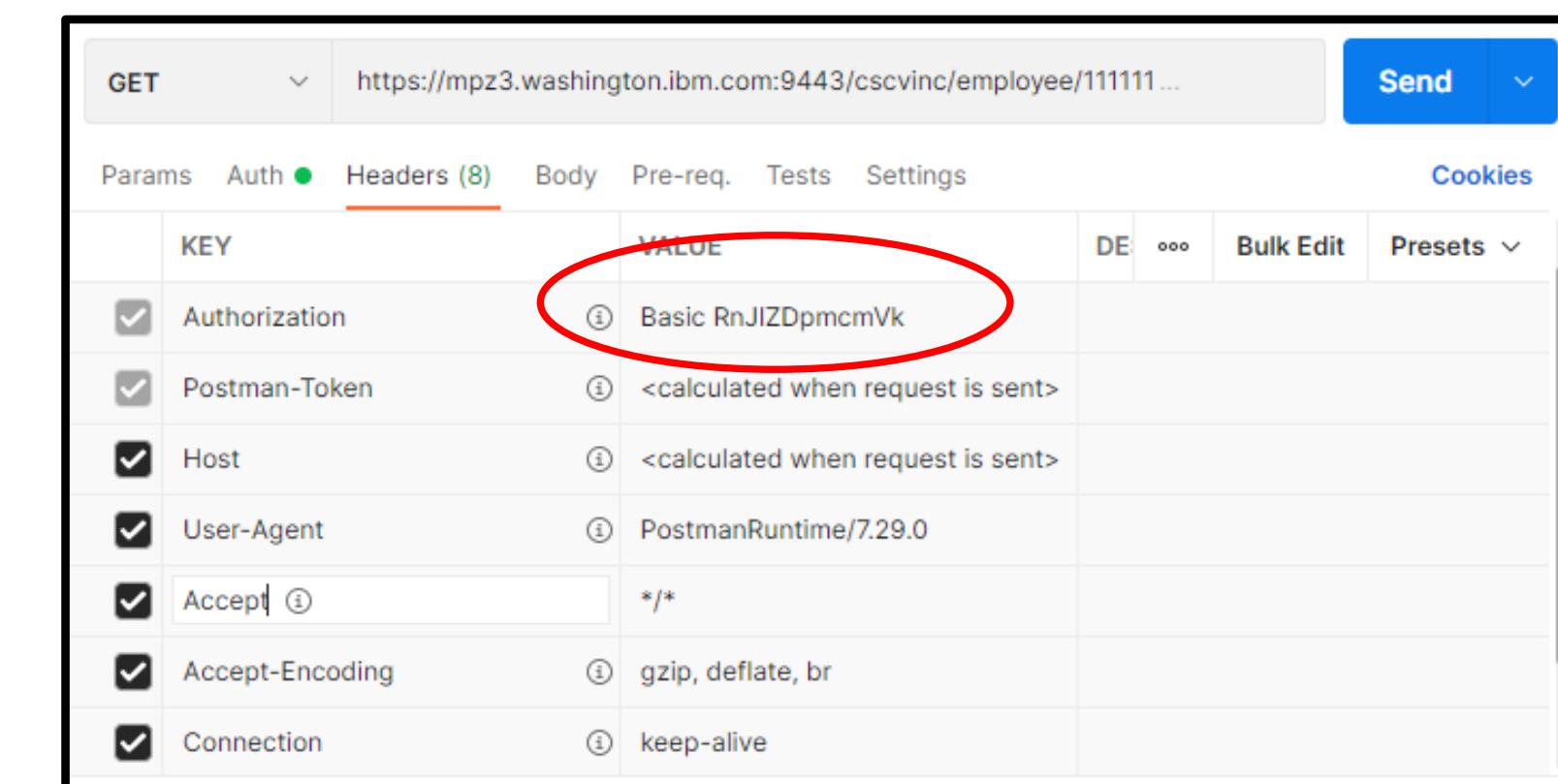
<safRegistry id="saf" />
<safAuthorization racRouteLog="ASIS" />
<safCredentials unauthenticatedUser="WSGUEST"
  profilePrefix="BBGZDFLT" />
```

- ❑ When sending a request to a Liberty server, basic authentication information (identity and password) is provided in the HTTP header in a *Basic Authorization* token with the identity and password encoded or formatted using Base64.

- An example with Postman:



The screenshot shows the Postman interface for a GET request to <https://mpz3.washington.ibm.com:9443/cscvinc/employee/111111...>. The 'Auth' tab is selected, and 'Basic Auth' is chosen from the dropdown. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables.' Below the dropdown, it says: 'The authorization header will be automatically generated when you send the request.' Fields for 'Username' (Fred) and 'Password' (redacted) are filled. A 'Show Password' checkbox is present.



The screenshot shows the Postman interface with the 'Headers' tab selected. A red circle highlights the 'Authorization' header row, which has a value of 'Basic RnJZDpmcmVk'. Other headers listed include Postman-Token, Host, User-Agent, Accept, Accept-Encoding, and Connection.

KEY	VALUE
Authorization	Basic RnJZDpmcmVk
Postman-Token	<calculated when request is sent>
Host	<calculated when request is sent>
User-Agent	PostmanRuntime/7.29.0
Accept	/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive



There are multiple ways to provide an identity and password

- When sending a request to a Liberty server running z/OS Connect, basic authentication information (identity and password) is provided in the HTTP header in a Basic Authorization token with the identity and password encoded or formatted using Base64.
 - Examples using the API Explorer feature , cURL, and a Java client.

The screenshot shows the IBM API Explorer interface for the 'cscvinc' service. On the left, there's a list of operations: POST /cscvinc/employee, DELETE /cscvinc/employee/{employee}, and GET /cscvinc/employee/{employee}. The GET operation is selected. On the right, there's a 'Model' section with a JSON schema for an employee object. Below it, a 'Response Content Type' dropdown is set to 'application/json'. Under 'Parameters', there are two entries: 'Authorization' with value 'Basic dXNlcpwYXNzd29yZA==' and 'employee' with value '000050'. A red box highlights the 'Authorization' parameter. A modal dialog box titled 'mpz3.washington.ibm.com:9443' appears, asking for sign-in with 'Username' (user1) and 'Password' (*****). A red box highlights this dialog. To the right of the dialog, a 'curl' command is shown in a terminal window:

```
c:\z>curl -X GET --user FRED:FRED --insecure https://mpz3.washington.ibm.com:9443/cscvinc/employee/111111 {"cscvincSelectServiceOperationResponse": {"cscvincContainer": {"response": {"CEIBRESP": 0, "CEIBRESP2": 0, "USERID": "CICSUSER", "file": {"employee": {"id": "111111", "name": "C. BAKER", "address": "OTTAWA, ONTARIO", "phoneNumber": "51212003", "date": "26 11 81", "amount": "$0011.00"}}}}}
```

The screenshot shows a Java code editor with a file named 'ZeeGet.java'. The code is a simple HTTP client using the URL class and HttpURLConnection. It sets the response content type to 'application/json' and encodes the 'Authorization' header using Base64. A red box highlights the 'Authorization' header line. The code is as follows:

```
49 URL url = new URL("https://wg31.washington.ibm.com:9453/db2/department?dept1=C01&dept2=C01");
50 System.out.println("URL: " + url);
51 HttpsURLConnection conn = (HttpsURLConnection) url.openConnection();
52 conn.setRequestMethod("GET");
53 conn.setRequestProperty("Content-Type", "application/json");
54 byte[] bytesEncoded = Base64.encodeBase64("Fred:fredpwd".getBytes());
55 conn.addRequestProperty("Authorization", new String(bytesEncoded));
56
57 try {
58
59     if (conn.getResponseCode() != 200) {
60         throw new RuntimeException("Failed : HTTP error code : " + conn.getResponseCode());
61     }
62     BufferedReader bufferedReader = new BufferedReader(new InputStreamReader((conn.getInputStream())));
63     String output;
64     StringBuilder stringBuffer = new StringBuilder();
65     while ((output = bufferedReader.readLine()) != null) {
66         stringBuffer.append(output);
67     }
68     JSONObject json = new JSONObject(stringBuffer.toString());
69     JSONArray jsonArray = json.getJSONArray("ResultSet 1 Output");
70     JSONObject jsonEntry = new JSONObject();
71     for (int index = 0; index < jsonArray.length(); index++) {
72         jsonEntry = jsonArray.getJSONObject(index);
73         if (jsonEntry.has("employeeNumber")) {
```

Tech-TIP: A RACF PassTicket provides an alternative to a password

- A PassTicket is generated by or for a client by using a secured sign-on key (whose value is masked or encrypted) to encrypt a valid *RACF identity* combined with the *application name* of the targeted resource. Also embedded in the PassTicket is a time stamp (based on the current Universal Coordinated Time (UCT)) which sets the time when the PassTicket will expire (usually 10 minutes).
- Access to PassTickets is managed using the RACF PTKTDATA class.
- For z/OS Connect, a RACF PassTicket can be used for basic authentication when connecting from any REST client on any platform to a z/OS Liberty server and for requests from a z/OS Connect server accessing IMS and Db2.
- ***PassTickets do not have to be generated on z/OS using RACF services.*** IBM has published the algorithm used to generate a PassTickets, see manual *z/OS Security Server RACF Macros and Interfaces, SA23-2288-40*. *Github has examples using Java, Python and other example are available on other sites.*



Tech/Tip: RACF resources for using PassTickets

- ❑ A PTKTDATA resource is defined using the *appName* assigned to the target subsystem:

```
RDEFINE PTKTDATA appName SSIGNON(KEYMASK(keymaskValue))
    APPLDATA('NO REPLAY PROTECTION')
```

Where:

- appName* is an application name assigned to the resource, e.g., BBGZDFLT
keymaskValue is the value of the secured sign-on application key, a 64-bit hex value
replayProtection indicates if a pass ticket can be reused

- ❑ Access to using PassTickets is controlled by another PTKTDATA resource, *IRRPTAUTH.appName.identity*. UPDATE access is required. For example, to use PassTickets to access a z/OS Connect server the resources below need to be defined and access permitted.

```
<safRegistry id="saf" />
    <safAuthorization racRouteLog="ASIS" />
    <safCredentials unauthenticatedUser="WSGUEST"
        profilePrefix="BBGZDFLT" />
```

```
RDEFINE PTKTDATA BBGZDFLT SSIGNON(0123456789ABCDEF)
    APPLDATA('NO REPLAY PROTECTION') UACC(NONE)
RDEFINE PTKTDATA IRRPTAUTH.BBGZDFLT.* UACC(NONE)
PERMIT IRRPTAUTH.BBGZDFLT.* ID(LIBSERV) CLASS(PTKTDATA) ACCESS(UPDATE)
PERMIT IRRPTAUTH.BBGZDFLT.USER1 ID(USER1) CLASS(PTKTDATA) ACCESS(UPDATE)
```



z/OS Connect Security server XML Authentication Configuration (OpenAPI 2)

- requireAuth - requires the client to provide credentials

```
<zosconnect_zosConnectManager  
    requireAuth="true|false"  
    requireSecure="true"/>  
  
<zosconnect_zosConnectAPIs>  
    <zosConnectAPI name="catalog"  
        requireAuth="true|false"  
        requireSecure="true"/>  
</zosconnect_zosConnectAPIs>  
  
<zosconnect_services>  
    <service id="selectByEmployee"  
        name="selectEmployee"  
        requireAuth="true|false"  
        requireSecure="true"/>  
</zosconnect_services>  
  
<zosconnect_apiRequesters>  
    requireAuth="true|false"  
    <apiRequester name="cscvincapi_1.0.0"  
        requireAuth="true|false"  
        requireSecure="true"/>  
</zosconnect_apiRequesters>
```

Globally, requires that users specify security credentials to be authenticated order to access APIs, services and API requesters, unless overridden on the specific resource definitions.

Requires that users specify security credentials to be authenticated in order to access the API.

Requires that users specify security credentials to be authenticated in order to directly access the service. This attribute is ignored when the service is invoked from an API, then only the API requireAuth attribute is relevant.

Requires that users specify security credentials to be authenticated in order to access all API requesters. If the requireAuth attribute is not set, the global setting on the zosconnect_zosConnectManager element is used instead, unless the requireAuth attribute is overridden on the specific API requester.

The requireAuth attribute controls whether an inbound request must provide credentials using one of the three authentication methods, e.g., basic, client certificate, or third-party token.

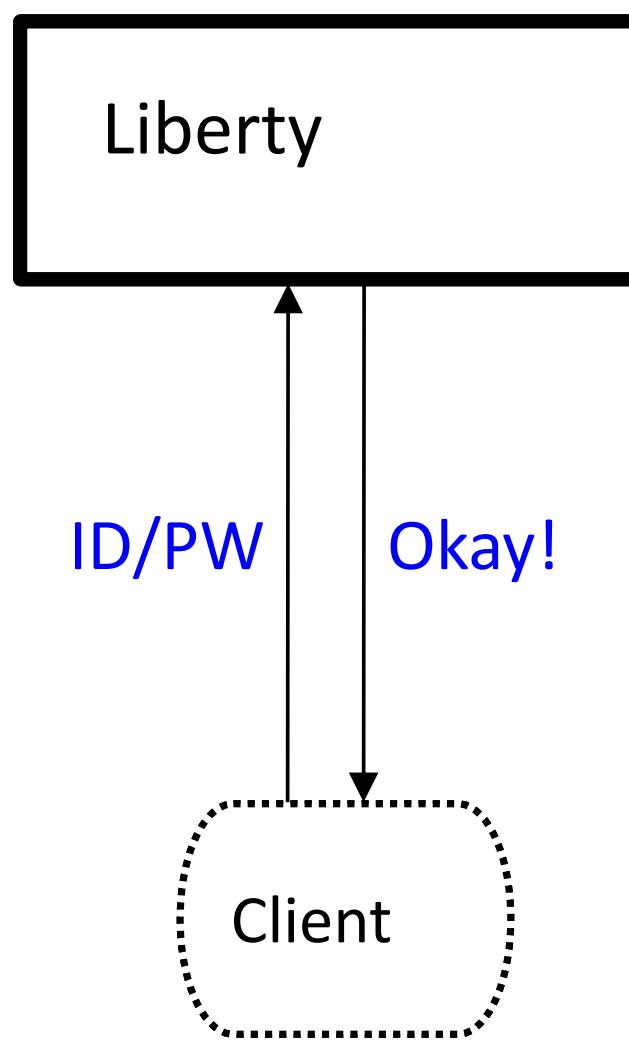
Note that there are no equivalent configuration elements for an z/OS Connect OpenAPI 3 server.



Authentication - TLS Mutual Authentication

Several different ways this can be accomplished:

Basic Authentication



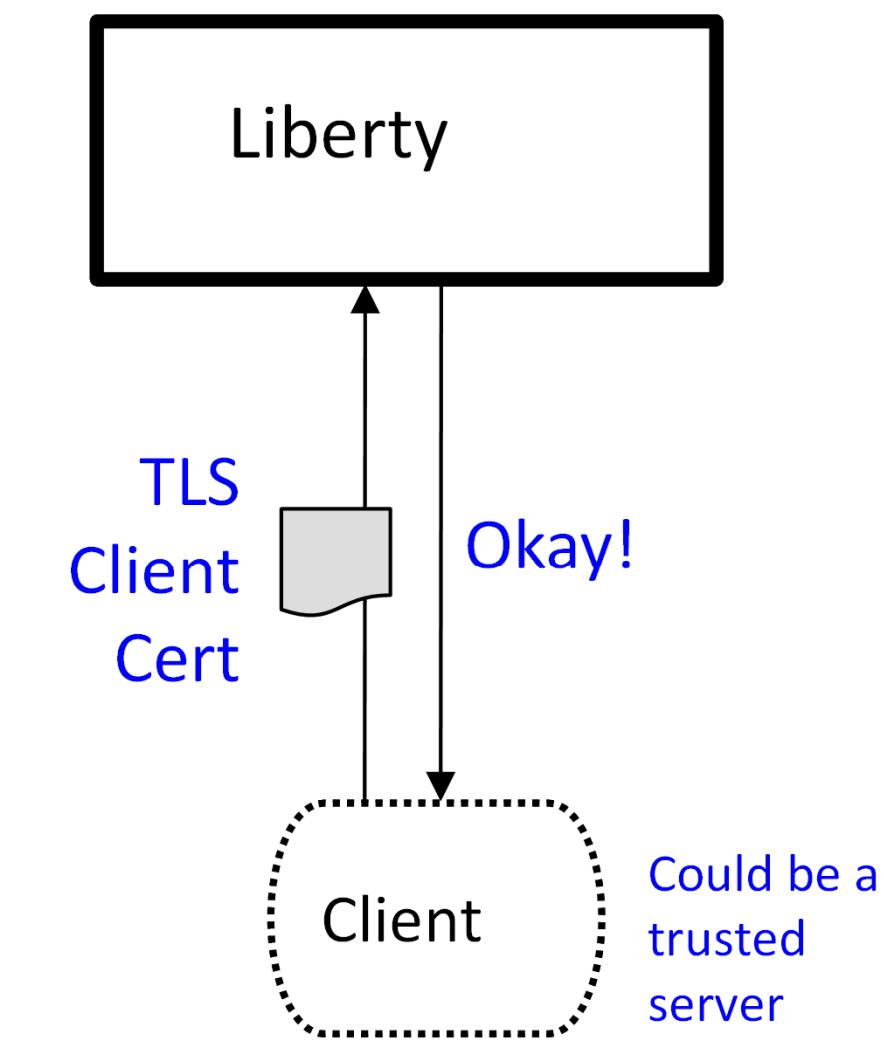
Server prompts for ID/PW

Client supplies ID/PW or ID/PassTicket

Server checks registry:

- Basic (server.xml)
- SAF

Client Certificate



Server prompts for client certificate.

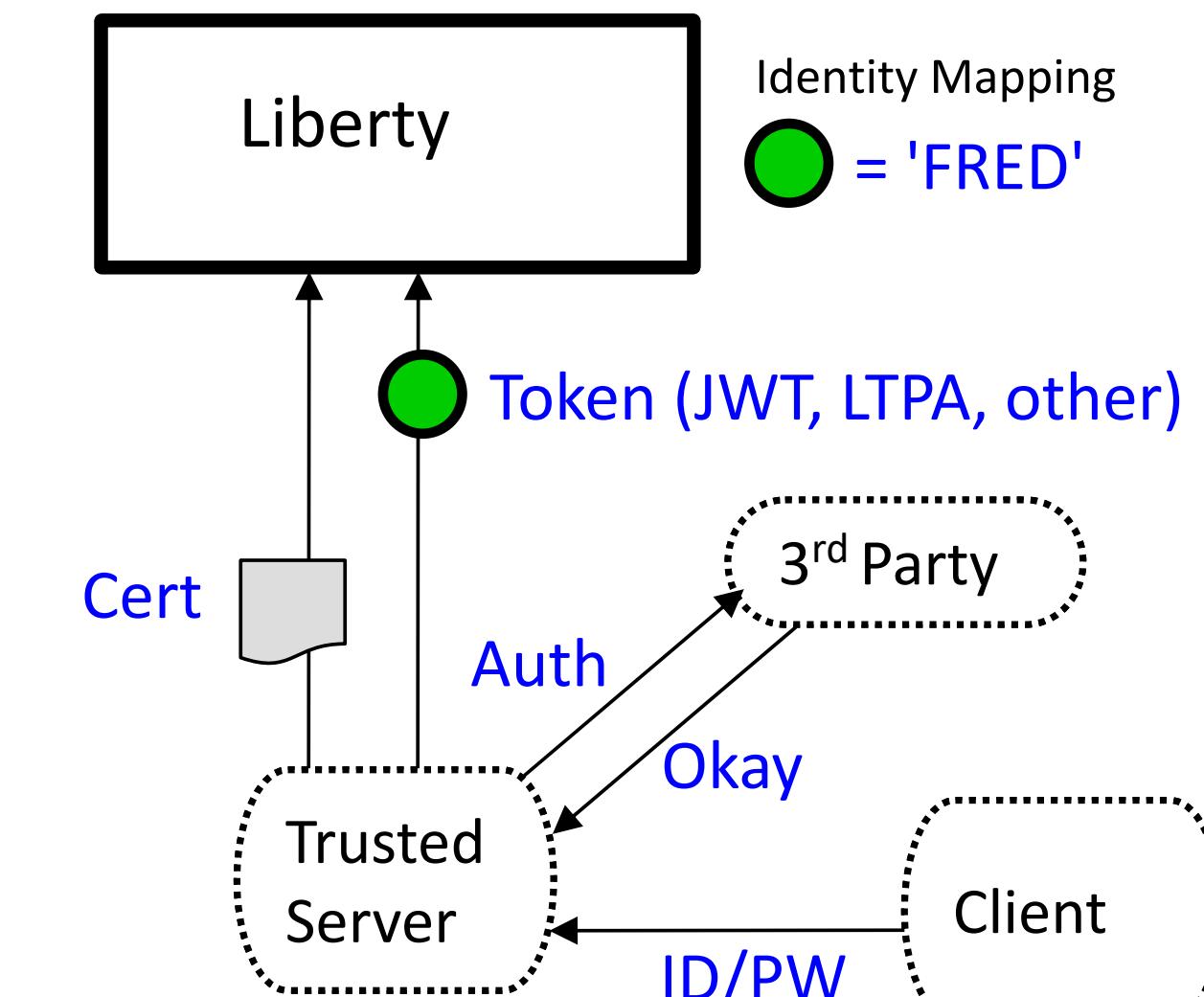
Client supplies personal certificate

Server validates client certificate and maps it to an identity

Registry options:

- SAF

Third Party Authentication



Client authenticates to 3rd party sever

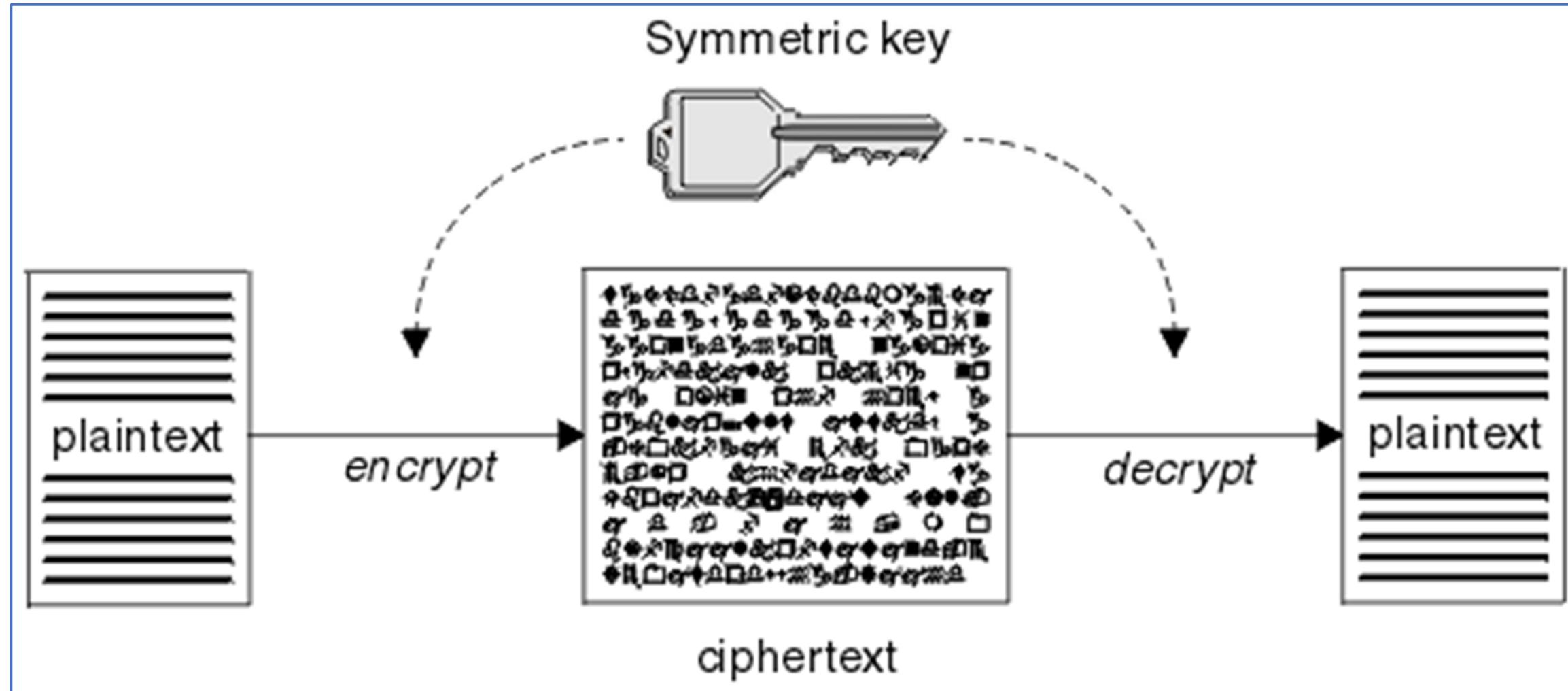
Client receives a trusted 3rd party token

Token flows to Liberty z/OS and is mapped to an identity

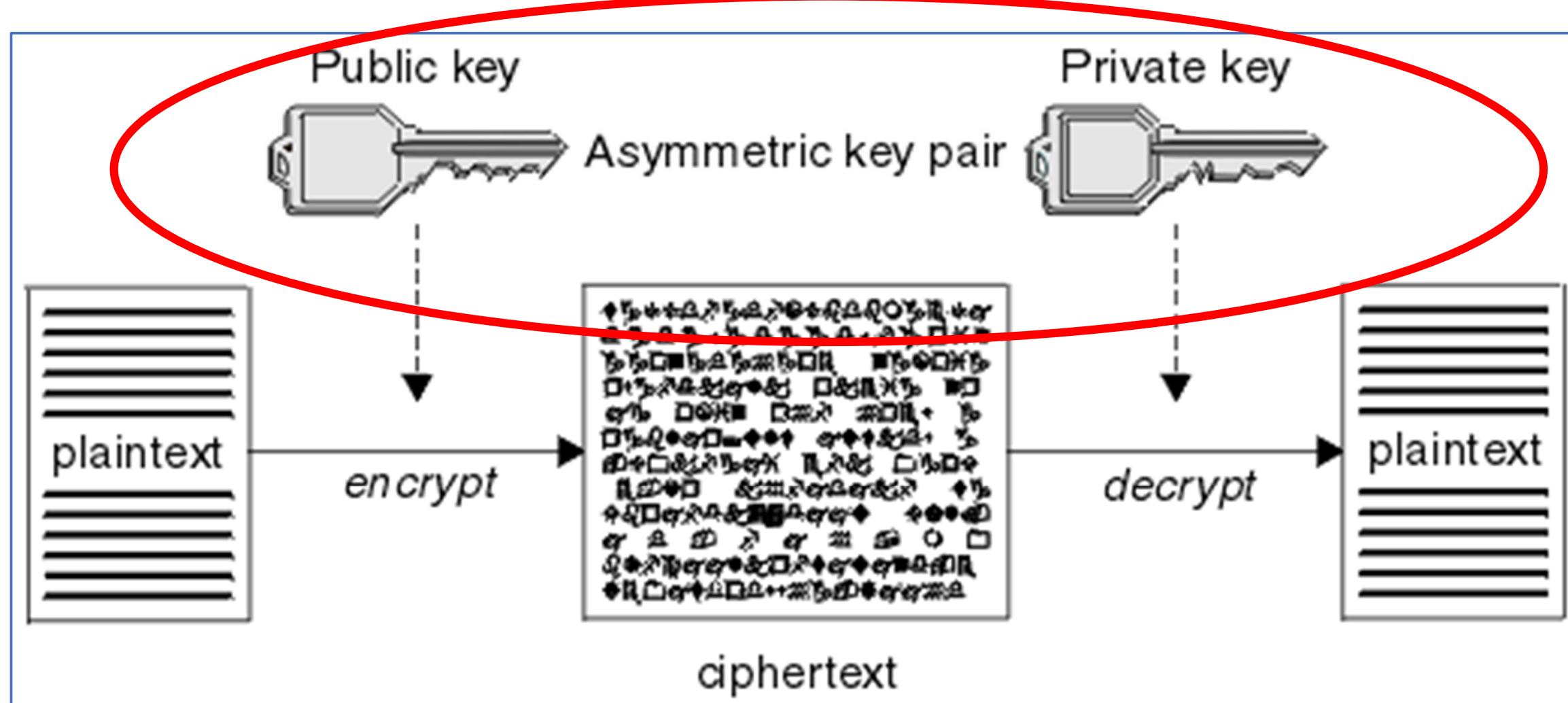
Registry options:

- We may not need to know these details.

Tech-Tip: Symmetric key v. Asymmetric key pairs



A symmetric key is a key shared by the endpoints. Both endpoints use the same key to encrypt and decrypt messages.



An asymmetric key pair is the preferred solution. There is no risk of compromise by sending a symmetric or shared key outside of a protected communication flow.

A message encrypted with a public key can only be decrypted by endpoint that has the private key. The privacy of the messages is ensured.

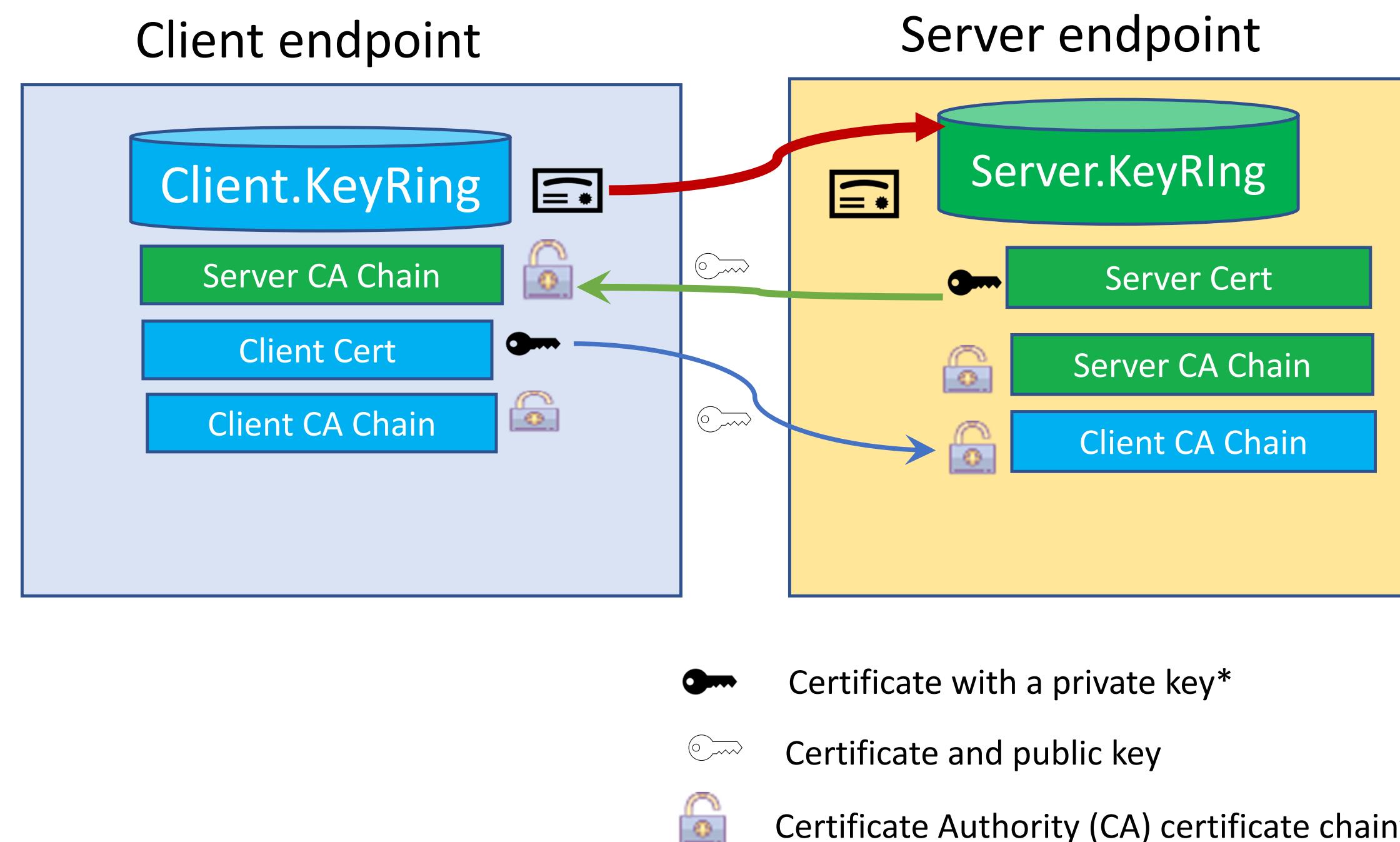
If an endpoint can successfully decrypt a message message encrypted received with a private key, the endpoint sending the message has successfully asserted its validity by proving it has the private used to encrypt the message.

The basic TLS Handshake Flow (HTTPS)

The HTTPS protocol involves a TLS handshake –

Server Authentication (always occurs when HTTPS is the protocol)

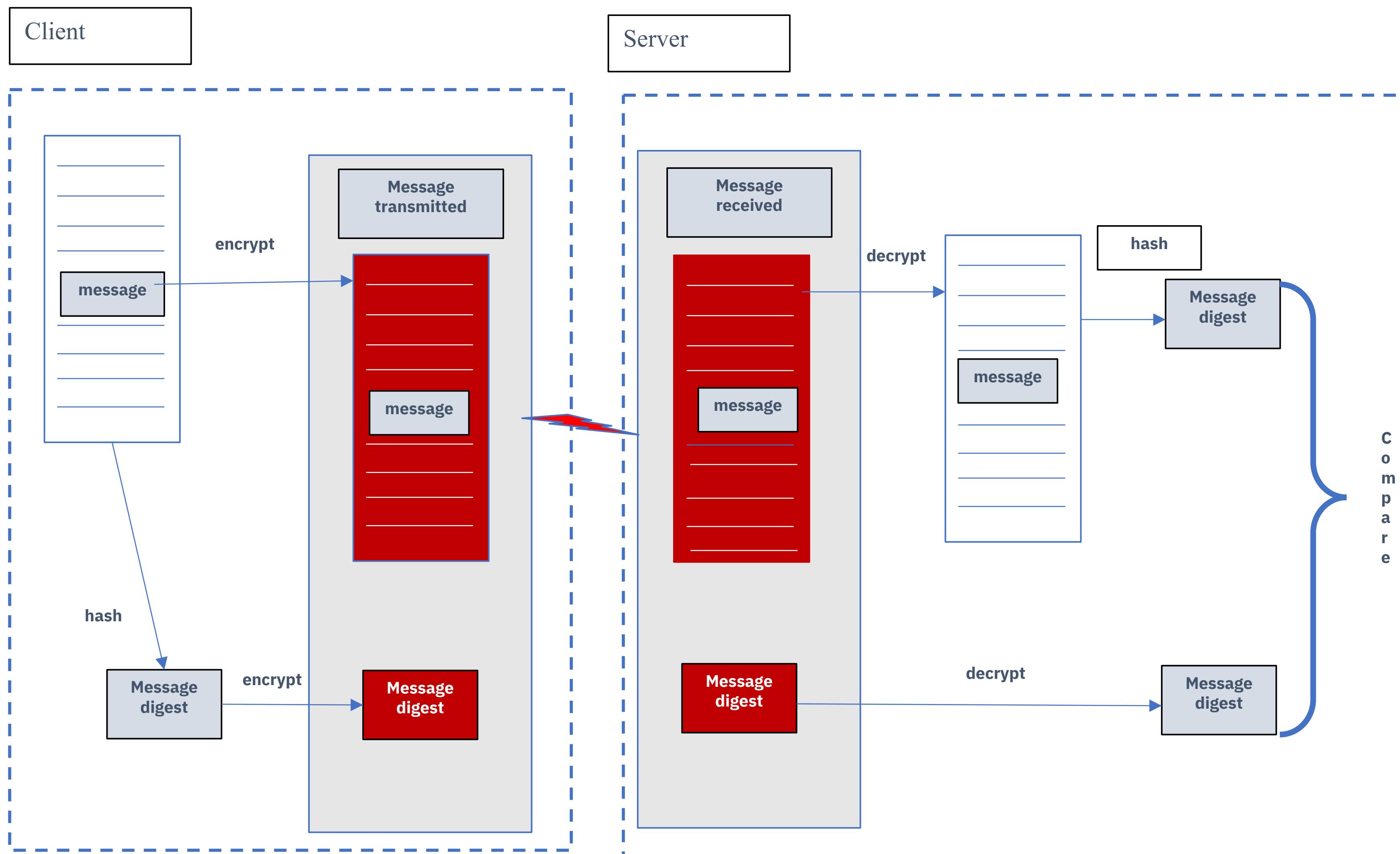
Mutual Authentication (optional, at the request of the server endpoint)



*For server and/or mutual authentication to work, the endpoint sending the client certificate must use a personal certificate with a private key. The private key is required to decrypt (or encrypt) a message digest that is sent from the other endpoint during the handshake flow. Generation of a message digest also requires access to the CA certificate used to sign the certificate.

#Refers to the set or of certificates used to issue the server or client personal certificate including any intermediate certificates up to and including the root CA.

Tech-Tip: Message Integrity and Encryption (client to server endpoint)



Tech-Tip: Accessing a certificate's private key in non-virtual key rings

Two types of RACF profiles resources are used to control access to key ring and certificates

- **RDATALIB** for controlling access to a specific key ring
- **FACILITY** for controlling access to key rings globally

User certificates (connected to the key ring with usage PERSONAL)

- Global profiles uses the **FACILITY** resource **IRR.DIGTCERT.LISTRING**:
 - **READ** access is required to access one's own key ring and private key
 - **UPDATE** access is required to access another user's key private key
- Specific key rings uses the RDATALIB class **<ring owner>.<ring name>.LST**
 - **READ** access is required to access one's own private key
 - **UPDATE** access is required to access another identity's private keys

CERTAUTH and SITE certificates (connected to the key ring with usage PERSONAL)

- Global profiles uses the **FACILITY** resource **IRR.DIGTCERT.GENCERT**:
 - **CONTROL** access is required to access a CERTAUTH or SITE certificate private key ring
- Specific key rings uses the RDATALIB class **<ring owner>.<ring name>.LST**
 - **CONTROL** access is required to access the private keys of CERTAUTH and SITE certificates

Remember: When switching from global FACILITY class profiles to specific ring RDATALIB class profiles, the RDATLIB resources will be checked first

Tech/Tip: RACF digital certificate (RACDCERT) command review

```
RACDCERT ID(LIBSERV) GENCERT SUBJECTSDN(CN('wg31.washington.ibm.com') +
O('IBM') OU('LIBERTY')) WITHLABEL('Liberty Server Cert') ALTNAMES(DOMAIN('wg31z.washington.ibm.com'))
RACDCERT ID(LIBSERV) GENREQ(LABEL('Liberty Server Cert')) DSN(CERT.REQ)
```

Send the certificate to your Certificate Authority to be signed

```
racdcert CERTAUTH withlabel('Liberty CA') add('USER1.LIBCA.PEM') TRUST
racdcert id(LIBSERV) withlabel('Liberty Server Cert') add('LIBSERV.P12') password('secret') TRUST
```

```
/* Create Liberty key ring and connect CA and personal certificates */
racdcert id(libserv) addring(Liberty.KeyRing)
racdcert id(libserv) connect(ring(Liberty.KeyRing) label('CICS CA') certauth usage(certauth))
racdcert id(libserv) connect(ring(Liberty.KeyRing) label('Liberty CA') certauth usage(certauth))
/* Connect default personal certificate */
racdcert id(libserv) connect(ring(Liberty.KeyRing) label('Liberty Client Cert') default
```

```
setropts raclist(digtcert) refresh
```

Broadcom Support web pages

Site of *What ACF2 security setup is needed for IBM's z/OS Connect Enterprise Edition V3.0?*

<https://knowledge.broadcom.com/external/article/128597/what-acf2-security-setup-is-needed-for-i.html>

Site of *ACF2 setup for z/OS Connect Enterprise Edition V3.0*

<https://knowledge.broadcom.com/external/article/142172/acf2-setup-for-zos-connect-enterprise-ed.html>

Site of *Setting up Liberty Server for z/OS with Top Secret*

<https://knowledge.broadcom.com/external/article/37272/setting-up-liberty-server-for-zos-with-t.html>

Tech/Tip: Anatomy of a RACF Personal Digital Certificate

Digital certificate information for user ATSSERV:

Label: **RPServer-Server**
Certificate ID: 2QfB4+Lixdn12dfihZmlhZlg4oWZpYZ
Status: **TRUST**
Start Date: 2020/11/12 00:00:00
End Date: **2029/12/31 23:59:59**
Serial Number:
 >01<
Issuer's Name:
 >**CN=RPServer-CertAuth.OU=CertAuth**<
Subject's Name:
 >**CN=RPServer-Server.OU=ATS.O=IBM.C=USA**<
Subject's AltNames:
 Domain: **wg31.washington.ibm.com** ←
Signing Algorithm: sha1RSA
Key Type: RSA
Key Size: 2048
Private Key: **YES**
Ring Associations:
 Ring Owner: ATSSERV
 Ring:
 >**RpServer.KeyRing**<
 Ring Owner: LIBSERV
 Ring:
 >**RpServer.KeyRing**<

Some clients are more sensitive than others when checking common names (CN). They will check the endpoint's actual host name versus the CN and if they do not match, the certificate is rejected. The *AltName* attribute can be used to resolve this issue.

Tech/Tip: RACF Certificate Filtering and Mapping

Filters for mapping certificates can be created with a RACDCERT command.

- Enter command RACDCERT ID MAP to create a filter that assigns RACF identity ATSUSER to any digital certificate signed with the ATS client signer certificate and where the subject is organizational unit ATS in organization IBM.

```
racdcert id(atsuser) map sdnfilter('OU=ATS.O=IBM')
idnfilter('CN=ATS Client CA.OU=ATS.O=IBM') withlabel('ATS USERS')
```

- Enter command RACDCERT ID MAP to create a filter that assigns RACF identity OTHUSER to any digital certificate signed by the ATS client signer certificate and where the subject is in organization IBM.

```
racdcert id(othuser) map sdnfilter('O=IBM')
idnfilter('O=IBM') withlabel('IBM USERS')
```

- Refresh the in-storage profiles for digital certificate maps.

```
SETRPTS RACLST(DIGTNMAP) REFRESH
```



Liberty JSSE (HTTPS) server XML configuration

```
<!-- Enable features -->
<featureManager>
    <feature>transportSecurity-1.0</feature>
</featureManager>

<sslDefault sslRef="DefaultSSLSettings"
    outboundSSLRef="OutboundSSLSettings" />

<ssl id="DefaultSSLSettings"
    keyStoreRef="CellDefaultKeyStore"
    trustStoreRef="CellDefaultKeyStore"
    clientAuthenticationSupported="true"
    clientAuthentication="true"/>

<keyStore id="CellDefaultKeyStore"
    location="safkeyring:///Liberty.KeyRing"
    password="password" type="JCERACFKS"
    fileBased="false" readOnly="true" />

<ssl id="OutboundSSLSettings"
    keyStoreRef="OutboundKeyStore"
    trustStoreRef="OutboundKeyStore"/>

<keyStore id="OutboundKeyStore"
    location="safkeyring:///zCEE.KeyRing"
    password="password" type="JCERACFKS"
    fileBased="false" readOnly="true" />
```

SSL repertoires

Key ring for server certificate
send to for clients

Key ring for client connections to
server endpoints

safkeyring:///KeyRing v safkeyring://owner/KeyRing

Tech/Tip: Regarding *clientAuthentication* and *clientAuthenticationSupported*. Understand the implications of the interactions between these attributes. There may instances where you want to use HTTPS, but not always with mutual authentication
Consider setting *clientAuthentication* to false when setting *clientAuthenticationSupported* to true.



Tech/Tip: Combining TLS mutual and basic authentication

```
/******  
/* SET SYMBOLS  
/******  
//EXPORT EXPORT SYMLIST=(*)  
// SET CURL= '/usr/lpp/rocket/curl'  
/******  
/* CURL Procedure  
/******  
//CURL PROC  
//CURL EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
// PEND  
/******  
/* STEP CURL - use cURL to deploy API cscvinc  
/******  
//DEPLOY EXEC CURL  
BPXBATCH SH export CURL=&CURL; +  
$CURL/bin/curl -X PUT -s +  
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +  
https://wg31.washington.ibm.com:9445/zosConnect/apis/cscvinc?status=sto+  
pped > null; +  
$CURL/bin/curl -X DELETE -s +  
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +  
https://wg31.washington.ibm.com:9445/zosConnect/apis/cscvinc > null; +  
$CURL/bin/curl -X POST -s +  
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +  
--data-binary @/u/johnson/cscvinc.aar +  
--header "Content-Type: application/zip" +  
https://wg31.washington.ibm.com:9445/zosConnect/apis  
/******  
/* STEP CURL - use cURL to invoke the API cscvinc  
/******  
//INVOKE EXEC CURL  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH export CURL=&CURL; $CURL/bin/curl -X GET -s +  
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +  
https://wg31.washington.ibm.com:9445/cscvinc/employee/000100
```

```
<httpEndpoint id="defaultHttpEndpoint"  
host="*"  
httpPort="9080"  
httpsPort="9443" />  
  
<sslDefault sslRef="DefaultSSLSettings"  
outboundSSLRef="DefaultSSLSettings" />  
  
<ssl id="DefaultSSLSettings"  
keyStoreRef="CellDefaultKeyStore"  
trustStoreRef="CellDefaultKeyStore"  
clientAuthenticationSupported="true"  
clientAuthentication="true"/>  
  
<keyStore id="CellDefaultKeyStore"  
location="safkeyring:///Liberty.KeyRing"  
password="password" type="JCERACFKS"  
fileBased="false" readOnly="true" />
```

```
<httpEndpoint id="AdminHttpEndpoint"  
host="*"  
httpPort="-1"  
httpsPort="9445"  
sslOptionsRef="mySSLOptions"/>  
  
<ssLOptions id="mySSLOptions"  
sslRef="BatchSSLSettings"/>  
  
<ssl id="BatchSSLSettings"  
keyStoreRef="CellDefaultKeyStore"  
trustStoreRef="CellDefaultKeyStore"  
clientAuthenticationSupported="true"  
clientAuthentication="false"/>
```

<https://www.rocketsoftware.com/products/rocket-open-source/rocket-open-appdev-z>



The Liberty JSSE server XML configuration for outbound connections

```
<!-- Enable features -->
<featureManager>
    <feature>transportSecurity-1.0</feature>
</featureManager>

<ssl id="cicsTLSSettings"
    keyStoreRef="CICSKeyStore"
    trustStoreRef="CICSKeyStore"
    clientKeyAlias="Liberty Client Cert"/>
<keyStore id="CICSKeyStore"
    location="safkeyring:///Liberty.CICS.KeyRing"
    password="password" type="JCERACFKS"
    fileBased="false" readOnly="true" />
<ssl id="db2TLSSettings"
    keyStoreRef="Db2KeyStore"
    trustStoreRef="Db2KeyStore"
    clientKeyAlias="Liberty Client Cert"/>
<keyStore id="Db2KeyStore"
    location="safkeyring:///Liberty.Db2.KeyRing"
    password="password" type="JCERACFKS"
    fileBased="false" readOnly="true" />
<ssl id="otherTLSSettings"
    keyStoreRef="OtherKeyStore"
    trustStoreRef="OtherKeyStore">
    <outboundConnection
        host="wg31.washington.ibm.com"
        port="9555"
        clientCertificate="Client Cert"/>
</ssl>
<keyStore id="OtherKeyStore"
    location="safkeyring:///Other.KeyRing"
    password="password" type="JCERACFKS"
    fileBased="false" readOnly="true" />
```

```
<sslDefault sslRef="DefaultSSLSettings"
    outboundSSLRef="OutboundSSLSettings" />

<zosconnect_authorizationServer sslCertsRef="SSL repertoire"/>
<zosconnect_cicsIpicConnection sslCertsRef="cicsTLSSettings"/>
<zosconnect_db2Connection sslCertsRef="db2TLSSettings"> * 
<zosconnect_endpointConnect sslCertsRef= "SSL repertoire"/>
<zosconnect_zosConnectRestClient sslCertsRef="SSL repertoire"/>
<zosconnect_zosConnectServiceRestClientConnection sslCertsRef="SSL repertoire"/>
```

F BAQSTRT,REFRESH,KEYSTORE
F BAQSTRT,REFRESH,KEYSTORE, ID=CICSKeyStore
F BAQSTRT,REFRESH,KEYSTORE, ID=Db2KeyStore
F BAQSTRT,REFRESH,KEYSTORE, ID=OtherKeyStore



Tech-Tip: Enabling hardware cryptography

jvm.options

```
-Djava.security.properties=${server.config.dir}/java.security
```

java.security

```
security.provider.1=com.ibm.crypto.hdwrCCA.provider.IBMJCECCA  
security.provider.2=com.ibm.crypto.provider.IBMJCE  
security.provider.3=com.ibm.jsse2.IBMJSSEProvider2  
security.provider.4=com.ibm.security.jgss.IBMJGSSProvider  
.....
```

Enabling the IBMJCECCA provider

```
<keyStore id="CellDefaultKeyStore"  
         location="safkeyringhw://Liberty.KeyRing"  
         password="password" type="JCECCARACFKS"  
         fileBased="false" readOnly="true" />
```

Enabling the IBMJCEHYBRID provider

```
<keyStore id="CellDefaultKeyStore"  
         location="safkeyringhybrid://Liberty.KeyRing"  
         password="password" type="JCEHYBRIDRACFKS"  
         fileBased="false" readOnly="true" />
```

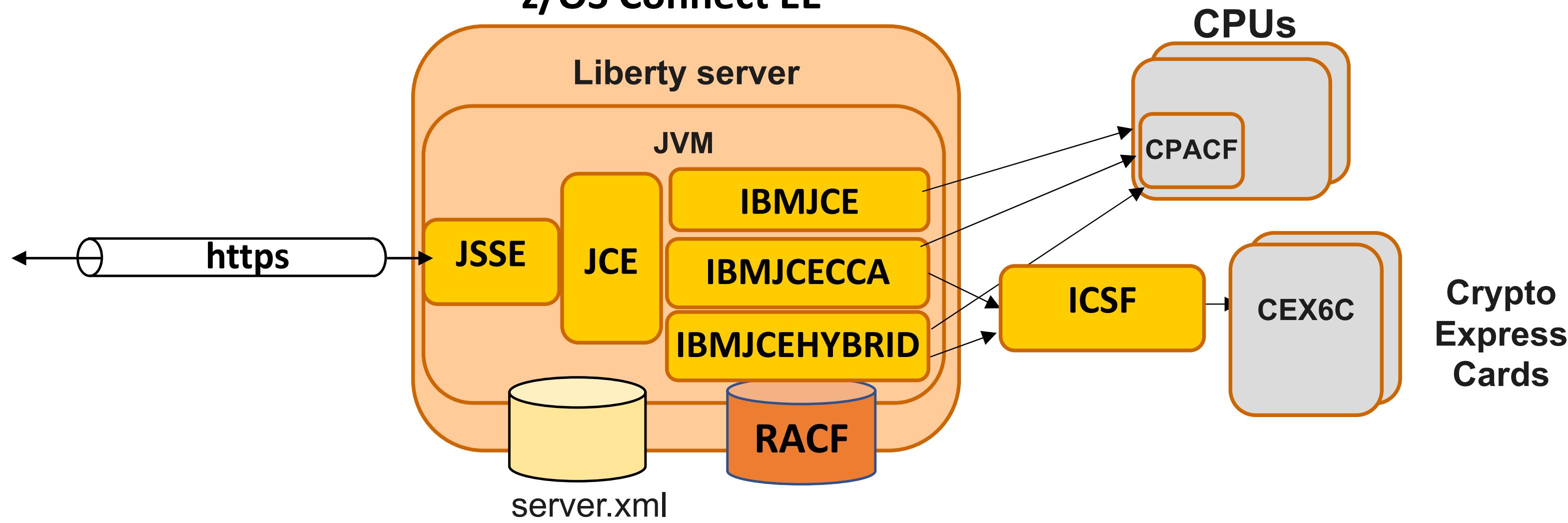
See URL <https://www.ibm.com/support/pages/node/6209109> for details on implementing IBMJCECCA and IBMJCEHYBRID hardware encryption providers



Liberty and using Java Secure Socket Extension (JSSE)

The server XML configuration defines the HTTPS ports, key rings, and other JSSE attributes

z/OS Connect EE

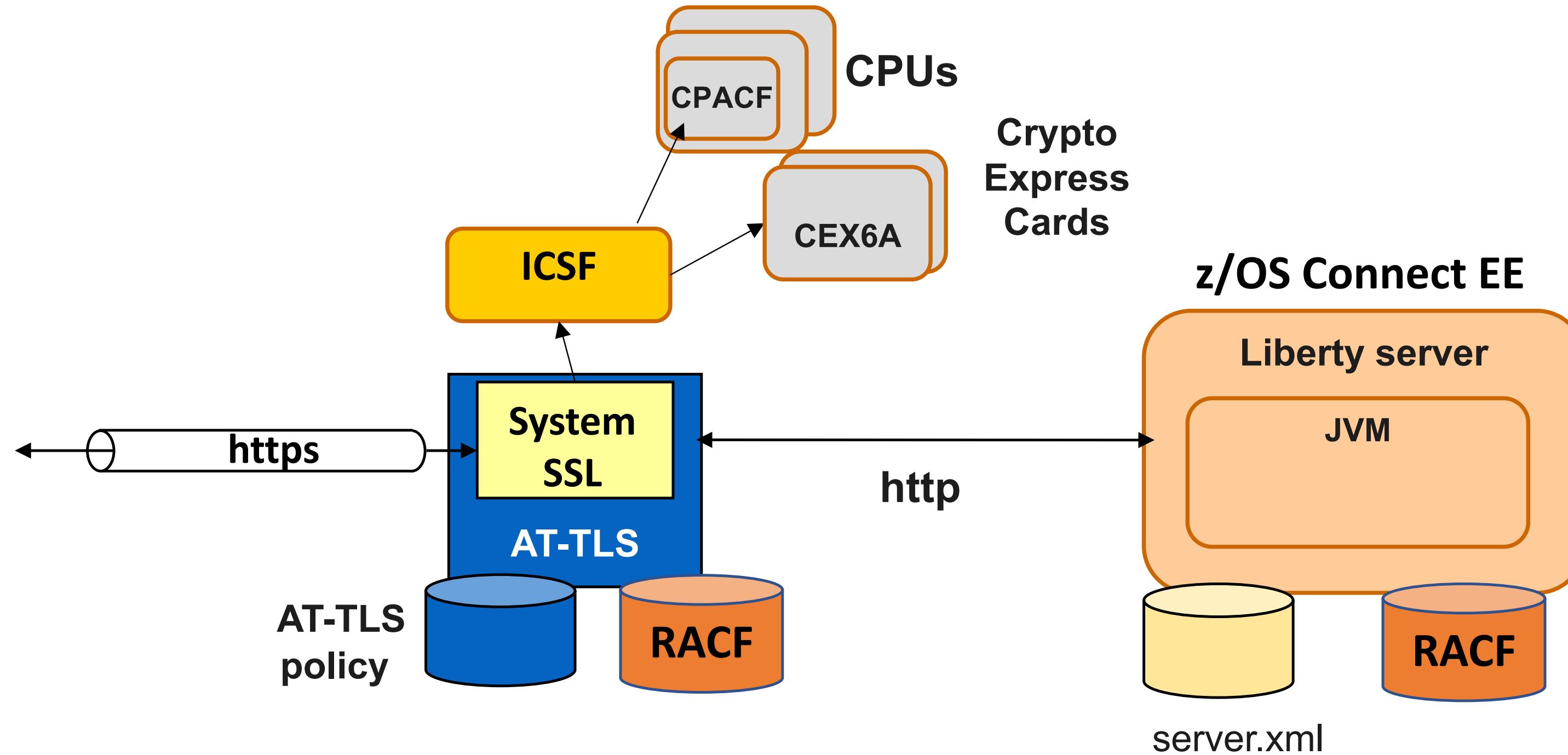


- z/OS Connect EE support for TLS is based on **Liberty** server support
- **Java Secure Socket Extension (JSSE)** API provides framework and Java implementation of TLS protocols used by Liberty HTTPS support
- **Java Cryptography Extension (JCE)** is standard extension to the Java Platform that provides implementation for cryptographic services
- **IBM Java SDK for z/OS** provides three different JCE providers, **IBMJCE**, **IBMJCECCA** and **IBMJCEHYBRID**.
- The JCE providers access **CPACF (CP Assist for Cryptographic Functions)** directly, therefore keep your Java service levels current.



Liberty and using AT-TLS

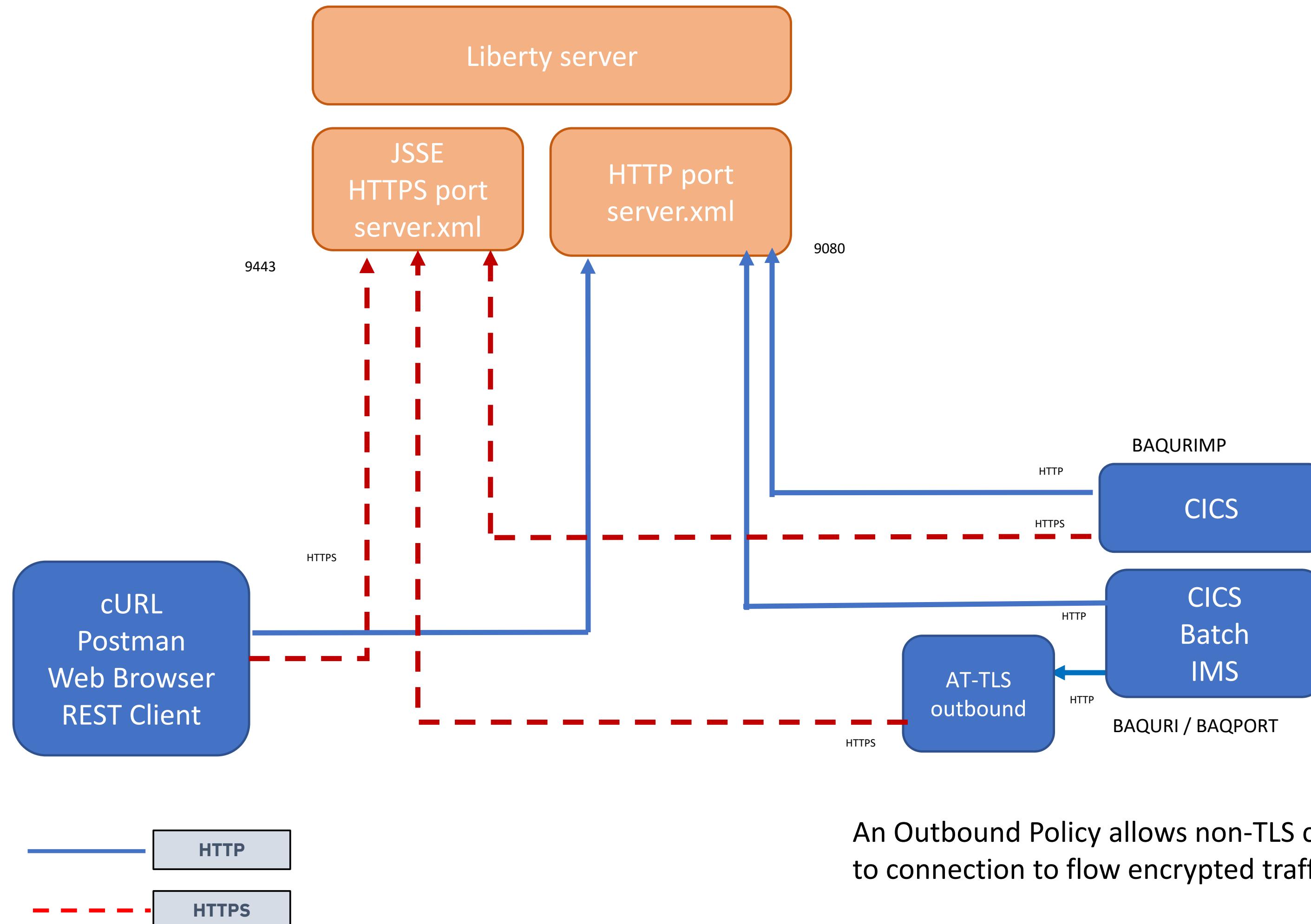
The server XML configuration uses no HTTPS protocol, key rings or other JSSE attributes



- **Application Transparent TLS (AT-TLS)** creates a secure session on behalf of z/OS Connect
- Only define http ports in server.xml (z/OS Connect does not know that TLS session exists)
- Define TLS protection for all applications (including z/OS Connect) in **AT-TLS policy**
- AT-TLS uses **System SSL** which exploits the CPACF and Crypto Express cards via ICSF

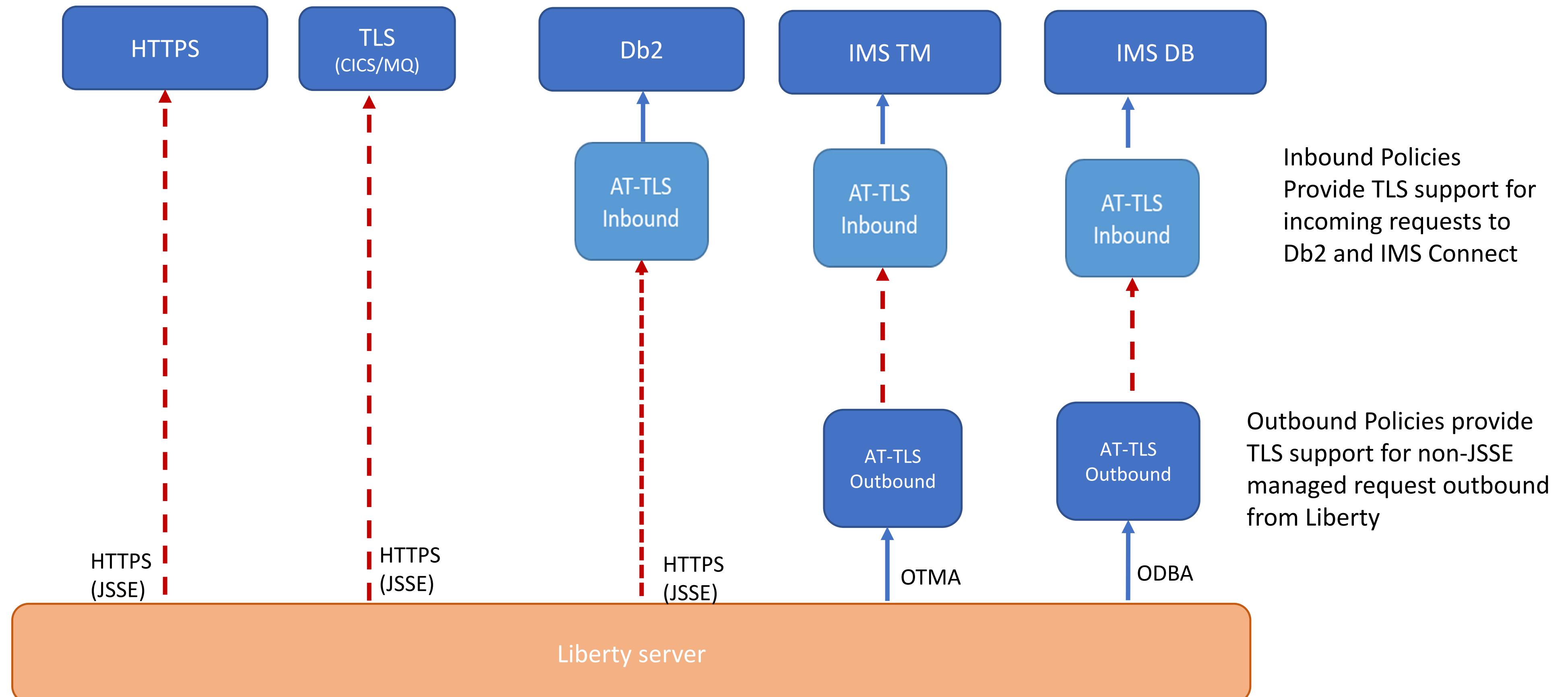


Tech-Tip: TLS client encryption to a Liberty server scenarios





Tech-Tip: TLS encryptions from a Liberty server (HTTPS/native to HTTPS/TLS/OTMA/ODBA)





z/OS Connect Security server XML TLS Security Configuration (OpenAPI 2)

- requireSecure - requires the client to connect using HTTPS

```
<zosconnect_zosConnectManager  
    requireAuth="true"  
    requireSecure="true|false"/>  
  
<zosconnect_zosConnectAPIs>  
    <zosConnectAPI name="catalog"  
        requireAuth="true"  
        requireSecure="true|false"/>">/>  
</zosconnect_zosConnectAPIs>  
  
<zosconnect_services>  
    <service id="selectByEmployee"  
        name="selectEmployee"  
        requireAuth="true"  
        requireSecure="true|false"/>  
</zosconnect_services>  
  
<zosconnect_apiRequesters>  
    requireAuth="true|false"  
    <apiRequester name="cscvincapi_1.0.0"  
        requireAuth="true"  
        requireSecure="true|false"/>  
</zosconnect_apiRequesters>
```

Globally, requires that client connections use HTTPS, unless overridden on the specific resource definitions.

Requires that client connections use HTTPS (true) or HTTP(false) in order to access the API.

Requires that client connections use HTTPS (true) or HTTP(false) to directly access the service. This attribute is ignored when the service is invoked from an API, then only the API requireSecure attribute is relevant.

Requires that client connections use HTTPS (true) or HTTP(false) to access the PI requesters. If the requireSecure attribute is not set, the global setting on the zosconnect_zosConnectManager element is used instead, unless the requireSecure attribute is overridden on the specific API requester.

The requireSecure attribute controls whether an inbound must be using HTTPS(true) or if HTTP(false) is allowed.

Note that there are no equivalent configuration elements for an z/OS Connect OpenAPI 3 server.

**Let's explore using TLS for
encryption and data integrity
using samples in various scenarios**

Using this Liberty JSSE server XML configuration



```
<!-- Enable features -->
<featureManager>
    <feature>transportSecurity-1.0</feature>
</featureManager>

<sslDefault sslRef="DefaultSSLSettings"
    outboundSSLRef="OutboundSSLSettings" />

<ssl id="DefaultSSLSettings"
    keyStoreRef="CellDefaultKeyStore"
    trustStoreRef="CellDefaultKeyStore"
    clientAuthenticationSupported="true"
    clientAuthentication="true"
    serverKeyAlias="Liberty Server Cert"/>

<keyStore id="CellDefaultKeyStore"
    location="safkeyring:///Liberty.KeyRing"
    password="password" type="JCERACFKS"
    fileBased="false" readOnly="true" />

<ssl id="OutboundSSLSettings"
    keyStoreRef="OutboundKeyStore"
    trustStoreRef="OutboundKeyStore"
    clientKeyAlias="Liberty Client Cert"/>

<keyStore id="OutboundKeyStore"
    location="safkeyring:///zCEE.KeyRing"
    password="password" type="JCERACFKS"
    fileBased="false" readOnly="true" />

<zosconnect_authorizationServer sslCertsRef="SSL repertoire"/>
<zosconnect_cicsIpicConnection sslCertsRef="SSL repertoire"/>
<zosconnect_endpointConnect sslCertsRef="SSL repertoire"/>
<zosconnect_zosConnectRestClient sslCertsRef="SSL repertoire"/>
<zosconnect_zosConnectServiceRestClientConnection sslCertsRef="SSL repertoire"/>
```

SSL repertoires

Using the contents of these key rings

Liberty's outbound key ring

Digital ring information for user LIBSERV:				
Ring:	Certificate Label Name	Cert Owner	USAGE	DEFAULT
>zCEE.KeyRing<				
zCEE CA		CERTAUTH	CERTAUTH	NO
Liberty CA		CERTAUTH	CERTAUTH	NO
zCEE Client Cert		ID (LIBSERV)	PERSONAL	YES
xyz Client Cert		ID (LIBSERV)	PERSONAL	No
DB2 CA		CERTAUTH	CERTAUTH	NO
MQ CA		CERTAUTH	CERTAUTH	NO
CICS CA		CERTAUTH	CERTAUTH	NO
Ring:	Certificate Label Name	Cert Owner	USAGE	DEFAULT
>Liberty.KeyRing<				
Liberty Server Cert		ID (LIBSERV)	PERSONAL	YES
Liberty CA		CERTAUTH	CERTAUTH	NO
zCEE CA		CERTAUTH	CERTAUTH	NO
CICS CA		CERTAUTH	CERTAUTH	NO
Digital ring information for user CICSSTC:				
Ring:	Certificate Label Name	Cert Owner	USAGE	DEFAULT
>CICS.KeyRing<				
CICS CA		CERTAUTH	CERTAUTH	NO
CICS Client Cert		ID (CICSSTC)	PERSONAL	YES
Liberty CA		CERTAUTH	CERTAUTH	NO
zCEE CA		CERTAUTH	CERTAUTH	NO
Digital ring information for user DB2USER:				
Ring:	Certificate Label Name	Cert Owner	USAGE	DEFAULT
>Db2.KeyRing<				
DB2 CA		CERTAUTH	CERTAUTH	NO
zCEE CA		CERTAUTH	CERTAUTH	NO
DB2USER		ID (DB2USER)	PERSONAL	YES

Liberty's outbound key ring to CICS

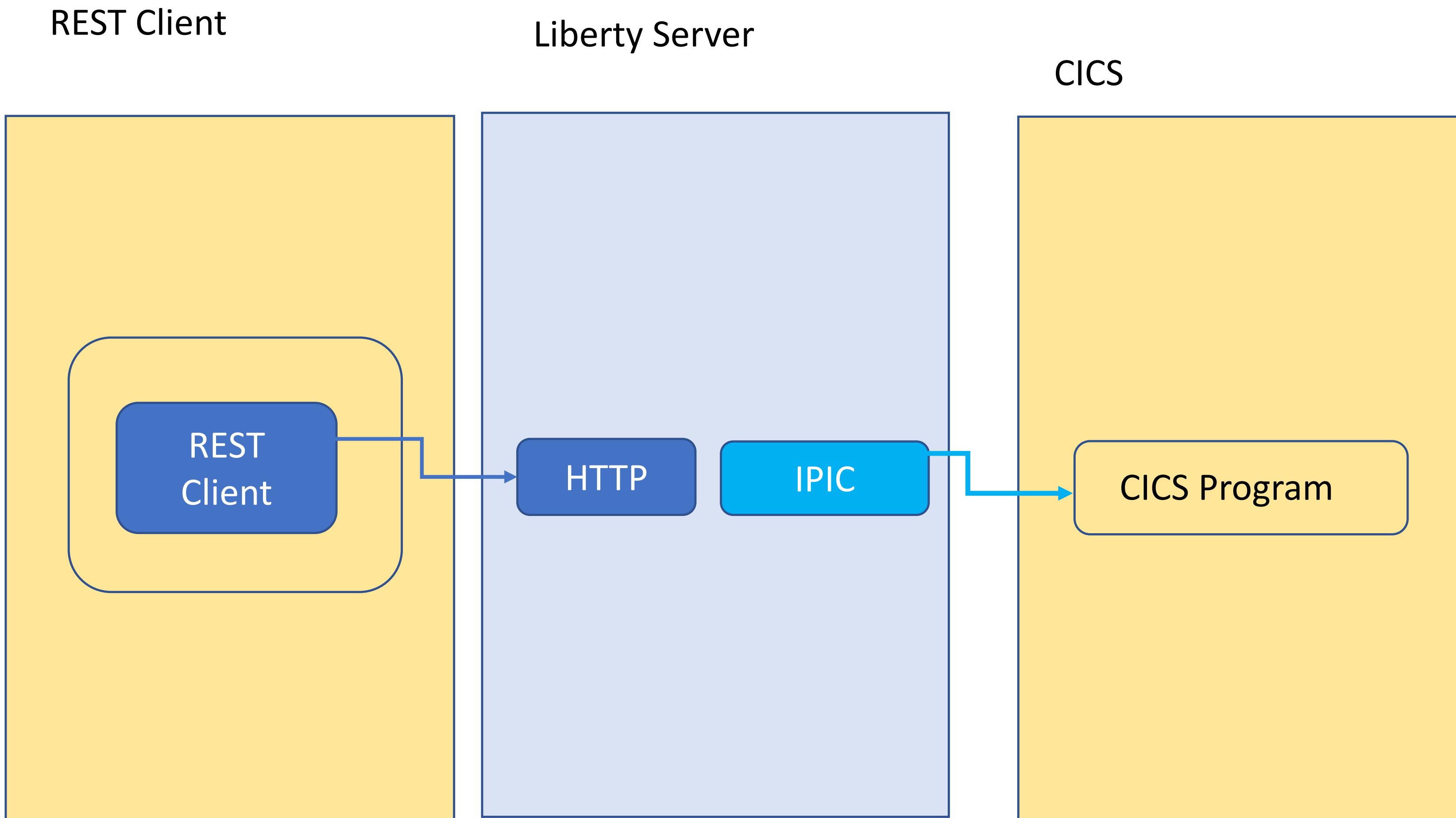
Liberty's outbound key ring to Db2

Tech-Tip: when Liberty is the client endpoint, and more than one personal certificate is connected to a key ring. Use the SSL repertoire *clientKeyAlias* attributes to select the personal certificate to be used in a handshake.

Tech-Tip: when Liberty is the server endpoint, and more than one personal certificate is connected to a key ring. Use the SSL repertoire *serverKeyAlias* attributes to select the personal certificate to be used in a handshake.

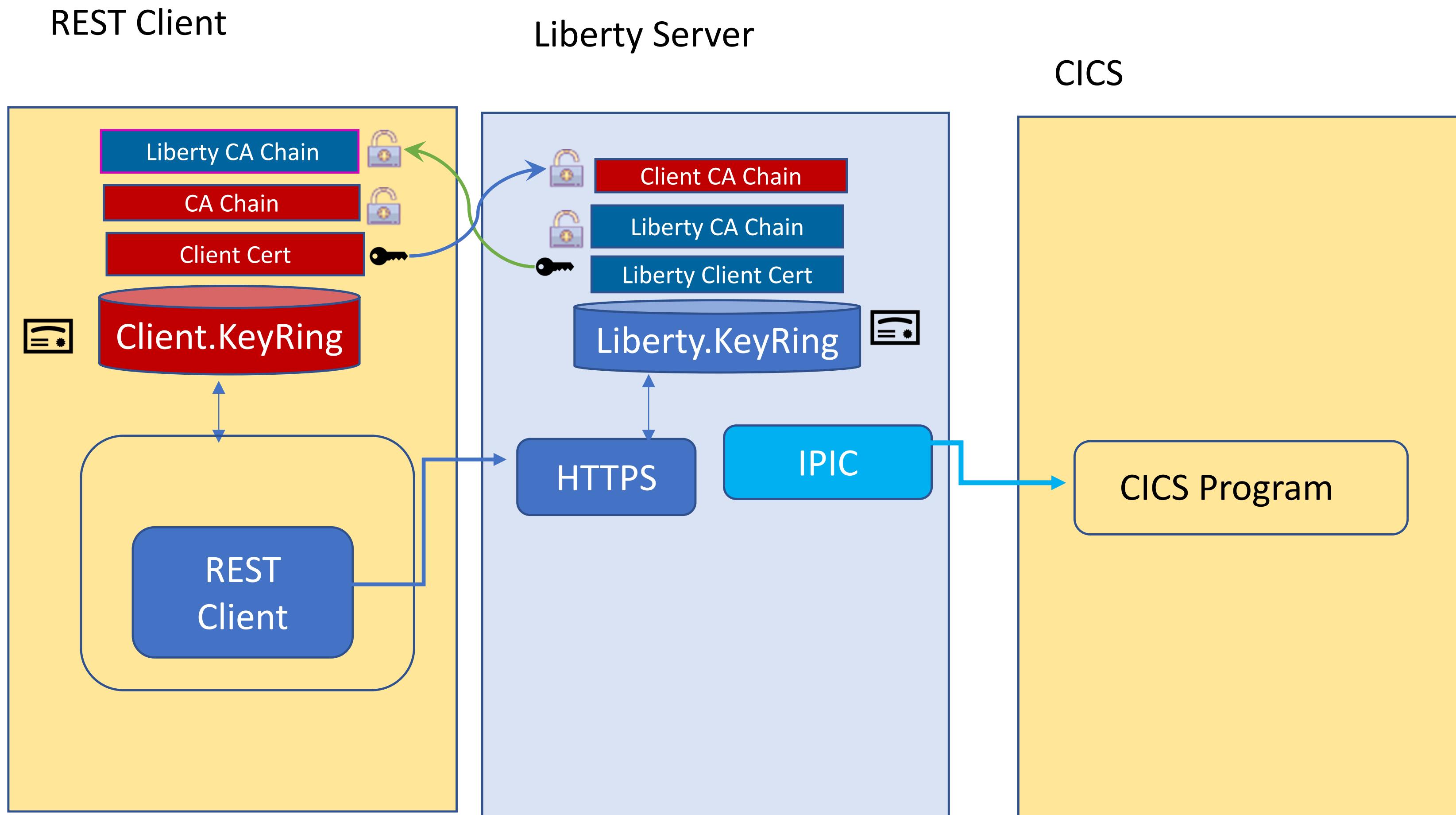


No TLS between any endpoints



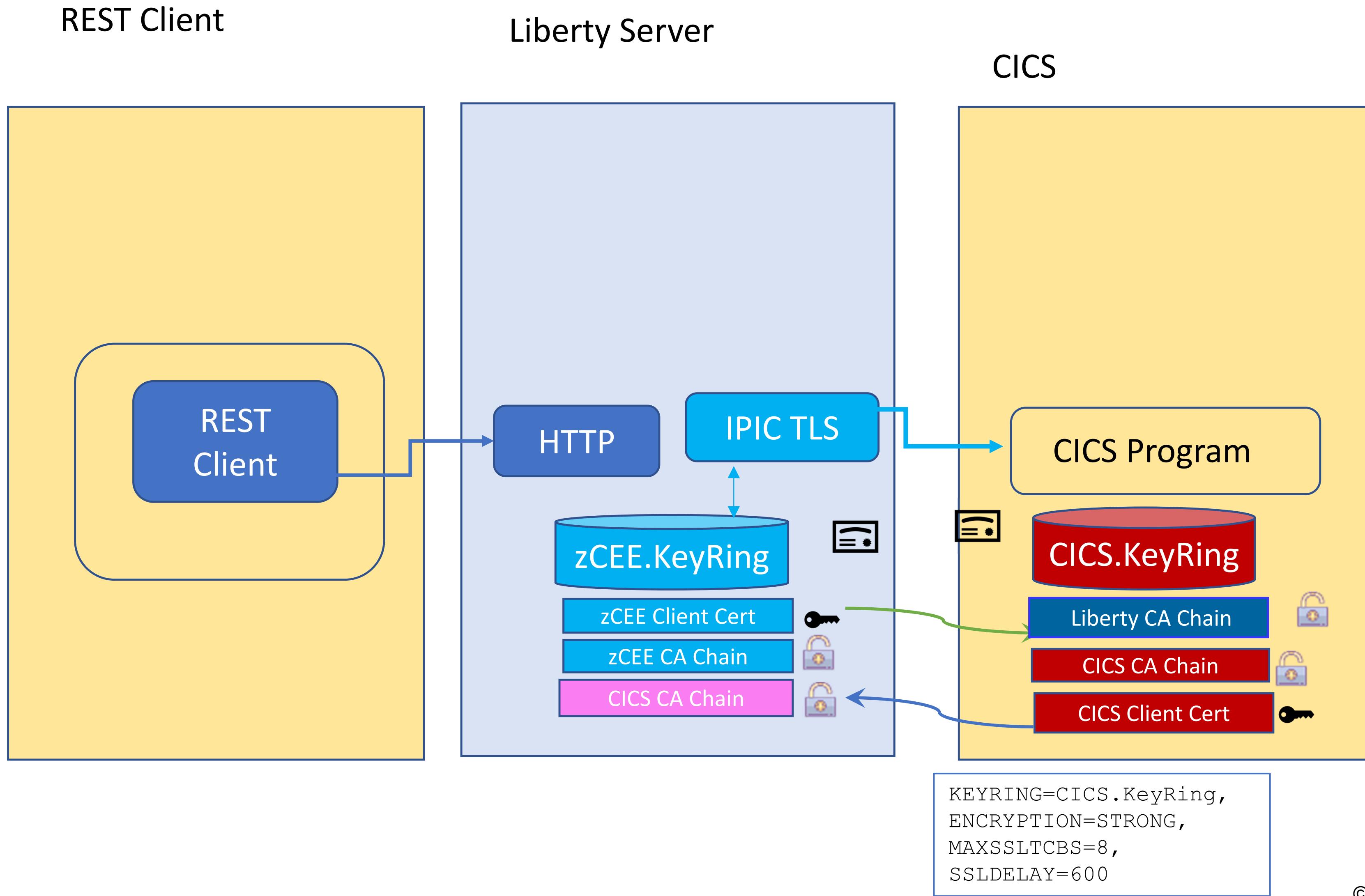


TLS handshake between the client and Liberty server



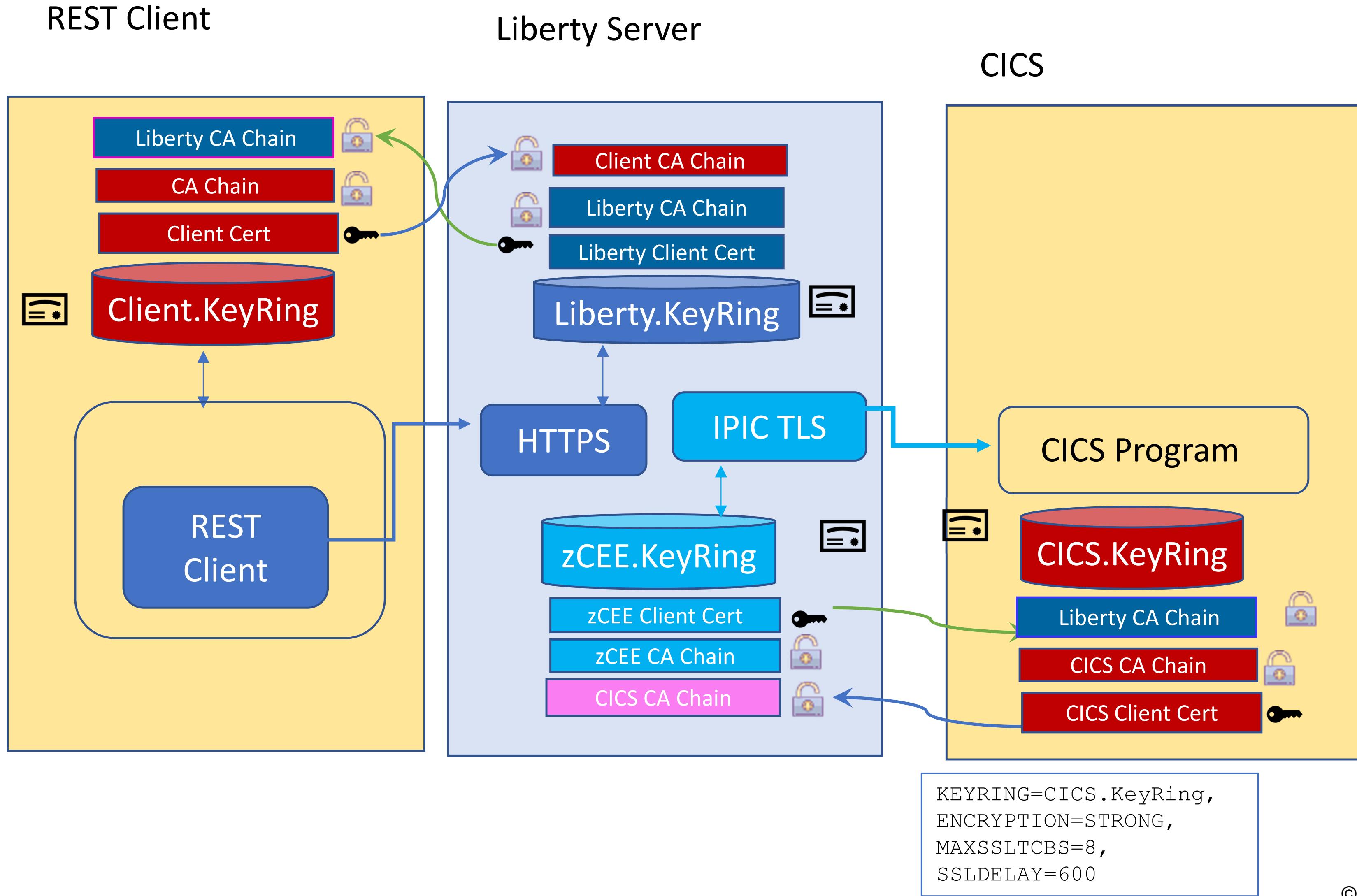


TLS handshake between the Liberty server and the target endpoint





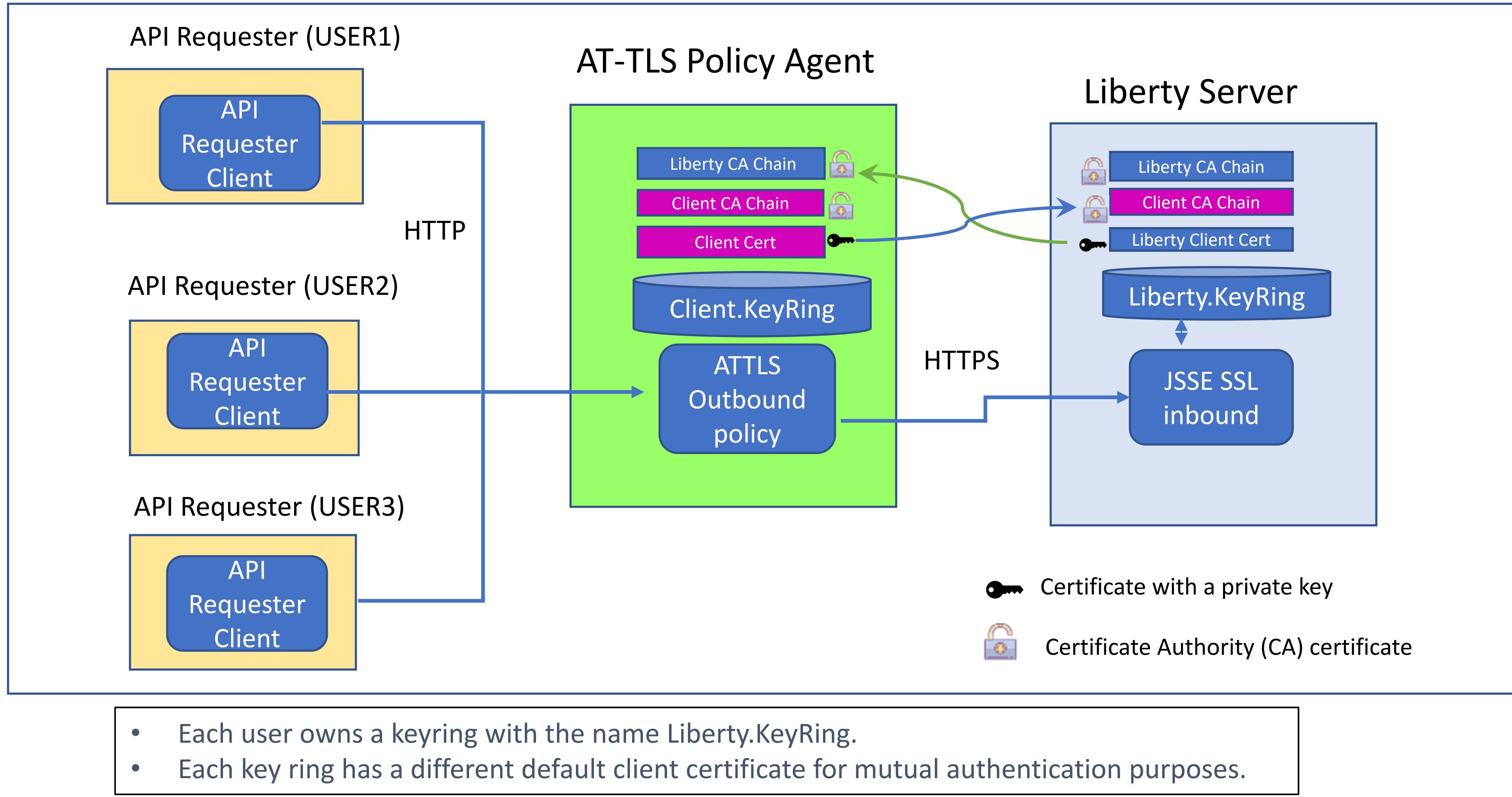
TLS handshakes between all endpoints





AT-TLS - outbound policy handshake scenario

Use of a common key ring name for multiple client identities

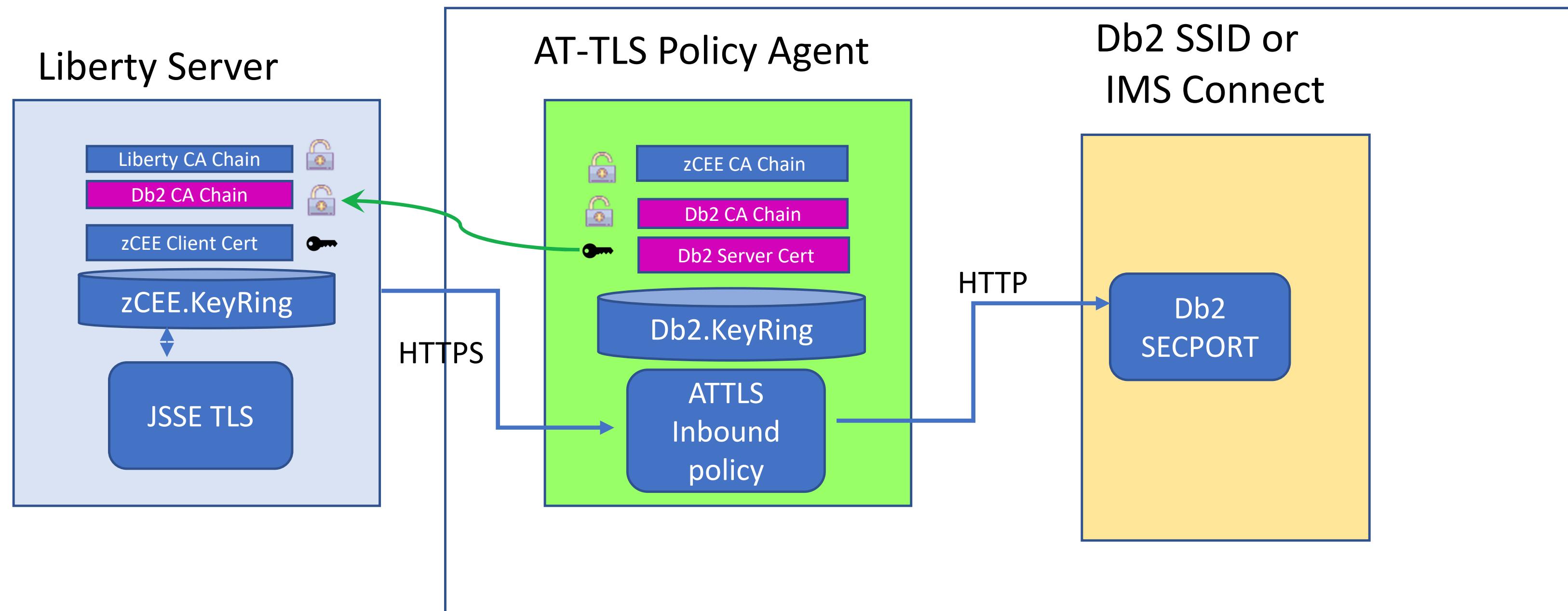


This is a situation when AT-TLS mutual authentication has a benefit.



AT-TLS - inbound policy handshake scenario (Db2 and IMS)

Policy Agent uses an inbound policy and acts a surrogate TLS server



Note that DB2 is AT-TLS aware
IMS is AT-TLS unaware

Certificate with a private key

Certificate Authority (CA) certificate



Ciphers

- During the TLS handshake, the TLS protocol and data exchange cipher are negotiated
- Choice of cipher and key length has an impact on performance
- You can restrict the protocol (TLS) and ciphers to be used
- Example setting server.xml file

```
<ssl id="DefaultSSLSettings" keyStoreRef="defaultKeyStore"  
sslProtocol="TLSv1.2"  
enabledCiphers="TLS_RSA_WITH_AES_256_CBC_SHA256  
TLS_RSA_WITH_AES_256_GCM_SHA384"/>
```

- This configures use of TLS 1.2 and two supported ciphers
- It is recommended to control what ciphers can be used in the server rather than the client

For cipher details, see IBM SDK Java 8.0.0 Cipher Suites at URL

https://www.ibm.com/support/knowledgecenter/SSYKE2_8.0.0/com.ibm.java.security.component.80.doc/security-component/jsse2Docs/ciphersuites.html

Tech/Tip: A note on cipher suite names

A CipherSuite is a suite of cryptographic algorithms used by a TLS connection. A suite comprises three distinct algorithms:

- The key exchange and authentication algorithm, used during the handshake
- The encryption algorithm, used to encipher the data
- The MAC (Message Authentication Code) algorithm, used to generate the message digest

There are several options for each component of the suite, but only certain combinations are valid when specified for a TLS connection. The name of a valid CipherSuite defines the combination of algorithms used. For example, the CipherSuite ***TLS_RSA_WITH_AES_128_CBC_SHA*** specifies:

- The RSA key exchange and authentication algorithm
- The AES encryption algorithm, using a 128-bit key and cipher block chaining (CBC) mode
- The SHA-1 Message Authentication Code (MAC)

Note				
To use some CipherSuites, the 'unrestricted' policy files need to be configured in the JRE. For more details of how policy files are set up in an SDK or JRE, see the <i>IBM SDK Policy files</i> topic in the <i>Security Reference for IBM SDK, Java Technology Edition</i> for the version you are using.				
Table 1. CipherSpecs supported by IBM MQ and their equivalent CipherSuites				
CipherSpec	Equivalent CipherSuite (IBM JRE)	Equivalent CipherSuite (Oracle JRE)	Protocol	FIPS 140-2 compatible
ECDHE_ECDSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	TLS 1.2	yes
ECDHE_ECDSA_AES_128_CBC_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	TLS 1.2	yes
ECDHE_ECDSA_AES_128_GCM_SHA256	SSL_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	TLS 1.2	yes
ECDHE_ECDSA_AES_256_CBC_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	TLS 1.2	yes
ECDHE_ECDSA_AES_256_GCM_SHA384	SSL_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	TLS 1.2	yes
ECDHE_ECDSA_NULL_SHA256	SSL_ECDHE_ECDSA_WITH_NULL_SHA	TLS_ECDHE_ECDSA_WITH_NULL_SHA	TLS 1.2	no
ECDHE_ECDSA_RC4_128_SHA256	SSL_ECDHE_ECDSA_WITH_RC4_128_SHA	TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	TLS 1.2	no
ECDHE_RSA_3DES_EDE_CBC_SHA256	SSL_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	TLS 1.2	yes
ECDHE_RSA_AES_128_CBC_SHA256	SSL_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	TLS 1.2	yes
ECDHE_RSA_AES_128_GCM_SHA256	SSL_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	TLS 1.2	yes
ECDHE_RSA_AES_256_CBC_SHA384	SSL_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	TLS 1.2	yes
ECDHE_RSA_AES_256_GCM_SHA384	SSL_ECDHE_RSA_WITH_AES_256_GCM_SHA384	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	TLS 1.2	yes
ECDHE_RSA_NULL_SHA256	SSL_ECDHE_RSA_WITH_NULL_SHA	TLS_ECDHE_RSA_WITH_NULL_SHA	TLS 1.2	no
ECDHE_RSA_RC4_128_SHA256	SSL_ECDHE_RSA_WITH_RC4_128_SHA	TLS_ECDHE_RSA_WITH_RC4_128_SHA	TLS 1.2	no

<https://www.ibm.com/docs/en/ibm-mq/9.3?topic=java-tls-cipherspecs-ciphersuites-in-mq-classes>

mitchj@us.ibm.com

© 2017, 2024 IBM Corporation
Slide 93



CICS IPIC using TLS

The server.xml file is the key configuration file:

The diagram illustrates the configuration of CICS IPIC using TLS across three different interfaces:

- Liberty Admin Center:** Shows the "Server Config" interface for the file `ipicSSLIDProp.xml`. The XML code defines a `<server>` block with a `description` of "CICS IPIC ID propagation connections". It includes a `<featureManager>` section with a `<feature>` for `zosconnect:cicsService-1.0`. The `<zosconnect_cicsIplicConnection id="catalog">` section specifies a host of "wg31.washington.ibm.com", port "1493", and a transid of "MIJO". The `sslCertsRef` attribute is set to "cicsSSLSettings".

```
<server description="CICS IPIC ID propagation connections">
  <!-- Enable features -->
  <featureManager>
    <feature>zosconnect:cicsService-1.0</feature>
  </featureManager>
  <zosconnect_cicsIplicConnection id="catalog">
    host="wg31.washington.ibm.com"
    port="1493"
    zosConnectNetworkId="CSCVINC"
    zosConnectApplid="CSCVINC"
    transid="MIJO"
    transidUsage="EIB_AND_MIRROR"
    sslCertsRef="cicsSSLSettings"/>
</server>
```
- Configuration interface:** Shows the "inquireSingle Service" configuration window. Under "Required Configuration", it shows "Coded character set identifier (CCSID)" as 37 and "Connection reference" as "catalog". Under "Optional Configuration", it shows "Transaction ID" and "Transaction ID usage".
- WG31 Command Line Interface:** Shows the output of the `TCPIPPS` command. It lists various TCP/IP parameters, including `Ssltype(Ssl)`, `Transid(CISSL)`, `Authenticate(Noauthenticate)`, `CipherSuite(CipherSuite)`, and `Host(ANY)`. Two specific entries are circled in red: `Ssltype(Ssl)` and `Host(ANY)`.

A callout box at the bottom right points to the "Define IPIC/TLS connections to CICS" section.

Tech/Tip: Cipher Suite numbers (CICS TCPIPSERVICE):

Table 2. 2-character and 4-character cipher suite definitions for SSL V3, TLS V1.0, TLS V1.1, TLS V1.2, and TLS V1.3

2-character cipher number	4-character cipher number	Short name	Description ¹	FIPS 140-2	Base security level	Security level 3
00	0000	TLS_NULL_WITH_NULL_NULL	No encryption or message authentication and RSA key exchange		X	X
01	0001	TLS_RSA_WITH_NULL_MD5	No encryption with MD5 message authentication and RSA key exchange		X	X
02	0002	TLS_RSA_WITH_NULL_SHA	No encryption with SHA-1 message authentication and RSA key exchange		X	X
03	0003	TLS_RSA_EXPORT_WITH_RC4_40_MD5	40-bit RC4 encryption with MD5 message authentication and RSA (export) key exchange		X	X
04	0004	TLS_RSA_WITH_RC4_128_MD5	128-bit RC4 encryption with MD5 message authentication and RSA key exchange			X
05	0005	TLS_RSA_WITH_RC4_128_SHA	128-bit RC4 encryption with SHA-1 message authentication and RSA key exchange			X

<https://www.ibm.com/docs/en/zos/3.1.0?topic=programming-cipher-suite-definitions>



MQ JMS using TLS

The screenshot shows the Service Project Editor's Configuration tab for a "twoWay Service". It lists various configuration parameters:

- Connection factory JNDI name: jms/qmgrCf
- Request destination JNDI name: jms/requestQueue
- Reply destination JNDI name: jms/replyQueue
- Wait interval: 3000
- MQMD format: MQSTR
- Coded character set identifier (CCSID): 37
- Is message persistent:
- Reply selection: msgIDToCorrelID
- Expiry: -1

Below this is a "LIBERTY.SSL.SVRCONN - Properties" dialog, specifically the "SSL" tab. It shows the selected cipher spec as "TLS_RSA_WITH_AES_256_CBC_SHA256".

The server.xml file is the key configuration file:

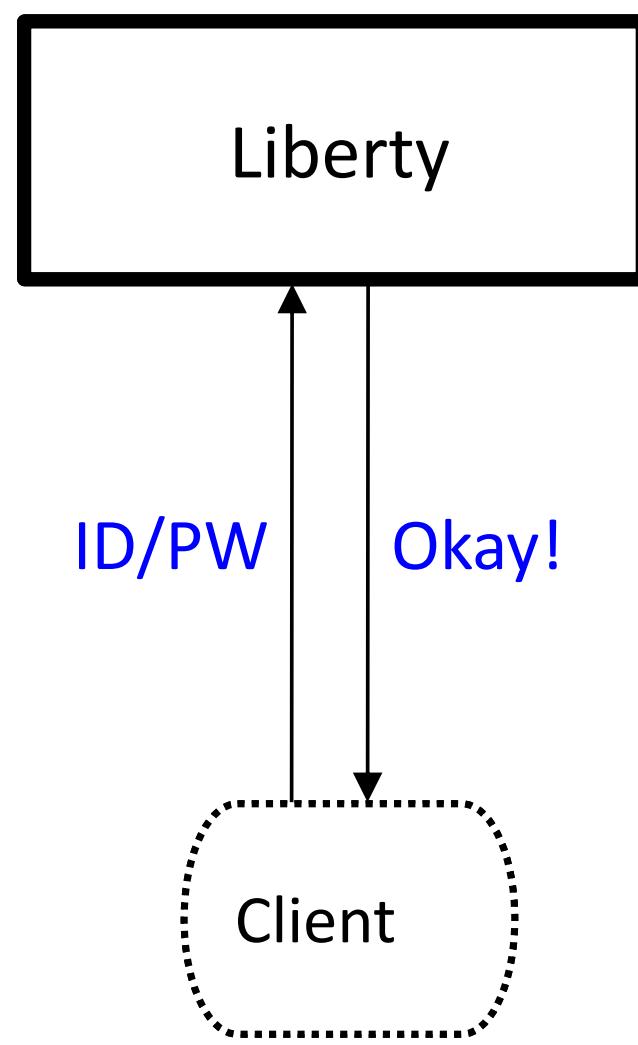
```
<server description="MQ Service Provider">
<featureManager>
<feature>zosconnect:mqService-1.0</feature>
</featureManager>
<variable name="wmqJmsClient.rar.location" value="/u/johnson/jca/wmq.jmsra.rar"/>
<wmqJmsClient nativeLibraryPath="/usr/lpp/mqm/V9R1M1/java/lib"/>
<zosconnect_services>
<service name="mqPutService">
<property name="useCallerPrincipal" value="true"/>
</service>
</zosconnect_services>
<connectionManager id="ConMgr1" maxPoolSize="5"/>
<jmsConnectionFactory id="qmgrCf" jndiName="jms/qmgrCf">
<connectionManagerRef="ConMgr1">
<properties.wmqJMS transportType="CLIENT" queueManager="ZMQ1" channel="LIBERTY.SSL.SVRCONN" hostName="wg31.washington.ibm.com" sslcipherSuite="SSL_RSA_WITH_AES_256_CBC_SHA256" port="1422" />
</jmsConnectionFactory>
<jmsQueue id="q1" jndiName="jms/default">
<properties.wmqJms baseQueueName="ZCEE.DEFAULT.MQZCEE.QUEUE" CCSID="37"/>
</jmsQueue>
</server>
```



Authentication - Third Party Authentication

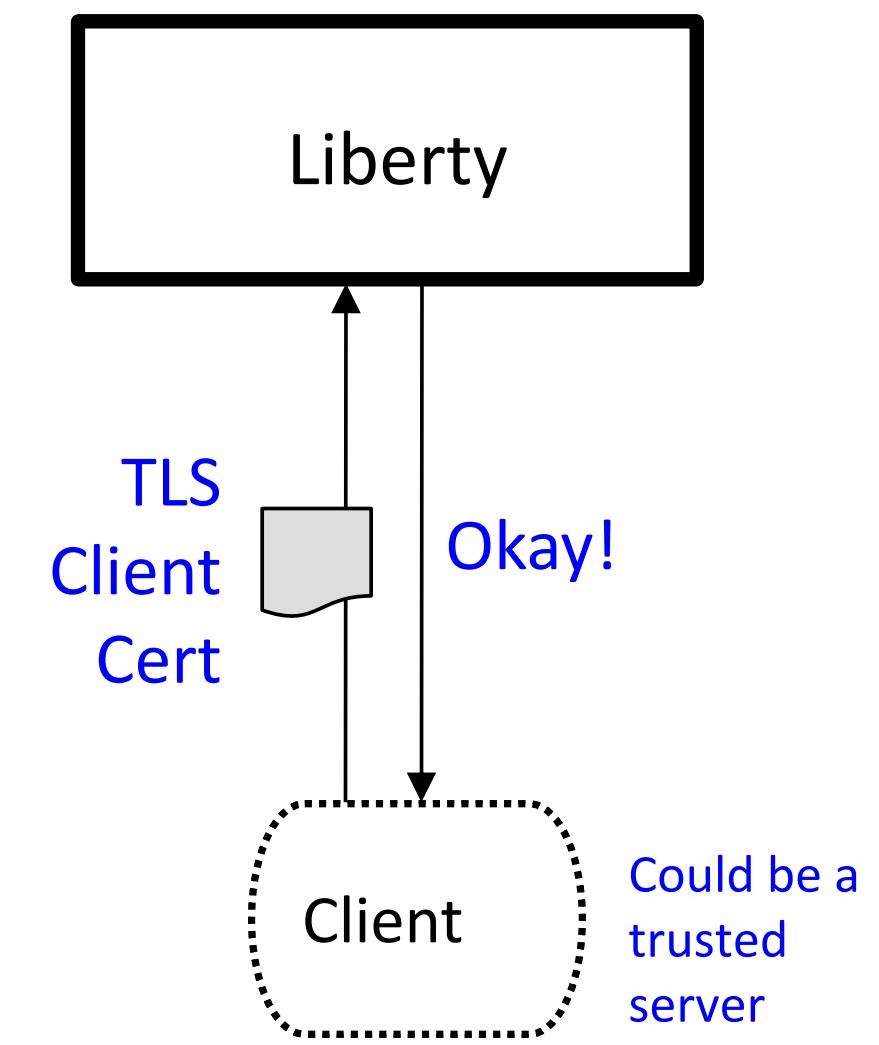
Several different ways this can be accomplished:

Basic Authentication



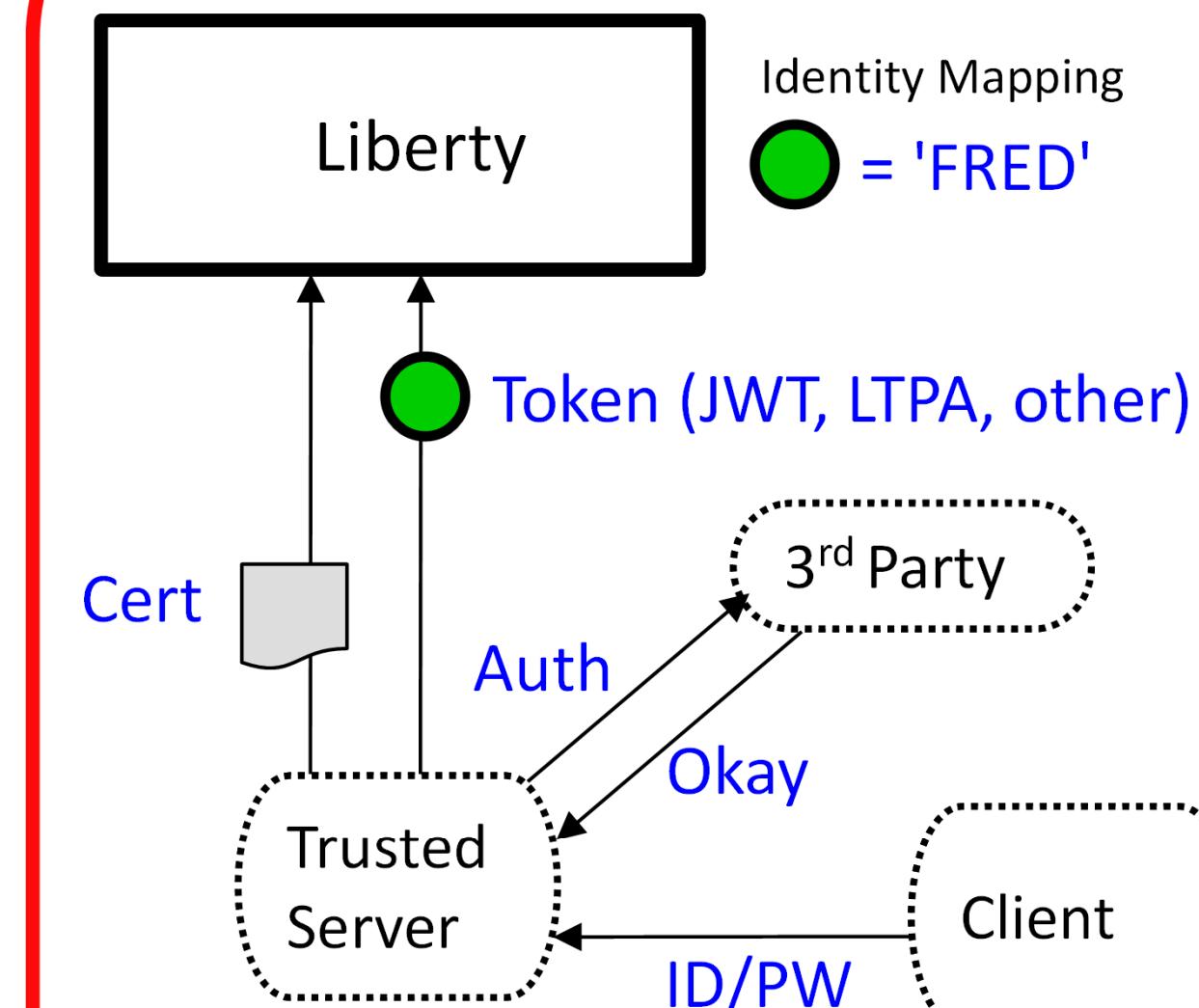
Server prompts for ID/PW
Client supplies ID/PW or ID/PassTicket
Server checks registry:
• Basic (server.xml)
• SAF

Client Certificate



Server prompts for client certificate.
Client supplies certificate
Server validates client certificate and maps to an identity
Registry options:
• SAF

Third Party Authentication



**Client authenticates to 3rd party sever
Client receives a trusted 3rd party token
Token flows to Liberty z/OS and is mapped to an identity
Registry options:
• We may know these detail.**



Third Party Authentication Examples

The screenshot shows the UPS Sign Up page. At the top, there's a yellow banner with the text "UPS is open for business: Service impacts related to Coronavirus ...More". Below the banner, the UPS logo is displayed. A "Sign Up" button is prominent. Below it, a link "Already have an ID? Log in" is shown. The page instructs users to "Use one of these sites." followed by icons for Google, Facebook, Amazon, and Apple. It also says "Or enter your own information." and lists fields for Name*, Email*, User ID*, Password*, and Phone. A "Show" link is next to the Password field.

The screenshot shows the "Log In to myNCDMV" page. It features a "Log In" tab and a "Sign Up" tab. The "Log In" section contains fields for "Email Address" (with "name@example.com" entered) and "Password" (with "*****" entered). There's a "Remember Me" checkbox and "Log In" and "Forgot Password" buttons. Below these, there's a "Continue as Guest" link. The "Sign Up" section is partially visible on the right. The background of the page is a photograph of autumn foliage.



Open security standards

- OAuth is an open standard for access delegation, used as a way to grant websites or applications access to their information without requiring a password.
- **From the OpenID Core specification:** OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It enables Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.
- **OAuth 2.0 Core (RFC 6749) Specifications:** <https://tools.ietf.org/html/rfc6749>
- **OpenID Connect Core Specifications:** https://openid.net/specs/openid-connect-core-1_0.html

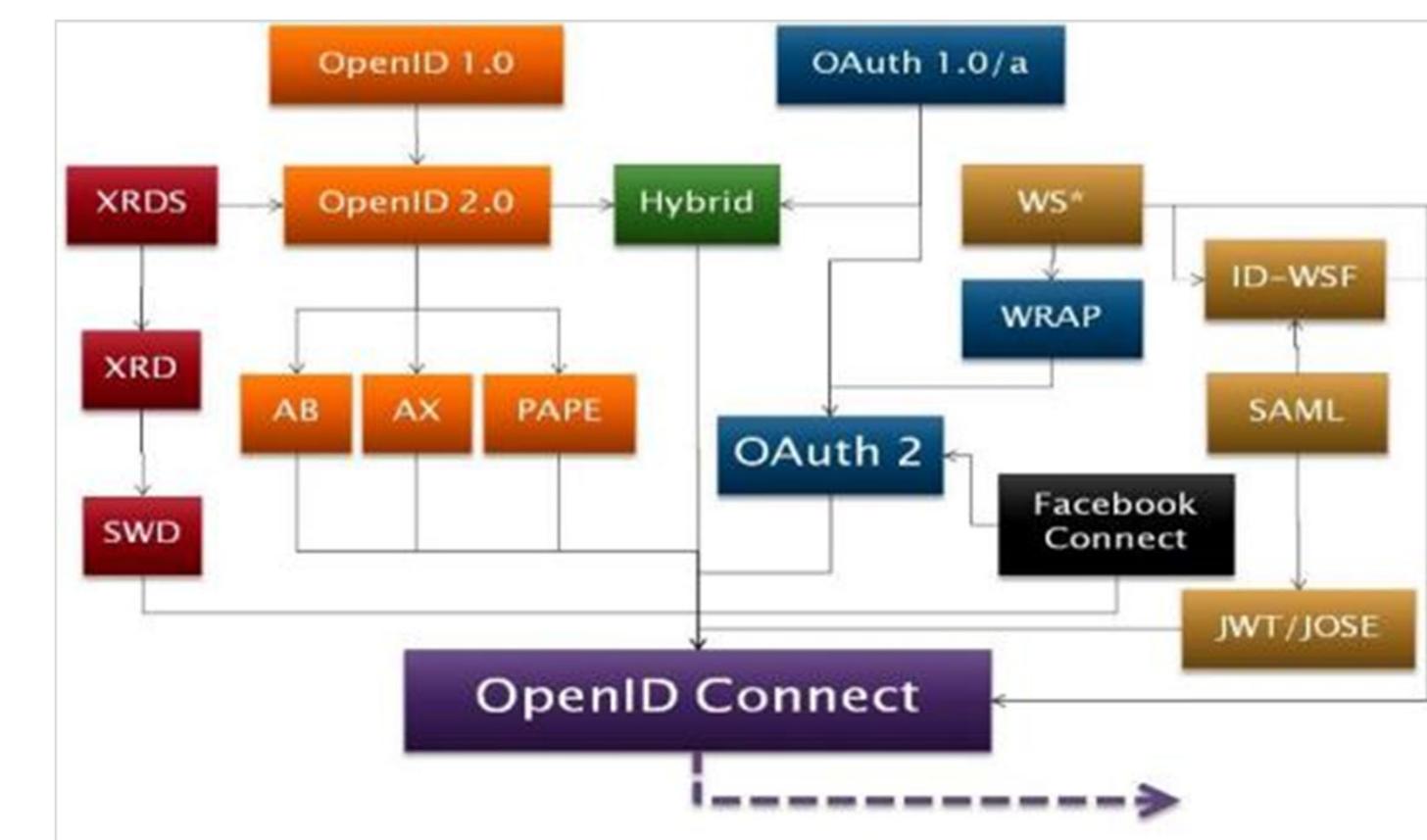
See the YouTube videos:

OAuth 2.0 and OpenID Connect (in plain English)

<https://www.youtube.com/watch?v=996OjexHze0>

OpenID Connect on Liberty

<https://www.youtube.com/watch?v=fuajCS5bG4c>





OpenID Connect/OAuth

- **From the z/OS Connect Knowledge Center:** z/OS Connect EE security can operate with traditional z/OS security, for example, System Authorization Facility (SAF) and also with open standards such as Transport Layer Security (TLS), JSON Web Token (JWT), and **OpenID Connect**.
- **From the OpenID Core specification:** OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It enables Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.
- **OAuth 2.0 Core (RFC 6749) Specifications:** <https://tools.ietf.org/html/rfc6749>
- **OpenID Connect Core Specifications:** https://openid.net/specs/openid-connect-core-1_0.html
- **Again, for a very good explanation of this topic see YouTube video OAuth 2.0 and OpenID Connect (in plain English)**
<https://www.youtube.com/watch?v=996OjexHze0>

What is a JWT (JSON Web Token) ?

- JWT is a compact way of representing claims that are to be transferred between two parties
- Normally transmitted via HTTP header
- Consists of three parts
 - Header
 - Payload
 - Signature

The screenshot shows the jwt.io debugger interface. At the top, it says "Encoded" and displays a long string of characters: eyJraWQiOiiI0cWpYLWJrWE9Vd19GX...vT_Ez0fD-. At the bottom of this string, there is a timestamp: "Mon Nov 02 2020 11:05:58 GMT-0500 (Eastern Standard Time)". A red oval highlights this timestamp. To the right, under "Decoded", the token is shown as JSON. The "HEADER" section contains: { "kid": "4qjX-bkX0Uw_F_uccjRMkB9ivMjXSQwj0RrkyRJq8DM", "alg": "RS256" }. The "PAYLOAD" section contains: { "sub": "Fred", "token_type": "Bearer", "scope": ["openid", "profile", "email"], "azp": "rpSsl", "iss": "https://wg31.washington.ibm.com:26213/oidc/endpoint/OP", "aud": "myZcee", "exp": 160433158, "iat": 160433158, "realmName": "zCEERealm", "uniqueSecurityName": "Fred" }.

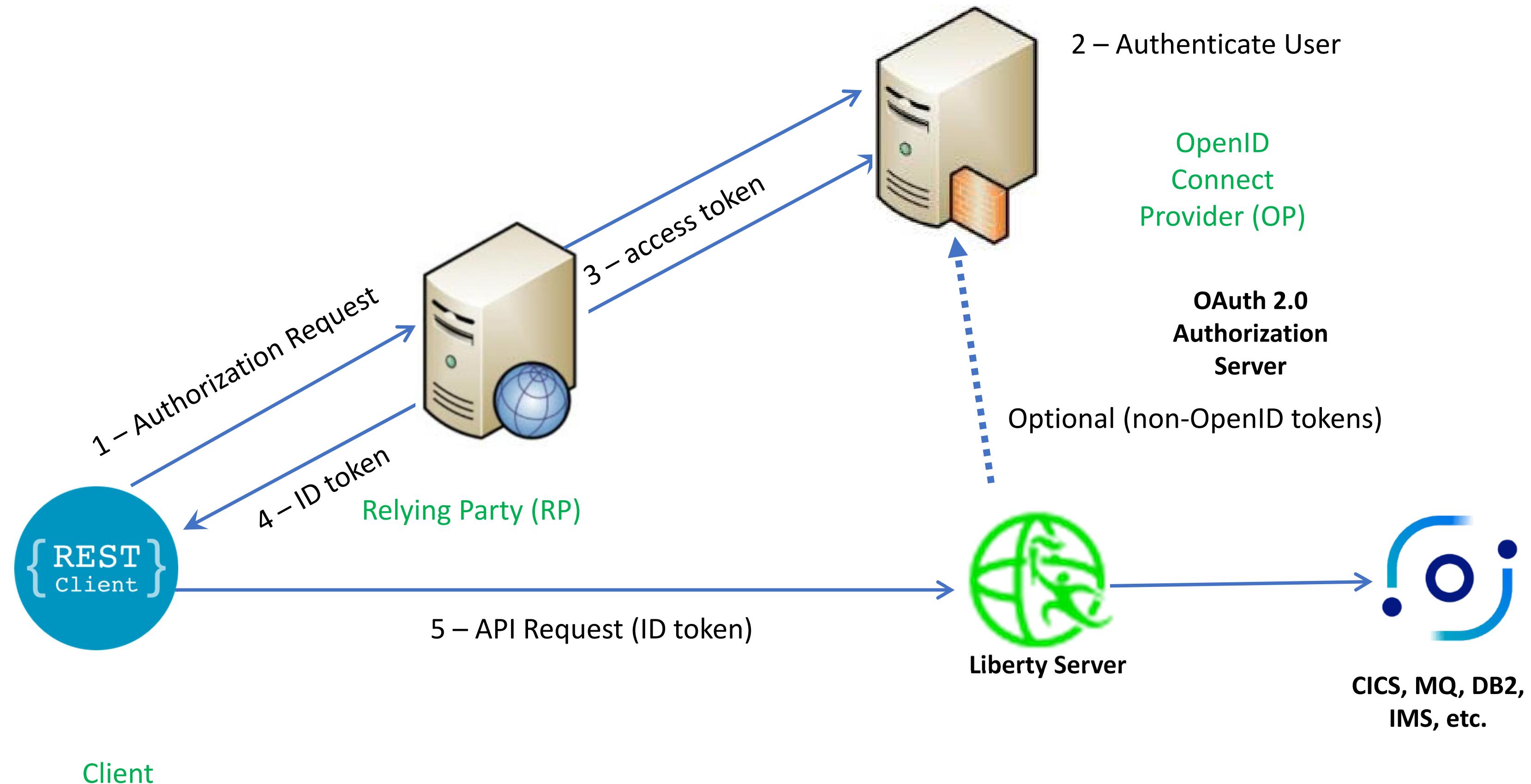
Values derived from the OAUTH configuration:

- signatureAlgorithm="RS256"
- accessTokenLifetime="300"
- resourceIds="myZcee"

<https://jwt.io>



Typical Authorization Flow for an OpenID Connect token to a z/OS Connect API Provider



Some basic OAuth/OpenID Connect terms

- **Authorization server** - The server that issues access tokens to the client after authenticating the resource owner and obtaining authorization. *In a z/OS Connect EE API requester scenario, the authorization server is called by the z/OS Connect EE server to retrieve an access token.*
- **Authorization Endpoint** - A service or endpoint on an OAuth authorization server that accepts an authorization request from a client to perform authentication and authorization of a user. The authorization endpoint returns an authorization grant, or code, to the client in the Authorization Code Flow. In the Implicit Flow, the authorization endpoint returns an access token to the client.
- **Token Endpoint** – A service or endpoint on an OP that accepts an authorization grant, or code, from a client in exchange for an access token, ID token, and refresh token
- **Access Token** – A credential that is used to access protected resources. An access token is a string that represents an authorization that is issued to the client. The access token is usually opaque to the client (it does not have to be opaque) and can be JSON Web Token (JWT). See URL <https://tools.ietf.org/html/rfc6749> Section 1.4 for more information.
- **OAuth token** - With OAuth 2.0, access tokens are used to access protected resources. An access token is normally a string that represents an authorization that is issued to the client. The string is usually opaque to the client. Opaque tokens may require that the token recipient call back to the server that issued the token. *However, an access token can also be in the form of a JSON Web Token (JWT) which does not require a call back (introspection).*
- **Scope** - Privilege or permission that allows access to a set of resources of a third party.

Some basic OAuth/OpenID Connect terms

- **Relying Party (RP)** – An entity that relies on an OP to authenticate a user and obtain an authorization to access a user's resource.
For z/OS Connect API Requester, it is the Liberty server configured as an OpenID Connect Client, e.g., using <openidConnectClient/> XML configuration elements.
- **OpenID Connect Provider (OP)** - An OAuth 2.0 authorization server that is capable of providing claims to a client or Relying Party (RP) , *an OpenID component.*
- **Resource owner** - An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end user. *In a z/OS Connect EE API requester scenario, the resource owner might be the user of the CICS, IMS, or z/OS application.*
- **Resource server** - The server that hosts the protected resources and accepts and responds to protected resource requests by using access tokens. *In a z/OS Connect API provider, the resource server is the z/OS Connect server. In a z/OS Connect EE API requester scenario, the resource server is the request endpoint for the remote RESTful API*
- **ID Token** - is an OpenID Connect token that is an extension to OAuth 2.0 specification access tokens. This token is a JSON Web Token (JWT). See URL https://openid.net/specs/openid-connect-core-1_0.html#IDToken for more information about the extensions.



Tech/Tip: Let's explore a flow using a Liberty OpenID Provider as an example

This Liberty server configuration provides a good example of the workings of an authorization server.

```
<httpEndpoint host="*" httpPort="26212" httpsPort="26213" id="defaultHttpEndpoint"/>

<openidConnectProvider id="OP"
    signatureAlgorithm="RS256"
    keyStoreRef="jwtStore"
    oauthProviderRef="OIDCssl" >
</openidConnectProvider>

<oauthProvider id="OIDCssl"
    httpsRequired="true"
    jwtAccessToken="true"
    autoAuthorize ="true"
    accessTokenLifetime="300">

    <!-- Define OIDC Client for zCEE Authentication -->
    <autoAuthorizeClient>zCEEclient</autoAuthorizeClient>
    <localStore>
        <client name="zCEEClient"
            secret="secret"
            displayname="zCEEClient"
            scope="openid"
            enabled="true"
            resourceIds="myZcee"/>
    </localStore>
</oauthProvider>
```



Key Points:

- **keyStoreRef** - A keystore containing the private key necessary for signing with an asymmetric algorithm.
- **jwtAccessToken** - generate a JSON Web Token, serialize it as a string and put in the place of the access token.

Tech/Tip: Generating a JWT using Liberty's as an example OPID provider

The Liberty server authorization server's XML configuration

```
<!--Key store that contains certificate used to sign JWT-->
<keyStore fileBased="false" id="jwtStore"
  location="safkeyring://JWT.KeyRing"
  password="password" readOnly="true" type="JCERACFKS"/>

<!-- Define a basic user registry -->
<basicRegistry id="basicRegistry"
  realm="zCEERealm">
  <user name="auser" password="pwd"/>
  <user name="distributed_User1" password="pwd"/>
  <user name="Fred" password="fredpwd"/>
  <user name="distuser1" password="pwd"/>
  <user name="distuser2" password="pwd"/>
</basicRegistry>
```

```
RACMAP ID(FRED) MAP USERDIDFILTER(NAME('Fred'))
  REGISTRY(NAME('*')) WITHLABEL('zCEE JWT FRED')
RACMAP ID(USER1) MAP USERDIDFILTER(NAME('distributed_User1'))
  REGISTRY(NAME('*')) WITHLABEL('zCEE JWT distributedUser1')
RACMAP ID(USER1) MAP USERDIDFILTER(NAME('distuser1'))
  REGISTRY(NAME('*')) WITHLABEL('zCEE JWT distuser1')
RACMAP ID(USER2) MAP USERDIDFILTER(NAME('distuser2'))
  REGISTRY(NAME('*')) WITHLABEL('zCEE JWT distuser2')
```

Tech/Tip: RACMAP Command Summary

```
RACMAP ID(USER1) MAP USERDIDFILTER(NAME('distuser1'))
  REGISTRY(NAME('*')) WITHLABEL('zCEE token user1')
RACMAP ID(USER1) MAP USERDIDFILTER(NAME('distribute_User1'))
  REGISTRY(NAME('zCEERealm')) WITHLABEL('zCEE user1')
RACMAP ID(USER1) MAP USERDIDFILTER(NAME('UID=user1,CN=User Name,OU=IBM ATG,O=IBM,C=US'))
  registry(name('*')) withlabel('USER X500 DN')
RACMAP ID(ATSUSER) MAP USERDIDFILTER(NAME('OU=IBM ATS,O=IBM,C=US'))
  registry(name('*')) withlabel('ATS USER')
RACMAP ID(IBMUSER) MAP USERDIDFILTER(NAME('O=IBM,C=US'))
  registry(name('*')) withlabel('IBM USER')
```

```
RACMAP ID(USER1) LISTMAP(LABEL('USER X500 DN'))

RACMAP ID(USER1) DELMAP (LABEL('zCEE distuser1'))

RACMAP QUERY USERDIDFILTER(NAME('USER1')) REGISTRY(NAME('*'))
```

RACMAP ID(USER1) LISTMAP

Label: zCEE token user1
 Distributed Identity User Name Filter:
 >distuser1<
 Registry Name:
 >*<

Label: zCEE user1
 Distributed Identity User Name Filter:
 >distribute_User1<
 Registry Name:
 >zCEERealm<

Label: USER X500 DN
 Distributed Identity User Name Filter:
 >UID=user1,CN=User Name,OU=IBM ATG,O=IBM,C=US<
 Registry Name:
 >*<



Liberty OpenID Client identity mapping configuration attributes

Decoded EDIT THE PAYLOAD AND SECRET

```
HEADER: ALGORITHM & TOKEN TYPE

{
  "kid": "kvjtqdLMjOTWiJrj0r73fu2MMt-FjiQrxU0YBzJLR4o",
  "alg": "RS256"
}

PAYLOAD: DATA

{
  "sub": "auser",
  "token_type": "Bearer",
  "scope": [
    "openid",
    "profile",
    "email"
  ],
  "azp": "rpSsl",
  "iss": "https://wg31.washington.ibm.com:26213
/oidc/endpoint/OP",
  "aud": "myZcee",
  "exp": 1646761228,
  "iat": 1646760928,
  "realmName": "zCEERealm",
  "uniqueSecurityName": "auser"
}
```

```
<safRegistry id="saf" />
<safAuthorization racRouteLog="ASIS" />
<safCredentials unauthenticatedUser="WSGUEST"
  mapDistributedIdentities="true" ←
  profilePrefix="BBGZDFLT" />
```

Use distributed identity filters to map the distributed identities to SAF user IDs, using IDIDMAP resources and the RACMAP command.

```
<authFilter id="ATSAuthFilter">
  <requestUrl id="ATSDemoUrl"
    name="ATSRefererUri"
    matchType="contains"
    urlPattern="/cscvinc/employee|/db2/employee|/mqapi/loan"/>
</authFilter>
<openidConnectClient id="ATS"
  httpsRequired="true"
  authFilterRef="ATSAuthFilter"
  inboundPropagation="required"
  scope="openid profile email"
  audiences="myZcee"
  issuerIdentifier="https://wg31.washington.ibm.com:26213/oidc/endpoint/OP
  mapIdentityToRegistryUser="false" ←
  signatureAlgorithm="RS256"
  userIdentityToCreateSubject="sub"
  trustAliasName="JWT-Signer-Certificate"
  trustStoreRef="jwtTrustStore"
  authnSessionDisabled="true"
  disableLtpaCookie="true">
</openidConnectClient>
<keyStore fileBased="false" id="jwtTrustStore"
  location="safkeyring:///JWT.KeyRing"
  password="password" readOnly="true" type="JCERACFKS"/>
```

Specifies whether to map the identity to a registry user. If this is set to false, then the user registry (SAF) is not used to create the user subject.



Liberty OpenID Client identity mapping configuration attributes (JWK)

```
{  
    "kid": "574eafad-fcb5-412e-97a3-8100a1c1fa5b",  
    "alg": "RS256"  
}  
  
{  
    "sub": "mitchj",  
    "aud": "myZCEE",  
    "iss": "https://wg31.washington.ibm.com:26213/oidc/endpoint/OP",  
    "exp": 1610451176,  
    "iat": 1610451876  
}
```

```
<openidConnectClient  
    id="ATSJWK"  
    clientId="RS-JWT-ZCEE"  
    httpsRequired="true"  
    authFilterRef="jwkAuthFilter"  
    inboundPropagation="required"  
    signatureAlgorithm="RS256"  
    userIdentifier="sub"  
    mapIdentityToRegistryUser="true"  
    issuerIdentifier="https://wg31.washington.ibm.com:26213/oidc/endpoint/OP"  
    disableItpaCookie="true"  
    audiences="myZcee"  
    tokenReuse="true"  
    jwkEndpointUrl="https://wg31.washington.ibm.com:26213/oidc/endpoint/OP/jwk"  
    jwkClientId="jwtClient"  
    jwkSecret="jwtSecret"/>  
</openidConnectClient>
```



JWT used in scenario – putting it all together

```
{
  "alg": "RS256"
}

{
  "sub": "Edward Johnson",
  "token_type": "Bearer",
  "azp": "rpSsl",
  "iss": "https://wg31.washington.ibm.com:26213/oidc/endpoint/OPssl",
  "aud": "myZcee",
  "realmName": "zCEERealm",
  "uniqueSecurityName": "Edward Johnson"
}
RSASHA256(base64UrlEncode(header) + base64UrlEncode(payload))
```

- The header contains an **alg** (algorithm) element value **RS256**
 - **RS256** (RSA Signature with SHA-256) is an asymmetric algorithm which uses a **public/private** key pair
 - **ES512** (Elliptic Curve Digital Signature Algorithm with SHA-512) [link for more info](#)
 - **HS256** (HMAC with SHA-256) is a symmetric algorithm with only one (**secret**) key
- The **iss** (issuer) claim identifies the principal that issued the JWT
- The **sub** (subject) claim **distuser** identifies the principal that is the subject of the JWT
- The **aud** (audience) claim **myZcee** identifies the recipients for which the JWT is intended



Configuring authentication with JWT

Liberty can perform user authentication with JWT using the support that is provided by the *openidConnectClient-1.0* feature. The **<openidConnectClient>** element is used to accept a JWT token as an authentication token

```
<openidConnectClient id="RPssl" inboundPropagation="required"
    signatureAlgorithm="RS256" trustAliasName="JWT-Signer"
    trustStoreRef="jwtTrustStore"
    userIdentityToCreateSubject="sub" mapIdentityToRegistryUser="false"
    issuerIdentifier="https://wg31.washington.ibm.com:26213/oidc/endpoint/OPssl"
    authnSessionDisabled="true" audiences="myZcee"/>
```

- ***inboundPropagation*** is set to required to allow z/OS Connect EE to use the received JWT as an authentication token
- ***signatureAlgorithm*** specifies the algorithm to be used to verify the JWT signature
- ***trustStoreRef*** specifies the name of the keystore element that defines the location of the validating certificate
- ***trustAliasName*** gives the alias or label of the certificate to be used for signature validation
- ***userIdentityToCreateSubject*** indicates the claim to use to create the user subject
- ***mapIdentityToRegistryUser*** indicates whether to map the retrieved identity to the registry user
- ***issuerIdentifier*** defines the expected issuer
- ***authnSessionDisabled*** indicates whether a WebSphere custom cookie should be generated for the session
- ***audiences*** defines a list of target audiences



Use authorization filters to associate request for security configurations

Authentication filter can be used to filter criteria that are specified in the **authFilter** element to determine whether certain requests are processed by certain providers, such as OpenID Connect, for authentication.

```
<openidConnectClient id="RPssl" inboundPropagation="required"
    signatureAlgorithm="RS256" trustAliasName="JWT-Signer"
    trustStoreRef="jwtTrustStore"
    userIdentityToCreateSubject="sub" mapIdentityToRegistryUser= "true"
    issuerIdentifier="https://wg31.washington.ibm.com:26213/oidc/endpoint/OPssl"
    authnSessionDisabled="true" audiences="myZcee"
    authFilterRef="JwtAuthFilter"/>
<openidConnectClient id="RPsslG" . . . authFilterRef= "API Gateway" />
<openidConnectClient id="RPsslURL" . . . authFilterRef= "URLFilter" />
<authFilter id="API Gateway">
    <remoteAddress id="ApiAddress" ip="10.7.1.*" matchType="equals"/>
</authFilter>
<authFilter id="URLFilter">
    <requestUrl id="URL" urlPattern="/cscvinc/employee|/db2/employee|/mqapi/loan" />
    matchType="equals"/> </authFilter>
<authFilter id="JwtAuthFilter" >
    <requestHeader id="authHeader" name="Authorization" value="Bearer" matchType="contains"/>
</authFilter>
```

Some alternative filter types

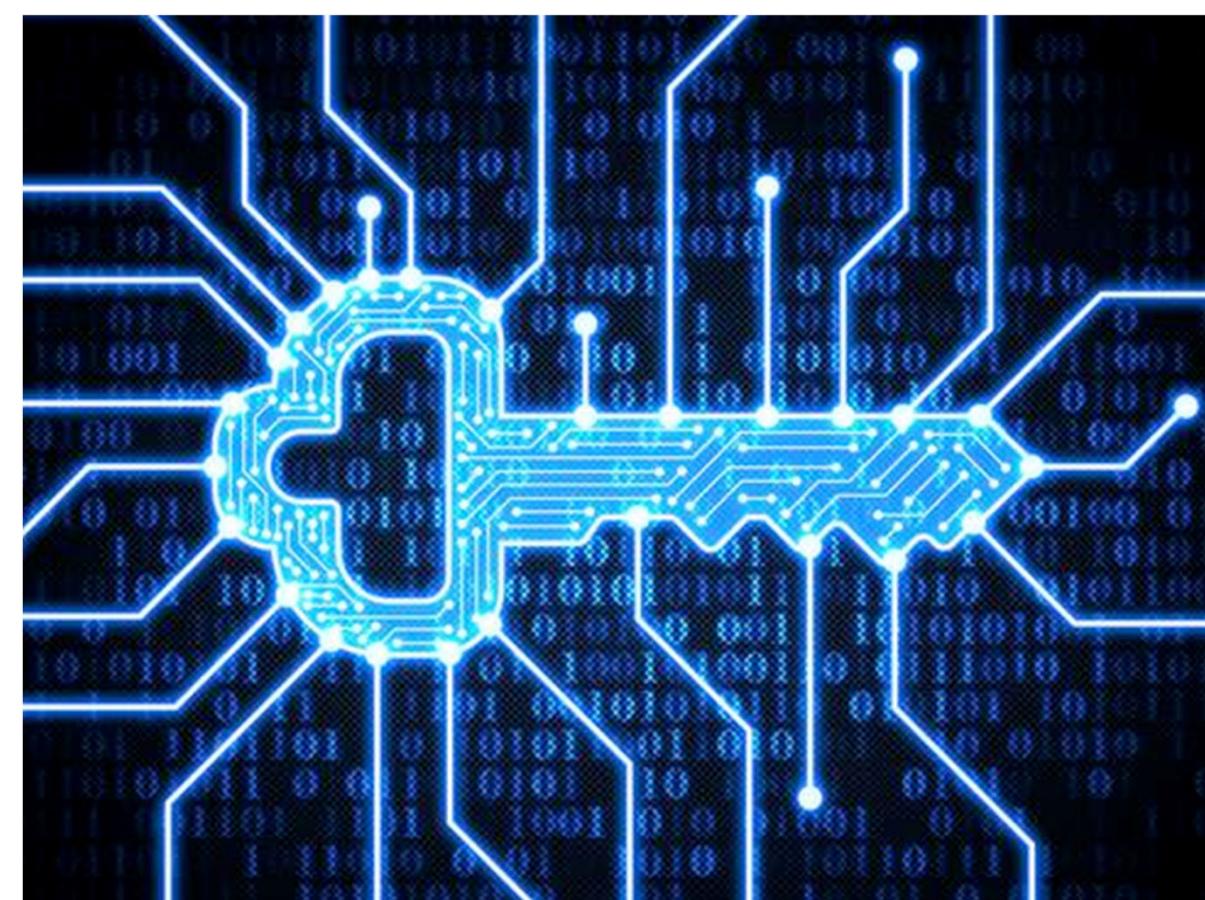
- A **remoteAddress** element is compared against the TCP/IP address of the client that sent the request.
- The **host** element is compared against the "Host" HTTP request header, which identifies the target host name of the request.
- The **requestUrl** element is compared against the URL that is used by the client application to make the request.

General security terms or considerations

Security involves

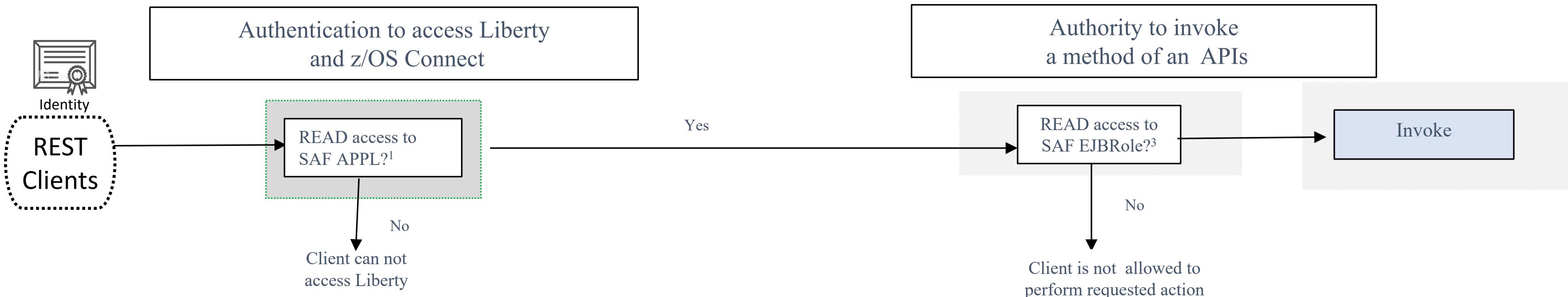
- Identifying who or what is requesting access (**Authentication**)
 - Basic Authentication
 - Mutual Authentication using Transport Layer Security (TLS), formerly known as SSL
 - Third Party Tokens
- Ensuring that the message has not been altered in transit (**Data Integrity**) and ensuring the confidentiality of the message in transit (**Encryption**)
 - TLS (encrypting messages and using a digital signature)

- Controlling access (**Authorization**)
 - Is the authenticated identity authorized to access to z/OS Connect
 - Is the authenticated identity authorized to access a specific API, Services, etc.

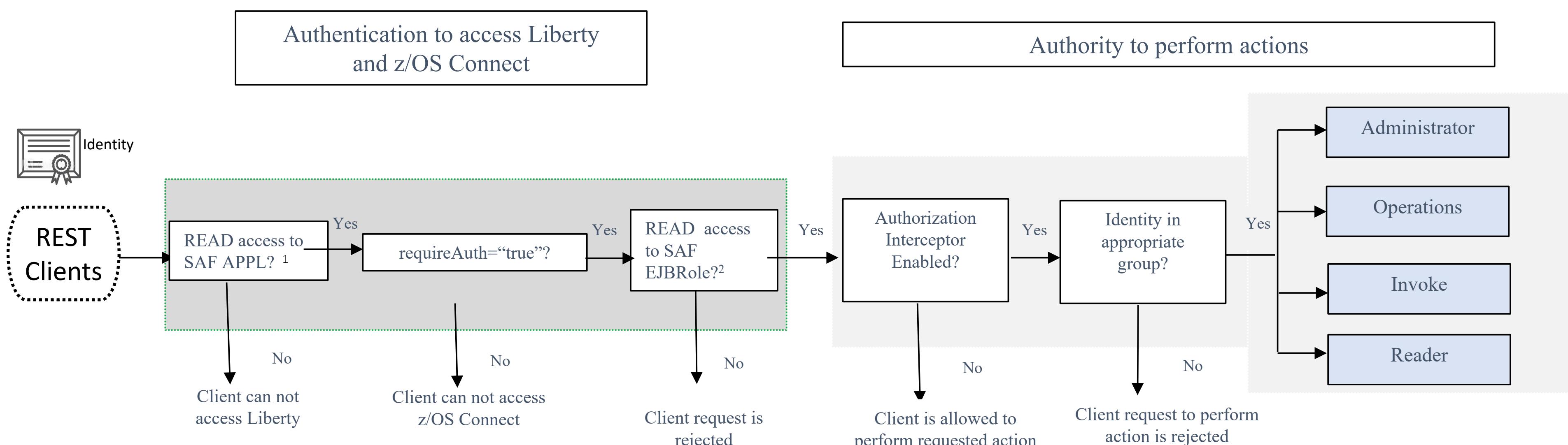




Security flow – authentication/authorization



(OpenAPI 3)



(OpenAPI 2)

¹RDEFINE APPL *profilePrefix*

²RDEFINE EJBROLE *profilePrefix.zos.connect.access.roles.zosConnectAccess*

³REDEFINE EJBROLE *profilePrefix.resourceName.role*

Security for OpenAPI 3 z/OS Connect APIs is configured using Liberty elements



```
<safCredentials unauthenticatedUser="WSGUEST" profilePrefix="BBGZDFLT" />  
  
<webApplication id="catalogManager" name="catalogManager"  
    location="${server.config.dir}/apps/api.war" contextRoot="/catalogManager" />  
  
<safRoleMapper profilePattern=%profilePrefix%.%resourceName%.%role%
```

The *name* attribute of the *webApplication* for the deployed WAR file determines the name of the EJBRoles used manage access to the API's methods.

*From the example OpenApi document, the value for %role% would be either **Manager** or **Staff**.*

So, the required SAF EJB roles to be defined would be:

- *BBGZDFLT.catalogManager.Manager*
- *BBGZDFLT.catalogManager.Staff*

*REDFINE EJBROLE BBGZDFLT.catalogManager.Manager
REDFINE EJBROLE BBGZDFLT.catalogManager.Staff*

Access to use the GET method to invoke /items would require read access to EJB role
BBGZDFLT.catalogManager.Manager.

Access to use the GET method to invoke /items/{id} and the POST method to invoke /orders would require read access to EJB role *BBGZDFLT.catalogManager.Staff*.



API Requester – authorization (OpenAPI 3 only)

```
<safCredentials unauthenticatedUser="WSGUEST" profilePrefix="BBGZDFLT" />

<safRoleMapper profilePattern=%profilePrefix%.%resourceName%.%role%>

<webApplication location="${server.config.dir}/apps/cscvinc.war">
  <appProperties> <property name="connectionRef" value="cscvincConnection"/> </appProperties>
</webApplication>

<webApplication name="catalogManager" location="${server.config.dir}/apps/catalog.war">
  <appProperties> <property name="connectionRef" value="catalogConnection"/> </appProperties>
</webApplication>
```

The *resourceName* defaults to the name of the WAR file if no name attribute is provided, otherwise the *resourceName* is value of the *name* attribute.

So, the required SAF EJB roles to be defined would be (*invoke* is the only role).

- *BBGZDFLT.cscvinc.invoke*
- *BBGZDFLT.catalogManager.invoke*

Authorization to invoke the API requester would require that the authenticated identity be a member of the STAFFGROUP or identity FRED.



Security for MQ Console and REST

```
/usr/lpp/mqm/web/mq/etc/mqweb.xml
<enterpriseApplication id="com.ibm.mq.console" location="${wlp.install.dir}/mq/apps/com.ibm.mq.webconsole.ear"
name="com.ibm.mq.console" . . .
</enterpriseApplication>

<enterpriseApplication id="com.ibm.mq.rest" location="${wlp.install.dir}/mq/apps/com.ibm.mq.rest.ear"
name="com.ibm.mq.rest" . . .
</enterpriseApplication>
```

```
/var/mqm/servers/mqweb/mqwebuser.xml
<safCredentials profilePrefix="MQWEB" unauthenticatedUser="WSGUEST"/
```

So, the required SAF EJB roles to be defined would be:

- REDFINE EJBROLE *MQWEB.com.ibm.mq.console.MQWebAdmin*
- REDFINE EJBROLE *MQWEB.com.ibm.mq.console.MQWebAdminRO*
- REDFINE EJBROLE *MQWEB.com.ibm.mq.console.MQWebUser*
- REDFINE EJBROLE *MQWEB.com.ibm.mq.rest.MFTWebAdmin*
- REDFINE EJBROLE *MQWEB.com.ibm.mq.rest.MFTWebAdminRO*
- REDFINE EJBROLE *MQWEB.com.ibm.mq.rest.MQWebAdmin*
- REDFINE EJBROLE *MQWEB.com.ibm.mq.rest.MQWebAdminRO*
- REDFINE EJBROLE *MQWEB.com.ibm.mq.rest.MQWebUser* .

<https://www.ibm.com/docs/en/ibm-mq/9.3?topic=roles-mq-console-rest-api>

mitchj@us.ibm.com

© 2017, 2024 IBM Corporation
Slide 117



z/OS Connect authentication/authorization is based on group access

z/OS Connect uses group security for controlling authorization for accessing APIs. There are sets of default global groups for functional roles are configured in a `zosConnectManager` configuration element as shown below:

```
<zosconnect_zosConnectManager  
    globalInterceptorsRef="interceptorList_g"  
    globalAdminGroup="SYSPGRP" globalOperationsGroup="GBLOPERS"  
    globalInvokeGroup="GBLINVKE" globalReaderGroup="GBLRDR"/>
```

There are four classes of groups available controlling z/OS Connect functions, administration, operations, invoking and reader in our server. An authenticated identity membership in one or more of these groups provides access to the corresponding function to that identity.

There is also a way to provide an alternative set of groups for functional roles for specific APIs, services, and API requesters in subordinate configuration elements in our server.

```
<zosConnectAPI name="cscvinc"  
    adminGroup="CSCADMIN" operationsGroup="CSCOPERS"  
    invokeGroup="CSCINVKE" readerGroup="CSCREADR"/>  
  
<service name="cscvincSelectService"  
    adminGroup="CSCADMIN" operationsGroup="CSCOPERS"  
    invokeGroup="CSCINVKE" readerGroup="CSCREADR"/>  
  
<apiRequester name="cscvinc_1.0.0"  
    adminGroup="CSCADMIN" operationsGroup="CSCOPERS"  
    invokeGroup="CSCINVKE" readerGroup="CSCREADR"/>
```



z/OS Connect interceptors - server XML example

```
<zosconnect_zosConnectManager  
    globalInterceptorsRef="interceptorList_g"  
    globalAdminGroup="SYSPGRP"  
    globalOperationsGroup="GBLOPERS"  
    globalInvokeGroup="GBLINVKE"  
    globalReaderGroup="GBLRDR"/>  
  
<zosconnect_authorizationInterceptor id="auth"/>  
<zosconnect_auditInterceptor id="audit"/>  
<zosconnect_zosConnectInterceptors id="interceptorList_g"  
    interceptorRef="auth"/>  
<zosconnect_zosConnectInterceptors id="interceptorList_a"  
    interceptorRef="auth,audit"/>  
  
<zosconnect_zosConnectAPIs>  
    <zosConnectAPI name="catalog"  
        runGlobalInterceptorsRef="true"  
        adminGroup="aapigrp1,aapigrp2"  
        operationsGroup="oapigrp1,oapigrp2"  
        invokeGroup="iapigrp1,oapigrp2"  
        readerGroup="rapigrp1,rapigrp2"/>  
</zosconnect_zosConnectAPIs>  
  
<zosconnect_apiRequesters>  
    <apiRequester name="cscvincapi_1.0.0"  
        runGlobalInterceptorsRef="false"  
        interceptorsRef="interceptorList_a"  
        adminGroup="aaprgrp1,aaprgrp2"  
        operationsGroup="oaprgrp1,oaprgrp2"  
        invokeGroup="iaprgrp1,oaprgrp2"  
        readerGroup="raprgrp1,raprgrp2"/>  
</zosconnect_apiRequesters>  
  
<zosconnect_services>  
    <service id="selectByEmployee" name="selectEmployee"  
        runGlobalInterceptorsRef="false"  
        interceptorsRef="interceptorList_a"  
        adminGroup="asrvgrp1,asrvgrp2"  
        operationsGroup="osrvgrp1,osrvgrp2"  
        invokeGroup="isrvgrp1,isrvgrp2"  
        readerGroup="rsrvrgrp1,rsrvgrp2"/>  
</zosconnect_services>
```

Global interceptor list –
authorization
interceptor only

Alternative interceptor
list – authorization and
audit interceptors

This avoids duplication
of interceptors

Note that these are z/OS
Connect configuration
elements. Documented in the
z/OS Connect KC

z/OS Connect OpenAPI 2 RESTful Administrative APIs



z/OS Connect administration API		
Interface providing meta-data and life-cycle operations for z/OS Connect services, APIs and API requesters.		
APIs : Operations for working with APIs		
GET	/apis	Show/Hide List Operations Expand Operations Returns a list of all the deployed z/OS Connect APIs
POST	/apis	Deploys a new API into z/OS Connect
DELETE	/apis/{apiName}	Undeploys an API from z/OS Connect
GET	/apis/{apiName}	Returns detailed information about a z/OS Connect API
PUT	/apis/{apiName}	Updates an existing z/OS Connect API
Services : Operations for working with services		
GET	/services	Show/Hide List Operations Expand Operations Returns a list of all the deployed z/OS Connect services
POST	/services	Deploys a new service into z/OS Connect
DELETE	/services/{serviceName}	Undeploys a service from z/OS Connect
GET	/services/{serviceName}	Returns detailed information about a z/OS Connect service
PUT	/services/{serviceName}	Updates an existing z/OS Connect service
GET	/services/{serviceName}/schema/{schemaType}	Returns the request or response schema for a z/OS Connect service
API Requesters : Operations that work with API Requesters.		
GET	/apiRequesters	Show/Hide List Operations Expand Operations Returns a list of all the deployed z/OS Connect API Requesters
POST	/apiRequesters	Deploys a new API Requester into z/OS Connect and invoke an API Requester call
DELETE	/apiRequesters/{apiRequesterName}	Undeploys an API Requester from z/OS Connect
GET	/apiRequesters/{apiRequesterName}	Returns the detailed information about a z/OS Connect API Requester
PUT	/apiRequesters/{apiRequesterName}	Updates an existing z/OS Connect API Requester



z/OS Connect Authorization Functions

Operations - Ability to perform all z/OS Connect EE operations and actions except for function *Invoke*. The following operations/actions are allowed:

APIs:

- *To obtain a list of all APIs (GET).**
- For a specific API, get its details and API Swagger document (GET) and *deploy (POST)**, update (PUT), start(PUT), stop(PUT), and delete(DELETE) it.

Services:

- *To obtain a list of all services or statistics for all services (GET).**
- For a specific service, get its details, request and response schemas, statistics (GET) and *deploy(POST)**, update(PUT), start(PUT), stop(PUT), and delete(DELETE) it.

API Requesters:

- *To obtain a list of all API requesters (GET).**
- For a specific API requester, get its details (GET) and *deploy (POST)**, update(PUT), start(PUT), stop(PUT), and delete(DELETE) it.

*These APIs use either the POST or GET method to invoke the REST APIs whose URIs have no path parameter. Therefore, the name of the API, or service or API Requester is not available. For authorization, only the default or global groups list can be used since no specific group list can be determined (for deployment, the name is embedded in the archive file).



z/OS Connect Authorization Levels

Reader - Ability for:

APIs:

- *To obtain a list of all APIs (GET) . **
- For a specific API, get its details and API Swagger document (GET).

Services:

- *To obtain a list of all services (GET) . **
- For a specific service, get its details and request and response schemas (GET).

API Requesters:

- *To obtain a list of all API requesters (GET) . **
- For a specific API requester, get its details (GET) .

Invoke - Ability to invoke user APIs, services and/or API requesters (POST,PUT,GET,DELETE,+).

Admin - All z/OS Connect EE actions are allowed, including all corresponding *Operations*, *Invoke*, and *Reader* actions configured for the same z/OS Connect resource.

*These APIs use either the POST or GET method to invoke the REST APIs whose URIs have no path parameter. Therefore, the name of the API, service or API Requester is not available. For authorization, only the default or global groups list since no specific group list can be determined (for deployment, the name is embedded in the archive file).

Example of z/OS Connect Authorization Levels



(this config has issues)

```
<zosconnect_zosConnectManager
    globalInterceptorsRef="interceptorList_g"
    globalAdminGroup="SYSPGRP" globalOperationsGroup="GBLOPERS"
    globalInvokeGroup="GBLINVKE" globalReaderGroup="GBLRDR"/>

<zosconnect_zosConnectAPIs>
    <zosConnectAPI name="cscvinc"
        adminGroup="CSCADMIN" operationsGroup="CSCOPERS"
        invokeGroup="CSCINVKE" readerGroup="CSCREADR"/>
    <zosConnectAPI name="db2employee"
        adminGroup="DB2ADMIN" operationsGroup="DB2OPERS"
        invokeGroup="DB2INVKE" readerGroup="DB2READR"/>
</zosconnect_zosConnectAPIs>

<zosconnect_services>
    <service name="cscvincSelectService"
        adminGroup="CSCADMIN" operationsGroup="CSCOPERS"
        invokeGroup="CSCINVKE" readerGroup="CSCREADR"/>
    <service name="selectEmployee"
        adminGroup="DB2ADMIN" operationsGroup="DB2OPERS"
        invokeGroup="DB2INVKE" readerGroup="DB2READR"/>
</zosconnect_services>

<zosconnect_apiRequesters>
    <apiRequester name="cscvincSelectService"
        adminGroup="CSCADMIN" operationsGroup="CSCOPERS"
        invokeGroup="CSCINVKE" readerGroup="CSCREADR"/>
    <apiRequester name="selectEmployee"
        adminGroup="DB2ADMIN" operationsGroup="DB2OPERS"
        invokeGroup="DB2INVKE" readerGroup="DB2READR"/>
</zosconnect_apiRequesters>
```

This works as you expect once the artifacts are deployed but:

- Only members of groups SYSPGRP, GBLOPERS or GBLRDR can connect to a z/OS server from the API toolkit (the tooling attempts a GET request for a list of all deployed services and APIs).
- Only members of groups SYSPGRP or GBLOPERS can deploy new z/OS Connect API, service or API requester artifacts (POST access for operations is not available until after the artifact is deployed)

Tech-Tip: When groups are specified for zosConnectAPI, service, or apiRequester configuration elements, the global groups are ignored for certain functions. Other functions, e.g., deploy new artifact, get a list or service statistics, only use the global group membership.

z/OS Connect Authorization Summary (OpenAPI 2)



- Members of groups SYSPGRP, GBLOPERS, DB2ADMIN or DB2OPERS can not manage (e.g., change, stop or delete) z/OS Connect artifacts managed by group CSCOPERS or CSCADMIN.
- Members of groups SYSPGRP, GBLOPERS, CSCADMIN or CSCOPERS can not manage (e.g., change, stop or delete) z/OS Connect artifacts managed by group DB2OPERS or DB2ADMIN.
- Only members of group CSCADMIN, CSCINV, DB2ADMIN or DB2INVKE can invoke the artifacts defined in the subordinate element:
- Members of group CSCADMIN or CSCINVKE can invoke artifacts managed by CSCINVKE
- Members of group DB2ADMIN or DB2INVKE can invoke artifacts managed by DB2INVKE
- Members of groups SYSPGRP or GBLINVKE can not invoke any artifacts protected these specific subordinate groups.
- Only members of groups SYSPGRP, GBLOPERS or GBLRDR can connect to a z/OS server from the API toolkit.
- Only members of groups SYSPGRP or GBLOPERS can deploy new z/OS Connect API, service or API requester artifacts.



Tech-Tip: Solution for z/OS Connect Authorization Levels (OpenAPI 2)

```
<zosconnect_zosConnectManager  
    globalInterceptorsRef="interceptorList_g"  
    globalAdminGroup="SYSPGRP" globalOperationsGroup="GBLOPERS , CSCOPERS , DB2OPERS"  
    globalInvokeGroup="GBLINVKE" globalReaderGroup="GBLRDR"/>  
  
<zosconnect_zosConnectAPIs>  
    <zosConnectAPI name="cscvinc" operationsGroup="CSCOPERS" invokeGroup="CSCINV" />  
    <zosConnectAPI name="db2employee" operationsGroup="DB2OPERS" invokeGroup="DB2INVKE" />  
</zosconnect_zosConnectAPIs>  
  
<zosconnect_services>  
    <service name="cscvincSelectService" operationsGroup="CSCOPERS" invokeGroup="CSCINV" />  
    <service name="selectEmployee" operationsGroup="DB2OPERS" invokeGroup="DB2INVKE" />  
</zosconnect_services>  
  
<zosconnect_apiRequesters>  
    <apiRequester name="cscvincSelectService" operationsGroup="CSCOPERS" invokeGroup="CSCINV" />  
    <apiRequester name="selectEmployee" operationsGroup="DB2OPERS" invokeGroup="DB2INVKE" />  
</zosconnect_apiRequesters>
```

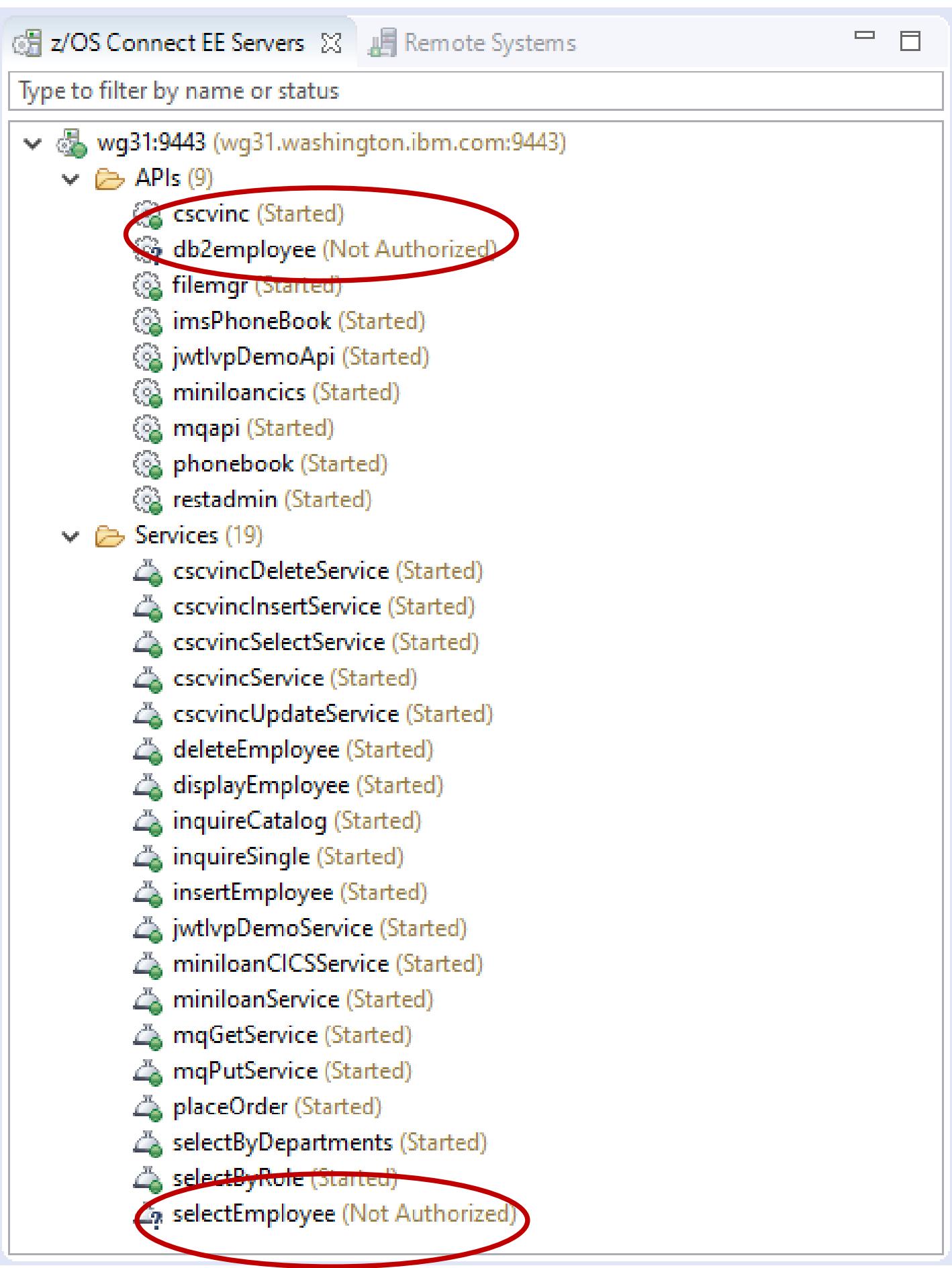
- Now members of groups SYSPGRP, **GBLOPERS, CSCOPERS, DB2OPERS** and GBLRDR can connect to a z/OS server from the API toolkit.
- Members of groups SYSPGRP, **GBLOPERS, CSCOPERS**, and **DB2OPERS** can deploy new artifacts.
- Only members of group **CSCOPERS** and **DB2OPERS** can manage artifacts after they are deployed.



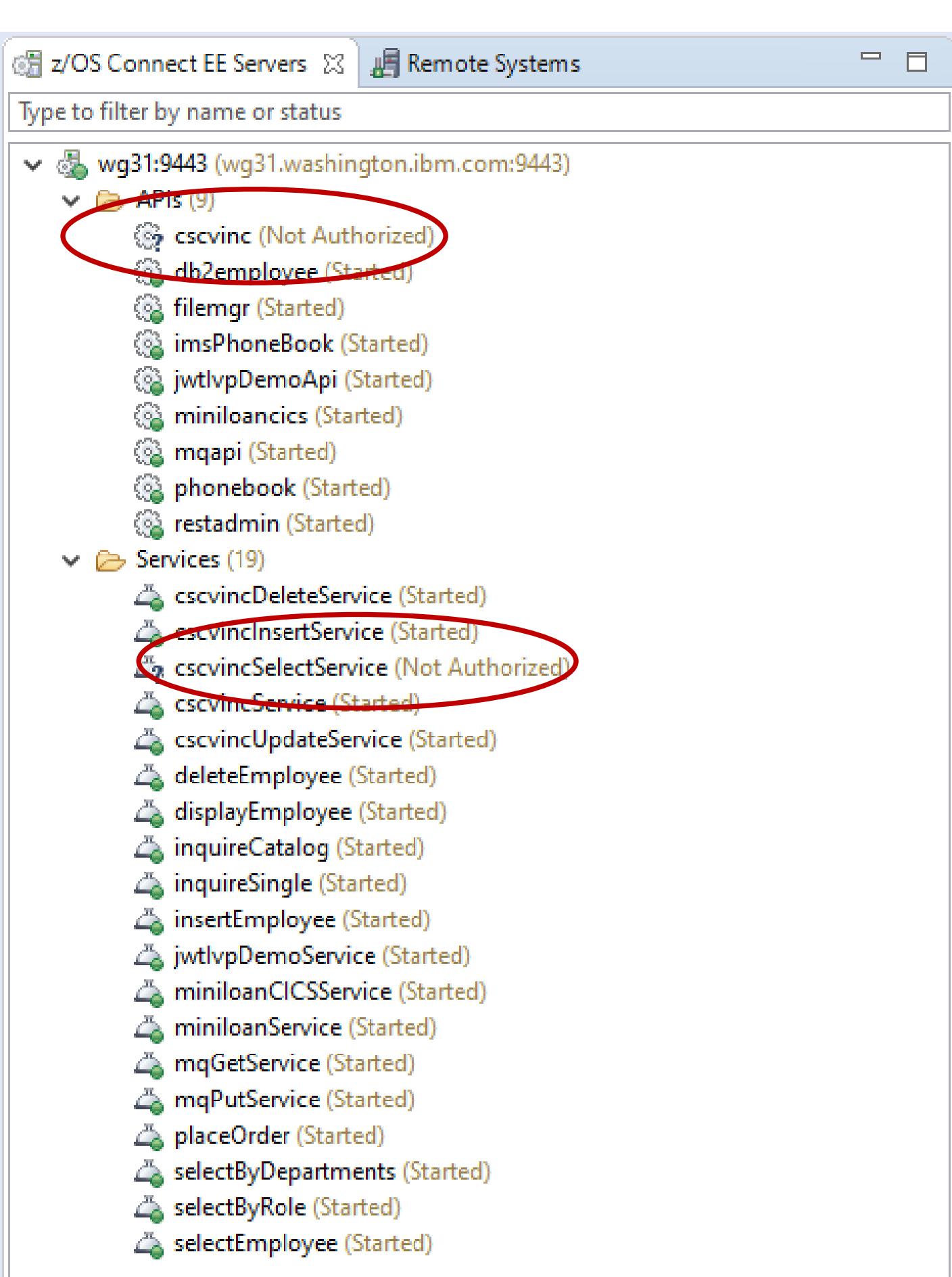
Tech-Tip: z/OS Toolkit and authorization status (OpenAPI 2)

Members of CSCOPERS and DB2OPERS can now connect to a server from the API Toolkit

CSCOPERS



DB2OPERS



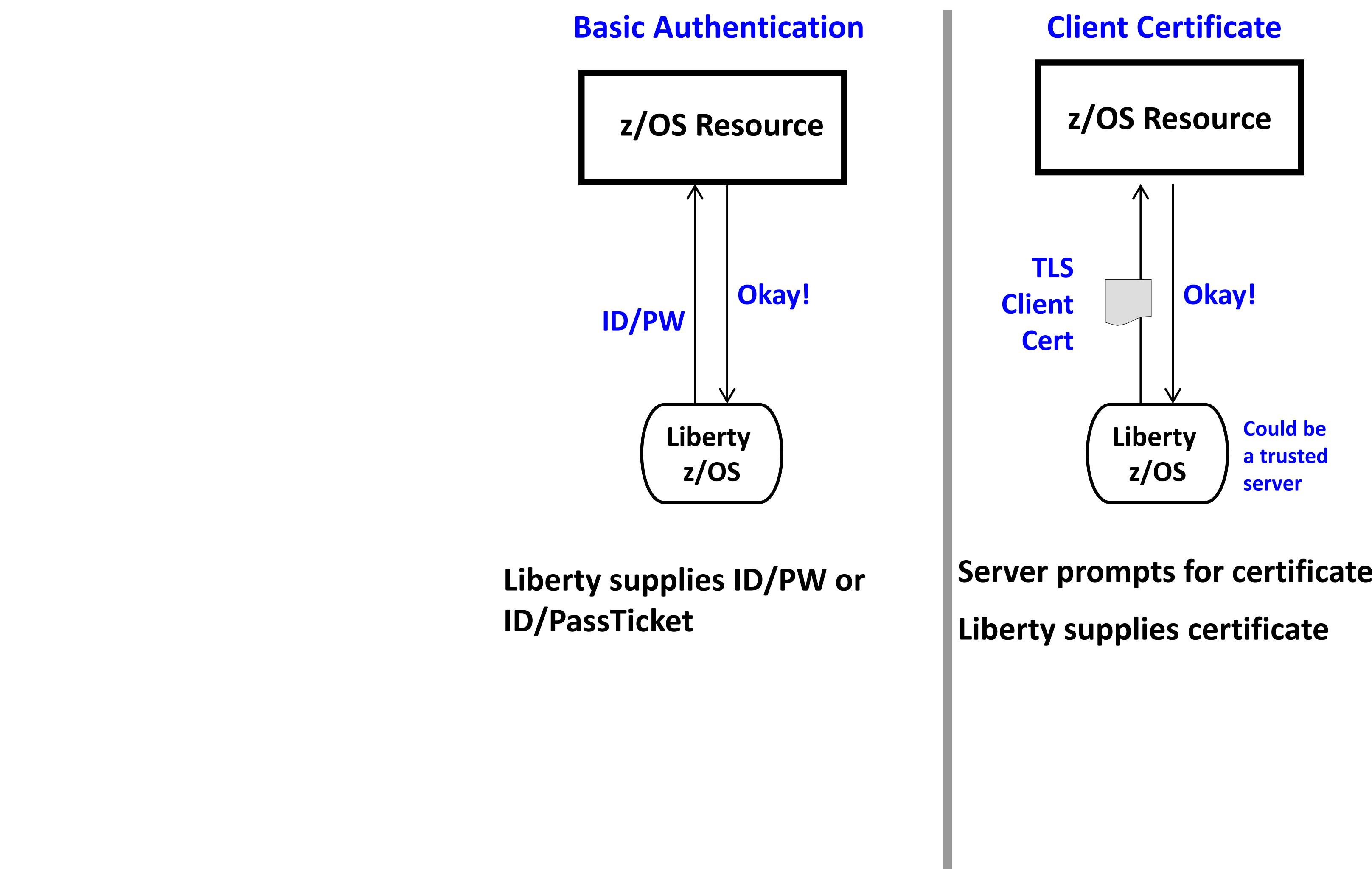


Security when accessing z/OS subsystems



Accessing z/OS resources

Several different ways this can be accomplished:





Liberty Basic authentication - Identity and Password

Server XML Configuration elements where basic authentication can be provided.

```
<resourceAdapter autoStart="true" id="eciResourceAdapter" location="/usr/lpp/cicstg/ctg93/deployable/cicseci.rar"/>
<library id="DB2JCCLib">
    <fileset dir="/usr/lpp/db2/jdbc/classes includes="db2jcc4.jar db2jcc_license_cisuz.jar"/>
</library>

<connectionFactory id="cics">
<properties.eciResourceAdapter connectionURL="tcp://wg31.Washington.ibm.com" port="2006" serverName="CICSZ" userName="identity"
password="password"/>
</connectionFactory>

<connectionFactory id="imsTM"> containerAuthDataRef="IMScredentials">
<authData id="IMScredentials" user= "identity" password= "password"/>

<connectionFactory id="imsDB">
<properties.imsudbJLocal databaseName="DFSIIVPA" user="identity" password="password"/>
</connectionFactory>

<jmsQueueConnectionFactory jndiName="MQ">
    <properties.wasJms userName="identity" password="password" />
</jmsQueueConnectionFactory>

<dataSource id="DefaultDataSource" jndiName="jdbc/db2"">
<jdbcDriver libraryRef="DB2JCCLib"/>
    <properties.db2.jcc databaseName="SAMPLEDB" serverName="localhost" portNumber="50000"
        user="identity" password="password"/>
</dataSource>
```



z/OS Connect Basic authentication - Identity and Password

Server XML Configuration elements where basic authentication can be provided.

```
<connectionFactory id="imsTM"> containerAuthDataRef="IMScredentials">
<authData id="IMScredentials" user= "identity" password= "password"/>

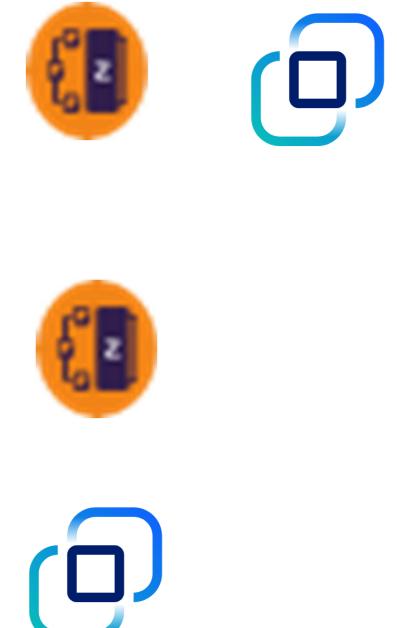
<connectionFactory id="imsDB">
<properties.imsudbJLocal databaseName="DFSIIVPA" user="identity" password="password"/>
</connectionFactory>

<jmsQueueConnectionFactory jndiName="MQ">
    <properties.wasJms userName="identity" password="password" />
</jmsQueueConnectionFactory>

<zosconnect_cicsIpicConnection id="CICS" authDataRef="CICScredentials"/>
<zosconnect_authData id="CICScredentials" user= "identity" password= "password"/>

<zosconnect_zosConnectServiceRestClientConnection id="Db2" basicAuthRef="db2Auth"/>
<zosconnect_zosConnectServiceRestClientBasicAuth id="db2Auth"
    userName="identity" password="password"/>

<zosconnect_db2Connection id="Db2" host="wg31.Washington.ibm.com" port='2446'
    userName="identity" password="password"/>
```



The value of the password can be encoded in the server XML configuration file. Using the **securityUtility** shipped with WebSphere Liberty Profile.



Using securityUtility to encrypt passwords

Best practice : use encryption for passwords instead of base64 encoding

- **SecurityUtility** – located in <wlp_install_dir>/wlp/bin Usage: securityUtility {encode|createSSLCertificate|help} [options]

- For encryption, use encode --key=encryption_key

- Specifies the key to be used when encoding using AES encryption. This string is hashed to produce an encryption key that is used to encrypt and decrypt the password. The key can be provided to the server by defining the variable **wlp.password.encryption.key** whose value is the key. If this option is not provided, a default key is used.

```
./securityUtility encode --encoding=aes --key=myKey myPassWord  
{aes}AHO0aXdiVD96u4oMRhoKeYH3U7aDqtFXTuHFBsO98Wlb
```

- Support was added at Liberty 22.0.0.1 for storing an AES password encryption key in a SAF key ring, see URL
<https://www.ibm.com/docs/en/was-liberty/zos?topic=slia-storing-aes-password-encryption-key-in-saf-key-ring>

```
./securityUtility encode --encoding=aes --keyring=safkeyring://JOHNSON/Liberty.KeyRing --keyringType=JCERACFKS  
--keyLabel="Johnson Client Cert" myPassWord
```

- Also supports 1-way hash encoding – for passwords in server.xml with basicRegistry

- For hash, use encode --encoding=hash

```
./securityUtility encode --encoding=hash XXXXXXXX
```

```
{hash}ATAAAAAIcqTmHn5qZahAAAAAIMjzy+hP8YFaIO6LiCreVe4etRLUS9a25eVuYtx6WKiv
```

See the WebSphere Application Server for z/OS Liberty *securityUtility* command at URL:

<https://www.ibm.com/docs/en/was-liberty/zos?topic=applications-securityutility-command>



Tech-Tip: Sample JCL - Executing the Liberty *securityUtility* command

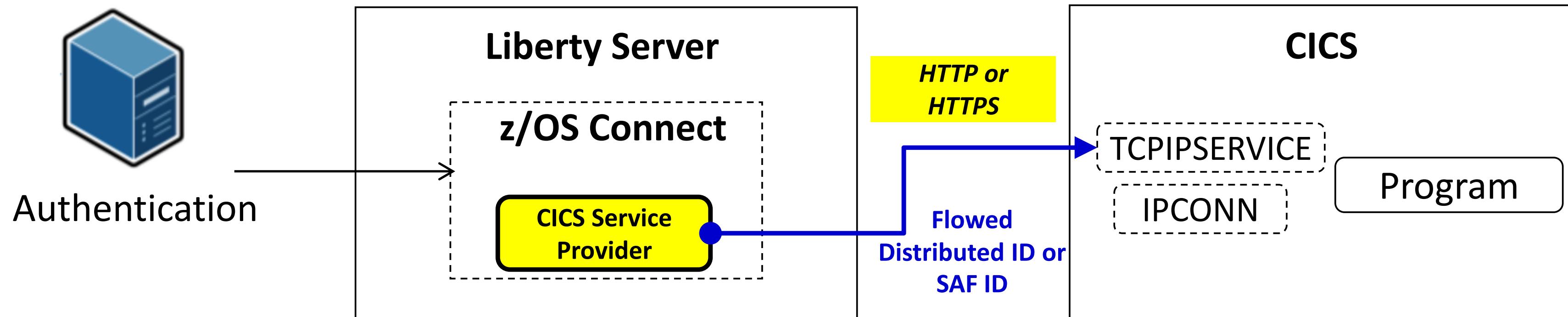
```
//*****  
/* Use securityUtility to encrypt a password using an  
/* encryption key of a certificate  
//*****  
//IKJEFT01 EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *  
BPXBATCH SH +  
/usr/lpp/IBM/zosconnect/v3r0/wlp/bin/securityUtility encode +  
--encoding=aes +  
--keyring=safkeyring://JOHNSON/Liberty.KeyRing +  
--keyringType=JCERACFKS --keyLabel="Johnson Client Cert" +  
passwordToEncrypt
```

```
<featureManager>  
  <feature>zosPasswordEncryptionKey-1.0</feature>  
</featureManager>  
  
<zosPasswordEncryptionKey  
keyring="safkeyring://JOHNSON/Liberty.KeyRing"  
label="Johnson Client Cert" type="JCERACFKS"/>
```

```
//*****  
/* Use securityUtility to encrypt a password using an  
/* encryption key string  
//*****  
//IKJEFT01 EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *  
BPXBATCH SH +  
/usr/lpp/IBM/zosconnect/v3r0/wlp/bin/securityUtility encode +  
--encoding=aes -key myEncryptionKey +  
passwordToEncrypt
```

```
wlp.password.encryption.key=myEncryptionKey
```

Flowing a user ID with CICS service provider



Distributed identities can be propagated to CICS and then mapped to a RACF user ID by CICS. You can then view the distinguished name and realm for a distributed identity in the association data of the CICS task. **Important:** If the z/OS Connect server is not in the same Sysplex as the CICS system, you must use an IPIC TLS (JSSE) connection that is configured with client authentication.

If a SAF ID is used for authentication (e.g., basic authentication with a SAF registry) then the SAF ID is passed to CICS.



Flowing an identity to CICS

The zosconnect_cicsIpicConnection element is the key :

Required Configuration

- Coded character set identifier (CCSID): 37
- Connection reference: catalog

Optional Configuration

- Transaction ID:
- Transaction ID usage:

```

<zosconnect_cicsIpicConnection id="catalog"
host="wg31.washington.ibm.com"
zosConnectNetworkid="CSCVINC"
zosConnectApplid="CSCVINC"
port="1493"/>

```

CICS IPConn

```

I TCPIPS - OVERTYPE TO MODIFY
Tcpipservice(CSCVINC)
Openstatus( Open )
Port(01493)
Protocol(Ipic)
Ssltype(Ssl)
Transid(CISS)
Authenticate(Noauthentic)
Connections(00000)
Backlog( 01024 )
Maxdatalen( 000000 )
Urm( DFHISAPIP )
Privacy(Supported)
Ciphers(3538392F3233)
Host(ANY)
Ipaddress(192.168.17.201)
Hosttype(Any)
Ipresolved(192.168.17.201)
Ipfamily(Ipv4family)

```

CICS TCPIPService

```

I TCPIPS - OVERTYPE TO MODIFY
Tcpipservice(CSCVINC)
Openstatus( Open )
Port(01493)
Protocol(Ipic)
Ssltype(Ssl)
Transid(CISS)
Authenticate(Noauthentic)
Connections(00000)
Backlog( 01024 )
Maxdatalen( 000000 )
Urm( DFHISAPIP )
Privacy(Supported)
Ciphers(3538392F3233)
Host(ANY)
Ipaddress(192.168.17.201)
Hosttype(Any)
Ipresolved(192.168.17.201)
Ipfamily(Ipv4family)

```

CICS TCPIPService

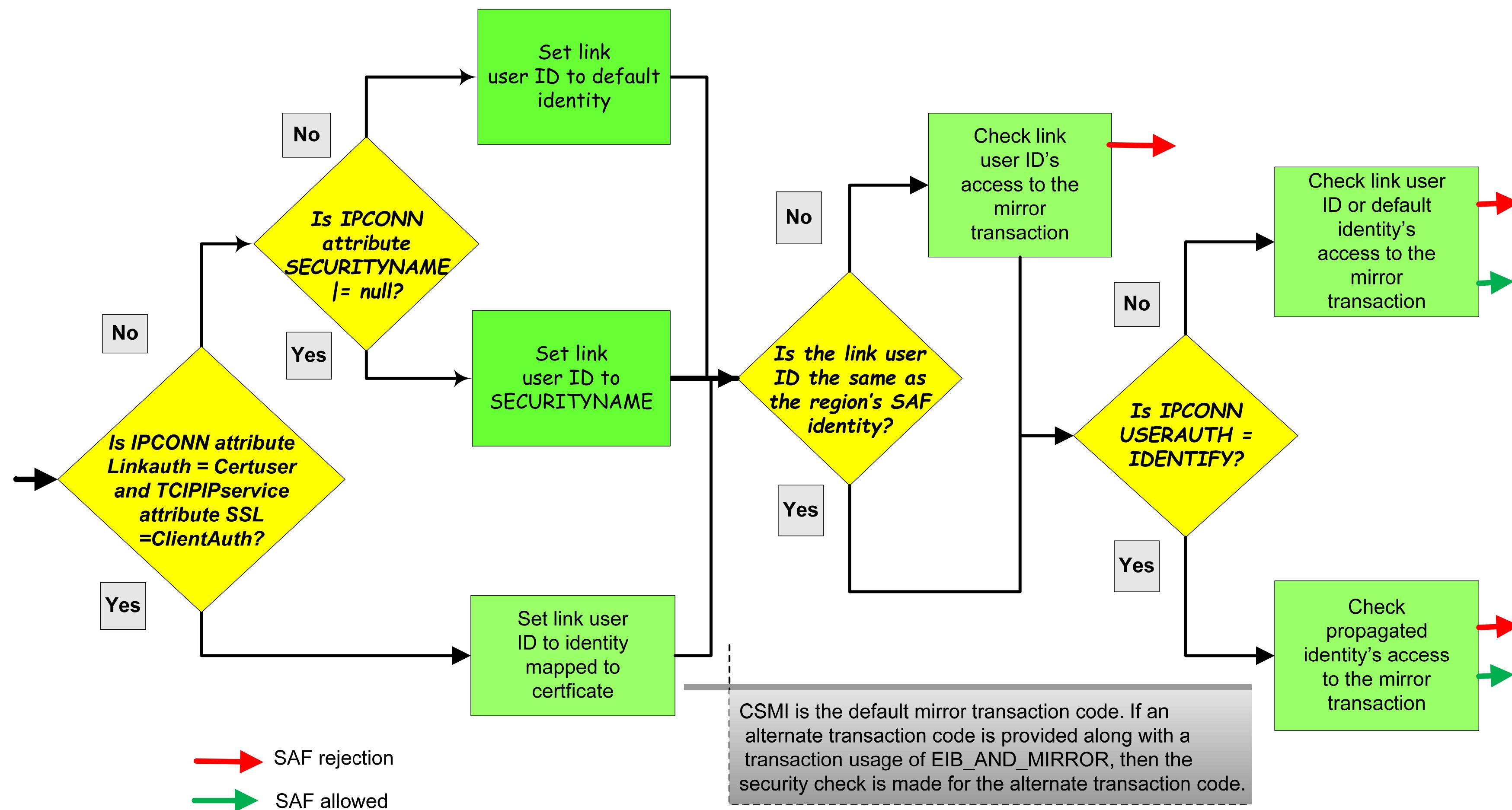
```

I IPConn - OVERTYPE TO MODIFY
Ipconn(CSCVINC)
Applid(CSCVINC)
Networkid(CSCVINC)
Servstatus( Inservice )
Connstatus( Released )
Ssltype(Nossal)
Purgeype( )
Ha(Notrequired)
Receivecount(001)
Sendcount(000)
Tcpipservice(CSCVINC)
Port()
Host()
Hosttype()
Ipresolved(0.0.0.0)
Ipfamily(Unknown)
Pendstatus( Notpending )
+ Recovstatus( Norecovdata )

```



Tech/Tip: CICS IPIC Security with USERAUTH(VERIFY)



CICS IPConn Resources



```
<zosconnect_cicsIpicConnection  
id="cscvinc"  
host="wg31.washington.ibm.com"  
zosConnectApplid="ZCAPPPL"  
zosConnectNetworkid="ZCNETID"  
port="1491"/>
```

zosConnectApplid must match APPLID
in an IPConn resource

zosConnectNetworkid must match
NETWORKID in an IPConn resource

```
DEFINE IPConn (ZOSCONN)  
GROUP (SYSPGRP)  
APPLID (ZCAPPPL)  
NETWORKID (ZCNETID)  
TCPIPSERVICE (ZOSCONN)  
LINKAUTH (SECUSER | CERTUSER)  
USERAUTH (IDENTIFY)  
IDPROP (REQUIRED | OPTIONAL)
```

LINKAUTH Determines the user identity to be used for link security. The value is either **CERTUSER** or **SECUSER**. A value of **CERTUSER** sets the link identity to the identity associated with the client certificate received from the client endpoint (TLS mutual authentication is required). A value of **SECUSER** sets the link identity to the value of the *SECURITYNAME* attribute as defined in the IPConn resource.

USERAUTH Identifies how the identity under which the attached transaction attach security will run. Since a password is not available, a value of **VERIFY** is not possible. A value of **LOCAL** means the current link identity is used. A value of **DEFAULTUSER** means the CICS default identity is used. For identity propagation purposes, the value of **USERAUTH** should be **IDENTIFY** (no password will be required) so the identity provided by the client is used for executing the attached transaction. TLS must be used if the client is in a different Sysplex.

IDPROP Determines whether the original distributed identity authenticated by the z/OS Connect server is also propagated to CICS in addition to the mapped identity used for z/OS Connect authorization checks. A value of **NOTALLOWED** does not propagate the original distributed identity. A value of **OPTIONAL** will propagate to CICS the original distributed identity, if available. A value of **REQUIRED** requires that the original distributed identity be propagated to CICS. TLS must be used if the client is in a different Sysplex.

CERTIFICATE Provides the label of the certificate connected to the CICS key ring to be used for server endpoint certificate during a TLS handshake.



Identity Propagation and CICS High Availability

Assume the service installed in a server files use the following *Connection reference* values:

- cscvinc
- catalog
- miniloan

If identity propagation is required for all connection, then the CICS IPCONN resources defined in the CICs that correspond to a `zosconnect_cicsIpicConnection` configuration elements must be dedicated to that z/OS Connect server and connection reference can not be reused.

Simplify administration by still sharing a common `cicsIpicConnection` XML configuration element by using variables and a bootstrap properties file or “variables” XML file

Server baqsvr1's bootstrap.properties

```
ipicPort=1491  
cicsHost=dvipa.washington.ibm.com  
serverPrefix=baqsvr1
```

Server baqsvr2's bootstrap.properties

```
cicsHost=dvipa.washington.ibm.com  
ipicPort=1491  
serverPrefix=baqsvr2
```

Server baqsvr3's bootstrap.properties

```
cicsHost=dvipa.washington.ibm.com  
ipicPort=1491  
serverPrefix=baqsvr3
```

ipicIDProp.xml

```
<zosconnect_cicsIpicConnection id="cscvinc"  
host="${cicsHost}"  
zosConnectNetworkid="${wlp.server.name}"  
zosConnectApplid="${wlp.server.name}"  
sharedPort="true" port="${ipicPort}"/>  
<zosconnect_cicsIpicConnection id="catalog"  
host="${cicsHost}"  
zosConnectNetworkid="${serverPrefix}C"  
zosConnectApplid="${serverPrefix}C"  
sharedPort="true" port="${ipicPort}"/>  
<zosconnect_cicsIpicConnection id="miniloan"  
host="${cicsHost}"  
zosConnectNetworkid="${serverPrefix}M"  
zosConnectApplid="${serverPrefix}M"  
sharedPort="true" port="${ipicPort}"/>
```

→ baqsvr1 or baqsvr2

→ baqsvr1C or baqsvr2C

→ baqsvr1M or baqsvr2M



Tech-Tip: CICS IPConn and TCPIPSERVICE resources for HA

CICS Specific TCPIPSERVICE - IPIC

```
TCpipservice : IPIC1
GROup       : SYSPGRP
Urm          ==> DFHISAIP
POrtnumber   ==> 01492
STatus       ==> Open
PROtocol     ==> IPic
TRansaction  ==> CISS
Host         ==> ANY
Ipaddress    ==> ANY
SPeciftcp   ==>
```

CICS Generic TCPIPSERVICE - IPICG

```
TCpipservice : IPICG1
GROup       : SYSPGRP
Urm          ==> DFHISAIP
POrtnumber   ==> 01491
STatus       ==> Open
PROtocol     ==> IPic
TRansaction  ==> CISS
Host         ==> ANY
Ipaddress    ==> ANY
SPeciftcp   ==> IPIC
```

A client connects first to the CICS region's generic port (1491) and then the CICS region redirects the client to the region's specific port (1492).

I IPConn ACQ

```
STATUS: RESULTS - OVERTYPE TO MODIFY
Ipc(BAQSVR1 ) App(BAQSVR1) Net(BAQSVR1) Ins Acq Nos
      Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR1C) App(BAQSVR1C) Net(BAQSVR1C) Ins Acq Nos
      Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR1M) App(BAQSVR1M) Net(BAQSVR1M) Ins Acq Nos
      Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR2 ) App(BAQSVR2) Net(BAQSVR2) Ins Acq Nos
      Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR2C) App(BAQSVR2C) Net(BAQSVR2C) Ins Acq Nos
      Rece(001) Sen(000) Tcp(IPIC)
Ipc(BAQSVR2M) App(BAQSVR2M) Net(BAQSVR2M) Ins Acq Nos
      Rece(001) Sen(000) Tcp(IPIC)
```

Number of
IPConn resources
equals the number
of zCEE server
times the number of
unique connection
references

¹CICS requires the specific TCPIPSERVICE be installed before the corresponding generic TCPIPSERVICE resource. TCPIPServices are installed in alphabetically order, so the name of specific service must be alphabetically prior to the name of the generic TCPIPSERVICE.



CICS IPIC connection processing for high availability load balancing*

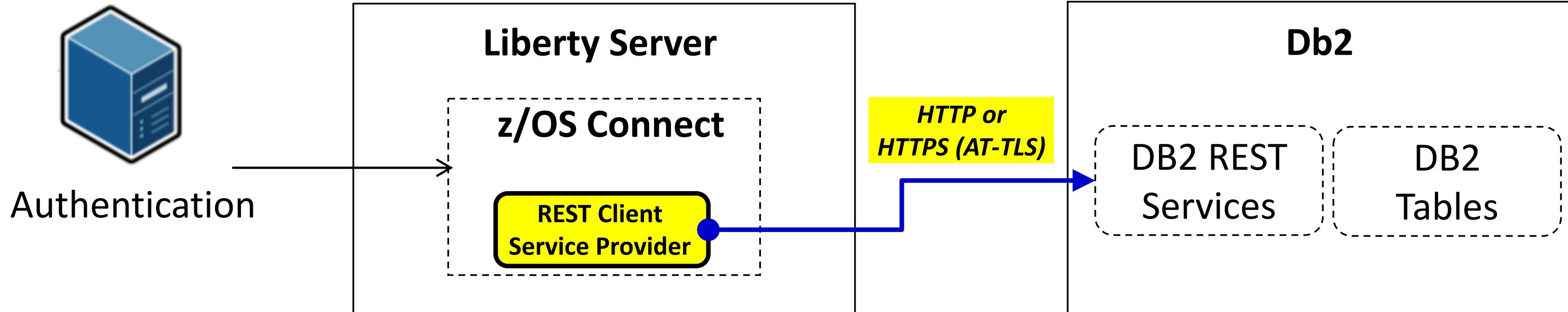
If the *reconnectInterval* attribute is set, at the specified time interval, a check is made to see if a new connection attempt should be attempted. A new connection is established if the current connection properties are not the preferred connection properties:

- If *reconnectInterval*, *preferredSpecificHost* and *preferredSpecificPort* are not set,
 - New connection attempts are disabled (this is the default behavior).
- If *reconnectInterval* is set and *preferredSpecificHost* and *preferredSpecificPort* are not set,
 - A new connection is attempted at the interval specified by the *reconnectInterval* time. Use this to enable regular connection rebalancing.
- If *reconnectInterval* and *preferredSpecificPort* are set and *preferredSpecificHost* is not set,
 - A new connection is attempted at the expiration time interval and if the current connected port in use does not match the preferred port
 - Relevant when shared port is for a single LPAR
 - Specific CICS region is preferred
- If *reconnectInterval* and *preferredSpecificHost* are set and *preferredSpecificPort* is not set
 - A new connection is attempted at the expiration time interval and if the current host in use does not match the preferred port
 - Relevant when shared port is across Sysplex
 - Any CICS region on a specific LPAR is preferred
- If *reconnectInterval*, *preferredSpecificHost* and *preferredSpecificPort* are all set
 - A new connection is attempted at the expiration time interval time and if both the current host and port in use do not match the preferred host and port
 - Relevant when shared port is on a single LPAR or across a Sysplex
 - Specific CICS region is preferred.

When the reconnection attempt results in a new connection to a CICS region, new requests are sent over the new connection. Previous connections will continue and when all requests have completed processing, the previous or old connection will be closed.



Flowing the identity for the REST client SP (Db2)



```
<zosconnect_zosConnectServiceRestClientConnection id="Db2Conn"  
host="wg31.washington.ibm.com"  
port="2446"  
basicAuthRef="dsn2Auth" />  
<zosconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth"  
userName="USER1"  
password="USER1"/>
```

Authentication options:

1. User ID / password
2. TLS Client Certificate (JSSE)
3. PassTicket support

Specify a user identity and password to be used in the HTTP header with the Db2 REST Service

```
<zosconnect_zosConnectServiceRestClientConnection id="Db2Conn"  
host="wg31.washington.ibm.com"  
port="2446"  
basicAuthRef="dsn2Auth" />  
  
<zosconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth"  
appName="DSN2APPL"/>
```

z/OS Connect requests a PassTicket from RACF

Server XML - Accessing a Db2 REST service



*selectEmployee Service

Service Project Editor: Configuration

Required Configuration

Enter the required configuration for this service.

Connection reference: db2conn

Definition Configuration

DSNL004I -DSN2 DDF START

COMPLETE

LOCATION DSN2LOC

LU

USIBMWZ.DSN2APPL

GENERICLU -NONE

DOMAIN

WG31.WASHINGTON.IBM.COM

TCPPORT 2446

SECPORT 2445

RESPORT 2447

```
<zosconnect_zosConnectServiceRestClientConnection id="db2conn"
    host="wg31.washington.ibm.com"
    port="2446"
    basicAuthRef="dsn2Auth" />

<zosconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth"
    applName="DSN2APPL"/>
```

```
<featureManager>
    <feature>zosconnect:db2-1.0</feature>
</featureManager>

<zosconnect_credential user="\${DB2_USERNAME}"
    password="\${DB2_PASSWORD}" id="commonCredentials" />

<zosconnect_db2Connection id="db2Conn" host="\${DB2_HOST}"
    port="\${DB2_PORT}" credentialRef="commonCredentials" />
```



PassTickets and Db2

- ☐ Basic authentication Db2 using a PassTicket depends on the Db2 configuration.

```
DSNL080I -DSN2 DSNLTDDF DISPLAY DDF REPORT FOLLOWS:  
DSNL081I STATUS=STARTD  
DSNL082I LOCATION LUNAME GENERICCLU  
DSNL083I DSN2LOC USIBMWZ.DSN2APPL USIBMWZ.DSN0APPL  
DSNL084I TCPPORT=2446 SECPORT=2445 RESPOR=2447 IPNAME=-NONE  
DSNL085I IPADDR=:192.168.17.201  
DSNL086I SQL DOMAIN=WG31.WASHINGTON.IBM.COM  
DSNL105I CURRENT DDF OPTIONS ARE:  
DSNL106I PKGREL = COMMIT  
DSNL106I SESSIDLE = 001440  
DSNL099I DSNLTDDF DISPLAY DDF REPORT COMPLETE
```

```
DSNL080I -DSNC DSNLTDDF DISPLAY DDF REPORT FOLLOWS:  
DSNL081I STATUS=STARTD  
DSNL082I LOCATION LUNAME GENERICCLU  
DSNL083I DSN2LOC -NONE -NONE  
DSNL084I TCPPORT=2446 SECPORT=2445 RESPOR=2447 IPNAME=DB2IPNM  
DSNL085I IPADDR=:192.168.17.252  
DSNL086I SQL DOMAIN=WG31.WASHINGTON.IBM.COM  
DSNL086I RESYNC DOMAIN=WG31.WASHINGTON.IBM.COM  
DSNL089I MEMBER IPADDR=:192.168.17.252  
DSNL105I CURRENT DDF OPTIONS ARE:  
DSNL106I PKGREL = COMMIT  
DSNL106I SESSIDLE = 001440  
DSNL099I DSNLTDDF DISPLAY DDF REPORT COMPLETE
```

```
RDEFINE PTKTDATA DSN2APPL SSIGNON(0123456789ABCDEF)  
APPLDATA('NO REPLAY PROTECTION') UACC(NONE)
```

```
RDEFINE PTKTDATA IRRPTAAUTH.DSN2APPL.* UACC(NONE)  
PERMIT IRRPTAAUTH.DSN2APPL.* ID(LIBSERV) CLASS(PTKTDATA)  
ACCESS(UPDATE)
```

Which value should be used for *applName* is determined for use in RACF resources is determined as shown below.

- ☐ If *GENERICCLU* is defined, use the second part of *GENERICCLU* for *applName*, e.g., ***DSN0APPL***
- ☐ If *GENERICCLU* is not defined, use the second part of *LUNAME* for *applName*, e.g., ***DSN2APPL***
- ☐ If neither *GENERICCLU* or *LUNAME* is defined, use the value of the *IPNAME* for *applName*, e.g., ***DB2IPNM***



Tech/Tip: Db2 REST Security

- Access to Db2 REST services requires READ access to the Db2 subsystem DSNR REST resource. i.e., permit READ access to this resource to the identity in question, for example

PERMIT DSN2.REST CLASS(DSNR) ID(USER2) ACC(READ) where DSN2 is the Db2 subsystem ID
SETROPTS RACLIST(DSNR) REFRESH

- Db2 package access is also required. If a user is not able to display a valid Db2 REST services in the z/OS Connect Db2 services development tooling or by using a **POST** to the Db2 provided REST interface URL of <http://wg31.washington.ibm.com:2446/services/DB2ServiceDiscover>, then they may not have sufficient access to the package containing the service.

For example, if service *zCEEService.selectEmployee* is defined to Db2 but not visible in the z/OS Connect tooling or if a **GET** request to URL <http://wg31.washington.ibm.com:2446/services/zCEEService/selectEmployee> fails with message:

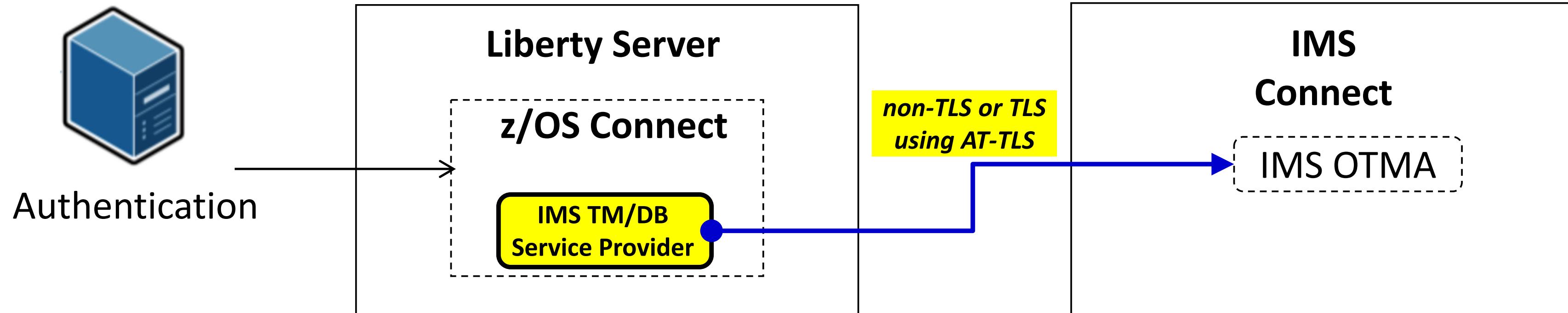
```
{  
  "StatusCode": 500,  
  "StatusDescription": "Service zCEEService.selectEmployee discovery failed due to  
  SQLCODE=-551 SQLSTATE=42501, USER2 DOES NOT HAVE THE PRIVILEGE TO PERFORM OPERATION EXECUTE  
  PACKAGE ON OBJECT zCEEService.selectEmployee. Error Location:DSNLJACC:35"  
}
```

The user needs to be granted execute authority on package *zCEEService.selectEmployee* with command:

GRANT EXECUTE ON PACKAGE "zCEEService"."selectEmployee" TO USER2 or
GRANT EXECUTE ON PACKAGE "zCEEService"."* TO USER2



Flowing an identity to IMS Connect (TM)



```
HWS=(ID=IMS15HWS,XIBAREA=100,RACF=Y,RRS=Y)  
TCPIP=(HOSTNAME=TCPIP,PORTID=(4000,LOCAL),RACFID=JOHNSON,TIMEOUT=5000)  
DATASTORE=(GROUP=OTMAGRP,ID=IVP1, MEMBER=HWSMEM, DRU=HWSYDRU0,  
TMEMBER=OTMAMEM, APPL=IMSTMAPL)
```

Authentication options:
1. User ID / password
2. PassTicket support

```
<connectionFactory containerAuthDataRef="Connection1_Auth" id="IVP1">  
<properties.gmoa hostName="wg31.washington.ibm.com" portNumber="4000"/>  
</connectionFactory>  
<authData id="Connection1_Auth" user="USER1" password="{xor}GhIPExAGDwg="/>
```

Specify a user identity and password to be used in the request to IMS Connect

```
<connectionFactory containerAuthDataRef="Connection1_Auth" id="IVP1">  
<properties.gmoa hostName="wg31.washington.ibm.com" portNumber="4000  
applicationName="IMSTMAPL"/>  
</connectionFactory>
```

Request a PassTicket
And use it in the request to IMS Connect

PassTickets and IMS



- Basic authentication to IMS Connect using a PassTicket depends on the APPL parameters configured in IMS Connect.

```
HWS= (ID=IMS15HWS, XIBAREA=100, RACF=Y, RRS=Y)
TCPIP= (HOSTNAME=TCPIP, PORTID=(4000, LOCAL), RACFID=JOHNSON, TIMEOUT=5000)
DATASTORE= (GROUP=OTMAGRP, ID=IVP1, MEMBER=HWSMEM, DRU=HWSYDROU0,
TMEMBER=OTMAMEM, APPL=IMSTMAPL)
ODACCESS= (ODBMAUTOCONN=Y, IMSPLEX=(MEMBER=IMS15HWS, TMEMBER=PLEX1),
DRDAPORT=(ID=5555, PORTMOT=6000), ODBMTMOT=6000, APPL=IMSDBAPL)
```

RDEFINE PTKTDATA IMSTMAPL SSIGNON(0123456789ABCDEF) APPLDATA('NO REPLAY PROTECTION') UACC(NONE)

RDEFINE PTKTDATA IRRPTAUTH.IMSTMAPL.* UACC(NONE)

PERMIT IRRPTAUTH.IMSTMAPL.* ID(LIBSERV) CLASS(PTKTDATA) ACCESS(UPDATE)

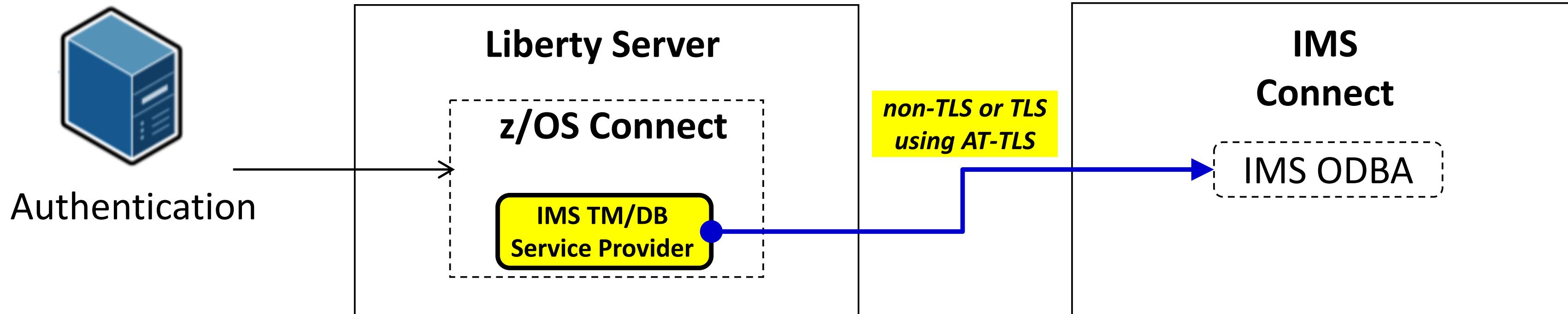
RDEFINE PTKTDATA IMSDBAPL SSIGNON(0123456789ABCDEF) APPLDATA('NO REPLAY PROTECTION') UACC(NONE)

RDEFINE PTKTDATA IRRPTAUTH.IMSDBAPL.* UACC(NONE)

PERMIT IRRPTAUTH.IMSDBAPL.* ID(LIBSERV) CLASS(PTKTDATA) ACCESS(UPDATE)



Flowing an identity to IMS Connect (DB)



```
HWS=(ID=IMS15HWS,XIBAREA=100,RACF=Y,RRS=Y)  
TCPIP=(HOSTNAME=TCPIP,PORTID=(4000,LOCAL),RACFID=JOHNSON,TIMEOUT=5000)  
ODACCESS=(ODBMAUTOCONN=Y,IMSPLEX=(MEMBER=IMS15HWS,TMEMBER=PLEX1),  
DRDAPORT=(ID=5555,PORTTMOT=6000),ODBMTMOT=6000,APPL=IMSDBAPL)
```

Authentication options:

1. User ID / password
2. PassTicket support

```
<connectionFactory id="DFSIVPACConn"> <properties.imsudbJLocal  
databaseName="DFSIVPA" datastoreName="IVP1" portNumber="5555"  
driverType="4" datastoreServer="wg31.washington.ibm.com" flattenTables="True"  
user="USER1 " password="USER1" />  
</connectionFactory>
```

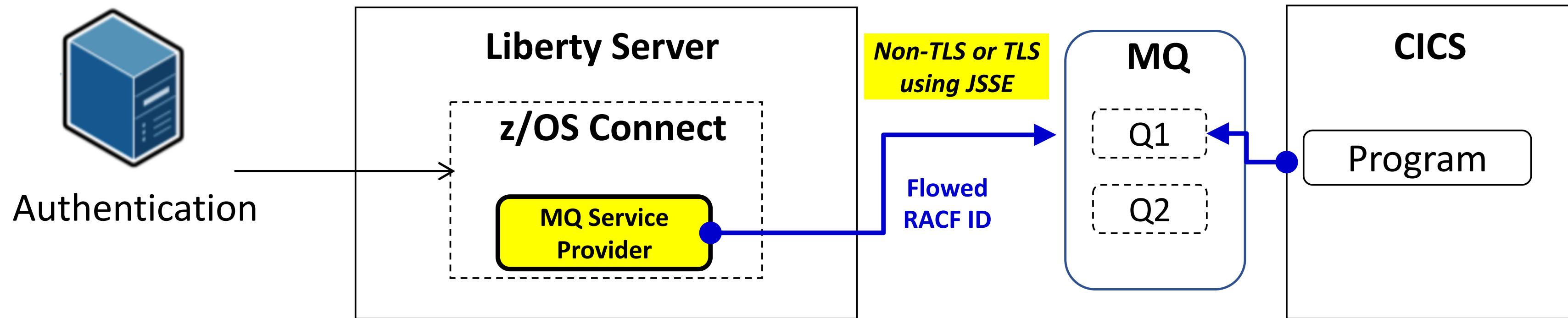
Specify a user identity and password to be used in the request to IMS Connect

```
<connectionFactory id="DFSIVPACConn"> <properties.imsudbJLocal  
databaseName="DFSIVPA" datastoreName="IVP1" portNumber="5555"  
datastoreServer="wg31.washington.ibm.com" driverType="4" flattenTables="True"  
applicationName="IMSDBAPL" "/>  
</connectionFactory>
```

Request a PassTicket
And use it in the request to IMS Connect



Flowing a user ID with MQ service provider



Set `useCallerPrincipal=true` to flow the authenticated RACF user ID, note that this is set at the service.

```
<zosconnect_services>
  <service name="mqPut">
    <property name="destination" value="jms/default"/>
    <property name="useCallerPrincipal" value="true"/>
  </service>
</zosconnect_services>
```

Define identity propagation to MQ



API Policies

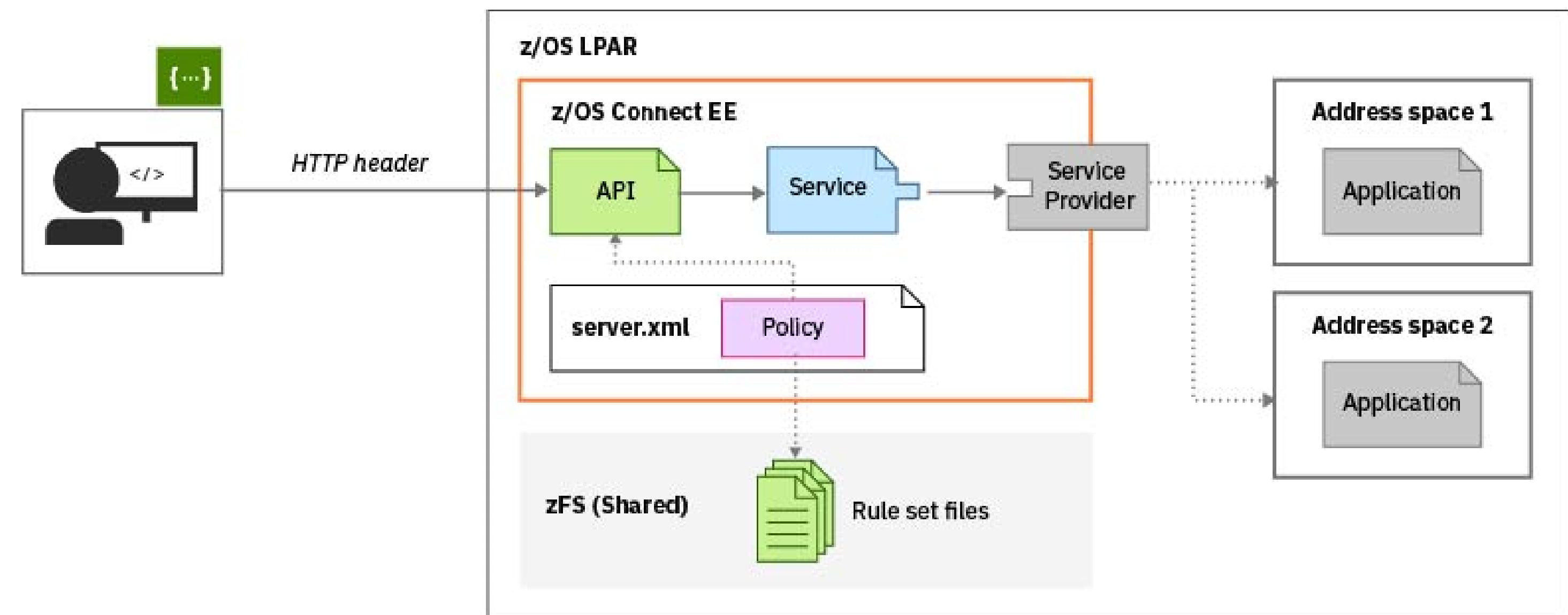
- HTTP header properties can be used to select alternative for IMS (V3.0.4) , CICS (V3.0.10), Db2 (V3.0.36) or MQ (V3.0.39)
- Policies can be configured globally for every API in the server or for individual APIs (V3.0.11)

CICS attributes
• cicsCcsid
• cicsConnectionRef
• cicsTransId

IMS attributes
• imsConnectionRef
• imsInteractionRef
• imsInteractionTimeout
• imsLtermOverrideName
• imsTranCode
• imsTranExpiration

Db2 attributes
• db2ConnectionRef
• db2CollectionID

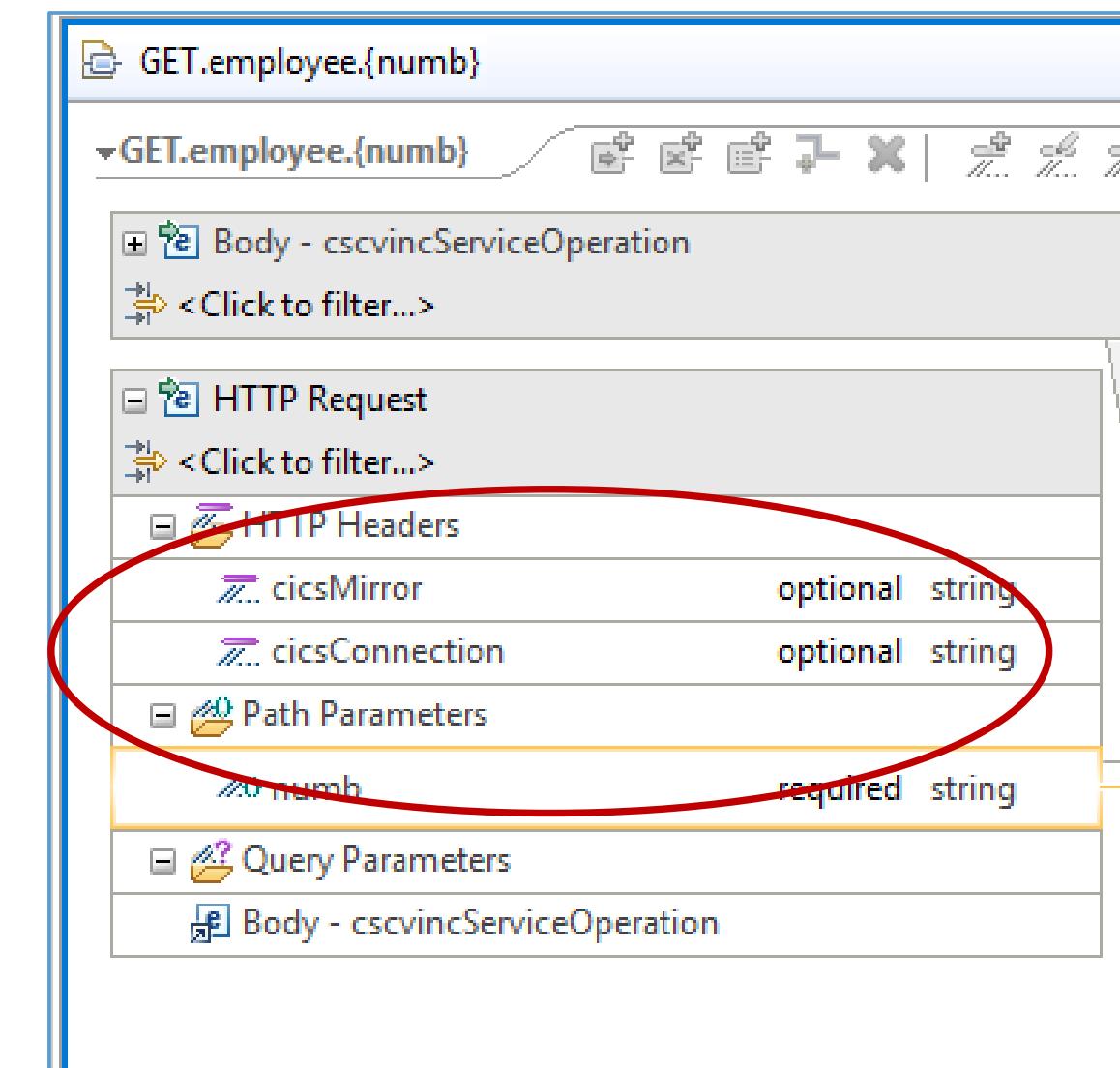
MQ attributes
• mqConnectionFactory
• mqDestination
• mqReplyDestination





A sample API Policies for CICS

```
<ruleset name="CICS rules">
  <rule name="csmi-rule">
    <conditions>
      <header name="cicsMirror" match="ANY_VALUE"/> *
    </conditions>
    <actions>
      <set property="cicsTransId" value="${cicsMirror}"/>
    </actions>
  </rule>
  <rule name="connection-rule">
    <conditions>
      <header name="cicsConnection" value="" match="ANY_VALUE"/>
    </conditions>
    <actions>
      <set property="cicsConnectionRef" value="${cicsConnection}" />
    </actions>
  </rule>
</ruleset>
```



Curl

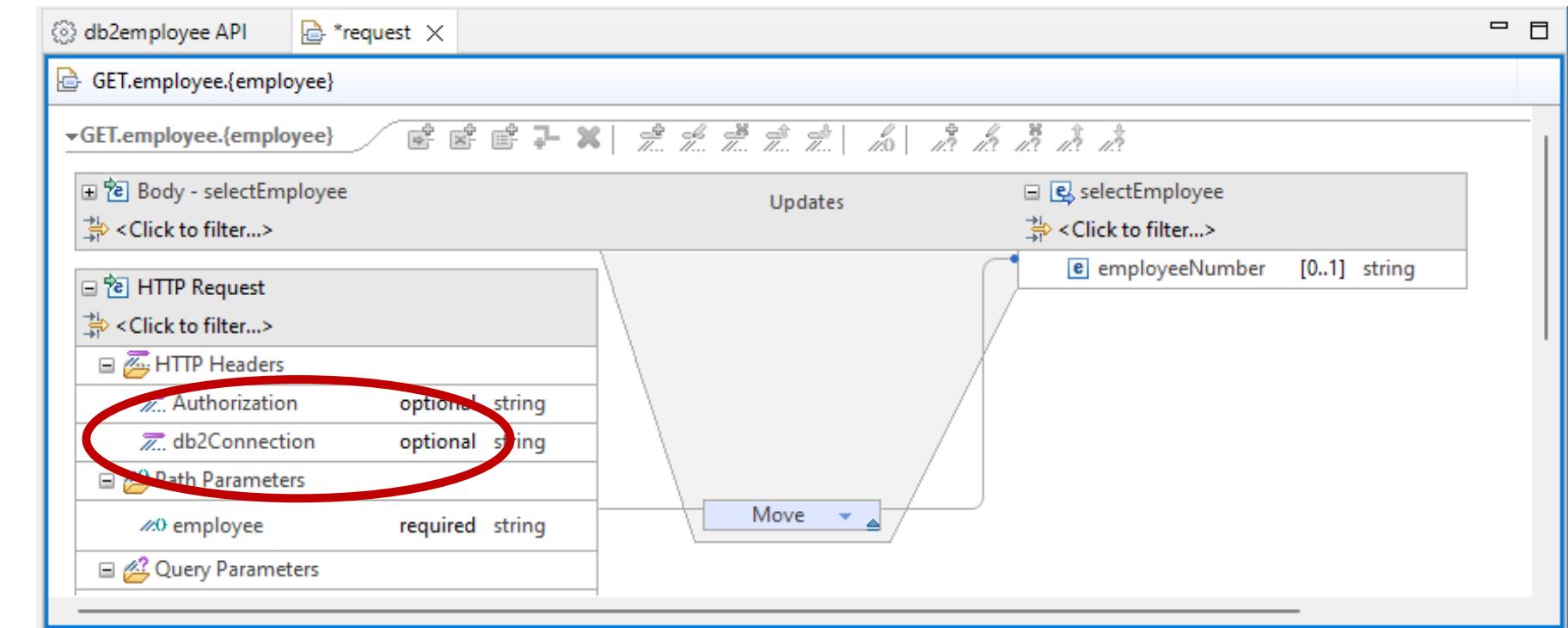
```
curl -X GET --header 'Accept: application/json' --header 'cicsMirror: MIJO' --header 'cicsConnection: cscvinc' 'https://m...
```

*Transaction MIJO needs to be a clone of CSMI (e.g., invoke program DFHMIRS)



A sample API Policies for Db2

```
<ruleset name="Db2 rules">
  <rule name="connection-rule">
    <conditions>
      <header name="db2Connection" value="" match="ANY_VALUE"/>
    </conditions>
    <actions>
      <set property="db2ConnectionRef" value="${db2Connection}"/>
    </actions>
  </rule>
</ruleset>
```



```
<zosconnect_zosConnectServiceRestClientConnection id="Db2Conn"
  sslCertsRef="db2SSLSettings" host="wg31.washington.ibm.com" port="2445" basicAuthRef="dsn2Auth" />
<zosconnect_zosConnectServiceRestClientBasicAuth id="dsn2Auth" applName="DSN2APPL"/>
<ssl id="db2SSLSettings" keyStoreRef="Db2KeyStore" trustStoreRef="Db2KeyStore"/>

<zosconnect_zosConnectServiceRestClientConnection id="fred"
  sslCertsRef="fredSSLSettings" host="wg31.washington.ibm.com" port="2445" />
<ssl id="fredSSLSettings" keyStoreRef="Db2KeyStore" trustStoreRef="Db2KeyStore" clientKeyAlias="FRED"/>

<zosconnect_zosConnectServiceRestClientConnection id="user1"
  sslCertsRef="user1SSLSettings" host="wg31.washington.ibm.com" port="2445" />
<ssl id="user1SSLSettings" keyStoreRef="Db2KeyStore" trustStoreRef="Db2KeyStore" clientKeyAlias="USER1"/>

<keyStore id="Db2KeyStore" location="safkeyring:///zCEE.Db2.KeyRing"
  password="password" type="JCERACFKS" fileBased="false" readOnly="true" />
```

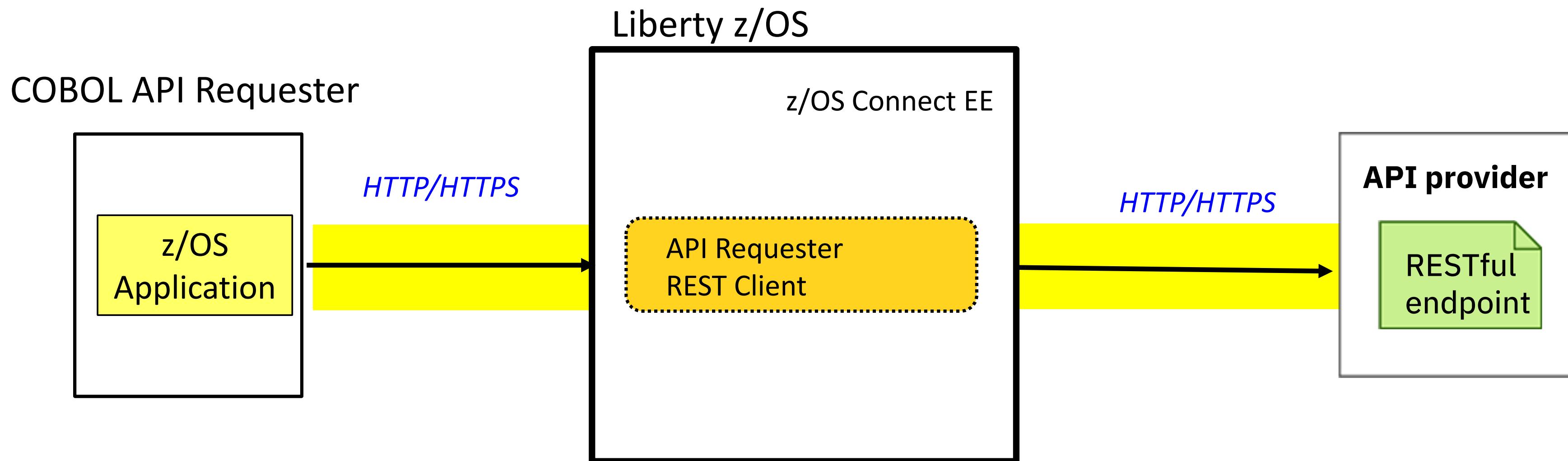
Requires APAR PH53162

Security when accessing non-z/OS systems

z/OS Connect API Requester Security



End to end API requester to API Provider connection overview



MVS Batch, IMS HTTP and Db2 stored procedure connection details provided by:

- Environment Variables (BAQURI, BAQPORT)
 - Via JCL
 - LE Options (CEEROPTS)
 - Programmatically (CEEENV)
- HTTP or HTTPS

CICS HTTP connection details provided by:

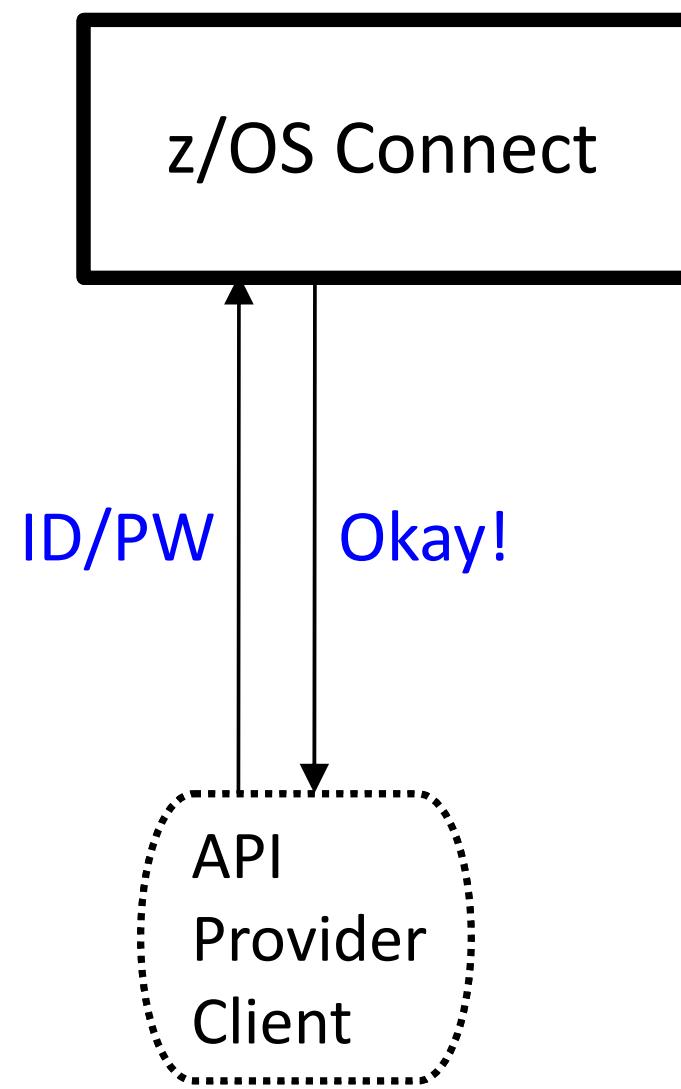
- CICS URIMAP resource (default BAQURIMP)
 - HOST
 - PORT
 - SCHEME (HTTP/HTTPS)



API Requester – Security from the application to the z/OS Connect server

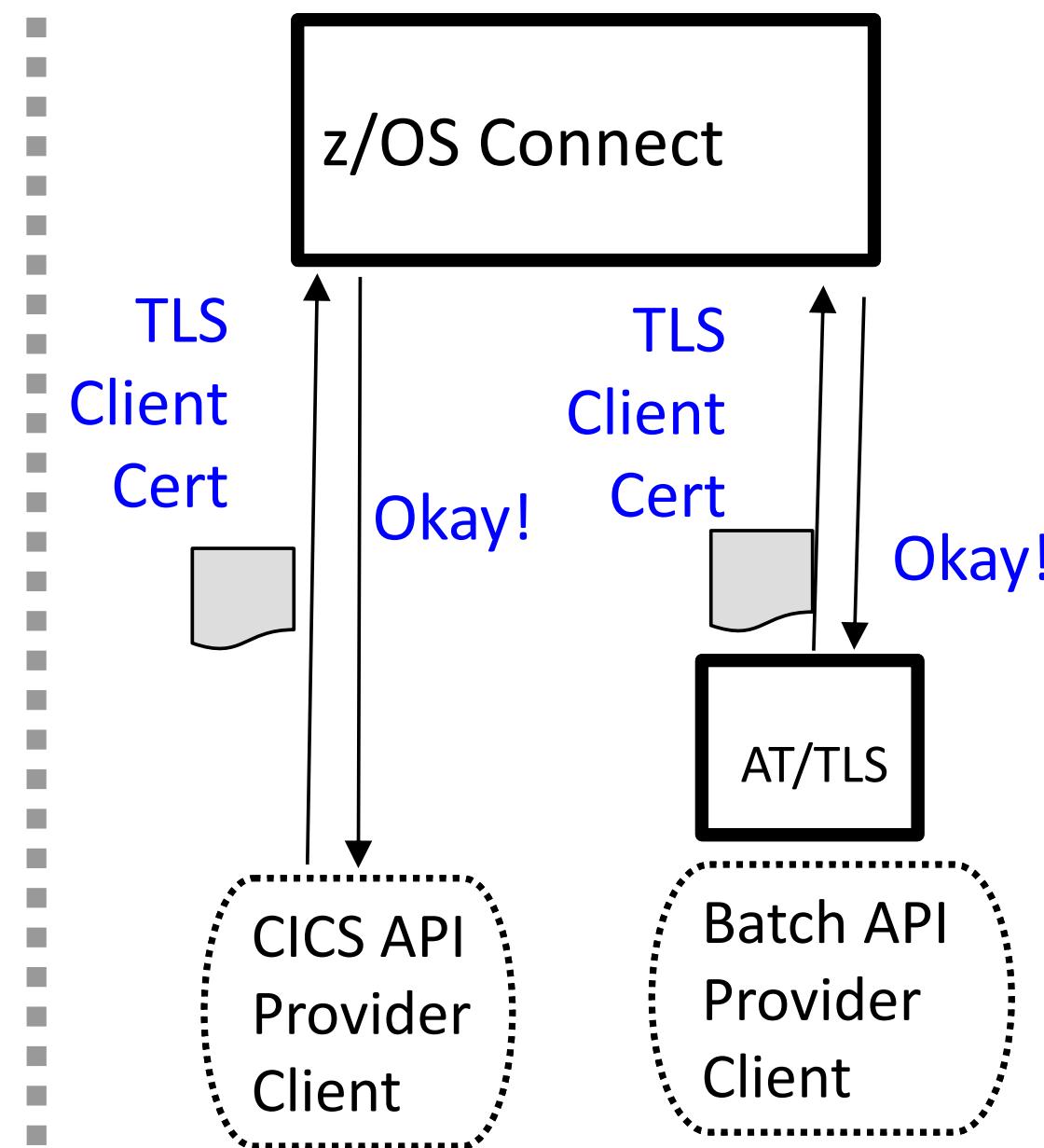
Two options for providing credentials for authentication

Basic Authentication



Application provides
ID/PW or ID/PassTicket

Client Certificate



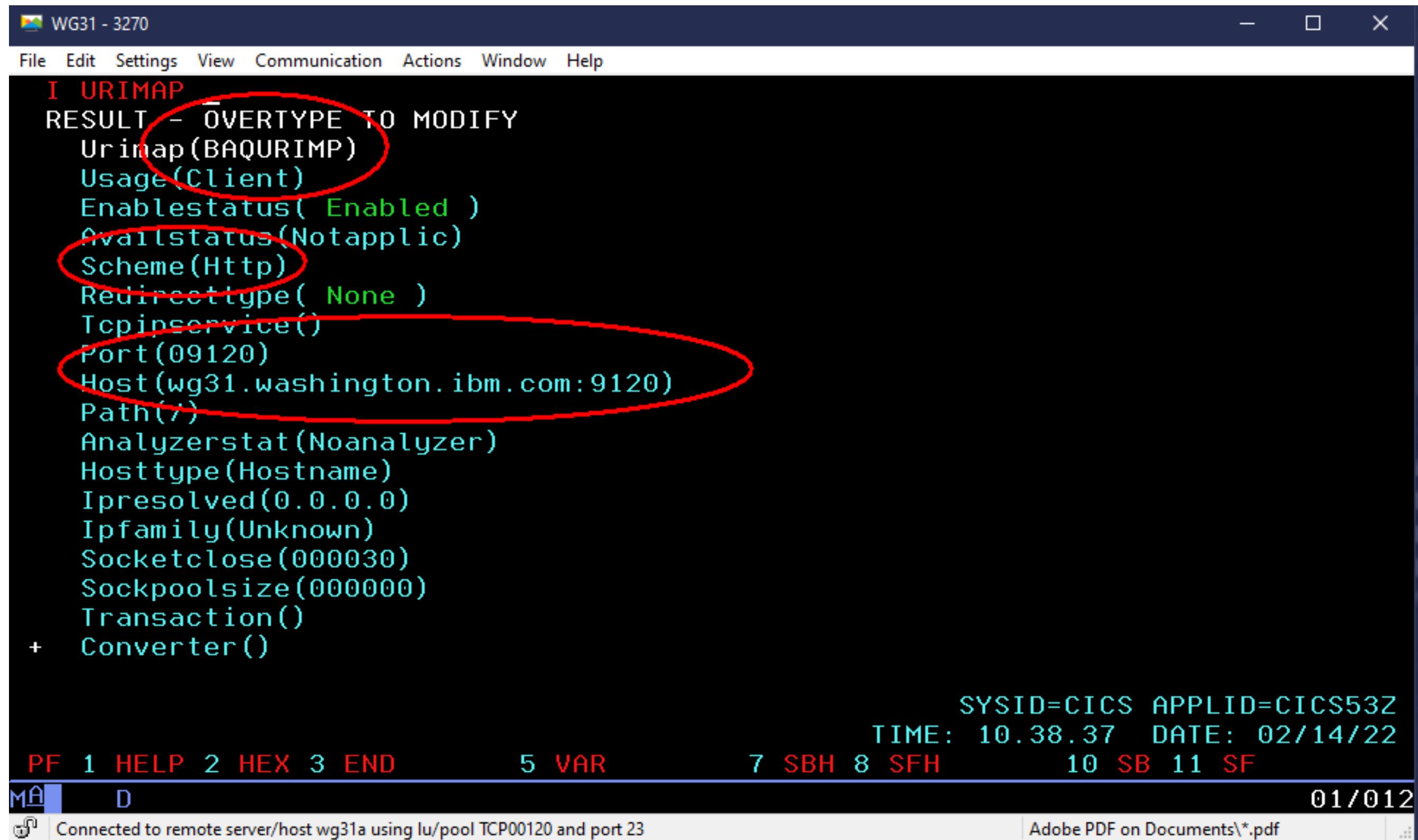
z/OS Connect requests a client certificate

CICS or AT/TLS supplies a client certificate



Configuring connections to the z/OS API requester server

Default CICS URI MAP*



```
WG31 - 3270
File Edit Settings View Communication Actions Window Help
I URIMAP
RESULT - OVERTYPE TO MODIFY
Urimap(BAQURIMP)
Usage(Client)
Enablestatus( Enabled )
Availstatus(Notapplic)
Scheme(Http)
Redirecttype( None )
Tcpinservice()
Port(09120)
Host(wg31.washington.ibm.com:9120)
Path(/)
Analyzerstat(Noanalyzer)
Hosttype(Hostname)
Ipresolved(0.0.0.0)
Ipfamily(Unknown)
Socketclose(000030)
Sockpoolsize(000000)
Transaction()
+ Converter()

SYSID=CICS APPLID=CICS53Z
TIME: 10.38.37 DATE: 02/14/22
PF 1 HELP 2 HEX 3 END      5 VAR      7 SBH 8 SFH      10 SB 11 SF
M A D
Connected to remote server/host wg31a using lu/pool TCP00120 and port 23
Adobe PDF on Documents\*.pdf
```

LE Environment Variables

```
//DELTAPI EXEC PGM=DELTAPI,PARM='323232'
//STEPLIB DD DISP=SHR,DSN=USER1.ZCEE.LOADLIB
//          DD DISP=SHR,DSN=ZCEE30.SBAQLIB
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//CEEOPTS DD *
POSIX(ON),
ENVAR("BAQURI=wg31.washington.ibm.com",
"BAQPORT=9120")
```

* V3.0.37 added support for a CICS application to specify or request a specific URIMAP resource the using BAQ-ZCON-SERVER-URI variable in BAQRINFO



A COBOL API Requester using basic authentication

- A MVS batch or IMS requester application sends basic authentication information (identity and password) by using environment variables.
 - BAQUSERNAME
 - BAQPASSWORD

- The environment variables can be provided in JCL using CEEOPTS DD statement:

```
//CEELOPTS DD *  
  POSIX(ON),  
  ENVAR("BAQURI=wg31.washington.ibm.com",  
 "BAQPORT=9080",  
 "BAQUSERNAME=USER1",  
 "BAQPASSWORD=USER1")
```

Note that the z/OS Connect communications stub generates the Authentication header token we saw earlier

- Or, provided by using a CEEROPT or CEEUOPT module:

```
CEEROPT CSECT  
CEEROPT AMODE ANY  
CEEROPT RMODE ANY  
CEELOPT POSIX=((ON),OVR),  
      ENVAR=(('BAQURI=wg31.washington.ibm.com',  
      'BAQPORT=9120',  
      'BAQUSERNAME=USER1',  
      'BAQPASSWORD=USER1'),OVR),  
      RPTOPTS=((ON),OVR)  
END
```

Tech/Tip: This is good opportunity to use a pass ticket rather than a password



Environment variables for non-CICS clients

Use these runtime environment variables when connecting to a z/OS Connect server

BAQPASSWORD - Specifies the password, in clear text, for the specified BAQUSERNAME to be authenticated with the z/OS Connect server. The username and password that are used for basic authentication, when SSL mutual authentication is not enabled.

BAQPORT - Specifies the port number for the z/OS Connect server.

BAQTIMEOUT - An optional 4-byte integer to set a timeout value in seconds for waiting for an API response. Valid range is 1 - 2,678,400 seconds. The default timeout value is 10 seconds.

BAQURI - Specifies either an IPv4 or IPV6 address, or a hostname of the host where the z/OS Connect server resides.

BAQUSERNAME - Specifies the username for connections if basic authentication is used.

BAQVERBOSE - An optional value to turn on verbose messages to assist debugging of runtime and configuration issues. Valid values are **OFF**, **ON**, **ERROR**, **AUDIT** and **ALL**. See URL <https://www.ibm.com/docs/en/zos-connect/zosconnect/3.0?topic=car-configuring-other-zos-applications-access-zos-connect-api-calls> for more information.



Tech/Tip: Generating PassTickets on z/OS

- On z/OS, a COBOL user application can generate a pass tickets by calling RACF service IRRSPK00:

```
77 COMM-STUB-PGM-NAME          PIC X(8) VALUE 'BAQCSTUB'.
77 PTKT-STUB-PGM-NAME          PIC X(8) VALUE 'ATSPTKTC'.

*-----*
***** L I N K A G E   S E C T I O N *****
*-----*
LINKAGE SECTION.

*-----*
* P R O C E D U R E S
*-----*
PROCEDURE DIVISION using PARM-BUFFER.

*-----*
MAINLINE SECTION.

*-----*
* Common code
*-----*
* initialize working storage variables
INITIALIZE GET-REQUEST.
INITIALIZE GET-RESPONSE.
CALL PTKT-STUB-PGM-NAME.
```

```
JOHNSON . PASSTCKT . SOURCE (ATSPTKTC)
*-----*
* Build IRRSPK00 parameters
*-----*
MOVE 0 to ws-length
MOVE LENGTH OF identity to identity-length.
INSPECT FUNCTION REVERSE (identity)
      TALLYING ws-length FOR ALL SPACES.
SUBTRACT ws-length FROM identity-length.
MOVE 0 to ws-length
MOVE LENGTH OF applid to applid-length.
INSPECT FUNCTION REVERSE (applid)
      TALLYING ws-length FOR ALL SPACES.
SUBTRACT ws-length FROM applid-length.
MOVE 8 to passTicket-length.
MOVE 'NOTICKET' to passTicket.
MOVE X'0003' to irr-functionCode.
MOVE X'00000001' to irr-ticketOptions.
SET irr-ticketOptions-ptr to ADDRESS OF irr-ticketOptions.
*-----*
* Call RACF service IRRSPK00 to obtain a pass ticket based
*   on identity and applid
*-----*
PERFORM CALL-RACF.
IF irr-safrc NOT = zero then
  DISPLAY "SAF_return_code:      " irr-safrc
  DISPLAY "RACF_return_code:     " irr-racfrc
  DISPLAY "RACF_reason_code:    " irr-racfrsn
End-if
. .
*-----*
* Call IRRSPK00 requesting a pass ticket
*-----*
CALL-RACF.
CALL W-IRRSPK00 USING irr-workarea,
  IRR-ALET, irr-safrc,
  IRR-ALET, irr-racfrc,
  IRR-ALET, irr-racfrsn,
  IRR-ALET, irr-functionCode,
  irr-optionWord,
  IRR-PASSTICKET,
  irr-ticketOptions-ptr,
  IRR-IDENTITY,
  IRR-APPLID
```

Tech/Tip: API Requester - HTTP v HTTPS



MVS Batch and IMS with and without an outbound AT-TLS policy

```
CEE0PTS DD *
  POSIX(ON),
  ENVAR("BAQURI=wg31.washington.ibm.com",
  "BAQPORT=9080")
```

```
CEE0PTS DD *
  POSIX(ON),
  ENVAR("BAQURI=wg31.washington.ibm.com",
  "BAQPORT=9443")
```

CICS URIMAPS

The image shows two CICS URIMAP configuration screens side-by-side, both titled 'WG31'. The left screen shows configuration for port 9080, while the right screen shows configuration for port 9443. Both screens display the same basic structure: a header, a 'DESCRIPTION' section, a 'UNIVERSAL RESOURCE IDENTIFIER' section, and a '+ OUTBOUND CONNECTION POOLING' section. The 'UNIVERSAL RESOURCE IDENTIFIER' section is circled in red in both screenshots, highlighting the difference between the two configurations.

WG31 (Left): Configuration for port 9080

```
OVERTYPE TO MODIFY
CEDA ALter UriMap( BAQURIMP )
  UriMap      : BAQURIMP
  Group       : SYSGRP
  Description ==> URIMAP for z/OS Connect EE server
  Status      ==> Enabled      Enabled | Disabled
  Usage       ==> Client       Server | Client | Pipeline |
                           | Jvmserver
  UNIVERSAL RESOURCE IDENTIFIER
    Scheme     ==> HTTP        HTTP | HTTPS
    Port       ==> 09120       No | 1-65535
    Host       ==> wg31.washington.ibm.com
    Path       ==> /
    (Mixed Case) ==>
    ==>
    ==>
    ==>
+ OUTBOUND CONNECTION POOLING
SYSID=CICS APPL
```

WG31 (Right): Configuration for port 9443

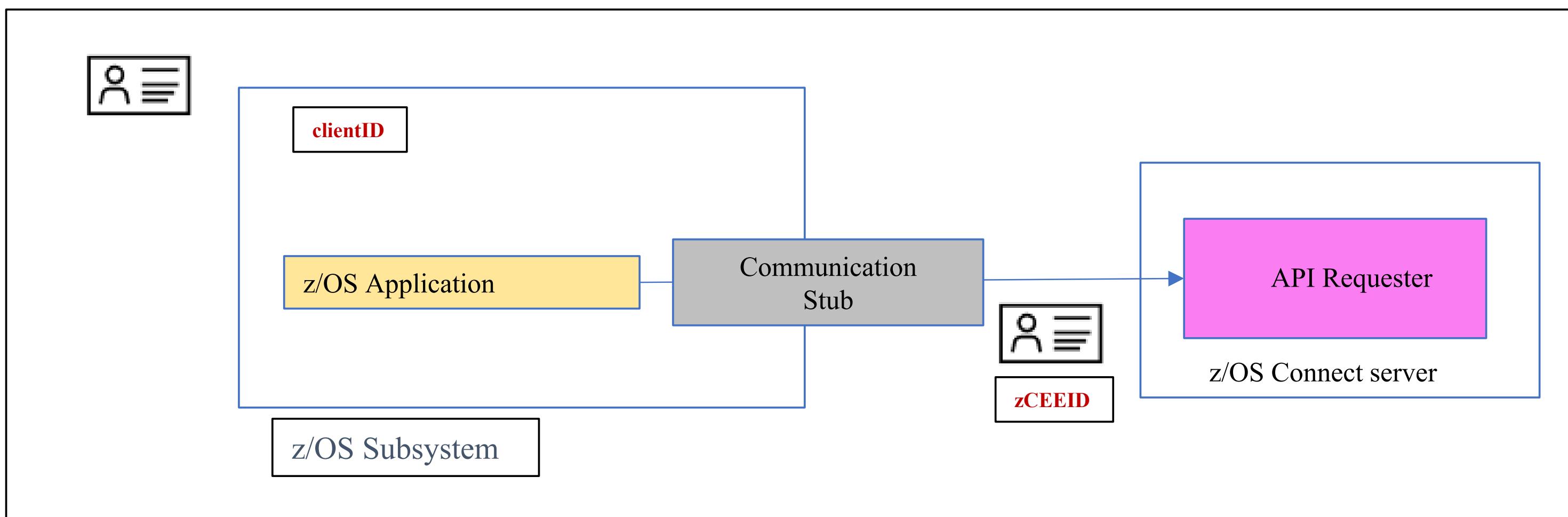
```
OVERTYPE TO MODIFY
CEDA ALter UriMap( BAQURIMP )
  UriMap      : BAQURIMP
  Group       : SYSGRP
  Description ==> URIMAP for z/OS Connect EE server
  Status      ==> Enabled      Enabled | Disabled
  Usage       ==> Client       Server | Client | Pipeline | Atom
                           | Jvmserver
  UNIVERSAL RESOURCE IDENTIFIER
    Scheme     ==> HTTPS       HTTP | HTTPS
    Port       ==> 09443       No | 1-65535
    Host       ==> wg31.washington.ibm.com
    Path       ==> /
    (Mixed Case) ==>
    ==>
    ==>
    ==>
+ OUTBOUND CONNECTION POOLING
SYSID=CICS APPLID=CICSS53Z
```

Field BAQ-ZCON-SERVER-URI was added to BAQRINFO in V3.0.37.

MOVE "URIMAP01" TO BAQ-ZCON-SERVER-URI.



API Requester - basic authentication and identity assertion



zCEEID – The identity that is used for authenticating connectivity the z/OS subsystem to the zCEE server. It is configured using basic authentication or for CICS, TLS client authentication. For MVS batch, IMS and Db2 stored procedures, the ***zCEEID*** is provided by the environment variable **BAQUSERNAME**. For CICS, the value for ***zCEEID*** is usually provided by the identity mapped to the CICS client certificate.

clientID – the identity under which the z/OS application is executing.

- For CICS, the CICS task identity
- For IMS, the transaction owner
- For batch, the job card USERID

requireAuth	idAssertion	Actions performed by z/OS Connect
true	OFF	Identity assertion is disabled. The zCEE server authenticates <i>zCEEID</i> and checks whether <i>zCEEID</i> has the authority to invoke an API requester.
	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server authenticates <i>zCEEID</i> and checks whether <i>zCEEID</i> is a surrogate of <i>clientID</i> . If <i>zCEEID</i> is a surrogate of <i>clientID</i> , the server further checks whether <i>clientID</i> has the authority to invoke an API requester; otherwise, a BAQR7114E message occurs.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server authenticates <i>zCEEID</i> and directly checks whether <i>clientID</i> has the authority to invoke an API requester.
false	OFF	Identity assertion is disabled. A BAQR0407W message occurs.
	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server checks whether <i>clientID</i> has the authority to invoke an API requester, and a warning message occurs to indicate that the ASSERT_ONLY value is used instead of the ASSERT_SURROGATE value.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server checks whether <i>clientID</i> has the authority to invoke an API requester.

```

<zosconnect_zosConnectManager
    requireAuth="true|false"
    requireSecure="true|false" />

<zosconnect_apiRequesters idAssertion="OFF">

<zosconnect_apiRequester name="cscvinc_1.0.0"
    requireAuth="true|false"
    requireSecure="true|false" />
    idAssertion="ASSERT_ONLY"> *

<zosconnect_apiRequester name="db2employee_1.0.0"
    requireAuth="true|false"
    requireSecure="true|false" />
    idAssertion="ASSERT_SURROGATE"> *

</zosconnect_apiRequesters>

```

Tech-Tip: Identity assertion requires setting a program control extended attribute

As root or superuser, set the *libifaedjreg64.so* program control extended attribute bit

- *Permit the server's identity to the required FACILITY resource*

```
PERMIT BPX.SERVER CLASS(FACILITY) ID(LIBSERV) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

- *Define a SURROGAT profile for the asserted identity and permit access to connection identity*

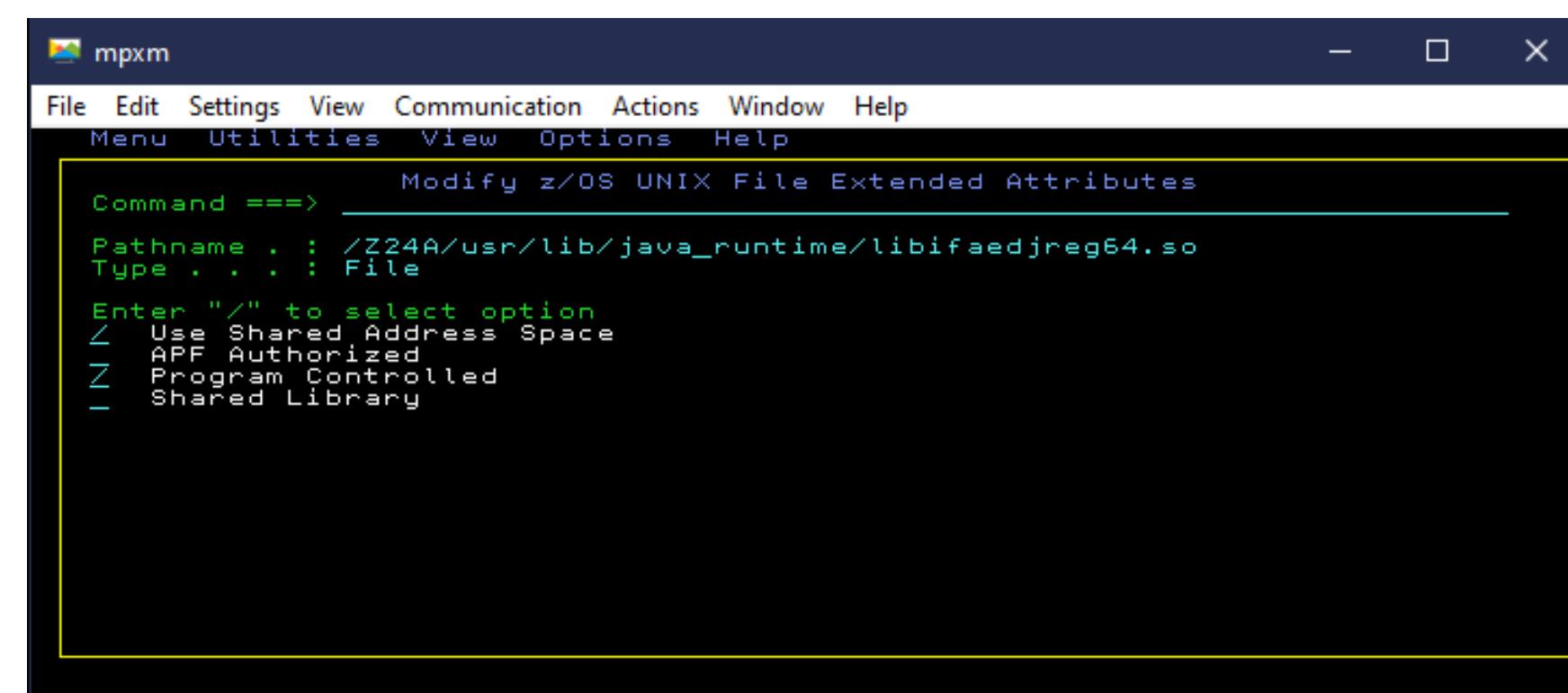
```
RDEFINE SURROGAT clientID.BAQASSRT UACC(NONE) OWNER(SYS1)
PERMIT clientID.BAQASSRT CLASS(SURROGAT) ACCESS(READ) ID(zCEEID)
```

OR

```
RDEFINE SURROGAT *.BAQASSRT UACC(NONE) OWNER(SYS1)
PERMIT *.BAQASSRT CLASS(SURROGAT) ACCESS(READ) ID(zCEEID)
SETROPTS RACLIST(SURROGAT) REFRESH
```

- *Enable the program control bit for Java shared object ifaedjreg64*

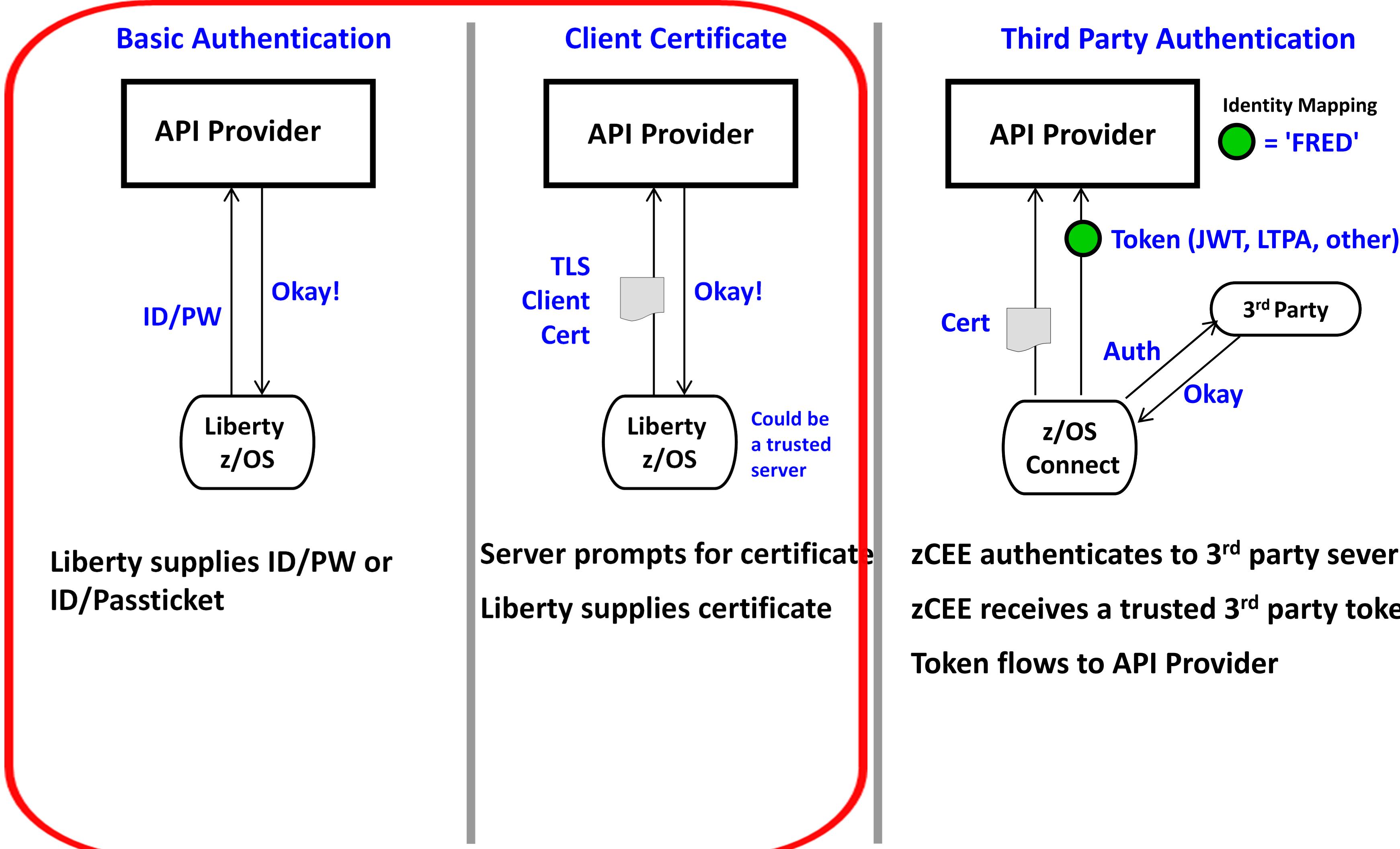
```
su          (switching to root is required)
cd /usr/lib/java_runtime
extattr +p libifaedjreg64.so
```





Accessing non-z/OS resources

Several different ways this can be accomplished:





Configuring Basic and/or TSL support

Basic authentication with HTTP protocol

```
<zosconnect_endpointConnection id="cscvincAPI"  
    host="http://wg31.washington.ibm.com" port="9080"  
    authenticationConfigRef="myAuthData" />  
  
<zosconnect_authData id="myAuthData"  
    user="zCEEclient" password="secret"/>
```

TLS with HTTPS protocol

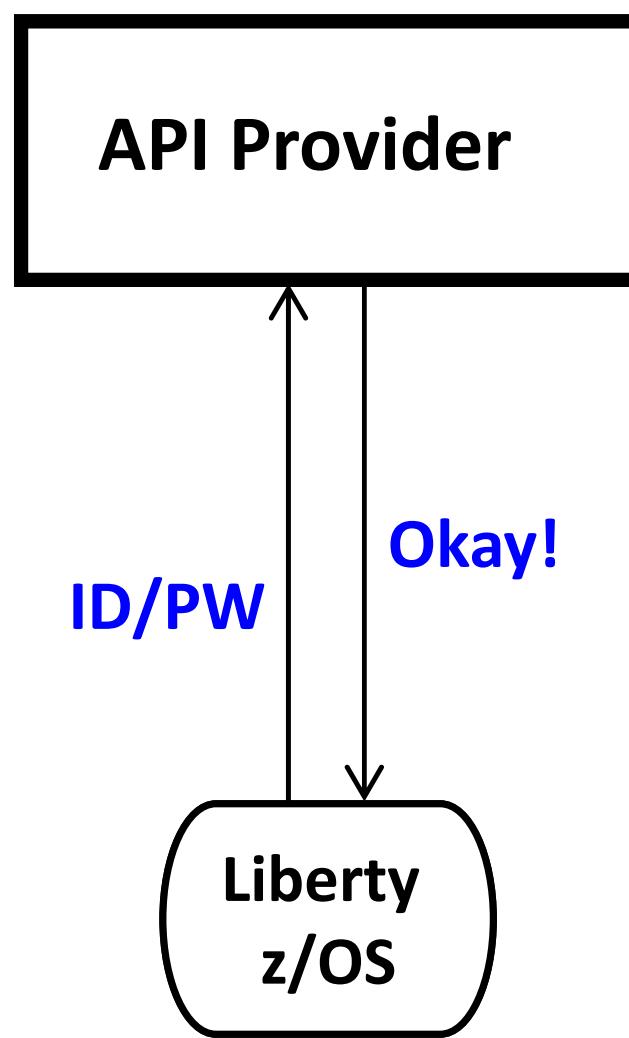
```
<zosconnect_endpointConnection id="cscvincAPI"  
    host="https://wg31.washington.ibm.com" port="9443"  
    authenticationConfigRef="myAuthData" 1  
    sslCertsRef="OutboundSSLSettings" />  
  
<zosconnect_authData id="myAuthData" 1  
    user="zCEEclient" password="secret"/>
```

¹ Optional, if mutual authentication is enabled by the server endpoint

API Requester – Security from the z/OS Connect server to the API provider

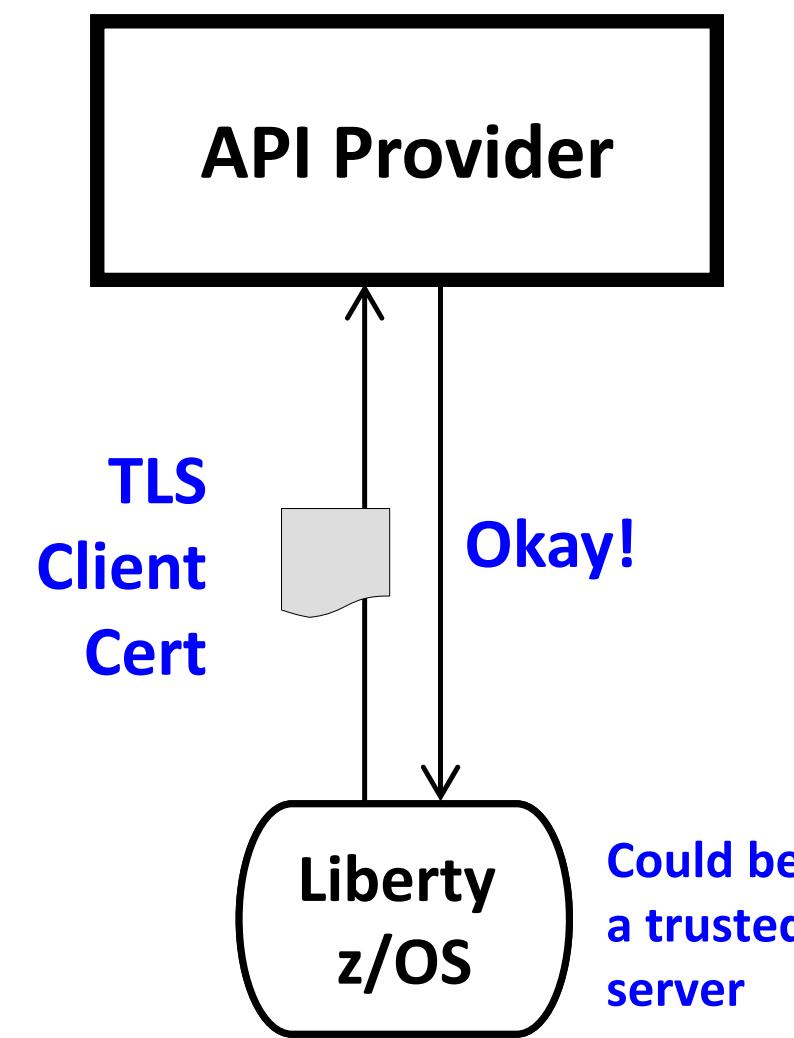
Several different ways this can be accomplished:

Basic Authentication



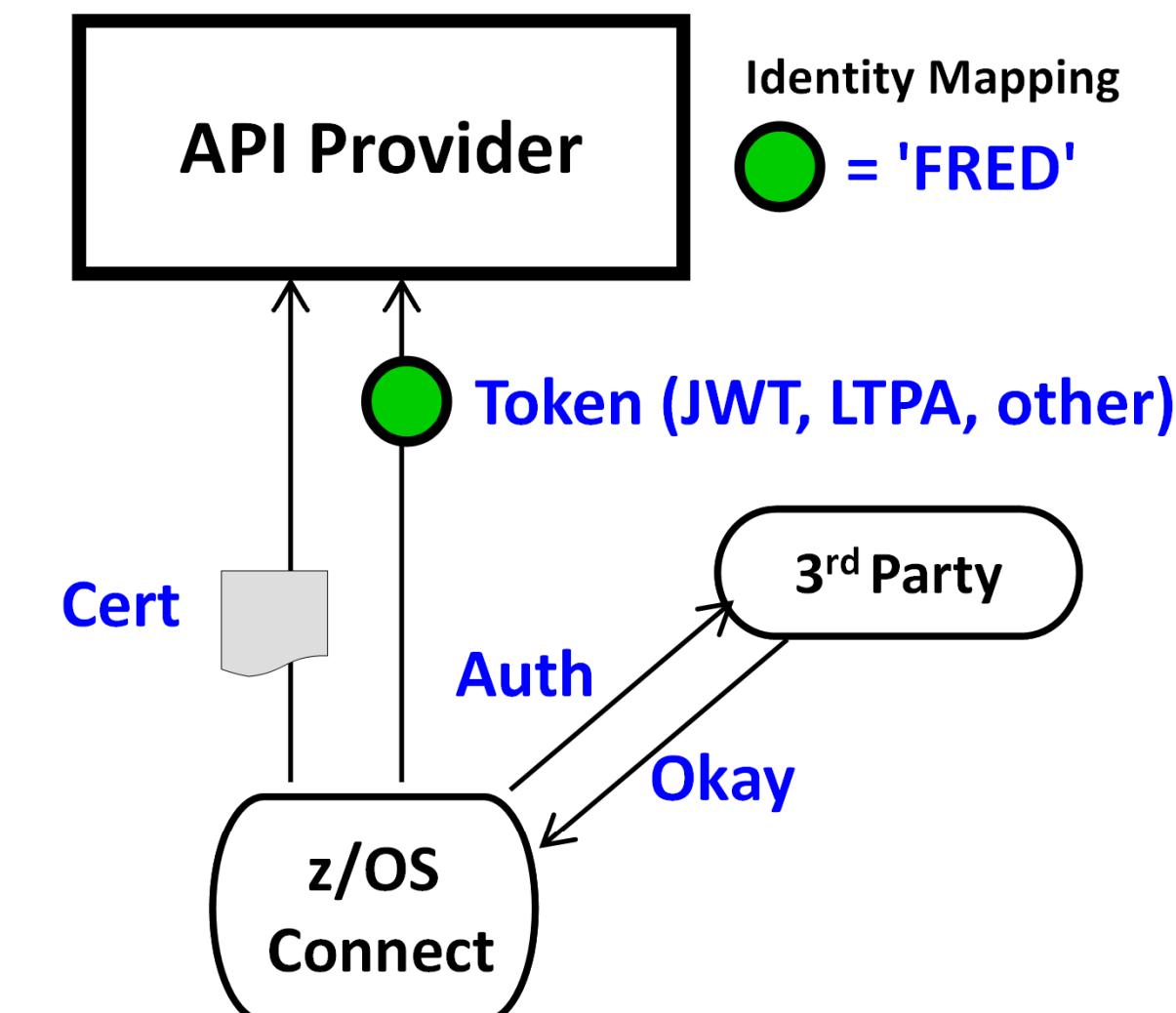
zCEE supplies ID/PW or
ID/Passticket

Client Certificate



Server prompts for certificate
zCEE supplies certificate

Third Party Authentication



zCEE authenticates to 3rd party sever
zCEE receives a trusted 3rd party token
Token flows to API Provider

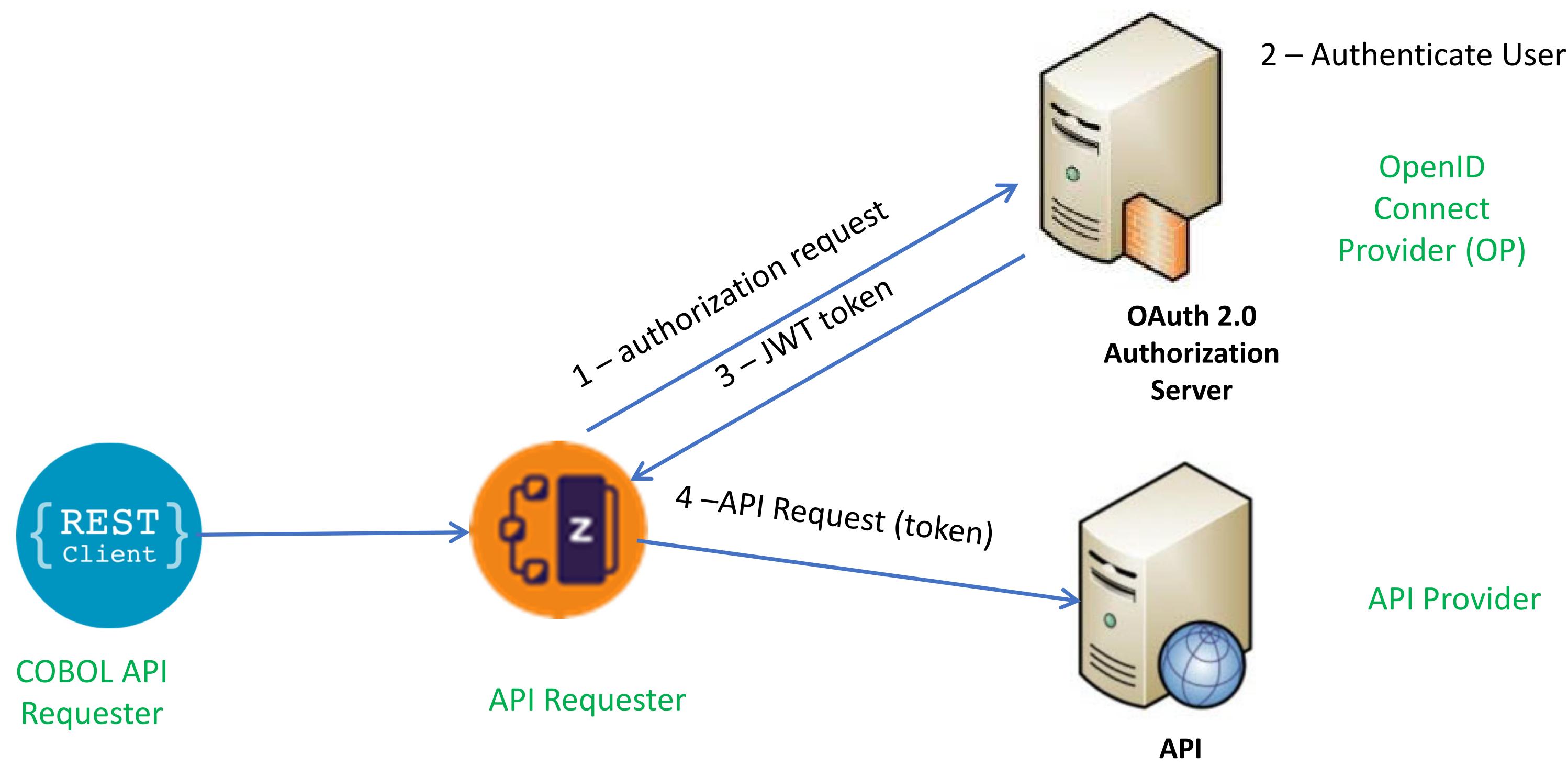
z/OS Connect API Requester - Token Support



z/OS Connect EE provides *three* ways of calling an API secured with a token

1. Use the OAuth 2.0 support when the request is part of an OAuth 2.0 flow. With OAAUTH configured, the token can be an opaque token or a JWT token.
2. In a non-OAuth 2.0 scenario, a JWT token is used in a custom flow, for example: when you need to specify the HTTP verb that is used in the request to the authentication server.
 - When you need to specify the HTTP verb that is used in the request to the authentication server
 - When you need to specify how the JWT is returned from the authentication server (for example, in an HTTP header or in a custom field in a JSON response message).
 - When you need to use a custom header name for sending the JWT to the request endpoint.
3. Use the locally generated JWT support when you need to send a JWT that is generated by the z/OS Connect EE server.

z/OS Connect OAuth Flow for API requester



Grant Types:

- client_credentials
- password



OAuth Grant Types Supported by z/OS Connect

client_credentials - the identity associated with the combination of the CICS, IMS, or z/OS application, and the z/OS Connect EE server that calls the RESTful API on behalf of the CICS, IMS, or z/OS application When this grant type is used, the z/OS Connect EE server sends the client credentials and the access scope to the authorization server.

```
<zosconnect_oAuthConfig id="myoAuthConfig"  
    grantType="client_credentials"  
    authServerRef="myoAuthServer"/>
```

password - The identity of the user of the CICS, IMS, or z/OS application, or it might be another entity. When this grant type is used, the z/OS Connect EE server sends the resource owner's credentials, the client credentials, and the access scope to the authorization server.

```
<zosconnect_oAuthConfig id="myoAuthConfig"  
    grantType="password"  
    authServerRef="myoAuthServer"/>
```

Configuring OAuth support – BAQRINFO copy book



Grant Type: *password* - The identity of the user provided by the CICS, IMS, or z/OS application, or it might be another entity. Client_credentials can be supplied by the program or in the server XML configuration.

Grant Type: *client_credentials* - the identity associated with the combination of the CICS, IMS, or z/OS application, and the z/OS Connect EE server that calls the RESTful API on behalf of the CICS, IMS, or z/OS application

Scope is always required.

OAuth 2.0 specification entity	password	client_credentials	Where Set
Client ID	required	Required	server.xml or by application
Client Secret	optional	Required	server.xml or by application
Username	required	N/A	by application
Password	required	N/A	by application



Obtaining a JWT using request parameters

The image displays two terminal windows from the z/OS environment. The left window shows the definition of a host API named ZCEE30.SBAQCOB(BAQHCONC) with various parameters for OAuth and BAQZ trace settings. The right window shows a CBL APOST script for USER1.ZCEE30.SOURCE(GETAPI) which includes authentication credentials (JWT-USER and JWT-PSWD), moves of token and password to BAQ REQ PARM fields, and a call to BAQEXEC to invoke the API endpoint.

```
BROWSE ZCEE30.SBAQCOB(BAQHCONC)
Command ===>
* Host API Request parameter names
 77 BAQR-OAUTH-USERNAME PIC X(22)
  VALUE 'BAQHAPI-oAuth-Username'.
 77 BAQR-OAUTH-PASSWORD PIC X(22)
  VALUE 'BAQHAPI-oAuth-Password'.
 77 BAQR-OAUTH-SCOPE PIC X(19)
  VALUE 'BAQHAPI-oAuth-Scope'.
 77 BAQR-OAUTH-CLIENT-ID PIC X(22)
  VALUE 'BAQHAPI-oAuth-ClientId'.
 77 BAQR-OAUTH-CLIENT-SECRET PIC X(26)
  VALUE 'BAQHAPI-oAuth-ClientSecret'.
 77 BAQR-OAUTH-RESOURCE PIC X(22)
  VALUE 'BAQHAPI-oAuth-Resource'.
 77 BAQR-OAUTH-AUDIENCE PIC X(22)
  VALUE 'BAQHAPI-oAuth-Audience'.
 77 BAQR-OAUTH-CUSTOM-PARMS PIC X(25)
  VALUE 'BAQHAPI-oAuth-CustomParms'.
 77 BAQR-JWT-USERNAME PIC X(22)
  VALUE 'BAQHAPI-Token-Username'.
 77 BAQR-JWT-PASSWORD PIC X(22)
  VALUE 'BAQHAPI-Token-Password'.

* Host API ZCON parameter names
 77 BAQZ-TRACE-VERBOSE PIC X(21)
  VALUE 'BAQHAPI-Trace-Verbose'.
 77 BAQZ-SERVER-URIMAP PIC X(21)
  VALUE 'BAQHAPI-Server-URIMAP'.
 77 BAQZ-SERVER-HOST PIC X(19)

File Edit Settings View Communication Actions Window Help
Menu Utilities Compilers Help

wg31 master
Line 0000000020 Col 001 080
Scroll ==> PAGE

EDIT          USER1.ZCEE30.SOURCE(GETAPI) - 01.02           Columns 00001 00072
File Edit Edit_Settings Menu Utilities Compilers Test Help
File Edit Edit_Settings Menu Utilities Compilers Test Help
Command ===>
***** **** Top of Data ****
000001      CBL APOST
000002
000003      * Authentication server credentials
000004      01 JWT-USER PIC X(10) VALUE 'myUsername'.
000005      01 JWT-PSWD PIC X(10) VALUE 'myPassword'.
000006
000007
000008
000009      * Send JWT credentials to z/OS Connect
000010      MOVE BAQR-TOKEN-USERNAME TO
000011          BAQ-REQ-PARM-NAME OF BAQ-REQ-PARMS(1)
000012      SET BAQ-REQ-PARM-ADDRESS OF
000013          BAQ-REQ-PARMS(1) TO ADDRESS OF JWT-USER
000014      MOVE LENGTH OF JWT-USER TO
000015          BAQ-REQ-PARM-LENGTH OF BAQ-REQ-PARMS(1)
000016      MOVE BAQR-TOKEN-PASSWORD TO
000017          BAQ-REQ-PARM-NAME OF BAQ-REQ-PARMS(2)
000018      SET BAQ-REQ-PARM-ADDRESS OF
000019          BAQ-REQ-PARMS(2) TO ADDRESS OF JWT-PSWD
000020      MOVE LENGTH OF JWT-PSWD TO
000021          BAQ-REQ-PARM-LENGTH OF BAQ-REQ-PARMS(2)
000022
000023      * Call the API endpoint using BAQEXEC
000024
000025
000026
000027

File Edit Settings View Communication Actions Window Help
File Edit Edit_Settings Menu Utilities Compilers Test Help
MA A
Connected to remote server/host wg31z using lu/pool TCP00111 and port 23
28/019
Connected to remote server/host wg31z using lu/pool TCP00111 and port 23
```

Configuring OAuth support – z/OS Connect API Requester



```
<zosconnect_endpointConnection id="cscvincAPI"
    host="http://wg31.washington.ibm.com" port="9080"
    authenticationConfigRef="myoAuthConfig"/>

<zosconnect_oAuthConfig id="myoAuthConfig"
    grantType="client_credentials|password"
    authServerRef="myoAuthServer"/>

<zosconnect_authorizationServer id="myoAuthServer"
    tokenEndpoint="https://wg31.washington.ibm.com:59443/oidc/endpoint/OP/token1
    basicAuthRef="tokenCredential" 2
    sslCertsRef="OutboundSSLSettings" />

<zosconnect_authData id="tokenCredential" 2
    user="zCEEClient" password="secret"/>
```

```
openidConnectProvider id="OP"
    signatureAlgorithm="RS256"
    keyStoreRef="jwtStore"
    oauthProviderRef="OIDCssl" >
</openidConnectProvider>
```

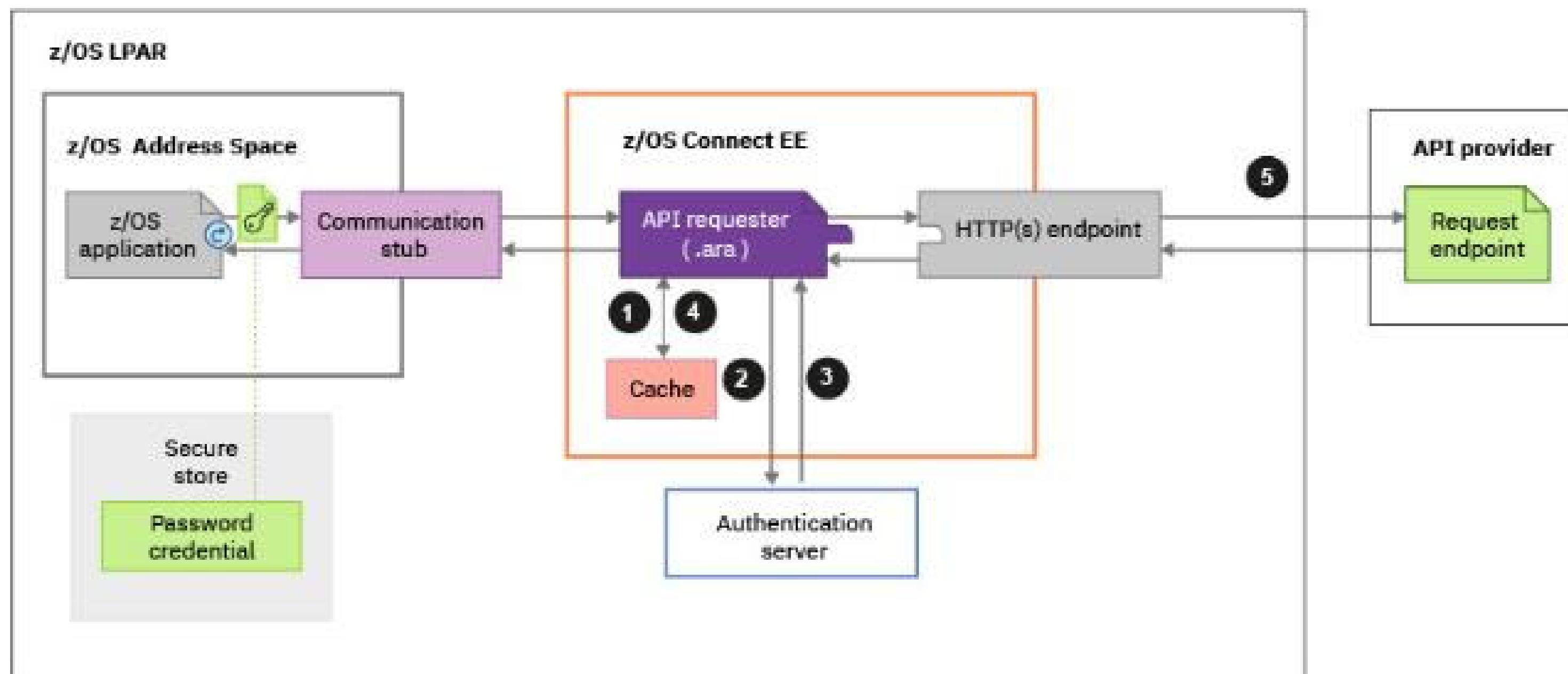
¹See URL https://www.ibm.com/support/knowledgecenter/SS7K4U_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp_oidc_token_endpoint.html

² These credentials can be specified by the application

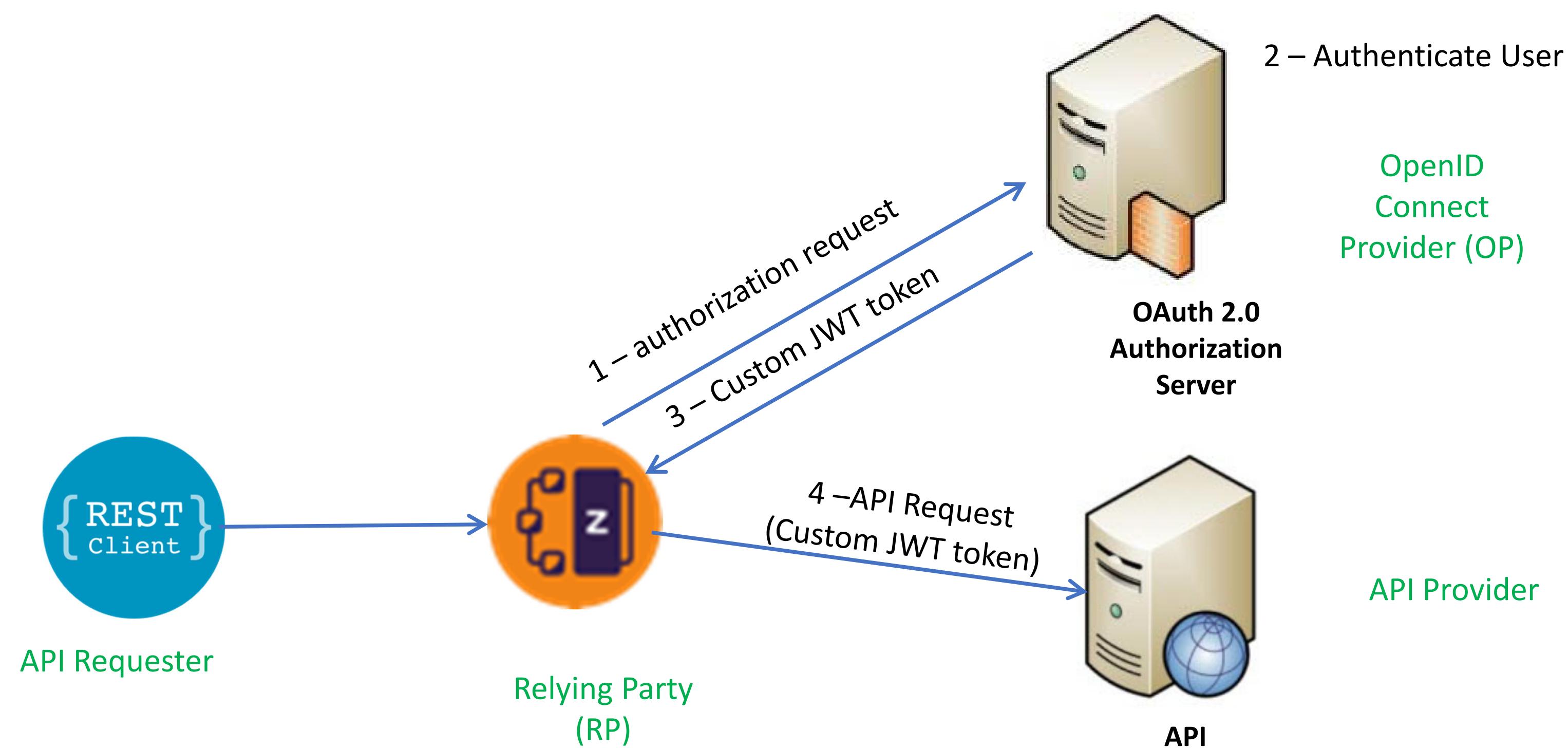


Calling an API with using a JWT custom flow

- ❑ In a non-OAuth 2.0 scenario, a JWT token is used in a custom flow, for example:
 - When you need to specify the HTTP verb that is used in the request to the authentication server.
 - When you need to specify how the JWT is returned from the authentication server (for example, in an HTTP header or in a custom field in a JSON response message).
 - When you need to use a custom header name for sending the JWT to the request endpoint.



z/OS Connect OAuth Custom Flow





API Requester – JWT Custom flow

```
wg31 master
File Edit Settings View Communication Actions Window Help
Menu Utilities Compilers Help
BROWSE ZCEE30.SBAQC0B(BAQRINFO)
Command ==> -
 01 BAQ-REQUEST-INFO.
    03 BAQ-REQUEST-INFO-COMP-LEVEL PIC S9(9) COMP-5 SYNC VALUE 4.
    03 BAQ-REQUEST-INFO-USER.
      05 BAQ-OAUTH.
        07 BAQ-OAUTH-USERNAME PIC X(256).
        07 BAQ-OAUTH-USERNAME-LEN PIC S9(9) COMP-5 SYNC
          VALUE 0.
        07 BAQ-OAUTH-PASSWORD PIC X(256).
        07 BAQ-OAUTH-PASSWORD-LEN PIC S9(9) COMP-5 SYNC
          VALUE 0.
        07 BAQ-OAUTH-CLIENTID PIC X(256).
        07 BAQ-OAUTH-CLIENTID-LEN PIC S9(9) COMP-5 SYNC
          VALUE 0.
        07 BAQ-OAUTH-CLIENT-SECRET PIC X(256).
        07 BAQ-OAUTH-CLIENT-SECRET-LEN PIC S9(9) COMP-5 SYNC
          VALUE 0.
        07 BAQ-OAUTH-SCOPE-PTR USAGE POINTER.
        07 BAQ-OAUTH-SCOPE-LEN PIC S9(9) COMP-5 SYNC
          VALUE 0.
      05 BAQ-AUTHTOKEN.
        07 BAQ-TOKEN-USERNAME PIC X(256).
        07 BAQ-TOKEN-USERNAME-LEN PIC S9(9) COMP-5 SYNC
          VALUE 0.
        07 BAQ-TOKEN-PASSWORD PIC X(256).
        07 BAQ-TOKEN-PASSWORD-LEN PIC S9(9) COMP-5 SYNC
          VALUE 0.
      05 BAQ-ZCON-SERVER-URI PIC X(256)
          VALUE SPACES.
Line 0000000028 Col 001 080
Scroll ==> PAGE
MA A
Connected to remote server/host wg31z using lu/pool TCP00145 04/015
```

BAQRINFO copy book

COBOL application

```
MOVE "ATSTOKENUSERNAME" to envVariableName.
PERFORM CALL-CEEENV THRU CALL-CEEENV-END
MOVE VAR(1:valueLength) to BAQ-TOKEN-USERNAME
MOVE valueLength TO BAQ-TOKEN-USERNAME-LEN
MOVE "ATSTOKENPASSWORD" to envVariableName.
PERFORM CALL-CEEENV THRU CALL-CEEENV-END
MOVE VAR(1:valueLength) to BAQ-TOKEN-PASSWORD
MOVE valueLength to BAQ-TOKEN-PASSWORD-LEN
```

Note that this example is using environment variables to provide token credentials, as documented in the z/OS Connect Advanced Topics Guide.



API Requester – JWT Custom flow

WG31 - 3270

```
Menu Utilities Compilers Help
BROWSE ZCEE30.SBAQCOB(BAQHCONC) Line 0000000020 Col 001 080
Command ===>
* Host API Request parameter names
 77 BAQR-DAUTH-USERNAME      PIC X(22)
    VALUE 'BAQHAPI-oAuth-Username'.
 77 BAQR-DAUTH-PASSWORD      PIC X(22)
    VALUE 'BAQHAPI-oAuth-Password'.
 77 BAQR-DAUTH-SCOPE          PIC X(19)
    VALUE 'BAQHAPI-oAuth-Scope'.
 77 BAQR-DAUTH-CLIENT-ID     PIC X(22)
    VALUE 'BAQHAPI-oAuth-ClientId'.
 77 BAQR-DAUTH-CLIENT-SECRET  PIC X(26)
    VALUE 'BAQHAPI-oAuth-ClientSecret'.
 77 BAQR-DAUTH-RESOURCE       PIC X(22)
    VALUE 'BAQHAPI-oAuth-Resource'.
 77 BAQR-DAUTH-AUDIENCE       PIC X(22)
    VALUE 'BAQHAPI-oAuth-Audience'.
 77 BAQR-DAUTH-CUSTOM-PARMS   PIC X(25)
    VALUE 'BAQHAPI-oAuth-CustomParms'.
 77 BAQR-TOKEN-USERNAME       PIC X(22)
    VALUE 'BAQHAPI-Token-Username'.
 77 BAQR-TOKEN-PASSWORD       PIC X(22)
    VALUE 'BAQHAPI-Token-Password'.
 77 BAQR-TOKEN-CUSTOM-PARMS   PIC X(25)
    VALUE 'BAQHAPI-Token-CustomParms'.
 77 BAQR-TOKEN-CUSTOM-HEADERS PIC X(27)
    VALUE 'BAQHAPI-Token-CustomHeaders'.

* Host API ZCON parameter names
 77 BAQZ-TRACE-VERBOSE        PIC X(21)
    VALUE 'BAQHAPI-Trace-Verbose'.
 77 BAQZ-SERVER-URIMAP         PIC X(21)
    VALUE 'BAQHAPI-Server-URIMAP'.
 77 BAQZ-SERVER-HOST           PIC X(19)
    VALUE 'BAQHAPI-Server-Host'.
 77 BAQZ-SERVER-PORT           PIC X(19)
    VALUE 'BAQHAPI-Server-Port'.
 77 BAQZ-SERVER-TIMEOUT        PIC X(22)
    VALUE 'BAQHAPI-Server-Timeout'.
 77 BAQZ-SERVER-USERNAME        PIC X(23)
    VALUE 'BAQHAPI-Server-Username'.
```

MA A
Connected to remote server/host wg31 using lu/pool TCP00112 and port 23 | Adobe PDF on

WG31 - 3270

```
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT JOHNSON.ZCEE.SOURCE(BAQZUSER) - 01.01 Columns 00001 00072
Command ===> Scroll ===> PAGE
***** **** Top of Data ****
==MSG> -CAUTION- Data contains invalid (non-display) characters. Use command
==MSG>     ==> FIND P'.' to position cursor to these
000001 IDENTIFICATION DIVISION.
000002 PROGRAM-ID. HBRMINM.
000003 ENVIRONMENT DIVISION.
000004 DATA DIVISION.
000005 WORKING-STORAGE SECTION.
000006 01 MY-USER PIC (10) VALUE 'myUsername'.
000007 01 MY-PSWD PIC (10) VALUE 'myPassword'.
000008 ...
000009 PROCEDURE DIVISION.
000010
000011 ....
000012 ....
000013 ***
000014 MOVE BAQR-TOKEN-USERNAME TO
000015     BAQ-ZCON-PARM-NAME OF BAQ-ZCON-PARMS(1).
000016 SET BAQ-ZCON-PARM-ADDRESS OF BAQ-ZCON-PARMS(1) TO
000017     address of MY-USER.
000018 MOVE LENGTH OF MY-USER TO
000019     BAQ-ZCON-PARM-LENGTH(1) OF BAQ-ZCON-PARMS(1).
000020
000021 MOVE BAQR-TOKEN-PASSWORD TO
000022     BAQ-ZCON-PARM-NAME OF BAQ-ZCON-PARMS(2).
000023 SET BAQ-ZCON-PARM-ADDRESS OF BAQ-ZCON-PARMS(2) TO
000024     ADDRESS OF MY-USER.
000025 MOVE LENGTH OF MY-USER TO
000026     BAQ-ZCON-PARM-LENGTH(1) OF BAQ-ZCON-PARMS(2).
***** **** Bottom of Data ****
```

MA A
Connected to remote server/host wg31 using lu/pool TCP00112 and port 23 | Adobe PDF on Documents*.pdf 05/009



Configuring JWT Custom flow

```
<zosconnect_endpointConnection id="cscvincAPI"
    host="http://wg31.washington.ibm.com" port="9080"
    authenticationConfigRef="myJWTConfig"/>

<zosconnect_authToken id="myJWTConfig" authServerRef="myJWTServer"
    header="myJWT-header-name"
    <tokenRequest/>      See next slide
    <tokenReponse/>      See next slide
</zosconnect_authToken>

<zosconnect_authorizationServer id="myJWTServer"
    tokenEndpoint=https://wg31.washington.ibm.com:59443/oidc/endpoint/OP/token1
    basicAuthRef="tokenCredential" 2
    sslCertsRef="OutboundSSLSettings" />

<zosconnect_authData id="tokenCredential" 2
    user="zCEEClient" password="secret"/>
```

¹See URL https://www.ibm.com/support/knowledgecenter/SS7K4U_liberty/com.ibm.websphere.wlp.zseries.doc/ae/twlp_oidc_token_endpoint.html

² These credentials can be specified by the application



Configuring Custom JWT flow

Request Token Example 1

```
<tokenRequest  
    credentialLocation="header"  
    header="Authorization"  
    requestMethod="GET" />
```

Response Token

```
<tokenResponse  
    tokenLocation="header"  
    header="JWTAuthorization" />
```

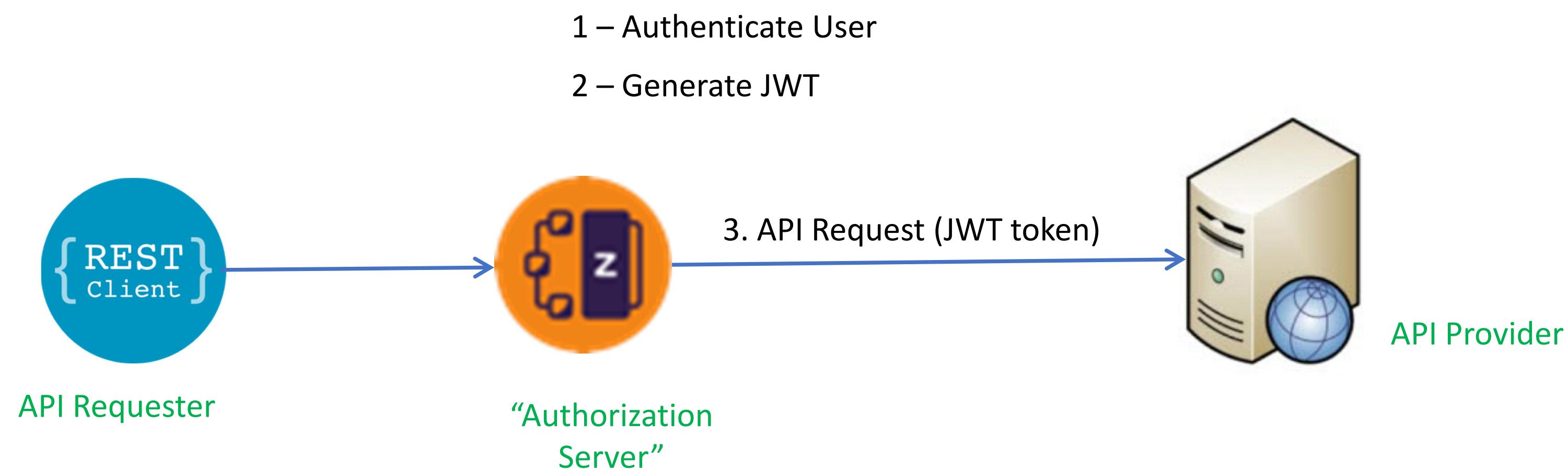
Response Token Example 2

```
<tokenRequest credentialLocation="body"  
    requestMethod="POST"  
    // Use XML escaped characters in requestBody  
    requestBody="
```

Response Token

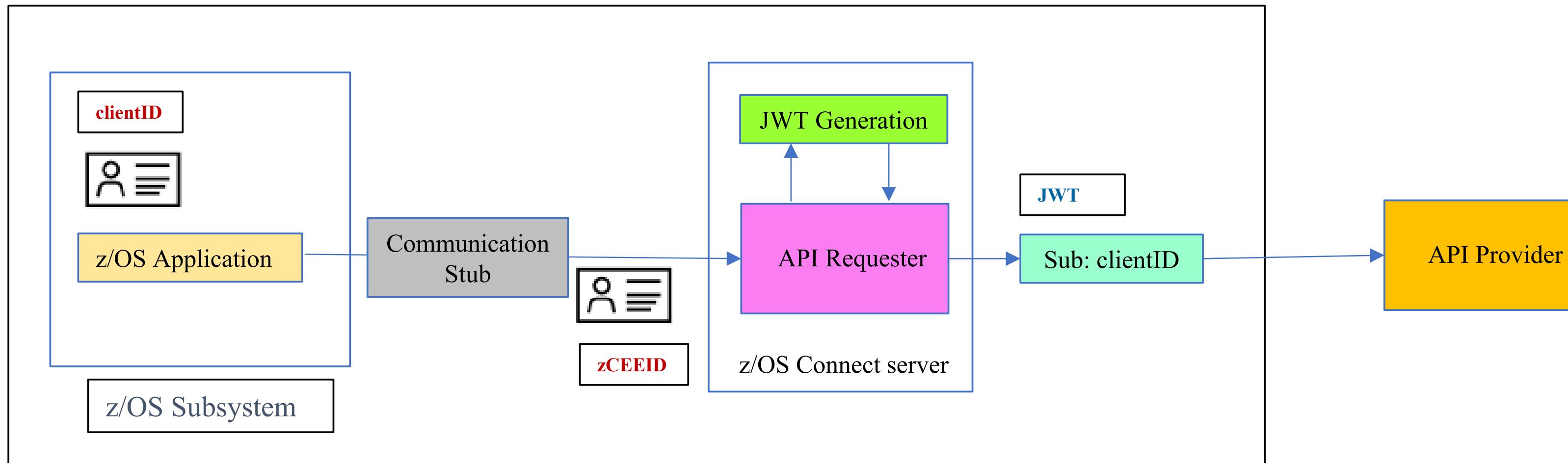
```
<tokenResponse  
    tokenLocation="body"  
    responseFormat="JSON"  
    tokenPath=".tokenname" />
```

z/OS Connect JWT Generation – V3.0.43





API Requester – JWT Generation



zCEEID – The identity that is used for authenticating connectivity the z/OS subsystem to the zCEE server. It is configured using basic authentication or for CICS, TLS client authentication.

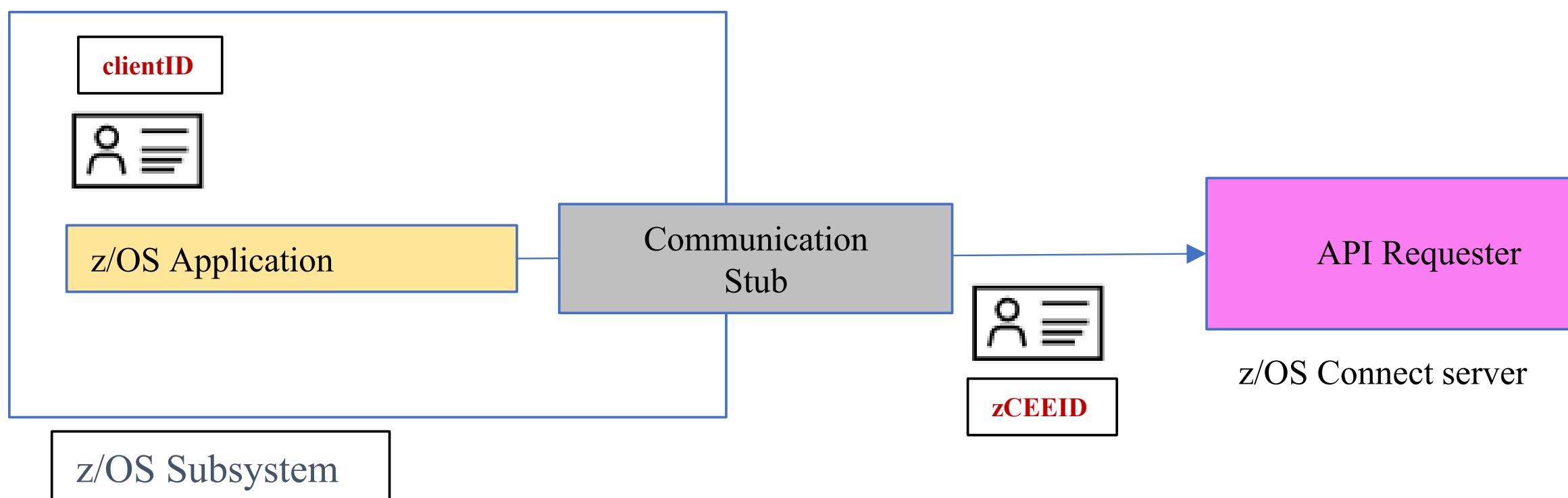
clientID – the identity under which the z/OS application is executing.

- For CICS, the task owner
- For IMS, the transaction owner
- For batch, the job owner

requireAuth	idAssertion	Actions performed by z/OS Connect
true	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server authenticates <i>zCEEID</i> and checks whether <i>zCEEID</i> is a surrogate of <i>clientID</i> . If <i>zCEEID</i> is a surrogate of <i>clientID</i> , the server further checks whether <i>clientID</i> has the authority to invoke an API requester; otherwise, a BAQR7114E message occurs.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server authenticates <i>zCEEID</i> and directly checks whether <i>clientID</i> has the authority to invoke an API requester
false	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server checks whether <i>clientID</i> has the authority to invoke an API requester, and a warning message occurs to indicate that the ASSERT_ONLY value is used instead of the ASSERT_SURROGATE value.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server checks whether <i>clientID</i> has the authority to invoke an API requester



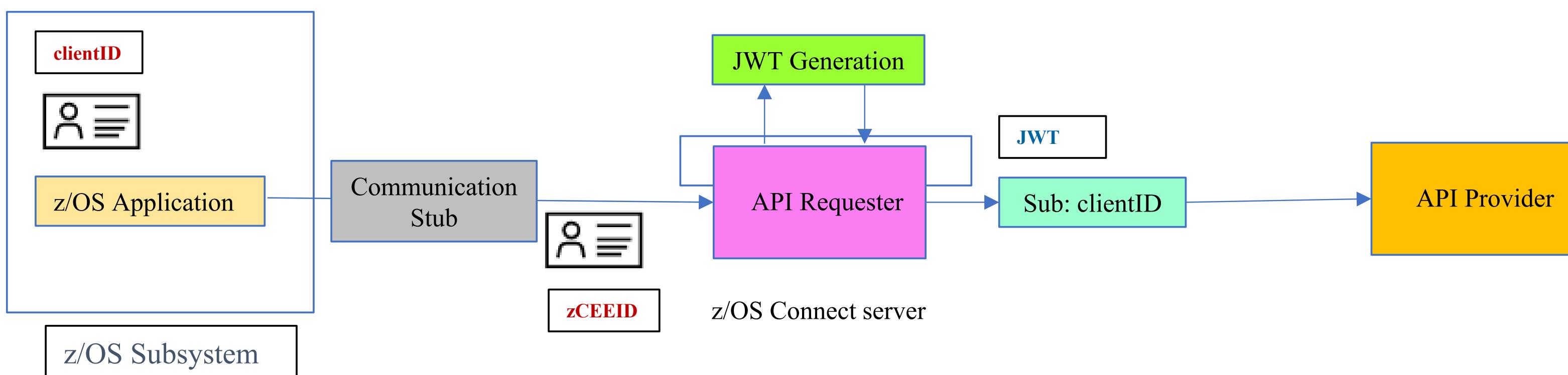
API Requester - authentication with identity assertion and JWT generation



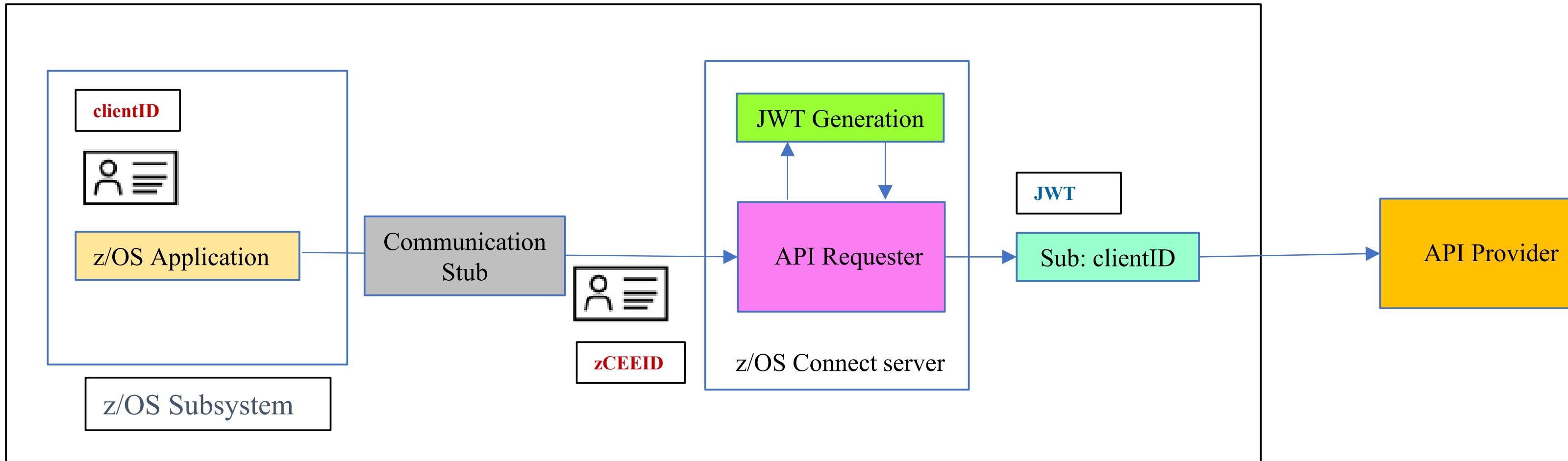
zCEEID – The identity that is used for authenticating connectivity the z/OS subsystem to the zCEE server. It is configured using basic authentication or for CICS, TLS client authentication.

clientID – the identity under which the z/OS application is executing.

- For CICS, the task owner
- For IMS, the transaction owner
- For batch, the job owner



API Requester – JWT Generation



zCEEID – The identity that is used for authenticating connectivity the z/OS subsystem to the zCEE server. It is configured using basic authentication or for CICS, TLS client authentication.

clientID – the identity under which the z/OS application is executing.

- For CICS, the task owner
- For IMS, the transaction owner
- For batch, the job owner

requireAuth	idAssertion	Actions performed by z/OS Connect
true	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server authenticates <i>zCEEID</i> and checks whether <i>zCEEID</i> is a surrogate of <i>clientID</i> . If <i>zCEEID</i> is a surrogate of <i>clientID</i> , the server further checks whether <i>clientID</i> has the authority to invoke an API requester; otherwise, a BAQR7114E message occurs.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server authenticates <i>zCEEID</i> and directly checks whether <i>clientID</i> has the authority to invoke an API requester
false	ASSERT_SURROGATE	Identity assertion is enabled. The zCEE server checks whether <i>clientID</i> has the authority to invoke an API requester, and a warning message occurs to indicate that the ASSERT_ONLY value is used instead of the ASSERT_SURROGATE value.
	ASSERT_ONLY	Identity assertion is enabled. The zCEE server checks whether <i>clientID</i> has the authority to invoke an API requester



Tech-Tip: JWT generation requires setting a program control extended attribute

As root or superuser, set the *libifaedjreg64.so* program control extended attribute bit

- *Permit the server's identity to the required FACILITY resource*

PERMIT BPX.SERVER CLASS(FACILITY) ID(LIBSERV**) ACCESS(READ)**

SETROPTS RACLIST(FACILITY) REFRESH

- *Define a SURROGAT profile for the asserted identity and permit access to connection identity*

RDEFINE SURROGAT **clientID.BAQASSRT UACC(NONE) OWNER(SYS1)**

PERMIT **clientID.BAQASSRT CLASS(SURROGAT) ACCESS(READ) ID(**zCEEID**)**

OR

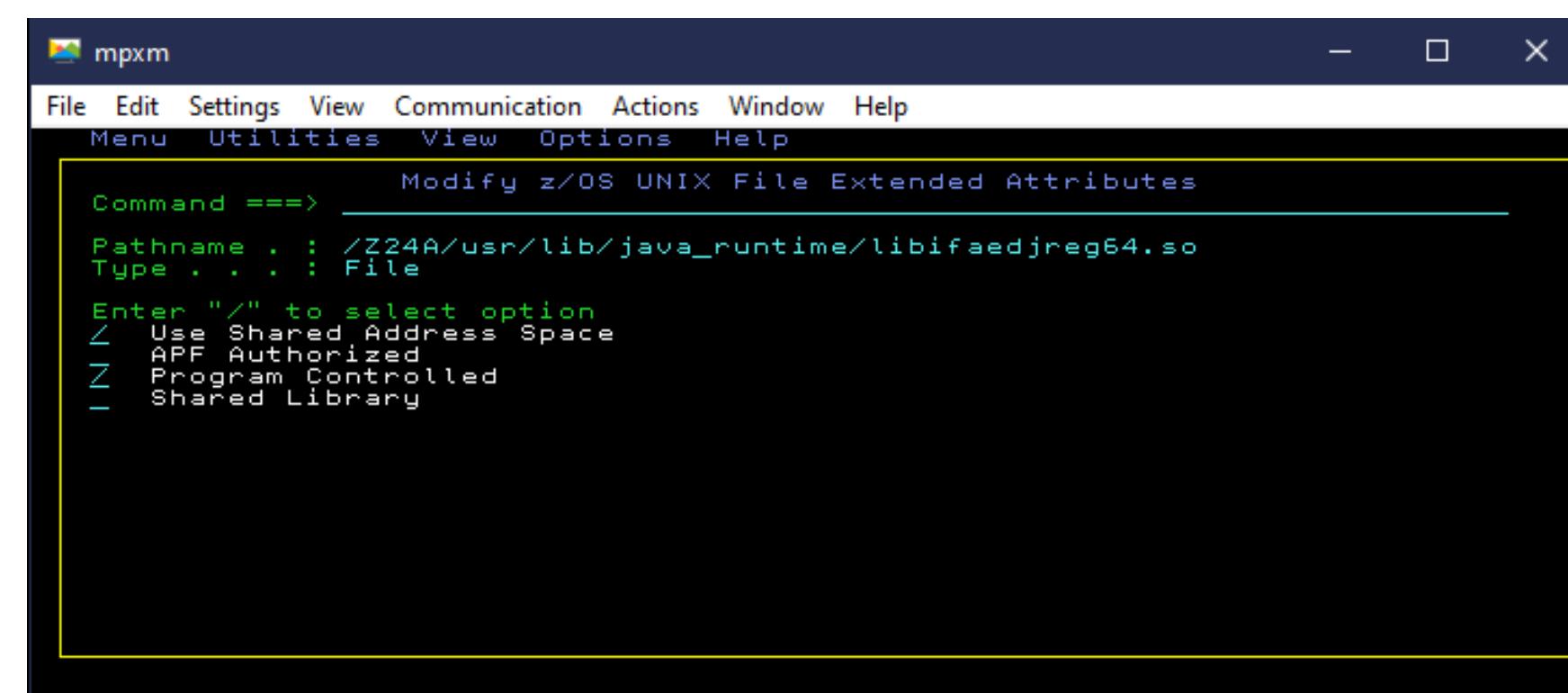
RDEFINE SURROGAT *.BAQASSRT UACC(NONE) OWNER(SYS1)

PERMIT *.BAQASSRT CLASS(SURROGAT) ACCESS(READ) ID(zCEEID**)**

SETROPTS RACLIST(SURROGAT) REFRESH

- *Enable the program control bit for Java shared object ifaedjreg64*

```
su  
cd /usr/lib/java_runtime  
extattr +p libifaedjreg64.so
```





Configuring JWT Generation support

```
<zosconnect_endpointConnection id="conn"  
    host="http://api.server.com" port="8080"  
    authenticationConfigRef="jwtConfig" />  
  
<zosconnect_authTokenLocal id="jwtConfig"  
    tokenGeneratorRef="jwtBuilder"  
    header="Authorization" >  
    <claims>{ "name":"JohnSmith,  
                "ID":"1234567890" }  
    </claims>  
  
<jwtBuilder id="jwtBuilder"  
    scope="scope1"  
    audiences="myApp1"  
    jti="true"  
    signatureAlgorithm="RS256"  
    keyStoreRef="myKeyStore"  
    keyAlias="jwtSigner"  
    issuer="z/OS Connect EE Default"/>
```

One or more Public claim (e.g., *aud,exp,nbf,iat,jti*) or
one or more private claims

The "sub" claim value will be application asserted user ID.

Configuring JWT Generation support



```
<zosconnect_endpointConnection id="conn1"  
    host="http://api.server.com" port="8080"  
    authenticationConfigRef="jwtConfig" />  
<zosconnect_endpointConnection id="conn2"  
    host="http://api.server.com" port="8080"  
    authenticationConfigRef="jwtConfig" />  
<zosconnect_authTokenLocal id="jwtConfig"  
    tokenGeneratorRef="jwtBuilder"  
    header="Authorization" >  
    <claims>{"scope":"Scope1"}</claims>  
<zosconnect_authTokenLocal id="jwtConfig"  
    tokenGeneratorRef="jwtBuilder"  
    header="Authorization" >  
    <claims>{"scope":"Scope2"}</claims>  
<jwtBuilder id="jwtBuilder"  
    scope="scope"  
    audiences="myApp1"  
    jti="true"  
    signatureAlgorithm="RS256"  
    keyStoreRef="myKeyStore"  
    keyAlias="jwtSigner"  
    issuer="z/OS Connect EE Default"/>
```



server XML Configuration

```
→<jwtBuilder id="jwtBuilder"
  scope="scope1"
  audiences="myApp1"
  jti="true"
  signatureAlgorithm="RS256"
  keyStoreRef="myKeyStore"
  keyAlias="jwtsigner"
  issuer="z/OS Connect EE Default"/>
  →<zosconnect_authTokenLocal id="jwtConfig"
    tokenGeneratorRef="jwtBuilder"
    header="JWTAuthorization" >
    <claims>{"name":"JohnSmith,
      "ID":"1234567890"}</claims>
  </ zosconnect_authTokenLocal >
  <zosconnect_endpointConnection id="conn"
    host="http://api.server.com" port="8080"
    authenticationConfigRef="jwtConfig" />
```

Configure the Liberty jwtBuilder element in server.xml.

Configure the zosconnect_authTokenLocal element, specifying any additional private claims required and the name of the header used to send the JWT to the endpoint.

header default value is Authorization

Finally, reference the JWT configuration from the zosconnect_endpointConnection element.

Useful Liberty functions/features and MVS commands



Use the adminCenter-1.0 feature to update the server XML from a browser

Administrators can use a web interface to maintain the server XML configuration.



The screenshot shows a web-based configuration tool titled "Server Config". The title bar includes icons for a briefcase, user profile, and settings, along with the text "Server Config". Below the title bar, the file name "adminCenter.xml" is displayed, with "Read only" and "Close" buttons to its right. The main content area is divided into two tabs: "Design" and "Source". The "Source" tab is selected, showing the XML code for the "adminCenter.xml" file. The code includes a "server" element with attributes like "description" and "writeDir", and nested elements like "featureManager" and "remoteFileAccess".

```
1<server description="Admin Center">
2
3  <!-- Enable features -->
4  <featureManager>
5    <feature>adminCenter-1.0</feature>
6  </featureManager>
7
8  <remoteFileAccess>
9    <writeDir>${server.config.dir}</writeDir>
10 </remoteFileAccess>
11
12</server>
13
```

```
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.Administrator OWNER(SYS1) UACC(NONE)
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.Reader OWNER(SYS1) UACC(NONE)
```

```
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.Administrator CLASS(EJBROLE) ID(FRED) ACCESS(READ)
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.Reader CLASS(EJBROLE) ID(FRED) ACCESS(READ)
```

```
SETR RACLIST(EJBROLE) REFRESH
```



Use Liberty's “adminCenter” Feature to manage server XML

- Web browser interface to the server's configuration files

The screenshot displays the IBM Liberty adminCenter interface. It features two main tabs: "Design" and "Source".

Design Tab: This tab contains several configuration settings:

- Include:** A list of XML files to include in the configuration.
- Require request authentication:** Set to "false".
- Preserve JSON object payload order:** Set to "false (default)".
- Preserve JSON payload character format:** A modal dialog is open, showing options "true" and "false (default)".
- Set response encoding to:** Set to "false (default)".
- Return all errors in JSON:** Set to "true (default)".

Source Tab: This tab shows the XML configuration file "server.xml". A content assist dropdown is open over the XML code, listing various API requester types. One item, "zosConnect_apiRequester", is highlighted.

A red circle highlights the tooltip "Press Ctrl+space for content assist." located at the top right of the Source tab area.

```
<server description="new server">
<include location="/var/zosconnect/servers/myServer/resources/imsmobile-config/services/ims-services.xml" optional="true"/>
<include location="/var/zosconnect/servers/myServer/resources/imsmobile-config/interactions/ims-interactions.xml" optional="true"/>
<include location="/var/zosconnect/servers/myServer/resources/imsmobile-config/connections/ims-connections.xml" optional="true"/>
<include location="${server.config.dir}/includes/safSecurity.xml"/>
<include location="${server.config.dir}/includes/saTrace.xml"/>
<include location="${server.config.dir}/includes/ipic.xml"/>
<include location="${server.config.dir}/includes/keyring.xml"/>
<include location="${server.config.dir}/includes/apiRequesterHTTPS.xml"/>
<include location="${server.config.dir}/includes/shared.xml"/>
<include location="${server.config.dir}/includes/oauth.xml"/>
<include location="${server.config.dir}/includes/audit.xml"/>
<include location="${server.config.dir}/includes/mq.xml"/>
<include location="${server.config.dir}/includes/db2.xml"/>
<include location="${server.config.dir}/includes/wlm.xml"/>
<include location="${server.config.dir}/includes/restConnector.xml"/>
<wsSecurityProvider>
<zosconnect_apiRequester>
<zosconnect_apiRequesters>
<zosconnect_auditInterceptor>
<zosconnect_authData>
<zosconnect_authorizationInterceptor>
<zosconnect_authorizationServer>
<zosconnect_authToken>
<zosconnect_zosConnectServiceRestClientBasicAuth />

<httpEndpoint host="*" httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint"/>

<cors allowCredentials="true" allowedHeaders="Origin, Content-Type, Authorization, Cache-Control, Expires, Pragma" allowedMethods="GET, POST, PUT, DELETE, HEAD, OPTIONS" maxAge="1728000" />
```



Use the restConnector-2.0 feature to see real time configuration details

A secure, REST administrative connector that enables remote access from a Java client or Web browser (GET only) or directly through an HTTPS call to the current runtime configuration.

The screenshot shows the 'Server Config' interface with the title 'restConnector.xml'. It has tabs for 'Design' and 'Source'. The 'Source' tab is selected, displaying XML code for a server configuration. A callout box highlights the XML path 'https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectManager' and its corresponding URL pattern 'https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectManager'. Below this, a box contains RACF commands defining security roles and resources for the restConnector-2.0 feature.

URI Path is the concatenation of the path `/ibm/api/config` with the server XML configuration element and any optional query strings.

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="REST Connector">
  <featureManager>
    <feature>restConnector-2.0</feature>
  </featureManager>
</server>
```

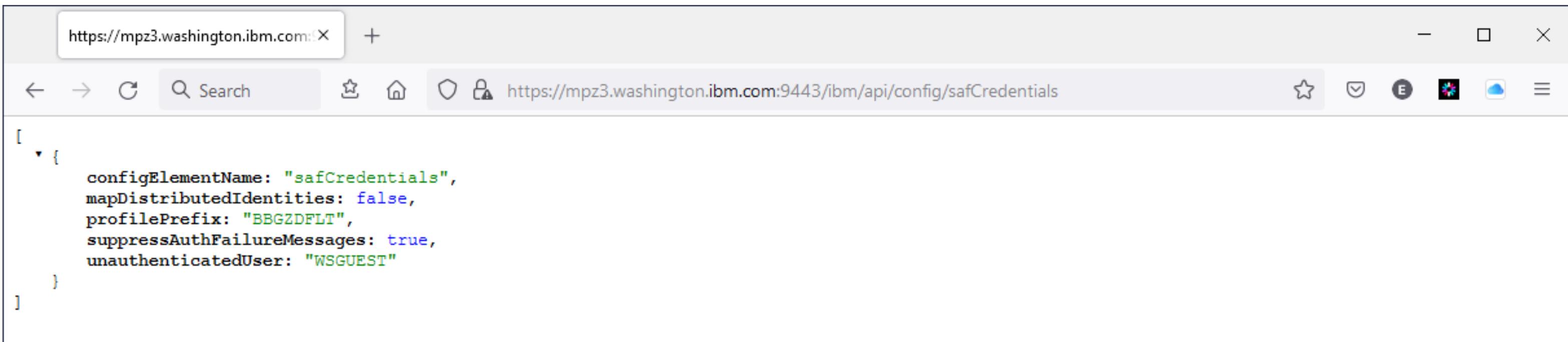
<https://mpz3.washington.ibm.com:9443/ibm/api/config/jmsQueue>
https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_cicsIpicConnection?port=1491
https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectServiceRestClientConnection
https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_cicsIpicConnection?id=miniloan
<https://mpz3.washington.ibm.com:9443/ibm/api/config/safCredentials>
<https://mpz3.washington.ibm.com:9443/ibm/api/config/connectionFactory>
https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectManager
<https://mpz3.washington.ibm.com:9443/ibm/api/config/keyStore>
<https://mpz3.washington.ibm.com:9443/ibm/api/config/ssl>
<https://mpz3.washington.ibm.com:9443/ibm/api/config/sslDefault>
https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectManager
https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_zosConnectAPIs
https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_services
https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_apiRequesters

```
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.Administrator OWNER(SYS1) UACC(NONE)
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.Reader OWNER(SYS1) UACC(NONE)
RDEFINE EJBROLE BBGZDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers OWNER(SYS1) UACC(NONE)
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.Administrator CLASS(EJBROLE) ID(ZCEEUSRS) ACCESS(READ)
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.Reader CLASS(EJBROLE) ID(ZCEEUSRS) ACCESS(READ)
PERMIT BBGZDFLT.com.ibm.ws.management.security.resource.allAuthenticatedUsers CLASS(EJBROLE) ID(ZCEEUSRS)
ACCESS(READ)
SETR RACLIST(EJBROLE) REFRESH
```



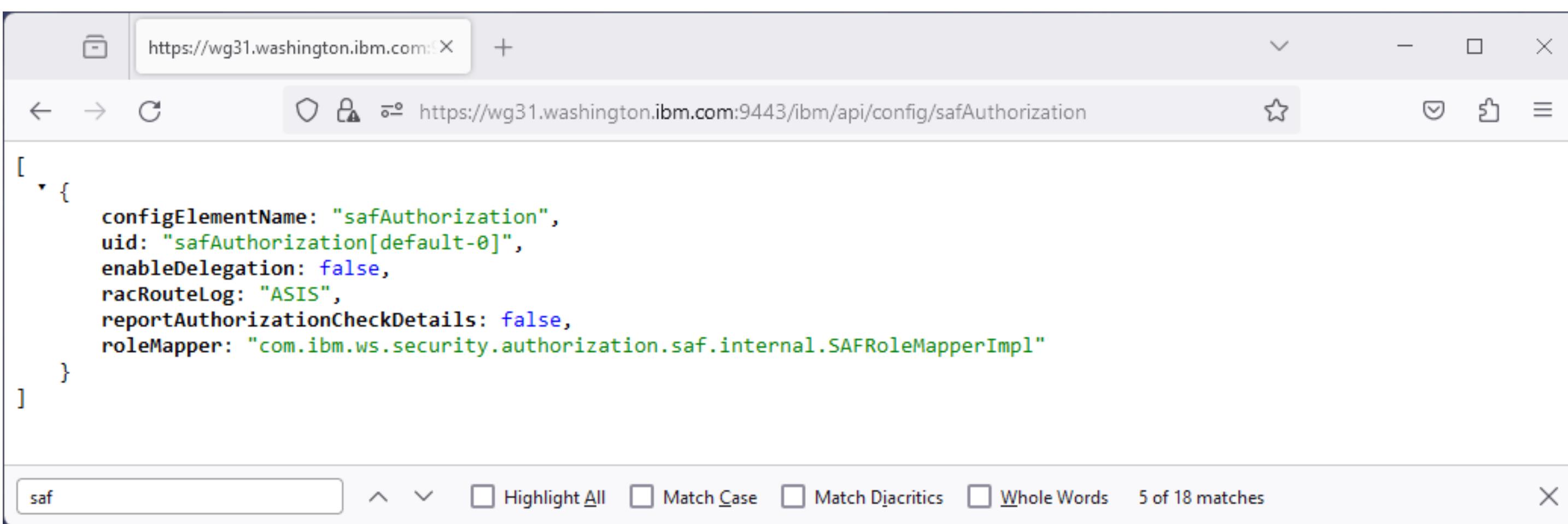
restConnector-2.0 examples – safCredentials/safAuthorization

`https://mpz3.washington.ibm.com:9443/ibm/api/config/safCredentials`



```
[{"configElementName": "safCredentials", "mapDistributedIdentities": false, "profilePrefix": "BBGZDFLT", "suppressAuthFailureMessages": true, "unauthenticatedUser": "WSGUEST"}]
```

`https://mpz3.washington.ibm.com:9443/ibm/api/config/safAuthorization`



```
[{"configElementName": "safAuthorization", "uid": "safAuthorization[default-0]", "enableDelegation": false, "racRouteLog": "ASIS", "reportAuthorizationCheckDetails": false, "roleMapper": "com.ibm.ws.security.authorization.saf.internal.SAFRoleMapperImpl"}]
```

saf

Highlight All Match Case Match Diacritics Whole Words 5 of 18 matches



restConnector-2.0 examples – zosconnect_cicsIpicConnection

https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_cicsIpicConnection?port=1491

The screenshot shows a web browser window displaying a JSON array of configuration elements. The URL in the address bar is https://mpz3.washington.ibm.com:9443/ibm/api/config/zosconnect_cicsIpicConnection?port=1491. The JSON data lists four connection configurations:

```
[{"configElementName": "zosconnect_cicsIpicConnection", "uid": "catalog", "id": "catalog", "connectionTimeout": 30000, "heartbeatInterval": 30000, "host": "wg31.washington.ibm.com", "port": 1491, "sharedPort": false, "transidUsage": "EIB_AND_MIRROR"}, {"configElementName": "zosconnect_cicsIpicConnection", "uid": "cscvinc", "id": "cscvinc", "connectionTimeout": 30000, "heartbeatInterval": 30000, "host": "wg31.washington.ibm.com", "port": 1491, "sharedPort": false, "transidUsage": "EIB_AND_MIRROR"}, {"configElementName": "zosconnect_cicsIpicConnection", "uid": "minilcan1", "id": "minilcan1", "connectionTimeout": 30000, "heartbeatInterval": 30000, "host": "wg31.washington.ibm.com", "port": 1491, "sharedPort": false, "transidUsage": "EIB_AND_MIRROR"}, {"configElementName": "zosconnect_cicsIpicConnection", "uid": "minilcan", "id": "minilcan", "connectionTimeout": 30000, "heartbeatInterval": 30000, "host": "wg31.washington.ibm.com", "port": 1491, "sharedPort": false, "transidUsage": "EIB_AND_MIRROR"}]
```



restConnector-2.0 examples – featureManager

<https://mpz3.washington.ibm.com:9443/ibm/api/config/featureManager>

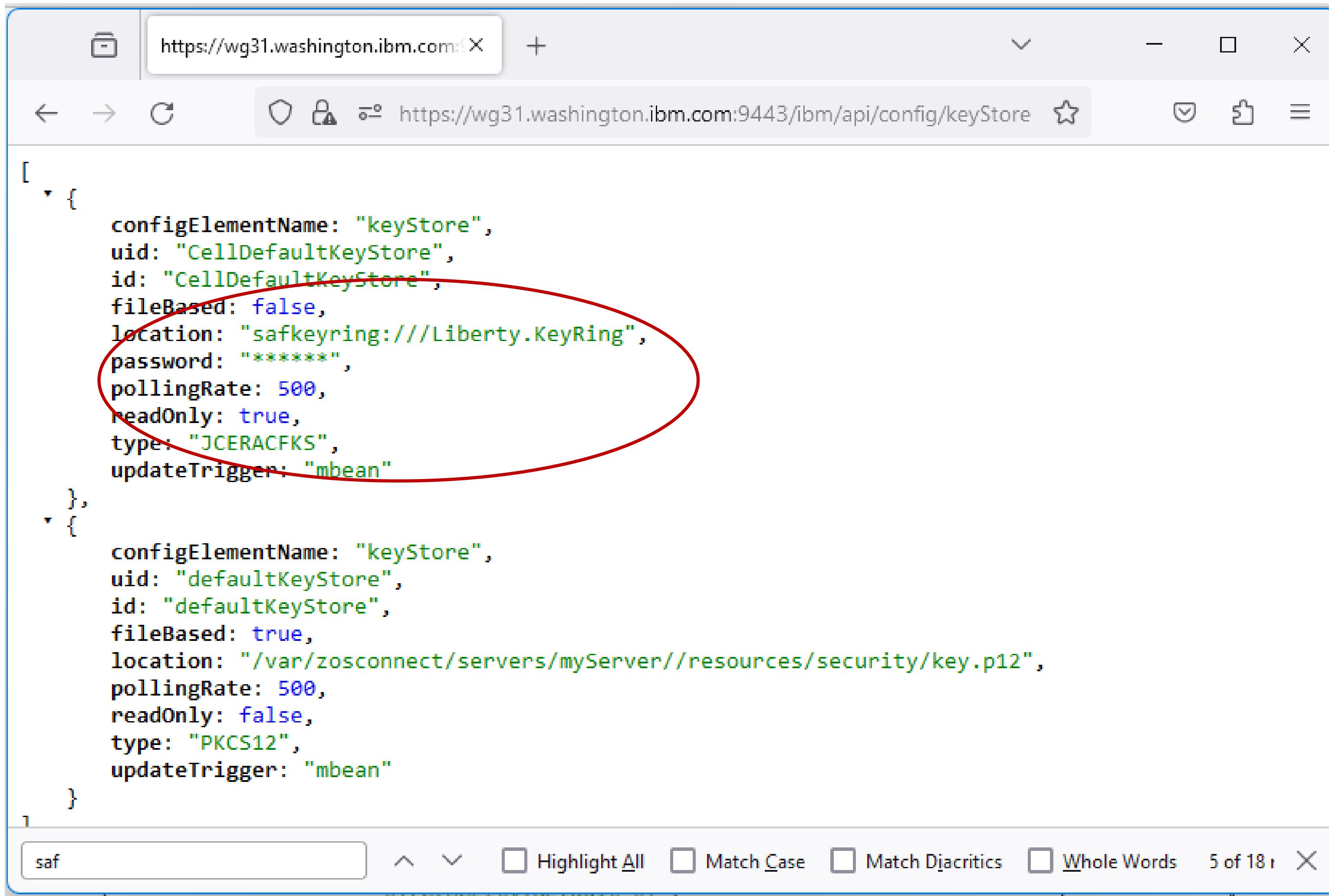
The screenshot shows a browser window with the URL <https://mpz3.washington.ibm.com:9443/ibm/api/config/featureManager>. The page displays a JSON object representing the featureManager configuration. The JSON structure is as follows:

```
[{"configElementName": "featureManager", "feature": ["appSecurity-2.0", "zosSecurity-1.0", "zosconnect:cicsService-1.0", "transportSecurity-1.0", "zosconnect:apiRequester-1.0", "zosconnect:apiRequester-1.0", "zosconnect:mqService-1.0", "zosWlm-1.0", "restConnector-2.0", "monitor-1.0", "zosRequestLogging-1.0", "adminCenter-1.0", "apiDiscovery-1.0", "zosconnect:zosConnect-2.0", "zosconnect:zosConnectCommands-1.0", "imsmobile:imsmobile-2.0"], "onError": "WARN"}]
```



restConnector-2.0 examples – keyStore

https://mpz3.washington.ibm.com:9443/ibm/api/config/keyStore



```
[  
  {  
    configElementName: "keyStore",  
    uid: "CellDefaultKeyStore",  
    id: "CellDefaultKeyStore",  
    fileBased: false,  
    location: "safkeyring:///Liberty.KeyRing",  
    password: "*****",  
    pollingRate: 500,  
    readOnly: true,  
    type: "JCERACFKS",  
    updateTrigger: "mbean"  
  },  
  {  
    configElementName: "keyStore",  
    uid: "defaultKeyStore",  
    id: "defaultKeyStore",  
    fileBased: true,  
    location: "/var/zosconnect/servers/myServer//resources/security/key.p12",  
    pollingRate: 500,  
    readOnly: false,  
    type: "PKCS12",  
    updateTrigger: "mbean"  
  }  
]
```



Use the **apiDiscovery-1.0** or **OpenAPI-3.0** features to execute RESTful APIs directly

The screenshot shows a browser window titled "IBM REST API Documentation" with the URL <https://mpz3.washington.ibm.com:9443/api/explorer/#/cscvinc>. The page displays the "Liberty REST APIs" section, specifically for the "cscvinc" service. It lists several API operations:

- cscvinc**:
 - POST /cscvinc/employee
 - DELETE /cscvinc/employee/{employee}
 - GET /cscvinc/employee/{employee}
 - PUT /cscvinc/employee/{employee}
- db2employee**:
 - Show/Hide | List Operations | Expand Operations
- filemgr**:
 - Show/Hide | List Operations | Expand Operations
- imsPhoneBook**:
 - Show/Hide | List Operations | Expand Operations
- jwtlvpDemoApi**:
 - Show/Hide | List Operations | Expand Operations
- miniloancics**:
 - Show/Hide | List Operations | Expand Operations
- mqapi**:
 - Show/Hide | List Operations | Expand Operations
- phonebook**:
 - Show/Hide | List Operations | Expand Operations



IBM MQ Administrative REST API

qmgr		Show/Hide List Operations Expand Operations
GET	/ibmmq/rest/v1/admin/qmgr	Retrieves details of all queue managers in the IBM MQ installation.
GET	/ibmmq/rest/v1/admin/qmgr/{qmgr}	Retrieves details of a specific queue manager in the IBM MQ installation.
*	GET /ibmmq/rest/v2/admin/qmgr	Retrieves details of all queue managers in the IBM MQ installation.
*	GET /ibmmq/rest/v2/admin/qmgr/{qmgr}	Retrieves details of a specific queue manager in the IBM MQ installation.
qmgr : action		Show/Hide List Operations Expand Operations
POST	/ibmmq/rest/v1/admin/action/qmgr/{qmgrName}/mqsc	Runs an MQSC command.
*	POST /ibmmq/rest/v2/admin/action/qmgr/{qmgrName}/mqsc	Runs an MQSC command.
queue		Show/Hide List Operations Expand Operations
GET	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue	Retrieves details of all queues.
POST	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue	Creates a queue.
DELETE	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue/{qName}	Deletes a queue.
GET	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue/{qName}	Retrieves details of a specific queue.
PATCH	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/queue/{qName}	Modifies a queue.
subscription		Show/Hide List Operations Expand Operations
GET	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/subscription	Retrieves details of all subscriptions.
GET	/ibmmq/rest/v1/admin/qmgr/{qmgrName}/subscription/{name}	Retrieves details of a specific subscription.

* If you are accessing a version earlier than V9.1.5 you must use v1

IBM MQ Messaging REST API Support



messaging

Show/Hide | List Operations | Expand Operations

DELETE	/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{qName}/message	Retrieves the next message from a specified queue.
GET	/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{qName}/message	Browses the next message from a specified queue.
POST	/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{qName}/message	Sends a message to a specified queue.
GET	/ibmmq/rest/v1/messaging/qmgr/{qmgrName}/queue/{qName}/messagelist	Browses messages from a specified queue.
* DELETE	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{qName}/message	Retrieves the next message from a specified queue.
* GET	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{qName}/message	Browses the next message from a specified queue.
* POST	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{qName}/message	Sends a message to a specified queue.
* GET	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/queue/{qName}/messagelist	Browses messages from a specified queue.
* POST	/ibmmq/rest/v2/messaging/qmgr/{qmgrName}/topic/{topicString}/message	Publishes a message to a specified topic.

* If you are accessing a version earlier than V9.1.5 you must use v1 rather than v2



Provide remote access to configuration/log information

The image displays three separate browser windows, each showing a different type of server configuration or log file via HTTPS:

- Top Left Window:** Shows the XML configuration file for a new server. The URL is <https://wg31.washington.ibm.com:9443/se>. The content includes various include statements pointing to configuration files like safSecurity.xml, ipicSSLIDProp.xml, keyringOutbound.xml, groupAccess.xml, shared.xml, oauth.xml, and adminCenter.xml.
- Top Right Window:** Shows the XML configuration for a web application named "serverConfig". The URL is <https://wg31.washington.ibm.com:9443/server/config>. The configuration includes enabling file serving and directory browsing, setting the extended document root to \${server.config.dir}, and defining a file-serving attribute.
- Bottom Window:** Shows a log file titled "messages.log". The URL is <https://wg31.washington.ibm.com:9443/server/config/logs/messages.log>. The log output provides detailed runtime information, including product version (WAS FOR Z/OS 20.0.0.6), Java version (1.8.0_261), and operating system details (z/OS 02.03.00; s390x) (en_US). It also shows log entries for various server components and security features.



Liberty MVS Commands

F BAQSTRT,CACHE,CLEAR,AUTH

Clears all users that are cached in the Liberty authentication cache

F BAQSTRT,REFRESH,CONFIG

Process pending configuration updates. Configuration processing applies to the server.xml file, any files it includes

F BAQSTRT,REFRESH,APPS

Process pending application updates. (Applicable to OpenAPI 3 servers only)

F BAQSTRT,REFRESH,KEYSTORE

Use the command to refresh the keystore instorage profiles for the server.

F BAQSTRT,REFRESH,KEYSTORE, ID=*OutboundKeyRing*

To refresh a specific keystore defined in the server XML with ID=OutboundKeyRing.

F BAQSTRT,CACHE,CLEAR,AUTH

Clears all users that are cached in the Liberty authentication cache.

F BAQSTRT,PAUSE

To pause the server

F BAQSTRT,STATUS

To display the current status of a server

F BAQSTRT,RESUME

To resume the server

For more details, see URL <https://www.ibm.com/docs/en/was-liberty/zos?topic=zos-modify-commands>



Liberty MVS Angel Commands

F BAQZANGL,DISPLAY,SERVERS

Displays a list of servers currently connected to the angel

F BAQZANGL,DISPLAY,SERVERS,PID

Displays a list of servers currently connected to the angel code along with the server's PIDs.

```
CWWKB0067I ANGEL DISPLAY OF ACTIVE SERVERS
CWWKB0080I ACTIVE SERVER ASID 4d JOBNAME ZCEEAPIR PID 16777398
CWWKB0080I ACTIVE SERVER ASID 4b JOBNAME ZCEEDVM PID 50331780
CWWKB0080I ACTIVE SERVER ASID 4f JOBNAME WLPRPSRV PID 138
CWWKB0080I ACTIVE SERVER ASID 4a JOBNAME ZCEESRVR PID 50331815
CWWKB0080I ACTIVE SERVER ASID 50 JOBNAME ZCEEOPID PID 33554605
CWWKB0080I ACTIVE SERVER ASID 4c JOBNAME ZCEEHATS PID 143
CWWKB0080I ACTIVE SERVER ASID 4e JOBNAME WLPOPSRV PID 33554565
CWWKB0080I ACTIVE SERVER ASID 58 JOBNAME MQWEBS PID 152
```

F BAQZANGL,VERSION

Displays the version level of the angel



z/OS Connect MVS Commands (OpenAPI 2)

```
<feature>zosconnect:zosConnectCommands-1.0</feature>
```

F BAQSTRT,ZCON,REFRESH

All updated z/OS Connect artifacts (APIs, services, and API Requesters) are reloaded.

F BAQSTRT,ZCON,CLEARTOKENCACHE

Clears all OAuth 2.0 access tokens and JWTs from the cache. The token cache is only applicable for OAuth 2.0 access tokens and JWTs that were generated either locally or by an external authentication server, when invoking API requesters.

F BAQSTRT,ZCON,CLEARSAFCACHE

Clears the SAF cache. The SAF cache contains SAF user IDs and any associated RACF groups in which the user ID resides. The SAF cache is only applicable to API requester, and only when ID assertion is enabled.

F BAQSTRT,REFRESH,APPS

Monitoring Java, Liberty and z/OS Connect



Java Health Center – Monitors the Java environment

Configuring the Monitoring Agent using JVM directives

Java Health Center Directives

- healthcenter:level=headless *run without a client*
- com.ibm.java.diagnostics.healthcenter.headless.output.directory=/var/zcee/hcd *directory where HCD will be stored*
- com.ibm.java.diagnostics.healthcenter.socket.readwrite=on *collect socket sent/receive data*
- com.ibm.java.diagnostics.healthcenter.headless.files.to.keep=2 *number of HCD files to retain*
- com.ibm.java.diagnostics.healthcenter.headless.delay.start=value=0 *delay start value in minutes*
- com.ibm.java.diagnostics.healthcenter.headless.run.pause.duration=0 *pause between runs, in minutes*
- com.ibm.java.diagnostics.healthcenter.headless.run.duration=0 *run duration, in minutes*
- com.ibm.java.diagnostics.healthcenter.headless.run.number.of.runs=0 *number of runs*
- com.ibm.diagnostics.healthcenter.readonly=on *no client connections allowed*

**Add directives to bootstrap.properties or a JVM properties file, e.g.,
/var/zcee/properties/zceeHCD.properties**

```
-Dcom.ibm.tools.attach.enable=yes  
-Xhealthcenter:level=headless  
-Dcom.ibm.java.diagnostics.healthcenter.headless.output.directory=/var/zcee/hcd  
-Dcom.ibm.java.diagnostics.healthcenter.socket.readwrite=on -Dcom.ibm.diagnostics.healthcenter.readonly=on  
-Dcom.ibm.java.diagnostics.healthcenter.headless.run.duration=5  
-Dcom.ibm.java.diagnostics.healthcenter.headless.run.number.of.runs=1 #
```

* All the health center directives should be on one line.

For details on these and other Health Center configuration properties, see URL
<https://www.ibm.com/docs/en/mon-diag-tools?topic=agent-health-center-configuration-properties>

Java Health Center – Monitoring Agent Configuration



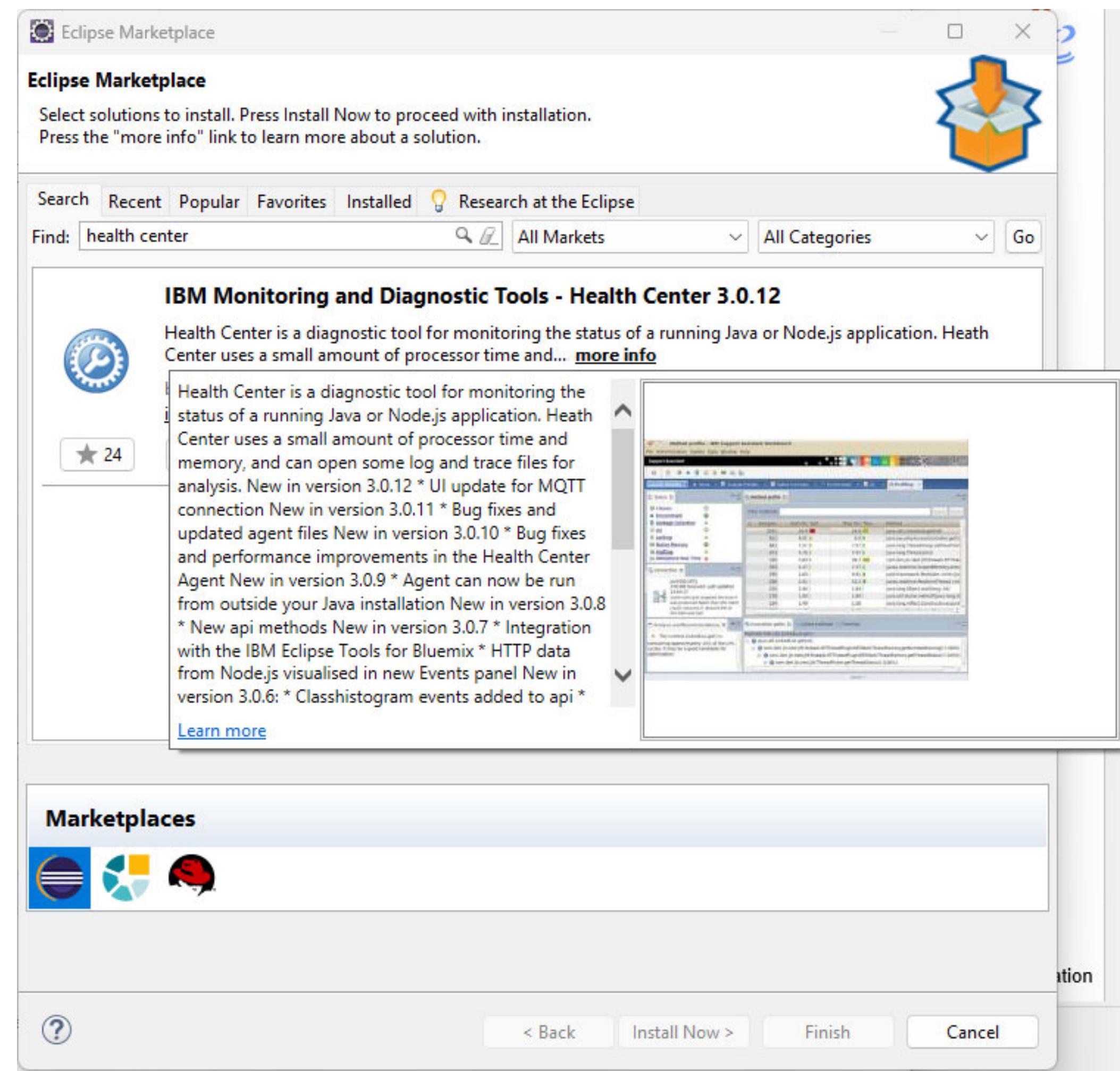
Set the `JVM_OPTIONS` environment variable to the properties file containing the health center directives

```
SYS1.PROCLIB(BAQSTRT)
//BAQSTRT PROC PARMS='myServer --clean'
//*
// SET ZCONHOME='/usr/lpp/IBM/zosconnect/v3r0'
//*
//ZCON      EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=8G,
//              PARM='PGM &ZCONHOME./bin/zosconnect run &PARMS.'
//STEPLIB   DD DISP=SHR,DSN=MQ91#.SCSQAUTH
//          DD DISP=SHR,DSN=MQ91#.SCSQANLE
//STDERR    DD SYSOUT=*,FREE=CLOSE,SPIN=(UNALLOC,1M)
//STDOUT    DD SYSOUT=*
//STDIN     DD DUMMY
//STDENV    DD *
_BPX_SHAREAS=YES
JAVA_HOME=/usr/lpp/java/J8.0_64/
WLP_USER_DIR=/var/zosconnect
JVM_OPTIONS=-Xoptionsfile=/var/zcee/properties/zceeHCD.properties
```

Java Health Center – Client Configuration



The Java health center client is available on the Eclipse Marketplace can be installed in most Eclipse workspace, e.g., IBM z/OS Explorer, etc.



Java Health Center – HEAP analysis example



The screenshot shows the Eclipse IDE interface with the IBM Monitoring and Diagnostic Tools - Health Center plugin installed. The main window displays a 'Heap and pause times' graph and a 'Summary' table.

Graph View:

- Y-axis: size (MB) and time (ms).
- X-axis: elapsed time (minutes).
- Legend: Used heap (after collection) (solid purple line), Heap size (dashed green line), Pause time (dotted blue line).
- The graph shows a general upward trend in heap usage over time, with several major pauses indicated by the pause time series.

Summary View:

Metric	Value
Concurrent collection count	10
GC Mode	Default (gencon)
Global collections - Mean garbage collection pause	6.29 ms
Global collections - Mean interval between collections	2110 ms
Global collections - Number of collections	12
Largest memory request	199 KB
Mean garbage collection pause	3.5 ms
Mean interval between collections	129 ms
Minor collections - Mean garbage collection pause	3.39 ms
Minor collections - Mean interval between collections	134 ms
Minor collections - Number of collections	310
Minor collections - Total amount flipped	338073 KB
Minor collections - Total amount tenured	52.64 MB
Number of collections	322
Number of collections triggered by allocation failure	312
Proportion of time spent in garbage collection pauses (%)	2.71%
Proportion of time spent unpause (%)	97.29%
Rate of garbage collection	2643 MB/minute
Total amount flipped	338073 KB

Help View:

- Topic: Using the garbage collection perspective
- Content:
 - Views for basic garbage collection information
 - Views for detailed garbage collection information

Java Health Center – Network analysis example



smf - Eclipse

File Edit Navigate Search Project Data Run Monitored System Window Help

Status Connection Sockets

CPU Classes Environment Events Garbage Collection I/O Locking Method Profiling Method Trace Native Memory Network Threads WebSphere Real Time

Socket ID filter: Apply Clear

ID	Type	IP Address	Port	Data sent	Data received	State	Thread [ID] Name
102	Client	0:0:0:ffff:c0a8:11c9	1491	116043 bytes	42284 bytes	Closed	[0x29d2fa00] Equino...
103	Client	0:0:0:ffff:c0a8:11c9	1491	116043 bytes	42284 bytes	Open	[0x2a00aa00] Default...
112	Server	0:0:0:ffff:c0a8:3c	65470	32953 bytes	38334 bytes	Open	[0x2a253d00] Shared...
127	Server	0:0:0:ffff:c0a8:3c	59411	87343 bytes	98768 bytes	Closed	[0x2a019f00] Default...
136	Server	0:0:0:ffff:c0a8:11c9	2446	87343 bytes	98768 bytes	Open	[0x2b38c800] Default...
138	ServerS...	0:0:0:0:0:0	9080	8818 bytes	8818 bytes	Open	[0x2a253d00] Shared...
144	Server	0:0:0:ffff:c0a8:3c	59412	4248 bytes	8818 bytes	Open	[0x2a019f00] Default...
164	ServerS...	0:0:0:0:0:0	9443	8818 bytes	8818 bytes	Open	[0x2a253d00] Shared...
176	Client	0:0:0:ffff:c0a8:11c9	4000	182558 bytes	186691 bytes	Closed	[0x2a00aa00] Default...
183	Client	0:0:0:ffff:c0a8:11c9	4000	182558 bytes	186691 bytes	Open	[0x2a14f400] Default...
186	Server	0:0:0:0:ffff:c0a8:11f3	7883	62048 bytes	116825 bytes	Open	[0x2a253d00] Shared...
196	Server	0:0:0:0:ffff:c0a8:3c	61723	1428 bytes	602 bytes	Closed	[0x29fcbb00] Default...
204	Server	0:0:0:0:ffff:c0a8:11f3	7880	1428 bytes	602 bytes	Open	[0x2a253d00] Shared...
215	Client	0:0:0:0:ffff:c0a8:11c9	1491	116825 bytes	62048 bytes	Open	[0x2b38c800] Default...
226	Server	0:0:0:0:ffff:c0a8:11f3	7863	2447 bytes	1059 bytes	Closed	[0x2a00aa00] Default...
227	Server	0:0:0:0:ffff:c0a8:11f3	9463	9892 bytes	8675 bytes	Open	[0x2aa3c100] Default...
228	Server	0:0:0:0:ffff:c0a8:11f3	7849	1059 bytes	9892 bytes	Closed	[0x29fcbb00] Default...
230	Server	0:0:0:0:ffff:c0a8:11f3	7850	54048 bytes	39936 bytes	Open	[0x2a00aa00] Default...
231	Server	0:0:0:0:ffff:c0a8:11f3	9463	10868 bytes	7460 bytes	Open	[0x2a14f400] Default...
233	Server	0:0:0:0:ffff:c0a8:11f3	9463	22059 bytes	11436 bytes	Open	[0x2a00aa00] Default...
234	Server	0:0:0:0:ffff:c0a8:11f3	7010	11436 bytes	22059 bytes	Closed	[0x2b20-0001] Default...

Sockets open Network I/O

number (amount)

elapsed time (minutes)

c0a8:11c9 = 192.168.17.201

Java Health Center – Method Profiling



The screenshot shows the Eclipse Java Health Center interface for method profiling. The main window has a title bar "smf - Eclipse" and a menu bar with File, Edit, Navigate, Search, Project, Data, Run, Monitored System, Window, Help.

The left sidebar contains a navigation tree with the following items:

- CPU
- Classes
- Environment
- Events
- Garbage Collection
- I/O
- Locking
- Method Profiling
- Method Trace
- Native Memory
- Network
- Threads
- WebSphere Real Time

The "Method Profiling" item is selected and highlighted in blue.

The central area displays two tables of method samples:

Samples	Self (%)	Self	Tree (%)	Tree	Method
2806	27.17	■	27.28	■	com.ibm.crypto.provider.MD5.a(byte[], int, int, byte[], int)
562	5.44	■	7.26	■	com.ibm.ws.logging.utils.FileLogHolder.writeRecord(java.lang.String)
440	4.26	■	21.36	■	com.ibm.ws.logging.internal.impl.BaseTraceService.publishTraceLogRecord(com.ibm.ws.logging...)
264	2.56	■	2.56	■	java.math.Division.monReduction(int[], java.math.BigInteger, int)
183	1.77	■	1.79	■	java.math.Multiplication.square(int[], int, int[])
172	1.67	■	2.32	■	javax.security.auth.Subject.toString(boolean)
150	1.45	■	1.47	■	java.math.DivisionLong.monReduceSq(long[], long[], long, int, long[])
130	1.26	■	1.83	■	com.ibm.crypto.provider.MD5.b(byte[], int, int, byte[], int)
128	1.24	■	1.55	■	com.ibm.crypto.provider.MD5.c(byte[], int, int, byte[], int)
115	1.11	■	1.14	■	java.math.DivisionLong.monMulSq(long[], int, long[])
102	0.99	■	5.32	■	com.ibm.ws.logging.utils.FileLogHolder.writeRecord(java.lang.String)
97	0.94	■	1.91	■	com.ibm.ws.logging.internal.impl.BaseTraceService.publishTraceLogRecord(com.ibm.ws.logging...)
92	0.89	■	1.31	■	org.eclipse.osni.interceptor

Samples	Self (%)	Self	Tree (%)	Tree	Method
1768	45.63	■	45.78	■	com.ibm.crypto.provider.MD5.a(byte[], int, int, byte[], int)
173	4.46	■	6.3	■	com.ibm.ws.logging.utils.FileLogHolder.writeRecord(java.lang.String)
152	3.92	■	18.68	■	com.ibm.ws.logging.internal.impl.BaseTraceService.publishTraceLogRecord(com.ibm.ws.logging...)
111	2.86	■	2.86	■	java.math.Division.monReduction(int[], java.math.BigInteger, int)
96	2.48	■	2.48	■	java.math.Multiplication.square(int[], int, int[])
56	1.45	■	2.04	■	com.ibm.crypto.provider.X.add(com.ibm.crypto.provider.EllipticPoint)
54	1.39	■	1.45	■	java.math.DivisionLong.monReduceSq(long[], long[], long, int, long[])
54	1.39	■	1.94	■	javax.security.auth.Subject.toString(boolean)
53	1.37	■	1.45	■	java.math.DivisionLong.monMulSq(long[], int, long[])
51	1.32	■	1.63	■	com.ibm.crypto.provider.P256PrimeField.a(int[])
43	1.11	■	3.59	■	java.math.Multiplication.multPAP(int[], int[], int[], int, int)
39	1.01	■	5.01	■	com.ibm.ws.logging.internal.impl.BaseTraceFormatter.formatObj(java.lang.Object)
27	0.7	■	1.42	■	com.ibm.ws.logging.internal.impl.BaseTraceFormatter.createFormattedMessage(java.util.Logging

The bottom section contains two line graphs:

- Samples over time:** A line graph showing the number of samples (#) versus elapsed time (minutes). A red circle highlights a peak around 2:30 minutes. A blue arrow points from this peak to the second graph.
- Invocation paths:** A line graph showing the number of samples (#) versus elapsed time (minutes), showing a similar trend to the first graph.

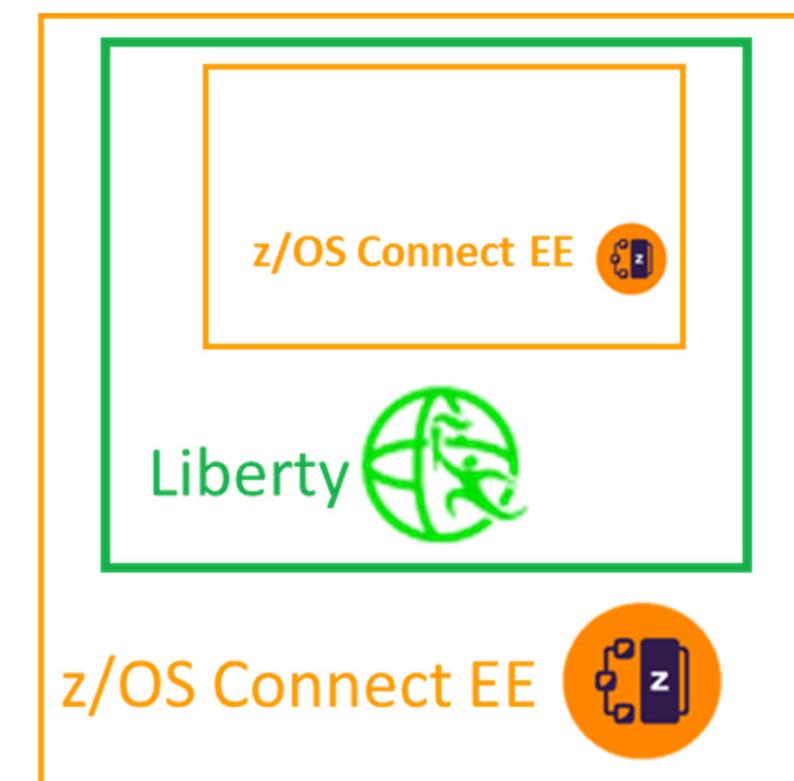
Tech-Tip: Sample JCL - Restarting the Java Health Center collection

SDSF PROCESS DISPLAY MPZ3		ALL		LINE 1-5 (5)								
				SCROLL ===> CSR								
NP	JOBNAME	Status	Owner	State	CPU-Time	PID	PPID	ASID	ASIDX	LatchWaitPID	Command	
BAQSTRT	WAITING FOR CHILD	LIBSERV	LIBSERV	1W	40.01	69050	83955129	42	002A		/bin/sh /usr/lpp/IBM/zosconnect/v3r0/bin	
BAQSTRT	OTHER KERNEL WAIT	LIBSERV	LIBSERV	HK	40.01	16846267	69050	42	002A		/usr/lpp/java/J8.0_64/bin/java -javagen	
BAQZANGL	SWAPPED, RUNNING	LIBANGE	LIBANGE	1RI	0.01	50399398	83953829	77	004D		/usr/lpp/IBM/zosconnect/v3r0/wplib/nat	
BAQZANGL	SWAPPED, FILE SYS KERNEL WAIT	LIBANGE	LIBANGE	1FI	0.01	83953829	1	77	004D		BPXBATA2	
BAQSTRT	FILE SYS KERNEL WAIT	LIBSERV	LIBSERV	1F	40.01	83955129	1	42	002A		BPXBATSL	

```
*****
product = WAS FOR Z/OS 21.0.0.9, z/OS Connect 03.00.52 (wlp-1.0.56.cl210920210909-1618)
wlp.install.dir = /shared/IBM/zosconnect/v3r0/wlp/
server.config.dir = /var/zosconnect/servers/myServer/
java.home = /shared/java/J8.0_64
java.version = 1.8.0_301
java.runtime = Java(TM) SE Runtime Environment (8.0.6.36 - pmz6480sr6fp36-20210913_01(SR6 FP36))
os = z/OS (02.03.00; s390x) (en_US)
process = 16846267@wg31
*****
```

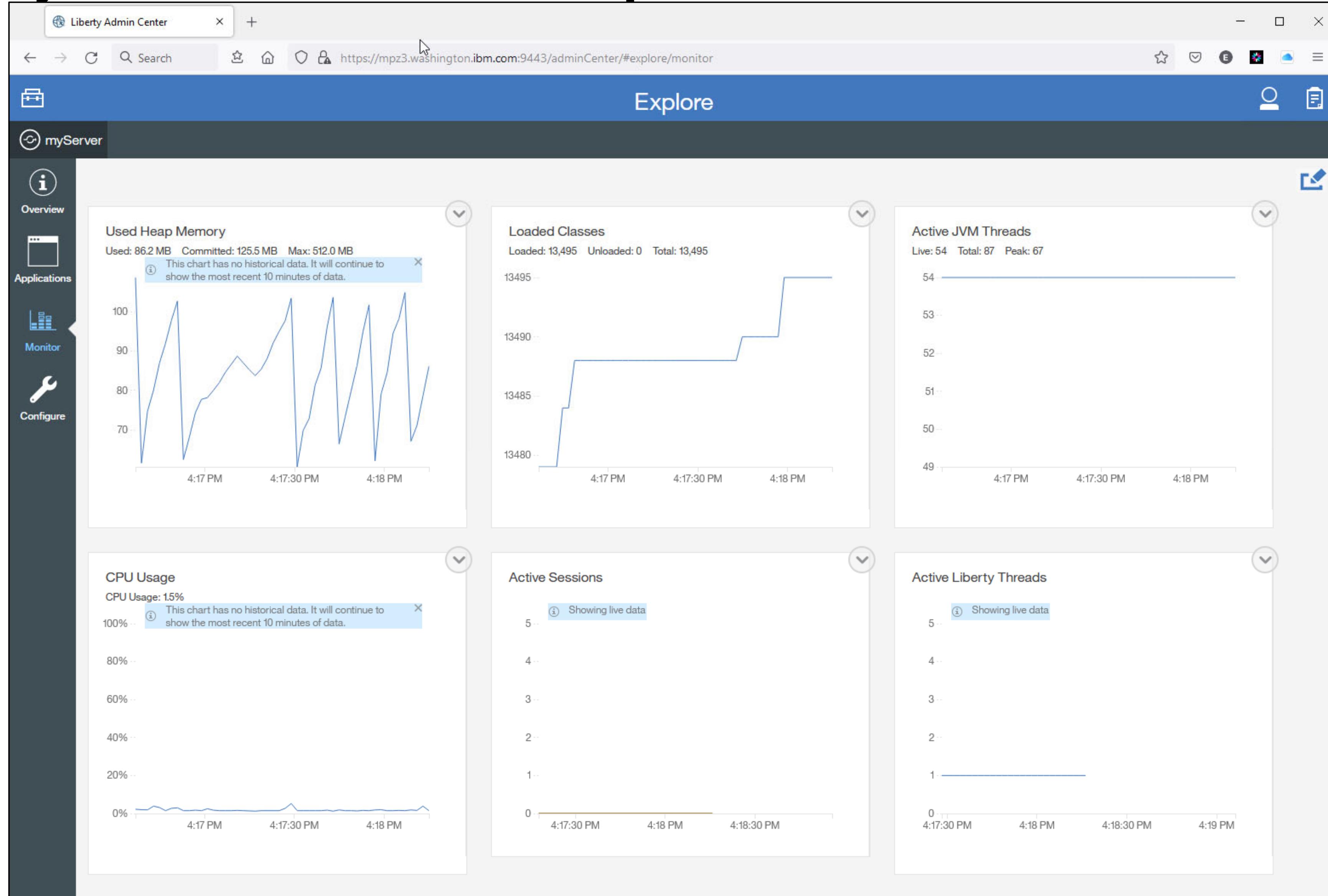
```
//JOHNONS JOB (ACCOUNT),NOTIFY=&SYSUID,REGION=0M,
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),USER=LIBSERV
//JAVA      EXEC PGM=IKJEFT01,REGION=0M
//SYSERR    DD   SYSOUT=*
//STDOUT     DD   SYSOUT=*
//SYSTSPRT  DD   SYSOUT=*
//SYSTSIN    DD   *
BPXBATCH SH +
java -jar /usr/lpp/java/J8.0_64/lib/ext/healthcenter.jar +
ID=16846267 level=headless +
-Dcom.ibm.java.diagnostics.healthcenter.headless.run.number.of.runs=1
```

The job must be executed under the same identity under which the server is running.





Liberty Admin Center feature provides real time monitoring



mitchj@us.ibm.com

Workload Manager - Definitions

WLM Report Classes

mpz3

Report-Class View Notes Options Help

Report Class Selection List Row 1 to 12 of 12

Command ==> _____

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete, /=Menu Bar

-- Last Change --

Action	Name	Description	User	Date
BAQSTC			JOHNSON	2021/09/04
WMQFTE			JOHNSON	2011/08/31
WMQFTER			JOHNSON	2011/08/31
WMQFTEZ			JOHNSON	2011/08/31
ZCEEADM			JOHNSON	2021/08/02
ZCEEAPIR			JOHNSON	2021/08/05
ZEECICS			JOHNSON	2021/08/05
ZCEEDB2			JOHNSON	2021/08/05
ZCEEIMS			JOHNSON	2021/08/05
ZCEEMQ			JOHNSON	2021/08/05
ZCEEOTHR			JOHNSON	2021/08/02
ZCEESTC			JOHNSON	2021/09/02

***** Bottom of data *****

10/004

Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23

WLM Service Classes

mpz3

Service-Class Xref Notes Options Help

Modify a Service Class Row 1 to 2 of 2

Command ==> _____

Service Class Name : OPS_HIGH

Description System Tasks Velocity 70

Workload Name STC_WKL (name or ?)

Base Resource Group (name or ?)

Cpu Critical NO (YES or NO)

I/O Priority Group NORMAL (NORMAL or HIGH)

Honor Priority DEFAULT (DEFAULT or NO)

Specify BASE GOAL information. Action Codes: I=Insert new period, E>Edit period, D=Delete period.

-- Period -- Goal --

Action	#	Duration	Imp.	Description
	1		1	Execution velocity of 70

***** Bottom of data *****

19/004

Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23

mitchj@us.ibm.com

WLM "CB" Classification Rules

mpz3

Subsystem-Type Xref Notes Options Help

Modify Rules for the Subsystem Type Row 1 to 8 of 16

Command ==> _____

Subsystem Type . . . CB Fold qualifier names? N (Y or N)

Description WLP/zCEE Transactions

Action codes: A=After C=Copy M=Move I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
More ==>

Action	Type	Name	Start
1	CN	myServer	_____
2	TC	TCADM	_____
2	TC	TCAPIR	_____
2	TC	TCCICS	_____
2	TC	TCDB2	_____
2	TC	TCIMS	_____
2	TC	TCMQ	_____
2	TC	TCOTHR	_____

-----Class-----

Service	Report
OPS_HIGH	ZCEEOTHR
OPS_HIGH	BAQSTC
OPS_HIGH	ZCEEADM
OPS_HIGH	ZCEEAPIR
OPS_HIGH	ZEECICS
OPS_HIGH	ZCEEDB2
OPS_HILO	ZCEEIMS
OPS_MED	ZCEEMQ
OPS_LOW	ZCEEOTHR

07/021

mpz3

File Edit Settings View Communication Actions Window Help

Subsystem-Type Xref Notes Options Help

Modify Rules for the Subsystem Type Row 9 to 16 of 16

Command ==> _____

Subsystem Type . . . CB Fold qualifier names? N (Y or N)

Description WLP/zCEE Transactions

Action codes: A=After C=Copy M=Move I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
More ==>

Action	Type	Name	Start
1	CN	zceex	_____
2	TC	TCADM	_____
2	TC	TCAPIR	_____
2	TC	TCDB2	_____
2	TC	TCCICS	_____
2	TC	TCIMS	_____
2	TC	TCMQ	_____
2	TC	TCOTHR	_____

-----Class-----

Service	Report
OPS_HIGH	ZCEEOTHR
OPS_HIGH	ZCEESTC
OPS_HIGH	ZCEEADM
OPS_HIGH	ZCEEAPIR
OPS_HIGH	ZEECICS
OPS_HILO	ZCEEDB2
OPS_HILO	ZCEEIMS
OPS_MED	ZCEEMQ
OPS_HILO	ZCEEOTHR

07/021

mpz3

File Edit Settings View Communication Actions Window Help

Subsystem-Type Xref Notes Options Help

Modify Rules for the Subsystem Type Row 9 to 16 of 16

Command ==> _____

Subsystem Type . . . CB Fold qualifier names? N (Y or N)

Description WLP/zCEE Transactions

Action codes: A=After C=Copy M=Move I=Insert rule
B=Before D=Delete row R=Repeat IS=Insert Sub-rule
More ==>

Action	Type	Name	Start
1	CN	zceex	_____
2	TC	TCADM	_____
2	TC	TCAPIR	_____
2	TC	TCDB2	_____
2	TC	TCCICS	_____
2	TC	TCIMS	_____
2	TC	TCMQ	_____
2	TC	TCOTHR	_____

-----Class-----

Service	Report
OPS_HIGH	ZCEEOTHR
OPS_HIGH	ZCEESTC
OPS_HIGH	ZCEEADM
OPS_HIGH	ZCEEAPIR
OPS_HIGH	ZEECICS
OPS_HILO	ZCEEDB2
OPS_HILO	ZCEEIMS
OPS_MED	ZCEEMQ
OPS_HILO	ZCEEOTHR

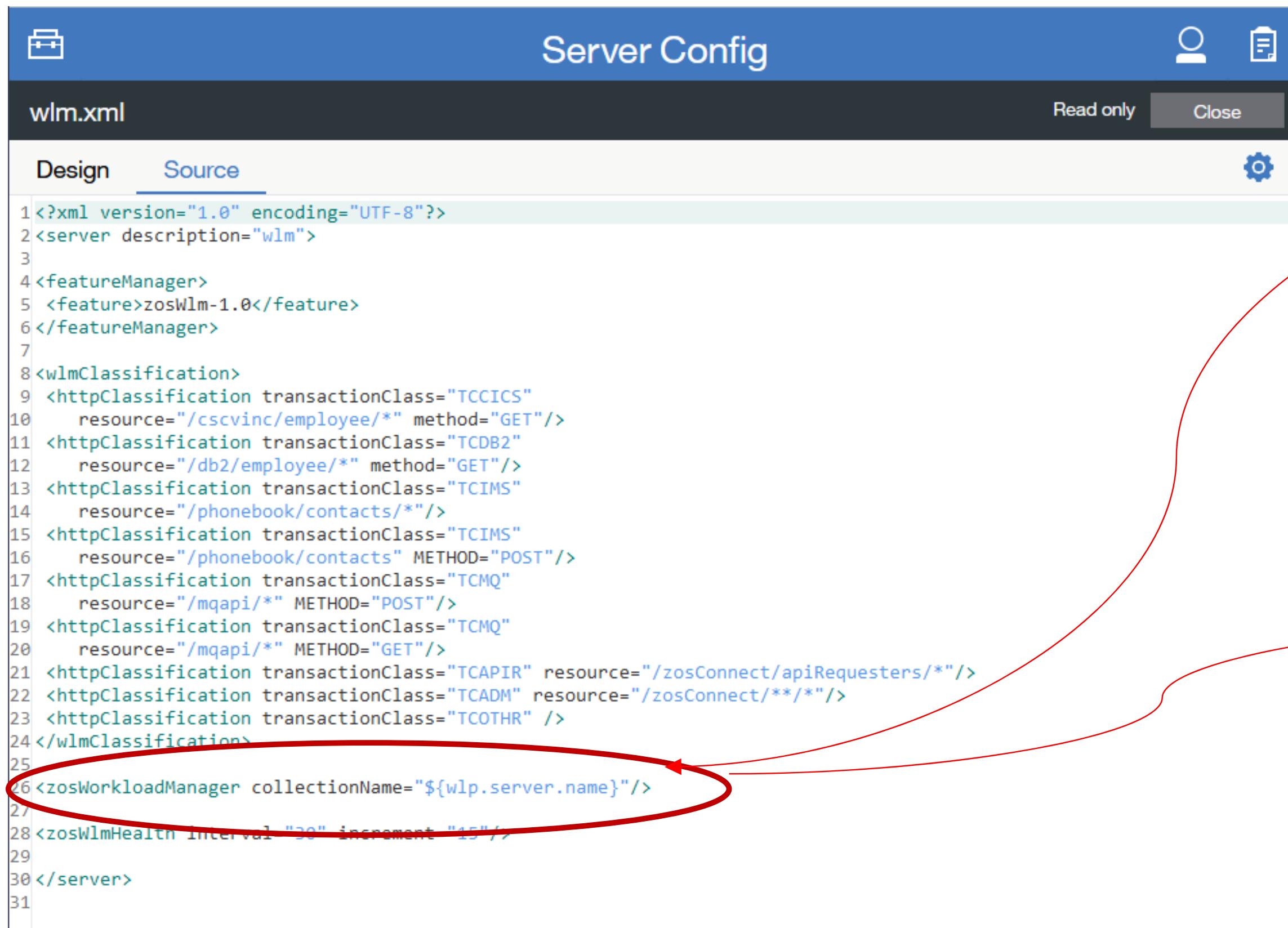
07/021

Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23

Workload Manager – WLM Classification server XML

The corresponding required server XML configuration

- Based on HTTP path matching (port and/or method can also be specified)
- The default value for the *wlmClassification* name is the name of the server
- See URL <https://www.ibm.com/docs/en/was-liberty/zos?topic=zos-wlm-classification> for more information
- The *transactionClass* attribute is required to ensure an enclave is created.

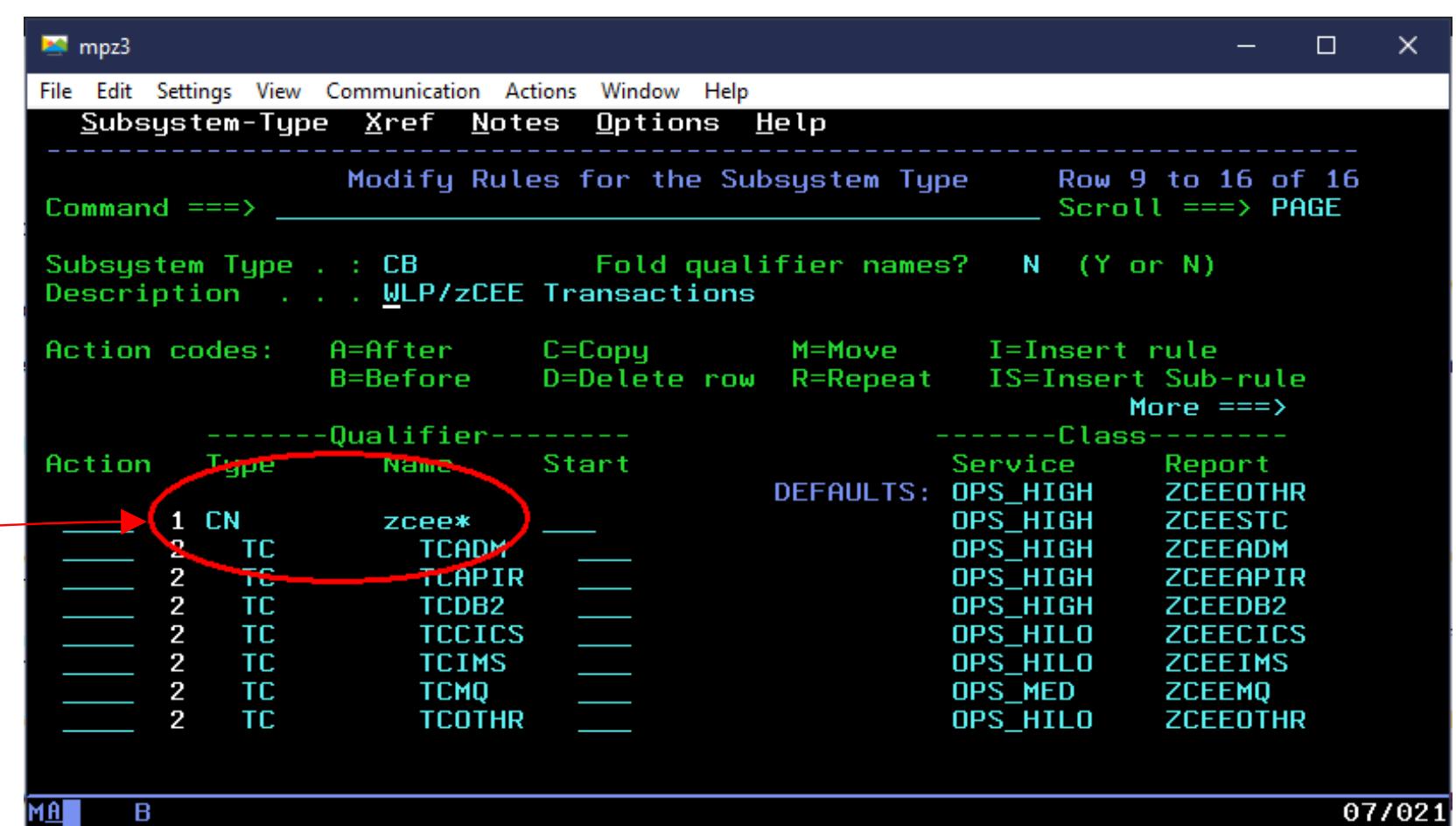


```

<?xml version="1.0" encoding="UTF-8"?>
<server description="wlm">
  <featureManager>
    <feature>zosWlm-1.0</feature>
  </featureManager>
  <wlmClassification>
    <httpClassification transactionClass="TCCICS"
      resource="/cscvinc/employee/*" method="GET"/>
    <httpClassification transactionClass="TCDB2"
      resource="/db2/employee/*" method="GET"/>
    <httpClassification transactionClass="TCIMS"
      resource="/phonebook/contacts/*"/>
    <httpClassification transactionClass="TCIMS"
      resource="/phonebook/contacts" METHOD="POST"/>
    <httpClassification transactionClass="TCMQ"
      resource="/mqapi/*" METHOD="POST"/>
    <httpClassification transactionClass="TCMQ"
      resource="/mqapi/*" METHOD="GET"/>
    <httpClassification transactionClass="TCAPIR" resource="/zosConnect/apiRequesters/*"/>
    <httpClassification transactionClass="TCADM" resource="/zosConnect/**/*"/>
    <httpClassification transactionClass="TCOTHR" />
  </wlmClassification>
  <zosWorkloadManager collectionName="${wlp.server.name}"/>
  <zosWlmHealth interval "30" increment "15"/>
</server>

```

Related to WLM CN name.



Action	Type	Name	Start	Service	Report
1	CN	zceex*		OPS_HIGH	ZCEEOTHR
2	TC	TCADM		OPS_HIGH	ZCEEESTC
2	TC	TCDB2		OPS_HIGH	ZCEEADM
2	TC	TCCICS		OPS_HILO	ZCEEAPIR
2	TC	TCIMS		OPS_HILO	ZCEEIMS
2	TC	TCMQ		OPS_MED	ZCEEMQ
2	TC	TCOTHR		OPS_HILO	ZCEEOTHR



Workload Manager – Active HTTP Classification

https://mpz3.washington.ibm.com:9443/ibm/api/config/httpClassification

The screenshot shows a web browser window displaying a JSON array of configuration elements for "httpClassification". Each element is defined by the following fields:

- configElementName: "httpClassification"
- uid: "wlmClassification[default-0]/httpClassification[default-4]", "wlmClassification[default-0]/httpClassification[default-5]", "wlmClassification[default-0]/httpClassification[default-6]", "wlmClassification[default-0]/httpClassification[default-7]", or "wlmClassification[default-0]/httpClassification[default-8]"
- host: "*" (wildcard)
- method: "POST", "GET", "*", or "*/*"
- port: "*" (wildcard)
- resource: "/mqapi/*", "/zosConnect/apiRequesters/*", "/zosConnect/**/*", or "/*"
- transactionClass: "TCMQ", "TCAPIR", "TCADM", or "TCOTHR"

```
[{"configElementName": "httpClassification", "uid": "wlmClassification[default-0]/httpClassification[default-4]", "host": "*", "method": "POST", "port": "*", "resource": "/mqapi/*", "transactionClass": "TCMQ"}, {"configElementName": "httpClassification", "uid": "wlmClassification[default-0]/httpClassification[default-5]", "host": "*", "method": "GET", "port": "*", "resource": "/mqapi/*", "transactionClass": "TCMQ"}, {"configElementName": "httpClassification", "uid": "wlmClassification[default-0]/httpClassification[default-6]", "host": "*", "method": "*", "port": "*", "resource": "/zosConnect/apiRequesters/*", "transactionClass": "TCAPIR"}, {"configElementName": "httpClassification", "uid": "wlmClassification[default-0]/httpClassification[default-7]", "host": "*", "method": "*", "port": "*", "resource": "/zosConnect/**/*", "transactionClass": "TCADM"}, {"configElementName": "httpClassification", "uid": "wlmClassification[default-0]/httpClassification[default-8]", "host": "*", "method": "*", "port": "*", "resource": "/*", "transactionClass": "TCOTHR"}]
```

RMF SMF Type 72 Service Class Reports

mpz3

File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help

SDSF OUTPUT DISPLAY JOHNSONR JOB12740 DSID 112 LINE CHARS 'CICS' FOUND
COMMAND INPUT ===>
POLICY=WSCPOL

REPORT CLAS

- TRANSACTIONS --		TRANS-TIME	HHH.MM.SS.FFFFFFF	TRA
Avg	0.02	ACTUAL	108891	TOT
MPL	0.02	EXECUTION	108856	MOB
ENDED	96	QUEUED	34	CAT
END/S	0.16	R/S AFFIN	0	CAT
#SWAPS	0	INELIGIBLE	0	
EXCTD	0	CONVERSION	0	
		STD DEV	762583	
----SERVICE----	SERVICE TIME	--APPL %--	--P	
IOC	0 CPU	1.967 CP	0.02 AAP	BLK
CPU	1739K SRB	0.000 IIPCP	0.02 ENQ	
MSO	0 RCT	0.000 IIP	0.31 CRM	
SRB	0 IIT	0.000 AAPCP	0.00 LCK	
TOT	1739K HST	0.000 AAP	N/A SUP	
/SEC	2898 IIP	1.844		
ABSRPTN	166K AAP	N/A		
TRX SERV	166K			

MA A
Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23

mpz3

File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help

SDSF OUTPUT DISPLAY JOHNSONR JOB12740 DSID 112 LINE CHARS 'APIR' FOUND
COMMAND INPUT ===>
SCROLL ==> PAGE
POLICY=WSCPOL

REPORT CLASS=ZCEEAPIR PERIOD=1

- TRANSACTIONS --		TRANS-TIME	HHH.MM.SS.FFFFFFF	TRANS-APPL%-----CP-IIPCP/AAPCP-IIP/AAP	---ENCLAVES---	
Avg	0.14	ACTUAL	424835	TOTAL 0.12	0.12 0.73	
MPL	0.14	EXECUTION	424707	MOBILE 0.00	0.00 0.00	
ENDED	200	QUEUED	126	CATEGORYA 0.00	0.00 0.00	
END/S	0.33	R/S AFFIN	0	CATEGORYB 0.00	0.00 0.00	
#SWAPS	0	INELIGIBLE	0			
EXCTD	0	CONVERSION	0			
		STD DEV	1.381943			
----SERVICE----	SERVICE TIME	--APPL %--	--PROMOTED--	--DASD I/O--	----STORAGE----	-PAGE-IN RATES-
IOC	0 CPU	5.073 CP	0.12	BLK 0.000	SSCHRT 2.4	AVG 0.00
CPU	4485K SRB	0.000 IIPCP	0.12	ENQ 0.000	RESP 0.4	TOTAL 0.00
MSO	0 RCT	0.000 IIP	0.73	CRM 0.000	CONN 0.3	SHARED 0.00
SRB	0 IIT	0.000 AAPCP	0.00	LCK 0.000	DISC 0.0	
TOT	4485K HST	0.000 AAP	N/A SUP	0.000 Q+PEND	0.0	
/SEC	7474 IIP	4.363		IOSQ 0.0		
ABSRPTN	53K AAP	N/A				
TRX SERV	53K					

MA A
Connected to remote server/host mpz3 using lu/pool MPZ30008 and port 23



Liberty SMF 120 Subtype 11

WebSphere Liberty Profile (WLP) can generate various types of SMF 120 records. Support for a SMF 120 record relevant for z/OS Connect was added in WLP V16.0.0.2. This record, a SMF 120 Subtype 11, is generated for each HTTP request received by the Liberty server. For more details and a description of the contents of this record, see URL <https://www.ibm.com/support/pages/liberty-zos-smf-120-11-version-2>



The screenshot shows the 'Server Config' interface with a blue header bar containing icons for a briefcase, user profile, and save. The title 'Server Config' is in the center. Below the header, a dark navigation bar has 'smf.xml' on the left and 'Read only' and 'Close' buttons on the right. A gear icon is in the top right corner of the main content area. The main area contains two tabs: 'Design' (selected) and 'Source'. The 'Source' tab displays the XML configuration code:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <server description="SMF">
4   <featureManager>
5     <feature>monitor-1.0</feature>
6     <feature>zosRequestLogging-1.0</feature>
7   </featureManager>
8
9 </server>
10
```

Useful Plug-ins for WAS z/OS SMF 120.9 Browser

<https://www.ibm.com/support/pages/node/6355403>



Liberty SMF 120 Subtype 11 – WP102312 Plugin

AutoSave (Off) Search Mitch Johnson MJ

File Home Insert Page Layout Formulas Data Review View Help ACROBAT

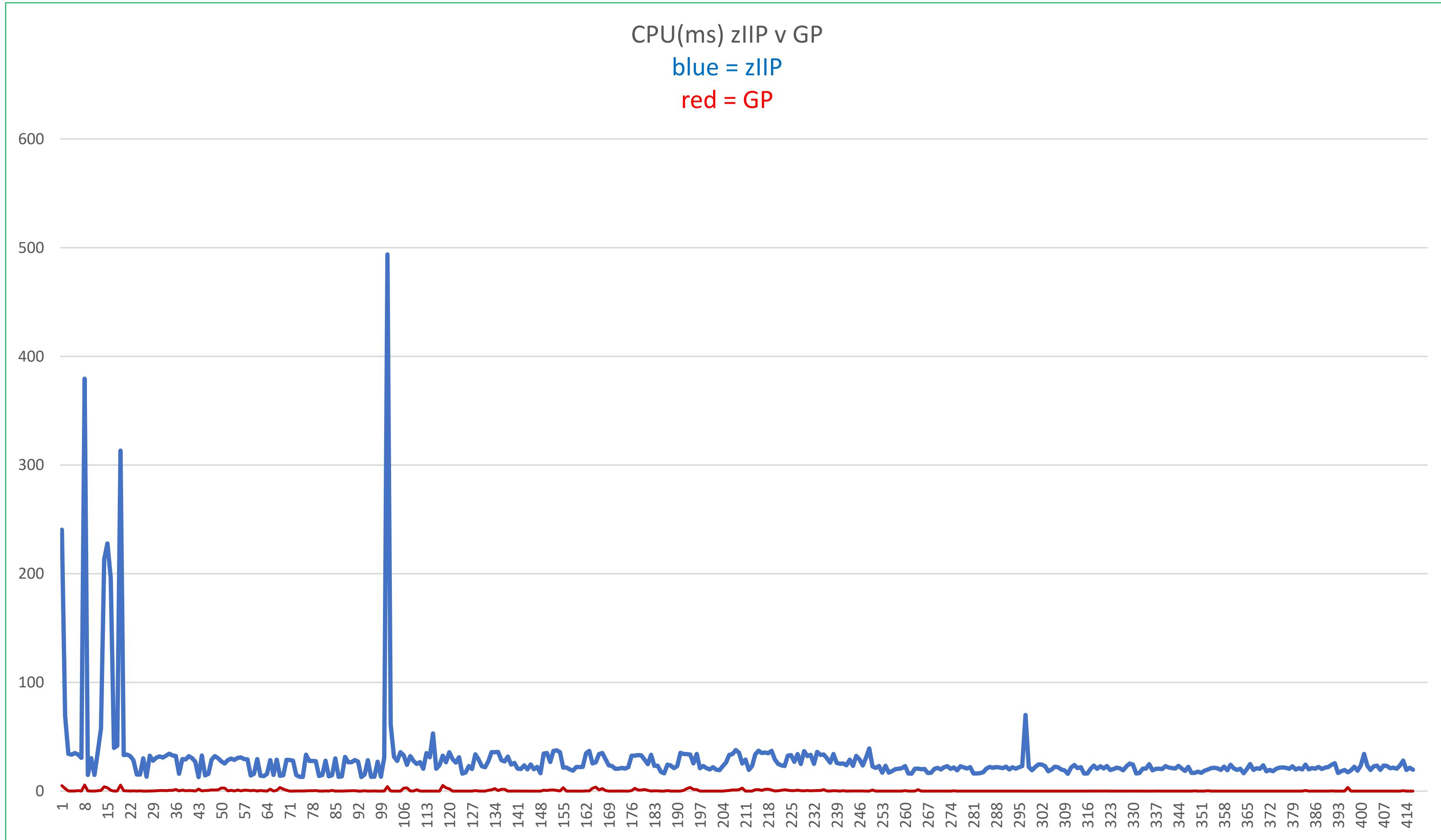
Cut Copy Format Painter Paste Font Alignment Number Styles Cells Editing Ideas Sensitivity

AS9 166

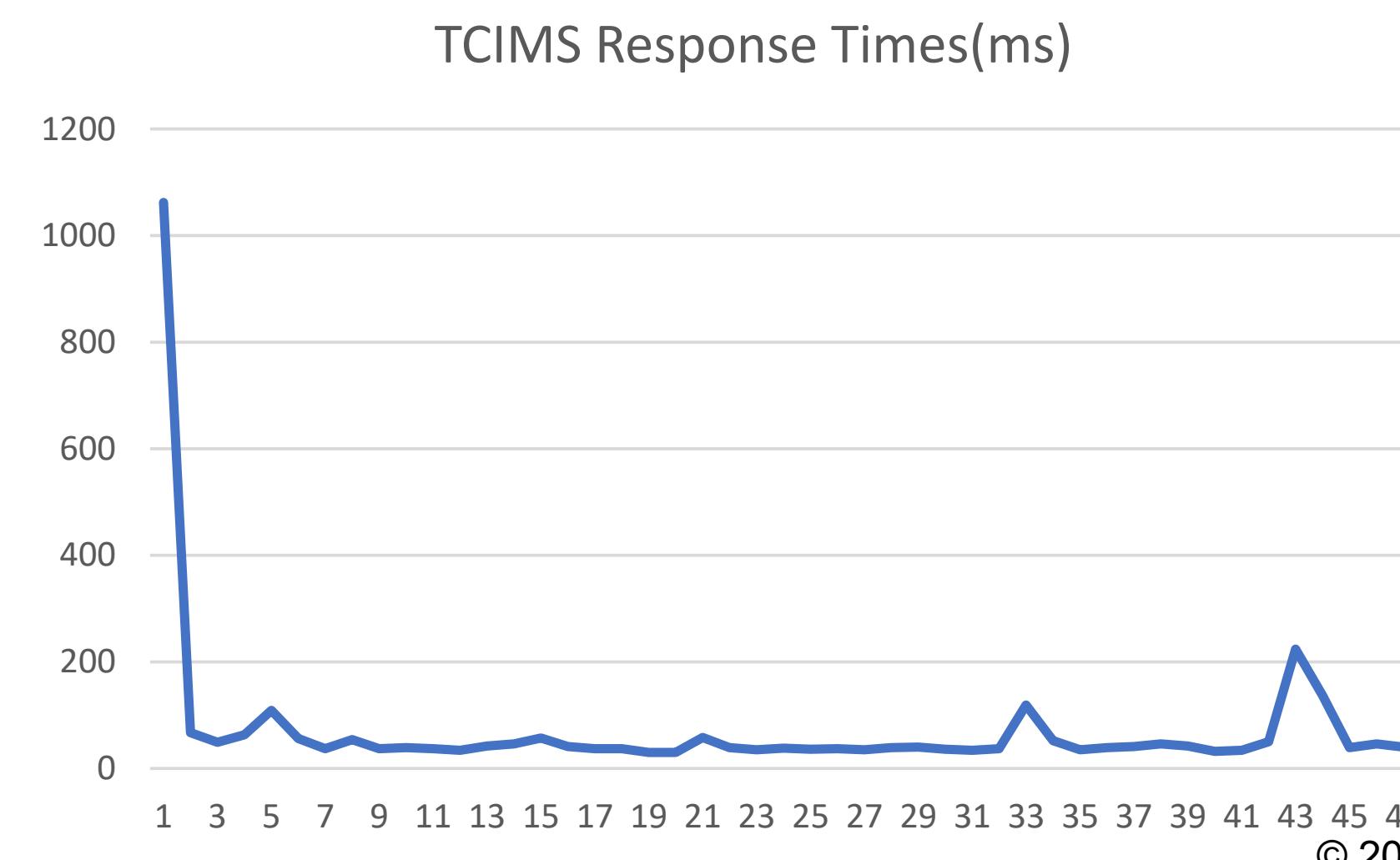
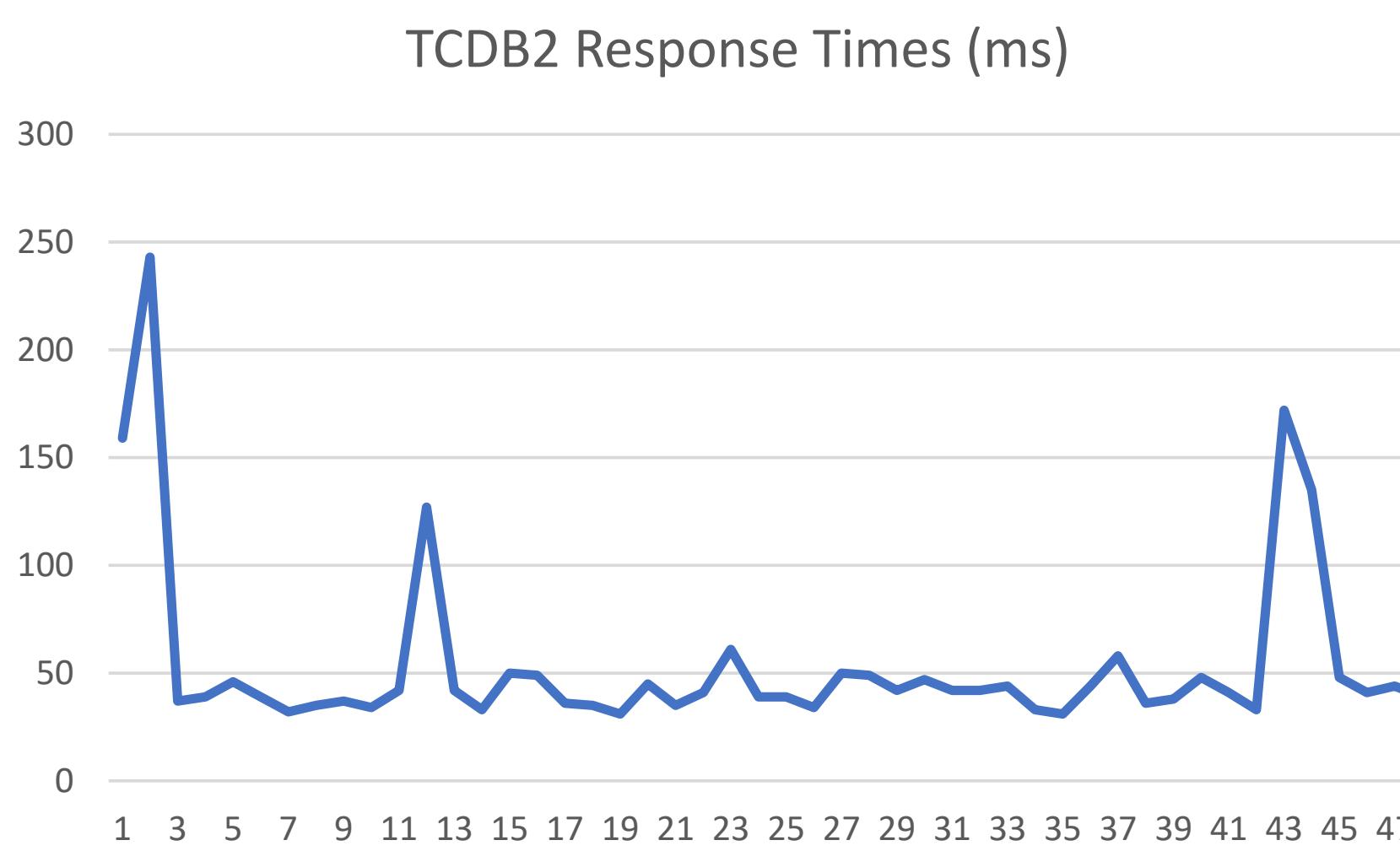
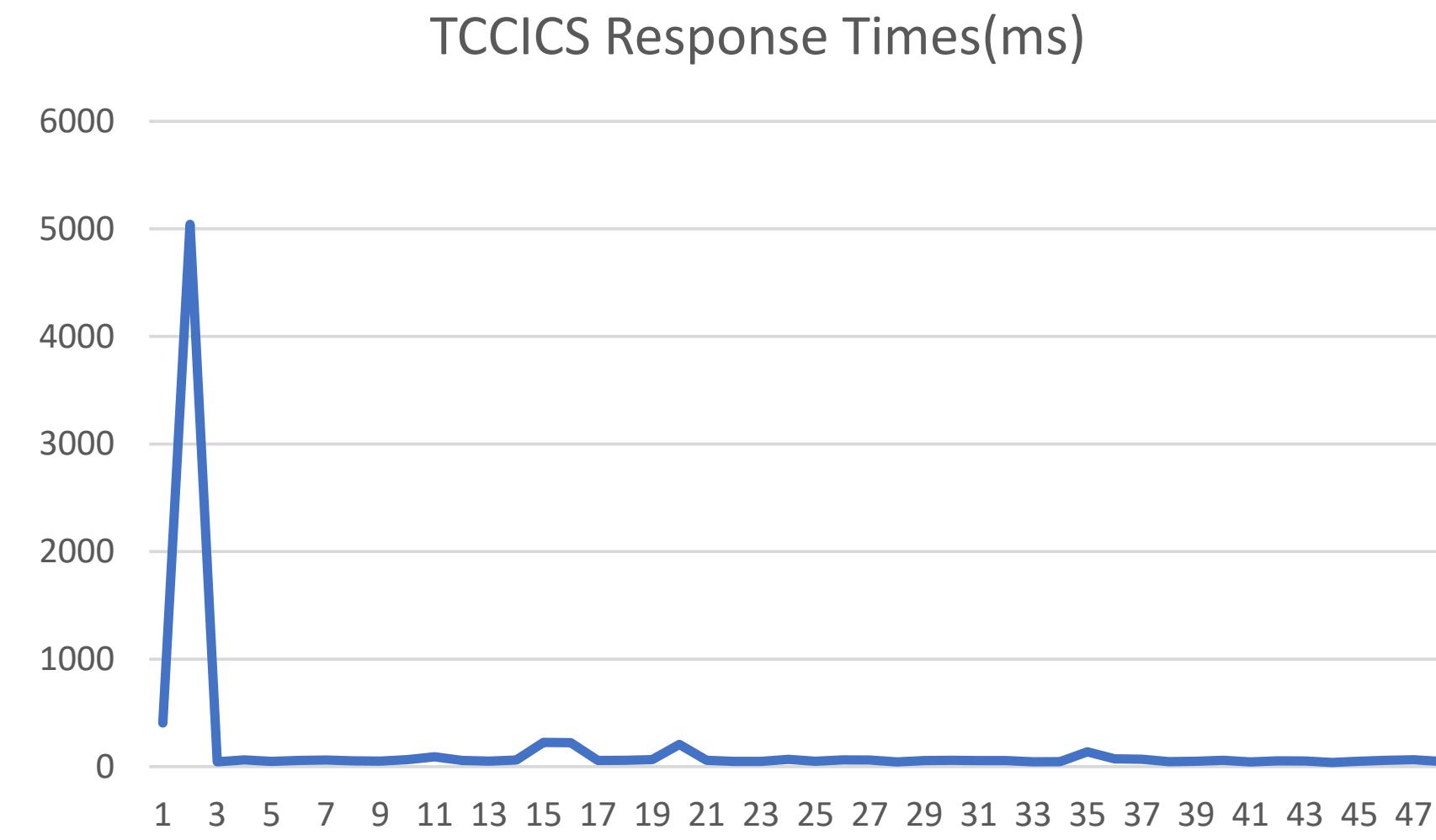
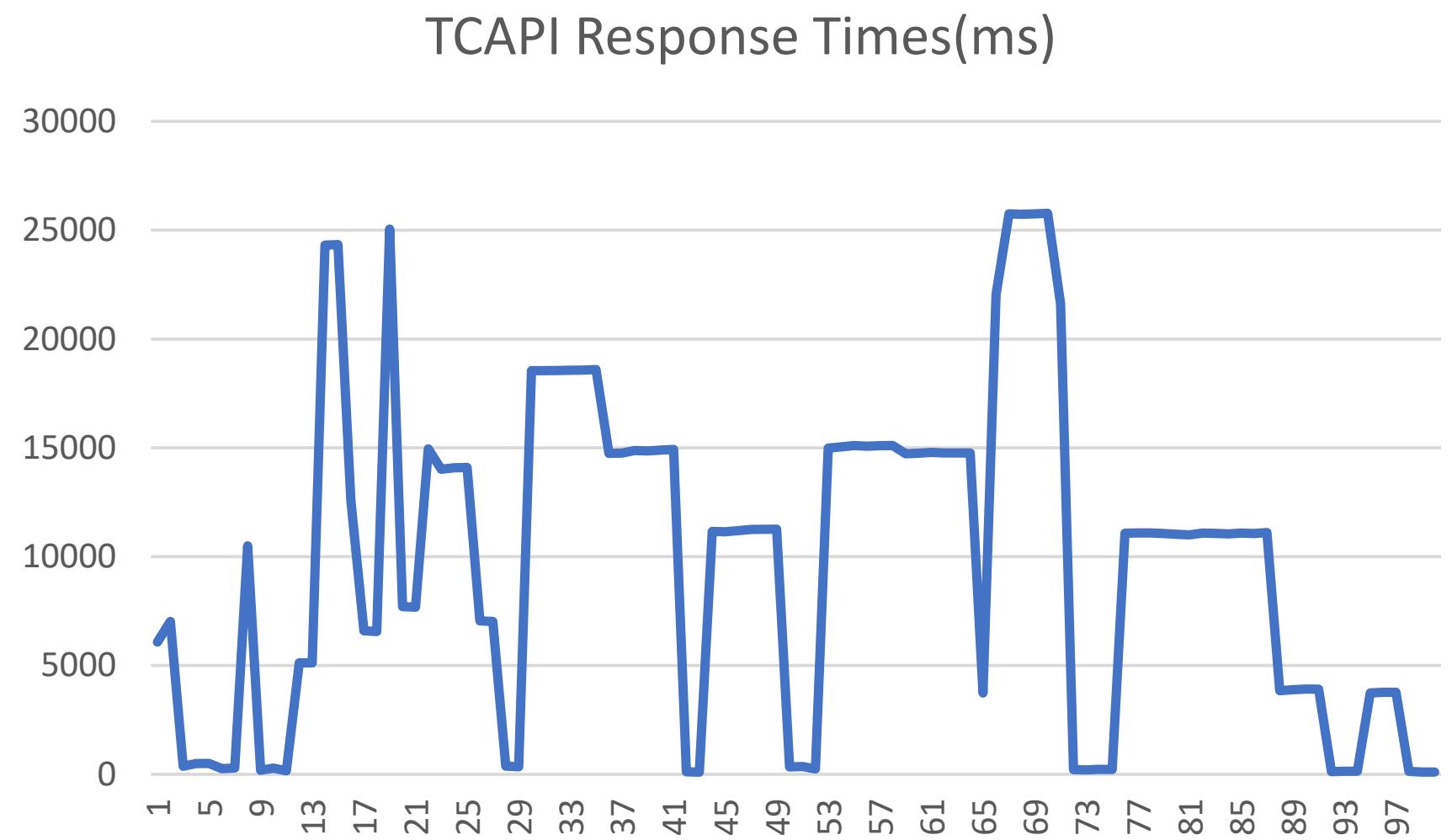
B	C	E	P	Q	R	S	T	U	V	W	Z	AA	AB	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	
1	SystemName	SysplexName	JobName	StartTime	StartTime	EndTime	EndTime	Respo	TranClass	TotalCPUStart	TotalCPUEnd	TotalCPU(ms)	TotalGP(ms)	TotalOffload(ms)	userid	mappedUser	requestUser	host	port	uri	response	targetPort	remotePort	RemoteAddr	
2	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	6080	TCAPIR	3314772936	4.32E+09	245.5195	5.0110927	240.508381	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4283	192.168.17.243
3	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7030	TCAPIR	178821759	471750165	71.51572	2.334169	69.18156	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4286	192.168.17.243
4	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	374	TCAPIR	4327455460	4.469E+09	34.44008	0.10757129	34.332504	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4301	192.168.17.243
5	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	495	TCAPIR	2762287407	2.9E+09	33.65053	0.057430662	33.5931	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4304	192.168.17.243
6	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	500	TCAPIR	4484655211	4.629E+09	35.15451	0.12540185	35.020004	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4303	192.168.17.243
7	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	262	TCAPIR	4637789017	4.777E+09	34.10823	0.42818993	33.680042	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4305	192.168.17.243
8	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	293	TCAPIR	542458283	668050357	30.66213	0.053870115	30.608257	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4306	192.168.17.243
9	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	10493	TCAPIR	3802597962	5.38E+09	385.0374	5.576215	379.46115	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4285	192.168.17.243
10	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	185	TCAPIR	5384541333	5.446E+09	15.04486	0.15656103	14.888303	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4308	192.168.17.243
11	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	282	TCAPIR	1028119195	1.153E+09	30.38298	0.04661279	30.336363	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4309	192.168.17.243
12	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	163	TCAPIR	901260513	962209631	14.88016	0	14.8801565	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4310	192.168.17.243
13	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	5126	TCAPIR	3137255105	3.284E+09	35.92899	0.33009765	35.598892	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4313	192.168.17.243
14	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	5122	TCAPIR	4890213483	5.128E+09	58.01673	0.61064285	57.40609	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4314	192.168.17.243
15	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	24315	TCAPIR	13036032356	1.393E+10	217.4406	4.0119	213.4287	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4282	192.168.17.243
16	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	24338	TCAPIR	1463812131	2.41E+09	230.9845	3.1036336	227.8809	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4284	192.168.17.243
17	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	12587	TCAPIR	1160912461	1.967E+09	196.8579	0.7669092	196.09096	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4315	192.168.17.243
18	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	6599	TCAPIR	5303866625	5.467E+09	39.78177	0.020269532	39.761494	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4316	192.168.17.243
19	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	6565	TCAPIR	6143860672	6.315E+09	41.86705	0.16208105	41.704967	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4317	192.168.17.243
20	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	25052	TCAPIR	2622790027	3.928E+09	318.7149	5.489483	313.22546	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4281	192.168.17.243
21	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7709	TCAPIR	4477460136	4.615E+09	33.52233	0.35891944	33.163406	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4322	192.168.17.243
22	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7682	TCAPIR	1973032107	2.112E+09	33.81701	0.19548193	33.621525	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4321	192.168.17.243
23	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	14950	TCAPIR	458083508	590213570	32.25832	0.0489917	32.209324	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4323	192.168.17.243
24	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	14016	TCAPIR	61401222	178390269	28.56178	0.2347461	28.327032	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4325	192.168.17.243
25	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	14088	TCAPIR	86069826	148846164	15.32625	0.0541626	15.272091	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4326	192.168.17.243
26	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	14097	TCAPIR	5471350509	5.535E+09	15.43587	0.21740967	15.218459	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4324	192.168.17.243
27	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7051	TCAPIR	5358173556	5.482E+09	30.16547	0.001757324	30.163715	USER1	/zosConn	mpz3.was	9080	/zosConnect/apiRequeste	166	9080	4328	192.168.17.243
28	MPZ3	MPZPLEX	BAQSTRT	Friday	Au	3.84E+12	Friday	Au	3.84E+12	7029	TCAPIR	2281578411													



Liberty SMF 120 type 11 – GP v zIIP comparison example



Liberty SMF 120 type 11 – Response times comparisons example



z/OS Connect SMF 123 server XML configuration (OpenAPI 2)



SMF 123 records have two subtypes, and each subtype can have different versions.

- SMF type 123 subtype 1 records - Version 1 contains some basic information about both API provider and API requester requests. Version 2 supersedes version 1 and contains more detailed information about each API provider request, including information about to which system of record (SOR) the request was sent
- *SMF type 123 subtype 2 records - Version 2 supersedes subtype 1 version 1 and contains more detailed information about each API requester request, including information about to what HTTP endpoint the request was sent.*

Server Config

audit.xml

Read only Close

Design Source

```
1<?xml version="1.0" encoding="UTF-8"?>
2<server description="SMF reporting">
3
4  <zosconnect_zosConnectManager
5      globalInterceptorsRef="interceptorList_g"/>
6
7  <zosconnect_authorizationInterceptor id="auth"
8      safCacheTimeout="600"/>
9
10 <zosconnect_auditInterceptor id="audit"
11     apiRequesterSmfVersion="2"
12     apiProviderSmfVersion="2"/>
13
14 <zosconnect_zosConnectInterceptors id="interceptorList_g"
15     interceptorRef="audit"/>
16
17</server>
18
```

Server Config

audit.xml

Read only Close

Design Source

Server

- z/OS Connect Manager
- z/OS Connect Authorization Interceptor auth
- z/OS Connect EE SMF Audit Interceptor audit

z/OS Connect Interceptors interceptorList_g

Sequence
0 (default)
The sequence in which this interceptor should be processed with respect to other configured interceptors implementing z/OS Connect's com.ibm.wsspi.zos.connect.Interceptor Service Provider Interface (SPI).

API provider SMF Version
2
The version of SMF 123 subtype 1 records to be written.

auditApiProviderRequestHeaders.name
(no value)
auditApiProviderRequestHeaders.desc

auditApiProviderResponseHeaders.name
(no value)
auditApiProviderResponseHeaders.desc

API requester SMF Version
2
The version of SMF 123 subtype 1 or subtype 2 records to be written.

z/OS Connect SMF 123 subtype 1 version 2 (OpenAPI 2) *



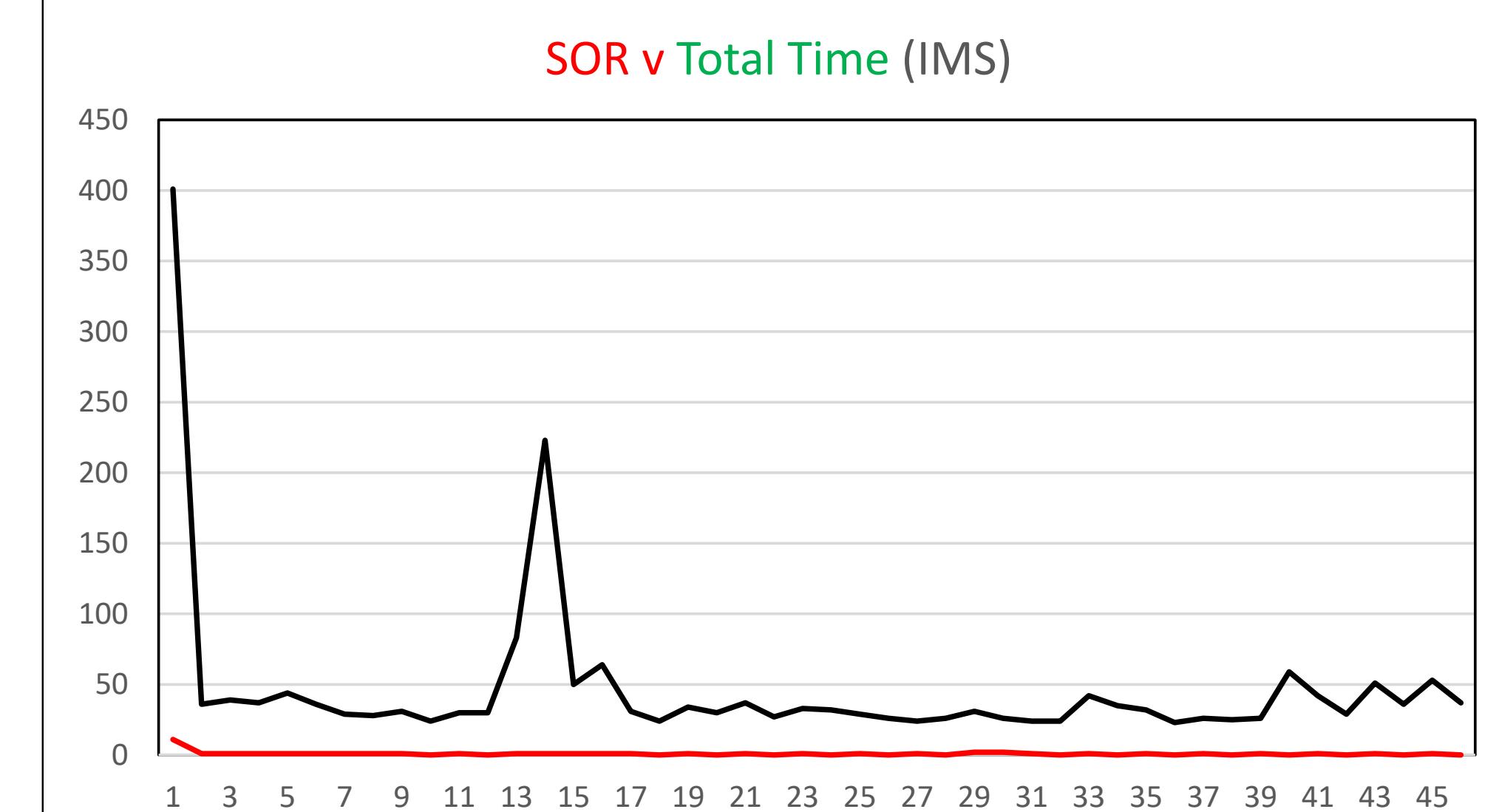
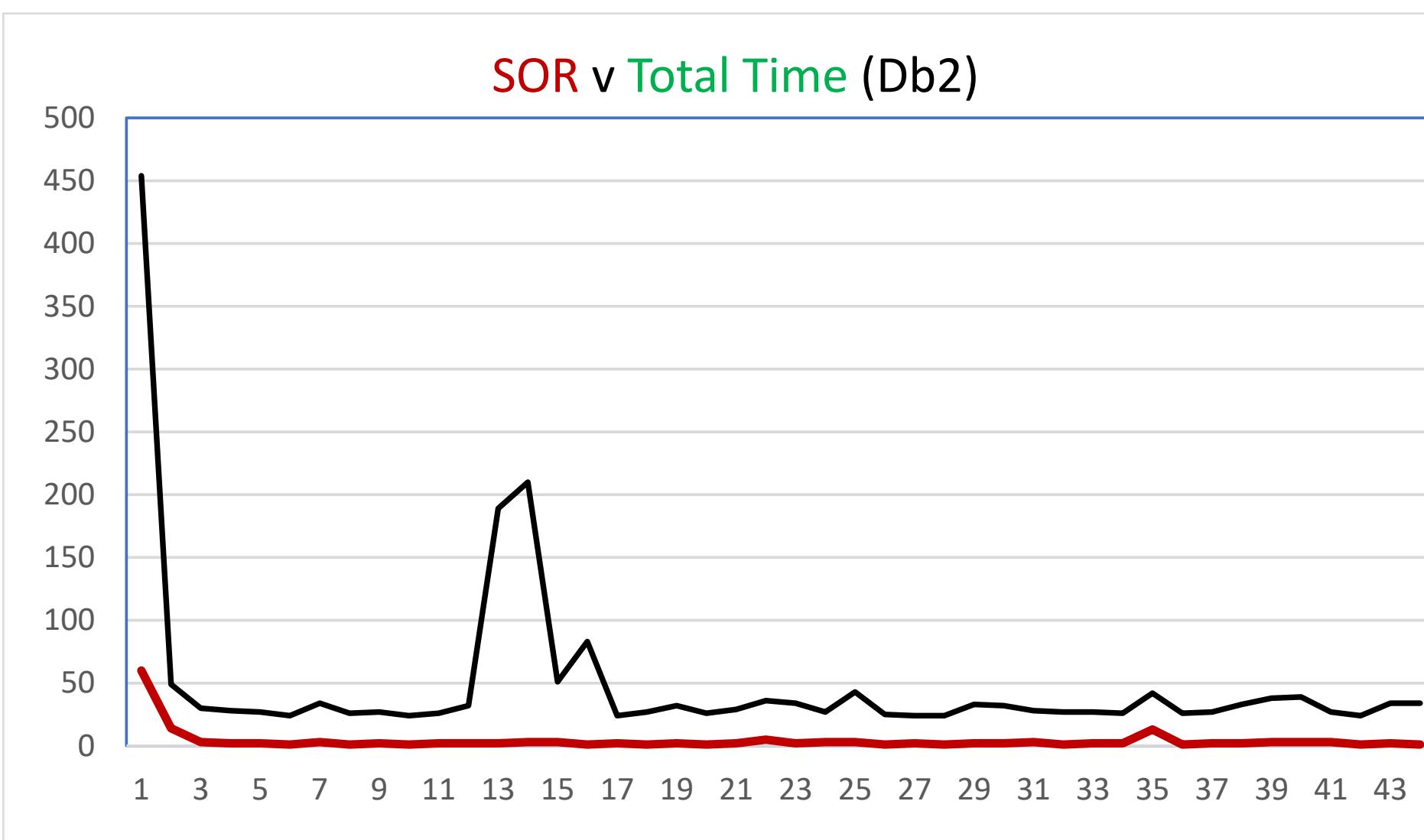
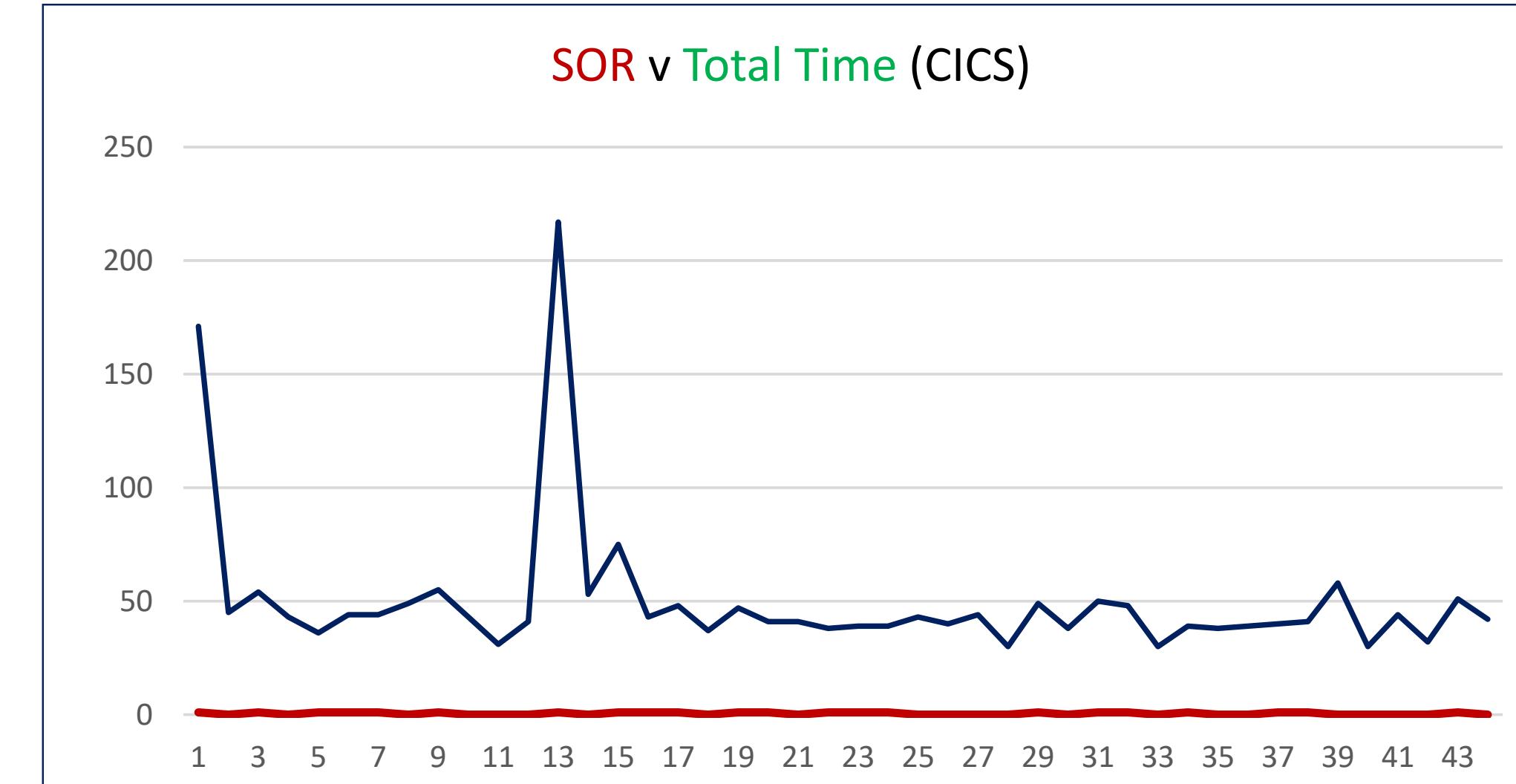
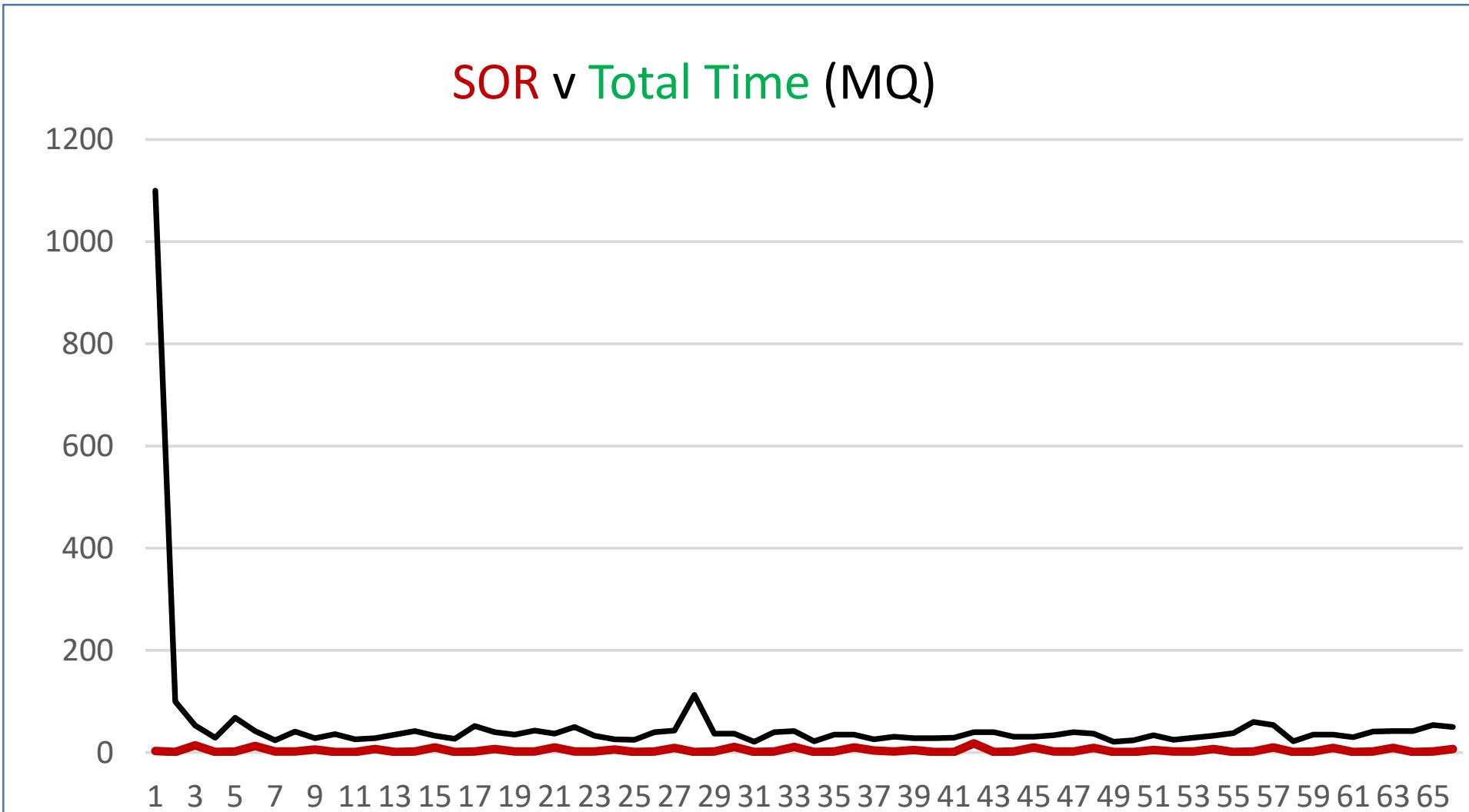
mitchj@us.ibm.com

Some fields have been hidden

* Generated by using a modified version of the BAQSMFX sample program.

© 2017, 2024 IBM Corporation
Slide 217

z/OS Connect SMF 123 subtype 1 version 2 graph examples (OpenAPI 2)



z/OS Connect SMF 123 subtype 2 version 2 (OpenAPI 2) *



AutoSave (Off) H Search smfout.csv Mitch Johnson MJ Share Comments

File Home Insert Page Layout Formulas Data Review View ACROBAT

Cut Copy Format Painter Paste Font Alignment Number Styles Cells Editing Ideas Sensitivity

Font: Calibri Size: 11 Bold Italic Underline Alignment: Wrap Text Number: General Conditional Formatting: Neutral

Format as Table: Normal, Bad, Good, Neutral, Calculation, Check Cell

Cells: Insert, Delete, Format, AutoSum, Fill, Clear

Editing: Sort & Filter, Find & Select, Ideas, Sensitivity

AP31 : 2021/08/23 18:16:02.725340 UTC

	A	B	C	D	U	V	W	X	Y	Z	AA	AI	AJ	AK	AL	AM	AAC	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY
27	SMF123_RSMF123_S	SMF123_SUBTYPE_VERSION																									
28	123	2	2																								
29																											
30	SID	SSI	TRIPLET_C	TRIPLET_C	HTTP_REQ_STAT	REQ_RET	REQ_PAYL	RESP_PAYL	USER_NA	USER_NA	ENDPOINT	ENDPOINT	TIME_ST	TIME_TII	TIME_AL	TIME_AM	TIME_ZCInbound	TIME_ENDPOI	StubTime	ZCInbound	TokenTim	EndPointTime	ZCOutbou	TotalTime(us)	TotalTime(s)	MVS_JOB M	
31	MPZ3	ZCON	2	40	200	200	NO	0	272	USER1	GET		2021/08/2021/02/202021/08/2318:	95384	108577	6734453	131423	25653	7103301					7.1023	USER1GE5 JC		
32	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	114313	7767	318	40583	2105	166270					0.1663	USER1GE5 JC		
33	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	112903	7193	130	51158	1905	175644					0.1756	USER1GE5 JC		
34	MPZ3	ZCON	2	40	200	200	NO	0	271	USER1	GET		2021/08/2021/02/202021/08/2318:	103999	102634	8843582	110850	3497	9166156					9.1662	USER1GE4 JC		
35	MPZ3	ZCON	2	40	200	200	NO	0	271	USER1	GET		2021/08/2021/02/202021/08/2318:	82840	4956	128	65685	1900	156097					0.1561	USER1GE4 JC		
36	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	116458	10778	288	58698	1778	189030					0.189	USER1GE5 JC		
37	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	149159	20483	614	102698	1760	277114					0.2771	USER1GE5 JC		
38	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	153803	23181	285	101022	1775	281176					0.2812	USER1GE4 JC		
39	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	140685	70595	11275606	113382	1920	11603168					11.6032	USER1GE1 JC		
40	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	108088	7624	222	65726	1746	184303					0.1843	USER1GE5 JC		
41	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	119784	9945	282	76225	1773	209052					0.2091	USER1GE4 JC		
42	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	94511	5061	132	44576	2427	147407					0.1474	USER1GE1 JC		
43	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	56951	10497	126	118293	1703	189186					0.1892	USER1GE5 JC		
44	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	55110	7646	210	122479	1616	187974					0.188	USER1GE4 JC		
45	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	119104	10588	354	109467	1604	242675					0.2427	USER1GE1 JC		
46	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	3051028	17103	9999318	222997	1770	13292831					13.2928	USER1GET JC		
47	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	129965	20381	121	212563	1870	366316					0.3663	USER1GE5 JC		
48	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	117036	17792	768	221666	1796	360790					0.3608	USER1GE4 JC		
49	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	121667	23095	468	217285	1673	366393					0.3664	USER1GE1 JC		
50	MPZ3	ZCON	2	40	200	200	NO	0	269	USER1	GET		2021/08/2021/02/202021/08/2318:	115629	13252	685	146376	1659	279825					0.2798	USER1GE1 JC		
51																											
52	REC_TYPE	SUBTYPE	SUBTYPE	SUBTYPE_VERSION																							

smfout

Ready

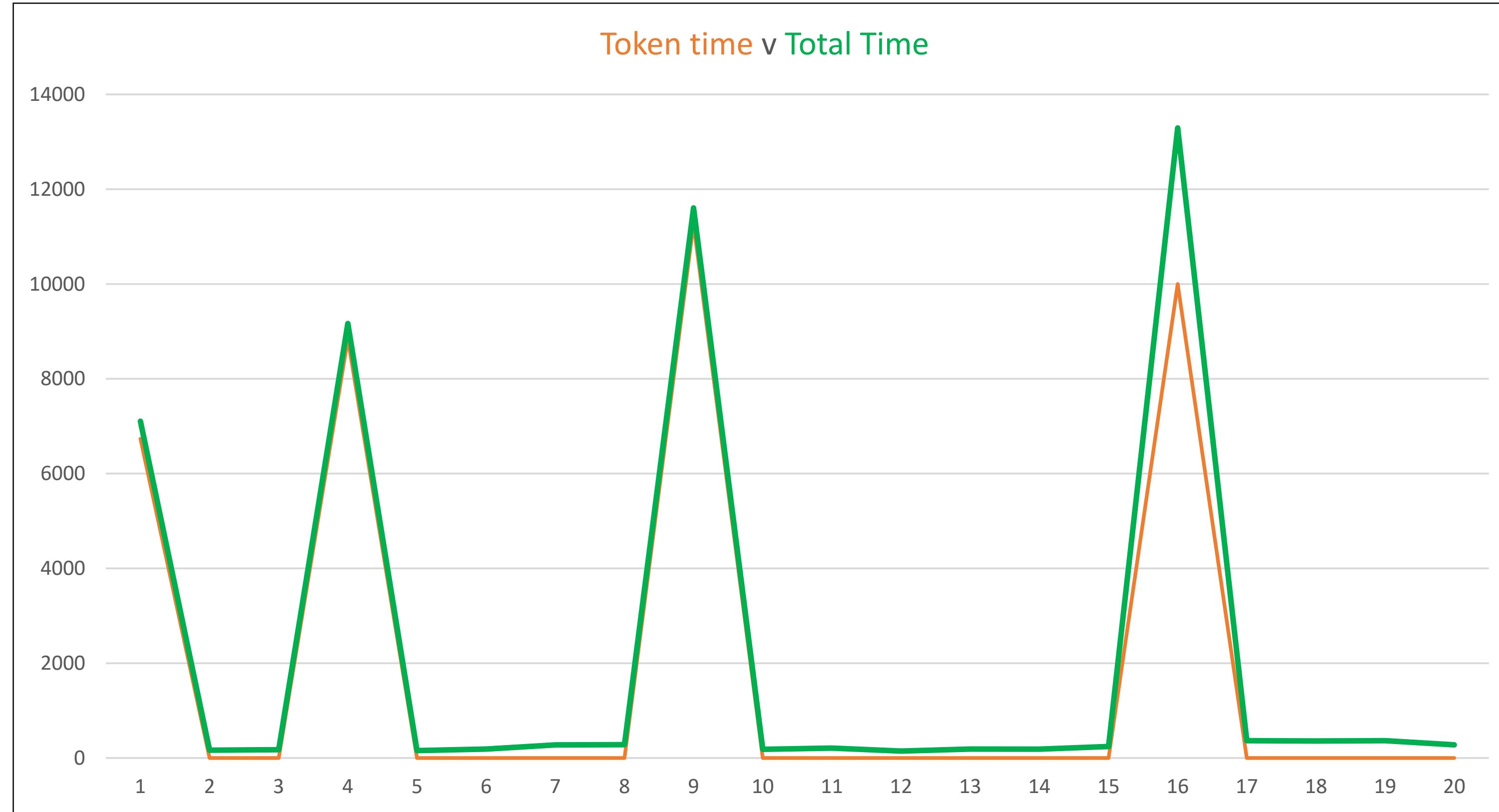
Some fields have been hidden

* Generated by using a modified version of the BAQSMFX sample program.

mitchj@us.ibm.com

© 2017, 2024 IBM Corporation
Slide 219

z/OS Connect SMF 123 subtype 2 version 2 graph example (OpenAPI 2)





Connection Management

Liberty default connection pool management <connectionManager>

- agedTimeout Amount of time before a connection can be discarded by pool maintenance.
- connectionTimeout Amount of time after which a connection request times out.
- maxIdleTime Amount of time a connection can be unused or idle until it can be discarded during pool maintenance.
- maxPoolSize Maximum number of physical connections for a pool.
- minPoolSize Minimum number of physical connections to maintain in the pool.
- purgePolicy Specifies which connections to destroy when a stale connection is detected in a pool.
- reapTime Amount of time between runs of the pool maintenance thread.

```
<connectionManger id="ConMgr1"  
    agedTimout=-1  
    connectionTimeout=30  
    maxIdleTIme=1800  
    maxPoolSize=50  
    minPoolSize=0  
    purgePolicy="EntirePool"  
    reapTIme=180/>
```



Connection Management for IMS TM

Use the connectionManagerRef attribute in an IMS ConnectionFactory to provide a connection pool for connections to IMS Connect.

```
<imsmobile_imsConnection connectionFactoryRef="IMSCF" connectionTimeout="-1"
    connectionType="IMSCONNECT" id="IMSCONN"/>

<connectionFactory id="IMSCF" connectionManagerRef="IMSTMConnMgr"
    containerAuthDataRef="Connection1_Auth" >
    <properties.gmoa hostName="wg31.washington.ibm.com" portNumber="4000"
        applicationName="IMSTMAPL"/>
</connectionFactory>

<connectionManger id="IMSTMConnMgr" agedTimout=-1 connectionTimeout=30
    maxIdleTIme=1800 maxPoolSize=50 minPoolSize=0 purgePolicy="EntirePool"
    reapTIme=180/>
```

The *maxPoolSize* in the *connectionManager* configuration element should not exceed the value of the IMS Connect *MAXSOC* attribute.



Connection Management for IMS DB

Use the `connectionManagerRef` attribute in an IMS ConnectionFactory to provide a connection pool for connections to IMS Connect.

```
<connectionFactory id="DFSIVPACConn" connectionManagerRef="IMSDBConnMgr" >
<properties.imsudbJLocal
    databaseName="DFSIVPA"
    datastoreName="IVP1"
    datastoreServer="wg31.washington.ibm.com"
    driverType="4"
    portNumber="5555"
    user="USER1"
    password="USER1"
    flattenTables="True"/>
</connectionFactory>

<connectionManger id="IMSDBConnMgr" agedTimout=-1 connectionTimeout=30
maxIdleTIme=1800 maxPoolSize=50 minPoolSize=0 purgePolicy="EntirePool"
reapTIme=180/>
```

The `maxPoolSize` in the `connectionManager` configuration element should not exceed the value of the IMS Connect `MAXSOC` attribute.



Connection Management for MQ

Use the connectionManagerRef attribute in a JMS ConnectionFactory to provide a connection pool for connections to a queue manager.

```
<jmsConnectionFactory id="qmgrCf" jndiName="jms/qmgrCf"  
connectionManagerRef="MQConnMgr">  
  <properties.wmqJMS transportType="CLIENT"  
    queueManager="ZMQ1"  
    channel="LIBERTY.DEF.SVRCONN"  
    hostName="wg31.washington.ibm.com"  
    port="1422" />  
</jmsConnectionFactory>  
  
<connectionManger id="MQConnMgr" agedTimout=-1 connectionTimeout=30  
  maxIdleTIme=1800 maxPoolSize=50 minPoolSize=0 purgePolicy="EntirePool"  
reapTIme=180/>
```

The *maxPoolSize* in the *connectionManager* configuration element should not exceed the value of the *MAXINST* or *MAXINSTC* attributes of the queue manager's server-connection channel.



Connection Management for outbound HTTP request, e.g., Db2, etc.

Outbound connections to Db2, authorization servers, API requesters servers are managed by z/OS Connect code (as is any endpoint configured by the use of a z/OS Connection configuration element).

Connections are managed and/or configured by the use of Java system parameters (-D) *http.maxConnections* and *http.keepAlive*.

- Dhttp.maxConnections=5
- Dhttp.keepAlive=true



TLS sessions

- When connections timeout, it is still possible to avoid the impact of full handshakes by reusing the TLS session id
- Configured by setting the `sslSessionTimeout` attribute on the `sslOptions` element to an amount of time
- Example setting `server.xml` file

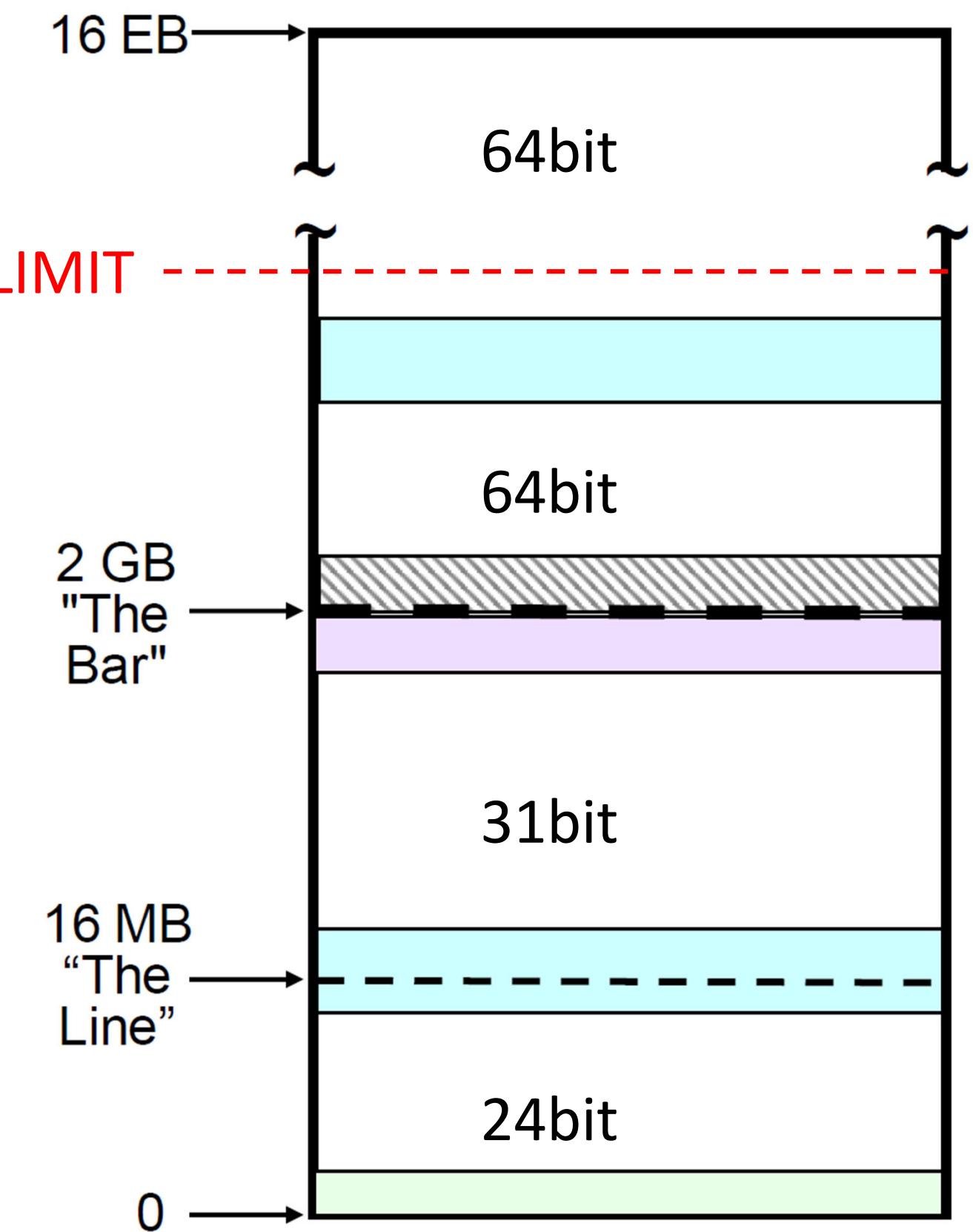
```
<httpEndpoint host="*" httpPort="80" httpsPort="443" id="defaultHttpEndpoint"  
httpOptionsRef="httpOpts" sslOptionsRef="mySSLOptions"/>  
  
<httpOptions id="httpOpts" keepAliveEnabled="true" maxKeepAliveRequests="100"  
persistTimeout="1m"/>  
  
<sslOptions id="mySSLOptions" sslRef="DefaultSSLSettings"  
sslSessionTimeout="10m"/>
```

- This sets the timeout limit of an TLS session to **10 minutes** (default is 8640ms)

MEMLIMIT - memory storage above-the-bar

```
//ZCON EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=4G,
//      PARM='PGM &ZCONHOME./bin/zosconnect run &PARMS.'
```

- Limits the amount of 64-bit storage
 - Only a limit, not pre-allocated
- z/OS uses above the bar storage for:
 - Native thread stack storage
- Java uses above the bar storage for:
 - Heap storage
 - Caches
 - Java thread

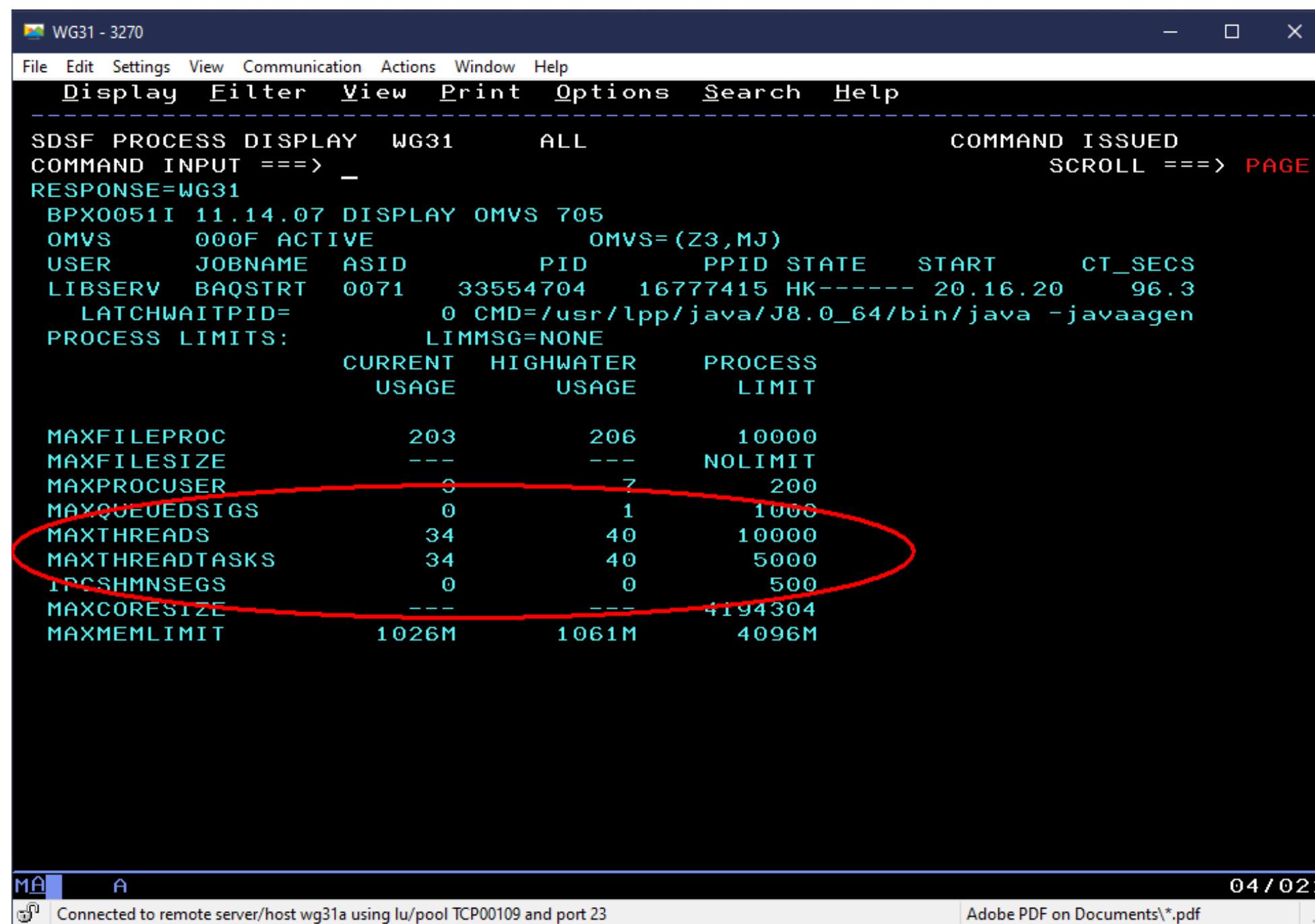


messages.log

CWWKB0125I: This server requested a REGION size of 0KB. The below-the-line storage limit is 8MB and the above-the-line storage limit is 1725MB.
 CWWKB0126I: MEMLIMIT=1000. MEMLIMIT CONFIGURATION SOURCE=JCL.

Native threads

- Native threads require 3Mb of above the bar storage (2Mb for LE and 1Mb for the JVM)
 - Monitor thread usage for the address space
 - *D OMVS,LIMITS,PID=<server pid>*



```

WG31 - 3270
File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help
SDSF PROCESS DISPLAY WG31 ALL
COMMAND INPUT ==> -
RESPONSE=WG31
BPX0051I 11.14.07 DISPLAY OMVS 705
OMVS 000F ACTIVE OMVS=(Z3,MJ)
USER JOBNAME ASID PID PPID STATE START CT_SECS
LIBSERV BAQSTRT 0071 33554704 16777415 HK----- 20.16.20 96.3
LATCHWAITPID= 0 CMD=/usr/lpp/java/J8.0_64/bin/java -javaagen
PROCESS LIMITS: LIMMSG=NONE
CURRENT HIGHWATER PROCESS
USAGE USAGE LIMIT
MAXFILEPROC 203 206 10000
MAXFILESIZE --- --- NOLIMIT
MAXPROCUSER 0 7 200
MAXQUEUEDSIGS 0 1 1000
MAXTHREADS 34 40 10000
MAXTHREADTASKS 34 40 5000
ITCASHMNSEGS 0 0 500
MAXCORESIZE --- --- 4194304
MAXMEMLIMIT 1026M 1061M 4096M

```

Connected to remote server/host wg31a using lu/pool TCP00109 and port 23 04 / 021

- MAXTHREADS must be greater than or equal to MAXTHREADTASK
- Take action when USAGE comes within 80-90% of maxThreads



Tech-Tip: Java heap storage

- Java heap is the area of memory managed by the Java Virtual Machine (JVM) where Java class objects and other objects instantiated by Java applications running in the JVM are stored and resides above the bar. The JVM obtains storage in the heap storage on behalf of the Java applications.
- A process known as garbage collection reclaims the storage when the object is no longer, for more information see URL https://docs.oracle.com/cd/E15289_01/JRSDK/garbage_collect.htm

Non-standard Java options related to garbage collection and heap storage*

- Xgcpolicy:gencon Garbage collection policy, the default is *gencon* and is the recommended garbage collection policy
- Xms<size> Initial heap size, defaults to *8MB* on z/OS
- Xmx<size> Maximum heap size, defaults to half the available memory with a minimum of *16 MB* and a maximum of *512 MB*

<https://www.ibm.com/docs/en/sdk-java-technology/8?topic=reference-default-settings>

Standard Java options related garbage collection reporting*

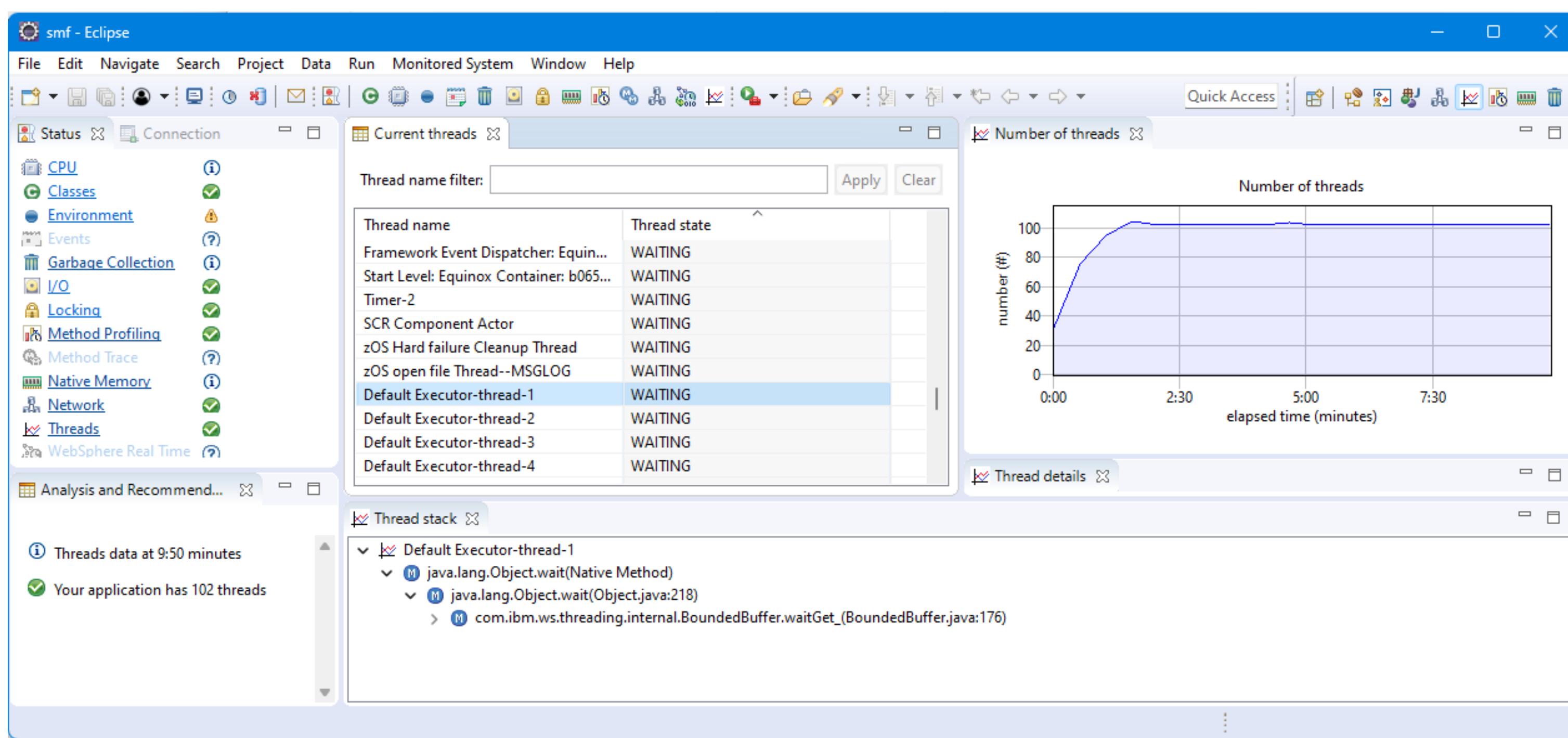
- verbose:gc Writes verbose garbage collection information.
- verbose:sizes Writes information to stderr describing the active memory usage settings.

<https://www.ibm.com/docs/en/sdk-java-technology/8?topic=options-standard>



Java threads

- Java threads handle application requests (executor threads), garbage collection and other Java housekeeping functions.
 - Each Java thread require 1.6Kb of Java heap storage
 - The maximum number of executor threads defaults to unlimited.
 - The maximum number of executor threads can be limited with configuration element `<executor maxThreads="300"/>`
 - The attribute *maxOpenConnections* attribute in the *tcpOptions* configuration element should be set to less than or equal to the value of the maximum number of executor threads.





MEMLIMIT Recommendations

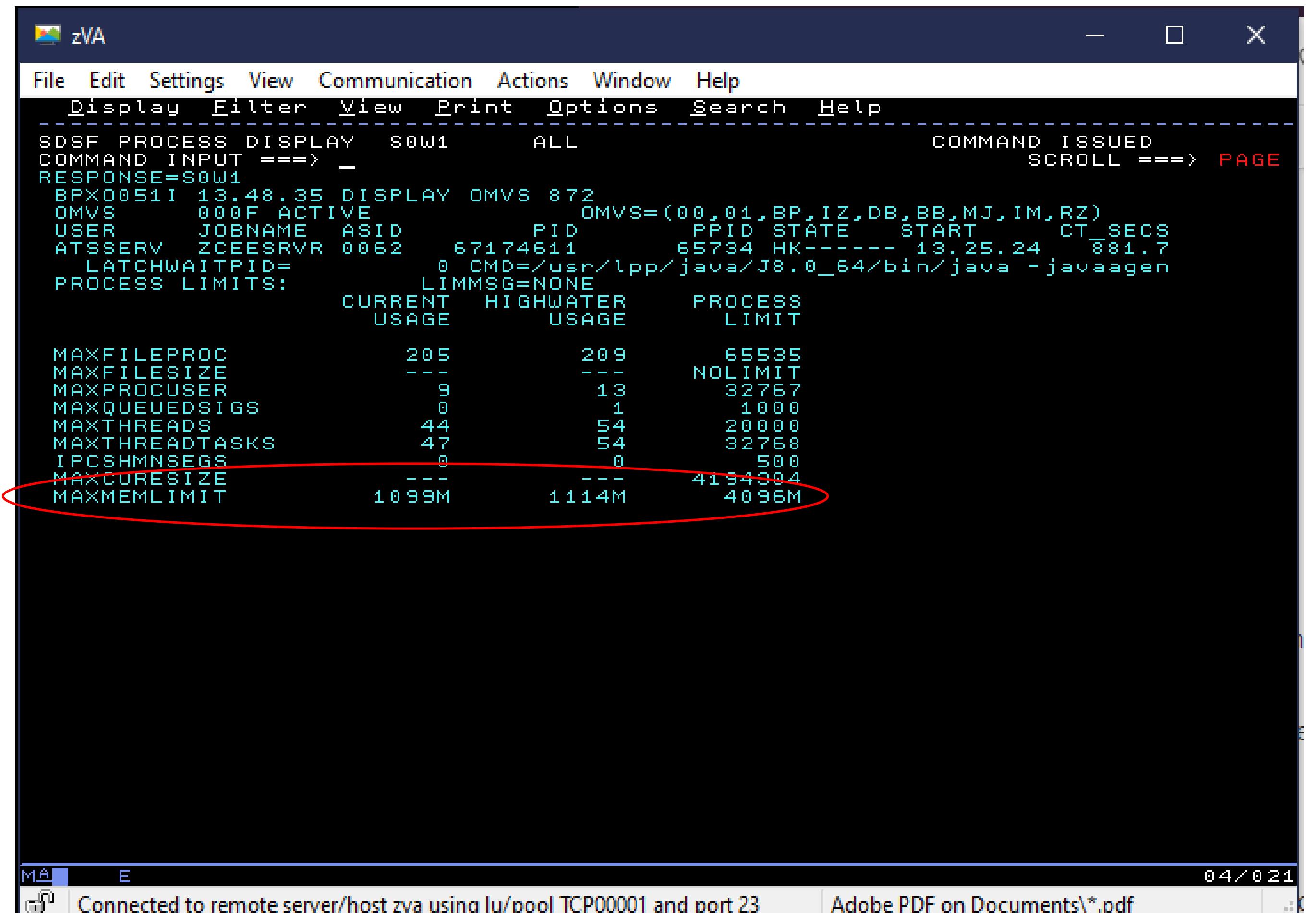
- Don't reach the maximum!
 - Results in Java Out Of Memory errors and system abends
 - z/OS Connect EE will stop processing API requests
- Ensure this doesn't happen
 - Limit the Liberty Default Executor thread pool
 - `maxThreads` default value is `-1` No Limit!
 - **MEMLIMIT** =
 - Maximum JVM Heap Size (`-Xmx`)
 - + 20% of the Maximum Heap Size (for JIT caches and other JVM requirements)
 - + Default Executor pool `maxThreads` * 3MB

`<executor maxThreads="300"/>`

Maximum JVM Heap Size – half the available memory with a minimum of 16 MB and a maximum of 512 MB

MEMLIMIT – management

- MEMLIMIT values
 - MEMLIMIT = maximum Java heap size + 50% of maximum heap size
 - or
 - MEMLIMIT = maximum Java heap size + 20% of Java heap size + (number of executor threads * 3Mb)
- Monitor periodically
 - To track high water mark with MVS command
D OMVS,LIMITS,PID=<server pid>
- Don't reach the maximum!
 - Results in Java Out Of Memory errors and system abends
 - Liberty will stop processing requests



```

zVA
File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help
SDSF PROCESS DISPLAY S0W1 ALL COMMAND ISSUED
COMMAND INPUT ===> -
SCROLL ===> PAGE
RESPONSE=S0W1
BPX0051I 13.48.35 DISPLAY OMVS 872
OMVS 000F ACTIVE OMVS=(00,01,BP,IZ,DB,BB,MJ,IM,RZ)
USER JOBNAM ASID PID PPID STATE START CT_SECS
ATSSERV ZCEESRVR 0062 67174611 65734 HK----- 13.25.24 881.7
LATCHWAITPID= 0 CMD=/usr/lpp/java/J8.0_64/bin/java -javaagen
PROCESS LIMITS: LIMMSG=NONE
CURRENT HIGHWATER PROCESS
USAGE USAGE LIMIT
MAXFILEPROC 205 209 65535
MAXFILESIZE -- -- NOLIMIT
MAXPROCUSER 9 13 32767
MAXQUEUEDSIGS 0 1 1000
MAXTHREADS 44 54 20000
MAXTHREADTASKS 47 54 32768
IPCSHMNSEGS 0 0 500
MAXCORESIZE -- -- 4194304
MAXMEMLIMIT 1099M 1114M 4096M

```

The screenshot shows the z/OS zVA interface with the DISPLAY OMVS command running. The output displays various system parameters and process limits. A red oval highlights the 'MAXMEMLIMIT' row in the 'PROCESS LIMITS:' section, which shows a current usage of 1099M, a high watermark of 1114M, and a limit of 4096M.



Inbound persistent connections

- Persistent connections can be used to avoid too many handshakes
- Configured by setting the `keepAliveEnabled` attribute on the `httpOptions` element to **true**
- Example setting `server.xml` file

```
<httpEndpoint host="*" httpPort="80" httpsPort="443" id="defaultHttpEndpoint"  
httpOptionsRef="httpOpts"/>  
  
<httpOptions id="httpOpts" keepAliveEnabled="true" maxKeepAliveRequests="500"  
persistTimeout="1m" />
```

- This sets the connection timeout to **1 minute** (default is 30 seconds) and sets the maximum number of persistent requests that are allowed on a single HTTP connection to **500**
- It is recommended to set a maximum number of persistent requests when connection workload balancing is configured
- It is also necessary to configure the client to support persistent connections

Where do I look when things go wrong?

Where to find information when a problem occurs.



messages.log

The screenshot shows a terminal window titled 'mpz3'. The menu bar includes 'File', 'Edit', 'Settings', 'View', 'Communication', 'Actions', 'Window', 'Help' at the top, and a secondary menu with 'File', 'Edit', 'Edit_Settings', 'Menu', 'Utilities', 'Compilers', 'Test', 'Help' below it. The main area displays a log file named 'messages.log' located at '/MPZ3/var/zosconnect/servers/myServer/logs/messages.log'. The log contains numerous messages starting with '000228 System Property com.ibm.ims.jdbcenvironment set to 'WAS''. It also includes entries about resource adapters, application resources, server configuration, and various TCP channels. A message at the bottom states 'The myServer server is ready to run a smarter planet. The myServer server started in 17.991 seconds.' The status bar at the bottom shows '04/015'.

```
VIEW /MPZ3/var/zosconnect/servers/myServer/logs/messages.log
Command ==> -
000228 System Property com.ibm.ims.jdbcenvironment set to 'WAS'
000229 JZCA7001I: Resource adapter imsudbJLocal installed in 5.685 seconds.
000230 CWWKZ0014W: The application resources could not be started as it could not be found at location /var/zosconnect/servers/mySe
000231 CWWKZ0018I: Starting application serverConfig.
000232 CWWKZ0135I: The serverConfig application is using the expanded directory at the /var/zosconnect/servers/myServer location.
000233 SRVE0169I: Loading Web Module: myServer.
000234 SRVE0250I: Web Module myServer has been bound to default_host.
000235 CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/server/config/
000236 SESN0176I: A new session context will be created for application key default_host/server/config
000237 SESN0172I: The session manager is using the Java default SecureRandom implementation for session ID generation.
000238 SRVE9103I: A configuration file for a web server plugin was automatically generated for this server at /var/zosconnect/serve
000239 CWWKZ0001I: Application serverConfig started in 0.036 seconds.
000240 SRVE9103I: A configuration file for a web server plugin was automatically generated for this server at /var/zosconnect/serve
000241 CWWK00219I: TCP Channel defaultHttpEndpoint has been started and is now listening for requests on host * (IPv6) port 9080.
000242 CWWK00219I: TCP Channel defaultHttpEndpoint-ssl has been started and is now listening for requests on host * (IPv6) port 94
000243 CWWKF0012I: The server installed the following features: [adminCenter-1.0, apiDiscovery-1.0, appSecurity-2.0, distributedMap
000244 CWWKF0008I: Feature update completed in 17.517 seconds.
000245 CWWKF0011I: The myServer server is ready to run a smarter planet. The myServer server started in 17.991 seconds.
000246 CWWKS2932I: The authorized version of the SAF user registry is activated. Authentication will proceed using authorized nativ
000247 CWWKS1100A: Authentication did not succeed for user ID user1. An invalid user ID or password was specified.
000248 CWWKS1100A: Authentication did not succeed for user ID user1. An invalid user ID or password was specified.
***** ***** Bottom of Data *****
```

SYSLOG/STC JESMSG LG DD

The screenshot shows the SDSF OPERLOG window for MPZ3 on 08/30/2021 at 08:00. The log displays various system events, including resource manager restarts, exit processing, and application server configuration. Key entries include:

- +CWLIB0103I: Resource manager BBG.DEFAULT.DA38C9E1985D0C11.IBM with 337
- the corresponding token ID of 01000001023805f80000000900000007 has successfully restarted with Resource Recovery Services (RRS). Number of unresolved units of recovery: 0
- ATR169I RRS HAS UNSET EXITS FOR RESOURCE MANAGER 338
- BBG.DEFAULT.DA38C9E1985D0C11.IBM REASON: UNREGISTERED
- +CWLIB0104I: Recovery processing for resource Manager 339
- BBG.DEFAULT.DA38C9E1985D0C11.IBM with the corresponding token ID of 01000001023805f80000000900000007 has completed.
- DSNU133I -DSNB DSNUVSMF - TRACE DATA LOST, SMF NOT ACCESSIBLE RC=24
- +CWWKZ0001I: Application serverConfig started in 0.036 seconds.
- +CWWKF001II: The myServer server is ready to run a smarter Planet. The 341
- myServer server started in 17.991 seconds.
- ICH408I USER(USER1) GROUP(SYS1) NAME() 342
- LOGON/JOB INITIATION - PASS PHRASE IS NOT VALID
- ICH408I USER(USER1) GROUP(SYS1) NAME() 343
- LOGON/JOB INITIATION - PASS PHRASE IS NOT VALID
- DSNU133I -DSNC DSNUVSMF - TRACE DATA LOST, SMF NOT ACCESSIBLE RC=24
- DSNU123I -DSNB DSNUVSMF - TRACE RECORDING HAS BEEN RESUMED ON SMF
- +CSQX251I ZMQA CSQXSTRL Listener started, TRPTYPE=TCP INDISP=QMGR
- +CSQX218E ZMQA CSQXLSTT Listener not started - unable to bind, 245

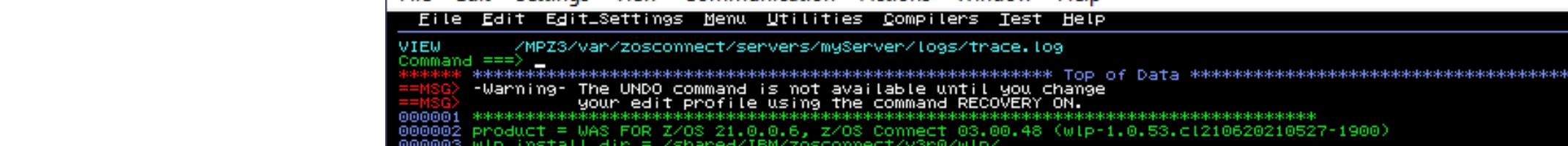
STC STDOUT DD

First Failure Data Collection (FFDC) dumps

 mpz3
File Edit Settings View Communication Actions Window Help
File Edit Settings Menu Utilities Compilers Test Help
VIEW /MPZ3/var/zosconnect/servers/myServer/logs/ffdc/ffdc_21.08.30_15.05.58.0.log Columns 00001 00124
Command ==> _ Scroll ==> PAGE
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001 -----start of DE processing----- = [8/30/21 15:05:58:198 GMT]
000002 Exception = javax.net.ssl.SSLHandshakeException
000003 Source = com.ibm.ws.channel.ssl.internal.SSLReadServiceContext\$SSLReadCompletedCallback
000004 probeid = 798
000005 Stack Dump = javax.net.ssl.SSLHandshakeException: Received fatal alert: unknown_ca
000006 at com.ibm.jsse2.g.a(g.java:28)
000007 at com.ibm.jsse2.g.a(g.java:48)
000008 at com.ibm.jsse2.bb.a(bb.java:33)
000009 at com.ibm.jsse2.g\$c.consume(g\$c.java:29)
000010 at com.ibm.jsse2.bb.a(bb.java:71)
000011 at com.ibm.jsse2.a0.a(a0.java:48)
000012 at com.ibm.jsse2.bg.g(bg.java:317)
000013 at com.ibm.jsse2.bg.f(bg.java:164)
000014 at com.ibm.jsse2.bg.e(bg.java:199)
000015 at com.ibm.jsse2.bg.unwrap(bg.java:106)
000016 at javax.net.ssl.SSLEngine.unwrap(SSLEngine.java:21)
000017 at com.ibm.ws.channel.ssl.internal.SSLReadServiceContext.decryptMessage(SSLReadServiceContext.java:1212)
000018 at com.ibm.ws.channel.ssl.internal.SSLReadServiceContext\$SSLReadCompletedCallback.complete(SSLReadServiceContext.java:1800)
000019 at com.ibm.ws.tcpchannel.internal.AioReadCompletionListener.futureCompleted(AioReadCompletionListener.java:138)
M C 04/015
Connected to remote server/host mpz3 using lu/pool MPZ30030 and port 23

```
File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help
SDSF OUTPUT DISPLAY BAQSTRRT STC10771 DSID 103 LINE 84      COLS 02- 133
COMMAND INPUT ==> SCROLL ==> CSR
YAUDIT  ** BAQR7130I: z/OS Connect EE API miniloanccs was registered successfully for API Discovery.
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/ibm/api/explorer/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/api/docs/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/api/explorer/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/ibm/api/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/IBMJMXConnectorREST/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/ibm/adminCenter/serverConfig-i
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/jwt/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/ibm/adminCenter/explore-1.0/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/adminCenter/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/zosConnect/apiRequesters/
YAUDIT  ** J2CA7001I: Resource adapter gmoa installed in 4.766 seconds.
YAUDIT  ** J2CA7001I: Resource adapter imsudbJLocal installed in 5.685 seconds.
YWARNNG  ** CWWKZ0014W: The application resources could not be found at location /var/zosconnect/servers/m
YAUDIT  ** CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/server/config/
YAUDIT  ** CWWKZ0001I: Application serverConfig started in 0.036 seconds.
YAUDIT  ** CWWKF0012I: The server installed the following features: YadminCenter-1.0, apiDiscovery-1.0, appSecurity-2.0, distributed
YAUDIT  ** CWWKF0011I: The myServer server is ready to run a smarter planet. The myServer server started in 17.991 seconds.
YAUDIT  ** CWWKS1100A: Authentication did not succeed for user ID user1. An invalid user ID or password was specified.
YAUDIT  ** CWWKS1100A: Authentication did not succeed for user ID user1. An invalid user ID or password was specified.
***** BOTTOM OF DATA *****
```

trace.out



The screenshot shows the MPZ3 application interface. The title bar says "mpz3". The menu bar includes "File", "Edit", "Settings", "View", "Communication", "Actions", "Window", and "Help". Below the menu is a toolbar with icons for "File", "Edit", "Edit_Settings", "Menu", "Utilities", "Compilers", "Test", and "Help". The main area displays a trace log from "/MPZ3/var/zosconnect/servers/myServer/logs/trace.log". The log contains several entries, including a warning about the UNDO command and details about the server configuration and Java runtime environment. The right side of the window has a status bar with "Columns 00001 00124" and "Scroll ==> PAGE".

```
VIEW /MPZ3/var/zosconnect/servers/myServer/logs/trace.log
Command ==> -
***** **** Top of Data ****
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000001 ****
000002 product = WAS FOR z/OS 21.0.0.6, z/OS Connect 03.00.48 (wlp-1.0.53.c1210620210527-1900)
000003 wlp.install.dir = /shared/IBM/zosconnect/v3r0/wlp/
000004 server.config.dir = /var/zosconnect/servers/myServer/
000005 java.home = /MA4RS1/usr/lpp/java/J8.0_64
000006 java.version = 1.8.0_301
000007 java.runtime = Java(TM) SE Runtime Environment (8.0.6.35 - pmz6480sr6fp35-20210714_01(SR6 FP35))
000008 os = z/OS (02.04.00; s390x) (en_US)
000009 process = 16843093@MPZ3
000010 trace.specification = **=info:com.ibm.zosconnect.wv==FINEST:zosConnect=all:zosConnectServiceCics=all
000011 ****
000012 [8/30/21 15:34:58:201 GMT] 00000064 id=e8994258 com.ibm.zosconnect.service.cics.internal.conn.isc.Connection > getVersion E
000013 [8/30/21 15:34:58:203 GMT] 00000064 id=e8994258 com.ibm.zosconnect.service.cics.internal.conn.isc.Connection < getVersion E
000014
000015 [8/30/21 15:34:58:203 GMT] 00000064 id=00000000 connect.service.cics.internal.conn.isc.headers.ISCHTTPHeader > getHeader E
000016
000017 [8/30/21 15:34:58:203 GMT] 00000064 id=00000000 connect.service.cics.internal.conn.isc.headers.IscV2HTTPHeader > <init> Entry
000018 [8/30/21 15:34:58:203 GMT] 00000064 id=438687cd connect.service.cics.internal.conn.isc.headers.IscV2HTTPHeader < <init> Exit
000019 com.ibm.zosc
```



Issues and problems can be categorized

- First realize that actual products problems do occur, but they are rare. In my experience most problems and issues can be resolved with a little investigation and some analysis. I have found that most problems and issues will fall in these categories.

- **Basic Security issues**
 - Insufficient access to local SAF resources, e.g., APPL, EJBCROLE, SERVER resources
 - Security issues related to XML configuration elements, safCredentials, sslDefault, keystore, etc.
- **Advanced Security issues**
 - Key ring access, e.g., FACILITY resources IRR.DIGTCERT or RDATALIB or IDIDMAP resources.
 - Key ring contents, e.g., missing certificates, key usage, personal and certificate authorities, private keys versus public keys.
 - Incorrect use of certificates in a TLS handshakes versus certificates used for token validation.
- **z/OS Connect XML Configuration issues**
 - Missing or misspelled configuration attributes (remember the Liberty XML parser is too forgiving)
- **External resource Issues**
 - Service provider configuration issues.
 - Timeouts
 - Network Firewalls
 - Resource Security
 - Other resource errors

Remember external symptoms will overlap. But the use of rigor in setting configuration standards and following a process in problem isolation/determination process will help reduce the impact of problems and issues.

messages.log - The anatomy of a message in the messages.log file



```
*****  
product = WAS FOR Z/OS 21.0.0.6, z/OS Connect 03.00.48 (wlp-1.0.53.cl210620210527-1900)  
wlp.install.dir = /shared/IBM/zosconnect/v3r0/wlp/  
server.config.dir = /var/zosconnect/servers/zceoepid/  
java.home = /MA4RS1/usr/lpp/java/J8.0_64  
java.version = 1.8.0_301  
java.runtime = Java(TM) SE Runtime Environment (8.0.6.35 - pmz6480sr6fp35-20210714_01(SR6 FP35))  
os = z/OS (02.04.00; s390x) (en_US)  
process = 16843186@MPZ3  
*****  
[9/3/21 13:38:02:831 GMT] 00000013 com.ibm.ws.kernel.launch.internal.FrameworkManager  
[9/3/21 13:38:04:439 GMT] 0000001f com.ibm.ws.config.xml.internal.XMLConfigParser  
[9/3/21 13:38:04:466 GMT] 0000001f com.ibm.ws.config.xml.internal.XMLConfigParser  
[9/3/21 13:38:04:470 GMT] 0000001f com.ibm.ws.config.xml.internal.XMLConfigParser  
[9/3/21 13:38:04:473 GMT] 0000001f com.ibm.ws.config.xml.internal.XMLConfigParser  
[9/3/21 13:38:04:476 GMT] 0000001f com.ibm.ws.config.xml.internal.XMLConfigParser  
[9/3/21 13:38:04:481 GMT] 0000001f com.ibm.ws.config.xml.internal.XMLConfigParser  
[9/3/21 13:38:04:610 GMT] 00000021 com.ibm.ws.zos.core.internal.NativeServiceTracker  
[9/3/21 13:38:04:612 GMT] 00000021 com.ibm.ws.zos.core.internal.NativeServiceTracker  
[9/3/21 13:38:04:628 GMT] 00000021 com.ibm.ws.zos.core.internal.NativeServiceTracker  
[9/3/21 13:38:04:679 GMT] 00000021 com.ibm.ws.zos.core.internal.NativeServiceTracker  
[9/3/21 13:38:04:680 GMT] 00000021 com.ibm.ws.zos.core.internal.NativeServiceTracker  
[9/3/21 13:38:04:680 GMT] 00000021 com.ibm.ws.zos.core.internal.NativeServiceTracker  
- - - - -  
[9/3/21 13:38:42:347 GMT] 00000040 om.ibm.ws.app.manager.rar.internal.RARApplicationHandlerImpl  
[9/3/21 13:38:42:419 GMT] 0000003e com.ibm.ws.jmx.connector.server.rest.RESTAppListener  
[9/3/21 13:38:42:422 GMT] 0000003e com.ibm.ws.jmx.connector.server.rest.RESTAppListener  
[9/3/21 13:38:42:428 GMT] 0000002c com.ibm.ws.tcpchannel.internal.TCPPort  
[9/3/21 13:38:42:431 GMT] 0000002c com.ibm.ws.tcpchannel.internal.TCPPort  
[9/3/21 13:38:42:437 GMT] 00000042 com.ibm.ws.webcontainer.osgi.mbeans.PluginGenerator  
[9/3/21 13:38:42:489 GMT] 0000002c com.ibm.ws.kernel.feature.internal.FeatureManager  
[9/3/21 13:38:42:490 GMT] 0000002c com.ibm.ws.kernel.feature.internal.FeatureManager  
[9/3/21 13:38:42:490 GMT] 0000002c com.ibm.ws.kernel.feature.internal.FeatureManager  
[9/3/21 13:41:31:640 GMT] 00000045 .security openidconnect.client.internal.OidcClientConfigImpl  
[9/3/21 13:41:31:691 GMT] 00000045 http.authentication.filter.internal.AuthenticationFilterImpl  
[9/3/21 13:41:32:824 GMT] 00000053 com.ibm.zosconnect.service.cics.internal.conn.isc.Connection  
*****  
A CWWKE0001I: The server zceoepid has been launched.  
A CWWKG0028A: Processing included configuration resource  
A CWWKB0125I: This server requested a REGION size of 0KB  
I CWWKB0126I: MEMLIMIT=2000. MEMLIMIT CONFIGURATION SOUR  
I CWWKB0122I: This server is connected to the default an  
I CWWKB0103I: Authorized service group KERNEL is availab  
I CWWKB0103I: Authorized service group LOCALCOM is avail  
I CWWKB0103I: Authorized service group PRODMGR is availa  
- - - - - 148 Line(s) not Displayed  
A J2CA7001I: Resource adapter imsudbJLocal installed in  
I CWWKX0103I: The JMX REST connector is running and is a  
I CWWKX0103I: The JMX REST connector is running and is a  
I CWWKO0219I: TCP Channel defaultHttpEndpoint has been s  
I CWWKO0219I: TCP Channel defaultHttpEndpoint-ssl has be  
I SRVE9103I: A configuration file for a web server plugi  
A CWWKF0012I: The server installed the following feature  
I CWWKF0008I: Feature update completed in 37.484 seconds  
A CWWKF0011I: The zceoepid server is ready to run a smar  
I CWWKS1700I: OpenID Connect client ATS configuration su  
I CWWKS4358I: The authentication filter ATSAuthFilter co  
I BAQR0680I: CICS connection cscvinc established with 10
```

- **WLP_LOGGING_CONSOLE_FORMAT - SIMPLE** - Use the simple logging format. As of Liberty release 20.0.0.6 (z/OS Connect V3.034), this format writes the messages to STDOUT and STDERR with time stamps included.



Basic security issues – Sometimes the problem is easy to find

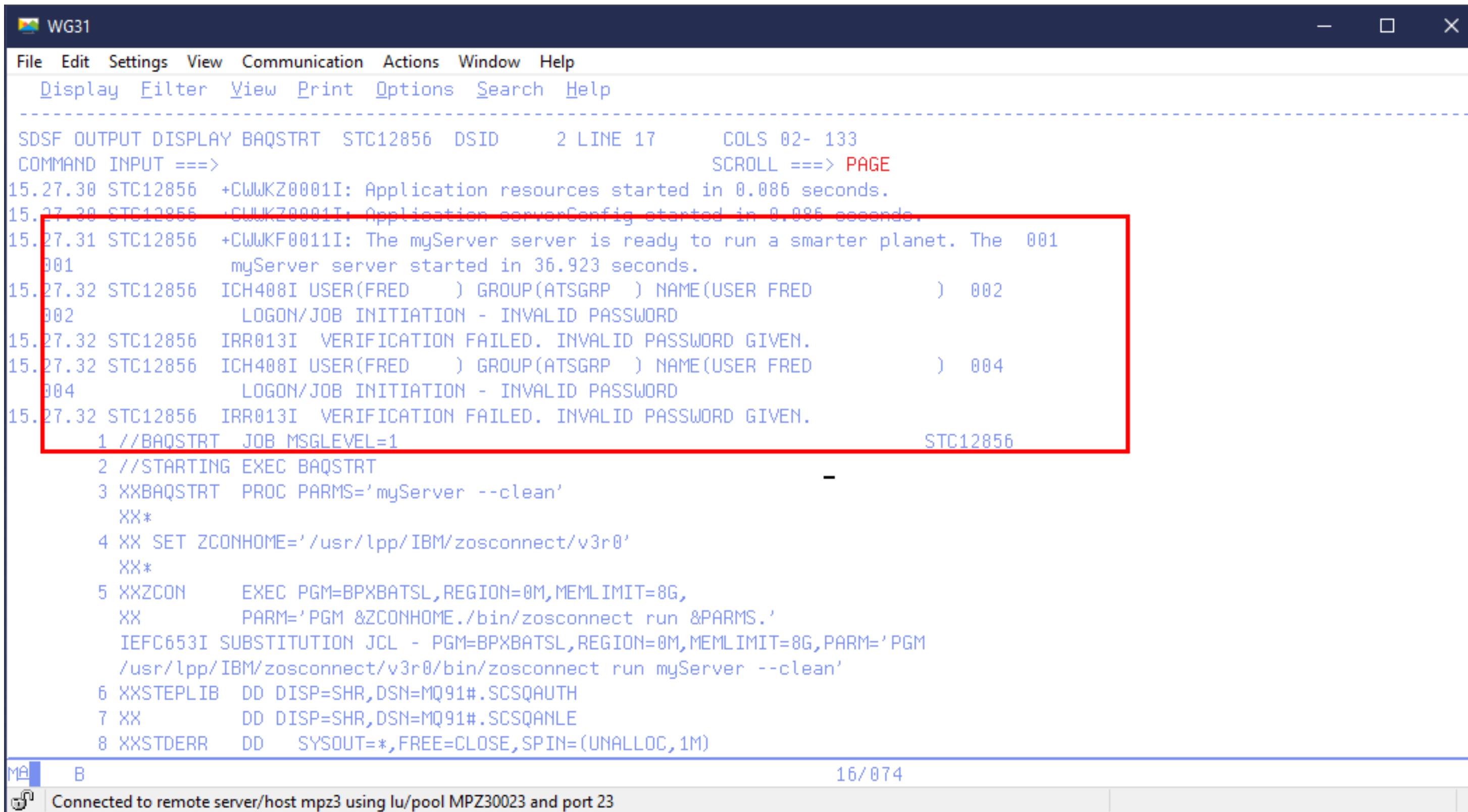
The STDOUT may show:

```
ÝAUDIT    " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified  
ÝAUDIT    " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified
```

And the messages.log displays:

```
CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.
```

But the JESMSGLOG and SYSLOG displays:



```
WG31  
File Edit Settings View Communication Actions Window Help  
Display Filter View Print Options Search Help  
-----  
SDSF OUTPUT DISPLAY BAQSTRT STC12856 DSID 2 LINE 17 COLS 02- 133  
COMMAND INPUT ==> SCROLL ==> PAGE  
15.27.30 STC12856 +CWWKZ0001I: Application resources started in 0.086 seconds.  
15.27.30 STC12856 +CWWKZ0001I: Application serverConfig started in 0.086 seconds.  
15.27.31 STC12856 +CWWJKF0011I: The myServer server is ready to run a smarter planet. The 001  
001 myServer server started in 36.923 seconds.  
15.27.32 STC12856 ICH408I USER(FRED ) GROUP(ATSGRP ) NAME(USER FRED ) 002  
002 LOGON/JOB INITIATION - INVALID PASSWORD  
15.27.32 STC12856 IRR013I VERIFICATION FAILED. INVALID PASSWORD GIVEN.  
15.27.32 STC12856 ICH408I USER(FRED ) GROUP(ATSGRP ) NAME(USER FRED ) 004  
004 LOGON/JOB INITIATION - INVALID PASSWORD  
15.27.32 STC12856 IRR013I VERIFICATION FAILED. INVALID PASSWORD GIVEN.  
1 //BAQSTRT JOB MSGLEVEL=1 STC12856  
2 //STARTING EXEC BAQSTRT  
3 XXBAQSTRT PROC PARMs='myServer --clean'  
XX*  
4 XX SET ZCONHOME='/usr/lpp/IBM/zosconnect/v3r0'  
XX*  
5 XXZCON EXEC PGM=BPXBATSL,REGION=0M,MEMLIMIT=8G,  
XX PARM='PGM &ZCONHOME./bin/zosconnect run &PARMS.'  
IEFC653I SUBSTITUTION JCL - PGM=BPXBATSL,REGION=0M,MEMLIMIT=8G,PARM='PGM  
/usr/lpp/IBM/zosconnect/v3r0/bin/zosconnect run myServer --clean'  
6 XXSTEPLIB DD DISP=SHR,DSN=MQ91#.SCSQAUTH  
7 XX DD DISP=SHR,DSN=MQ91#.SCSQANLE  
8 XXSTDERR DD SYSOUT=*,FREE=CLOSE,SPIN=(UNALLOC,1M)  
MA B  
Connected to remote server/host mpz3 using lu/pool MPZ30023 and port 23 16/074
```



Basic security issues – Sometimes you must dig a little more

The STDOUT may show:

```
ÝAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified  
ÝAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified
```

But there are no SAF messages in the SYSLOG:

While the messages.log displays a SAF return code and reason code:

```
VIEW      /MPZ3/var/zosconnect/servers/myServer/logs/messages.log          Columns 00100 00223  
Command ==> -  
000256  SAF return code 0x00000008. RACF return code 0x00000008. RACF reason code 0x00000020.  
000257  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000258  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000259  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000260  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000261  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000262  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000263  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000264  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000265  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000266  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000267  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000268  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000269  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000270  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000271  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000272  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000273  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000274  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000275  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
000276  CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZD  
000277  CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
***** ***** Bottom of Data *****
```

MA B

04/015

Connected to remote server/host mpz3 using lu/pool MPZ30023 and port 23

CWWKS2907E: SAF Service IRRSIA00_CREATE did not succeed because user FRED has insufficient authority to access APPL-ID BBGZDFLT. SAF return code 0x00000008. RACF return code 0x00000008. RACF reason code 0x00000020.

mitchj@us.ibm.com

Tech-Tip: And be aware of hex v. decimal in return and reason codes

RACF return code 0x00000008. RACF reason code 0x00000020.

Table 1. initACEE create return codes

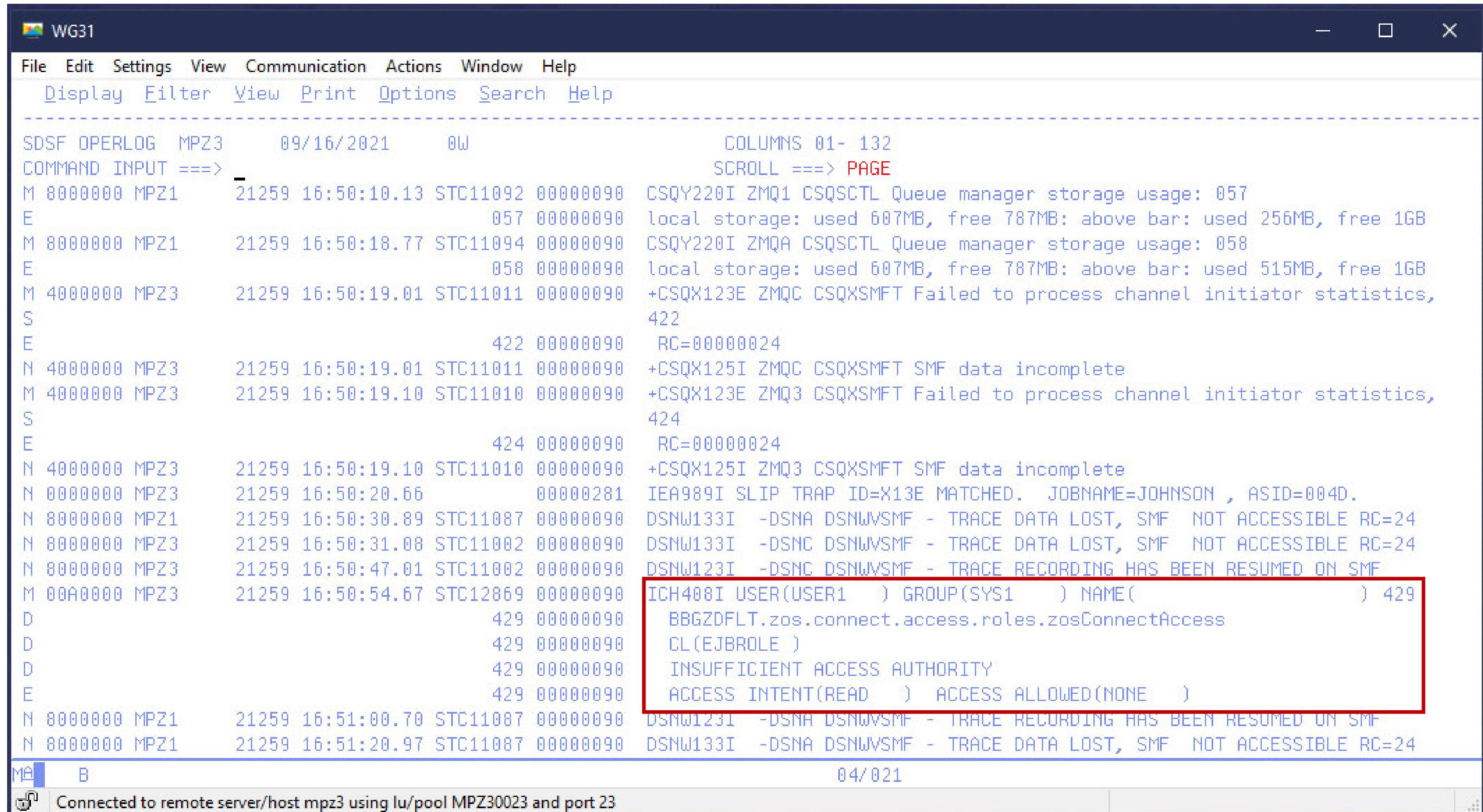
SAF return code	RACF® return code	RACF reason code	Explanation
0	0	0	The service was successful.
4	0	0	RACF is not installed.
8	8	4	Parameter list error occurred.
8	8	8	An internal error occurred during RACF processing.
8	8	12	Recovery environment could not be established.
8	8	16	User ID is not defined to RACF.
8	8	20	Password, Password Phrase or Pass Ticket is not valid.
8	8	24	Password or Password Phrase is expired.
8	8	28	User ID is revoked or user access to group is revoked.
8	8	32	The user does not have appropriate RACF access to either the SECLABEL, SERVAUTH profile, or APPL specified in the parmlist.
8	8	36	Certificate is not valid.
8	8	40	▷ No user ID is defined for this certificate. See Usage Note number 37. ▷
8	8	44	The client security label is not equivalent to the server's security label.
8	8	48	A managed ACEE is requested with a nested RACO in the Envir_In parameter.
8	12	InitUSP reason code	initUSP failed. See initUSP reason codes in Return and reason codes .

Hex '20' = Dec '32'

Root cause – No READ access to APPL resource BBGZDFLT

Basis security issues - Use the SYSLOG/JESMSGLG output

The SYSLOG shows a ICH408I message:



```

WG31
File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help
-----
SDSF OPERLOG MPZ3      09/16/2021     0W          COLUMNS 01- 132
COMMAND INPUT ==> -
M 8000000 MPZ1      21259 16:50:10.13 STC11092 00000090 CSQY220I ZMQ1 CSQSCTL Queue manager storage usage: 057
E                               057 00000090 local storage: used 607MB, free 787MB; above bar: used 256MB, free 1GB
M 8000000 MPZ1      21259 16:50:18.77 STC11094 00000090 CSQY220I ZMQA CSQSCTL Queue manager storage usage: 058
E                               058 00000090 local storage: used 607MB, free 787MB; above bar: used 515MB, free 1GB
M 4000000 MPZ3      21259 16:50:19.01 STC11011 00000090 +CSQX123E ZMQC CSQXSMFT Failed to process channel initiator statistics,
S                                         422
E                               422 00000090 RC=00000024
N 4000000 MPZ3      21259 16:50:19.01 STC11011 00000090 +CSQX125I ZMQC CSQXSMFT SMF data incomplete
M 4000000 MPZ3      21259 16:50:19.10 STC11010 00000090 +CSQX123E ZMQ3 CSQXSMFT Failed to process channel initiator statistics,
S                                         424
E                               424 00000090 RC=00000024
N 4000000 MPZ3      21259 16:50:19.10 STC11010 00000090 +CSQX125I ZMQ3 CSQXSMFT SMF data incomplete
N 0000000 MPZ3      21259 16:50:20.66      00000281 IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=JOHNSON , ASID=004D.
N 8000000 MPZ1      21259 16:50:30.89 STC11087 00000090 DSNW133I -DSNA DSNWVSMF - TRACE DATA LOST, SMF NOT ACCESSIBLE RC=24
N 8000000 MPZ3      21259 16:50:31.08 STC11002 00000090 DSNW133I -DSNC DSNWVSMF - TRACE DATA LOST, SMF NOT ACCESSIBLE RC=24
N 8000000 MPZ3      21259 16:50:47.01 STC11002 00000090 DSNW123I -DSNC DSNWVSMF - TRACE RECORDING HAS BEEN RESUMED ON SMF
M 00A0000 MPZ3      21259 16:50:54.67 STC12869 00000090 ICH408I USER(USER1 ) GROUP(SYS1 ) NAME(          ) 429
D                               429 00000090 BBGZDFLT.zos.connect.access.roles.zosConnectAccess
D                               429 00000090 CL(EJBROLE )
D                               429 00000090 INSUFFICIENT ACCESS AUTHORITY
E                               429 00000090 ACCESS INTENT(READ ) ACCESS ALLOWED(NONE )
N 8000000 MPZ1      21259 16:51:00.70 STC11087 00000090 DSNW123I -DSNA DSNWVSMF - TRACE RECORDING HAS BEEN RESUMED ON SMF
N 8000000 MPZ1      21259 16:51:20.97 STC11087 00000090 DSNW133I -DSNA DSNWVSMF - TRACE DATA LOST, SMF NOT ACCESSIBLE RC=24
-----
```

Connected to remote server/host mpz3 using lu/pool MPZ30023 and port 23

Symptom: client see HTTP 403 – Authorization Failed. There were no messages in STDOUT or messages.log locations. Root cause – No READ access to EJBROLE BBGZDFLT.zos.connect.access.roles.zosConnectAccess.



Basic security issues – Sometimes there is misdirection

The STDOUT may show:

WG31

File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help

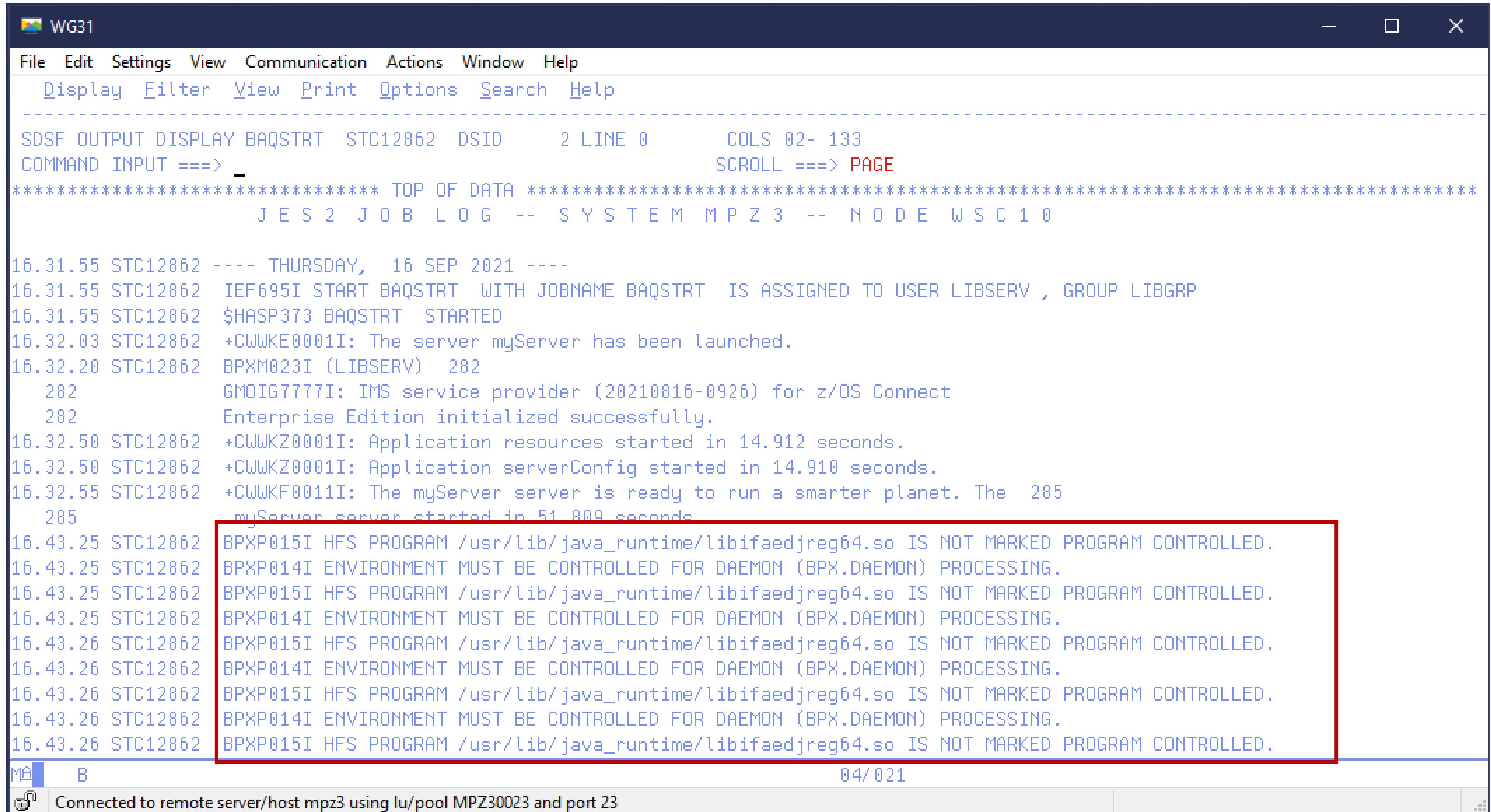
SDSF OUTPUT DISPLAY BAQSTRT STC12844 DSID 103 LINE 98 COLS 02- 133
COMMAND INPUT ==> SCROLL ==> PAGE

```
YAUDIT  " CWWKZ0001I: Application serverConfig started in 4.006 seconds.  
YAUDIT  " CWWKZ0001I: Application resources started in 4.007 seconds.  
YAUDIT  " CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/zosConnect/apiRequesters/  
YAUDIT  " CWWKT0016I: Web application available (default_host): http://dvipa.washington.ibm.com:9080/  
YAUDIT  " CWWKF0012I: The server installed the following features: YadmindCenter-1.0, apiDiscovery-1.0, appSecurity-2.0, distributed  
YAUDIT  " CWWKF0011I: The myServer server is ready to run a smarter planet. The myServer server started in 66.646 seconds.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
YAUDIT  " CWWKS1100A: Authentication did not succeed for user ID FRED. An invalid user ID or password was specified.  
***** BOTTOM OF DATA *****
```

MA B 04/021

Connected to remote server/host mpz3 using lu/pool MPZ30019 and port 23

Basic security issues - SYSLOG/JESMSGGLG output (even more misdirection)



```

WG31

File Edit Settings View Communication Actions Window Help
Display Filter View Print Options Search Help
-----+
SDSF OUTPUT DISPLAY BAQSTRT STC12862 DSID      2 LINE 0      COLS 02- 133
COMMAND INPUT ==> -
***** TOP OF DATA *****
J E S 2   J O B   L O G   --   S Y S T E M   M P Z 3   --   N O D E   W S C 1 0

16.31.55 STC12862 ---- THURSDAY, 16 SEP 2021 ----
16.31.55 STC12862 IEF695I START BAQSTRT WITH JOBNM BAQSTRT IS ASSIGNED TO USER LIBSERV , GROUP LIBGRP
16.31.55 STC12862 $HASP373 BAQSTRT STARTED
16.32.03 STC12862 +CWUJKE0001I: The server myServer has been launched.
16.32.20 STC12862 BPXM023I (LIBSERV) 282
282      GMOIG7777I: IMS service provider (20210816-0926) for z/OS Connect
282      Enterprise Edition initialized successfully.
16.32.50 STC12862 +CWUWKZ0001I: Application resources started in 14.912 seconds.
16.32.50 STC12862 +CWUWKZ0001I: Application serverConfig started in 14.910 seconds.
16.32.55 STC12862 +CWUWKF0011I: The myServer server is ready to run a smarter planet. The 285
285      myServer server started in 51.809 seconds
16.43.25 STC12862 BPXP015I HFS PROGRAM /usr/lib/java_runtime/libifaedjreg64.so IS NOT MARKED PROGRAM CONTROLLED.
16.43.25 STC12862 BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING.
16.43.25 STC12862 BPXP015I HFS PROGRAM /usr/lib/java_runtime/libifaedjreg64.so IS NOT MARKED PROGRAM CONTROLLED.
16.43.25 STC12862 BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING.
16.43.26 STC12862 BPXP015I HFS PROGRAM /usr/lib/java_runtime/libifaedjreg64.so IS NOT MARKED PROGRAM CONTROLLED.
16.43.26 STC12862 BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING.
16.43.26 STC12862 BPXP015I HFS PROGRAM /usr/lib/java_runtime/libifaedjreg64.so IS NOT MARKED PROGRAM CONTROLLED.
16.43.26 STC12862 BPXP014I ENVIRONMENT MUST BE CONTROLLED FOR DAEMON (BPX.DAEMON) PROCESSING.
16.43.26 STC12862 BPXP015I HFS PROGRAM /usr/lib/java_runtime/libifaedjreg64.so IS NOT MARKED PROGRAM CONTROLLED.

MA B          04/021
Connected to remote server/host mpz3 using lu/pool MPZ30023 and port 23

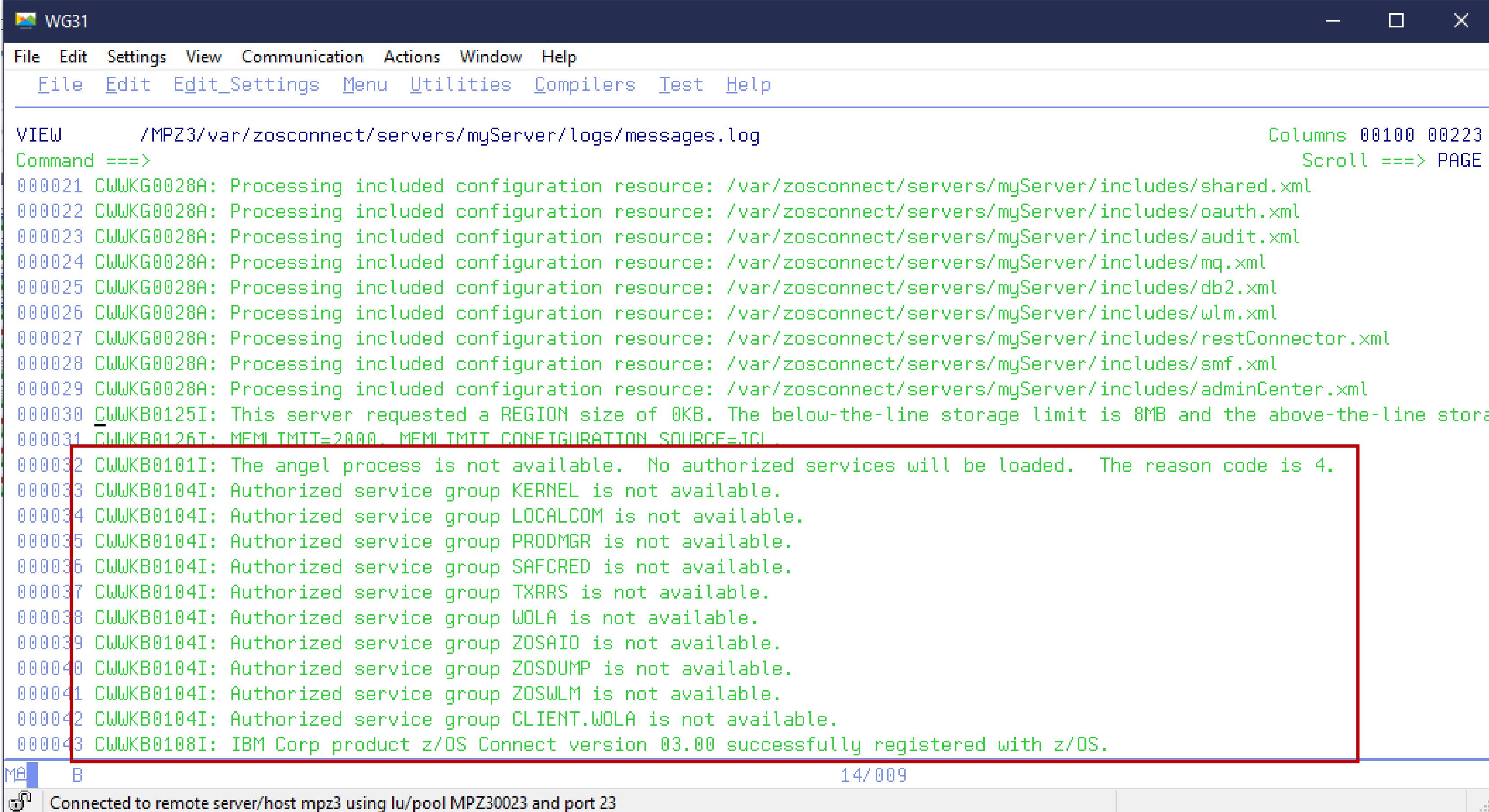
```

Symptom: Client unable to connect. STDOUT contains message *CWWKS1100A: Authentication did not succeed for user ID user1. An invalid user ID or password was specified.*



Basic security issues - SYSLOG/JESMSGGLG output (even more misdirection)

There is no need to set the extended protection attribute for this Java shared object executable.
The root cause was that the angel was not active.



```
VIEW      /MPZ3/var/zosconnect/servers/myServer/logs/messages.log          Columns 00100 00223
Command ==>
000021 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/shared.xml
000022 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/oauth.xml
000023 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/audit.xml
000024 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/mq.xml
000025 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/db2.xml
000026 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/wlm.xml
000027 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/restConnector.xml
000028 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/smf.xml
000029 CWUKG0028A: Processing included configuration resource: /var/zosconnect/servers/myServer/includes/adminCenter.xml
000030 CWWKB0125I: This server requested a REGION size of 0KB. The below-the-line storage limit is 8MB and the above-the-line stor
000031 CWWKB0126I: MFML TMFT=2000, MFML TMFT CONFIGURATION SOURCE=TCI
000032 CWWKB0101I: The angel process is not available. No authorized services will be loaded. The reason code is 4.
000033 CWWKB0104I: Authorized service group KERNEL is not available.
000034 CWWKB0104I: Authorized service group LOCALCOM is not available.
000035 CWWKB0104I: Authorized service group PRODMGR is not available.
000036 CWWKB0104I: Authorized service group SAFCRE is not available.
000037 CWWKB0104I: Authorized service group TXRRS is not available.
000038 CWWKB0104I: Authorized service group WOLA is not available.
000039 CWWKB0104I: Authorized service group ZOSAIO is not available.
000040 CWWKB0104I: Authorized service group ZOSDUMP is not available.
000041 CWWKB0104I: Authorized service group ZOSWLM is not available.
000042 CWWKB0104I: Authorized service group CLIENT.WOLA is not available.
000043 CWWKB0108I: IBM Corp product z/OS Connect version 03.00 successfully registered with z/OS.
```



External resource issues (HTTP 500)

The client sees:

```
HTTP/1.1 500 Internal Server Error
```

The STDOUT may show:

```
ÝWARNING `` BAQR0429W: API db2employee encountered an error while processing a request under URL  
https://mpz3.washington.ibm.com:9443/db2/employee/948478.
```

While the messages.log display

```
[9/16/21 21:00:55:811 GMT] 00000051 com.ibm.zosconnect.service.cics.internal.conn.ISCECIRequest E BAQR0657E: Transaction abend MIJO occurred in CICS while using CICS connection cscvinc and service cscvincDeleteService.  
[9/16/21 21:00:55:815 GMT] 00000051 com.ibm.zosconnect.internal.web.ServiceProxyServlet W BAQR0429W: API cscvinc encountered an error while processing a request under URL https://mpz3.washington.ibm.com:9443/cscvinc/employee/948478.
```

The STDOUT may show:

```
ÝWARNING `` BAQR0429W: API db2employee encountered an error while processing a request under URL  
https://mpz3.washington.ibm.com:9443/db2/employee/948478.
```

The messages.log displays:

```
[9/14/21 20:04:59:776 GMT] 00000048 osconnect.service.client.rest.internal.RestClientServiceImpl E BAQR0558E: The remote service invocation failed with [9/14/21 20:04:59:776 GMT] 00000048  
osconnect.service.client.rest.internal.RestClientServiceImpl E BAQR0558E: The remote service invocation failed with failed due to SQLCODE=-204 SQLSTATE=42704, USER1.EMPLOYEE IS AN UNDEFINED NAME. Error Location:DSNLJACC:35"
```



Tech-Tip: An HTTP 500 shortcut – look elsewhere

A HTTP status code 500 occurs when a failure occurred at an external endpoint. It does not matter if the external endpoint is a z/OS resources or a REST API provider, or an authorization server, etc.

The details of the failure may not be provided **directly** to z/OS Connect, just the fact that a failure has occurred. The failure could be a security issue, an abend or something entirely. z/OS Connect may or may not have directly access to any details of the failure (it depends on the service provider). It does not mean the details do not exist; the details are just readily available.

The shortcut to identify the issue is review the messages in the messages.log and check to see if there is corresponding FFDC (first failure data collection) dump.



What is a Java stack trace?

```
[9/6/21 22:51:19:981 GMT] 00000039 com.ibm.ejs.j2c.ConnectionEventListener
A J2CA0056I: The Connection Manager received
a fatal connection error from the Resource Adapter for resource null. The exception is: javax.resource.spi.EISSystemException: ICO0001E:
com.ibm.connector2.ims.ico.IMSTCPIPManagedConnection@c341a0aa.processOutputOTMAMsg(Connection, InteractionSpec, Record, Record) error. IMS
Connect returned an error: RETCODE=[4], REASONCODE=[NFNDDST] [Datastore not found.]
at com.ibm.connector2.ims.ico.IMSManagedConnection.processOutputOTMAMsg(IMSManagedConnection.java:4042)
at com.ibm.connector2.ims.ico.IMSTCPIPManagedConnection.callSendRecv(IMSTCPIPManagedConnection.java:241)
at com.ibm.connector2.ims.ico.IMSManagedConnection.call(IMSManagedConnection.java:1625)
at com.ibm.connector2.ims.ico.IMSConnection.call(IMSConnection.java:213)
at com.ibm.connector2.ims.ico.IMSInteraction.execute(IMSInteraction.java:586)
at com.ibm.ims.gateway.services.IMSGatewayServiceImpl.executeTranServiceInputTMRA(Unknown Source)
at com.ibm.ims.gateway.services.IMSGatewayServiceImpl.invokeTransactionService(Unknown Source)
at com.ibm.ims.gateway.services.IMSGatewayServiceImpl.invoke(Unknown Source)
at com.ibm.ims.zconnect.provider.clients.GatewayServiceClient.doPost(Unknown Source)
at com.ibm.ims.zconnect.provider.clients.IMSClient.doInvoke(Unknown Source)
at com.ibm.ims.gateway.config.services.IMSZServiceHandlerImpl.invoke(Unknown Source)
at com.ibm.ims.gateway.config.services.IMSZServiceImpl.invoke(Unknown Source)
at com.ibm.zosconnect.internal.ZosConnectServiceImpl.apiInvoke(Unknown Source)
at com.ibm.zosconnect.internal.ServiceManagerImpl.invoke(Unknown Source)
at com.ibm.zosconnect.internal.ApiManagerImpl.invokeApi(Unknown Source)
at com.ibm.zosconnect.internal.web.ServiceProxyServlet$3.run(Unknown Source)
at com.ibm.ws.webcontainer.async.ServiceWrapper.wrapAndRun(ServiceWrapper.java:236)
at com.ibm.ws.webcontainer.async.ContextWrapper.run(ContextWrapper.java:28)
at com.ibm.ws.webcontainer.async.WrapperRunnableImpl.run(WrapperRunnableImpl.java:89)
at com.ibm.ws.threading.internal.ExecutorServiceImpl$RunnableWrapper.run(ExecutorServiceImpl.java:238)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1160)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
at java.lang.Thread.run(Thread.java:825)
```

A J2CA0056I: The Connection Manager received
a fatal connection error from the Resource Adapter for resource null. The exception is: javax.resource.spi.EISSystemException: ICO0001E:
com.ibm.connector2.ims.ico.IMSTCPIPManagedConnection@c341a0aa.processOutputOTMAMsg(Connection, InteractionSpec, Record, Record) error. IMS
Connect returned an error: RETCODE=[4], REASONCODE=[NFNDDST] [Datastore not found.]

IMS service provider classes
z/OS Connect Java classes

A Google search of ICO00001E returned an explanation at URL: <https://www.ibm.com/docs/en/ims/13.1.0?topic=exceptions-ico0001e>

Root cause – Datastore mistyped in the interaction configuration

mitchj@us.ibm.com

© 2017, 2024 IBM Corporation
Slide 247



First Failure Data Collection (FFDC)

```
-----Start of DE processing----- = [9/7/21 14:19:29:291 GMT]
Exception = com.ibm.msg.client.jms.DetailedIllegalStateException
Source = com.ibm.zosconnect.service.mq.OneWayMQServiceInvocation
probeid = 0004
Stack Dump = com.ibm.msg.client.jms.DetailedIllegalStateException: JMSWMQ2002: Failed to get a message from destination 'ZCONN2.DEFAULT.MQZCEE.QUEUE'.
IBM MQ classes for JMS attempted to perform an MQGET; however IBM MQ reported an error.
Use the linked exception to determine the cause of this error.
at com.ibm.msg.client.wmq.common.internal.Reason.reasonToException(Reason.java:489)
at com.ibm.msg.client.wmq.common.internal.Reason.createException(Reason.java:215)
.
.
.
at com.ibm.zosconnect.service.mq.MQService.invoke(Unknown Source)
at com.ibm.zosconnect.internal.ZosConnectServiceImpl.apiInvoke(Unknown Source)
at com.ibm.zosconnect.internal.ServiceManagerImpl.invoke(Unknown Source)
at com.ibm.zosconnect.internal.ApiManagerImpl.invokeApi(Unknown Source)
at com.ibm.zosconnect.internal.web.ServiceProxyServlet$3.run(Unknown Source)
at com.ibm.ws.webcontainer.async.ServiceWrapper.wrapAndRun(ServiceWrapper.java:236)
at com.ibm.ws.webcontainer.async.ContextWrapper.run(ContextWrapper.java:28)
at com.ibm.ws.webcontainer.async.WrapperRunnableImpl.run(WrapperRunnableImpl.java:89)
at com.ibm.ws.threading.internal.ExecutorServiceImpl$RunnableWrapper.run(ExecutorServiceImpl.java:238)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1160)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
at java.lang.Thread.run(Thread.java:825)
Caused by: com.ibm.mq.MQException: JMSCMQ0001: IBM MQ call failed with compcode '2' ('MQCC_FAILED') reason '2016' ('MQRC_GET_INHIBITED').
at com.ibm.msg.client.wmq.common.internal.Reason.createException(Reason.java:203)
... 25 more
```

MQ service provider classes

Root cause – Queue was configured to disable the MQPUT request



The FFDC dump is more than just a Java stack trace

```
-----Start of DE processing----- = [9/7/21 20:26:12:394 GMT]
Exception = com.ibm.zosconnect.endpoint.connection.TokenConfigException
Source = com.ibm.zosconnect.endpoint.connection.internal.OAuthConfigImpl
probeid = 265
Stack Dump = com.ibm.zosconnect.endpoint.connection.TokenConfigException: BAQR1006E: An error occurred when z/OS Connect EE attempted to
access the authentication/authorization server. Error: javax.net.ssl.SSLHandshakeException: SSLHandshakeException invoking
https://wg31.washington.ibm.com:26213/oidc/endpoint/OP/token: com.ibm.jsse2.util.j: PKIX path building failed:
com.ibm.security.cert.IBMCertPathBuilderException: unable to find valid certification path to requested target
at com.ibm.zosconnect.endpoint.connection.internal.OAuthConfigImpl.requestAuthorizationServer(Unknown Source)
at com.ibm.zosconnect.endpoint.connection.internal.OAuthConfigImpl.getAuthData(Unknown Source)
at com.ibm.zosconnect.apirequester.internal.restclient.RestClientImpl.handleAuthConfig(Unknown Source)
at com.ibm.zosconnect.apirequester.internal.restclient.RestClientImpl.invoke(Unknown Source)
at com.ibm.zosconnect.apirequester.internal.ARInvokeHandler.handle(Unknown Source)
at com.ibm.zosconnect.apirequester.internal.ApiRequesterManagerImpl.invoke(Unknown Source)
at com.ibm.zosconnect.apirequester.internal.ApiRequesterManagerProxyImpl$1.run(Unknown Source)
.
.
.
Dump of callerThis
Object type = com.ibm.zosconnect.endpoint.connection.internal.OAuthConfigImpl
copyright_notice = "Licensed Materials - Property of IBM 5655-CE3 (c) Copyright IBM Corp. 2017, 2021 All Rights Reserved
tc = class com.ibm.websphere.ras.TraceComponent@2d85bcc
strings[0] = "TraceComponent[com.ibm.zosconnect.endpoint.connection.internal.OAuthConfigImpl, class
com.ibm.zosconnect.endpoint.connection.internal.OAuthConfigImpl, [zosConnectApiRequesterToken], com.ibm.zosconnect.endpoint
.connection.internal.resources.ZosConnectEndpointConnection, null]"
CFG_ELEMENT_ID = "id"
CFG_GRANTTYPE = "grantType"
id = "myoAuthConfig"
grantType = "password"
authServer = class com.ibm.zosconnect.endpoint.connection.internal.AuthorizationServerImpl@ed6c1e8c
.
.
.
sslCertsRef = "OutboundSSLSettings"
connectionTimeout = 30000
receiveTimeout = 60000
id = "myoAuthServer"
```

z/OS Connect Java classes



The FFDC dump for a network issue

```
-----Start of DE processing----- = [6/6/21 14:56:01:242 GMT]
Exception = java.net.UnknownHostException
Source = com.ibm.zosconnect.service.cics.internal.conn.isc.ConnectionManager
probeid = 131
Stack Dump = java.net.UnknownHostException: wg31.washington.ibm.com
at java.net.InetAddress.getAllByName0(InetAddress.java:1419)
at java.net.InetAddress.getAllByName(InetAddress.java:1323)
at java.net.InetAddress.getAllByName(InetAddress.java:1246)
at java.net.InetAddress.getByName(InetAddress.java:1196)
at com.ibm.zosconnect.service.cics.internal.conn.isc.ConnectionManager.createConnection(Unknown Source)
at com.ibm.zosconnect.service.cics.internal.conn.isc.ConnectionManager.getConnection(Unknown Source)
at com.ibm.zosconnect.service.cics.internal.conn.isc.SessionManager.getNewConversation(Unknown Source)
at com.ibm.zosconnect.service.cics.ServerECIRequest.executeISC(Unknown Source)
at com.ibm.zosconnect.service.cics.ServerECIRequest.execute(Unknown Source)
at com.ibm.zosconnect.service.cics.internal.CicsIpicConnection.flow(Unknown Source)
at com.ibm.zosconnect.service.cics.internal.CicsServiceImpl.flowRequest(Unknown Source)
at com.ibm.zosconnect.service.cics.internal.CicsServiceImpl.invoke(Unknown Source)
at com.ibm.zosconnect.internal.ZosConnectServiceImpl.apiInvoke(Unknown Source)
at com.ibm.zosconnect.internal.ServiceManagerImpl.invoke(Unknown Source)
at com.ibm.zosconnect.internal.ApiManagerImpl.invokeApi(Unknown Source)
```

Base Java classes
z/OS Connect Java classes

Root cause – Host wg31.washington.ibm.com was not configured in the DNS server

mitchj@us.ibm.com

© 2017, 2024 IBM Corporation
Slide 250

Use the messages.log and FFDC log together



The messages.log states a First Failure Data Collection dump of the issues has been created.

```
[9/12/21 14:56:45:613 GMT] 00000045 com.ibm.ws.logging.internal.impl.IncidentImpl           I FFDC1015I: An FFDC Incident has been  
created: "com.ibm.connector.DetailedResourceException: MQJCA1011: Failed to allocate a JMS connection., error code: MQJCA1011 An  
internal error caused an attempt to allocate a connection to fail. See the linked exception for details of the failure.  
com.ibm.ejs.j2c.poolmanager.FreePool.createManagedConnectionWithMCWrapper 199" at ffdc_21.09.12_14.56.45.0.log
```

```
[9/12/21 14:56:45:652 GMT] 00000045 com.ibm.ws.logging.internal.impl.IncidentImpl           I FFDC1015I: An FFDC Incident has been  
created: "com.ibm.msg.client.jms.DetailedJMSEException: MQJCA1011: Failed to allocate a JMS connection.
```

An internal error caused an attempt to allocate a connection to fail.

See the linked exception for details of the failure. com.ibm.zosconnect.service.mq.OneWayMQServiceInvocation 0004" at
ffdc_21.09.12_14.56.45.1.log

```
[9/12/21 14:56:45:652 GMT] 00000045 com.ibm.zosconnect.service.mq.MQServiceInvocation          E BAQM0056E: An unexpectedJMSEException  
occurred while processing a request for service 'mqGetService'. The exception message was 'MQJCA1011: Failed to allocate a JMS  
connection.'
```



The FFDC dump showing additional JMS information

```
-----Start of DE processing----- = [9/12/21 14:56:45:567 GMT]
Exception = com.ibm.mq.connector.DetailedResourceException
Source = com.ibm.ejs.j2c.poolmanager.FreePool.createManagedConnectionWithMCWrapper
probeid = 004
Stack Dump = com.ibm.mq.connector.DetailedResourceException: MQJCA1011: Failed to allocate a JMS connection., error code: MQJCA1011 An
internal error caused an attempt to allocate a connection to fail. See the linked exception for details of the failure.
at com.ibm.mq.connector.services.JCAExceptionBuilder.buildException(JCAExceptionBuilder.java:169)
at com.ibm.mq.connector.services.JCAExceptionBuilder.buildException(JCAExceptionBuilder.java:135)
at com.ibm.mq.connector.ConnectionBuilder.createConnection(ConnectionBuilder.java:162)
at com.ibm.mq.connector.outbound.ManagedConnectionFactoryImpl.createConnection(ManagedConnectionFactoryImpl.java:655)
at com.ibm.mq.connector.outbound.ManagedConnectionImpl.<init>(ManagedConnectionImpl.java:200)
at com.ibm.mq.connector.outbound.ManagedConnectionFactoryImpl.createManagedConnection(ManagedConnectionFactoryImpl.java:248)
at com.ibm.ejs.j2c.FreePool.createManagedConnectionWithMCWrapper(FreePool.java:1376)
at com.ibm.ejs.j2c.FreePool.createOrWaitForConnection(FreePool.java:1246)
at com.ibm.ejs.j2c.PoolManager.reserve(PoolManager.java:1438)
at com.ibm.ejs.j2c.ConnectionManager.allocateMCWrapper(ConnectionManager.java:574)
at com.ibm.ejs.j2c.ConnectionManager.allocateConnection(ConnectionManager.java:306)
at com.ibm.mq.connector.outbound.ConnectionFactoryImpl.createManagedJMSConnection(ConnectionFactoryImpl.java:309)
at com.ibm.mq.connector.outbound.ConnectionFactoryImpl.createConnectionInternal(ConnectionFactoryImpl.java:252)
at com.ibm.mq.connector.outbound.ConnectionFactoryImpl.createConnection(ConnectionFactoryImpl.java:225)
. . .
at java.lang.Thread.run(Thread.java:818)
Caused by: com.ibm.msg.client.jms.DetailedJMSEException: JMSFMQ6312: An exception occurred in the Java(tm) MQI.
The Java(tm) MQI has thrown an exception describing the problem.
See the linked exception for further information.
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
. . .
... 27 more
Caused by: com.ibm.mq.jmqi.JmqiException: CC=2;RC=2495;AMQ8568: The native JNI library 'mqjrrs64' was not found. For a client installation
this is expected. [3=mqjrrs64]
at com.ibm.mq.jmqi.local.LocalMQ.loadLib(LocalMQ.java:1178)
Caused by: java.lang.UnsatisfiedLinkError: /usr/lpp/mqm/V9R1M0/java/lib/libmqjrrs64.so (EDC5205S DLL module not found.)
```

Root cause – configuration issue in the MQ resource adapter configuration, e.g., nativeLibraryPath.

mitchj@us.ibm.com

© 2017, 2024 IBM Corporation
Slide 252

Tech/Tip: Details of the flow with mutual authentication (TLS 1.2)

1. A Client sends a request to server for a protected session in a ***ClientHello*** message. Included in the request is the TLS capabilities of the client (e.g., TLS 1.2 or 1.3) and a list of supported ciphers in preference order.
2. The server selects the TLS version and selects cipher from the list sent by the client and returns this information in a ***ServerHello*** message.
3. The server's certificate public information (including the **public key**) is sent to the client in a ***Certificate*** message.
4. The server sends cryptographic information for the client to use for encrypting a pre-master key in a ***Server key exchange*** message.
5. **For mutual authentication, the server sends a *CertificateRequest* message requesting a client's personal certificate.**
6. The server concludes by sending a ***ServerHelloDone*** message.
7. The client verifies the server's certificate with its trust store.
8. **If mutual authentication is requested, the client sends its public personal certificate information in a *Certificate* message**
9. The client then uses the **server's public key** to generate and encrypt a 48 byte "premaster secret" message which is sent to the server in a ***ClientKeyExchange*** message.
10. **When mutual authentication is requested, a digitally signature (hashed) of the concatenation of all previous handshake messages is encrypted with the client's private key sent in a *CertificateVerify* message.**
11. The ***Change Cipher*** message is used to change the cipher used during the handshake so all subsequent messages will be encrypted using a different cipher.
12. The server uses its **private key** to decrypt the "premaster secret" message (**only the private key can be used to decrypt the message**).
13. **If mutual authentication is requested, the server verifies the client's personal certificate with its key ring and uses the client's public key to decrypt and verify the message sent in the *CertificateVerify* message.**
14. Both the Client and Server use the "premaster secret" to compute a 'master secret', also known as "shared secret" or "session key" (symmetric encryption)
15. Client and server will use this "shared secret" or "session key" to encrypt messages sent between the endpoints.

Tech/Tip: cURL trace of a TLS Handshake

- * successfully set certificate verify locations:
 - * CAfile: certauth.pem
 - CApath: none
 - * TLSv1.3 (OUT), TLS handshake, Client hello (1):
 - * TLSv1.3 (IN), TLS handshake, Server hello (2):
 - * TLSv1.2 (IN), TLS handshake, Certificate (11):
 - * TLSv1.2 (IN), TLS handshake, Server key exchange (12):
 - * TLSv1.2 (IN), TLS handshake, Server finished (14):
 - * TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
 - * TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
 - * TLSv1.2 (OUT), TLS handshake, Finished (20):
 - * TLSv1.2 (IN), TLS handshake, Finished (20):
 - * SSL connection using TLSv1.2 / ECDHE-RSA-AES256-GCM-SHA384
 - * Server certificate:
 - * subject: O=IBM; OU=LIBERTY; CN=wg31.washington.ibm.com
 - * start date: Dec 23 04:00:00 2020 GMT
 - * expire date: Jan 1 03:59:59 2023 GMT
 - * common name: wg31.washington.ibm.com (matched)
 - * issuer: OU=LIBERTY; CN=CA for Liberty
 - * SSL certificate verify ok.
- * TLSv1.2 (IN), TLS handshake, Server key exchange (12):
 - ~~* TLSv1.2 (IN), TLS handshake, Request CERT (13):~~
 - * TLSv1.2 (IN), TLS handshake, Server finished (14):
 - ~~* TLSv1.2 (OUT), TLS handshake, Certificate (11):~~
 - * TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
 - ~~* TLSv1.2 (OUT), TLS handshake, CERT verify (15):~~
 - * TLSv1.2 (OUT), TLS change cipher, Change cipher spec (01):

TLS 1.2 <https://tools.ietf.org/html/rfc5246>

TLS 1.3 <https://tools.ietf.org/html/rfc8446>



A FFDC dump showing an SSL Handshake issue

```
. . . -----Start of DE processing----- = [6/16/21 17:59:45:534 GMT]
Exception = java.security.cert.CertPathValidatorException
Source = com.ibm.ws.ssl.core.WSX509TrustManager
probeid = checkServerTrusted
Stack Dump = java.security.cert.CertPathValidatorException: The certificate issued by CN=OpenIdProv, OU=CertAuth is not trusted; internal cause is: java.security.cert.CertPathValidatorException: Certificate chaining error
at com.ibm.security.cert.BasicChecker.<init>(BasicChecker.java:111)
at com.ibm.security.cert.PKIXCertPathValidatorImpl.engineValidate(PKIXCertPathValidatorImpl.java:220)
at java.security.cert.CertPathValidator.validate(CertPathValidator.java:278)
at com.ibm.jsse2.util.f.a(f.java:40)
at com.ibm.jsse2.util.f.b(f.java:143)
. .
e = class com.ibm.jsse2.util.f@5728f8dd
f = null
z = class java.lang.String[37]
tsCfgAlias = "OutboundKeyRing"
tsFile = "safkeyring:///zCEE.KeyRing"
extendedInfo = class java.util.HashMap@5ebd51b
serialVersionUID = 362498820763181265
```

Root cause – CA used to sign server certificate was not present in outbound key ring.

Tech-Tip: Use the Java JSSE debugging utility to enable SSL tracing at the Java level.

Use the Java runtime directive **-Djavax.net.debug** to enable this tracing by setting this directive value to **ssl**, e.g. **-Djavax.net.debug=ssl**. For more options regarding additional trace options SSL tracing available, see URL <https://www.ibm.com/docs/en/sdk-java-technology/8?topic=troubleshooting-debugging-utilities>

Using this directive requires the Java SDK be at Version 8, service release 6, fix pack 36 or later release level.



Tech/Tip: Use the Java directive javax.net.debug to enable Java SSL tracing

Add this directive to the JVM properties -Djavax.net.debug=ssl,handshake

```
.java:1168|JsseJCE: Using cipher DES/CBC/NoPadding from provider TBD via init
.java:1168|JsseJCE: Using cipher RC4 from provider TBD via init
.java:1168|JsseJCE: Using cipher DES/CBC/NoPadding from provider TBD via init
.java:1168|JsseJCE: Using cipher DESEde/CBC/NoPadding from provider TBD via init
-
-
-
.java:1168|JsseJCE: Using cipher AES/GCM/NoPadding from provider TBD via init
.java:1168|JsseJCE: Using cipher ChaCha20-Poly1305 from provider TBD via init
-
-
-
.java:1168|JsseJCE: Using KeyGenerator IbmTlsExtendedMasterSecret from provider TBD via init
.java:1168|JsseJCE: Using signature SHA1withECDSA from provider TBD via init
.java:1168|JsseJCE: Using signature NONEwithECDSA from provider TBD via init
-
-
-
.java:1168|Consuming ClientHello handshake message (
-
-
-
.java:1168|Consumed extension: supported_versions
.java:1168|Negotiated protocol version: TLSv1.2
-
-
-
.java:1168|Produced ServerHello handshake message (
-
-
-
.java:1168|Produced server Certificate handshake message (
-
-
-
.java:1168|Produced ECDH ServerKeyExchange handshake message (
-
-
-
.java:1168|Produced ServerHelloDone handshake message (
-
-
-
.java:1168|Consuming ECDHE ClientKeyExchange handshake message (
-
-
-
.java:1168|Consuming ChangeCipherSpec message
-
-
-
.java:1168|Consuming client Finished handshake message (
-
-
-
.java:1168|Produced ChangeCipherSpec message
.java:1168|Produced server Finished handshake message (
-
-
-
```

For more details, see URL <https://www.ibm.com/docs/en/sdk-java-technology/8?topic=troubleshooting-debugging-utilities>



Other common TLS handshake issues

- ***Error occurred during a read, exception:javax.net.ssl.SSLHandshakeException: null cert chain***

This exception occurs when the server configuration set to require client certificates (`clientAuthentication="true"`) and the client had no certificate to provide and no alternative authentication method was available.

- ***Error occurred during a read, exception:javax.net.ssl.SSLException: Received fatal alert: bad_certificate error (handshake), vc=1083934466
Caught exception during unwrap, javax.net.ssl.SSLException: Received fatal alert: bad_certificate***

This is usually caused when the client certificate presented to the server did not have a certificate authority(CA) certificate for the CA that signed the client's personal certificate in the server's trust store key ring.

- ***CWWKO0801E: Unable to initialize SSL connection. Unauthorized access was denied or security settings have expired. Exception is javax.net.ssl.SSLHandshakeException: no cipher suites in common***

- There may be many causes for this issue but first confirm the RACF identity under which the server is running has either READ access to FACILITY resources IRR.DIGTCERT.LISTRING and IRR.DIGTCERT.LIST or access to RDATALIB resources if virtual keyrings are being used.

The first FACILITY resource gives the identity access to their own key ring and the second allows access to the certificates. If if virtual keyrings are in use, then the identity needs READ or UPDATE authority to the <ringOwner>.<ringName>.LST resource in the RDATALIB class. READ access enables retrieving one's own private key, UPDATE access enables retrieving another's private key.

An alternative cause: For a TLS handshake to occur, the server must first have access to a private or site certificate that has a private key and the server must have access to that certificate's private key and no certificate with a private key is available.

- Another possibility is that the TLS handshake the negotiations between the client and server failed, e.g., `javax.net.ssl.SSLHandshakeException: Client requested protocol SSLv3 is not enabled or supported in server context`



trace.out – use as a last resort or at the request of Level 2

First, the current active trace specification settings can be display using the *restConnector* feature.

`https://mpz3.washington.ibm.com:9443/ibm/api/config/logging`



```
[{"configElementName": "logging", "appsWriteJson": false, "consoleFormat": "DEV", "consoleLogLevel": "AUDIT", "consoleSource": "message", "copySystemStreams": true, "isoDateFormat": false, "jsonAccessLogFields": "default", "jsonFieldMappings": "", "logDirectory": "/var/zosconnect/servers/myServer/logs", "maxFileSize": 20, "maxFiles": 2, "messageFileName": "messages.log", "messageFormat": "SIMPLE", "messageSource": "message", "suppressSensitiveTrace": false, "traceFileName": "trace.log", "tracerFormat": "ENHANCED", "traceSpecification": "*=info"}]
```

Enabling trace in z/OS Connect EE server

<https://www.ibm.com/docs/en/zosconnect/3.0?topic=problems-enabling-trace-in-zos-connect-ee>



Managing trace specifications

- Use “include” file to save commonly used trace specifications.
- Add the “include” after the sever has started to avoid tracing the startup activity.

server.xml

```
<include location="${server.config.dir}/includes/safTrace.xml"/>
```

safTrace.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="security trace">
<logging traceSpecification="com.ibm.ws.security.*=all:
    SSLChannel=all:SSL=all:zosConnectSaf=all:zosConnect=all"/>
</server>
```

cicsTrace.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="CICS trace">
<logging traceSpecification="zosConnectServiceCics=all:
    com.ibm.zosconnect.wv*=FINEST:zosConnect=all"/>
</server>
```

imsTrace.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="IMS trace">
<logging traceSpecification="com.ibm.ims.*=all:
    com.ibm.j2ca.RAIMSTM=all:com.ibm.zosconnect.wv*=FINEST:
    zosConnect=all"/>
</server>
```

Enables enhanced tracing

(after adding an “include” file)
F BAQSTART,REFRESH,CONFIG

Disable enhanced tracing

F BAQSTART,LOGGING='*=INFO'

Or

F BAQSTART,REFRESH,CONFIG
(after removing the “include” file)



trace.out file

```
mpz3
File Edit Settings View Communication Actions Window Help
File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT /MPZ3/var/zosconnect/servers/myServer/logs/trace.log Columns 00101 00252
Command ==> Scroll ==> PAGE
003697      > getSSLConfig: DefaultSSLSettings Entry
003698      < getSSLConfig Exit
003699      SSLConfig.toString() {
003700      - - - - - 4 Line(s) not Displayed
003701      > determineIfCSIV2SettingsApply Entry
003702      (com.ibm.ssl.remoteHost=*, com.ibm.ssl.direction=inbound, com.ibm.ssl.remotePort=9443, com.ibm.ssl.endPointName=defaultHttpEndpoint-ssl)
003703      < determineIfCSIV2SettingsApply (original settings) Exit
003704      - - - - - 43 Line(s) not Displayed
003705      3 keyStoreType: JCERACFKS
003706      3 trustStoreType: JCERACFKS
003707      - - - - - 44 Line(s) not Displayed
003708      3 keyStore: safkeyring://Liberty.KeyRing
003709      3 keyStoreName: CellDefaultKeyStore
003710      3 keyStorePassword: *****
003711      3 trustStore: safkeyring://Liberty.KeyRing
003712      3 trustStoreName: CellDefaultKeyStore
003713      3 trustStorePassword: *****
003714      - - - - - 2 Line(s) not Displayed
003715      (com.ibm.ssl.remoteHost=*, com.ibm.ssl.direction=inbound, com.ibm.ssl.remotePort=9443, com.ibm.ssl.endPointName=defaultHttpEndpoint-ssl)
003716      - - - - - 1 Line(s) not Displayed
004117 k      3 Error occurred during a read, exception:javax.net.ssl.SSLHandshakeException: Empty server certificate chain
004118      - - - - - 375 Line(s) not Displayed
004119      3 Caught exception during unwrap, javax.net.ssl.SSLHandshakeException: Empty server certificate chain
004120      - - - - - 1 Line(s) not Displayed
004121      (com.ibm.ssl.remoteHost=*, com.ibm.ssl.direction=inbound, com.ibm.ssl.remotePort=9443, com.ibm.ssl.endPointName=defaultHttpEndpoint-ssl)
004122      - - - - - 22 Line(s) not Displayed
004123      > isTransportSecurityEnabled Entry
004124      < isTransportSecurityEnabled true Exit
004125      - - - - - 1 Line(s) not Displayed
004126      > getSSLConfig: DefaultSSLSettings Entry
004127      < getSSLConfig Exit
004128      SSLConfig.toString() {
004129      - - - - - 4 Line(s) not Displayed
004130      > determineIfCSIV2SettingsApply Entry
004131      (com.ibm.ssl.remoteHost=*, com.ibm.ssl.direction=inbound, com.ibm.ssl.remotePort=9443, com.ibm.ssl.endPointName=defaultHttpEndpoint-ssl)
004132      < determineIfCSIV2SettingsApply (original settings) Exit
004133      - - - - - 43 Line(s) not Displayed
004134      3 keyStoreType: JCERACFKS
004135      3 trustStoreType: JCEPKEK
004136      - - - - - 44 Line(s) not Displayed
004137      3 keyStore: safkeyring://Liberty.KeyRing
004138      3 keyStoreName: CellDefaultKeyStore
004139      3 keyStorePassword: *****
004140      3 trustStore: safkeyring://Liberty.KeyRing
004141      3 trustStoreName: CellDefaultKeyStore
004142      3 trustStorePassword: *****
004143      - - - - - 2 Line(s) not Displayed
004144      (com.ibm.ssl.remoteHost=*, com.ibm.ssl.direction=inbound, com.ibm.ssl.remotePort=9443, com.ibm.ssl.endPointName=defaultHttpEndpoint-ssl)
004145      - - - - - 1 Line(s) not Displayed
004146      > isTransportSecurityEnabled Entry
004147      < isTransportSecurityEnabled true Exit
004148      - - - - - 4 Line(s) not Displayed
004149      > getSSLConfig: DefaultSSLSettings Entry
004150      < getSSLConfig Exit
004151      SSLConfig.toString() {
004152      - - - - - 43 Line(s) not Displayed
004153      > determineIfCSIV2SettingsApply Entry
004154      (com.ibm.ssl.remoteHost=*, com.ibm.ssl.direction=inbound, com.ibm.ssl.remotePort=9443, com.ibm.ssl.endPointName=defaultHttpEndpoint-ssl)
004155      < determineIfCSIV2SettingsApply (original settings) Exit
004156      - - - - - 44 Line(s) not Displayed
004157      3 keyStoreType: JCERACFKS
004158      3 trustStoreType: JCEPKEK
004159      - - - - - 2 Line(s) not Displayed
004160      3 keyStore: safkeyring://Liberty.KeyRing
004161      3 keyStoreName: CellDefaultKeyStore
004162      3 keyStorePassword: *****
004163      3 trustStore: safkeyring://Liberty.KeyRing
004164      3 trustStoreName: CellDefaultKeyStore
004165      3 trustStorePassword: *****
004166      - - - - - 1 Line(s) not Displayed
004167      (com.ibm.ssl.remoteHost=*, com.ibm.ssl.direction=inbound, com.ibm.ssl.remotePort=9443, com.ibm.ssl.endPointName=defaultHttpEndpoint-ssl)
004168      - - - - - 375 Line(s) not Displayed
004169 k      3 Error occurred during a read, exception:javax.net.ssl.SSLHandshakeException: Empty server certificate chain
004170      - - - - - 1 Line(s) not Displayed
004171      3 Caught exception during unwrap, javax.net.ssl.SSLHandshakeException: Empty server certificate chain
004172      - - - - - 22 Line(s) not Displayed
004173      (com.ibm.ssl.remoteHost=*, com.ibm.ssl.direction=inbound, com.ibm.ssl.remotePort=9443, com.ibm.ssl.endPointName=defaultHttpEndpoint-ssl)
004174      - - - - - 1 Line(s) not Displayed
004175      > isTransportSecurityEnabled Entry
004176      < isTransportSecurityEnabled true Exit
03/019
Connected to remote server/host mpz3 using lu/pool MPZ30006 and port 23
```

Use thread number and/or package name to control which trace records are displayed

Agenda

- Review the OMVS and Liberty basics
 - The OMVS, Java and Liberty execution environments
 - Creating Liberty servers
 - Managing the server XML configuration
- Liberty Security
 - Security provided by the Angel process
 - Liberty authentication and authorization using basic, digital certificates and third-party tokens
- Security when accessing z/OS subsystems
- Security when accessing non-z/OS subsystems
- Useful Liberty features and MVS commands
- Managing and Monitoring Liberty servers
 - WLM configurations
 - SMF options
 - Monitoring OMVS processes
 - Connection pooling options
 - Above the bar storage
 - High availability options
- Where do I look when things go wrong?
 - Problem determination techniques
 - Understand the anatomy of messages
- Appendix - Sample administrative JCL



z/OS Connect Wildfire Github Site <https://ibm.biz/BdPRGD>

The screenshot shows two GitHub repository pages side-by-side.

Left Repository: [ibm-wsc/zCONNEE-Wildfire-Workshop](#)

- Code tab is selected.
- Branch: master (1 branch)
- Tags: 0 tags
- Files:
 - AdminSecurity (circled in red)
 - OpenAPI2
 - cohal (circled in red)
 - xml (circled in red)
 - README.md
 - ZCADMIN - zOS Connect Administrat...
 - ZCEESEC - zOS Connect Security.pdf
 - ZCINTRO - Introduction to zOS Conn...
 - zOS Connect EE V3 Advanced Topics ...
 - zOS Connect EE V3 Getting Started.pdf
- README.md

Right Repository: [ibm-wsc/zCONNEE-Wildfire-Workshop](#) (Public)

- Code tab is selected.
- Branch: master (1 branch)
- Actions, Projects, Wiki, Security, Insights, Settings tabs are available.
- zCONNEE-Wildfire-Workshop / AdminSecurity /
- File list:
 - emitchj Add files via upload (e3f87ee on Apr 23) (History)
 - Customization Basic Configuration(1of2) (1).pdf
 - Customization Basic Configuration(1of2) (2).pdf
 - Customization Security and CICS.pdf
 - Customization Security and DB2.pdf
 - Customization Security and JWT Tokens.pdf
 - Customization Security and MQ.pdf
 - Customization Security when accessing an IMS Database....
 - Customization Security when accessing an IMS Transactio...
 - Customization Security with MVS Batch.pdf
 - admin

- Contact your IBM representative to schedule access to these exercises

© 2018, 2024 IBM Corporation

Page 262



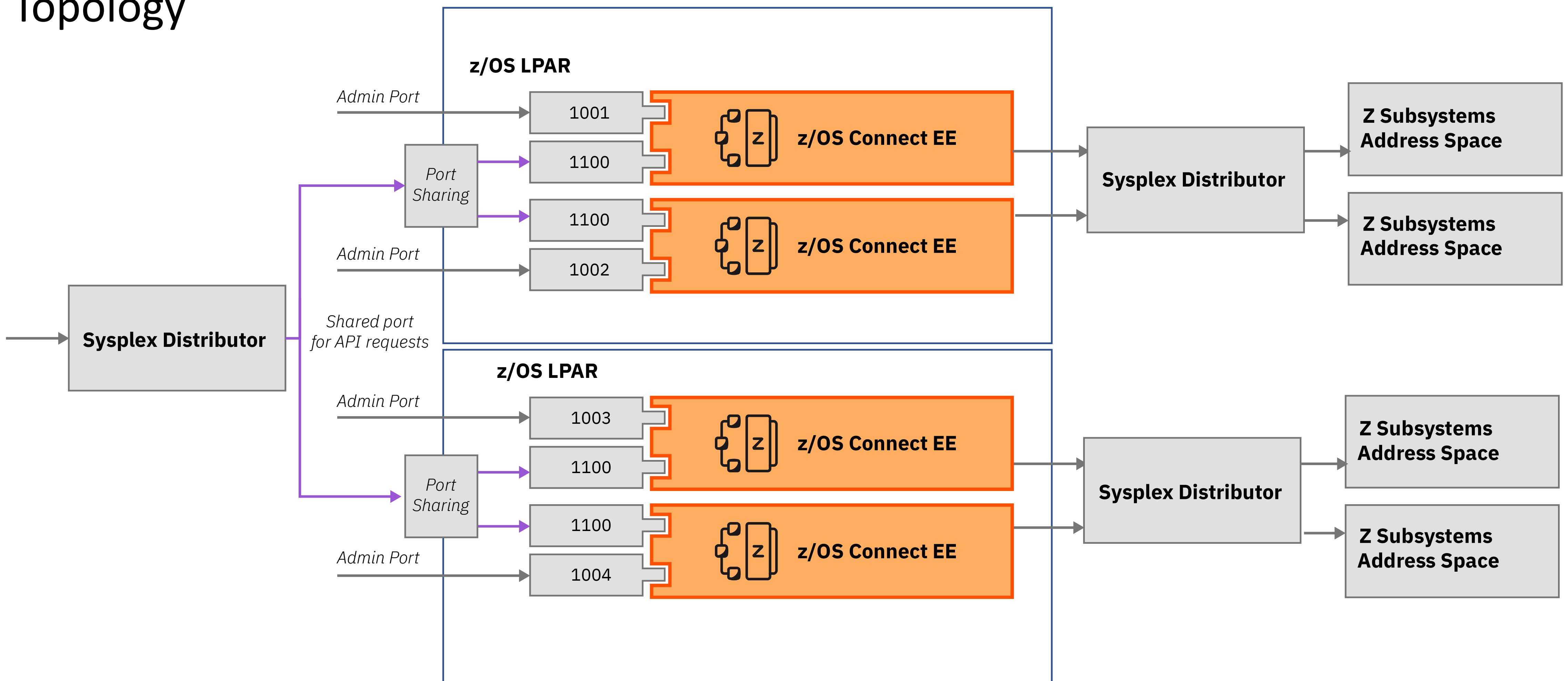
Thank you for listening and your questions.

Miscellaneous Odds and Ends



High Availability

- Topology



ibm.biz/zosconnect-ha-concepts

ibm.biz/zosconnect-scenarios

Sysplex DVIPAs



SYS1.TCPIP.TCPPARMS (IPNODES)

```
192.168.17.241 MPZ1.DMZ MPZ1 mpz1.washington.ibm.com  
192.168.17.242 MPZ2.DMZ MPZ2 mpz2.washington.ibm.com  
192.168.17.243 MPZ3.DMZ MPZ3 mpz3.washington.ibm.com  
192.168.17.240 dvipa dvipa.washington.ibm.com
```

SYS1.TCPIP.TCPPARMS (PROFMPZ3)

```
IPCONFIG SYSPLEXROUTING  
DYNAMICXCF 172.1.1.243 255.255.255.0 3  
VIPADYNAMIC  
VIPADEFINE 255.255.255.0 192.168.17.240  
VIPADISTRIBUTE DEFINE DISTM ROUNDROBIN|BASEWLM 192.168.17.240  
PORT 23 1416 1491 2446 9443 9453 9463  
DESTIP  
172.1.1.241  
172.1.1.242  
172.1.1.243  
ENDVIPADYNAMIC
```

SERVERWLM is not an option

HOMETEST

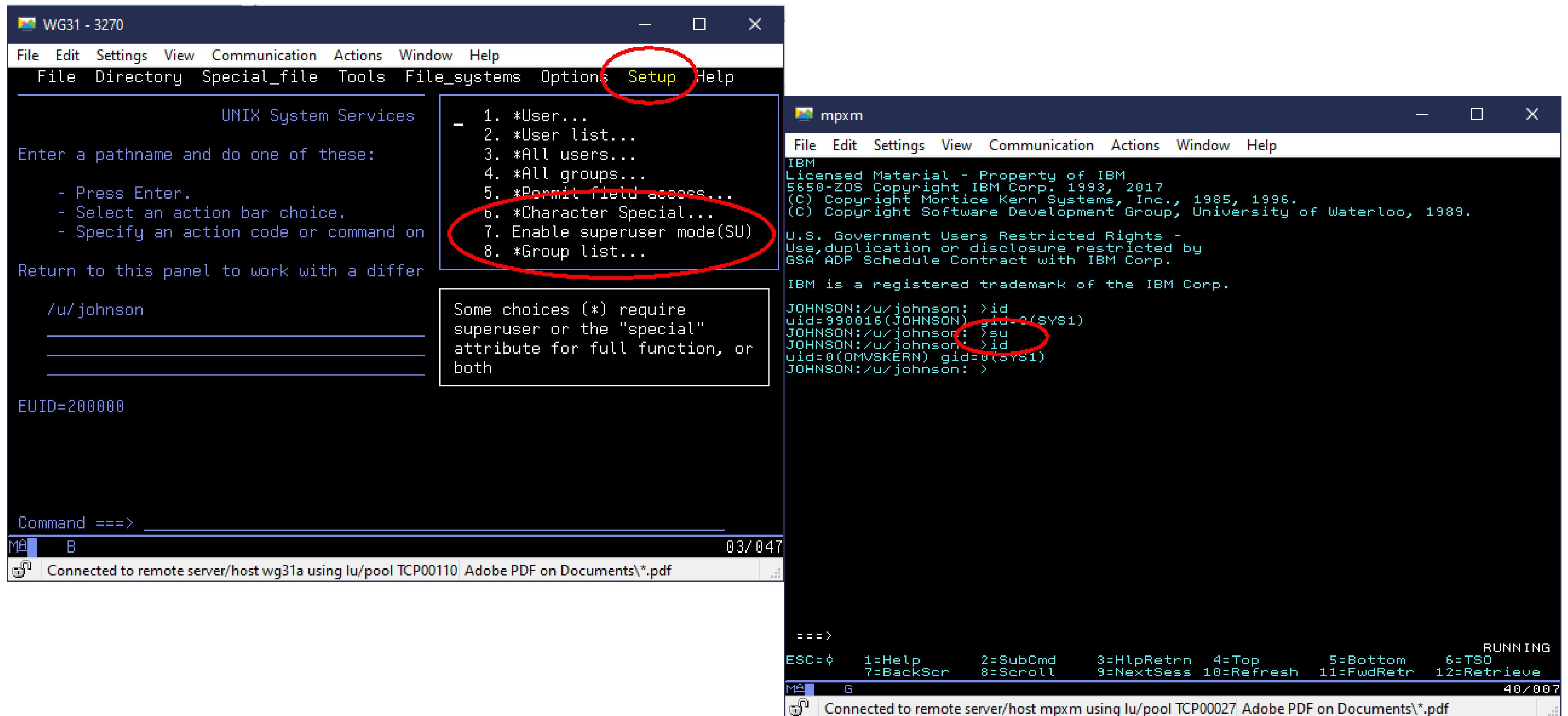
```
EZA0619I Running IBM MVS TCP/IP CS V2R4 TCP/IP Configuration Tester  
EZA0602I TCP Host Name is: MPZ3  
  
EZA0605I Using Name Server to Resolve MPZ3  
EZA0611I The following IP addresses correspond to TCP Host Name: MPZ3  
EZA0612I 192.168.17.243  
EZA0614I The following IP addresses are the HOME IP addresses defined in PROFILE.TCPIP:  
EZA0615I 192.168.17.243  
EZA0615I 172.1.1.243  
EZA0615I 192.168.17.240  
EZA0615I 127.0.0.1  
  
EZA0618I All IP addresses for MPZ3 are in the HOME list!  
EZA0622I Hometest was successful - all Tests Passed!
```

```
<zosconnect_cicsIpicConnection id="cscvinc"  
host="dvipa.washington.ibm.com"  
port="1491"/>  
<zosconnect_endpointConnection id="mqapi"  
host="http://dvipa.washington.ibm.com"  
port="9453"  
basicAuthRef="myBasicAuth"  
connectionTimeout="10s"  
receiveTimeout="20s" />
```

The screenshot shows a browser window titled "IBM REST API Documentation" with the URL <https://dvipa.washington.ibm.com:9443/api/explorer/>. The page displays a list of REST APIs under the heading "Liberty REST APIs". The APIs are categorized and listed as follows:

- cscvinc**:
 - POST /cscvinc/employee
 - DELETE /cscvinc/employee/{employee}
 - GET /cscvinc/employee/{employee}
 - PUT /cscvinc/employee/{employee}
- db2employee**:
 - Show/Hide | List Operations | Expand Operations
- filemgr**:
 - Show/Hide | List Operations | Expand Operations
- imsPhoneBook**:
 - Show/Hide | List Operations | Expand Operations
- jwtlvpDemoApi**:
 - Show/Hide | List Operations | Expand Operations
- miniloancics**:
 - Show/Hide | List Operations | Expand Operations
- mqapi**:
 - Show/Hide | List Operations | Expand Operations
- phonebook**:
 - Show/Hide | List Operations | Expand Operations

Tech/Tip: z/OS : Switching to root authority



Tech-Tip: Super user is required to set the program control extended attribute (`extattr +p`) bit for the Java shared object ***ifaedjreg64.so***. This extended attribute must be set for identity assertion in certain situations.

Sample JCL - Executing the z/OS Connect Build Toolkit on z/OS

```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=&SYSUID,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)  
//*****  
///* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET WORKDIR='/u/johnson/zconbt'  
// SET ZCONDIR='/usr/lpp/IBM/zosconnect/v3r0/zconbt/bin'  
//ZCONBT EXEC PGM=IKJEFT01,REGION=0M,MEMLIMIT=4G  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
  export WORKDIR=&WORKDIR; +  
  export ZCONDIR=&ZCONDIR; +  
  cd $WORKDIR; +  
  $ZCONDIR/zconbt.zos -p cscvinc.properties -f=cscvinc.ara; +  
  cp -v $WORKDIR/syslib/* //'JOHNSON.ZCONBT.COPYLIB'"
```

cscvinc.properties

```
apiDescriptionFile=./cscvinc.json  
dataStructuresLocation=./syslib  
apiInfoFileLocation=./syslib  
logFileDirectory=./logs  
language=COBOL  
connectionRef=cscvincAPI  
requesterPrefix=csc
```

This assumes the zconbt.zip files was expanded into directory /usr/lpp/IBM/zosconnect/v3r0/zconbt using command *jar -tf zconbt.zip* and that the property file and Swagger JSON document are encoded in ASCII in directory /u/johnson/zconbt.



Tech-Tip: Liberty's “adminCenter” Feature

- The Web browser interface feature “adminCenter” was used to display the server’s configuration files

The screenshot shows two side-by-side views of the "Server Config" tool in the IBM WebSphere Studio Application Developer interface. Both views are for the file "server.xml".

Left View (Design Tab):

- Header: "Server Config" and "server.xml".
- Tab: "Design" (highlighted with a red circle).
- Content:
 - Section: "Server".
 - Text: "Select an element on the left hand side tree to view its configuration."
 - Buttons: "Add child" and "Remove".
 - Section: "Description".
 - Text: "new server".

Right View (Source Tab):

- Header: "Server Config" and "server.xml".
- Tab: "Source" (highlighted with a red circle).
- Content:
 - Text: "Press Ctrl+space for content assist." and a gear icon.
 - Code listing:

```
1 <server description="new server">
2   <include location="/var/zosconnect/servers/myServer/resources/imsmobile-config/services/ims-services.xml" optional="true"/>
3   <include location="/var/zosconnect/servers/myServer/resources/imsmobile-config/interactions/ims-interactions.xml" optional="true"/>
4   <include location="/var/zosconnect/servers/myServer/resources/imsmobile-config/connections/ims-connections.xml" optional="true"/>
5   <include location="/var/zosconnect/servers/myServer/ims-admin-services.xml" optional="true"/>
6   <include location="${server.config.dir}/includes/safSecurity.xml"/>
7   <include location="${server.config.dir}/includes/safTrace.xml"/>
8   <include location="${server.config.dir}/includes/ipic.xml"/>
9   <include location="${server.config.dir}/includes/keyring.xml"/>
10  <include location="${server.config.dir}/includes/apiRequesterHTTPS.xml"/>
11  <include location="${server.config.dir}/includes/shared.xml"/>
12  <include location="${server.config.dir}/includes/oauth.xml"/>
13  <include location="${server.config.dir}/includes/audit.xml"/>
14  <include location="${server.config.dir}/includes/mq.xml"/>
15  <include location="${server.config.dir}/includes/db2.xml"/>
16  <include location="${server.config.dir}/includes/wlm.xml"/>
17  <include location="${server.config.dir}/includes/restConnector.xml"/>
18  <include location="${server.config.dir}/includes/smf.xml"/>
19  <include location="${server.config.dir}/includes/adminCenter.xml" />
20
21  <!-- Enable features -->
22  <featureManager>
23    <feature>apiDiscovery-1.0</feature>
24    <feature>zosconnect:zosConnect-2.0</feature>
25    <feature>zosconnect:zosConnectCommands-1.0</feature>
26    <feature>imsmobile:imsmobile-2.0</feature>
27  </featureManager>
28
29  <!-- To access this server from a remote client add a host attribute to the following element, e.g. host="*" -->
30  <httpEndpoint host="*" httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint"/>
31
32
```

Tech/Tip: Use the TCPIP resolver trace to display name resolution information

```
ALLOC FILE(SYSTCPT) DA(*)
ping wg31.washington.ibm.com
Resolver Trace Initialization Complete -> 2021/09/12 12:54:37.36

res_init Resolver values:
Setup file warning messages = No
CTRACE TRACERES option = No
Global Tcp/Ip Dataset = SYS1.TCPIP.TCPPARMS(TCPDAT3)
Default Tcp/Ip Dataset = SYS1.TCPIP.TCPPARMS(TCPDAT3)
Local Tcp/Ip Dataset = //DD:SYSTCPD
                         ==> SYS1.TCPIP.TCPPARMS(TCPDAT3)
Translation Table = SYS1.TCPIP.STANDARD.TCPXLBIN
UserId/JobName = JOHNSON
Caller API = TCP/IP Sockets Extended
Caller Mode = EBCDIC
System Name = WSC13 (from VMCF)
UnresponsiveThreshold = 25
(G) DataSetPrefix = SYS1.TCPIP
(G) HostName = MPZ3
. . .
res_query Failed: RetVal = -1, RC = 1, Reason = 0x78981005
res_querydomain Failed: RetVal = -1, RC = 1, Reason = 0x78981005
res_search Failed: RetVal = -1, RC = 1, Reason = 0x78981005
GetAddrInfo Closing IOCTL Socket 0x00000000
BPX1CLO: RetVal = 0, RC = 0, Reason = 0x00000000
GetAddrInfo Failed: RetVal = -1, RC = 1, Reason = 0x78AE1004
GetAddrInfo Ended: 2021/09/12 12:55:32.364732
*****
EZ2311I Unknown host 'WG31.WASHINGTON.IBM.COM'
```

Root cause – Host wg31.washington.ibm.com was missing from SYS1.TCPIP.TCPPARMS(IPNODES)

Sample JCL - Executing multiple OMVS commands in one step

```

//*****
///* SET SYMBOLS
//*****
//EXPORT EXPORT SYMLIST=(*)
// SET CURL= '/usr/lpp/rocket/curl'
//*****
///* CURL Procedure
//*****
//CURL PROC
//CURL EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
// PEND
//*****
///* STEP CURL - use curl to deploy API cscvinc
//*****
//DEPLOY EXEC CURL
BPXBATCH SH export CURL=&CURL; +
$CURL/bin/curl -X PUT -s +
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +
https://wg31.washington.ibm.com:9445/zosConnect/apis/cscvinc?status=stoped > null; +
$CURL/bin/curl -X DELETE -s +
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +
https://wg31.washington.ibm.com:9445/zosConnect/apis/cscvinc > null; +
$CURL/bin/curl -X POST -s +
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +
--data-binary @/u/johnson/cscvinc.aar +
--header "Content-Type: application/zip" +
https://wg31.washington.ibm.com:9445/zosConnect/apis
//*****
///* STEP CURL - use curl to invoke the API cscvinc
//*****
//INVOKE EXEC CURL
//SYSTSIN DD *,SYMBOLS=EXECSYS
BPXBATCH SH export CURL=&CURL; $CURL/bin/curl -X GET -s +
--cacert /u/johnson/CERTAUTH.PEM --user FRED:FRED +
https://wg31.washington.ibm.com:9445/cscvinc/employee/000100

```

Always be aware of the beginning and trailing spaces.

<https://www.rocketsoftware.com/platforms/ibm-z/curl-for-zos>



Sample JCL - Executing the Liberty *productInfo* command

```
//*****  
//* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET WLPDIR='/usr/lpp/IBM/zosconnect/v3r0/wlp'  
//PRODINFO EXEC PGM=IKJEFT01,REGION=0M,MEMLIMIT=4G  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
Export WLPDIR=&WLPDIR; +  
$WLPDIR/bin/productInfo version; +  
$WLPDIR/bin/productInfo featureInfo | grep cics; +  
$WLPDIR/bin/productInfo featureInfo | grep mq; +  
$WLPDIR/bin/productInfo featureInfo | grep ims; +  
$WLPDIR/bin/productInfo validate | grep 'Product validation'
```

```
productInfo featureInfo  
productInfo version  
productInfo validate
```

```
Product name: z/OS Connect  
Product version: 03.00.48  
Product edition: z/OS Connect Enterprise Edition
```

```
cicsService-1.0 ÿ1.0.0"  
wmqJmsClient-1.1 ÿ1.0.0"  
wmqJmsClient-2.0 ÿ1.0.0"  
Product Extension: mqzosconnect  
mqService-1.0 ÿ1.0.0"  
Product Extension: imsmobile  
imsmobile-2.0 ÿ2.0.0.202108160933"  
Product validation completed successfully.
```

Sample JCL - Copy WOLA executables from OMVS to a PDSE

```
//JOHNSONS JOB (ACCOUNT),JOHNSON,NOTIFY=JOHNSON,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)  
//*****  
///* SET SYMBOLS  
//*****  
//EXPORT EXPORT SYMLIST=(*)  
// SET DSNAME='USER1.WOLA2106.LOADLIB'  
// SET ZCEEPATH='/usr/lpp/IBM/zosconnect/v3r0'  
// SET JAVAHOME='/usr/lpp/java/J8.0_64'  
//*****  
///* Step ALLOC - Allocate a PDSE load library  
//*****  
//ALLOC EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *,SYMBOLS=EXECSYS  
DELETE '&DSNAME'  
SET MAXCC=0  
ALLOC DSNAME('&DSNAME') -  
    NEW CATALOG SPACE(2,1) DSORG(PO) CYLINDERS -  
    RECFM(U) DSNTYPE(LIBRARY)  
//*****  
///* Step WOLACOPY - copy the WOLA executables to the PDSE  
//*****  
//WOLACOPY EXEC PGM=IKJEFT01,REGION=0M  
//SYSTSPRT DD SYSOUT=*  
//SYSERR DD SYSOUT=*  
//STDOUT DD SYSOUT=*  
//SYSTSIN DD *,SYMBOLS=EXECSYS  
BPXBATCH SH +  
    export JAVA_HOME=&JAVAHOME; +  
    export DSNAME=&DSNAME; +  
    cp -Xv &ZCEEPATH/wlp/clients/zos/* "//$DSNAME"
```



Sample JCL - BBOSMFV (Extract Liberty SMF 120 Subtype 11 records)

```
//JOHNONS JOB (ACCOUNT),JOHNSON,NOTIFY=JOHNSON,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1, 1)  
//EXPORT EXPORT SYMLIST=(*  
// SET REPORT='LibertyExport'  
//JAVA EXEC PROC=JVMPRC86,  
// JAVACLS='com.ibm.ws390.sm.smfview.JclSmf'  
//STDENV DD DISP=SHR,DSN=JOHNSON.JCLLIB.CNTL(STDENV)  
//SMFDATA DD DISP=SHR,DSN=MPZ3.DUMPSMF  
//SMFENV DD *,SYMBOLS=EXEC SYS  
# Specify the plugin to use  
plugin=&REPORT  
# Specify where the output goes  
output=/u/johnson/&REPORT..csv  
# Uncomment (and change the value as appropriate) to filter  
#matchServer=BAQSTRT
```

```
JOHNSON.JCLLIB.CNTL(STDENV)  
. /etc/profile  
export JAVA_HOME=/usr/lpp/java/J8.0_64  
export PATH=/bin:"${JAVA_HOME}"/bin  
  
LIBPATH=/lib:/usr/lib:"${JAVA_HOME}"/bin  
LIBPATH="$LIBPATH":"${JAVA_HOME}"/lib/s390x  
LIBPATH="$LIBPATH":"${JAVA_HOME}"/lib/s390x/j9vm  
LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin/classic  
export LIBPATH="$LIBPATH":  
  
# Customize your CLASSPATH here  
APP_HOME=$JAVA_HOME  
CLASSPATH=$APP_HOME:"${JAVA_HOME}"/lib:"${JAVA_HOME}"/lib/ext  
CLASSPATH=/u/johnson/lib/bbosmfv.jar:$CLASSPATH  
CLASSPATH=/u/johnson/lib/WP102312_Plugins.jar:$CLASSPATH  
  
# Add Application required jars to end of CLASSPATH  
for i in "${APP_HOME}"/*.jar; do  
    CLASSPATH="$CLASSPATH":$i"  
done  
export CLASSPATH="$CLASSPATH":  
  
# Configure JVM options  
IJO="-Xms16m -Xmx128m"  
export IBM_JAVA_OPTIONS="$IJO "
```

Sample JCL – Using ADRDSSU to dump/restore MVS data sets



ZCEEDUMP

```
//EXPORT EXEC SYMLIST=(*)
// SET ZCEELVL=349
//DELETE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *,SYMBOLS=EXECSYS
      DELETE IBM.ZCEE30.BKUP&ZCEELVL.
      SET MAXCC=0
//DUMP      EXEC PGM=ADRDSSU,REGION=2048K
//SYSPRINT DD SYSOUT=*
//DUMPDD DD DSN=IBM.ZCEE30.BKUP&ZCEELVL.,
//          DISP=(NEW,CATLG),
//          UNIT=SYSDA,SPACE=(CYL,(3000,2000,0),RLSE)
//SYSIN    DD *,SYMBOLS=EXECSYS
      DUMP DATASET(INCLUDE(
      ZCEE30.SBAQ* -
      ZCEE30.WOLA*.** -
      OMVS.ZCEE*.** -
      )) OPTIMIZE(4) OUTDDNAME(DUMPDD) TOLERATE(ENQF)
```

ZCEERSTR

```
//RESTORE EXEC PGM=ADRDSSU,REGION=2048K
//SYSPRINT DD SYSOUT=*
//DUMPDD DD DISP=SHR,DSN=JOHNSON.ZCEE30.BKUP349
//SYSIN    DD *
      RESTORE DATASET(INCLUDE(**)) -
      INDDNAME(DUMPDD) OUTDYNAM(WAS004) -
      NULLSTORCLAS -
      REPLACE CATALOG TOLERATE(ENQF)
```

Sample JCL – Define and format a ZFS data set using IOEAGFMT

ZFS

```
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYSIN    DD *
      SET MAXCC=0
      DEFINE CLUSTER (NAME(OMVS.ZCEE.GROUP1.ZFS) -
                      LINEAR CYLINDERS(100 100) SHAREOPTIONS(3))
//CREATE EXEC PGM=IOEAGFMT,REGION=0M,
//  PARM=( '-aggregate OMVS.ZCEE.GROUP1.ZFS -compat' )
//SYSPRINT DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
```

Sample JCL – Generate WLM Workload Activity Reports

```
//JOHNONS JOB (ACCOUNT),NOTIFY=&SYSUID,REGION=0M,  
// CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)  
//DELETE EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
    DELETE JOHNSON.DUMPSMF.SORT  
//RMFSORT EXEC PGM=SORT,REGION=0M  
//SORTIN DD DISP=SHR,DSN=MPZ3.DUMPSMF  
//SORTOUT DD DISP=(,CATLG),DSN=JOHNSON.DUMPSMF.SORT,  
//           SPACE=(CYL,(100,50),RLSE),UNIT=SYSDA  
//SORTWK01 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(100))  
//SORTWK02 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(100))  
//SORTWK03 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(100))  
//SORTWK04 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(100))  
//SORTWK05 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(100))  
//SORTWK06 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(100))  
//SORTWK07 DD DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(100))  
//SYSPRINT DD SYSOUT=(,)  
//SYSOUT DD SYSOUT=(,)  
//SYSIN DD *  
    SORT FIELDS=(11,4,CH,A,7,4,CH,A),EQUALS  
    MODS E15=(ERBPPE15,36000,,N),E35=(ERBPPE35,3000,,N)  
//RMFPP EXEC PGM=ERBRMFPP,REGION=0M  
//SYSUDUMP DD SYSOUT=*  
//STEPLIB DD DSN=SYS1.COMBINED.LINKLIB,DISP=SHR  
//MFPIINPUT DD DISP=SHR,DSN=JOHNSON.DUMPSMF.SORT  
//MFPMMSGDS DD SYSOUT=*  
//SYSIN DD *  
    SYSOUT(O)  
    SYSRPTS(WLMGL(RCPER)) /*WORKLOAD ACTIVITY REPORT */
```



BAQSMFP output (OpenAPI 2)

```
*****
* SMF123.1 V2 Request Data Section *
*****
SMF123S1_REQ_TYPE = API (1)
SMF123S1_HTTP_RESP_CODE = 500
SMF123S1_REQ_TIMED_OUT = NO
SMF123S1_USER_NAME = FRED
SMF123S1_USER_NAME_MAPPED =
SMF123S1_CLIENT_IP_ADDR = 192.168.0.60
SMF123S1_API_NAME = db2employee
SMF123S1_API_VERSION = 1.0.0
SMF123S1_SERVICE_NAME = selectEmployee
SMF123S1_SERVICE_VERSION = 1.0.0
SMF123S1_REQ_METHOD = GET
SMF123S1_REQ_QUERY_STR =
SMF123S1_REQ_TARGET_URI = /db2/employee/000010
SMF123S1_REQ_PAYLOAD_LEN = 0
SMF123S1_RESP_PAYLOAD_LEN = 0
SMF123S1_TIME_ZC_ENTRY = 0x00DA2FB8 38ED5494 04000000 08880001
UTC_CONV_TIME_ZC_ENTRY = 2021/08/19 15:30:24.905545 UTC
SMF123S1_TIME_ZC_EXIT = 0x00DA2FB8 38F3883F A8000000 08880001
UTC_CONV_TIME_ZC_EXIT = 2021/08/19 15:30:24.930947 UTC
SMF123S1_TIME_SOR_SENT = 0x00DA2FB8 38F232A9 76000000 08A00001
UTC_CONV_TIME_SOR_SENT = 2021/08/19 15:30:24.925482 UTC
SMF123S1_TIME_SOR_RECV = 0x00DA2FB8 38F300A4 AA000000 08880001
UTC_CONV_TIME_SOR_RECV = 2021/08/19 15:30:24.928778 UTC
SMF123S1_SP_NAME = restclient-1.0
SMF123S1_SOR_REFERENCE = Db2Conn
SMF123S1_SOR_IDENTIFIER = Db2:DSN2LOC,wg31.washington.ibm.com:2446
SMF123S1_SOR_RESOURCE = services/zCEEService/selectEmployee
SMF123S1_REQ_ID = 302
SMF123S1_TRACKING_TOKEN = 0x42415131 77734859 41514159 314E6670 31395046
35304455 312B6E7A 51454241
514E6F76 75446A74 564A5145 41413D3D 40404040 40404040 40404040
SMF123S1_REQ_HDR1 =
SMF123S1_REQ_HDR2 =
SMF123S1_REQ_HDR3 =
SMF123S1_REQ_HDR4 =
SMF123S1_RESP_HDR1 =
SMF123S1_RESP_HDR2 =
SMF123S1_RESP_HDR3 =
```

```
*****
* SMF123.2 V2 Request Data Section *
*****
SMF123S2_REQ_APP_TYPE = ZOS (3)
SMF123S2_HTTP_RESP_CODE = 200
SMF123S2_REQ_STATUS_CODE = 200
SMF123S2_REQ_RETRY = NO
SMF123S2_REQ_PAYLOAD_LEN = 0
SMF123S2_RESP_PAYLOAD_LEN = 269
SMF123S2_USER_NAME = USER1
SMF123S2_USER_NAME_MAPPED =
SMF123S2_USER_NAME_ASSERTED = USER1
SMF123S2_API_REQ_NAME = cscvinc_1.0.0
SMF123S2_API_REQ_VERSION = 1.0.0
SMF123S2_ENDPOINT_REFERENCE = cscvincAPI
SMF123S2_ENDPOINT_HOST = https://mpz3.washington.ibm.com
SMF123S2_ENDPOINT_PORT = 9463
SMF123S2_ENDPOINT_FULL_PATH = /cscvinc/employee/111111
SMF123S2_ENDPOINT_METHOD = GET
SMF123S2_ENDPOINT_QUERY_STR =
SMF123S2_TIME_STUB_SENT = 0x00DA2FC1 7D34CE8B 4A000000 084C0001
UTC_CONV_TIME_STUB_SENT = 2021/08/19 16:11:52.420584 UTC
SMF123S2_TIME_ZC_ENTRY = 0x00DA2FC1 7D58AEE0 0E000000 08A00001
UTC_CONV_TIME_ZC_ENTRY = 2021/08/19 16:11:52.567534 UTC
SMF123S2_TIME_ZC_EXIT = 0x00DA2FC1 87DCB806 E6000000 08880001
UTC_CONV_TIME_ZC_EXIT = 2021/08/19 16:12:03.594112 UTC
SMF123S2_TIME_TOKEN_GET_START = 0x00DA2FC1 7D59D3A6 E6000000 08A00001
UTC_CONV_TIME_TOKEN_GET_START = 2021/08/19 16:11:52.572218 UTC
SMF123S2_TIME_TOKEN_GET_FINISH = 0x00DA2FC1 7D59DF85 CC000000 088C0001
UTC_CONV_TIME_TOKEN_GET_FINISH = 2021/08/19 16:11:52.572408 UTC
SMF123S2_TIME_ENDPOINT_SENT = 0x00DA2FC1 7D5A0328 04000000 088C0001
UTC_CONV_TIME_ENDPOINT_SENT = 2021/08/19 16:11:52.572978 UTC
SMF123S2_TIME_ENDPOINT_RECEIVED = 0x00DA2FC1 87DC8216 58000000 08880001
UTC_CONV_TIME_ENDPOINT_RECEIVED = 2021/08/19 16:12:03.593249 UTC
SMF123S2_MVS_JOBNAME = USER1GE2
SMF123S2_MVS_JOBID = JOB09543
SMF123S2_MVS_SYSNAME = MPZ3
SMF123S2_MVS_ASID = 54
SMF123S2_MVS_SID = MPZ3
SMF123S2_REQ_ID = 732
SMF123S2_TRACKING_TOKEN = 0x42415131 77734859 41514159 314E6670 31395046
514E6F76 77583159 7275414F 40404040 40404040 40404040 40404040
SMF123S2_REQ_HDR1 =
SMF123S2_REQ_HDR2 =
SMF123S2_REQ_HDR3 =
```

CICS Performance Analyzer

V5R4M0		CICS Performance Analyzer z/OS Connect Summary							
ZCEE0001 Printed at 13:35:01 8/21/2021		Data from 11:30:24 8/19/2021 to 12:11:24 8/19/2021				Page 1			
Initial CICS PA report									
JOBNAME : BAQSTRT SPNAME : imsmobile-2.0									
Request:	49	Fail:	0	Timed out:	0	Get:	49 Post:		
							0 Put: 0 Delete: 0		
----- Maximum value Request details -----									
SOR Sent Latency	.0326	Avg	.3781	Max	551	ZC Entry	19/08/2021 12:09:45.036778		
SOR Response	.0016		.0183		551		19/08/2021 12:09:45.036778		
ZC Exit Latency	.0025		.0048		504		19/08/2021 12:09:36.823661		
ZC Response	.0367		.3982		551		19/08/2021 12:09:45.036778		
ZC Time	.0351		.3799		551		19/08/2021 12:09:45.036778		
JOBNAME : BAQSTRT SPNAME : restclient-1.0									
Request:	50	Fail:	50	Timed out:	0	Get:	50 Post:		
							0 Put: 0 Delete: 0		
----- Maximum value Request details -----									
SOR Sent Latency	.0478	Avg	.5953	Max	488	ZC Entry	19/08/2021 12:09:33.386614		
SOR Response	.0027		.0127		594		19/08/2021 12:09:52.016624		
ZC Exit Latency	.0014		.0029		524		19/08/2021 12:09:40.369997		
ZC Response	.0519		.6004		488		19/08/2021 12:09:33.386614		
ZC Time	.0492		.5972		488		19/08/2021 12:09:33.386614		
JOBNAME : BAQSTRT SPNAME : CICS-1.0									
Request:	49	Fail:	0	Timed out:	0	Get:	49 Post:		
							0 Put: 0 Delete: 0		
----- Maximum value Request details -----									
SOR Sent Latency	.0300	Avg	.0589	Max	450	ZC Entry	19/08/2021 12:09:26.478282		
SOR Response	.0011		.0049		517		19/08/2021 12:09:39.019456		
ZC Exit Latency	.0077		.0138		450		19/08/2021 12:09:26.478282		
ZC Response	.0387		.0741		450		19/08/2021 12:09:26.478282		
ZC Time	.0376		.0727		450		19/08/2021 12:09:26.478282		

IBM z Omegamon for JVM

The image displays three windows from the IBM z Omegamon for JVM interface:

- Top Left Window:** "z/OS Connect Request Summary" (KJJZCSA). It shows a summary of requests over the last 30 minutes. The summary table includes columns for API Name, Service, SoR ID, Reference, and Resource. Below the table, it lists specific requests for services like catalog, cscvinc, db2employee, filea, filequeue, *ADMIN*, and phonebook.
- Top Right Window:** "Requests by Service Name" (KJJZCSS). It shows a summary of requests by service name over the last 30 minutes. The table includes columns for API Name, Service, SoR ID, Reference, and Resource. Below the table, it lists specific requests for inquireSingle, cscvincService, and selectEmployee.
- Bottom Window:** "z/OS Connect Request Detail" (KJJZCDD). This window provides a detailed log of a single request. The log entries include:


```

Event time..... 04/02/19 18:47:54.267
Request Type... API
API name..... cscvinc
Request URI... /cscvinc/employee/444444
Query String...
Method.... GET
Port..... 9453
HTTP code.... 200 (OK)
Timeout..... No
Service Name... cscvincService
Total Req Time.. 0.008006s
z/OS Conn Time.. 0.005515s
SoR Resp Time... 0.002491s
SoR ID..... USIBMWZ.CICS53Z
SoR Ref..... cscvinc
SoR Resource.... CSMI,CSCVINC
Remote Address.. 192.168.0.141
Request Length... 0
Response Length.. 302
Correlator..... e6e2d3d7d3c5e7400011000010d5ea50
Operation..... getCscvincService
Provider..... CICS-1.0
User ID..... Fred
      
```

IBM z Omegamon for JVM

The image displays four windows of the IBM z Omegamon for JVM tool, each showing a different network connection detail.

- Top Left Window:** Shows a "z/OS Connect Request Detail" for a filequeue API. The request was made via IBM MQ for z/OS at port 9453, resulting in a 200 (OK) response. The service name is FileaQueue.
- Top Right Window:** Shows a "z/OS Connect Request Detail" for a cscvinc API. The request was made via IBM MQ for z/OS at port 9453, resulting in a 200 (OK) response. The service name is cscvincService.
- Bottom Left Window:** Shows a "z/OS Connect Request Detail" for a db2employee API. The request was made via IBM MQ for z/OS at port 9453, resulting in a 200 (OK) response. The service name is selectEmployee.
- Bottom Right Window:** Shows a "z/OS Connect Request Detail" for a phonebook API. The request was made via IBM MQ for z/OS at port 9453, resulting in a 200 (OK) response. The service name is ivtnoService.

Each window includes a header with the session ID (WG31 - 3270), date/time (e.g., 04/02/2019 18:59:29), and various configuration parameters like Auto Update and SMF ID. The bottom of each window shows a status bar indicating a connection to a remote server/host using TCP port 23.