An integrated Single Sign-On Solution with Linux on z Systems, z/OS, and Microsoft Active Directory

Linux on z Systems Live Virtual Class | 07 December 2016

Marc Beyerle (marc.beyerle@de.ibm.com)

IBM Mainframe Specialist, Senior Java Performance Engineer

Deck version: 1.0
Deck date: 2016-12-07

IBM **Client Center** - Systems and Software, IBM Germany Lab

Trademarks

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

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

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

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

*BladeCenter®, CICS®, DataPower®, DB2®, e business(logo)®, ESCON, eServer, FICON®, IBM®, IBM (logo)®, IMS, MVS, OS/390®, POWER6®, POWER6+, POWER7®, Power Architecture®, PowerVM®, PureFlex, PureSystems, S/390®, ServerProven®, Sysplex Timer®, System p®, System p5, System x®, System z®, System z9®, System z10®, WebSphere®, X-Architecture®, z13™, z Systems®, z9®, z10, z/Architecture®, z/OS®, z/VM®, z/VSE®, zEnterprise®, zSeries®

The following are trademarks or registered trademarks of other companies.

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

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

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

Agenda

- Introduction & background
- Phase 1 (workshop): evaluating different integration technologies
- Phase 2 (Proof of Concept): implementing the proposed architecture
- Summary

Introduction & background

- **The customer is a *medium-sized European bank*, which is mostly active in its home country**
  - In its home country, the bank has a wide network of branches

- **Early 2015, the client decided to *modernize the front-end* for the *branch part* of their core banking solution**
  - Current application is based on Smalltalk, IBM® VisualAge® Generator, and 3270 screens
  - Goal: transition to a more "modern" front-end, which should be web- / browser-based

- **The bank wanted to evaluate several different options and their decision basically came down to two approaches: one based on *Microsoft® .NET®* running on *Intel® x86* and the other on *WebSphere® Application Server* (WAS) on *Linux® on z Systems®***
  - Client has experience in both environments and asked Microsoft and IBM for presenting their respective solutions
  - Requirement: both solutions had to be able to integrate seamlessly with the existing CICS® and DB2® for z/OS® back-end

Introduction & background, *continued*

- ***Important*: at this point in time, the client hadn't mentioned that they were actually looking for a *Single Sign-On* (SSO) solution**
  - In the preparation phase of the workshop, the client only stated that ***performance*** and ***security*** were their ***key concerns*** for the new front-end solution
  - Therefore, focus was put on performance and security of the the different integration technologies

- **Performed a good amount of *research* on the different technologies and engaged additional team members to support this project**
  - Martina von dem Bussche: security-related aspects for Linux on z and z/OS
  - Uwe Denneler: z Systems infrastructure setup and configuration
  - Tobias Leicher: integration into CICS Transaction Server for z/OS

Marc Beyerle
Java / WebSphere

Tina v.d. Bussche
Security / RACF

Uwe Denneler
z Systems infrastr.

Tobias Leicher
CICS / CTG

Agenda

- Introduction & background
- **Phase 1 (workshop): evaluating different integration technologies**
- Phase 2 (Proof of Concept): implementing the proposed architecture
- Summary

The Big Picture

HTTP(S)

?

Some protocol(*)

Microsoft .NET

Java

(*) can be one of:

- **CTG (IPIC, TCPIP, SNA)**
- **JSON**
- **Web Services**

Too many options!

HTTP(S)

?

Some protocol(*)

(*) can be one of:
- CTG (IPIC, TCPIP, SNA)
  - Local or remote (x2)
- JSON
- Web Services

22 = 2 x 11 (2)

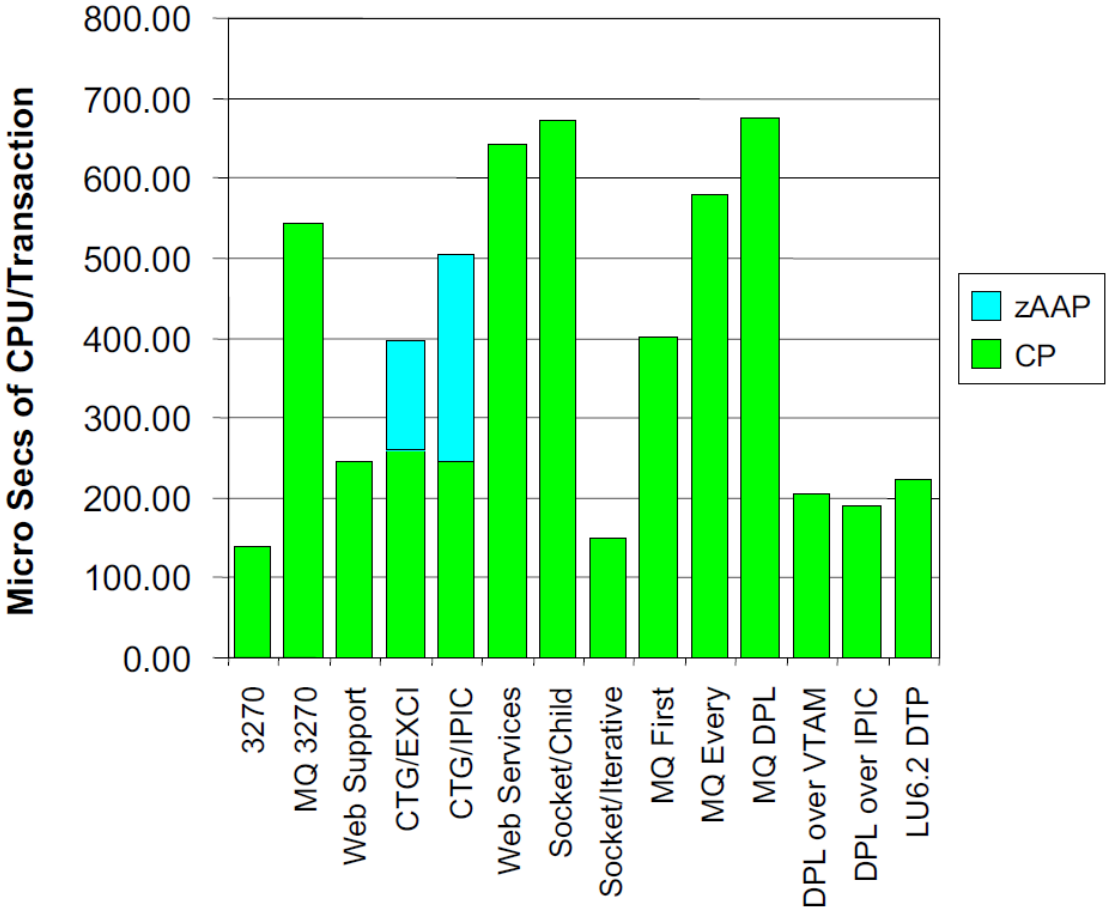
Integration technologies

- **In an on-site workshop at the client's IT headquarters, we presented the different protocol options, each with its own set of *pros* and *cons***
  - *IP Interconnectivity* (IPIC)
  - TCPIP (also known as *ECI over TCP/IP*)
  - SNA® *Advanced Program-to-Program Communication* (APPC)
  - *External CICS Interface* (EXCI)
  - *JavaScript® Object Notation* (JSON)
  - Web Services

- ***Evaluation criteria*: support for *2-Phase Commit* (2PC), support for *z/OS Identity Propagation*, zIIP offload potential, etc.**
  - IPIC turned out to be the most complete option from a functionality perspective and is also the protocol which is most widely used by customers in general

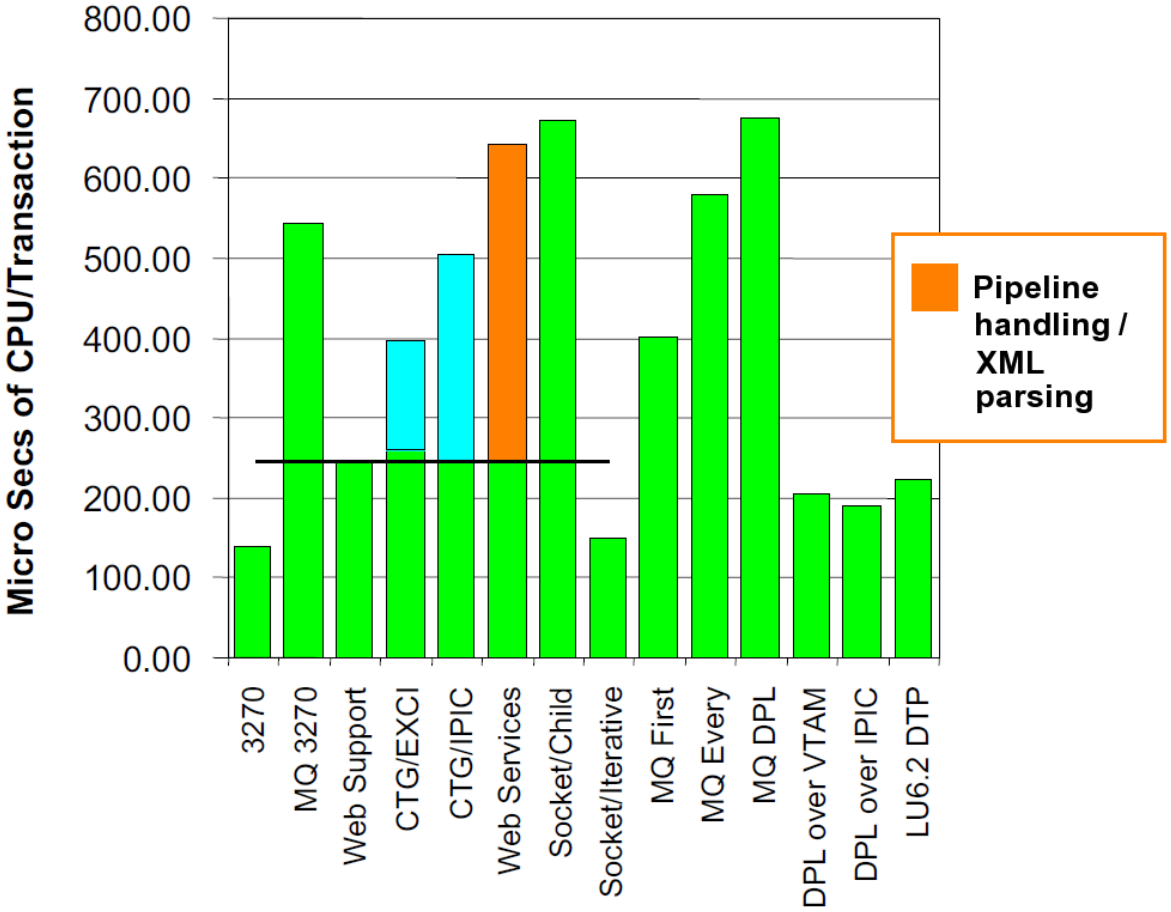
- **See the end of this presentation for a list of IBM *Redbooks*® and *Redpapers™* that were *very useful* for preparing this comparison**

Non-functional requirement: Performance



Source: IBM Redpaper *IBM CICS Performance Series: A Processor Usage Study of Ways into CICS*, http://www.redbooks.ibm.com/abstracts/redp4906.html

Non-functional requirement: Performance, *continued*



Caution: This is not an actual measurement, but a "guesstimation" based on discussions with CICS experts

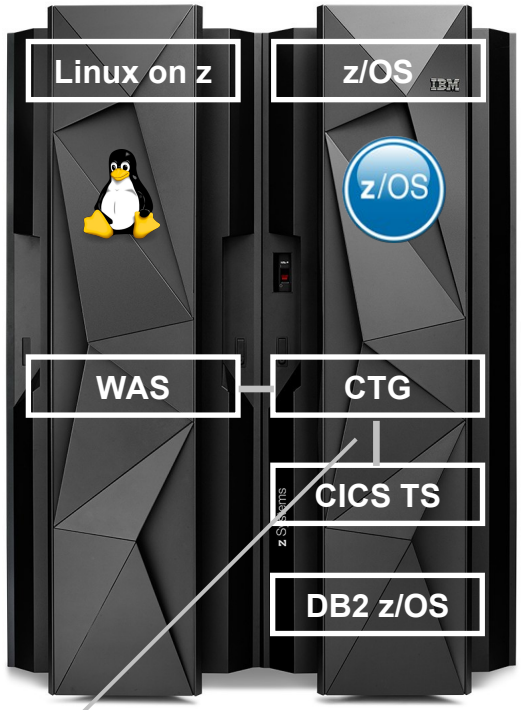
Integration technologies, *continued*

- **Of all the possible combinations, the option including WebSphere Application Server on *Linux on z Systems* and the *CICS Transaction Gateway* (CTG) on z/OS turned out to be the "best fit" for the customer's requirements**
  - Reason for this recommendation: combination of functionality (2PC, etc.), performance, security options, and integration into the z platform
  - Used proven ***Fit-for-Purpose* (F4P)** methodology for the evaluation
  - See the next slide for a high-level view of the recommended option

- **Reason for recommending *CTG on z/OS*: customer already had long-term experience with this setup**
  - In F4P terminology, this is considered a ***local factor***

Recommended option: WebSphere Application Server with remote CTG

HTTP(S)



IPIC / EXCI

Agenda

- Introduction & background
- Phase 1 (workshop): evaluating different integration technologies
- **Phase 2 (Proof of Concept): implementing the proposed architecture**
- Summary

Outcome of the on-site workshop

- **In the workshop, it turned out that the most important criteria for the client were actually (1) the possibility to have a *full audit trail* and (2) an *SSO solution* that integrates seamlessly with Microsoft's *Active Directory*® (AD)**
  - Performance and security were still considered important, but the full audit trail capability and SSO turned out to be even more important
  - AD is the client's central user repository

- **In general, security plays a *key role* in the bank – security does not only include things like authentication / authorization and encryption, but also the possibility to have a full audit trail**
  - For the full audit trail, customer wanted to use z/OS *Resource Access Control Facility* (RACF®), since this is their primary data source for security-related reports, evaluations, etc.

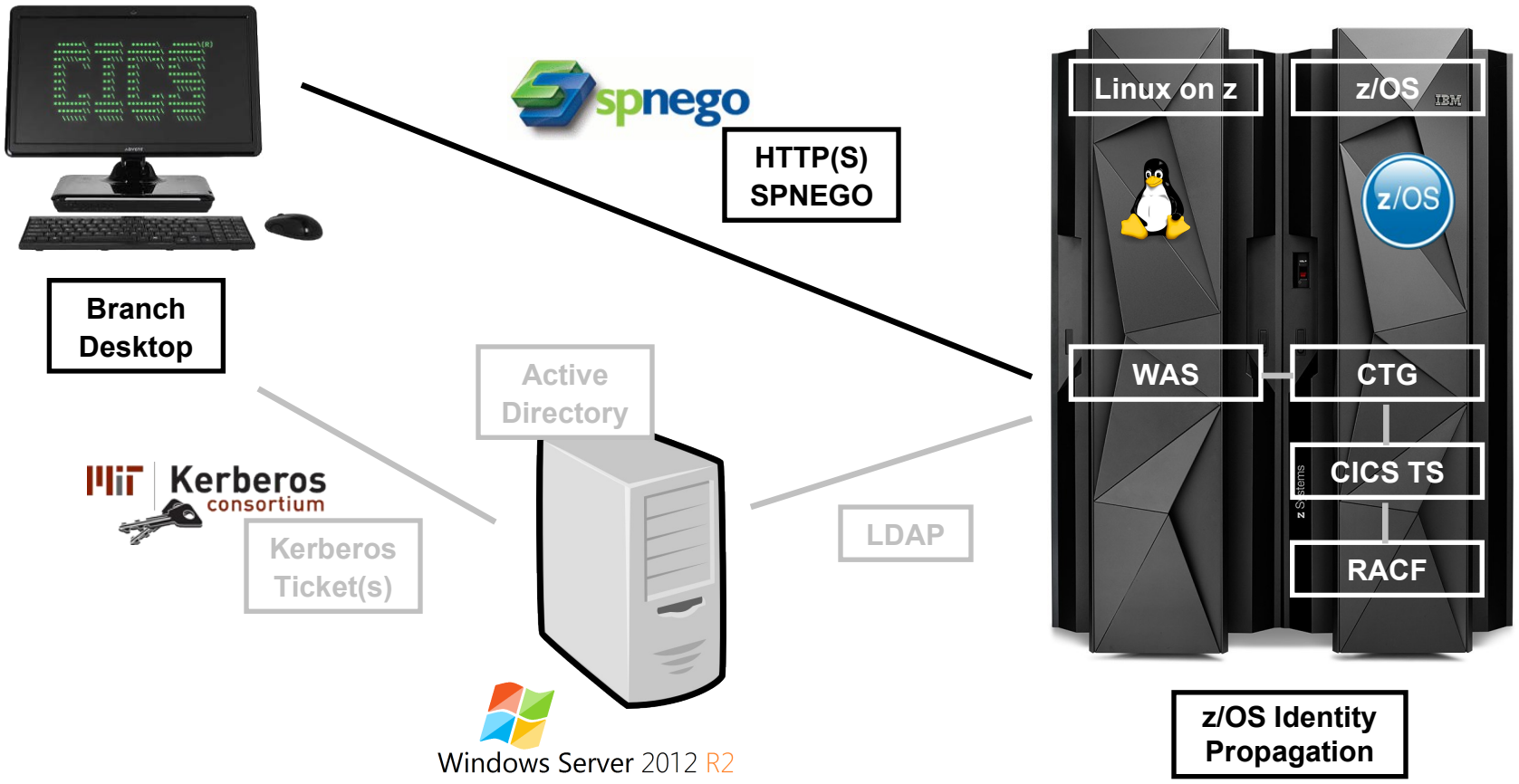
Technologies used in the Proof of Concept

- **Performed a good amount of *research* on both SSO and full audit trail**
  - On top of the mentioned Redbooks, a good source of information was also the WebSphere Application Server *Knowledge Center* on the Internet

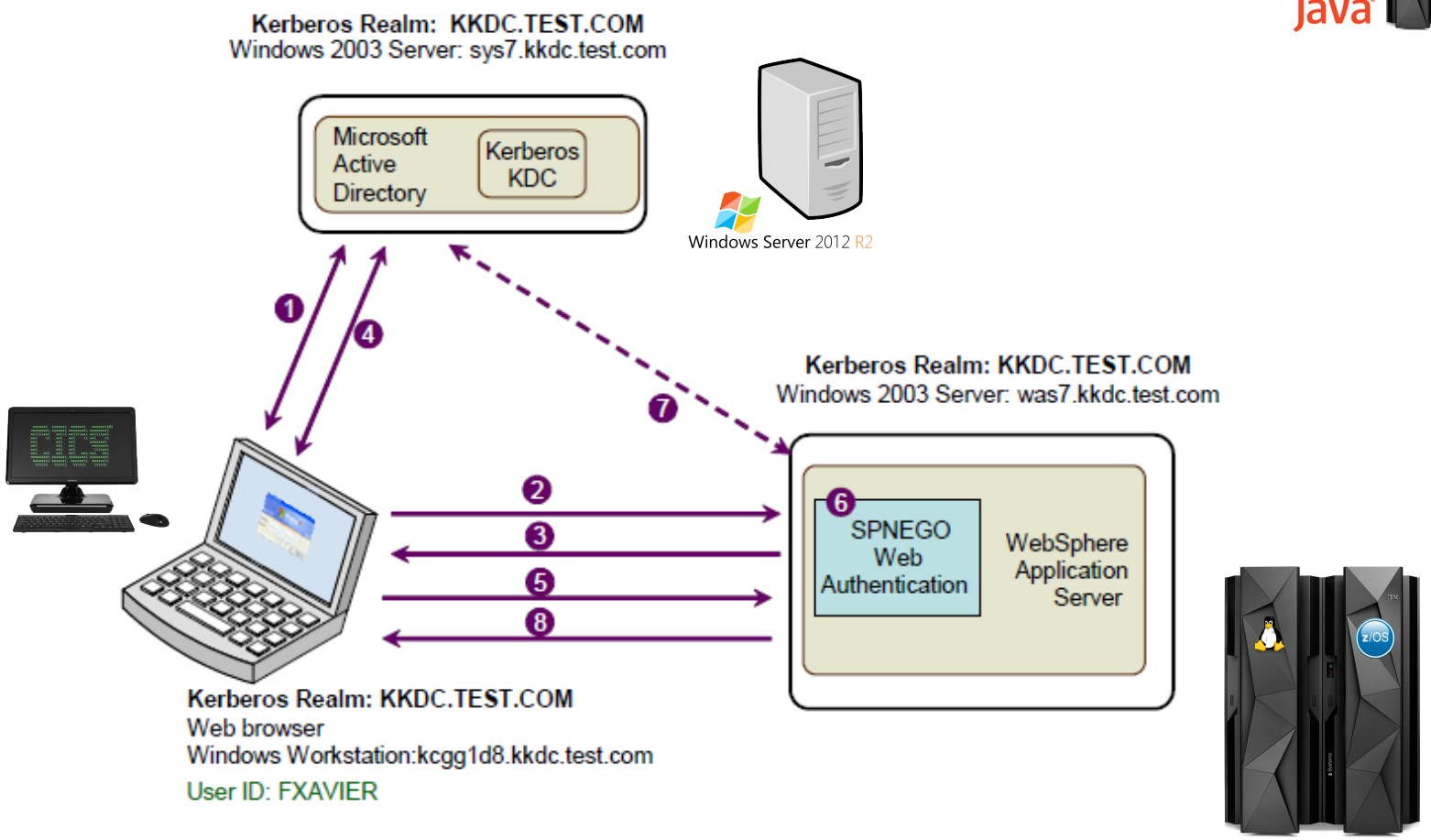
- ***Single Sign-On*: decided to use *Kerberos*® and *SPNEGO*, since this is the combination of technologies that is documented best when it comes to integration with Active Directory**
  - Other possible technologies included for example *Security Assertion Markup Language* (SAML™), but they are much more complicated to set up and not as well documented as Kerberos and *Simple and Protected GSSAPI Negotiation Mechanism* (SPNEGO)

- ***Full audit trail*: the one technology that allows for an easy integration into RACF and all the other components along the chain (WebSphere, CTG, CICS) is *z/OS Identity Propagation***
  - From the *z/OS Identity Propagation* Redbook: *"Identity propagation is the capability whereby a non z/OS identity, a distributed identity, is propagated into the z/OS environment…"*

Architecture for the Proof of Concept



Kerberos and SPNEGO flow



Source: IBM Redbook *Implementing Kerberos in a WebSphere Application Server Environment*, http://www.redbooks.ibm.com/abstracts/sg247771.html

Implementation details

- **First, we had to get hold of a *Microsoft Windows® Server 2012* (hardware + license) for Active Directory**
  - Why 2012? Because this is the version currently in use by the customer
  - Finally managed to get a temporary loaner from our colleagues in the *Technical Exploration Center* (TEC) in Ehningen – thank you very much for that!

- **Second, ensure that all components are in the same network and can actually *reach each other*: client workstation, WebSphere Application Server on Linux on z, CICS TS on z/OS, and the Microsoft AD Server**
  - Linux: entries in `/etc/hosts`
  - ***Important***: the Linux host name has to match the name that is later on encoded into the Kerberos *Service Principal Name* (SPN), otherwise the WebSphere configuration will fail
  - Linux host name in our setup: `wasdemo.mybank.test`
  - Microsoft Windows Server: DNS configuration
  - ***Important***: in Active Directory, you have to configure the ***encryption algorithms*** that can be used by Kerberos

Implementation details, *continued*

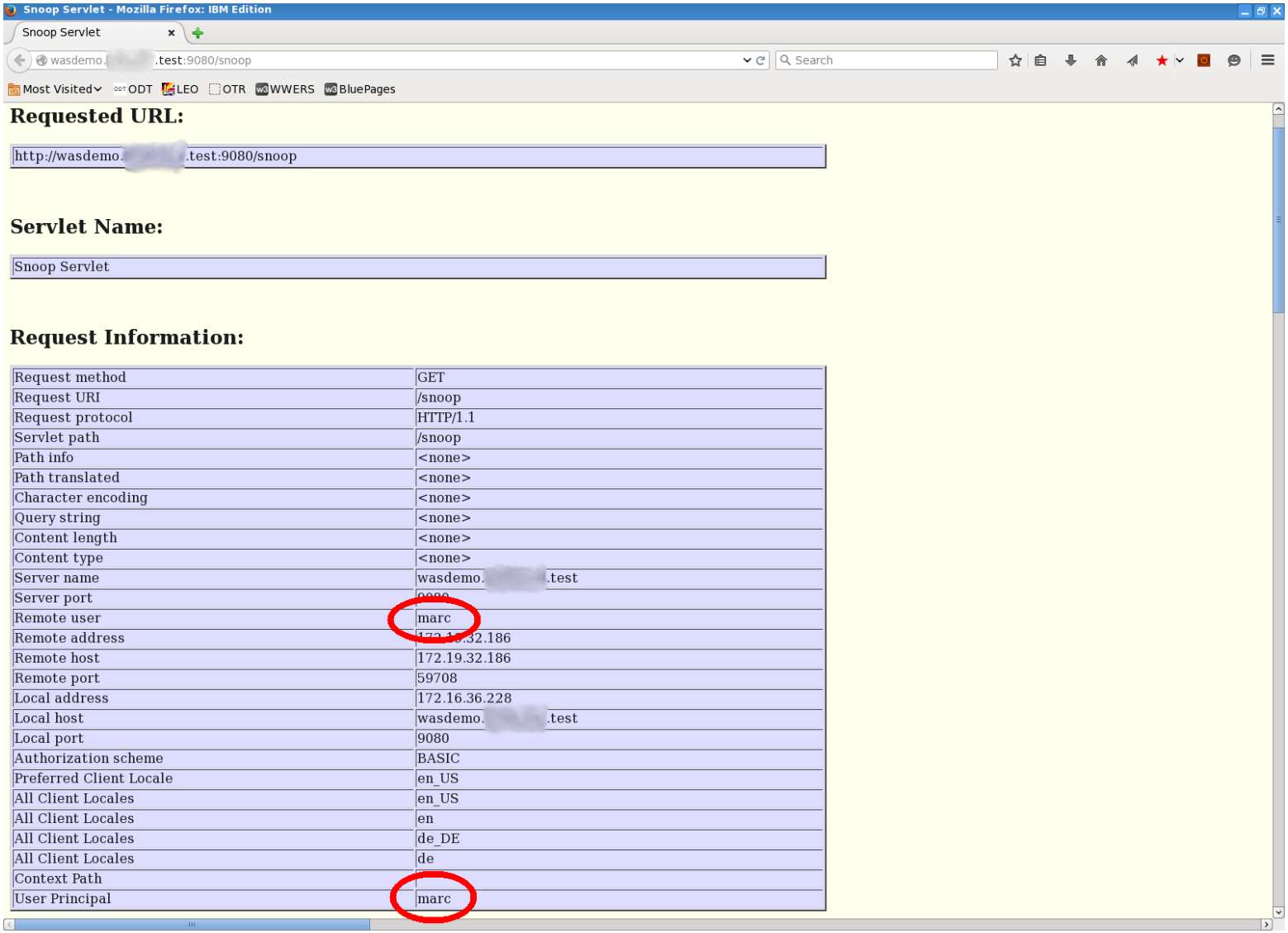
- **Create the WebSphere environment – cell, node, server, etc.**
  - ***Important***: start with *Administrative Security* turned off, otherwise you will lock yourself out of the Admin Console
  - Starting with security turned off is not an issue, since you have to enable Administrative Security anyway later on in the configuration process

- **Create (at least) *2 new AD users***
  - WebSphere administrator: `wasadmin`
  - Service user for Kerberos: `HTTP/wasdemo.mybank.test` (note the Linux host name in this Kerberos SPN)

- **On the Microsoft Windows Server, create the so-called *keytab* file**
  - Windows command line utility: `ktpass`
  - In this keytab file, the Kerberos SPN plus its associated keys are stored so that the service user can log on to the Active Directory later on without entering passwords

Implementation details, *continued*

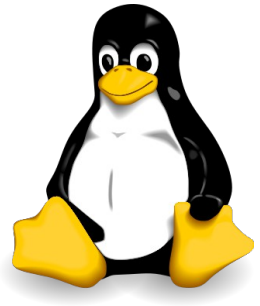
- **Create a *Kerberos configuration file* (typically called `krb5.conf` on Linux) for WebSphere Application Server**
  - Done using WebSphere scripting – `wsadmin.sh` and `$AdminTask`
  - Very well documented in the *Implementing Kerberos…* Redbook
- **Now simply follow chapter 7.3.2ff in the *Implementing Kerberos…* Redbook in order to *configure the WebSphere* Application Server *environment*: SSO, Active Directory, SPNEGO, and Kerberos**
  - As part of this configuration, Administrative Security has to be turned on
- **Configure the *browser* to make use of SPNEGO**
  - Steps are different for Internet Explorer® and Firefox®, see Appendix B in the Redbook
- **Now, you're able to "*automagically*" (i.e. via SSO) authenticate via Kerberos**



Nice to know

- **Although not required for the Proof of Concept, I managed to make use of Kerberos and SSO to automatically sign on to WebSphere's *Snoop* Servlet with my *Linux-based desktop* ☺**
  - This is done by using `libgssapi_krb5` and the corresponding command line tools - `kinit`, `klist`, `kdestroy`, etc.

```
[marc@LOCALHORST ~]$ export KRB5_CONFIG=/home/marc/krb5.conf
[marc@LOCALHORST ~]$ kinit -V marc
Using default cache: /tmp/krb5cc_500
Using principal: marc@MYBANK.TEST
Password for marc@MYBANK.TEST: …
Authenticated to Kerberos v5
[marc@LOCALHORST ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: marc@MYBANK.TEST
Valid starting Expires Service principal
03/03/16 16:33:42 03/04/16 02:33:37 krbtgt/MYBANK.TEST@MYBANK.TEST
renew until 03/04/16 16:33:42
[marc@LOCALHORST ~]$
```



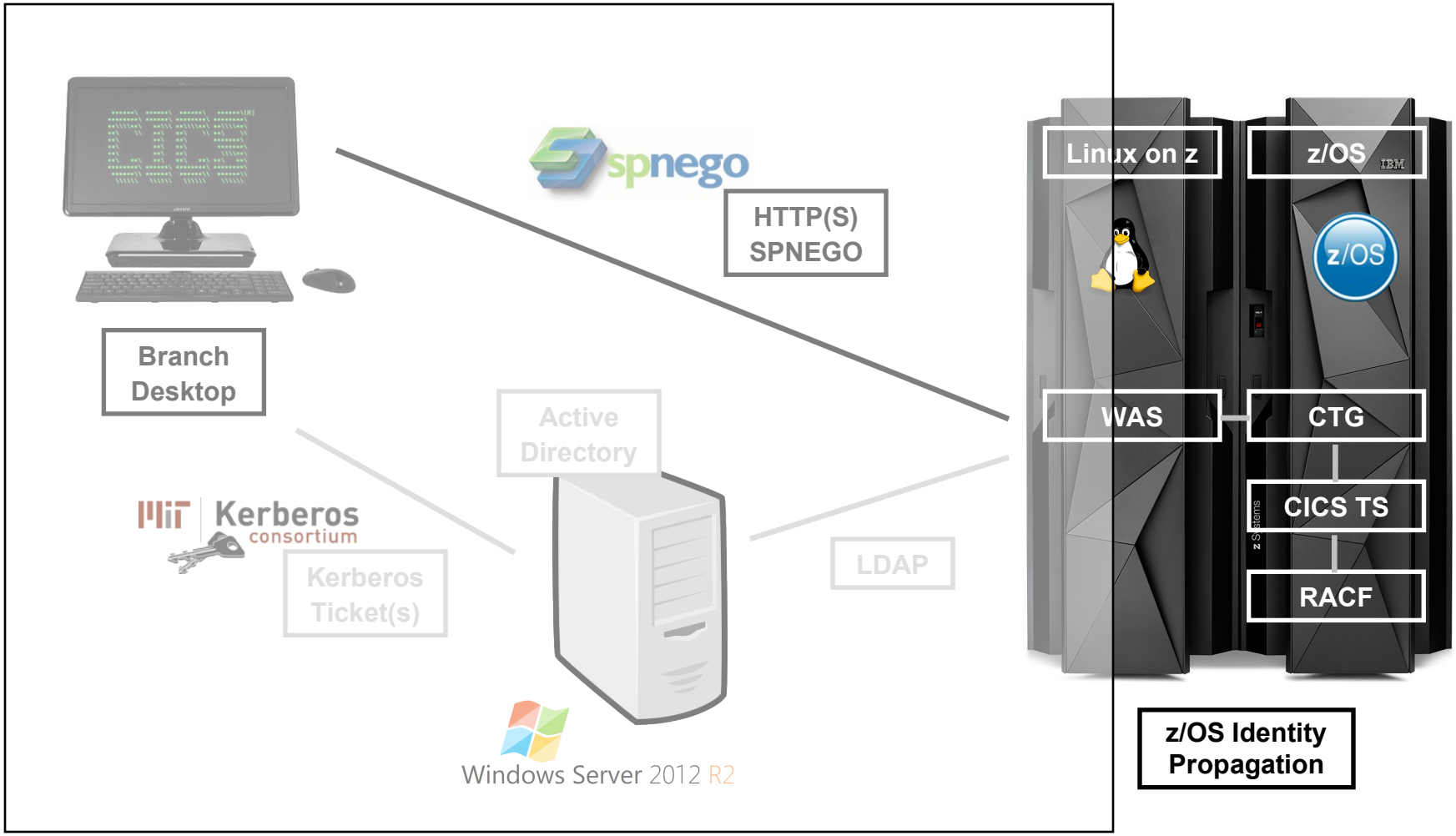
Nice to know, *continued*

- **After calling the Snoop Servlet, the Kerberos ticket cache contains an *additional ticket*, which was obtained as part of the SPNEGO *handshake* process**

```
[marc@LOCALHORST ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_500
Default principal: marc@MYBANK.TEST
Valid starting Expires Service principal
03/03/16 16:33:42 03/04/16 02:33:37 krbtgt/MYBANK.TEST@MYBANK.TEST
renew until 03/04/16 16:33:42
03/03/16 16:36:53 03/04/16 02:33:37 HTTP/wasdemo.mybank.test@MYBANK.TEST
renew until 03/04/16 16:33:42
[marc@LOCALHORST ~]$
```

- **Note the part in *red*: this is the fully qualified Kerberos SPN for SPNEGO**
  - SPNEGO requires the first part of this identifier to be `HTTP`

SSO configuration completed

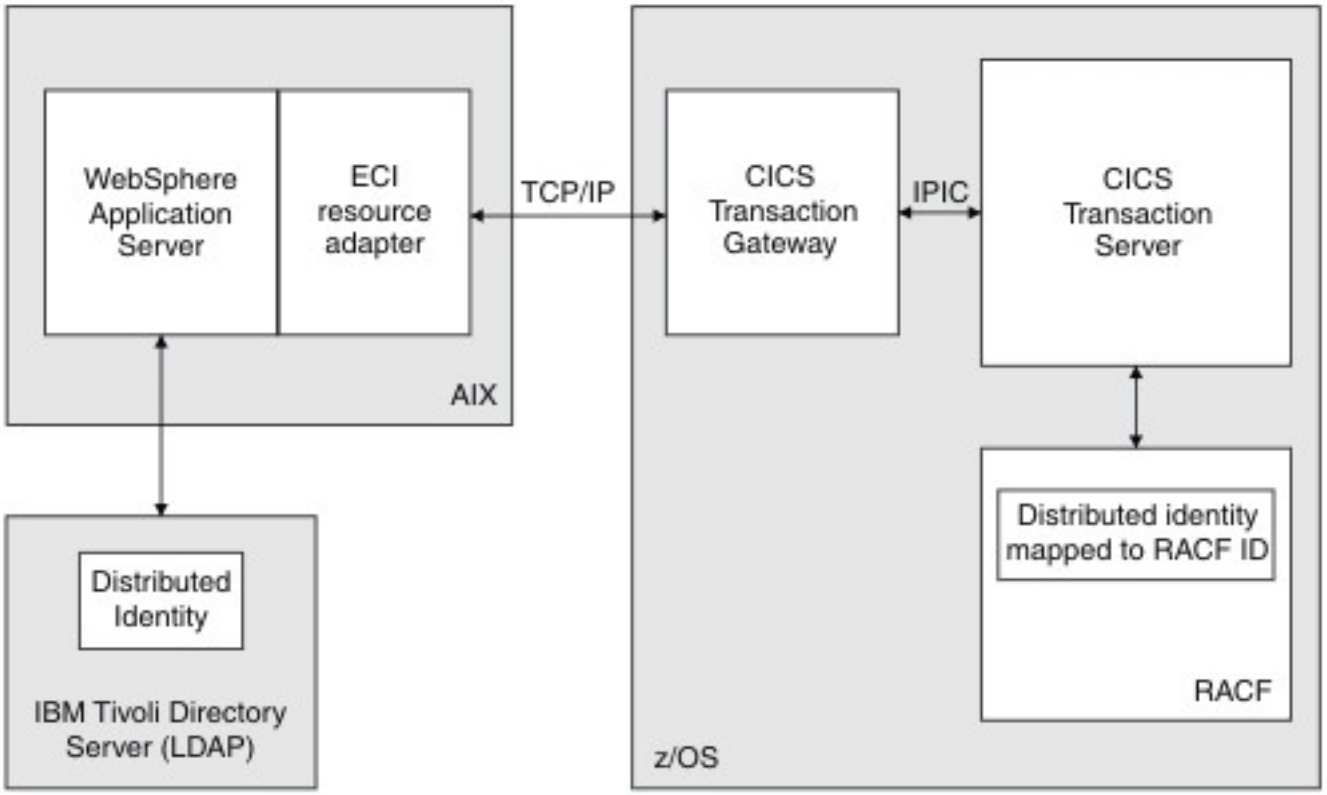


Implementation details, *continued*

- **After getting SPNEGO and Kerberos up and running, we had to configure the *CICS / CTG-related options* in WebSphere Application Server**
  - Install the CICS ECI resource adapter (`<CTG_HOME>/deployable/cicseci.rar`)
  - Create a J2C connection factory
  - Very well documented in the Redbook *The Complete Guide to CICS Transaction Gateway Volume 1 – Configuration and Administration*, section 12.5

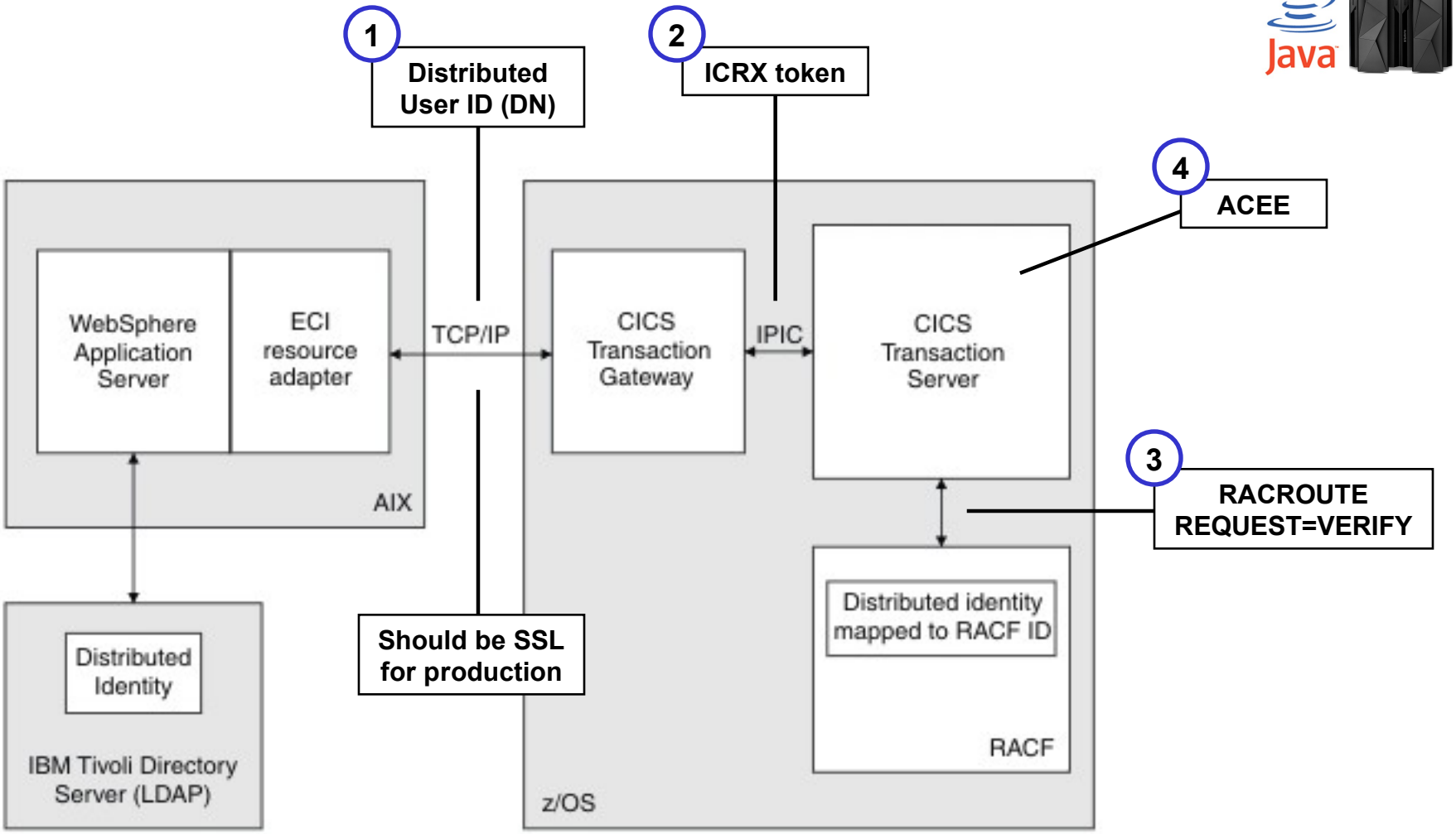
- **Next, we configured *CICS Transaction Server* and *CTG on z/OS* according to *Scenario 04* (SC04) in the CTG for z/OS Knowledge Center**
  - CTG: `APPLID`, `HOSTNAME`, `PORT`, etc.
  - CICS: `TCPIPService`, `IPConn`
  - Link to the Knowledge Center: https://www.ibm.com/support/knowledgecenter/SSZHJ2_9.1.0/scenarios/topics/sc_idprop_ovr.html

Scenario 04 (SC04)



Source: ***Scenario 04*** (SC04) in the IBM Knowledge Center for CICS Transaction Gateway for z/OS

Scenario 04 (SC04), *continued*



Source: ***Scenario 04*** (SC04) in the IBM Knowledge Center for CICS Transaction Gateway for z/OS

Implementation details, *continued*

- **After the CICS configuration, we defined the required *mappings in RACF*, in order to map Microsoft AD users to z/OS RACF users**
  - This is done by using the `RACMAP` command
  - You can for example use ***one-to-one mappings*** for specific users, ***many-to-one mappings*** for other users in the same Active Directory domain, and '*' for the ***default mapping*** for all other users
  - If you are using ISPF and / or TSO, `RACMAP` has to be defined in the TSO/E *APF-authorized command table* (see screenshot below)
  - See the Redbook *z/OS Identity Propagation* for further details and examples

```
 BROWSE    SYS1.PARMLIB(IKJTSO00) - 01.24            Line 00000000 Col 001 080
********************************* Top of Data ***********************************
/* DOC:  THIS MEMBER IS USED AT IPL TIME TO DEFINE THE AUTHORIZED    */
/*       COMMAND LIST, THE AUTHORIZED PROGRAM LIST, THE NOT          */
/*       BACKGROUND COMMAND LIST, THE AUTHORIZED BY THE TSO SERVICE  */
/*       FACILITY LIST, AND TO CREATE THE DEFAULTS THE SEND COMMAND  */
/*       WILL USE.                                                   */
/*                                                                   */
AUTHCMD NAMES(                     /* AUTHORIZED COMMANDS          */ +
   RACMAP                               /* IBM TIVOLI ZSECURE ADMIN */ +
   CKGRACF                              /* IBM TIVOLI ZSECURE ADMIN */ +
   B8RACF                             /* ZSECURE ADMIN RACF-OFFLINE */ +
   FTP                             /*                              */ +
```

Implementation details, *continued*

- **The last configuration step in WebSphere is to set up the identity propagation *login module***
  - Documented in the Knowledge Center for CICS Transaction Gateway: https://www.ibm.com/support/knowledgecenter/SSZHJ2_9.1.0/scenarios/topics/sc_idprop_was.html

- ***Caution*: for the `propIdentity` custom property, use the value `Caller`, and not `RunAs` as suggested in the Knowledge Center entry**
  - Found this in the Knowledge Center of an older version of CTG for z/OS

- **In order to have some application that accesses CTG on z/OS, we used `ECIIVT.ear`, the ECI resource adapter *installation verification application* that comes with CTG for z/OS**

Implementation details, *continued*

- **In order to actually see SSO and Kerberos working, I *slightly adapted* the `ECIIVT.ear` sample application**
  - Originally, the idea of `ECIIVT.ear` is to test whether the connection to CTG is actually working – the purpose is not to showcase SSO and / or integration of Kerberos and / or z/OS Identity Propagation
  - I defined the default Servlet as being ***protected*** – this means that only authenticated and authorized users can access the default Servlet, others get an error page
  - I used the special subject *All Authenticated in Application's Realm* in order to map all authenticated users to the security role required for accessing the default Servlet

- **Update the *filter criteria* for the URLs that are enabled for SPNEGO in order to include the path to the IVT application**
  - URLs are only enabled for SPNEGO by configuration, not by default

zSecure™ CARLa report on live SMF data


```
S M F   R E C O R D   L I S T I N G    9Mar16 16:59 to 10Mar16 09:58


Date/time           Typ User     Event    Eq  Description
10Mar16 09:58:25.45  80 MARC     ACCESS     0 RACF ACCESS success for MARC:
(READ,READ) on TCICSTRN CSMI
                       CN=beyerle,CN=Users,DC=mybank,DC=test
                       WIN-A8DPKGM1QA0.mybank.test:389


10Mar16 09:58:25.46  80 CICSUSER ACCESS     0 RACF ACCESS success for CICSUSER:
(READ,READ) on TCICSTRN CSMI




10Mar16 09:58:25.46  80 MARC     ACCESS     0 RACF ACCESS success for MARC:
(READ,READ) on TCICSTRN CSMI
                       CN=beyerle,CN=Users,DC=mybank,DC=test
                       WIN-A8DPKGM1QA0.mybank.test:389


10Mar16 09:58:58.70  80 NOACC    ACCESS     1 RACF ACCESS violation for NOACC:
(READ,NONE) on TCICSTRN CSMI
                       CN=marc,CN=Users,DC=mybank,DC=test
                       WIN-A8DPKGM1QA0.mybank.test:389
```

Interpreting the zSecure output

- ***Question*: Why is there a RACF *access violation* for `NOACC` in the last page?**

- ***Answer*: Because there is no *one-to-one* mapping defined for the Microsoft AD user `marc`. The user `marc` falls into the *many-to-one* mapping for the AD domain.**

```
RACMAP ID(MARC) USERDIDFILTER(name('CN=beyerle,CN=Users,DC=mybank,DC=test'))
  REGISTRY(name('WIN-A8DPKGM1QA0.mybank.test:389')) WITHLABEL('MYBANK01')

SETROPTS RACLIST(IDIDMAP) REFRESH


RACMAP ID(NOACC) USERDIDFILTER(name('CN=Users,DC=mybank,DC=test'))
  REGISTRY(name('WIN-A8DPKGM1QA0.mybank.test:389')) WITHLABEL('MYBANK03')

SETROPTS RACLIST(IDIDMAP) REFRESH
```

Agenda

- Introduction & background
- Phase 1 (workshop): evaluating different integration technologies
- Phase 2 (Proof of Concept): implementing the proposed architecture
- **Summary**

Summary

- **Admittedly, there are *quite a few configuration steps* required when it comes to setting up SSO and z/OS Identity Propagation**
  - Good news: everything was very well documented
- **Once the environment is set up, the scenario *works perfectly***
  - Both SSO and identity propagation work as expected and you can really follow closely what's going on under the covers
- **Largest part of the setup is *WebSphere*-related**
  - If you know what you're doing(*), the CICS / CTG and RACF parts can be configured pretty quickly and easily
- **All in all, this project was a *very nice learning experience* for me personally**
  - Never had to deal with so many different security aspects before

(*)That's the crux, of course ☺

Thank you

Resources

- **IBM Client Center – Systems and Software, IBM Germany Lab**
  - Part of the IBM Development Lab in Boeblingen, Germany
  - External homepage: http://www.ibm.com/ibm/clientcenter/boeblingen
  - IBM Intranet: http://clientcenter.de.ibm.com
  - Email: clientcenter@de.ibm.com

- **IBM Redbook *The Complete Guide to CICS Transaction Gateway Volume 1 – Configuration and Administration*, http://www.redbooks.ibm.com/abstracts/SG248160.html**

- **IBM Redpaper *IBM CICS Performance Series: A Processor Usage Study of Ways into CICS*, http://www.redbooks.ibm.com/abstracts/redp4906.html**

- **IBM Redbook *CICS and SOA: Architecture and Integration Choices*, http://www.redbooks.ibm.com/abstracts/sg245466.html**

- **IBM Redbook *Implementing IBM CICS JSON Web Services for Mobile Applications*, http://www.redbooks.ibm.com/redpieces/abstracts/sg248161.html**

- **IBM Redbook *z/OS Identity Propagation*, http://www.redbooks.ibm.com/abstracts/sg247850.html**

- **IBM Redbook *Implementing Kerberos in a WebSphere Application Server Environment*, http://www.redbooks.ibm.com/abstracts/sg247771.html**