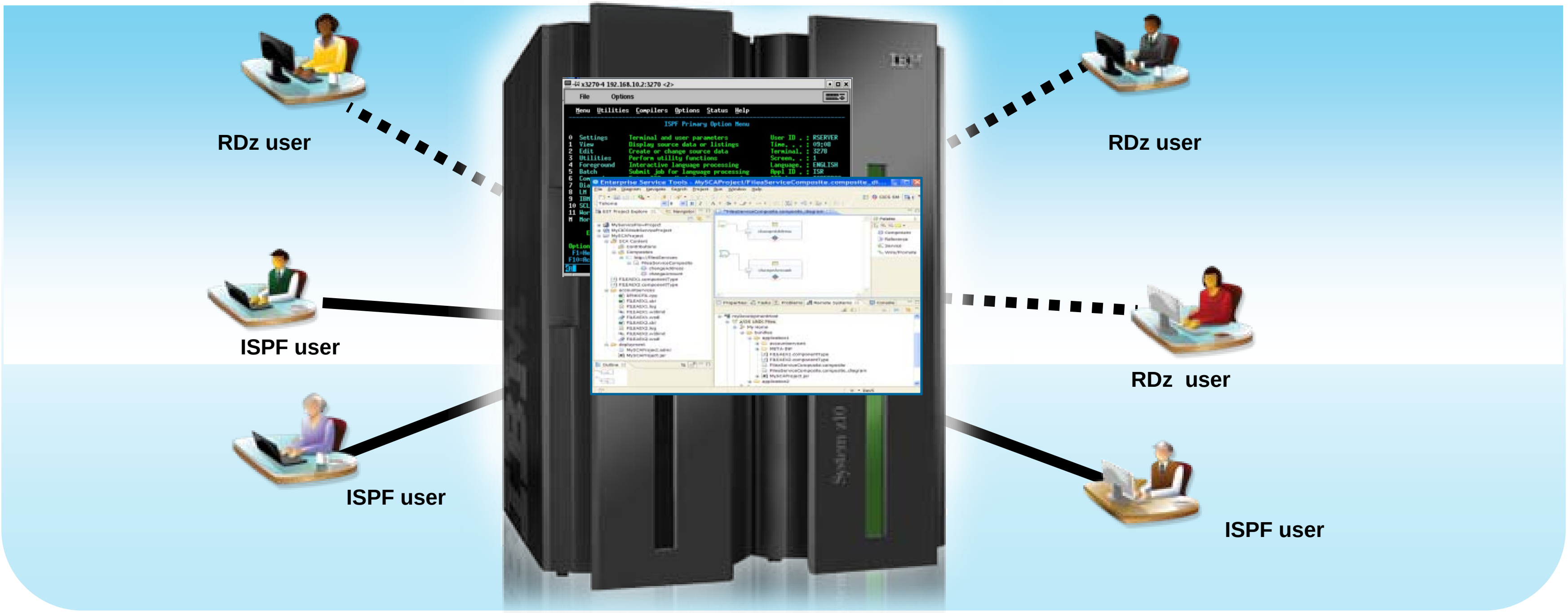
Creating a Personal z/OS® Development and Unit Test Environment – Lunch & Learn

Rosalind Radcliffe / Dana Boudreau

12 March 2012

Session #: 11091

Today’s mainframe development environment with RDz and ISPF


RDz user
ISPF user
ISPF user
RDz user
RDz user
ISPF user
Enterprise Service Tools - MySCAProject/FilesServiceComposite.composite_di...

```
x3270-4 192.168.10.2:3270 <2>
File     Options
 Menu  Utilities  Compilers  Options  Status  Help
                   ISPF Primary Option Menu
0  Settings      Terminal and user parameters         User ID . : RSERVER
1  View          Display source data or listings      Time. . . : 09:08
2  Edit          Create or change source data         Terminal. : 3278
3  Utilities     Perform utility functions            Screen. . : 1
4  Foreground    Interactive language processing      Language. : ENGLISH
5  Batch         Submit job for language processing   Appl ID . : ISR
```

Modern IDEs add value:

- Productivity enhancing tooling; more attractive tooling for new developers
- Ability to offload some development MIPS usage
- Integration with complete application lifecycle tools

But challenges remain:

- Business pressures to manage mainframe MIPS usage for development
- Unit test delays caused by dependencies on operations team

Modern Desktop Tools

What is RDz

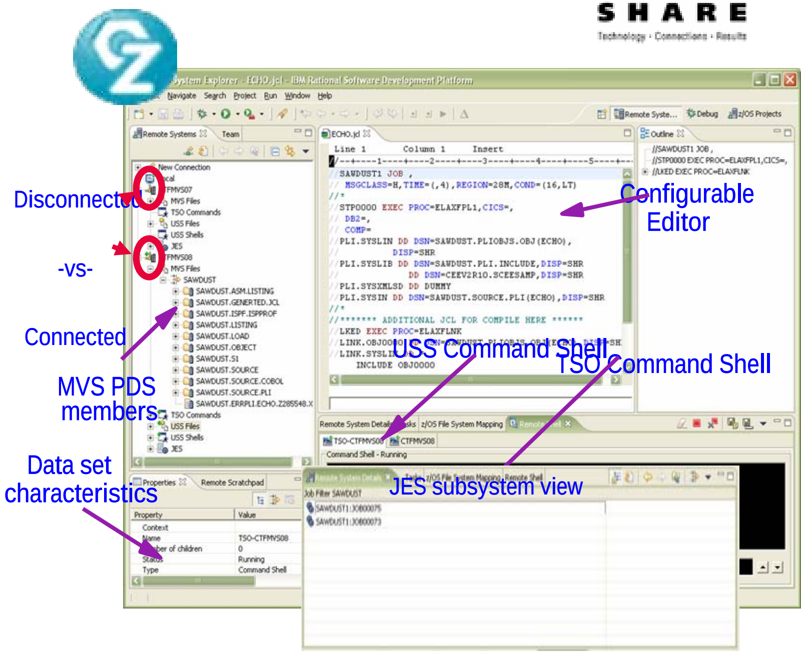
- Eclipse-based IDE speeding modern composite (SOA) application development

RDz supports Enterprise Modernization

- Supports common IDE for COBOL, PL/I, C, C++, HLASM, Java, EGL and web services
- Transforms UML to COBOL source code
- Provides interactive access to z/OS for development, debug, job generation, submission, monitoring, command execution
- Supports new and existing runtimes (CICS, IMS, Batch, USS, DB2 SP, WAS)

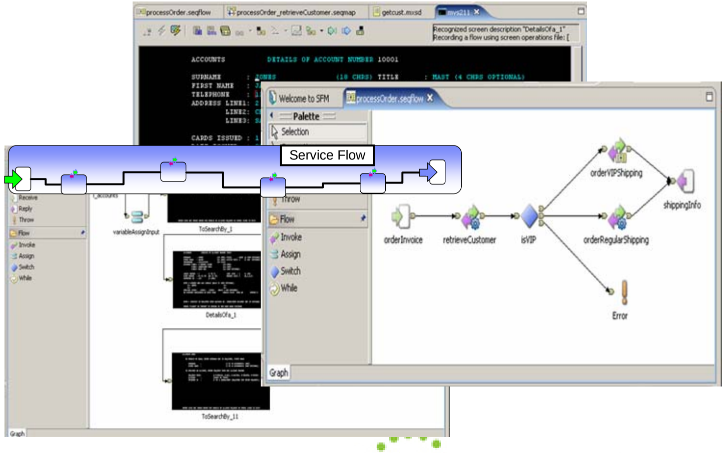
RDz supports SOA

- Enables CICS and IMS applications for web services and SOA
- Supports for J2EE, JCA, XML, web services



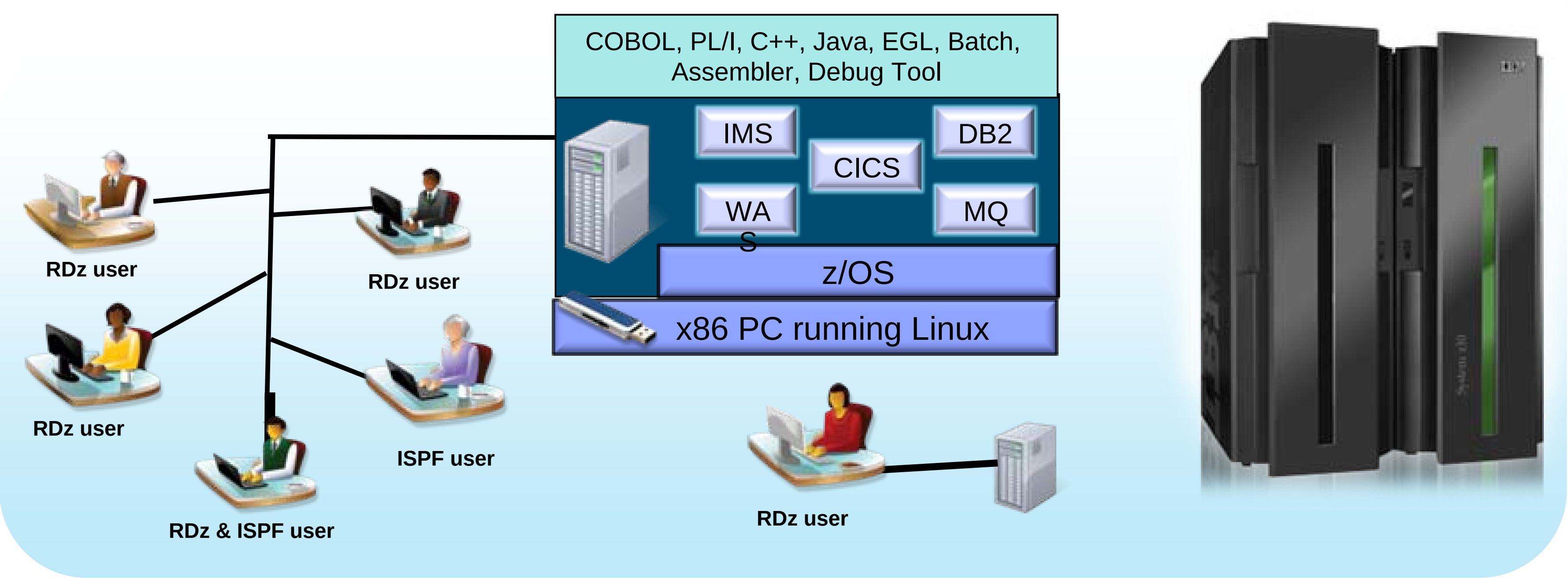
```
Line 1         Column 1      Insert
//--+----1----+----2----+----3----+----4----+----5----+--
//SAWDUST1 JOB ,
// MSGCLASS=H,TIME=(,4),REGION=28M,COND=(16,LT)
//*
//STP0000 EXEC PROC=ELAXFPL1,CICS=,
// DB2=,
// COMP=
//PLI.SYSLIN DD DSN=SAWDUST.PLIOBJS.OBJ(ECHO),
//          DISP=SHR
//PLI.SYSLIB DD DSN=SAWDUST.PLI.INCLUDE,DISP=SHR
//             DD DSN=CEEV2R10.SCEESAMP,DISP=SHR
//PLI.SYSXMLSD DD DUMMY
//PLI.SYSIN DD DSN=SAWDUST.SOURCE.PLI(ECHO),DISP=SHR
//*
//******* ADDITIONAL JCL FOR COMPILE HERE ******
//LKED EXEC PROC=ELAXFLNK
//LINK.OBJ0000 DD DSN=SAWDUST.PLIOBJS.OBJ(ECHO),DISP=SH
//LINK.SYSLIN DD *
    INCLUDE OBJ0000
```

Property	Value
Context	
Name	TSO-CTFMVS08
Number of children	0
Status	Running
Type	Command Shell



Announcing the RDz Unit Test Feature

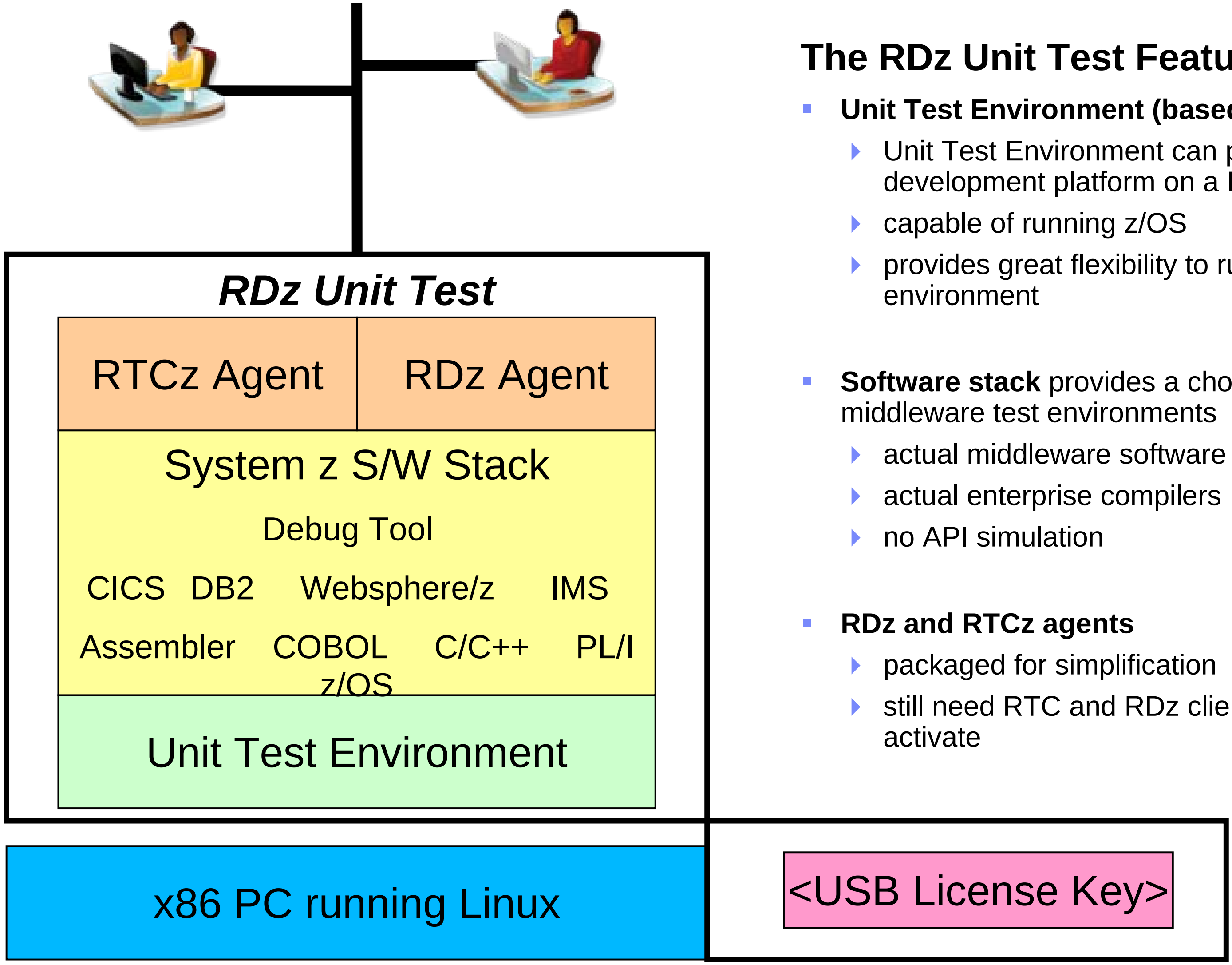
The ultimate in modern application development for System z



- Liberate developers to rapidly prototype new applications
- Develop and test System z applications anywhere, anytime!
- Free up mainframe development MIPS for production capacity
- Eliminate costly delays by reducing dependencies on operations staff

Note: This Program is licensed only for development and test of applications that run on IBM z/OS. The Program may not be used to run production workloads of any kind, nor more robust development workloads including without limitation production module builds, pre-production testing, stress testing, or performance testing.

RDz Unit Test Offering Description



The RDz Unit Test Feature consists of:

- **Unit Test Environment (based on zPDT)**
  - Unit Test Environment can provide a System z development platform on a PC
  - capable of running z/OS
  - provides great flexibility to run a customized environment
- **Software stack** provides a choice of IBM middleware test environments
  - actual middleware software (including z/OS)
  - actual enterprise compilers
  - no API simulation
- **RDz and RTCz agents**
  - packaged for simplification
  - still need RTC and RDz client license(s) to activate

RDz UT machine specifications

- Underlying Linux requirement
  - Red Hat Enterprise Linux 5.3 (RHEL 5.3) – 64-bit
  - OpenSUSE 10.3, 11.0, and 11.1 – 64-bit

- Base machine must have:
  - Processor
    - RDz-UT Standard - Minimum practical configuration is 2.4 Ghz Core 2 Duo
    - RDz-UT Specialty - Minimum practical configuration is 2.0 Ghz i5 / i7 4 core
  - RAM
    - 3 GB minimum
    - Recommended is 1 GB for base Linux + 2 GB per processor core enabled for RDz-UT
  - Disk space
    - 20 GB for base Linux / zPDT install
    - 50-120 GB for z/OS install
  - Network
    - 100MB / 1GB Ethernet adapter (shared by Linux and z/OS)
    - Wireless (OK for Linux, not recommended for z/OS)
  - USB
    - Required for 1091 hardware key

ADCD 1.11s Software Products Licensed for Use with RDZ-UT V.8.0.1

- z/OS Version 1 Release 11
- IBM Ported Tools for z/OS 1.14
- IBM Print Services Facility for z/OS V4.2
- IBM REXX/370 Compiler & IBM Library for REXX/370
- IBM AFP(TM) Font Collection for VSE
- IBM Enterprise COBOL for z/OS V4.2
- IBM XML Toolkit for z/OS V1.10
- IBM 31-bit SDK for z/OS, Java 2 Technology Edition V5.0
- IBM 64-bit SDK for z/OS, Java 2 Technology Edition V5.0
- IBM 31-bit SDK for z/OS, Java Technology Edition, V6
- IBM 64-bit SDK for z/OS, Java Technology Edition, V6
- IBM Debug Tool for z/OS V9.1
- IBM Enterprise PL/I for z/OS V3.8
- IBM Tivoli NetView for z/OS V5.3
- IBM z/OS Management Facility V1.10
- IBM Rational Developer for System z V7.5 (Remote System Explorer Component)
- IBM DB2 for z/OS V9.1
- IBM DB2 Administration Tool for z/OS V7.2
- IBM DB2 Object Comparison Tool for z/OS V7.2
- IBM DB2 Accessories Suite for z/OS V1.1
- IBM CICS Transaction Server for z/OS V4.1
- IBM Information Management System (IMS) Version 11.1.0
- IBM IMS Queue Control Facility for z/OS V2.1
- IBM IMS High Performance Change Accumulation Utility for z/OS V1.4
- IBM IMS High Performance Image Copy for z/OS V4.1.0
- IBM IMS(TM) Database Control Suite for z/OS V3.2
- IBM IMS Library Integrity Utilities for z/OS V2.1
- IBM WebSphere Application Server for z/OS V7.0
- IBM Tivoli Workload Scheduler for z/OS V8.5.0
- IBM WebSphere MQ for z/OS V7.0.0

* Custom software stacks my be available at additional cost

RDz UT - Limitations

- **The RDz UT environment does NOT support all System z function, such as:**
  - Physical Parallel, ESCON®, FCP, FICON® and High Performance FICON channels
  - Coupling links and coupling facilities
  - Sysplex
  - List-directed IPL
  - External Time Reference (ETR)
  - Server Time Protocol (STP)

- **RDz UT does not produce an environment equal to a larger System z**
  - Some aspects of a larger system are unlikely to be met in any very small environment.
    - Inability to verify and enhance the scalability of a program
    - Inability to run application programs that require hundreds of MIPS.
  - A UT system is not recommended for very fine-level performance tuning that is sensitive to memory location, cache functions, and pipeline optimization.
  - In addition, the UT platform does not nearly have the same quality of service as does a mainframe in terms of availability and connectivity.

- **Anyone needing any of the function outlined above should consider a traditional System z server environment such as Solution Edition for Application Development**

Updates with RDz UT 8.0.3

Available Nov, 10

- Product names changed for clarity
  - Rational Developer for System z Unit Test – Desktop User (was Standard Edition)
    - Sell to individuals looking for laptop configurations
    - Small-scale: 3-5 users per instance
    - Focus on CICS, IMS, DB2, Batch, Compiling
  - Rational Developer for System z Unit Test – Server User (was Speciality Edition)
    - Sell to teams looking for server configurations
    - Team-scale: 15-30 users per instance
    - Focus on CICS, IMS, DB2, Batch, WebSphere, Compiling and larger workloads
- License changes
  - RDz UT – Server User now licenses via “Authorized User Single Install”
  - 1 license = 1 user on 1 server
  - Eases compliance/audit, aligned with Jazz licensing
- Now includes DB2 10.1

Updates with RDz UT 8.0.3

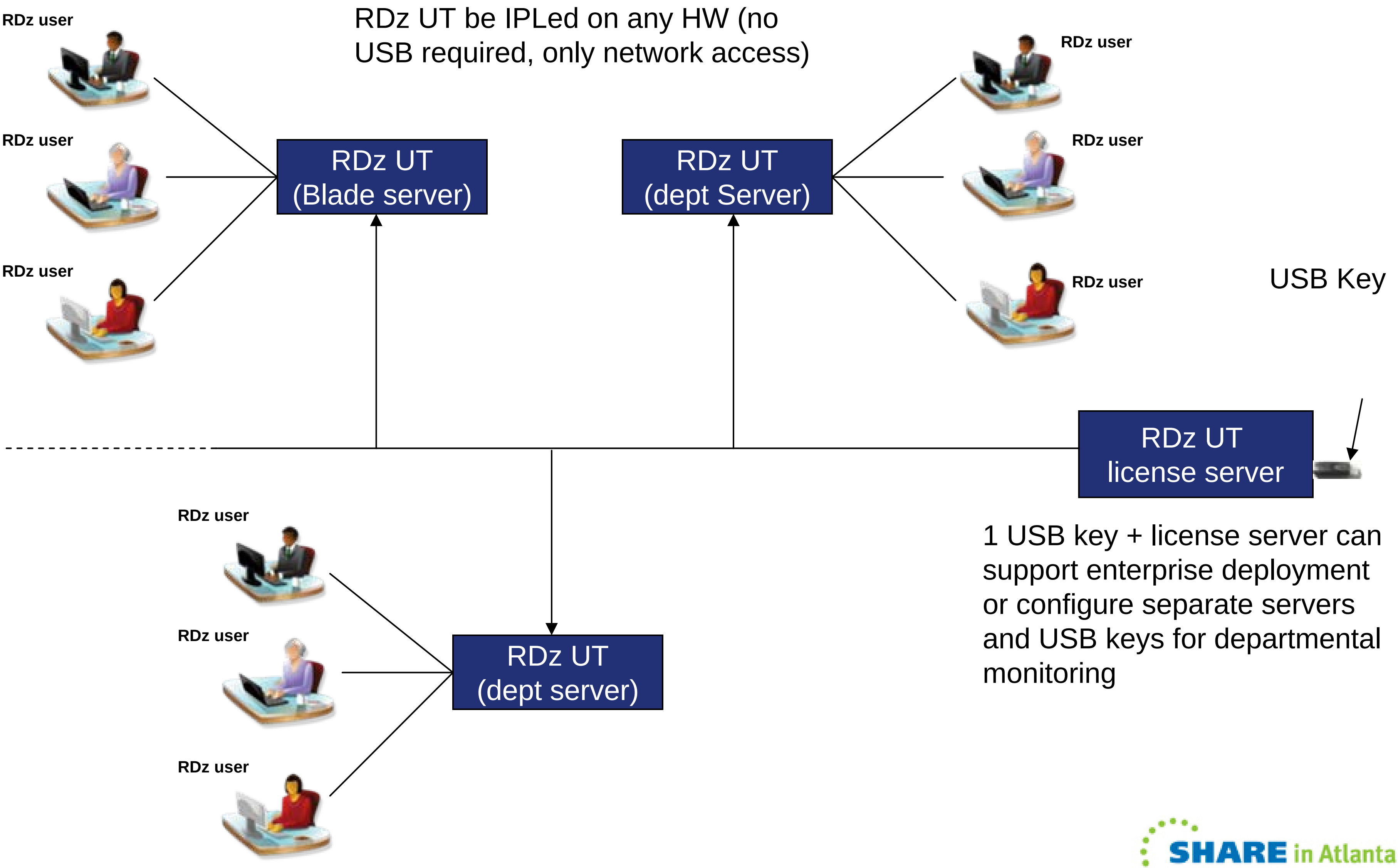
Available Nov, 10

- RDz UT License Server – Network-based, centralized license management system
  - Available for RDz UT Server Users
  - Easily IPL machines on hardware with limited/restricted USB access (e.g., x Series Blades in a blade center)
  - No longer require USB token to be physically plugged into every zPDT machine
  - Eases management and provisioning of multiple RDz UT instances
  - Future Plans: zBX and private cloud deployment configurations

- High Capacity License Tokens
  - Available for RDz UT server users
  - Allows multiple zPDT instances licensed from a single token
  - Supports up to 30 RDz UT servers (3CP) per token

- Scripts and tools to assist with setup and provisioning

Greater license flexibility



RDz UT be IPLed on any HW (no USB required, only network access)

1 USB key + license server can support enterprise deployment or configure separate servers and USB keys for departmental monitoring

OK … so what now

- Are you interested in testing your application changes on your PC/ off-mainframe?
  - RDz Unit Test provides a x86-based environment where you can test z/OS applications on a desktop or server machine
- Are your development systems overloaded?
  - E.g., My development system is so slow! It takes 1 hour to compile!!
  - E.g., There are not enough resources available for development
  - RDz Unit Test provides a low cost mechanism to create additional development capacity without mainframe HW upgrades.
- Do you have difficulty testing application changes? Is it too difficult or take too much time to make changes to your test environment? Are there too many requests for unique test environments than your ops team can fulfill?
  - RDz Unit Test provides a separate test environment under the development team's control allowing quick environment changes to be performed by development. The UT environment can be assigned to a small teams for unique test environments allowing more robust testing
- Are you being asked to use fewer MIPS for development
  - RDz Unit Test provides a 0-MIPS development environment, allowing development MIPS to be shifted back for additional production capacity
- Looking for a way to train new sys progs
  - RDz Unit Test provides a zOS environment where you can install, configure, toy with ideas to train new system programmers

Step 1 - Linux installation

- RDz UT requires a 64 bit host operating system
- 64 bit Open SuSE and 64 bit Red Hat Enterprise Linux require 64 bit hardware
  - RDz UT enforces the use of Open SuSE or RHEL as the host environment at install time
- Device drivers are biggest concern
  - Display
  - USB attachment
- Running RDz UT in a virtualized environment is possible
  - not certified or a supported configuration
  - Performance bottlenecks may be exposed depending on how physical resources are allocated
  - Some RDz-UT utilities must still be run natively
    - eg. SecureUpdateUtility

1090 (zPDT) installation / configuration

- Create file system to host z/OS volumes. Assign ownership of this filesystem to an id that will be used for IPL
- zPDT is a single executable file to install the 1090 base functions. The single file contains the following:
  - An sntl-sud rpm at the correct level (A driver for the 1091 token)
  - An shk-server rpm at the correct level (Another token program)
  - The primary z1090 rpm for openSUSE
  - The primary z1090 rpm for RHEL
  - An installer program that automatically installs the rpms. The correct z1090 rpm (RHEL or openSUSE) is automatically selected for your base Linux system.
- Install the zPDT code by running the executable file from root
  - Two prerequisite rpms are installed.
  - The 1090 executable files are loaded into /usr/z1090/bin.
  - A set of man files is loaded into /usr/z1090/man.
- Minor configuration of system files is required to complete installation

Setup a new z/OS release

- z/OS volumes are represented as Linux files
- z/OS system volumes (files) are shipped with RDz-UT
- Can create additional work volumes (files) if required
- User that ipls system “owns” the z/OS volume (file) resources
- We must create a device mapping file to map Linux files to z/OS volume names and device numbers.
  - Used to IPL z/OS system
  - Also defines CPU and memory configuration, network attachment, 3270 terminals, and other I/O devices
  - Called aprof1 by convention, but any name will do
- We verify configuration with a utility called AWSCKMAP
  - AWSCKMAP /media/rdzut/aprof1
  - Note 1091 token must be plugged in and initialized for this command to work

RDz UT Device maps – defining devices to z/OS

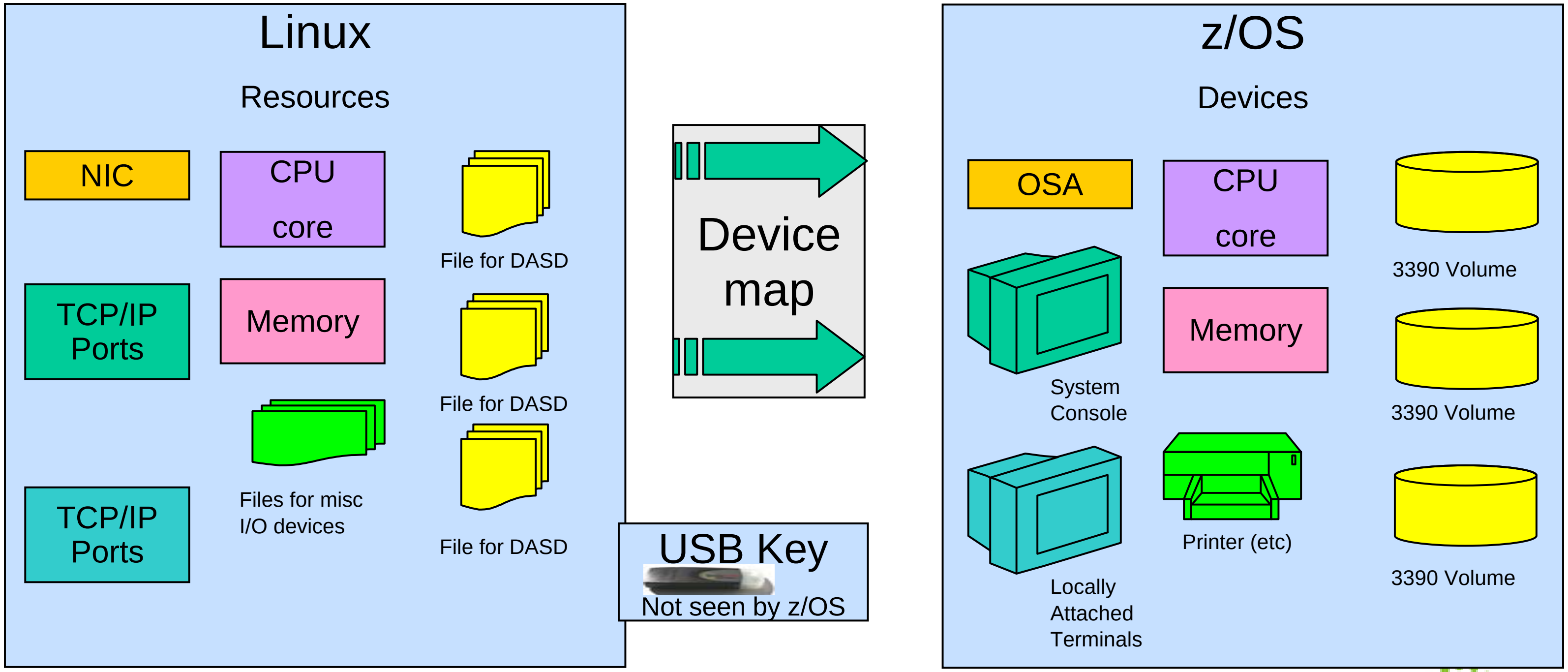
“Devmap” – will map the entities known to Linux, to the devices known to z/OS

z/OS volumes are contained in Linux files

z/OS Communication device is a Linux ethernet card or a logical tunnel device

z/OS printer or card readers can be Linux files

… Other devices are also possible such as SCSI attached tape drives.



Ipling the system

- Your first IPL
  - Open a Linux terminal window
  - Start zPDT emulator
    - awsstart aprof1 defines 1090 environment and terminals using devmap
  - Define some 3270 terminals
    - Powerterm & used for master console
    - Powerterm & used for local 3270 channel attached terminal
  - ipl 0a80 parm 0a82cs
    - ipls z/OS on sysres volume 0a80
    - points to ipl parameters on volume 0a82
    - points to initialization parameters in PDS member IEASYS82
      - System will use SYS1.PARMLIB concatenation defined when ADCD system was generated. In this case, for the first IPL IEASYS82 will be picked up from the PARMLIB supplied with the ADCD – 'ADCD.Z111.PARMLIB'.
      - IEASYS82 does a cold start of JES2 and brings up the system with a default configuration for VTAM and TCP/IP. Until customization of TCP/IP is completed, the only terminal interface that will work is a VTAM 3270 channel attached terminal defined with Powerterm (or x3270).

Operational semantics

- System volumes may be created and maintained by system programmers
  - z/OS volumes
  - Page, spool
  - Subsystems – CICS, IMS, DB2, WebSphere, MQ
- RDz-UT systems can be started, stopped and manipulated with little required z/OS skills
  - Startup/ Shutdown is automated through parmlib and MPF exits
- Common z/OS developer activities managed from RDz IDE – no native z/OS access required
  - File allocation – Sequential, PDS, Unix
  - File reads, edits
  - Compiles
  - Spool file manipulation
  - Job submission, execution
  - Native green screen access via HOD 3270 emulator
  - Access to DB2 resources via Data Explorer
  - Access to CICS resources via CICS Explorer

Multi User RDz UT environment

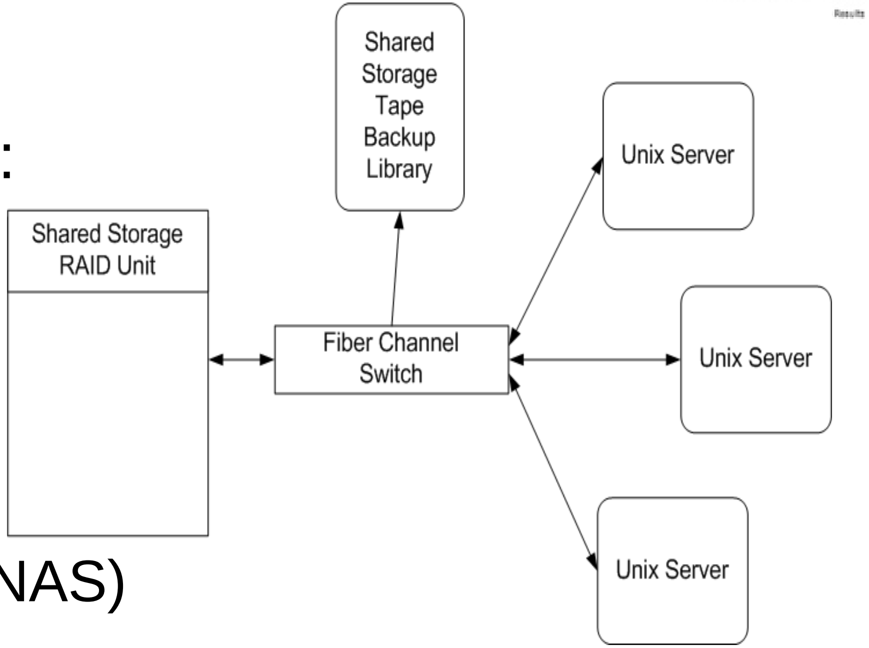
Goals:

- ✓ Common set of zOS volumes offering:
  - Common set of capabilities
  - Allow deployment to multiple RDz UT instances
  - Common set of data
  - Common set of software
- ✓ Store in a single place / common access storage
- ✓ Provision multiple RDz UT instances via a single client/admin mandate

Store in a single place / common access storage

Potential Hardware configurations:

- > Shared DASD
- > Storage Area Network (SAN)
- > Network Addressable Storage (NAS)



Shared Storage Tape Backup Library

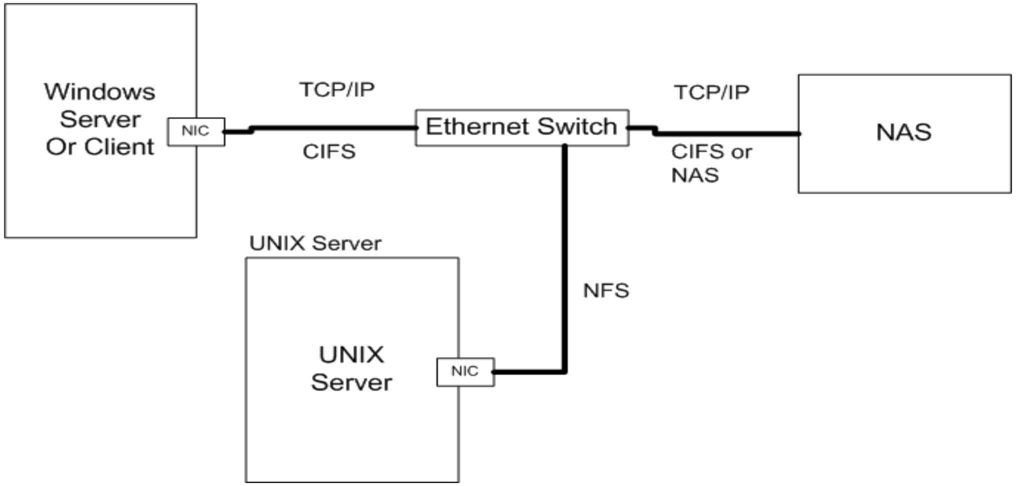
Shared Storage RAID Unit

Fiber Channel Switch

Unix Server

Unix Server

Unix Server



Windows Server Or Client

NIC

TCP/IP

CIFS

Ethernet Switch

TCP/IP

CIFS or NAS

NAS

UNIX Server

NFS

UNIX Server

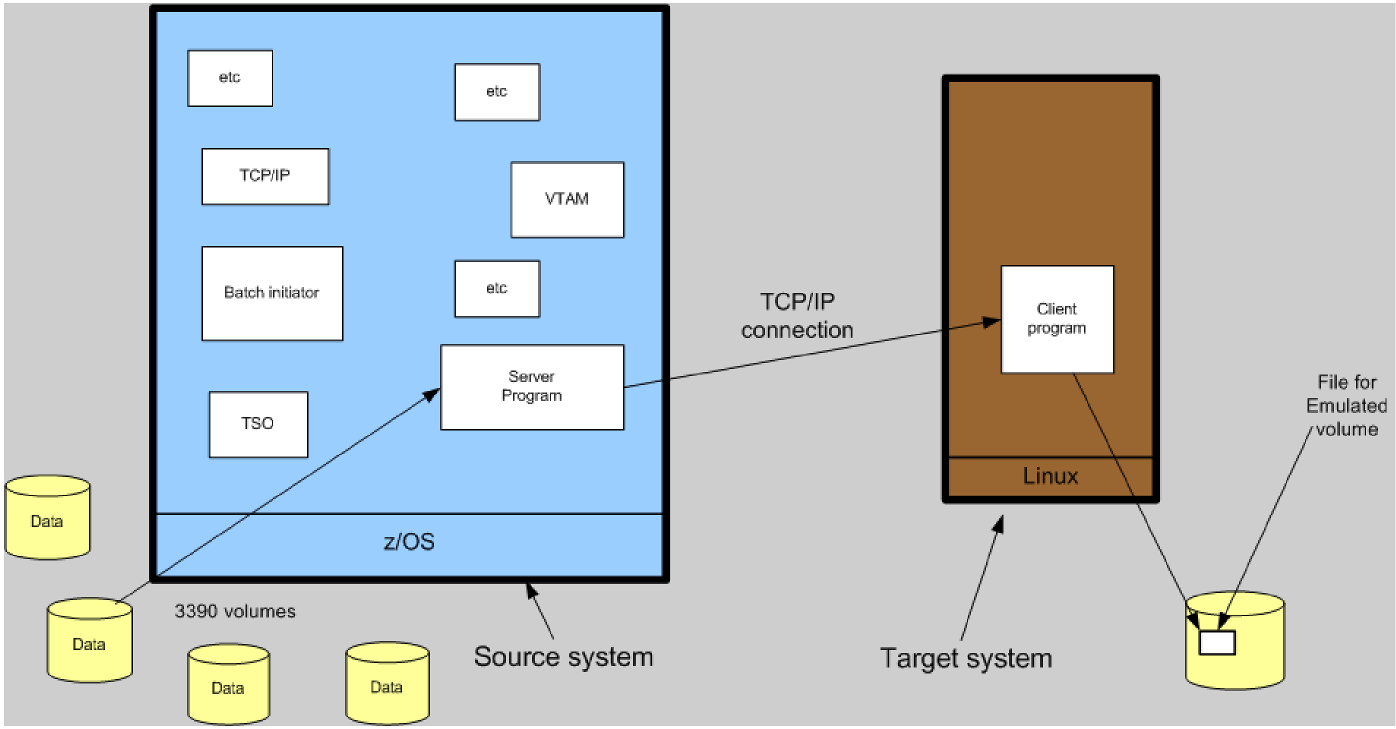
NIC

DEMO

- We will:
  1. Start the zPDT
  2. Define some 3270 terminals
  3. IPL the z/OS system
  4. Start DB2, CICS, MQ, and RDz daemons
  5. Create RDz host connection and access RDz server on RDz-UT
  6. Access z/OS PDS’s, Unix files, and Spool from RDz
  7. Access DB2 from RDz using Data Explorer
  8. Access CICS from RDz using CICS Explorer
  9. Open a host emulator session from RDz
  10. Start a PCOMM 3270 session
  11. Logon to TSO and access ISPF
  12. Use SDSF to view SYSLOG, SPOOL, active users
  13. Logon to CICS
  14. Do an orderly shutdown of z/OS and subsystems
  – Shutdown the zPDT emulator

Provisioning the RDz-UT system

- Create new z/OS volumes with RDz-UT utilities
- Use FTP to send individual files and datasets to RDz-UT system
- Use ADRDSSU DUMP/ RESTORE to move volume contents
- Use DASD migration utility provided with RDz-UT to create Linux volume file from native z/OS volume


etc
etc
TCP/IP
VTAM
Batch initiator
etc
Server Program
TSO
z/OS
Data
Data
3390 volumes
Data
Data
Source system
TCP/IP connection
Client program
Linux
Target system
File for Emulated volume