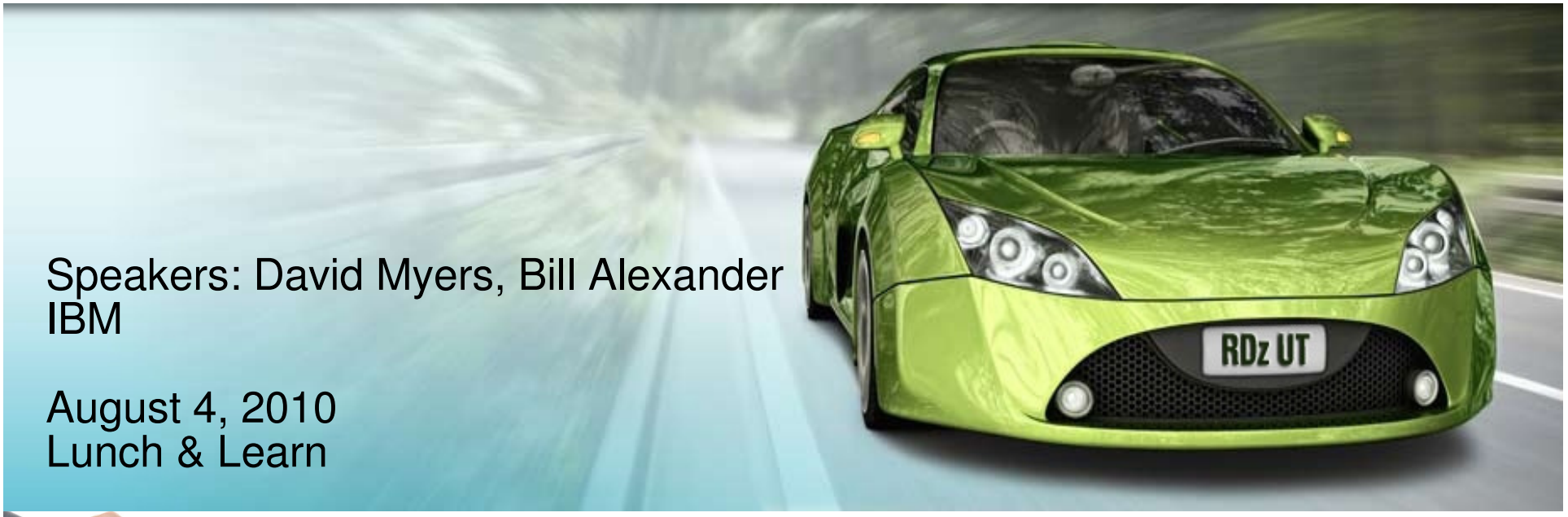
Rational Developer for System z Unit Test Feature: a new offering providing mainframe development flexibility



Speakers: David Myers, Bill Alexander
IBM

August 4, 2010
Lunch & Learn

SHARE in Boston

Agenda

- What is RDz Unit Test?
  - What comes with UT?
  - How do I license UT?
- Usage scenarios
- System requirements
- Installation/configuration
- Demo

Business constraints with mainframe development today

Which limits the amount of System z production workload coming online

"Operations tell me it will take two months to get my test system allocated."

"My development capacity charge-back is consuming my entire budget. I can't spend on tools."

"I can only test my batch applications in offline hours. Online apps consume the 9-5 cycles."

"We don't have the capital budget to obtain more mainframe test resources for my developers."

"It is difficult for my developers to learn the mainframe. Operations controls can prevent experimentation by developers.."

"I can't even work on Mondays! Production workload kicks me off."

"I want to try out creating Event Processing and ATOM apps, but my system isn't scheduled for a CICS update till 2012."

"The Mainframe isn't cool anymore."

Today's System z development environment

ISPF has provided consistent tooling for decades…but it is limiting

Drawbacks:

- Constant connection to mainframe is required
- Development shares machine with production use; gets lower priorities
- Inability to create cross-platform components
- ISPF green screen UI is unappealing to new hires
- MIPS usage for development vs. production usage

Today's mainframe development environment with RDz and ISPF

RDz provides relief…but customers want more

Modern IDEs add value:

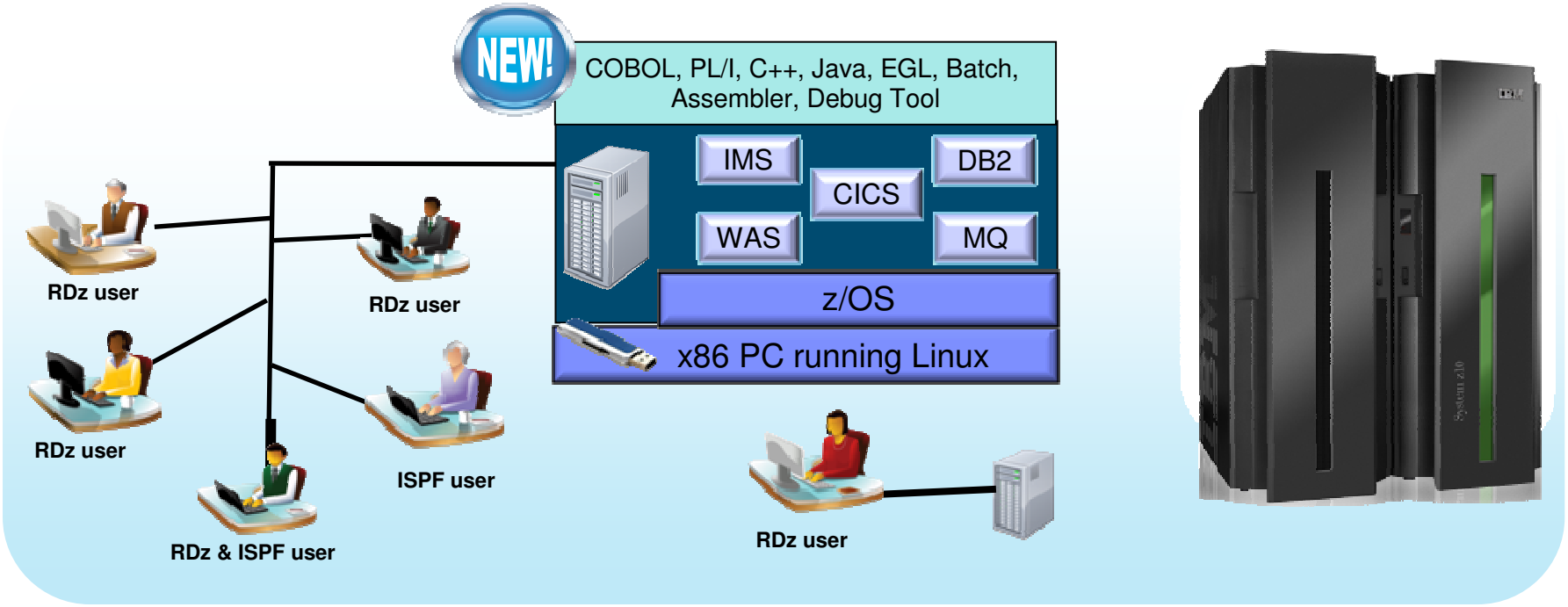
- Productivity enhancing tooling; more attractive tooling for new developers
- Ability to offload some development MIPS usage
- Integration with complete application lifecycle tools

But challenges remain:

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

Announcing the RDz Unit Test Feature

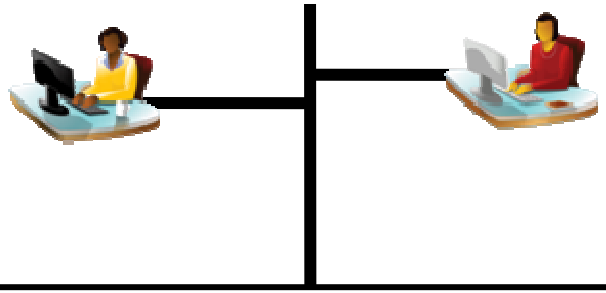
The ultimate in modern application development for System z

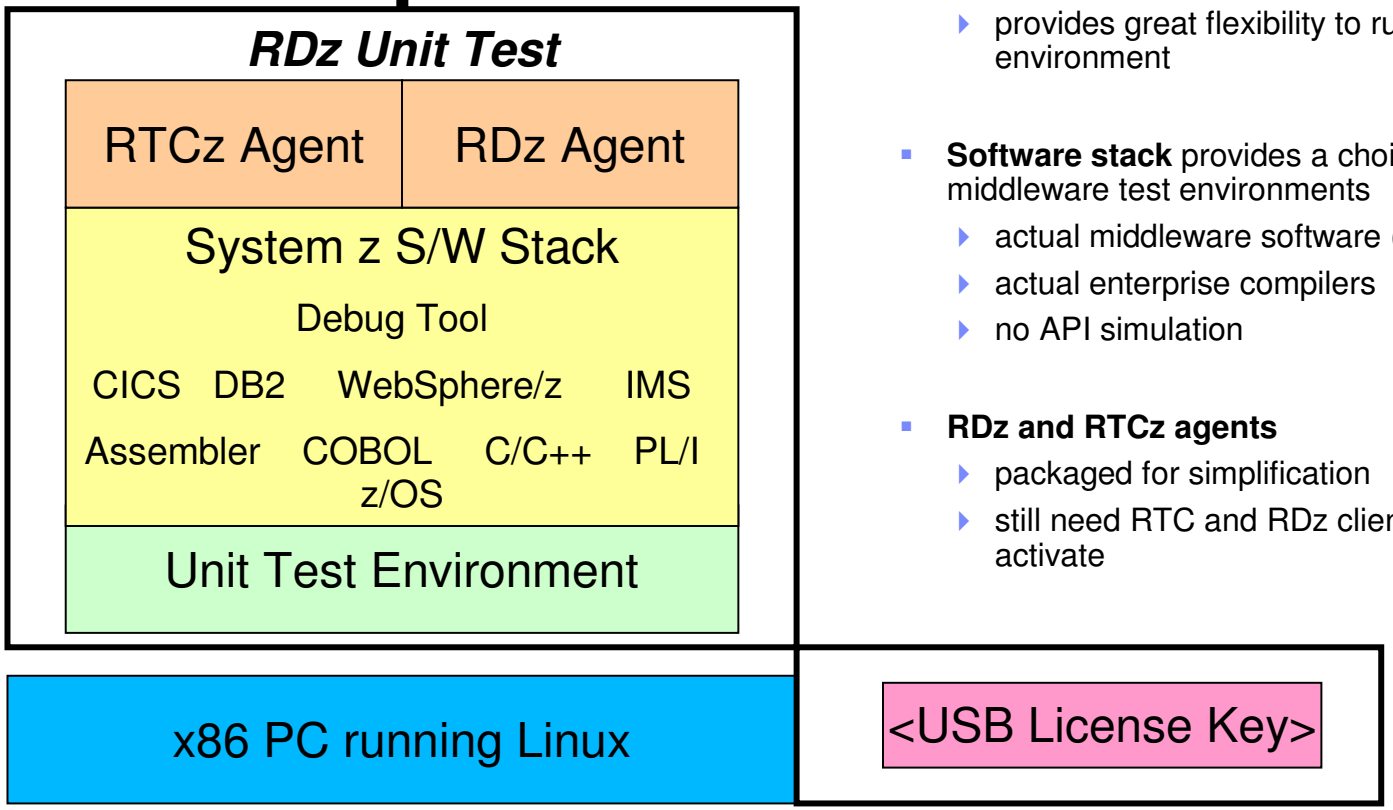


- Liberate developers to rapidly prototype new applications
- Develop and test System z applications anywhere, anytime!
- Free up mainframe development MIPS for production capacity
- Eliminate costly delays by reducing dependencies on operations staff

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 Unit Test 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
  - List-directed IPL
  - External Time Reference (ETR)
  - Server Time Protocol (STP)
  - MIDAWs
  - Logical channel subsystems
  - HiperSockets™
  - Multiple I/O paths per device
  - Not all CHSC functions are supported
  - Some IBM System z Crypto Express2
  - Some IBM 3088 CTC device

- **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.**

How will RDz UT help me?

- 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 too difficult or takes 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 (catastrophic event testing, system failure, impact analysis, etc)?
  - 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 additional development resources to be easily allocated

Licensing options

The Rational Developer for System z Unit Test environment is sold in two levels to offer flexibility of workload and configuration:

- *Standard:* enables a single virtual unit test engine environment
  - A single test engine environment is appropriate for small machines, such as a developer laptop.
  - One engine will facilitate most types of traditional System z application testing such as Batch, CICS, IMS, DB2, COBOL, PL/I, and Assembler.

- *Specialty:* enables a three virtual unit test engine environment
  - A three test engine environment is appropriate for server machines supporting a team of developers or specialty workload.
  - The three engines can be configured to test applicability of application code to zAAP or zIIP processors in addition to normal development and unit testing.
  - Three engines will allow a wider range of testing options including all the options of the Standard configuration, but also facilitating testing of Java, WebSphere, and more data processing options in DB2.

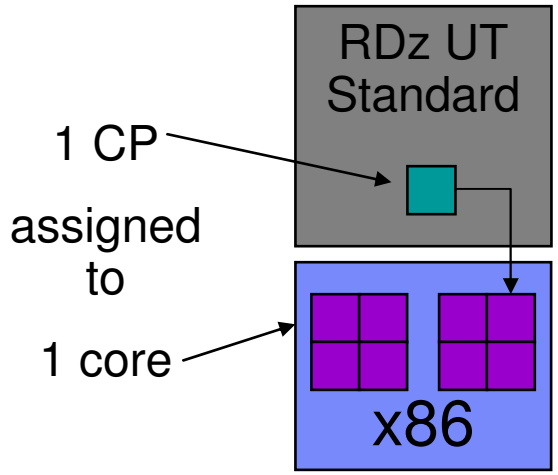
Note: In a multiuser configuration all users must be licensed to the same IBM Rational Developer for System z Unit Test level.

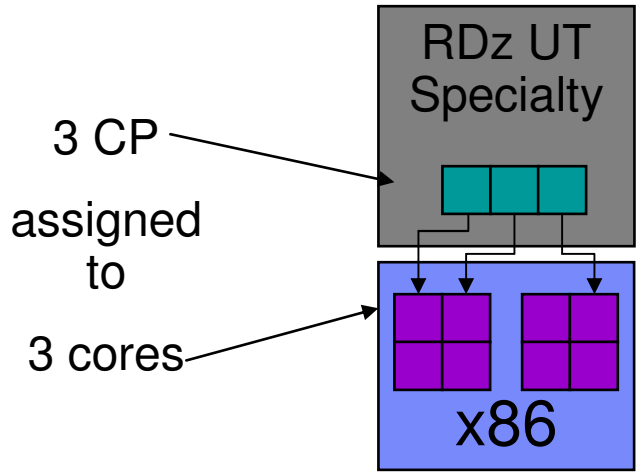
Standard vs. Specialty

Think as follows (simple examples)

- Standard allocates work to one (1) x86 core
- Specialty allocates work to three (3) x86 cores

- More cores allows more users, workload, performance, multitasking, etc





RDz UT environment licensing

- RDz Unit Test is enabled via a USB key
  - Key makes Unit Test feature operational
- USB key is shipped disabled, activated via Rational Key Center
- Physical delivery required for key and stack (DVDs)
  - No electronic download available

- Licensing
  - Authorized User Single Session (perpetual)
    or
    Authorized User Fixed Term Single Session (1yr)
  - Each user accessing the UT environment needs a license
  - RDz is a pre-requisite
- License to
  - Development workloads only
  - Specifically non-production in license (i.e., no end users of the applications allowed)
- No guarantee or warrantee to replicate mainframe function in entirety (see limitations listed earlier)
  - E.g., limited crypto subsystem support
- Development license for S/W stack (z/OS and middleware) content
  - No phone or defect support provided with purchase
  - Defects must be replicated in production system and fed through production support process
  - Online support forum available for QA

What does Authorized User Single Session mean??

- Legal Definition:
  - “Authorized User Single Session is a unit of measure by which the Program can be licensed. An Authorized User is a unique person who is given access to the Program. The Program may be installed on any number of computers or servers, but if the Authorized User simultaneously accesses the Program multiple times, either on the same or on multiple computers, each separate simultaneous access requires a separate entitlement. Licensee must obtain separate, dedicated entitlements for each Authorized User accessing the Program in any manner directly or indirectly (for example: via a multiplexing program, device, or application server) through any means. An entitlement for an Authorized User is unique to that Authorized User and may not be shared, nor may it be reassigned other than for the permanent transfer of the Authorized User Session entitlement to another person.
  - Any computing device that requests the execution of or receives for execution a set of commands, procedures, or applications from the Program or that is otherwise managed by the Program is considered a separate User of the Program and requires an entitlement as if that device were a person.

- What does that mean for RDz UT?
  - Assume each user license is a single TSO login
  - Each (TSO) user login needs a license

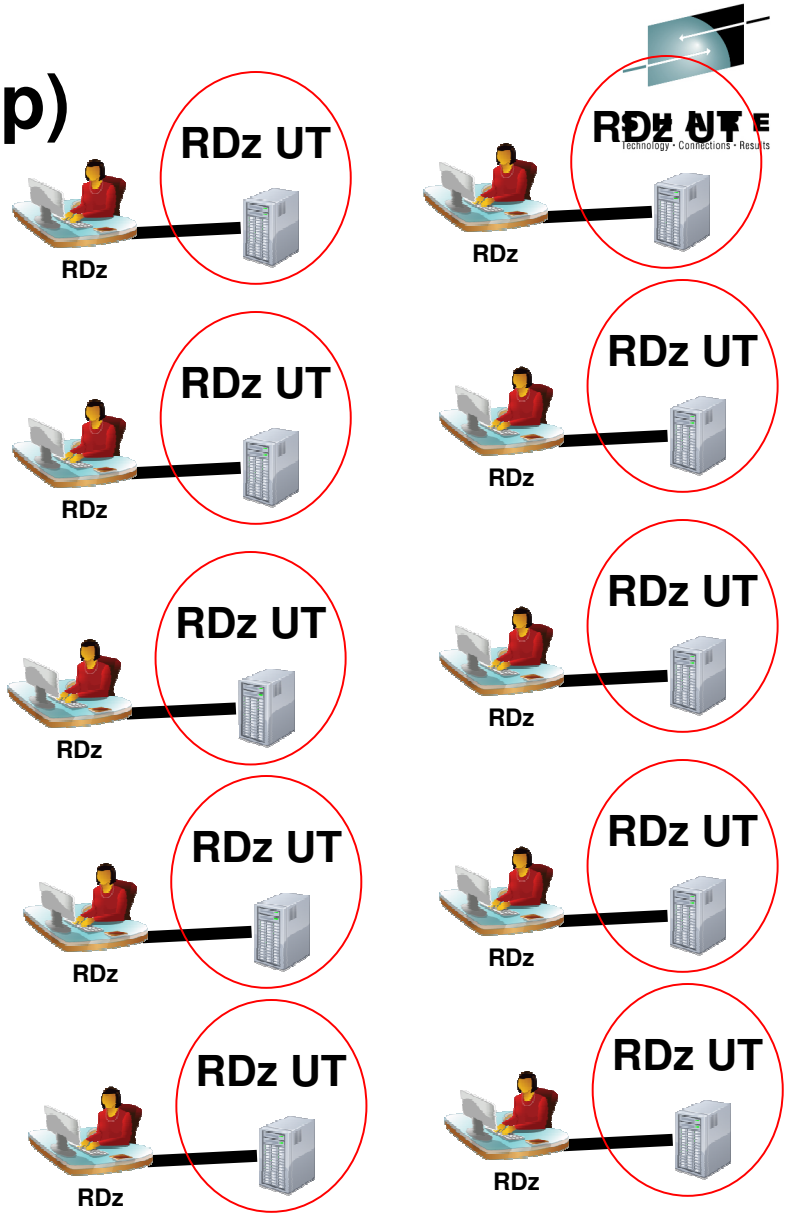
- How would I audit?
  - Run a report on defined RACF IDs in the system. Ensure IDs are tied to user count

- Floating user licenses (total number of active users) are in the roadmap for future release

Licensing Examples (desktop)

- 10 Developers
- Configuration: Each wants a desktop install

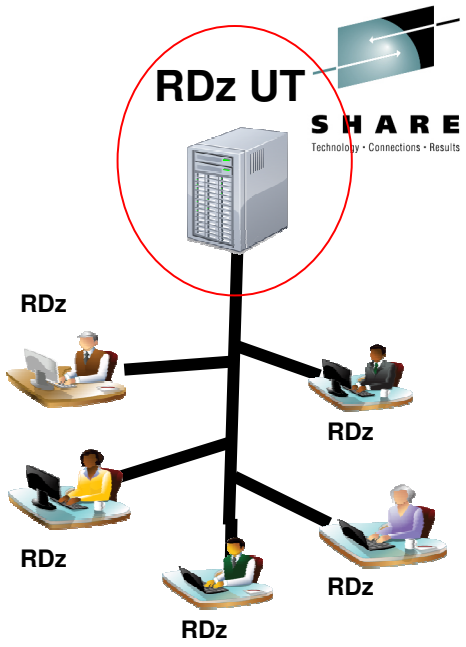
- Recommendation:
  - 10 RDz with Java or RDz with EGL
    - 1 for each developer
  - 10 RDz UT standard licenses
    - 1 for each developer
  - 10 USB keys
    - 1 for each desktop

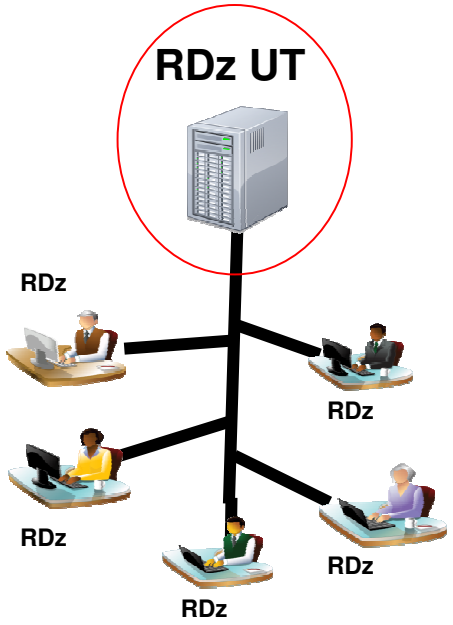


Licensing Examples (server)

- 20 Developers
- Configuration: 2 servers for 10 users

- Recommendation:
  - 20 RDz with Java or RDz with EGL
    - 1 for each developer
  - 20 RDz UT specialty licenses
    - 1 for each developer
  - 2 USB keys
    - 1 for each server

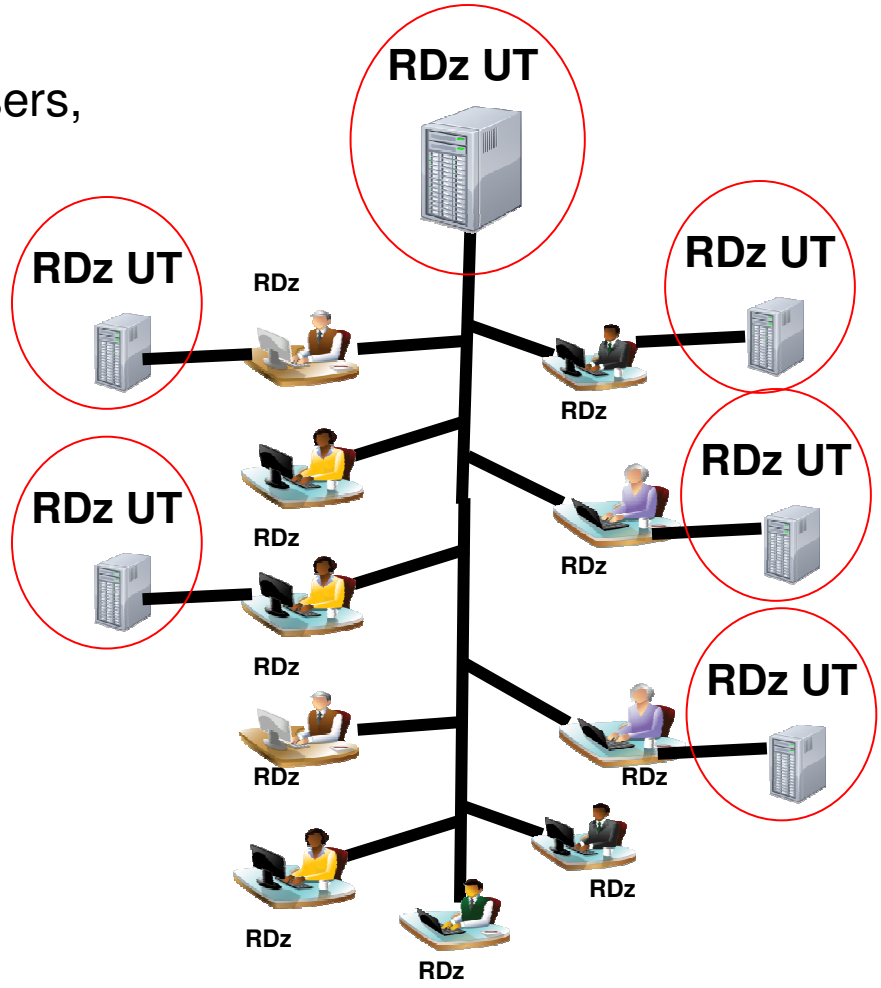




Licensing Examples (mixed use)

- 10 Developers
- Configuration: 1 shared server for all 10 users,
  5 desktop installs for 5 individual users

- Recommendation:
  - 10 RDz with Java or RDz with EGL
    - 1 for each developer
  - 15 RDz UT licenses
    - 10 RDz UT specialty (for server)
    - 5 RDz UT standard (for desktops)
  - 6 USB keys
    - 1 for server
    - 5 for desktops



Ordering considerations

- Each order needs at least 2 parts:
  - 1 license (D/E parts)
  - Some number of USB keys (B parts) [physical media packs]

- USB Keys (B parts) can be ordered after PO from Passport Advantage (like any physical media pack)
- USB Keys are sold in lots of 1, 5, 10, 25, or 50

- Make sure to choose the correct part to simplify orders!!
  - Qnty 10 of “1 USB key” == 10 boxes of 1 key each
  - Qnty 2 of “5 USB key” == 2 boxes of 5 keys each
  - Qnty 1 of “10 USB key” == 1 box of 10 keys
- Passport Advantage entitled customers to one complimentary (0-cost) B part order

Top 5 Questions

- **1) What is the maximum number of developers a RDz Unit Test server can support?**
  - This can vary depending on the underlying hardware and workload being tested. Desktop/Laptops can typically support 3-5 users. Low end server class machines can support 15-30 users. At this time, IBM has not provided a detailed analysis of RDz UT server sizing.
- **2) How can I get test data for use in RDz Unit Test?**
  - Customers can use existing tools like IDCAMS, DB2 utilities, etc. to extract test data and then download it to the RDz Unit Test machine. RDz Unit Test provides a volume duplication utility that can download an entire 3390 data volume as needed into the UT environment
- **3) Can I run other levels/backlevels of the middleware provided?**
  - Not currently. This is being investigated to determine whether this capability could be provided in the future.
- **4) Can I use other IBM tools in the UT environment?**
  - ESW zOTC or zMLC tools (like PD tools) cannot currently be used. There is no mechanism for IBM to license the software to the UT environment. Passport Advantage products can be used in the RDz UT environment. We are investigating providing this capability in the future.
- **5) Can I run third party software?**
  - Yes, if the third party license allows this. Customers must work with their software vendor to determine licensing considerations.

Additional Questions

- **6) Does this require system programming skills?**
  - z/OS does require system programming skills to set up the development and/or testing environments.
  - You can usually set up one box and transfer the configurations to another box easily.
- **7) What about security?**
  - RACF is installed, but with minimal configuration.
  - The sample configuration guide has suggestions for basic security.
  - Security is a site choice. The ability to customize z/OS on a platform designed for individuals or small teams may:
    - *Provide better testing opportunities*
    - *Provide customization for individual productivity gains*
    - *Provide opportunities to learn about z/OS fundamentals*

RDz UT Usage Scenarios – Common threads

- Where do builds take place?
  - Compilations can be done in RDz UT or the mainframe
  - Production builds should always take place on mainframe
- Where does source code reside?
  - This is a major design point.
    - Moving data can be manual, semi-automated or fully distributed.
- How does one move source or data from/to RDz UT?
  - Move data from/to RDz UT much like you would to an LPAR
    - RDz drag and drop, FTP, sftp, XMIT, NFS, DFS, etc
    - Non-VSAM files need format conversion before & after transfer
    - RDz UT provides facilities for transferring whole disks to RDz UT

RDz UT Usage Scenarios Examples

Scenario	Source Location	Build Location	Description
Mainframe Build	Mainframe Only	Mainframe	Outputs copied to RDz UT. All source updates made on host.
RDz UT Build	Mainframe and RDz UT	RDz UT	Source copied to and changed on RDz UT for local development. Source must be returned to host SCM.
Distributed Build	Mainframe and RDz UT and possibly elsewhere	Determined by SCM. Example: RTCz build agents	SCM takes care of source and build locations. Build requests are serviced on RDz UT, host or third party build machine. Outputs are made available to RDz UT for testing. This scenario automates the processes of the other two and is least prone to source code synchronization issues.

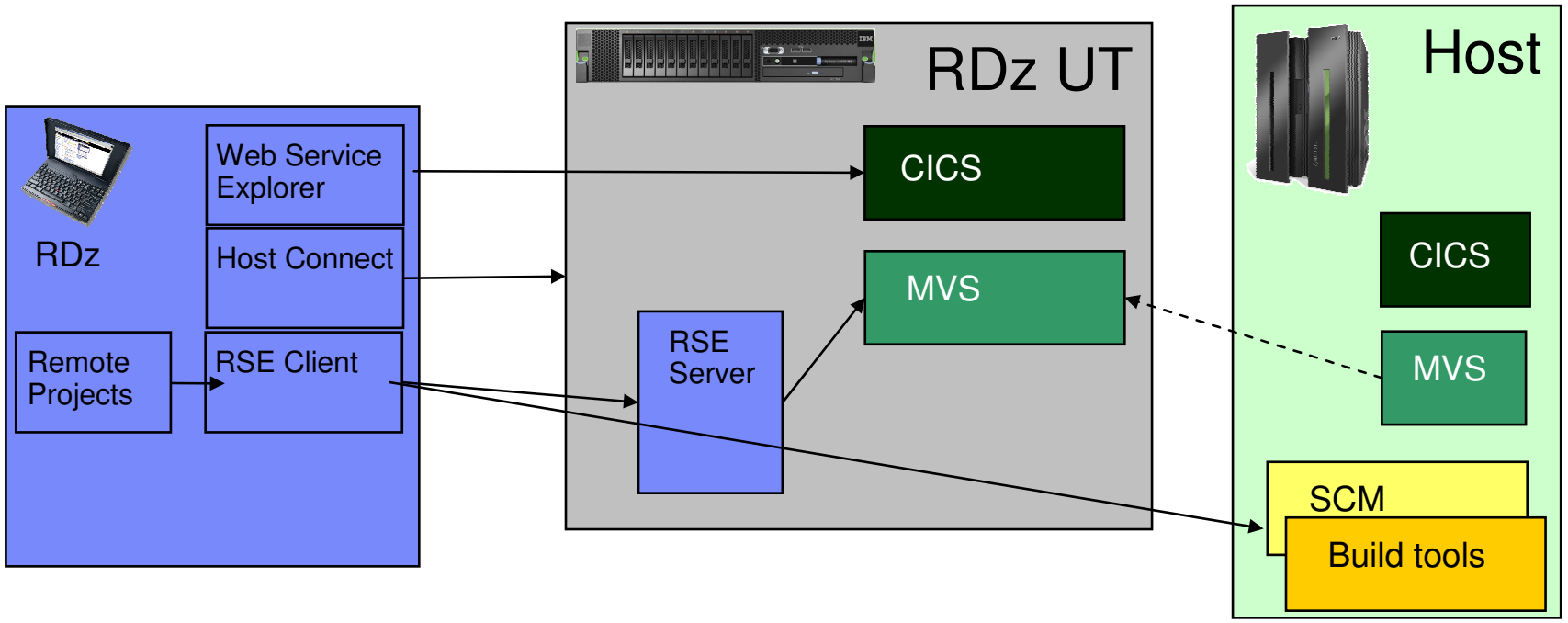
Mainframe build

Unit Test of code maintained on the host:

Develop code on mainframe

Copy compiled code and required data to RDz UT

Run tests on RDz UT as you would on a test LPAR



RDz UT Usage Scenarios – Mainframe build

- **Unit Test of code maintained on the host:**
  - Advantages
    - Conceptual simplicity
    - No need to manage source code on RDz UT
      (except as needed by debugger)
  - Disadvantages
    - Need limited resource reduction on mainframe
    - Programmer actively works with two different systems
      (made relatively transparent with RDz)

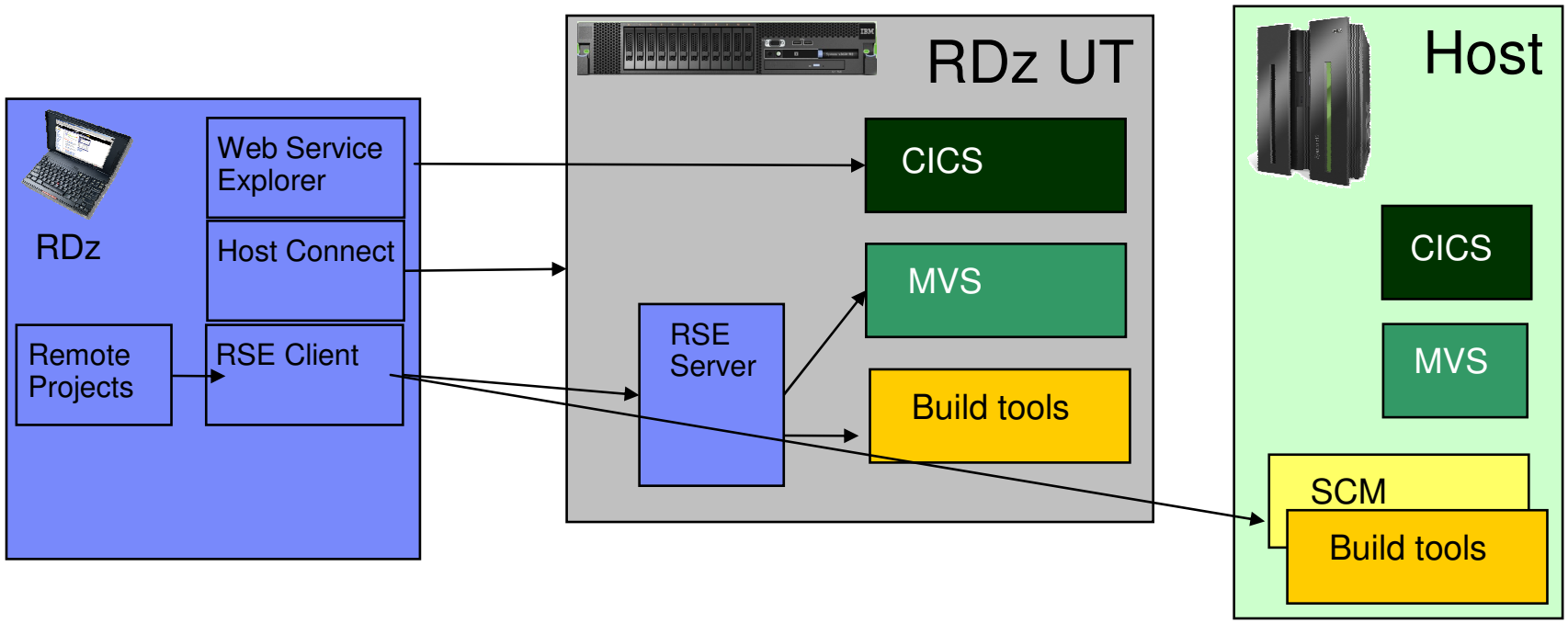
RDz UT based build

Development on RDz UT with host-based SCM:

Copy code and data to RDz UT as needed

Use RDz or other methods to run a standard compile/debug cycle

When tests and changes are complete, merge changes back to mainframe



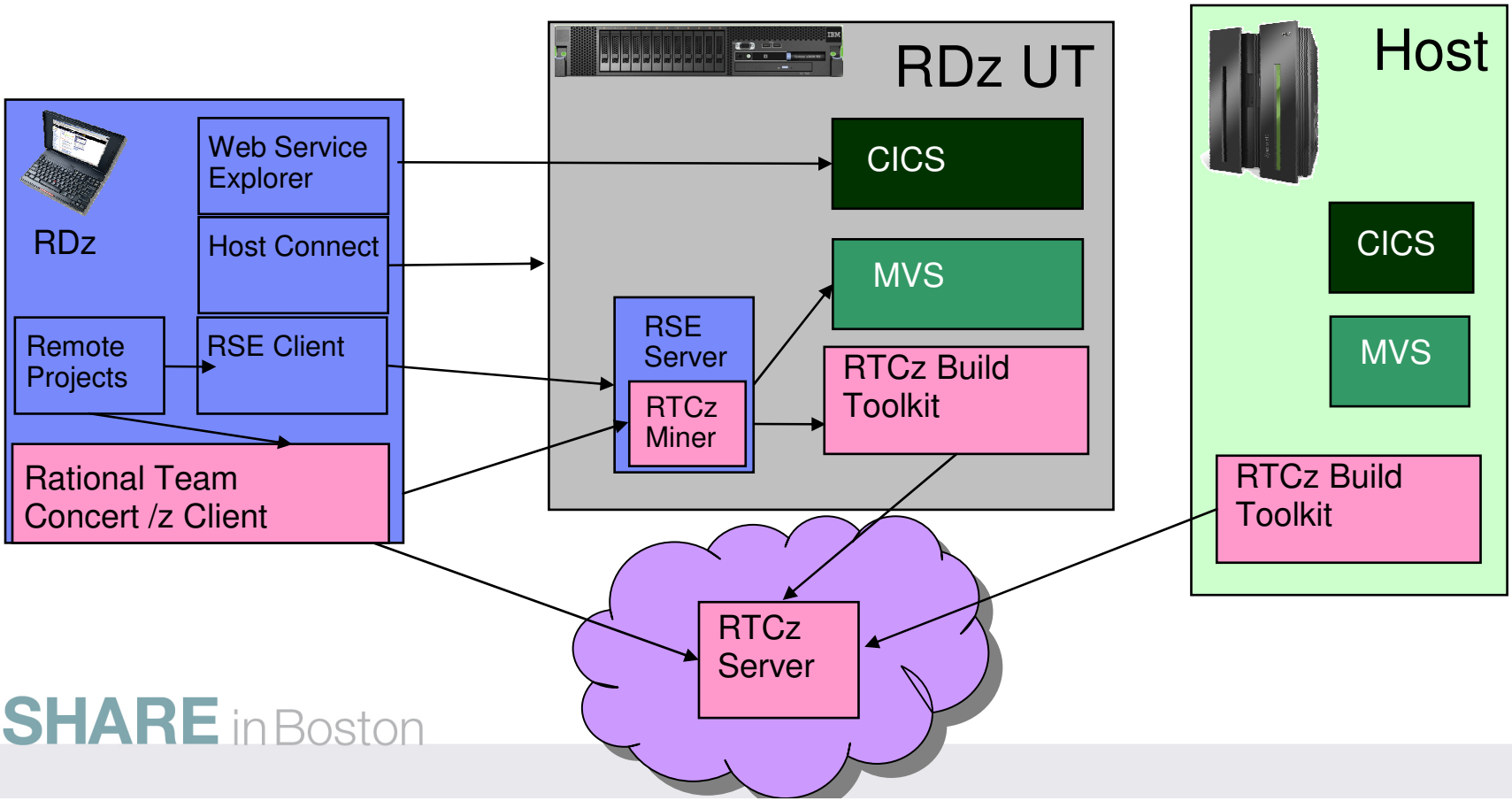
RDz UT Usage Scenarios – RDz UT based build

- **Development on RDz UT with host-based SCM**:
  - Advantages
    - Significant reduction in mainframe resource usage and contention
    - May allow developers to modify the environment to suit their individual needs more than they would be allowed to do in a restricted mainframe environment.
  - Disadvantages
    - Maintaining similar build environments across systems
    - Sharing code at the host requires check-out/check-in processes
    - May introduce additional backup and auditing requirements for RDz UT systems.

Distributed SCM and Build

Development on RDz UT with distributed SCM:

SCM manages movement of code to and from mainframe.
Builds may be done either on RDz UT or on mainframe (depending on SCM).
Test outputs installed by SCM on RDz UT.
Use RDz or other methods to run a standard compile/debug cycle



RDz UT Usage Scenarios – Distributed SCM and Build

- **Development on RDz UT with distributed SCM**:
  - Advantages
    - Most flexible for developers. No need to manage source code location or move build outputs.
    - Check-out/Check-in processes are built in to SCM.
    - Significant reduction in mainframe resource usage and contention
    - Allows developer flexibility in work methods and environment.
    - Most likely allows a standard client installation on the RDz UT system that is easily maintained.
  - Disadvantages
    - Initial complexity of setting up SCM at mainframe. (Not a function of RDz UT per se)
    - May introduce additional backup and auditing requirements for RDz UT systems.

RDz UT host machine specifications

- Underlying Linux host
  - Red Hat Enterprise Linux 5.3 (RHEL 5.3)
  - OpenSUSE 10.3, 11.0, and 11.1
  - Note: IBM does not support installation on other distributions.
- Base machine must have:
  - at least 3 GB of real memory:
    - 1 GB is required for the 64-bit Red Hat or openSUSE Linux
    - 1-2 GB is required for the System z operating system, depending on the size of the development system
  - at least 80 GB free disk space available after Linux
  - Note: RDz UT has been tested on:
    - Lenovo ThinkPad W Series
    - IBM System x 3500 M1, 3500 M2, 3650 M1, and 3650 M2

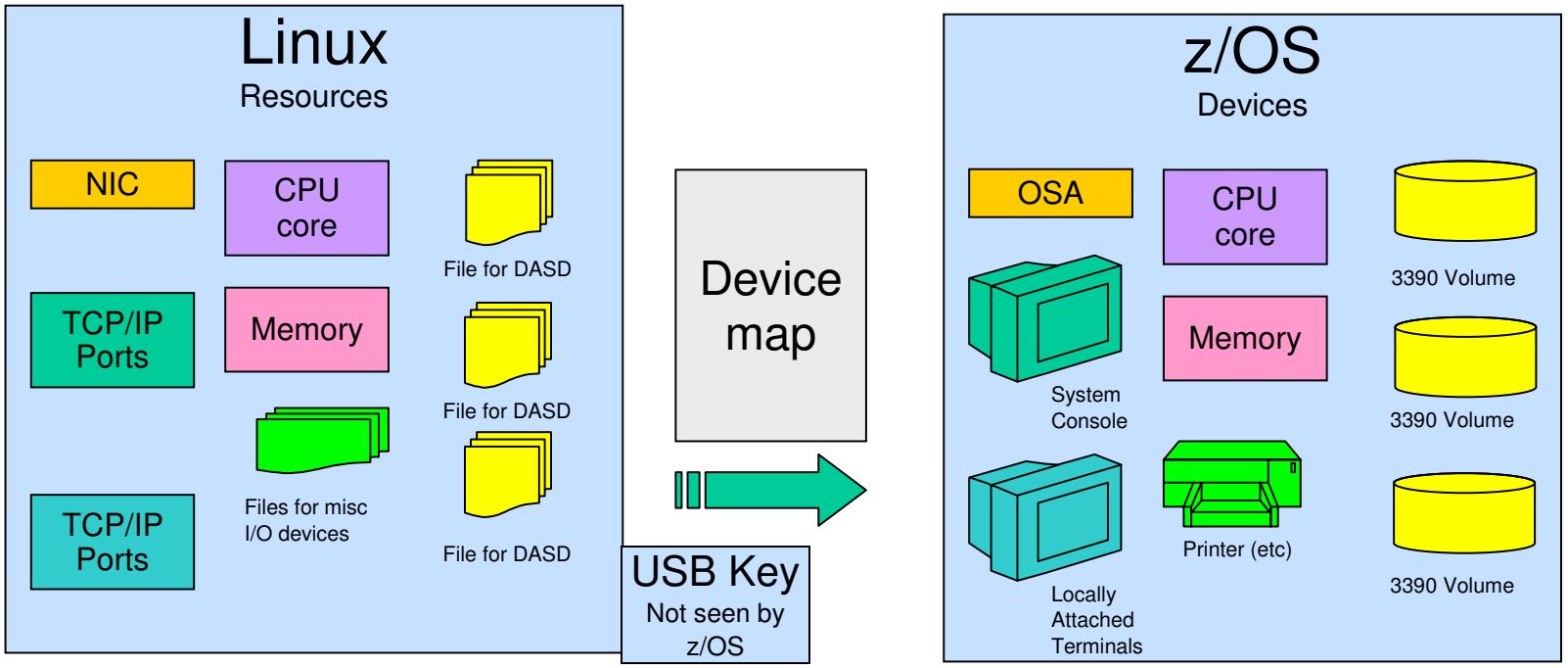
RDz UT installation overview

- The installation steps include:
  - Install Linux, including x3270 (or another 3270 emulator)
  - Create group ibmsys and userid ibmsys1
  - Install the System z Personal Development Tool package
  - Customize several Linux files (sysctl.conf, /etc/profile.local, /etc/profile, .bashrc)
  - Install a sample devmap
  - Activate your USB license keys
  - Install z/OS and other System z software
    - Unzip or untar the volumes
    - Customize or create a devmap
    - Start the zPDT and IPL your operating system
    - Install and configure RDz host, CICS, DB2, etc

RDz UT Device maps – defining devices to z/OS

"Devmap" maps entities known to Linux, to 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.



RDz UT Device maps – defining devices to z/OS

```
[system]
memory 2000m                    # define 2000 MB System z
processors 2                    # use 2 or 3, if appropriate
3270port 3270                   # port number for TN3270 connections

[manager]
name aws3274 0002               # define a few 3270 terminals
device 0700 3279 3274 mstcon
device 0701 3279 3274 tso
device 0702 3279 3274 tso

[manager]                       # define network adapter (OSA)
name awsosa 22 --path=F0 --pathtype=OSD    # QDIO mode
device 404 osa osa --unitadd=0
device 405 osa osa --unitadd=1
device 406 osa osa --unitadd=2

[manager]                       # define DASD volumes
name awsckd 0001
device 0a80 3390 3990 /home/ibmsys1/z1090/disks/ZBRES1
device 0a81 3390 3990 /home/ibmsys1/z1090/disks/ZBRES2
device 0a82 3390 3990 /home/ibmsys1/z1090/disks/ZBSYS1
device 0a83 3390 3990 /home/ibmsys1/z1090/disks/ZBUSS1
```

RDz UT setup and customization

- RDz UT: Configuration guide (SC14-7281)
  - Contains basic instructions and examples for system customization
    - Isolate base configuration files from customized files
    - Store user data on a separate disk
    - Configure TCP/IP settings
    - Configure security for critical system files
    - Create user IDs
    - Other minor but common changes to z/OS
  - System programmer assistance will be required to replicate specific conventions, security models, subsystems, etc

- Related Redbooks: System z Personal Development Tool

RDz UT maintenance

- The z/OS distribution uses a system of concatenated libraries:
  - User libraries — USER.LINKLIB
  - Distribution developers — ADCD.Z111.LINKLIB
  - z/OS system data sets — SYS1.LINKLIB

  - User libraries will be retained between release levels
  - Distribution library changes may require modifications to previously completed customizations
  - System libraries should ONLY be updated through SMP/E install, maintenance processes, or similar processes