

# **z/OS Basics: z/OS UNIX Shared File System environment and how it works**

Jim Showalter  
IBM

August 11, 2011  
Session Number 9875

# Trademarks

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

- DFS
- DFSMS
- DFSMSdss
- IBM
- MVS
- RACF
- RMF
- S/390
- z/OS
- zSeries

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.

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

## Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

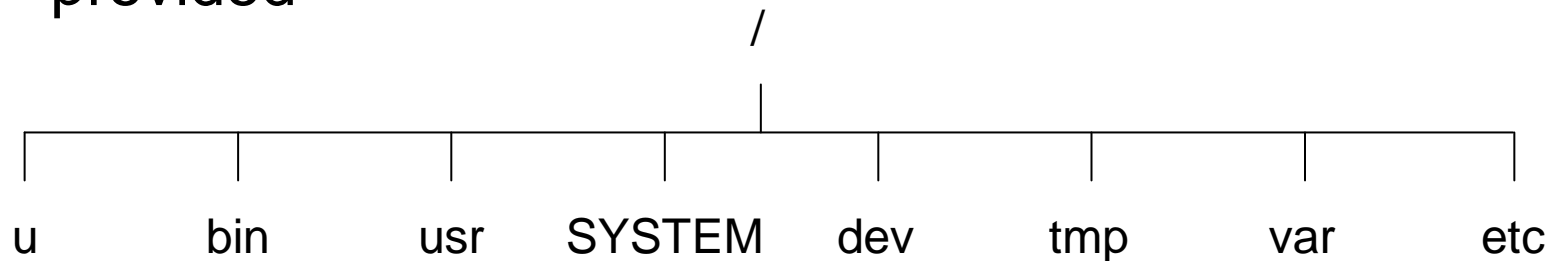
# Agenda

- Single System z/OS UNIX
- Multiple Systems with DASD sharing
- The z/OS UNIX Shared File System Environment
- Multiple Systems in a shared file system environment

# Single System z/OS UNIX

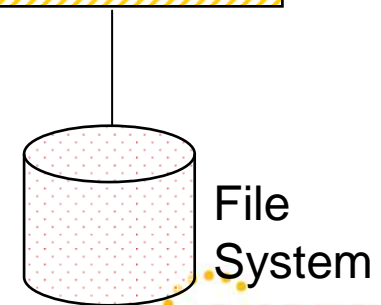
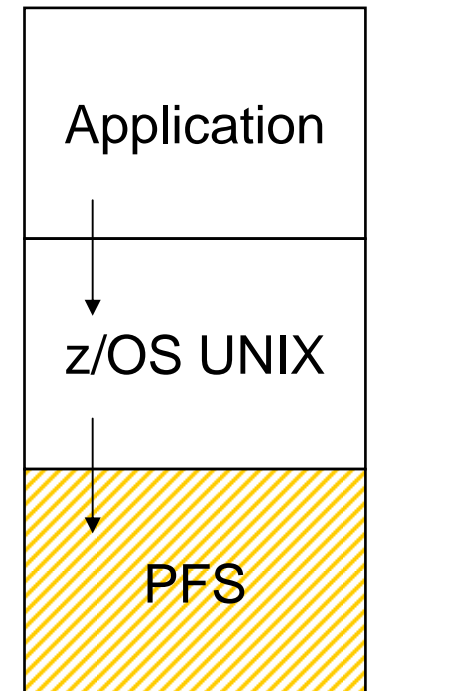
# Introduction to z/OS UNIX

- z/OS UNIX is an element of z/OS that provides a UNIX environment consisting of:
  - an Application Programming Interface (API), and
  - an interactive shell interface
  - TSO/E commands for file system management
  - ISPF shell environment (ishell)
- As a part of z/OS UNIX, a hierarchical file system is provided



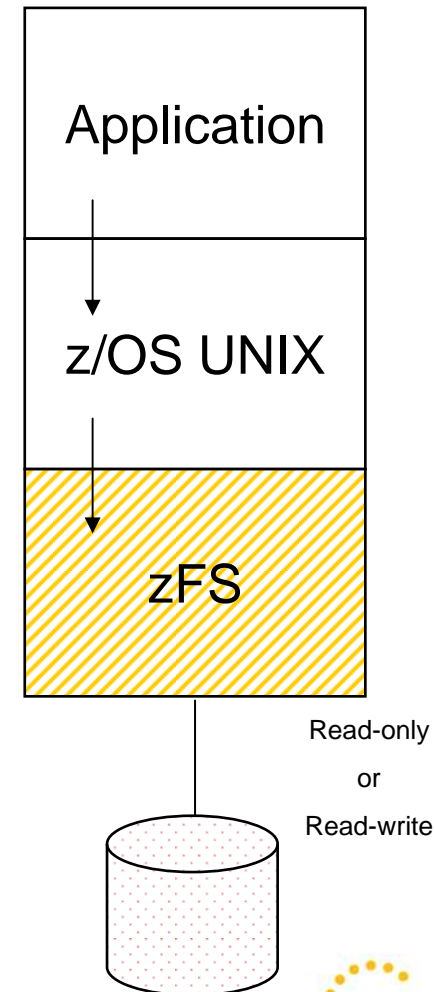
# z/OS file system components

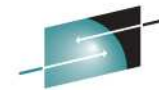
- z/OS UNIX System Services (z/OS UNIX)
  - Provides the z/OS UNIX APIs and the interface between applications or users to z/OS physical file systems
- z/OS Physical File Systems (PFS)
  - Provides the underlying code to access z/OS file systems (examples are zFS, HFS, TFS, AUTOMNT, NFS client)
- z/OS file systems
  - The actual DASD data sets that contain the z/OS UNIX files and directories



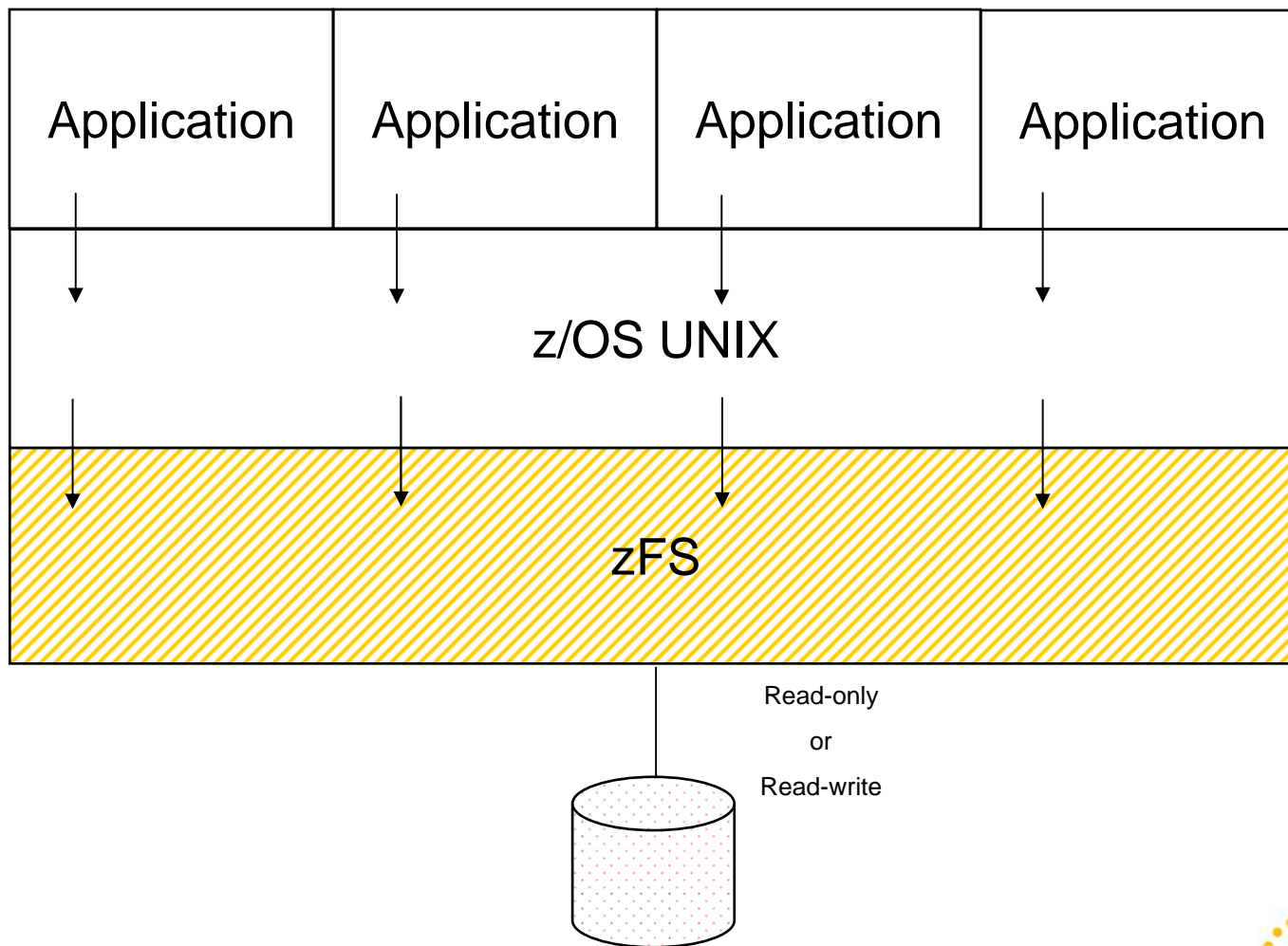
# Introduction to z/OS UNIX ...

- In z/OS, applications run in their own address space
- When an application invokes a z/OS UNIX API, control is passed into the z/OS UNIX address space
- If the z/OS UNIX API is a file request, application control is passed into the z/OS UNIX address space and then from there into the Physical File System (PFS) address space
- The PFS address space (in this case, zFS) handles reading and writing data to the actual disk and presents the data as a hierarchical file system
- All users on the system can access the same data in the file system (based on their authorization to the data)





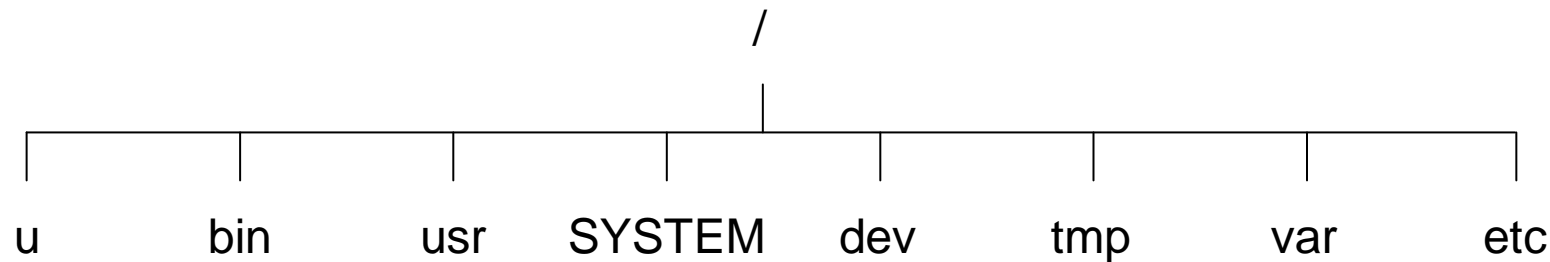
# Introduction to z/OS UNIX ...





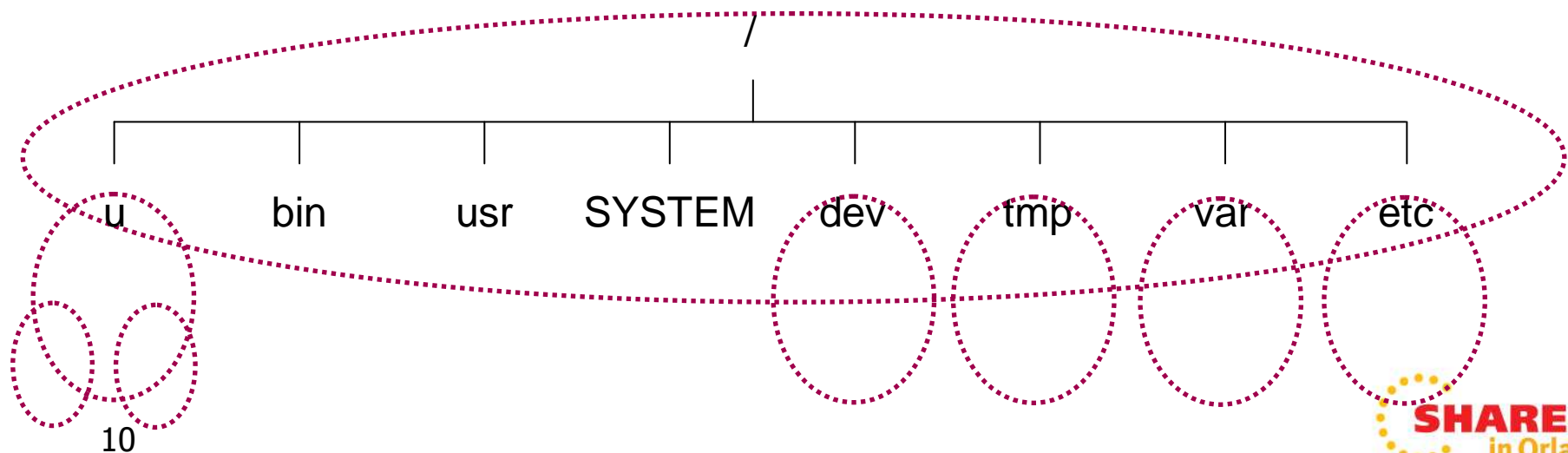
# The z/OS UNIX file system

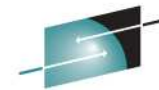
- This is how an end user views the z/OS UNIX hierarchical file system
- It looks like files and directories organized in a hierarchical tree



## The z/OS UNIX file system ...

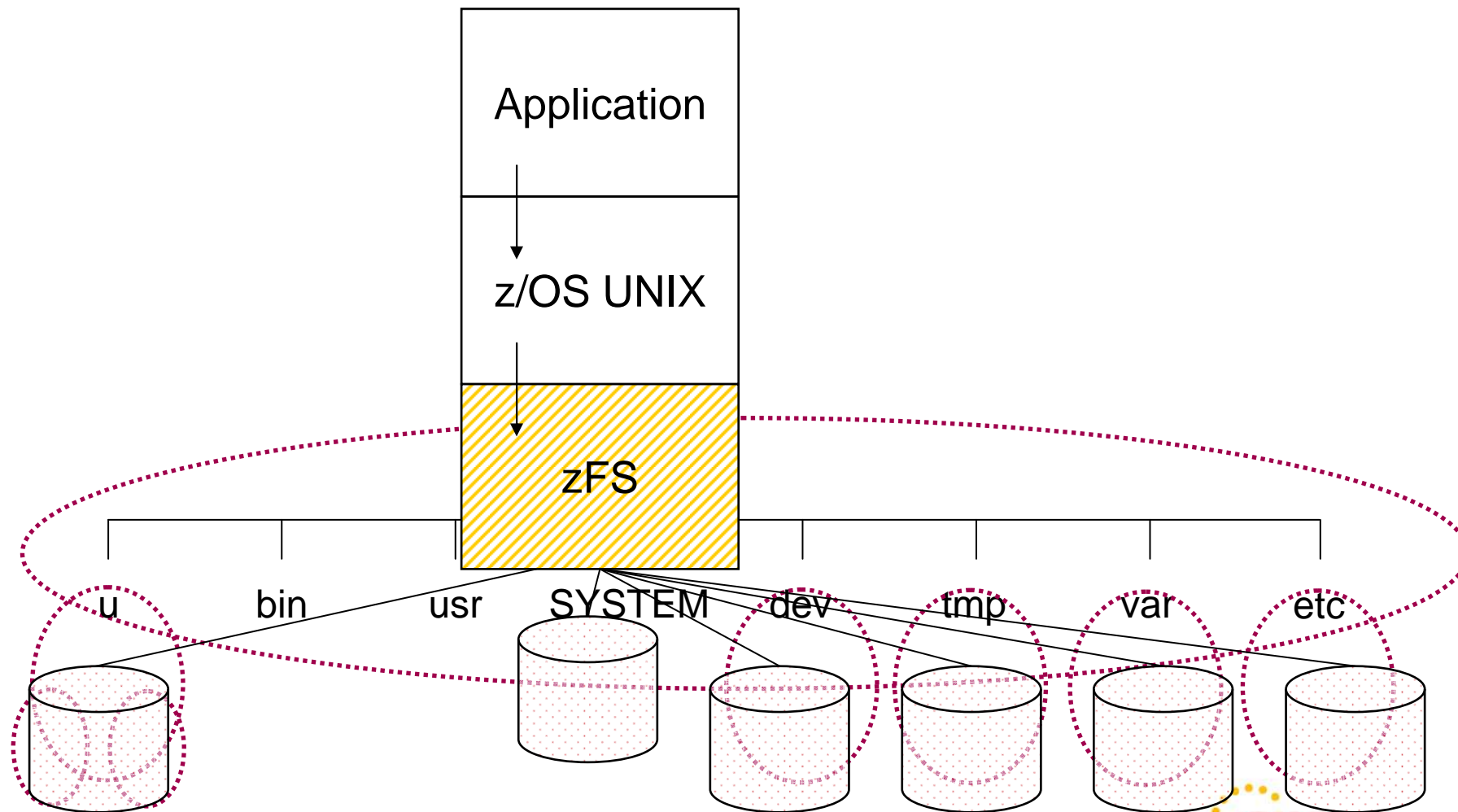
- The system programmer has to know that the hierarchical file system actually consists of many individually mounted file systems
- Each file system is (usually) a z/OS data set
- The hierarchical file system consists of a root file system with other file systems mounted on directories in the root file system





SHARE  
Technology - Connections - Results

# The z/OS UNIX file system ...



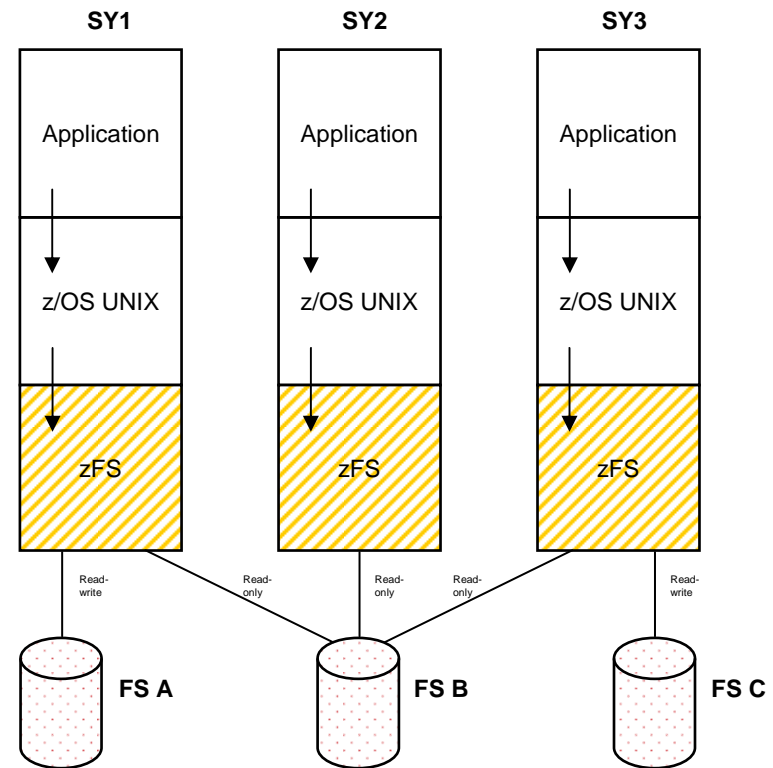
## The z/OS UNIX file system ...

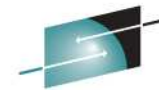
- The z/OS UNIX file system is configured by specifications in the BPXPRMxx Parmlib members
  - BPXPRMxx contains
    - FILESYSTYPE statements to initiate PFSEs
    - Mount statements for root and file systems below
  - BPXPRMxx list is specified in IEASYS00 as OMVS=(xx,yy,...)
  - When a file system is mounted, zFS allocates and opens the data set

# Multiple Systems with DASD sharing

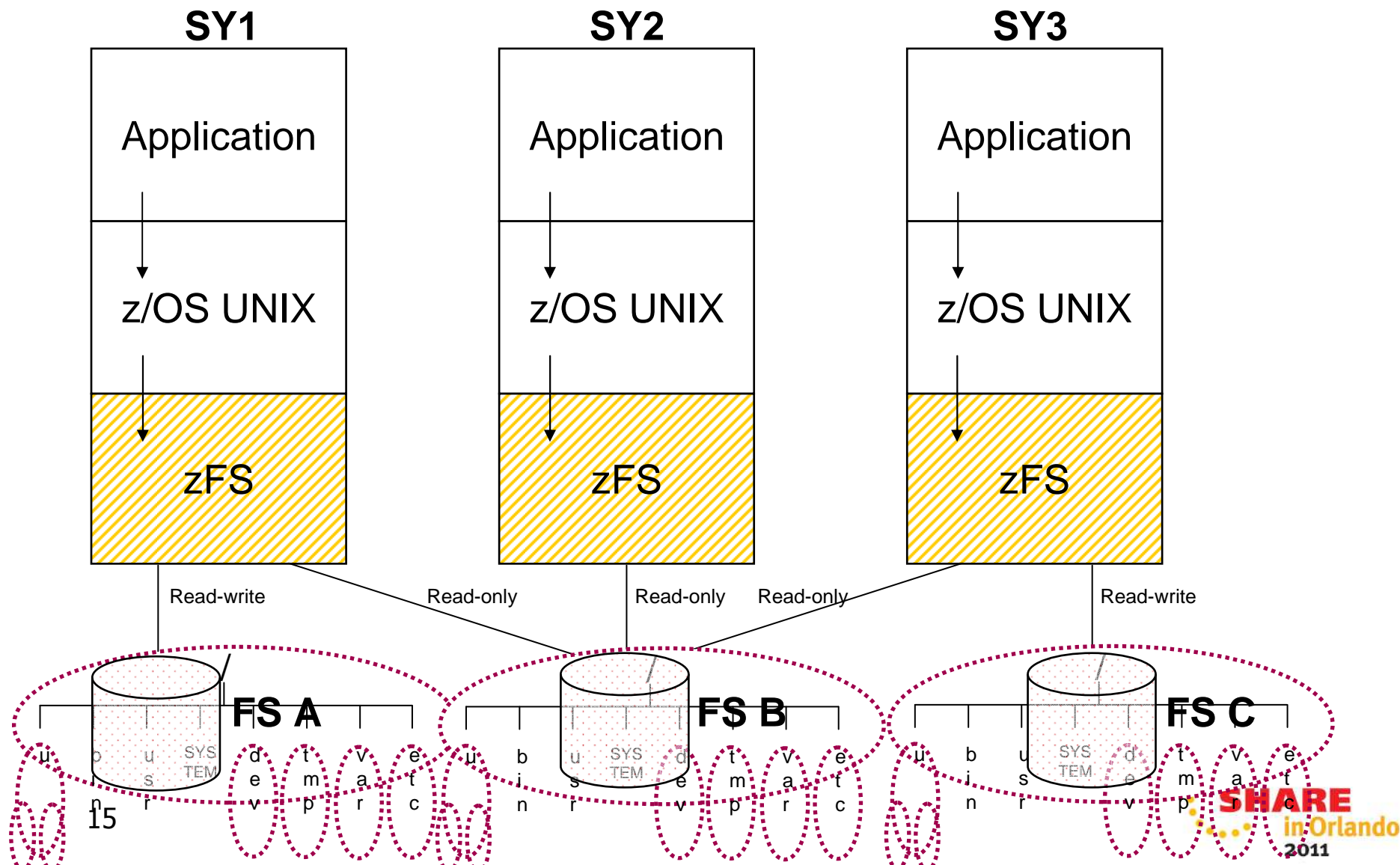
# Multiple systems with DASD sharing

- When you are logged on to a particular system (for example, SY1), you can only access a file system that is mounted on that system
- A file system can be read-only mounted on all systems, so it can be accessible from all systems
- A file system can only be read-write mounted on one system, so it can only be accessed from the one system it is mounted on
- To update (or read from) a read-write mounted file system you must log on to the system that it is mounted on





# Multiple systems with DASD sharing ...

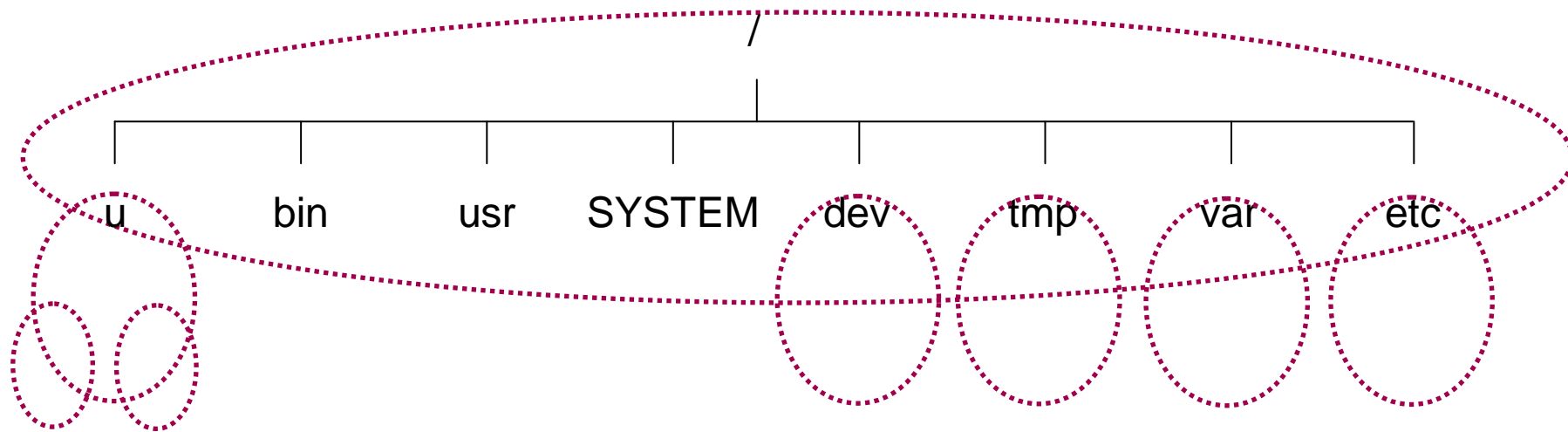


# The Shared File System Environment



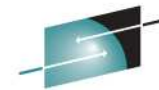
# The z/OS UNIX file system

- Again, here is what the z/OS UNIX file system looks like to the end user (and showing the mounted file systems)

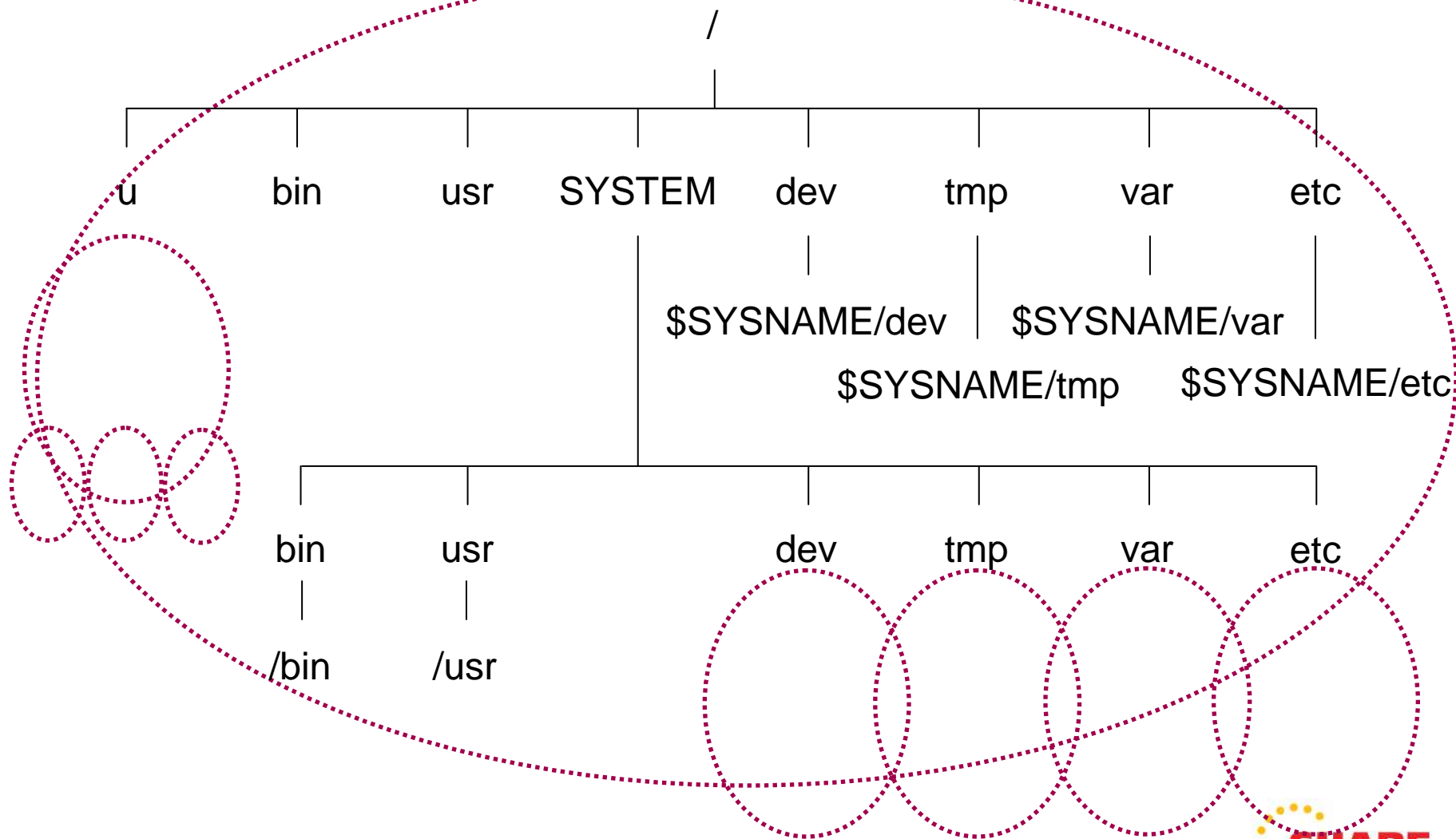


## The z/OS UNIX file system ...

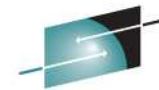
- The z/OS hierarchical file system is actually a bit more involved than the previous slide shows
- The sysplex shared file system environment needs to support multiple concurrent z/OS releases and even multiple concurrent service levels for different LPARs in a single file system hierarchy
- We also want to support system specific file systems
- The sysplex shared file system environment uses special mount points and symbolic links with system symbols to provide this
- Even a single system uses symbolic links to allow easy transition to the shared file system environment
- But, the end user view of the z/OS UNIX hierarchical file system does not change whether they are in a sysplex environment or not



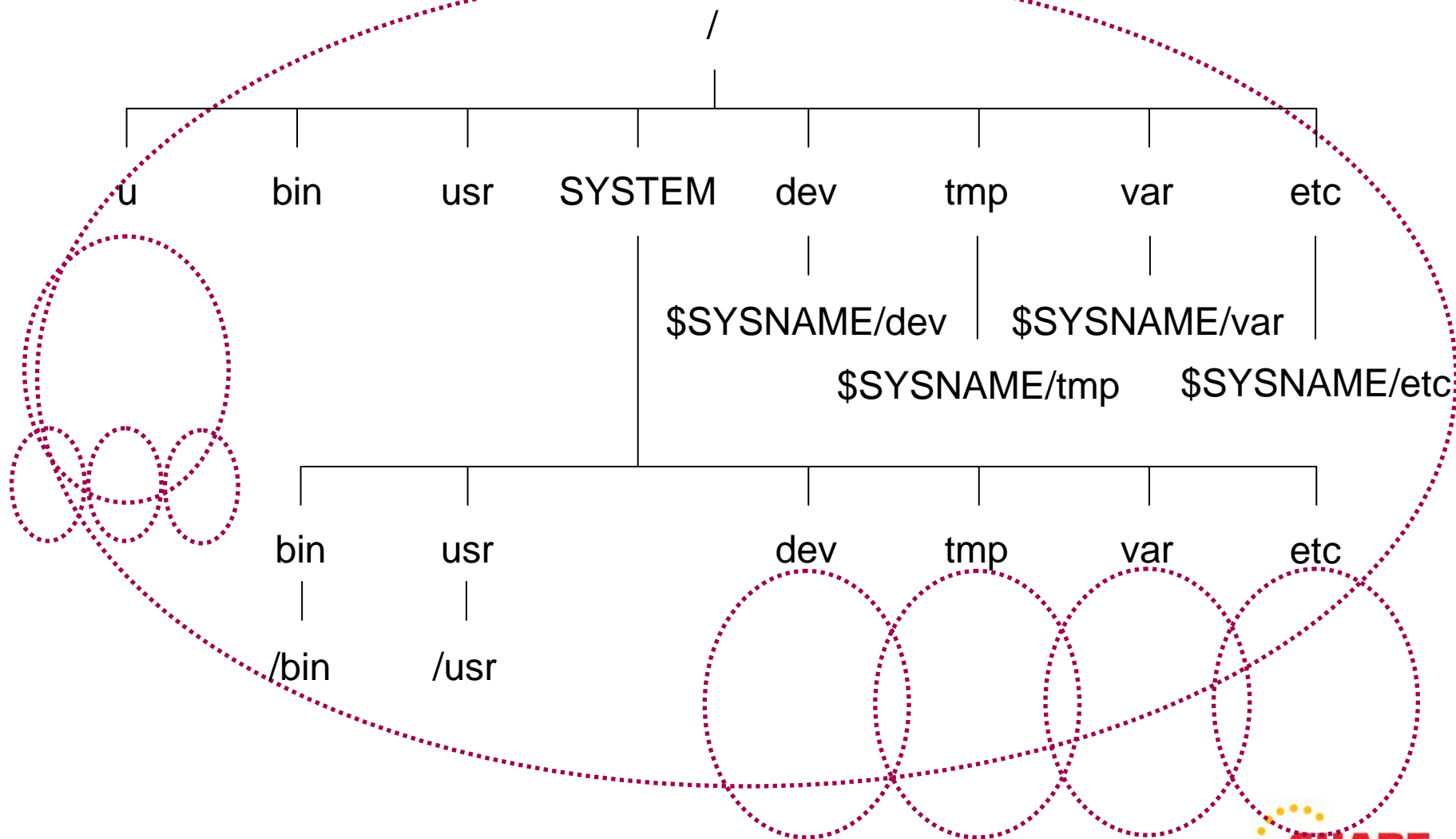
# The z/OS UNIX file system ...



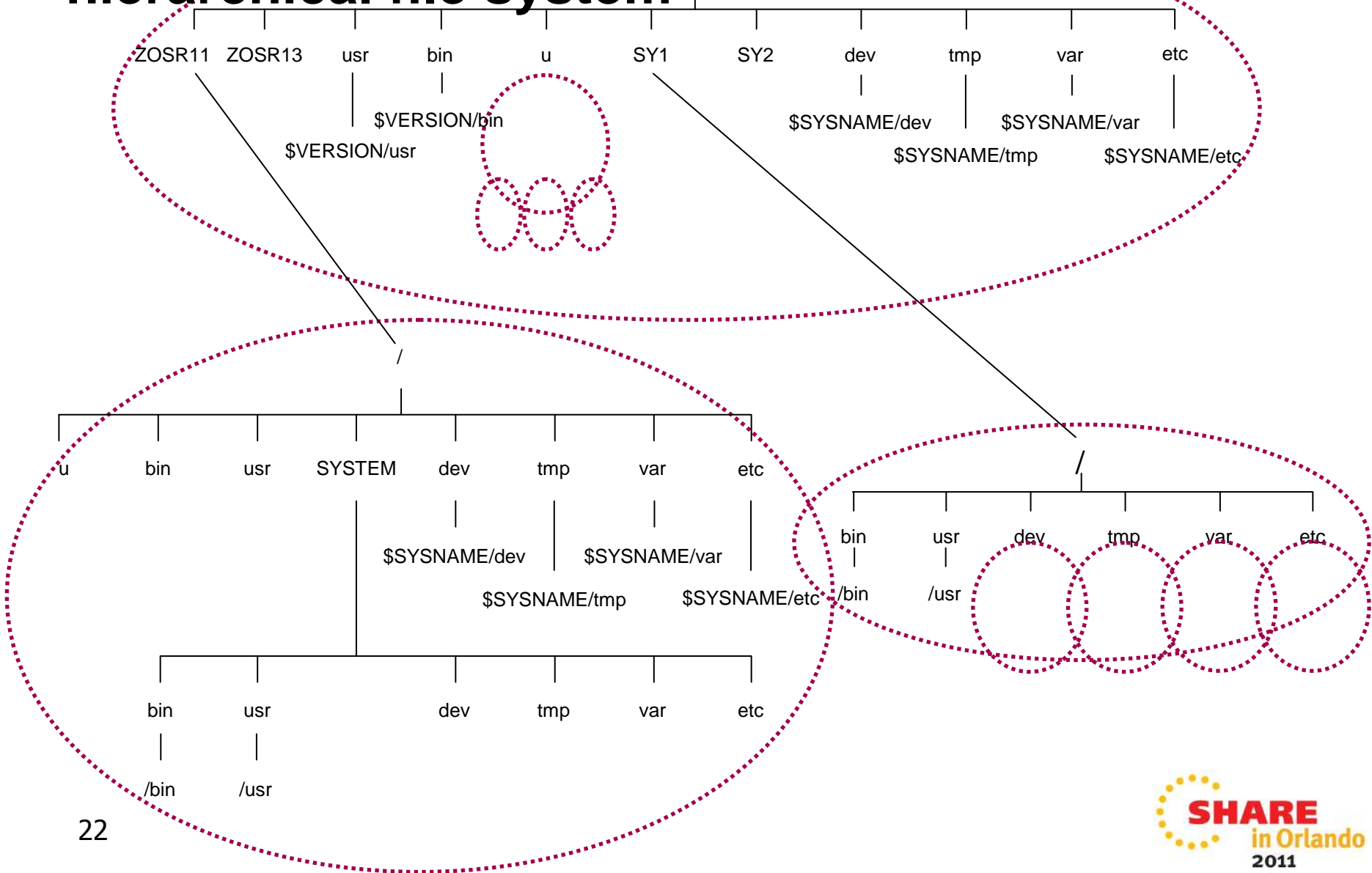
# Multiple Systems in a Shared File System Environment



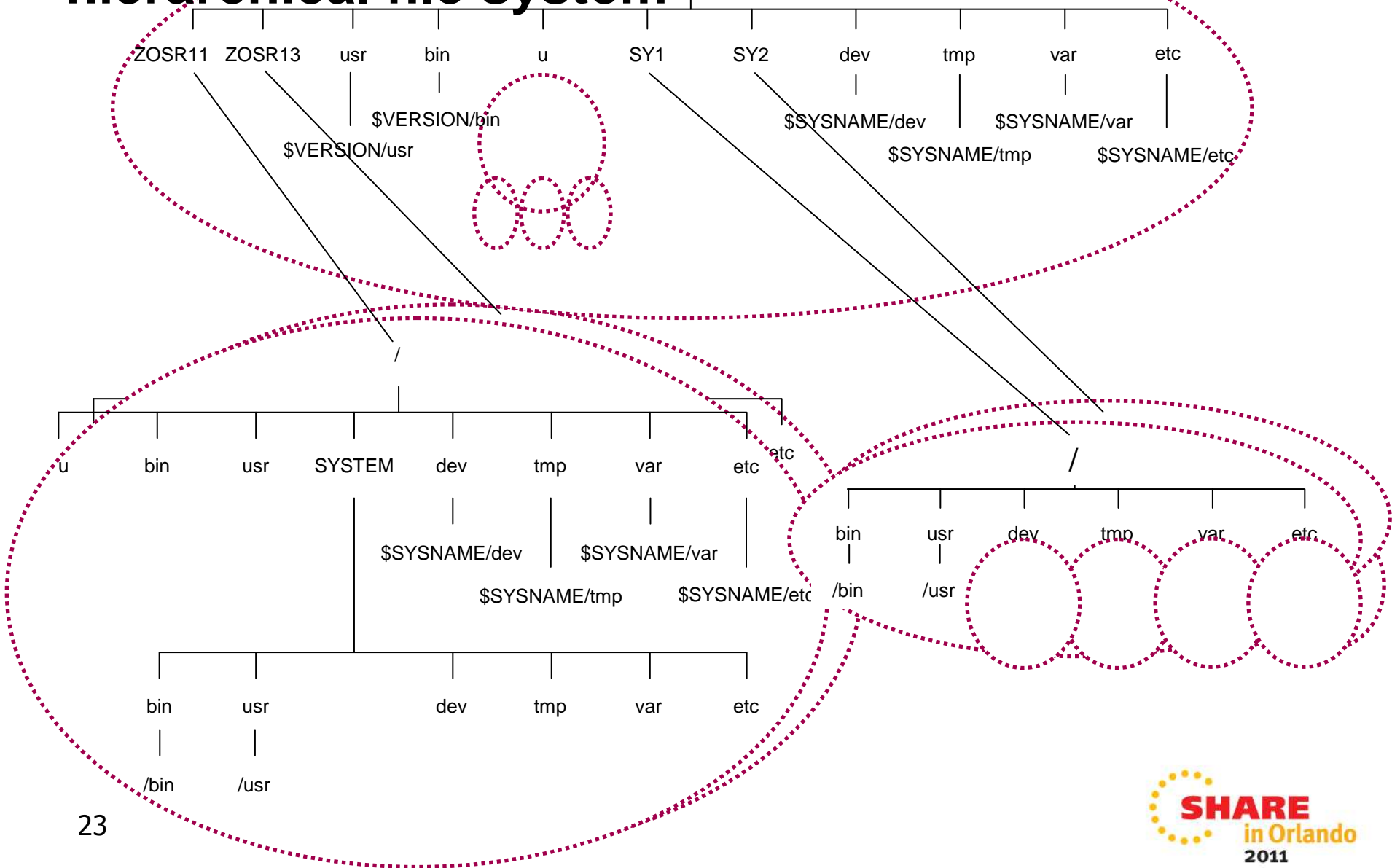
# The z/OS UNIX file system ...



# The z/OS UNIX sysplex shared file system hierarchical file system



# The z/OS UNIX sysplex shared file system hierarchical file system



# The z/OS UNIX file system in a shared file system environment



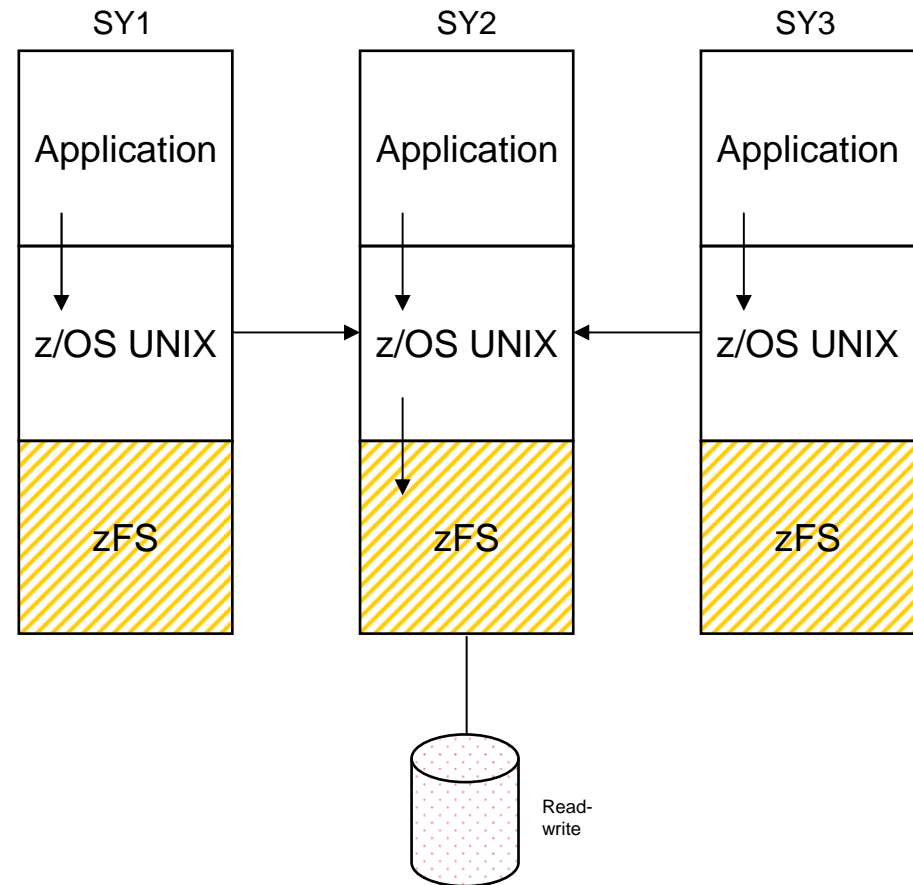
- The z/OS UNIX file system is configured by specifications in the BPXPRMxx Parmlib members
  - xx list is specified in IEASYS00 as OMVS=(xx,yy)
  - BPXPRMxx contains
    - FILESYSTYPE statements to initiate PFSes
    - Mount statements for root and lower file systems (At IPL, if a file system is already mounted, this is accepted silently)
    - SYSPLEX(YES) specifies shared file system environment
    - In this case a z/OS UNIX CDS (Couple Data Set) is required
    - VERSION('ZOSR13') specifies the value of \$VERSION (When SYSPLEX(YES) is specified, you must specify VERSION)
  - IEASYMxx contains
    - SYSDEF SYSNAME(SY1) specifies the value of \$SYSNAME
    - If the content of the symbolic link begins with \$SYSNAME and SYSPLEX is specified NO, then \$SYSNAME is replaced with /SYSTEM when the symbolic link is resolved.
  - When a file system is mounted, zFS allocates and opens the data set



# The z/OS UNIX shared file system environment



- In a parallel sysplex environment, z/OS UNIX can provide access to the entire file system hierarchy for all users from all LPARs in the sysplex
- This environment is called a **shared file system environment**
- z/OS UNIX provides this support by forwarding file requests from other LPARs (SY1 or SY3) to the LPAR designated as the z/OS UNIX file system owning LPAR (SY2 in this case)
- When you mount a file system, an owning LPAR is designated and the file system is mounted and available to all LPARs in the shared file system environment



# Benefits of the shared file system environment



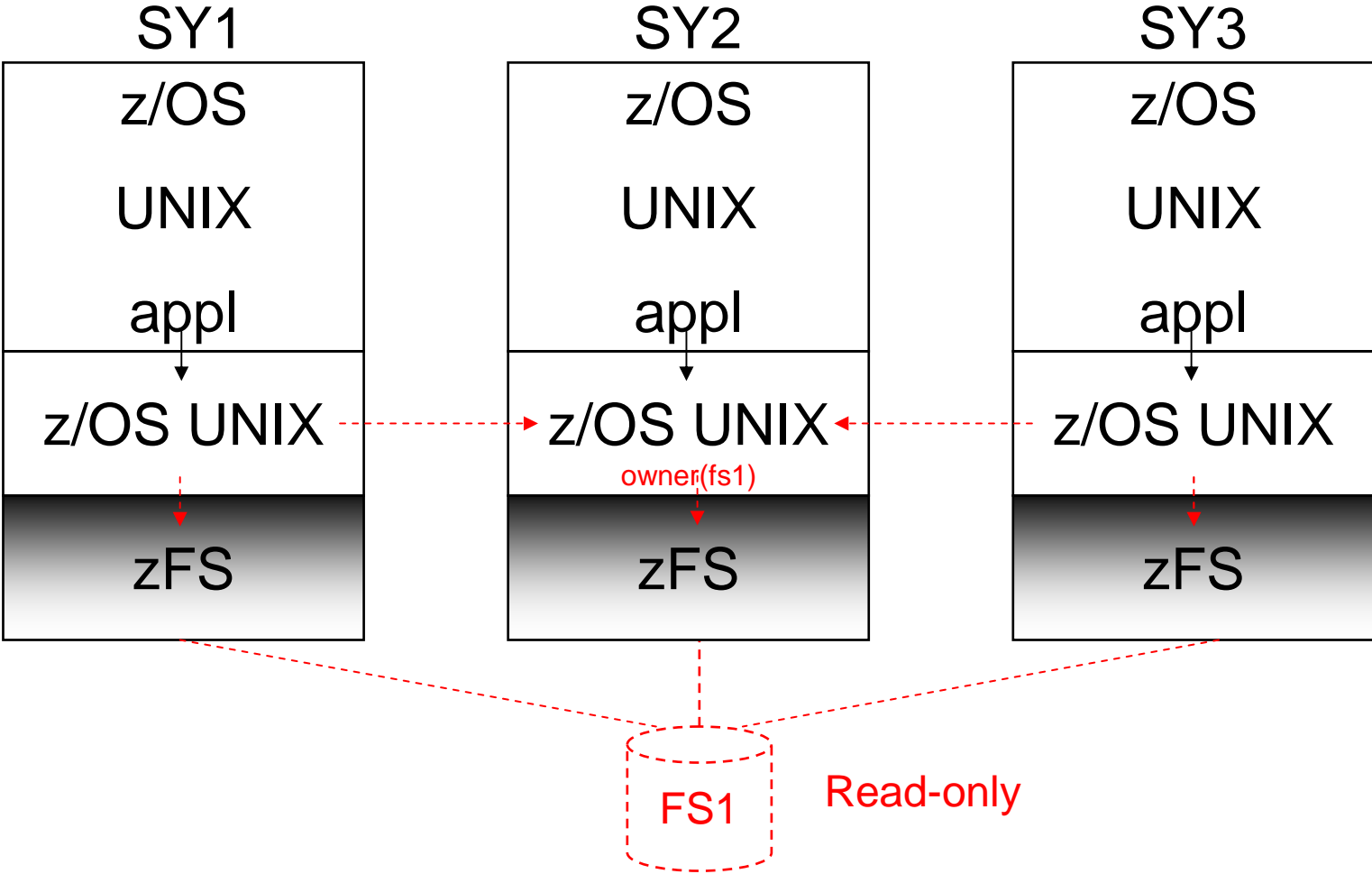
- **System independence**  
All z/OS UNIX data can be accessed from any system in the sysplex
- **Availability**  
If a system is shutdown or if it abnormally goes down, file system ownership movement occurs automatically and file systems remain accessible from the other systems (although temporary failures may be visible to applications during abnormal shutdown)
- **Flexibility**  
General users and applications automatically access the correct (system specific and/or version/release) file systems while administrators can access any file system
- **Transparency**  
Users and applications do not need to change to run in a shared file system environment (except to possibly handle and recover from the temporary failures)

# Publications

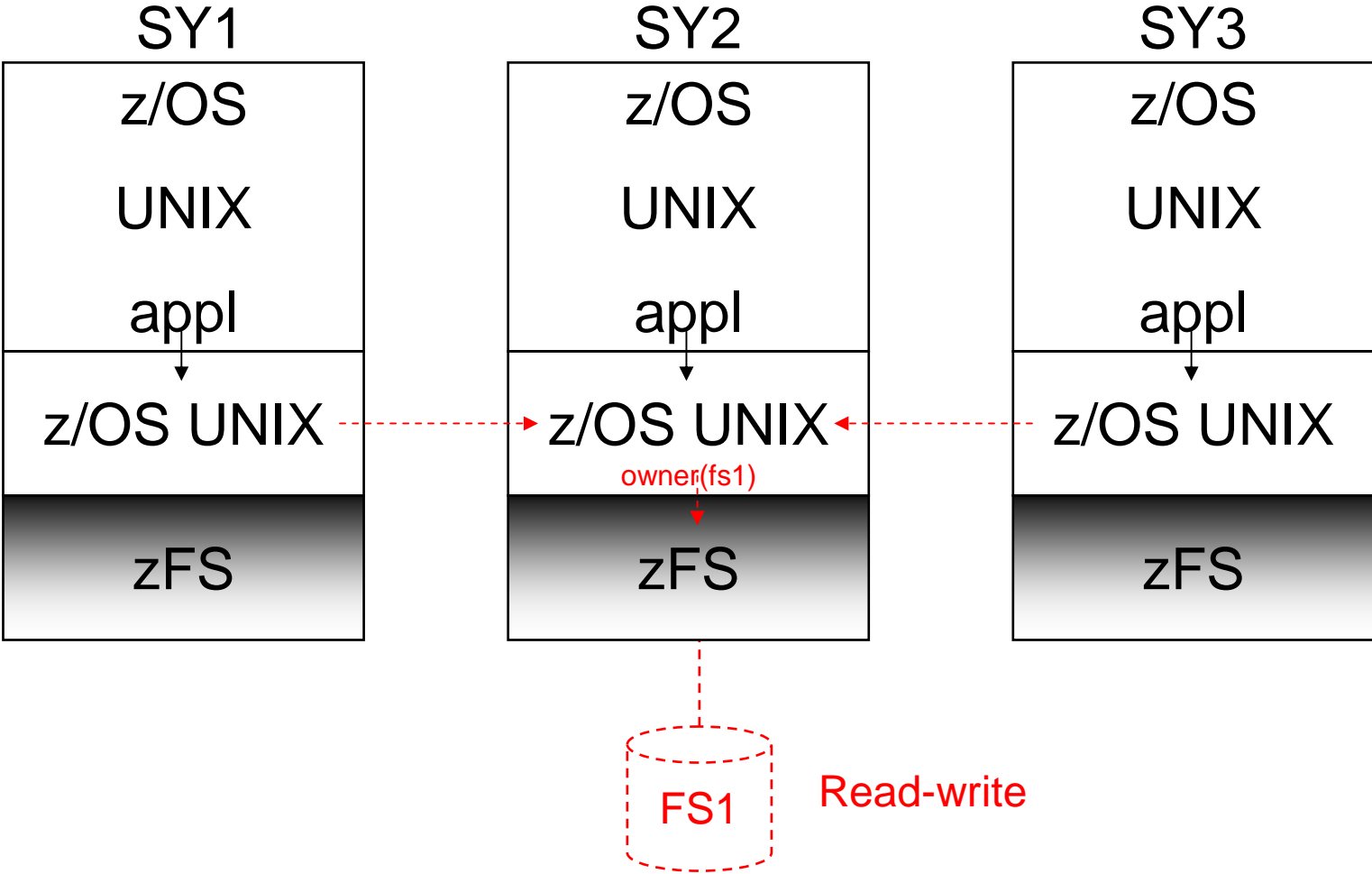
- z/OS UNIX System Services Planning (GA22-7800)  
General Administration of z/OS UNIX file systems
- z/OS UNIX Command Reference (SA22-7802)  
confighfs command for HFS
- z/OS MVS System Messages Volume 9 (IGF-IWM) (SA22-7639)  
IGWxxxxt messages for HFS
- z/OS UNIX System Services Messages and Codes (SA22-7807)  
z/OS UNIX return codes, z/OS UNIX reason codes, X'5Bxxxxrrr' reason codes for HFS
- z/OS Distributed File Service zSeries File System Administration (SC24-5989) – **was refreshed in April 2010**  
zFS Concepts and zfsadm command for zFS
- z/OS Distributed File Services Messages and Codes (SC24-5917)  
IOEZxxxxt messages and X'EFxxxxrrr' reason codes for zFS
- z/OS Distributed File Service zSeries File System Implementation (SG24-6580)
  - Redbook available (updated February 2010 to include z/OS V1R11)
  - <http://www.redbooks.ibm.com/abstracts/sg246580.html?Open>
- z/OS Version 1 Release 8 Implementation (SG24-7265)
  - Redbook available (contains zFS updates for z/OS V1R8)
  - <http://www.redbooks.ibm.com/abstracts/sg247265.html?Open>
- z/OS DFSMS™ Access Method Services for Catalogs (SC26-7394)  
IDCAMS utility
- z/OS DFSMS™ Storage Administration Reference (SC26-7402)  
ADRDSSU utility for backup

# Backup

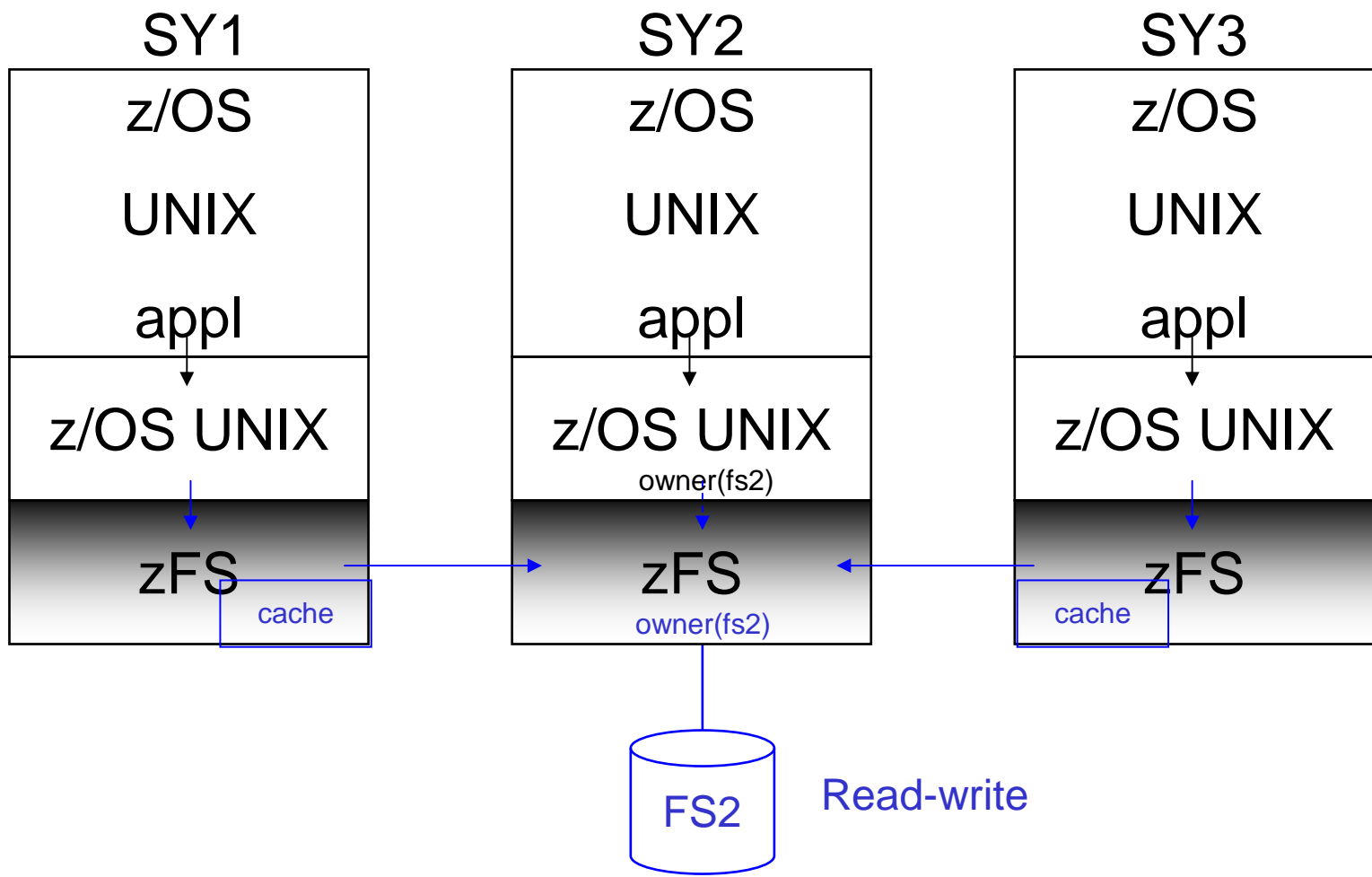
# Read-only mounted file system in a shared file system environment



# Read-write mounted file system in a shared file system environment



# Sysplex-aware read-write mounted file system in a shared file system environment (R11 zFS)

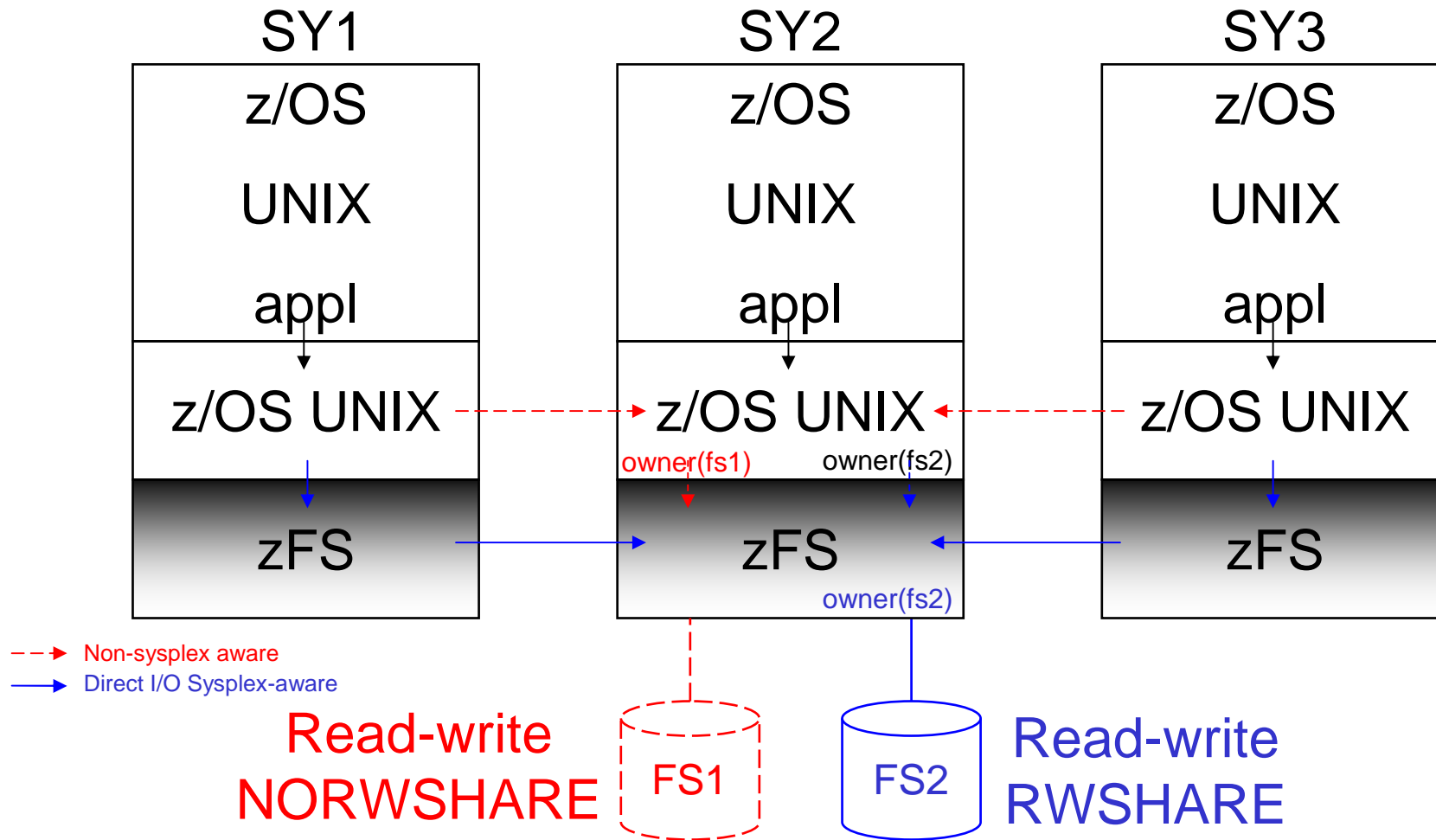


## Sysplex-aware read-write mounted file system in a shared file system environment

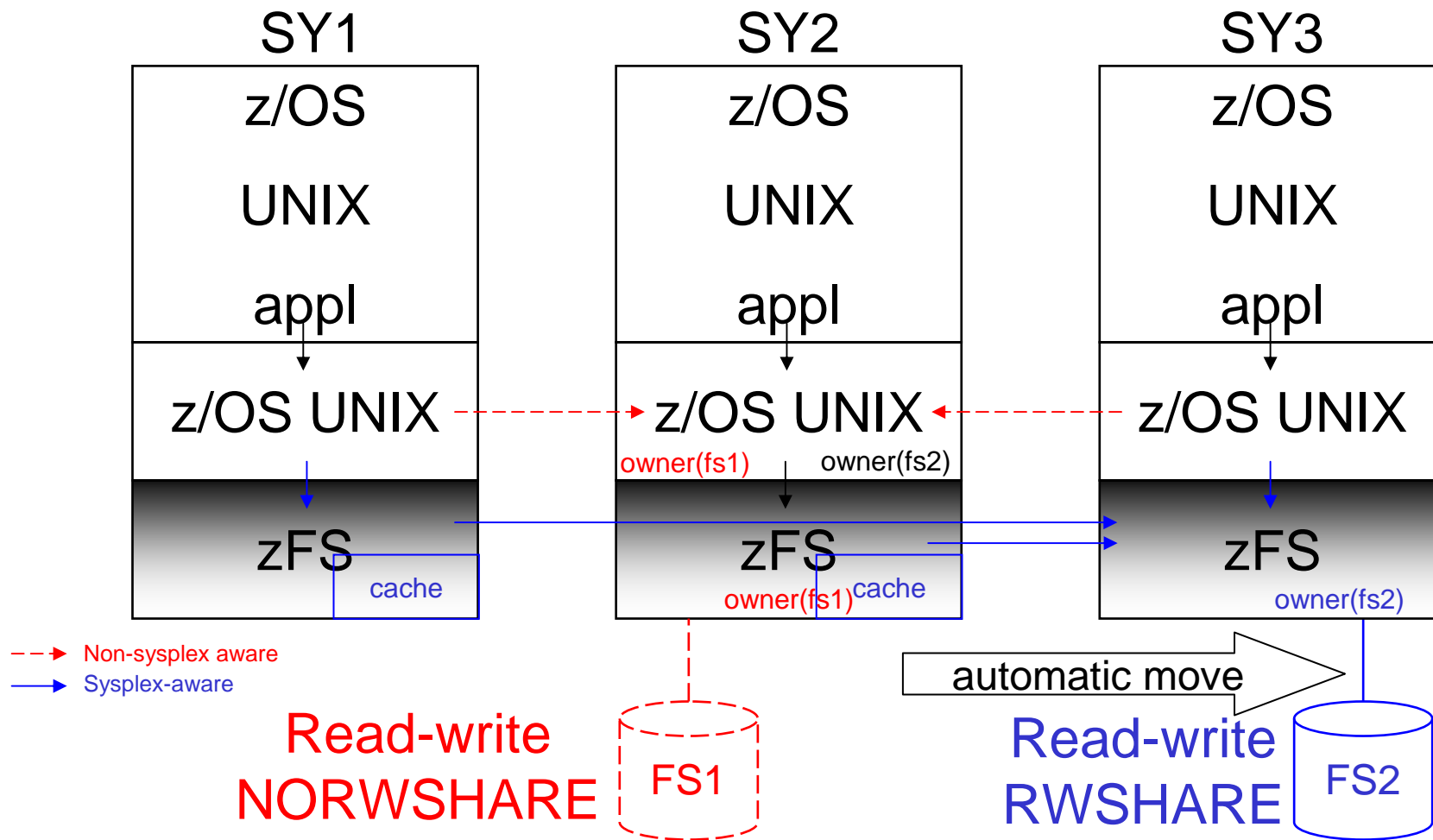
- Local mount  
This means that the mount request was sent by z/OS UNIX to the local zFS. The local zFS has the data set open. A **df -v** command for the file system would show Client=N. A file system is always locally mounted on the z/OS UNIX owning system.
- Global mount  
This means that the file system is available from any system in the shared file system environment. The file system is locally mounted on the z/OS UNIX owning system and may or may not be locally mounted on the other systems. A file system mounted in a shared file system environment is always globally mounted.



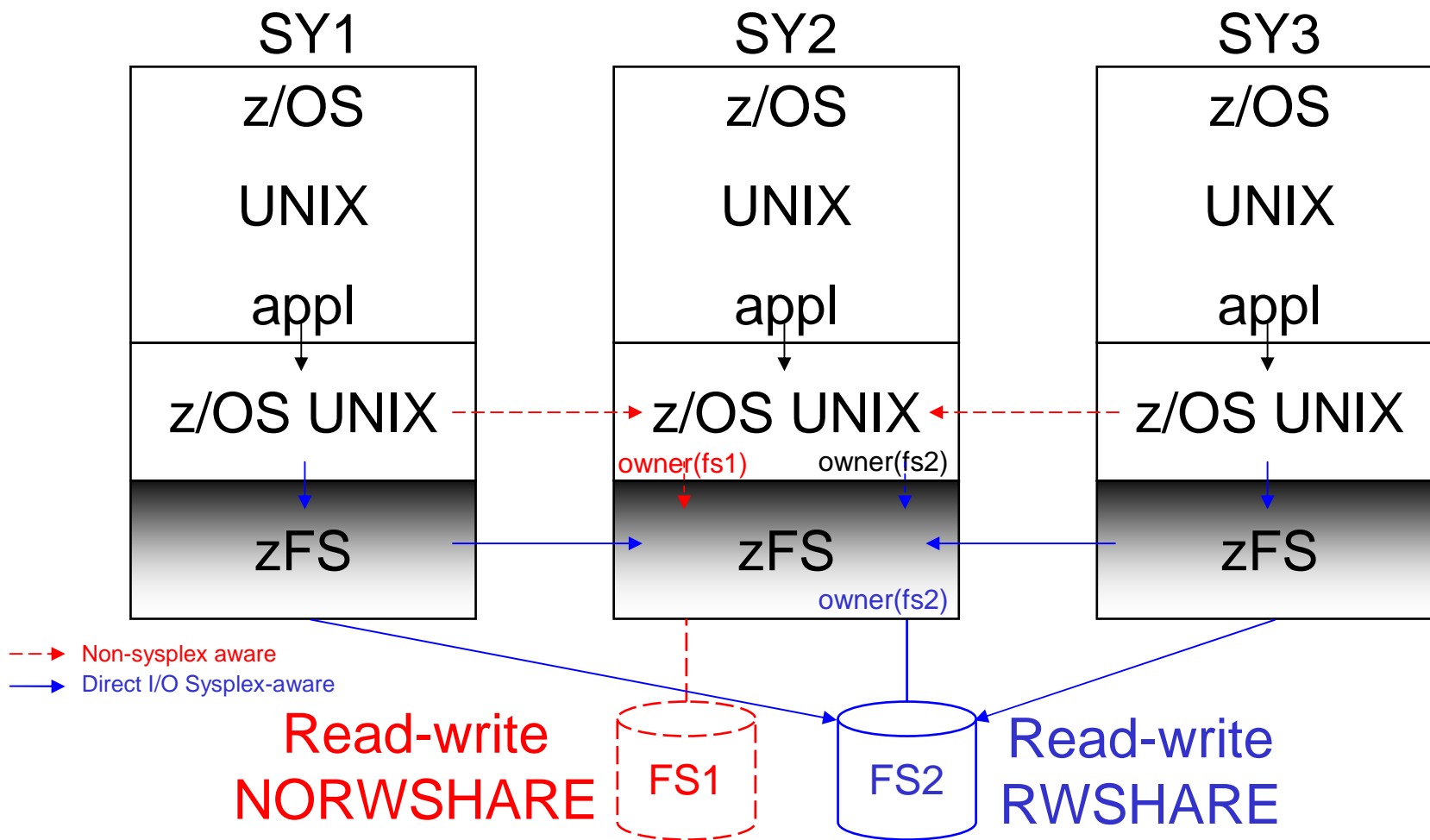
# Sysplex-aware and non-sysplex aware read-write mounted file systems in a shared file system environment (R11 zFS APAR OA29619)



# zFS ownership movement of a sysplex-aware read-write mounted file system in a shared file system environment



# Direct I/O sysplex-aware read-write mounted file system in a shared file system environment



## Adding a new system or version root

- In general, the contents of the sysplex root should only change when you need a new version root or system-specific root file system directory for your shared file system configuration. When a system is IPLed (initialized), the mount processing for the sysplex root file system will include defining the appropriate \$SYSNAME or \$VERSION directory in the sysplex root file system if the sysplex root is mounted as read/write. Assuming that you have the sysplex root file system mounted as read-only, the procedure to use to create a new version root or system-specific file system directory is as follows:
  - Remount the sysplex root file system to read-write
  - IPL the new system
  - Remount the sysplex root file system to read-only