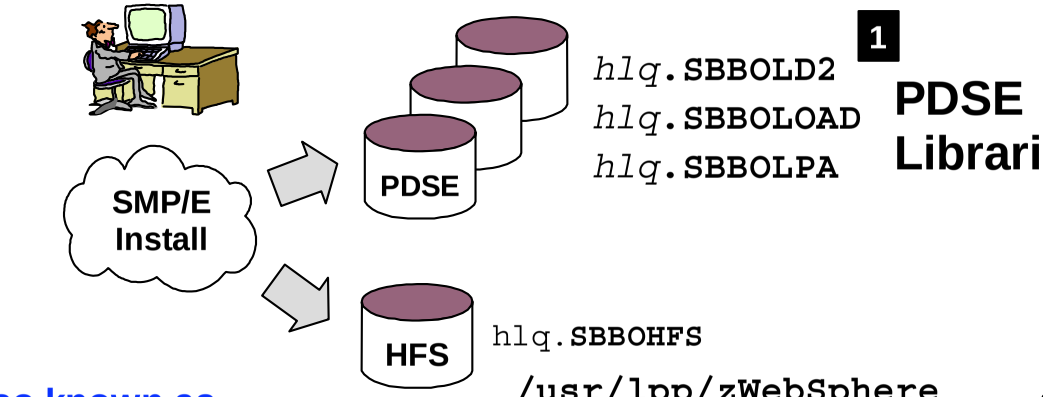
WebSphere Application Server

WebSphere for z/OS Version 5.1
Test, Production, Maintenance

IBM Americas Advanced Technical Support -- Washington Systems Center
Gaithersburg, MD, USA

Two "Install Environments"

When WebSphere Application Server for z/OS V5.1 is installed, SMP/E creates two "install environments":



Also known as

- "Code Base"
- "Install Base"
- "SMP/E data sets"

Full set of WebSphere for z/OS Version 5 SMP/E target data sets

```
WAS510.WAS.SBBOCLIB
WAS510.WAS.SBBODSD
WAS510.WAS.SBBOEXEC
WAS510.WAS.SBBOHFS
WAS510.WAS.SBBOJCL
WAS510.WAS.SBBOLD2
WAS510.WAS.SBBOLOAD
WAS510.WAS.SBBOLPA
WAS510.WAS.SBBOMAC
WAS510.WAS.SBBOMIG
WAS510.WAS.SBBOMJPN
WAS510.WAS.SBBOMLIB
WAS510.WAS.SBBOMSG
WAS510.WAS.SBBOPJPN
WAS510.WAS.SBBOPLIB
WAS510.WAS.SBBOSLB2
WAS510.WAS.SBBOSLIB
```

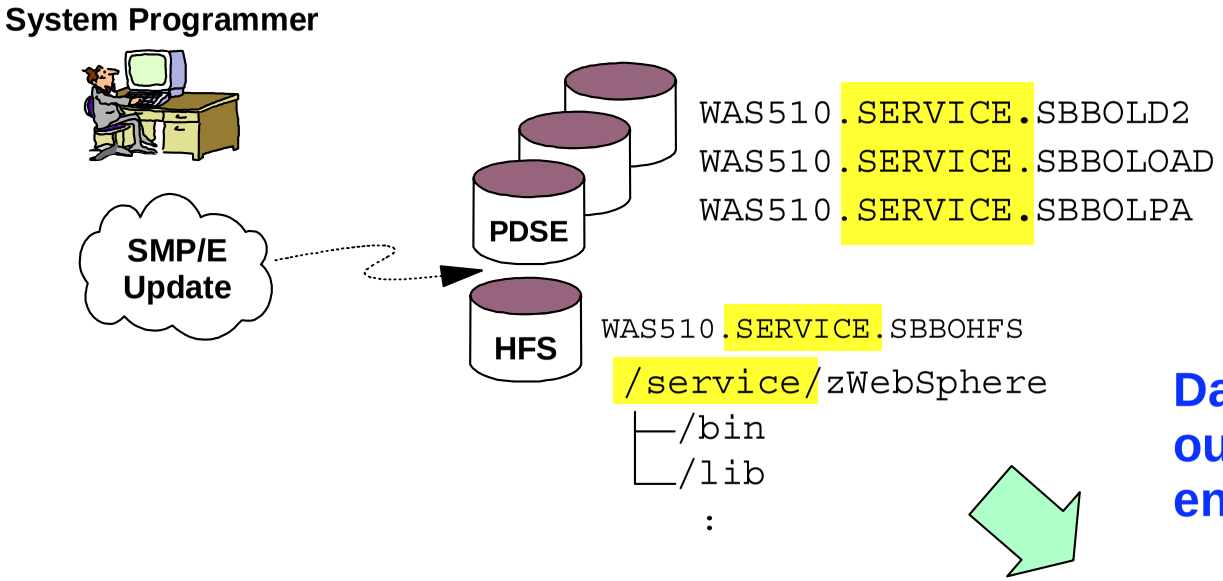
Key Points:

- WebSphere comes in two pieces -- typical MVS libraries and an HFS file system
- The "Install HFS" is different from the "Config HFS" you created for your cell

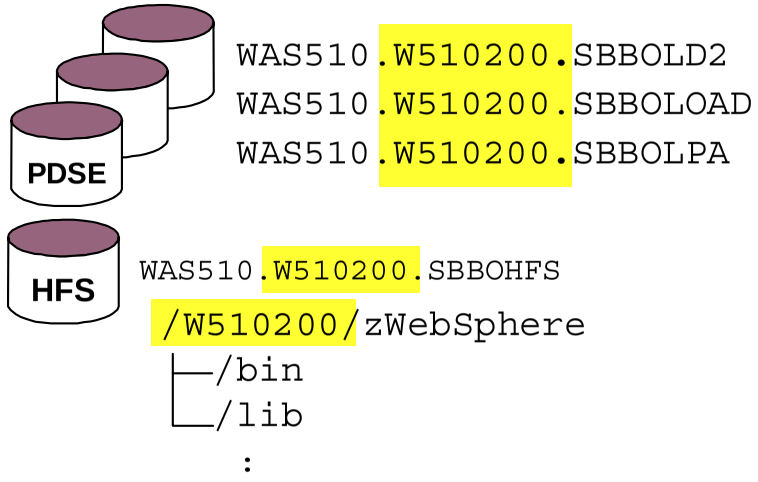
The fundamental issue we'll address is the relationship between your configuration and these "install environments"

Applying Maintenance to Service Env.

Maintenance is typically applied against "service" environment, then copied out to the test or production environment:



Data sets copied out to QA or Test environment



When we speak of "applying maintenance" to a test or production environment, we're speaking of bringing in a *copy* of the data sets from the "service" environment.

Multiple Versions Possible

There's nothing about WebSphere for z/OS Version 5.1 that prevents you from having multiple version levels on the same MVS image:

Production Version	New Maintenance Level Being Tested
WAS510.WAS.SBBOCLIB	WAS510.W510200.SBBOCLIB
WAS510.WAS.SBBODSD	WAS510.W510200.SBBODSD
WAS510.WAS.SBBOEXEC	WAS510.W510200.SBBOEXEC
WAS510.WAS.SBBOHFS	WAS510.W510200.SBBOHFS
WAS510.WAS.SBBOJCL	WAS510.W510200.SBBOJCL
WAS510.WAS.SBBOLD2	WAS510.W510200.SBBOLD2
WAS510.WAS.SBBOLOAD	WAS510.W510200.SBBOLOAD
WAS510.WAS.SBBOLPA	WAS510.W510200.SBBOLPA
WAS510.WAS.SBBOMAC	WAS510.W510200.SBBOMAC
WAS510.WAS.SBBOMIG	WAS510.W510200.SBBOMIG
WAS510.WAS.SBBOMJPN	WAS510.W510200.SBBOMJPN
WAS510.WAS.SBBOMLIB	WAS510.W510200.SBBOMLIB
WAS510.WAS.SBBOMSG	WAS510.W510200.SBBOMSG
WAS510.WAS.SBBOPJPN	WAS510.W510200.SBBOPJPN
WAS510.WAS.SBBOPLIB	WAS510.W510200.SBBOPLIB
WAS510.WAS.SBBOSLB2	WAS510.W510200.SBBOSLB2
WAS510.WAS.SBBOSLIB	WAS510.W510200.SBBOSLIB

Data sets have different qualifiers

Not quite as simple as that ... do have considerations related to:

- **Which copy -- if any -- resides in LPA/LNKLST**
- **How to "toggle" a configuration from one code base to another**
- **How to non-disruptively introduce new level into a production environment**

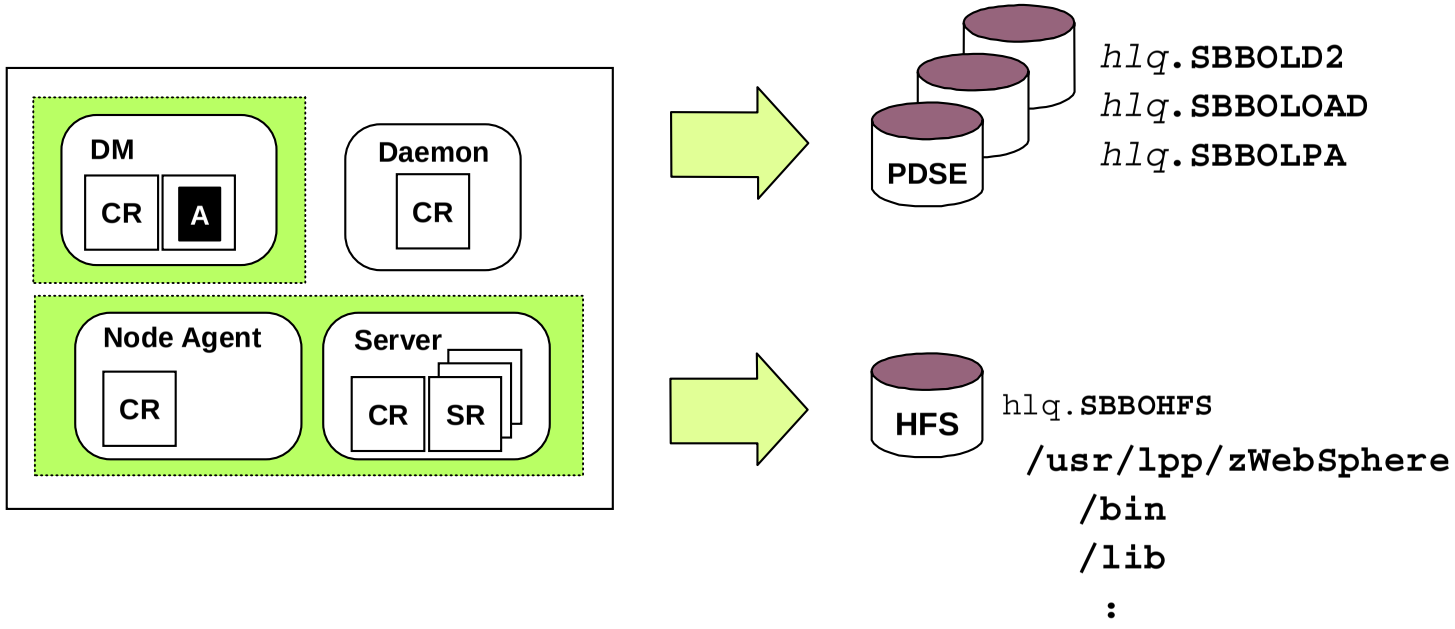
That's what this presentation is all about ...

Basics: How Config Accesses Code

Load libraries accessed in one of two ways:

- LPA/LNKLST
- STEPLIB

Fairly familiar stuff, but there are a few twists. We'll spend some time understanding this first.



HFS files are accessed via symbolic links in the configuration HFS

Some new things here ... we'll focus quite a bit of time on this area of the puzzle.

STEPLIB or LPA/LNKLST?

Do you remember when you first built your node with the ISPF panels? Whether STEPLIB is present or not is determined in "System Locations 1 of 2" panel:

```
Full Names of Data Sets

  PROCLIB:   SYS1.PROCLIB
  PARMLIB:   SYS1.PARMLIB
  SYSEXEC:   SYSS.WSC.SYSEXEC

  SCEERUN.:   CEE.SCEERUN
  SCEERUN2:   CEE.SCEERUN2
  SBBOLOAD:   WAS510.WAS.SBBOLOAD
  SBBOLD2.:   WAS510.WAS.SBBOLD2
  SBBOMIG.:   WAS510.WAS.SBBOMIG
  SBBOLPA.:   WAS510.WAS.SBBOLPA
  SGSKLOAD:   SYS1.GSK.SGSKLOAD
```

```
In LNKLST or
LPA (Y or N)?
     Y
     Y
     Y
     Y
     Y
     Y
     Y
```

Question asked is whether libraries in LPA/LNKLST ...

Y -- *no* STEPLIB built

N -- STEPLIB created

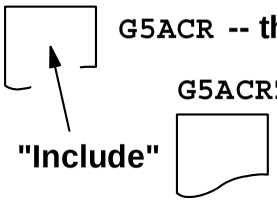
Wish to change a configuration to a new level of code?

Libraries in LPA/LNKLST?	***Refresh*** **contents of LPA/LNKLST**
STEPLIBs already present?	***Change*** **STEPLIB statements**
STEPLIBs not present but you want them?	***Add*** **STEPLIB statements**

Where can STEPLIB statements be found?

STEPLIB -- *Two* Places

JCL procedures are in two parts ... "Z member" is where STEPLIB would be:



`G5ACR` -- the main body of the JCL

`G5ACRZ` -- the "Z member" associated with the proc.

```
//*
//* Output DDs
//*
//CEEDUMP   DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT    DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT  DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//*
//*Steplib Setup
//*
//STEPLIB   DD DISP=SHR,DSN=WAS510.WAS.SBBOLD2
//          DD DISP=SHR,DSN=WAS510.WAS.SBBOLOAD
//          DD DISP=SHR,DSN=WAS510.WAS.SBBOLPA
```

All **procs have two-part JCL:**

- **Application server controllers**
- **Application server servants**
- **Deployment Manager controller**
- **Deployment Manager servant**
- **Node Agent controllers**
- **Daemon**

STEPLIB may also be in shell script `setupCmdLine.sh` found in `/bin` directory

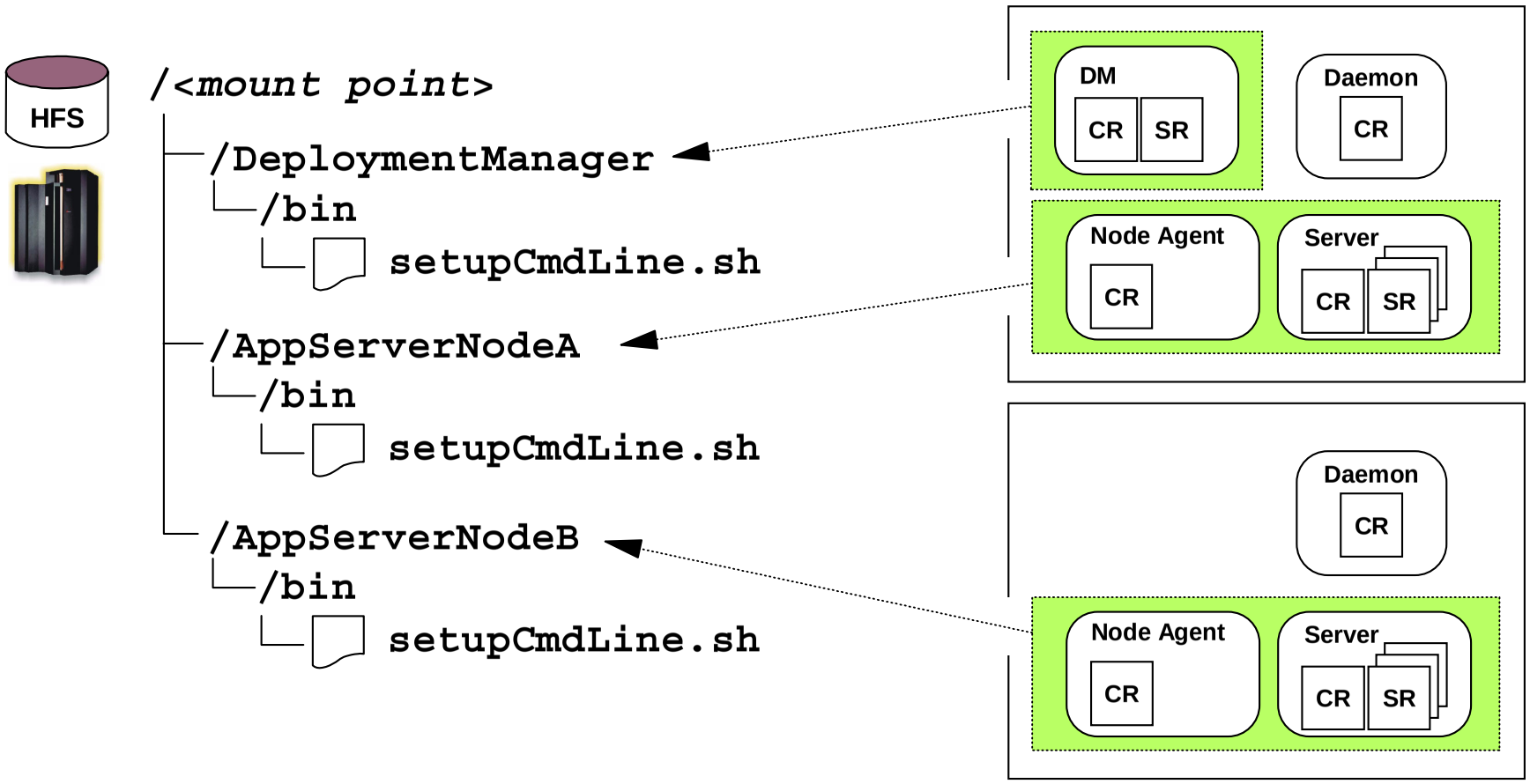
This establishes STEPLIB for all the shell scripts that are part of WAS.

```
 :
#
STEPLIB='WAS510.WAS.SBBOLD2':$STEPLIB
STEPLIB='WAS510.WAS.SBBOLOAD':$STEPLIB
STEPLIB='WAS510.WAS.SBBOLPA':$STEPLIB
export STEPLIB
 :
```

Careful! Multiple copies of this file ...

Multiple Copies of `setupCmdLine.sh`

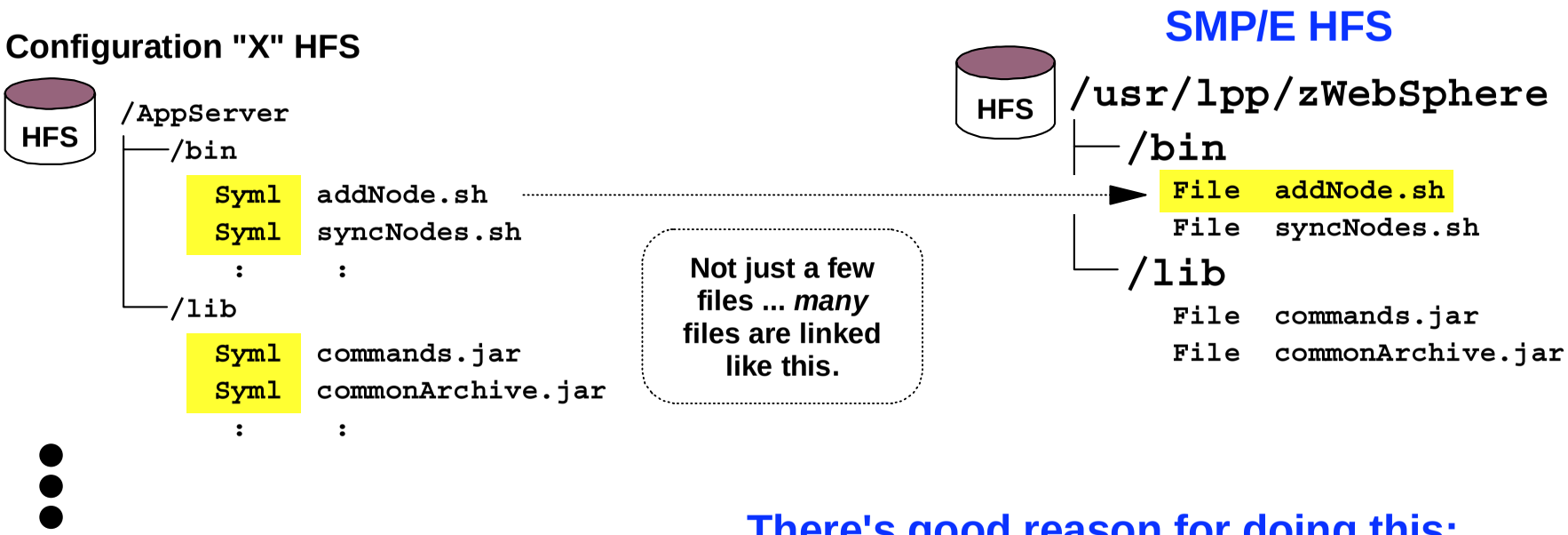
A multi-node configuration will have multiple copies of the `setupCmdLine.sh` shell script ... each node structure's `/bin` directory will contain a copy:



Key Message: any changes to STEPLIB -- either to add STEPLIB or to change STEPLIB -- involves changes to all "Z members" *and* all `setupCmdLine.sh` files.

Symlinks in the Configuration HFS

Many of the files in the Configuration HFS are really just symbolic links that point off to the actual file in the SMP/E (or "Install") HFS:



Configuration "Y" HFS

HFS
/AppServer
/bin
Syml addNode.sh
Syml syncNodes.sh
: :
/lib
Syml commands.jar
Syml commonArchive.jar
: :

There's good reason for doing this:

- **Eliminates unnecessary duplication of the same file in lots of different configuration HFS**
- **Gives SMP/E one place to update files rather than lots of different places**

Where's the pointer to the SMP/E HFS mount point specified? In the ISPF Dialogs ...

Contents of the Symbolic Links

Contents of the symbolic links in the Configuration HFS are determined by value you supply for "WebSphere SMP/E home directory" in panels:

```
System Locations (2 of 2)

   Specify the following for your customization, then press Enter
   to continue.

 Locations of HFS Resident Components

   WebSphere Application Server SMP/E home directory:
     /usr/lpp/zWebSphere/V5R1M0
   WebSphere JMS Client Java Feature SMP/E home directory:
     /usr/lpp/mqm/V5R3M1
   Java home directory:
     /usr/lpp/java2/J1.4
```

Whatever value you supply on panel is used to created all the symlinks in the node

ISHELL listing of a sample configuration HFS

```
   Type   Filename
_  Dir    .
_  Dir    ..
_  Syml   addNode.sh
_  Syml   admincons.pax
_  Syml   admincons.unpax.sh
_  Syml   applyPTF.sh
    :
```

```
Symbolic link contents:
/usr/lpp/zWebSphere/V5R1M0/bin/applyPTF.sh
```

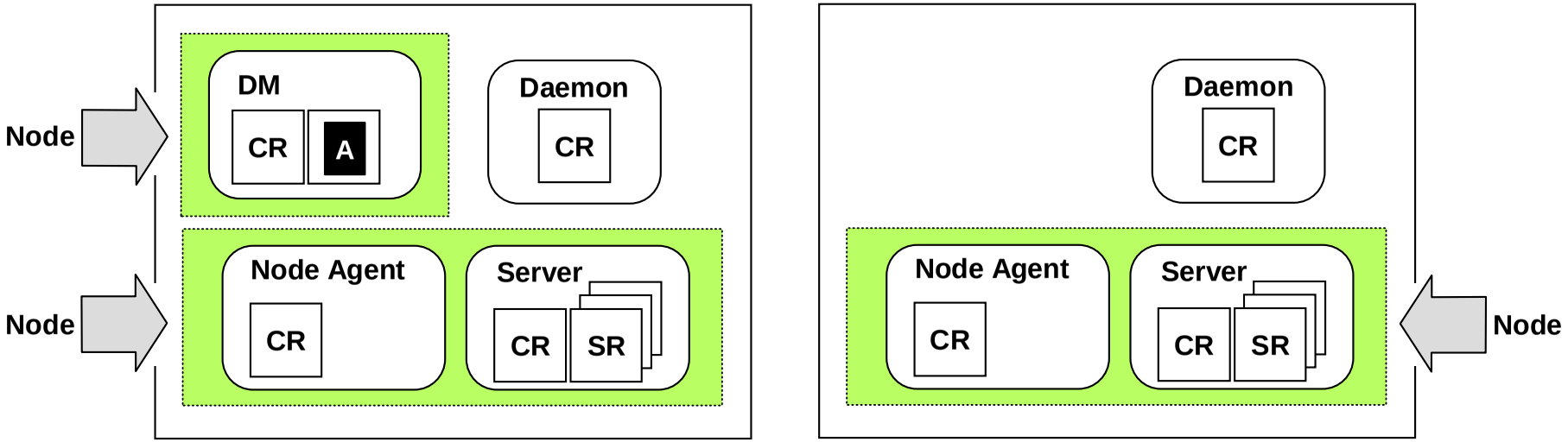
Once set ... they're set. Can't change them.

This has a profound impact on how different configurations are isolated from one another, and impacts how "non-disruptive" maintenance can be achieved.

Granularity of Symlinks: Node Level

WebSphere configurations are built by constructing nodes and then federating them into larger and larger cells:

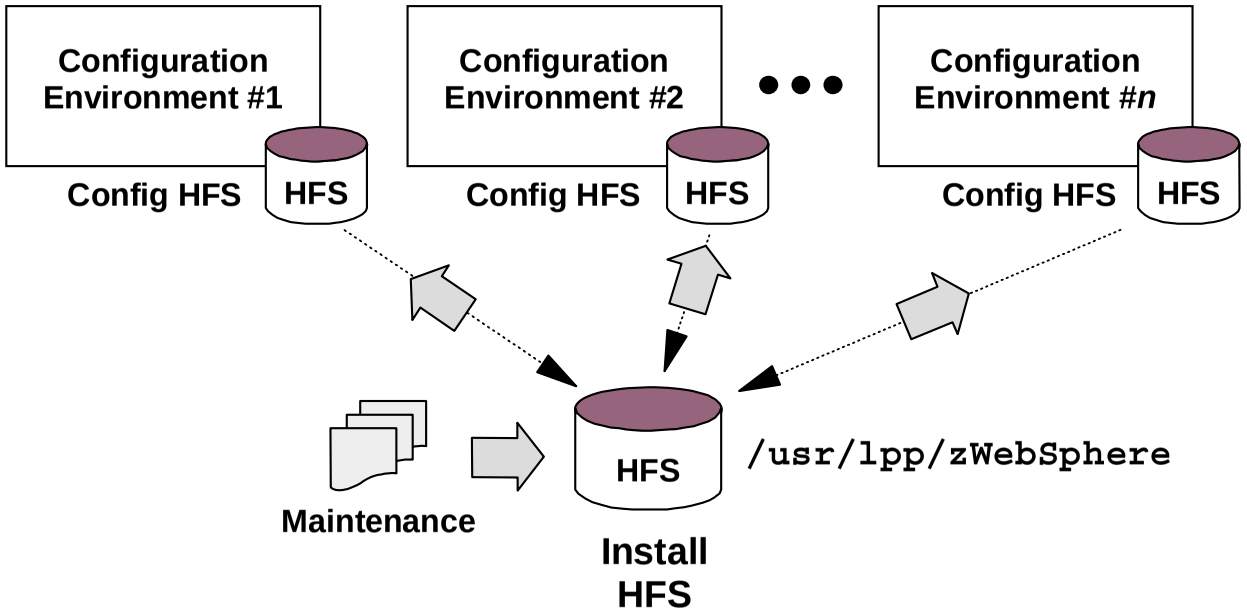
- **The symbolic links are created when the node is built**
- **Symbolic links are in the `/bin` and `/lib` directories ... *above* server level**
- **Therefore, symbolic links are no more granular than the node level**



Why is this important? Because it determines the granularity of isolation that can be achieve. And that leads directly into discussion of isolating test from production, and isolating node from node so maintenance can be "rolled" through a configuration.

Issue with Common SMP/E Mount Point

If you have a number of different configuration environments all pointing to the same "WebSphere SMP/E Home Directory," maintenance will apply to all:

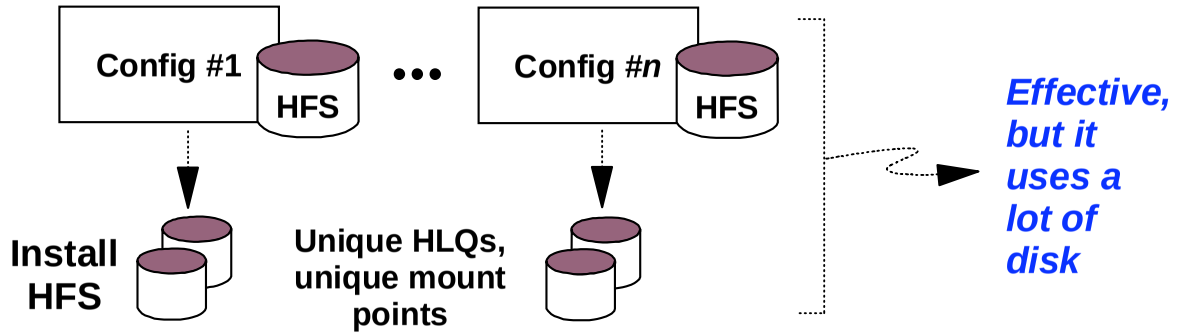


This may or may not be okay, depending on what you're trying to accomplish:

- **Multiple environments where you want the same code level? Okay.**
- **Test and Production environments? Not okay.**

Two ways to provide isolation between environments:

Provide separate install data sets for each:

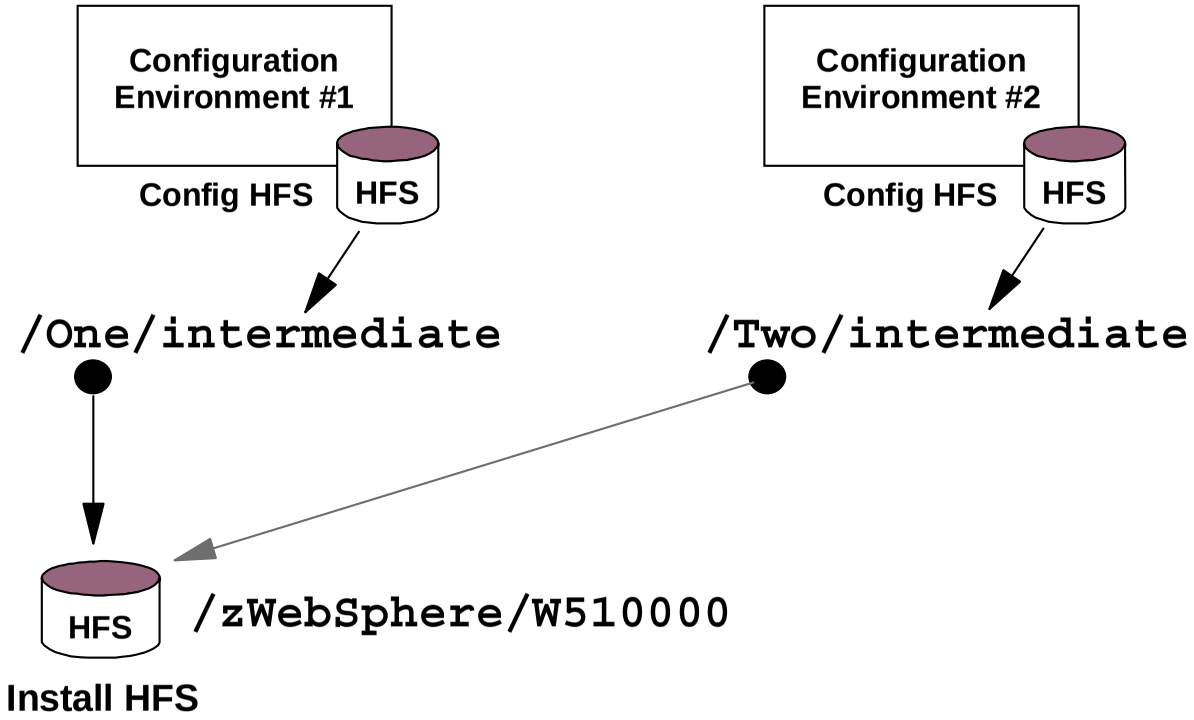


Effective, but it uses a lot of disk

Other method is to employ "intermediate symbolic links" ...

Intermediate Symbolic Links

Symbolic links may be "chained" ... this provides considerable flexibility:

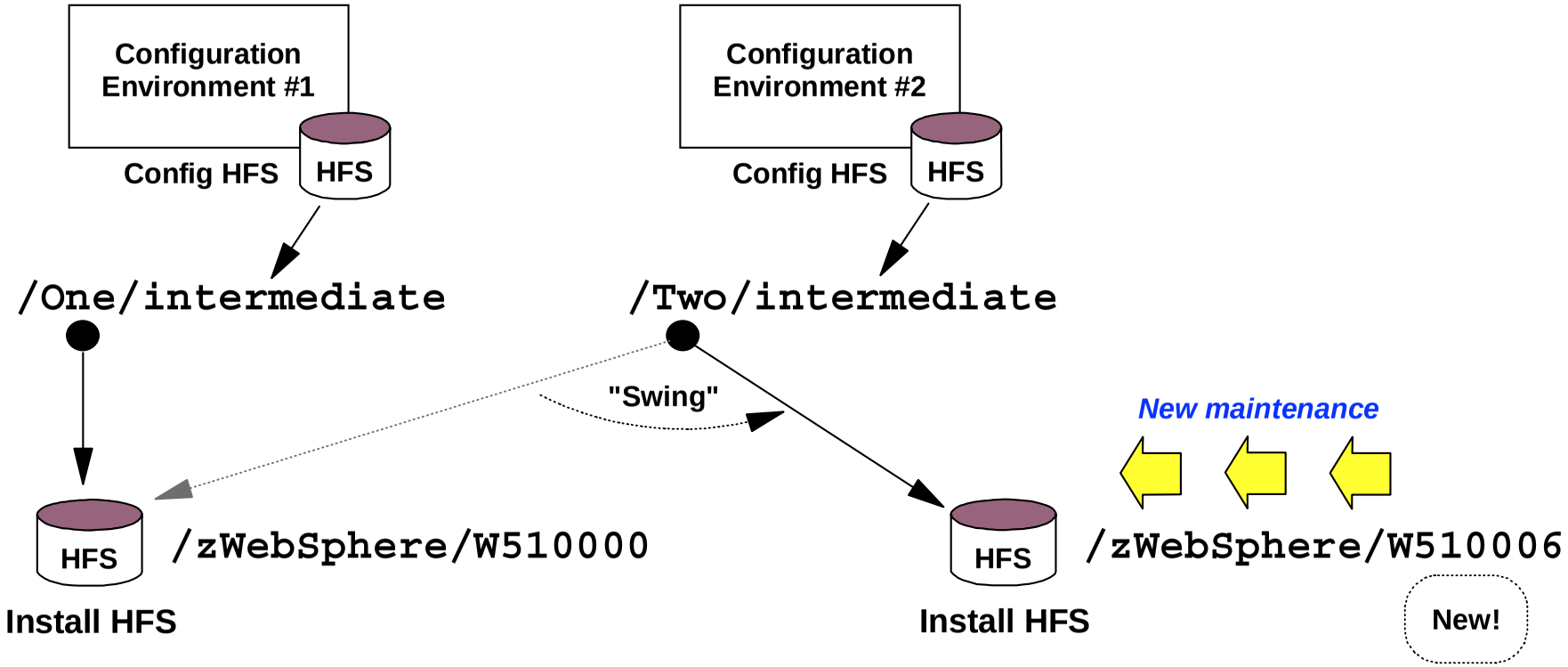


Create the intermediate symbolic link ahead of time and then go through the panels as usual. But rather than pointing directly to the Install HFS, point to the intermediate symbolic link for that configuration.

What's so special about that? Watch ...

Intermediate Symbolic Links

Now when maintenance brought in, link can be changed to new code base

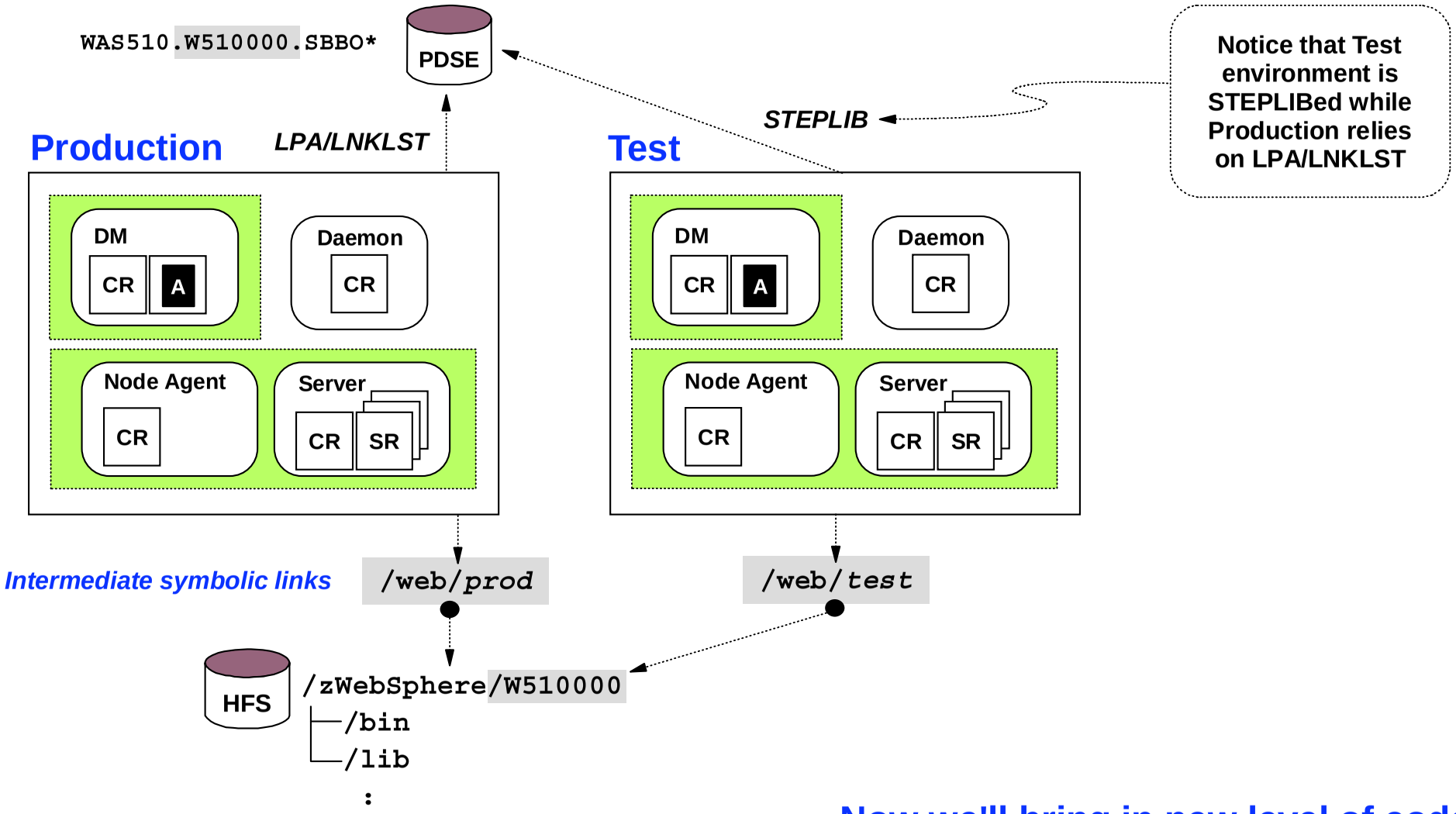


Intermediate symbolic link contents changed to point to new Install HFS

- Introduction of maintenance is under your control
- This allows you to "swing" a config to new maintenance
- This is the key to "rolling" maintenance through a cell (more in a bit)

Test and Production, Part 1

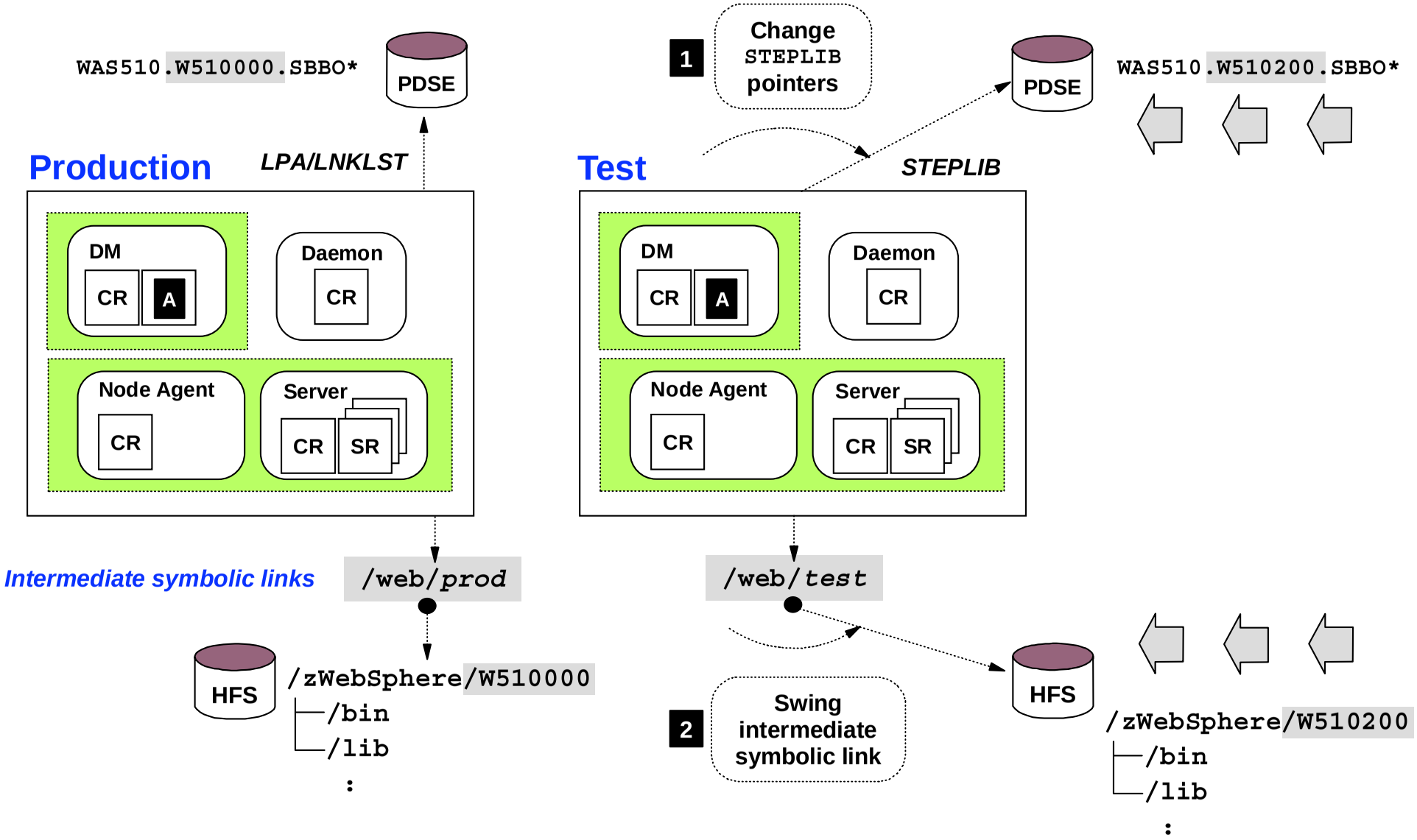
Two separate configuration environments, initially pointed to same code base:



Now we'll bring in new level of code and "swing" the Test environment ...

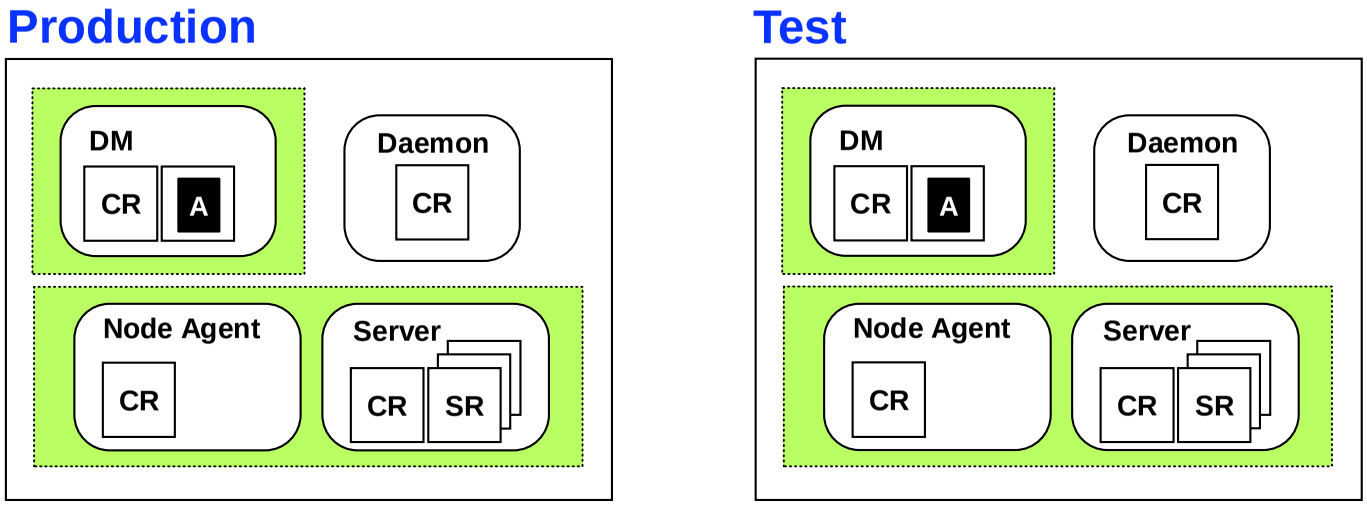
Test and Production, Part 2

Slide in second SMP/E data sets representing new maintenance:



Use of intermediate symbolic links permitted this flexibility

Keys to Test/Production Isolation



Pointers to PDSE Data Sets:

- One environment LPA/LNKLST, other STEPLIB, or
- Both environments STEPLIBed
- Can't both be LPA/LNKLST and achieve isolation
  - Unless on separate MVS images where separate refresh of LPA/LNKLST possible
- Implies different JCL procs for test and production
  - Likely to be the case anyway since different config mount points require different JCL start procedures (because of `SET ROOT=` value in JCL)

Pointers to SMP/E HFS:

- Have physically separate SMP/E HFS mounts points, or
- Use "intermediate symbolic links" to provide ability to isolate

Key is the ability to isolate the configuration to a different code base.

Other Symlinks that are Created

Be aware that JMS Client Java Feature and Java Home Directory on "System Locations 2 of 2" is used to create symlinks in the configuration HFS:

```
System Locations (2 of 2)

   Specify the following for your customization, then press Enter
   to continue.

 Locations of HFS Resident Components

   WebSphere Application Server SMP/E home directory:
     /usr/lpp/zWebSphere/V5R1M0
   WebSphere JMS Client Java Feature SMP/E home directory:
     /usr/lpp/mqm/V5R3M1
   Java home directory:
     /usr/lpp/java2/J1.4
```

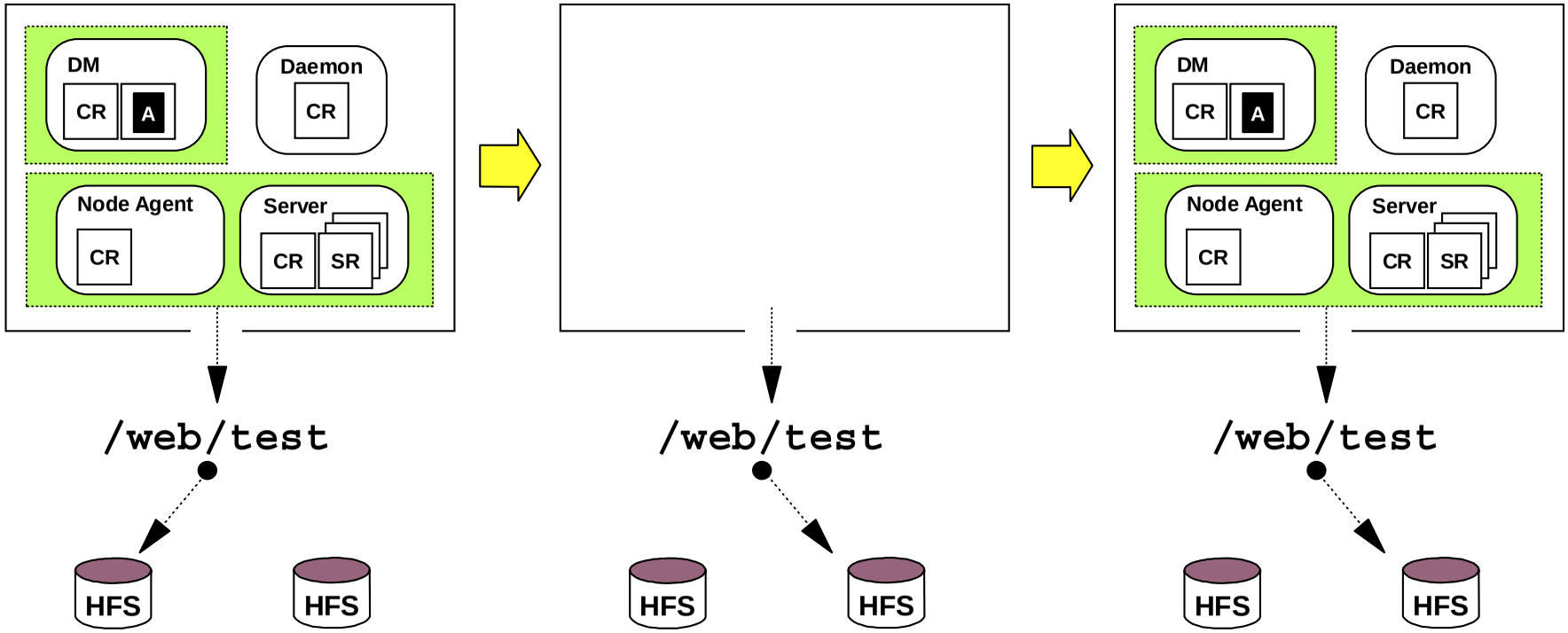
Issue is exactly the same as for "WebSphere SMP/E home directory"

Use intermediate symlinks to provide ability to isolate

Changing Intermediate Symlink Content

Two key points here:

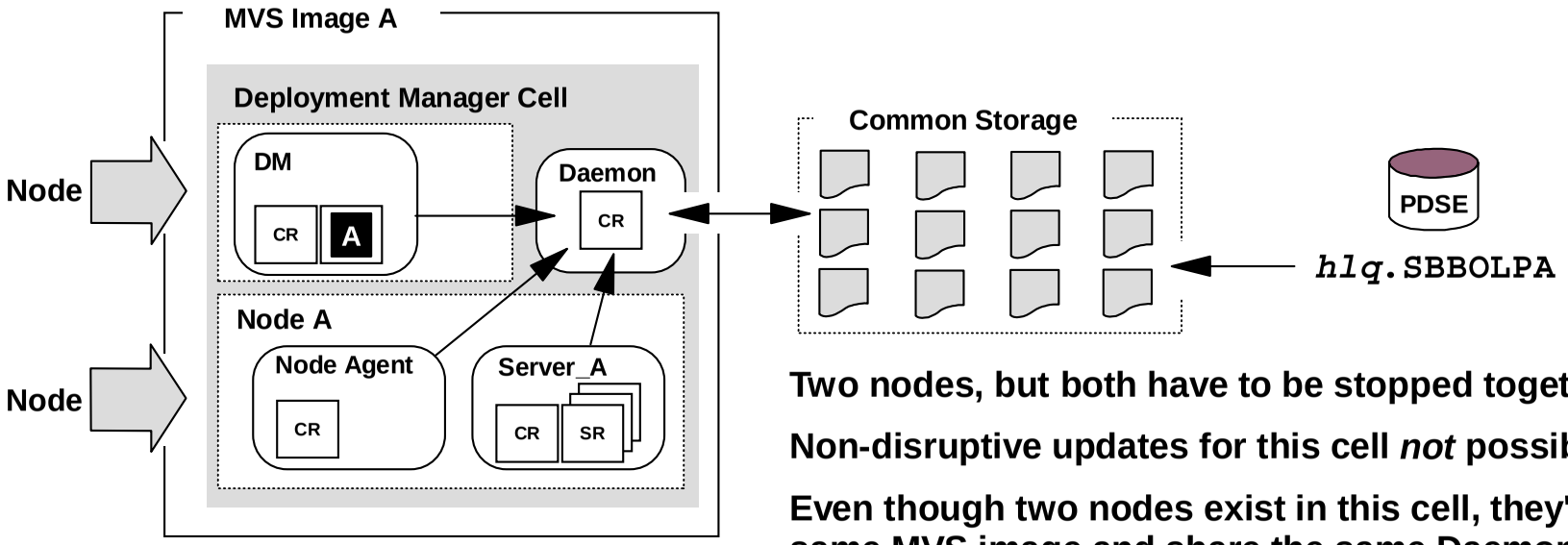
- Must delete intermediate symlink and recreate; can't simply change contents
- If server running that's using intermediate symlink, won't be able to delete



Recall that granularity of symlinks is at the node level. Means code can be introduced node by node. Be careful ... the Daemon comes into play now ...

Two Nodes, Same Cell, Same MVS

The Daemon holds `SBBOLPA` modules in common storage. Other servers in the same cell on that MVS image access those modules through Daemon.



Two nodes, but both have to be stopped together.

Non-disruptive updates for this cell *not* possible

Even though two nodes exist in this cell, they're on the same MVS image and share the same Daemon

Can't wait to update Daemon until after servers -- results in mixed environment: with servers at higher, LPA modules in Daemon's common storage at lower.

To update the code used by a Daemon means stopping/restarting that Daemon.

- **Stop a Daemon and all servers in that cell on the MVS image also stop**
- **Servers in cell on *other MVS images* have their own Daemons; they don't stop.**
- **Therefore, non-disruptive maintenance requires cell to span MVS images**

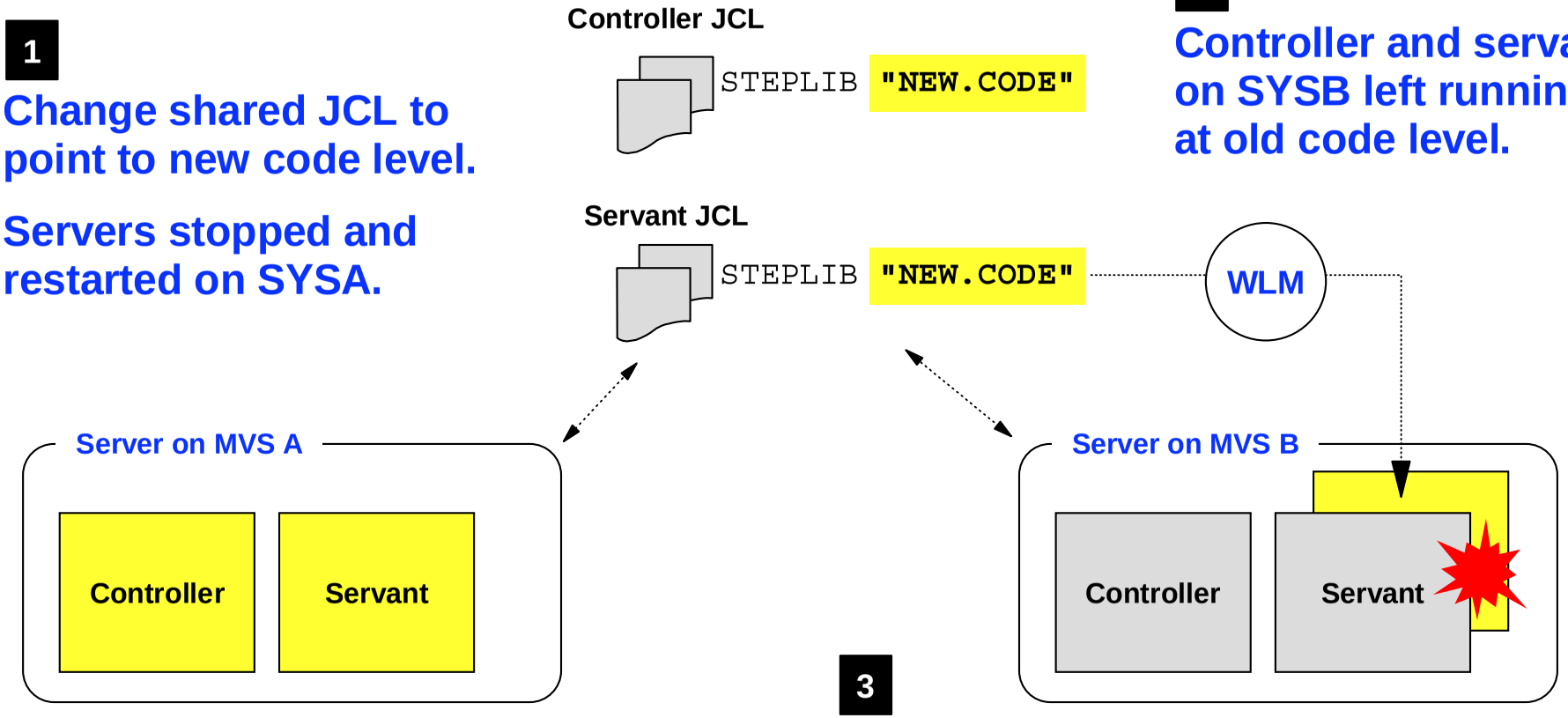
Separate JCL By Node May Be Needed

If your environment uses STEPLIB, need to have separate servant procs for each MVS image. This is because of potential for WLM to start servant at any time:

1

Change shared JCL to point to new code level.

Servers stopped and restarted on SYSA.



2

Controller and servant on SYSB left running at old code level.

3

WLM grabs now-updated JCL and starts a *second* servant region.

Result -- mixed environment

Refresh LPA/LNKLST System by System

If your environment uses LPA/LNKLST, then you can control when a system is toggled to new code by refreshing LPA/LNKLST:

Delete old modules from LPA

- Must be done module by module
- Two WebSphere libraries: **`SBBOLPA`** and **`SBBOLOAD`**
- Create member in **`PARMLIB`** with each module named in **`LPA DELETE`** statement
- Then **`SET PROG=`** that member to delete modules from LPA

Add new modules to LPA

- Easier -- may load whole libraries at once
- Two WebSphere libraries: **`SBBOLPA`** and **`SBBOLOAD`**
- Create member in **`PARMLIB`** pointing to new code level libraries
- Then **`SET PROG=`** that member to add modules to LPA

Create new LNKLST and activate

- One WebSphere libary: **`SBBOLD2`**
- Delete old **`LNKLST`** and add new
- Activate **`LNKLST`**

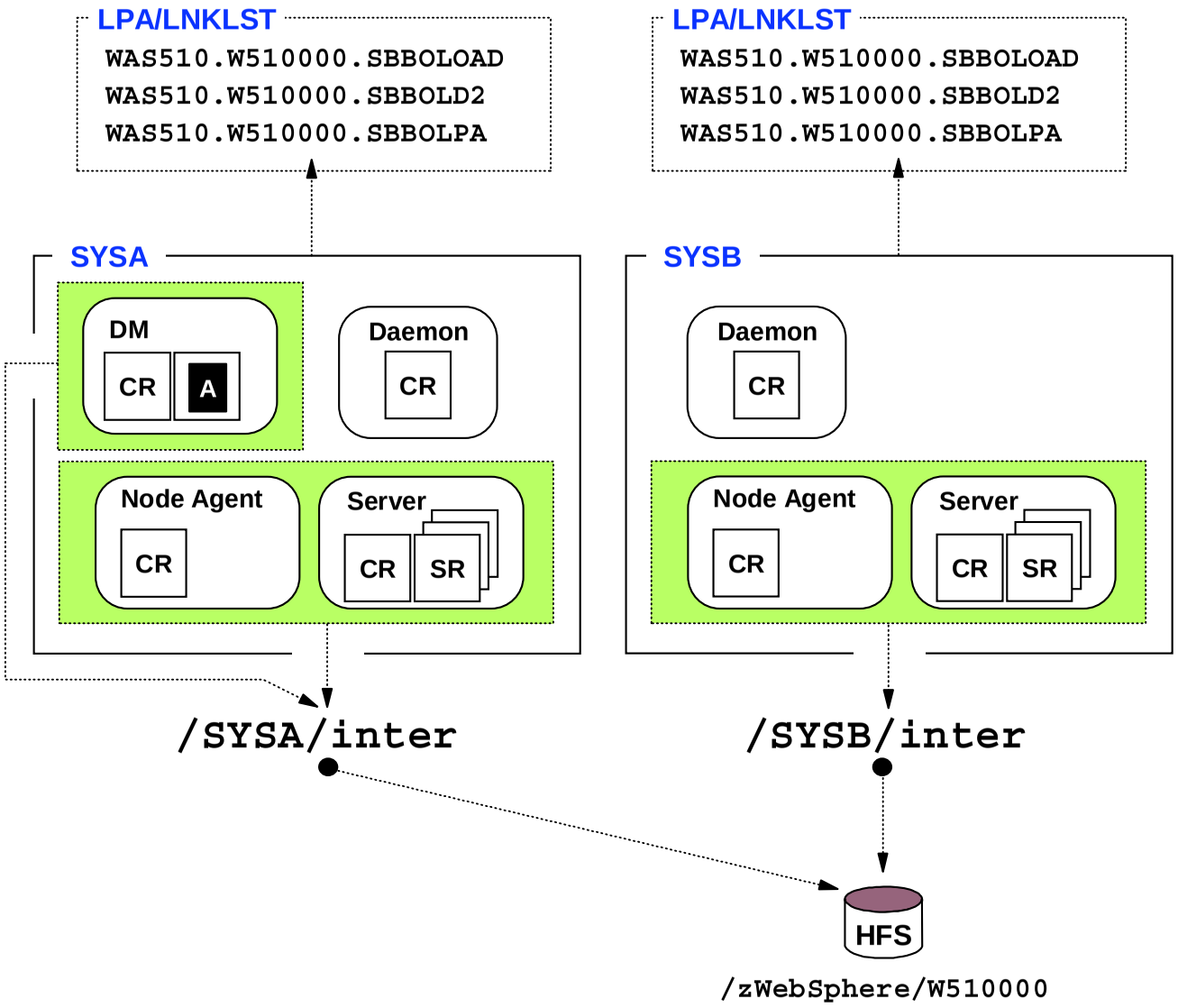
Other datasets that may be in LPA/LNKLST:

- **`SCEERUN`**
- **`SCEERUN2`**
- **`SGSKLOAD`**

See WP100396 for sample PARMLIB members that do these things.

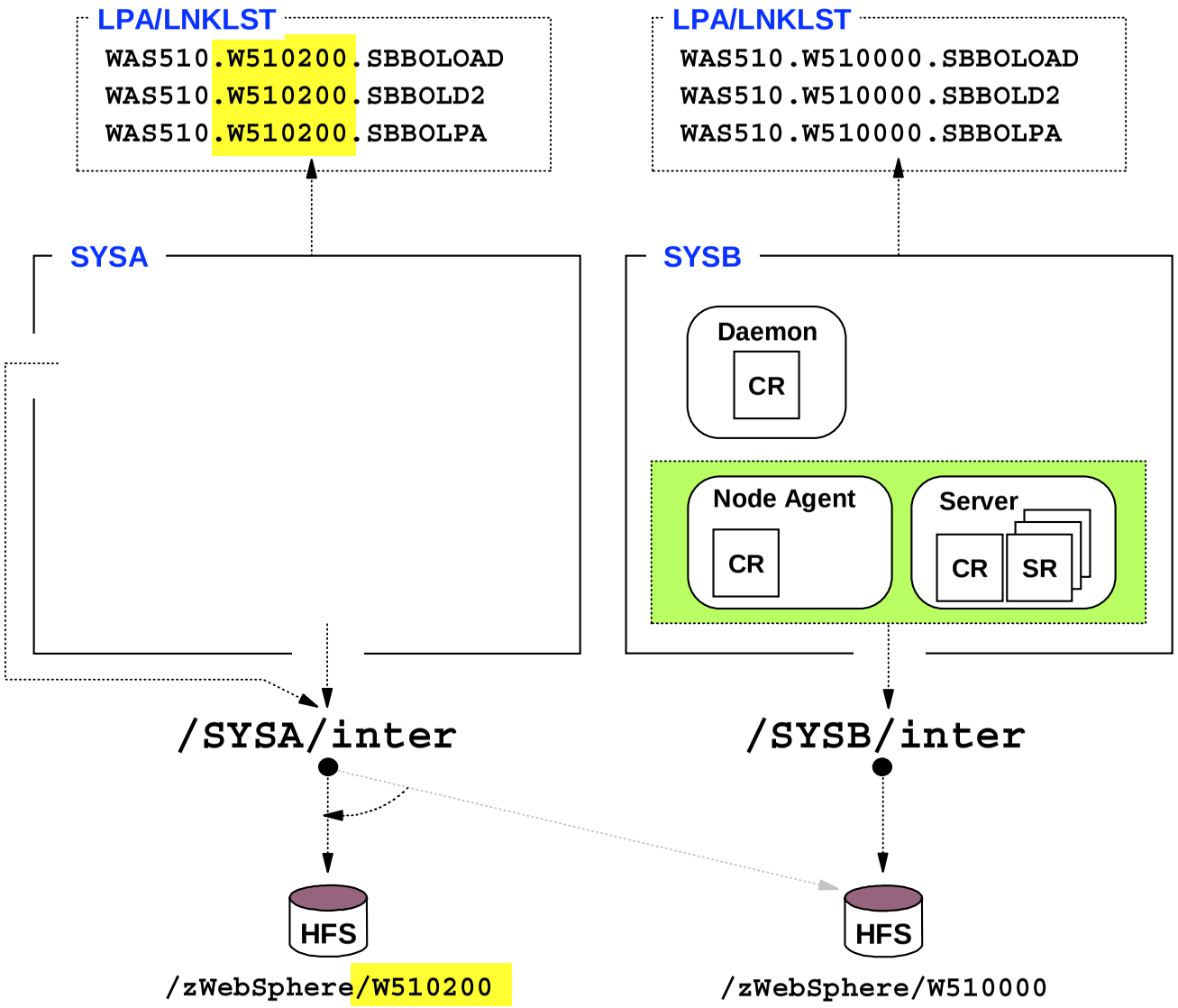
Example Non-Disruptive Rolling Maint.

Start: all nodes at W510000 level



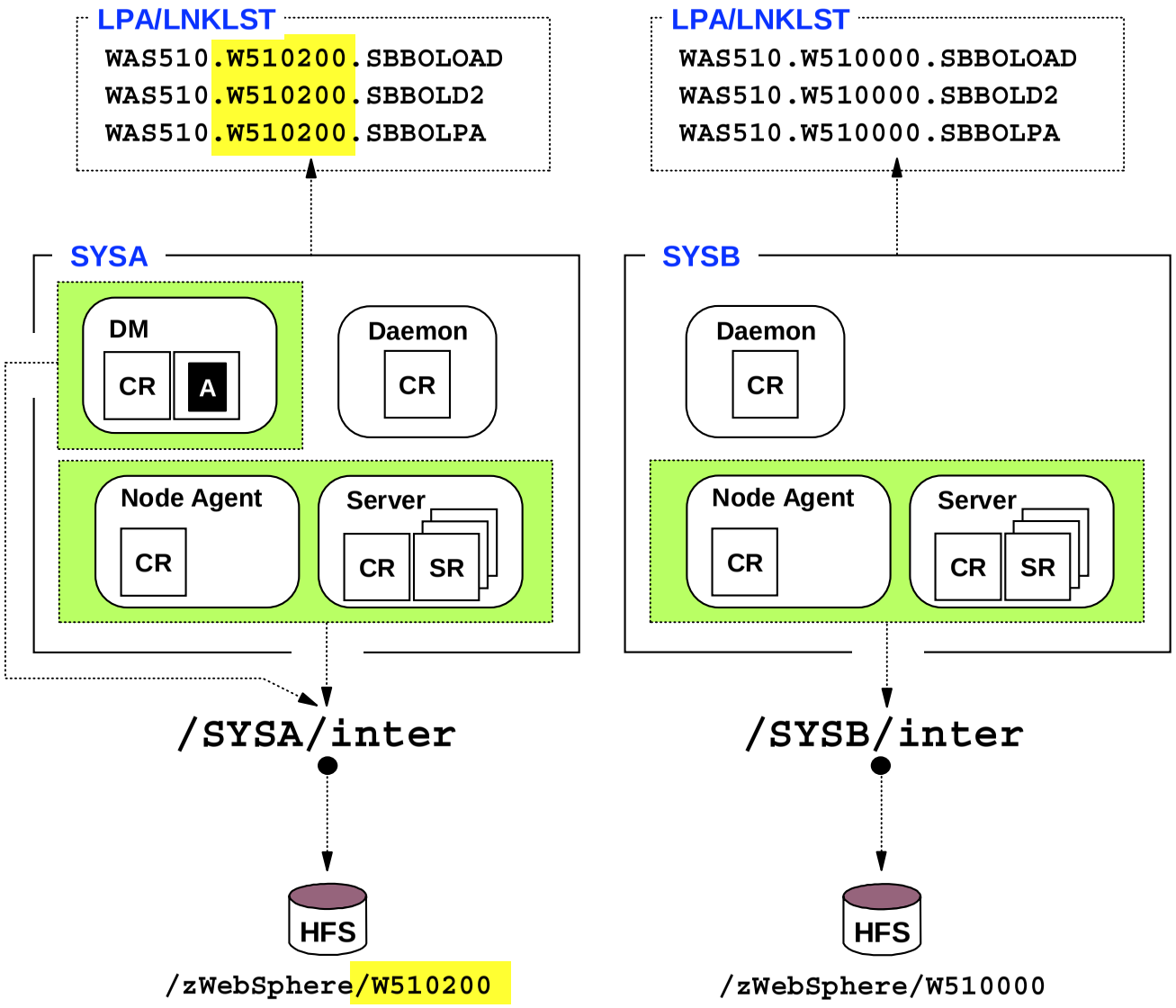
Switch SYSA Systems Nodes

Step 1 -- Stop servers on SYSA. Swing link to new code; refresh LPA/LNKLST



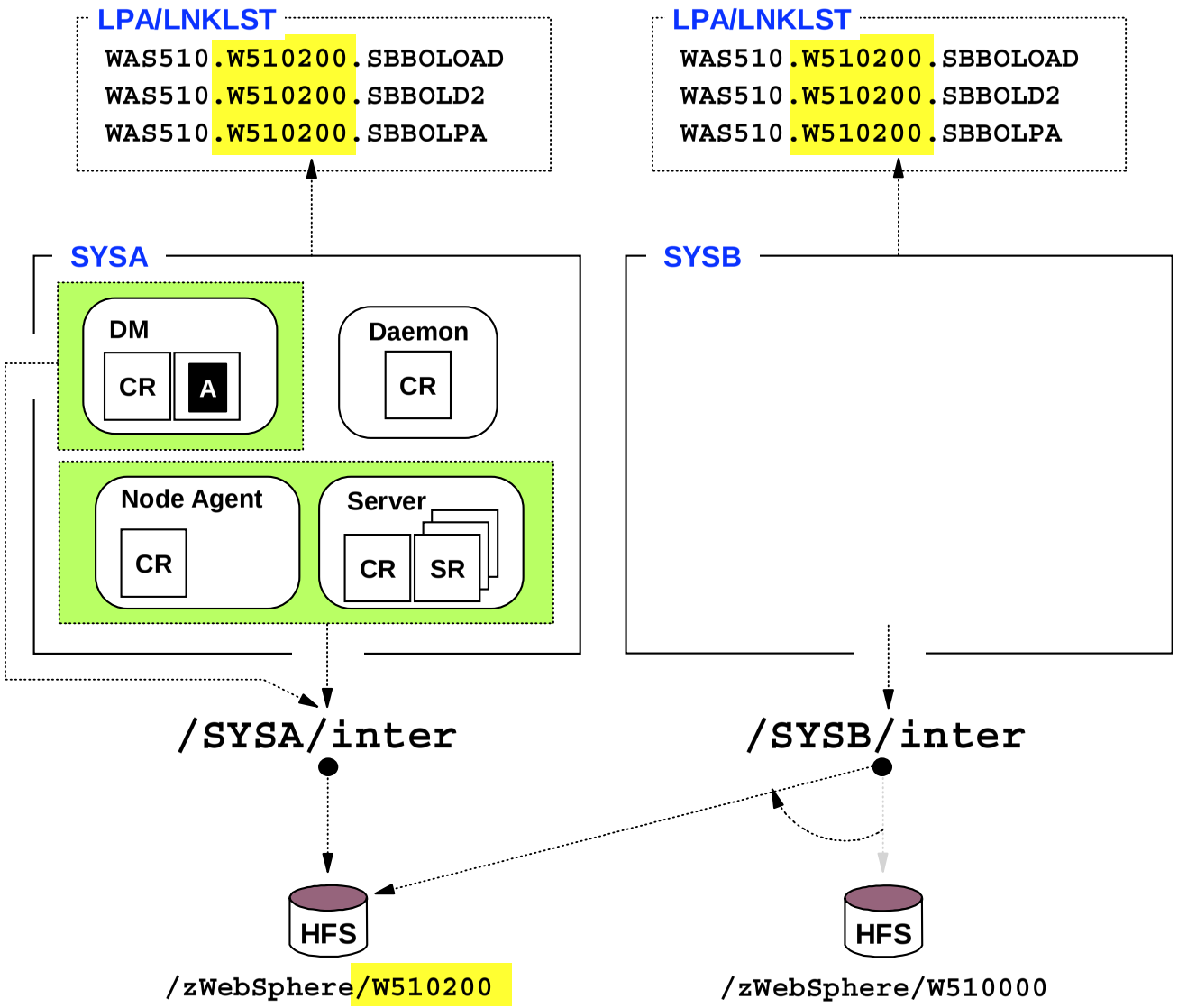
Restart Servers on SYSA

Step 2 -- Restart DMGR and servers on SYSA



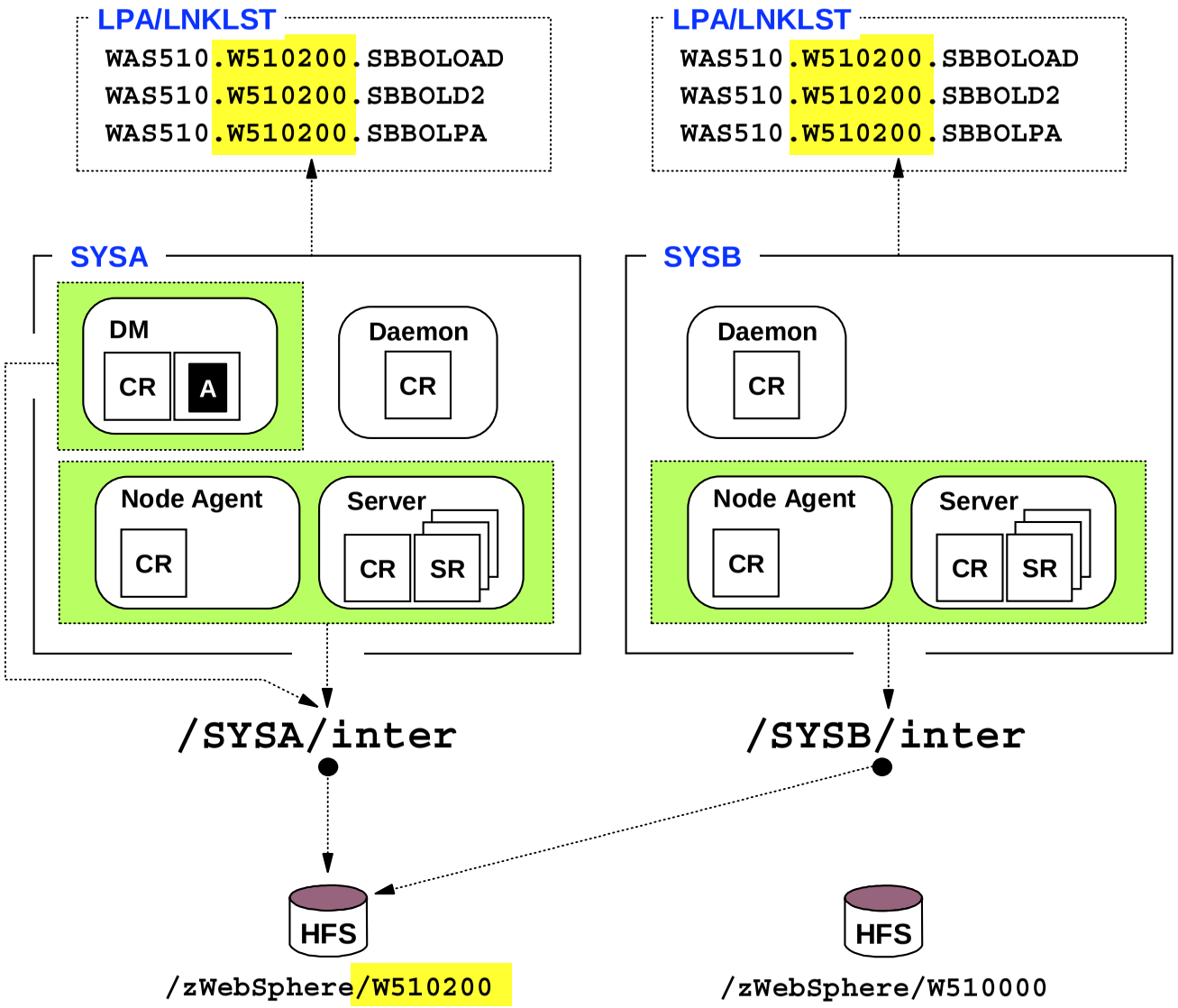
Switch SYSB System Node

Step 3 -- Stop servers on SYSB. Swing link to new code; refresh LPA/LNKLST



Restart Servers on SYSB

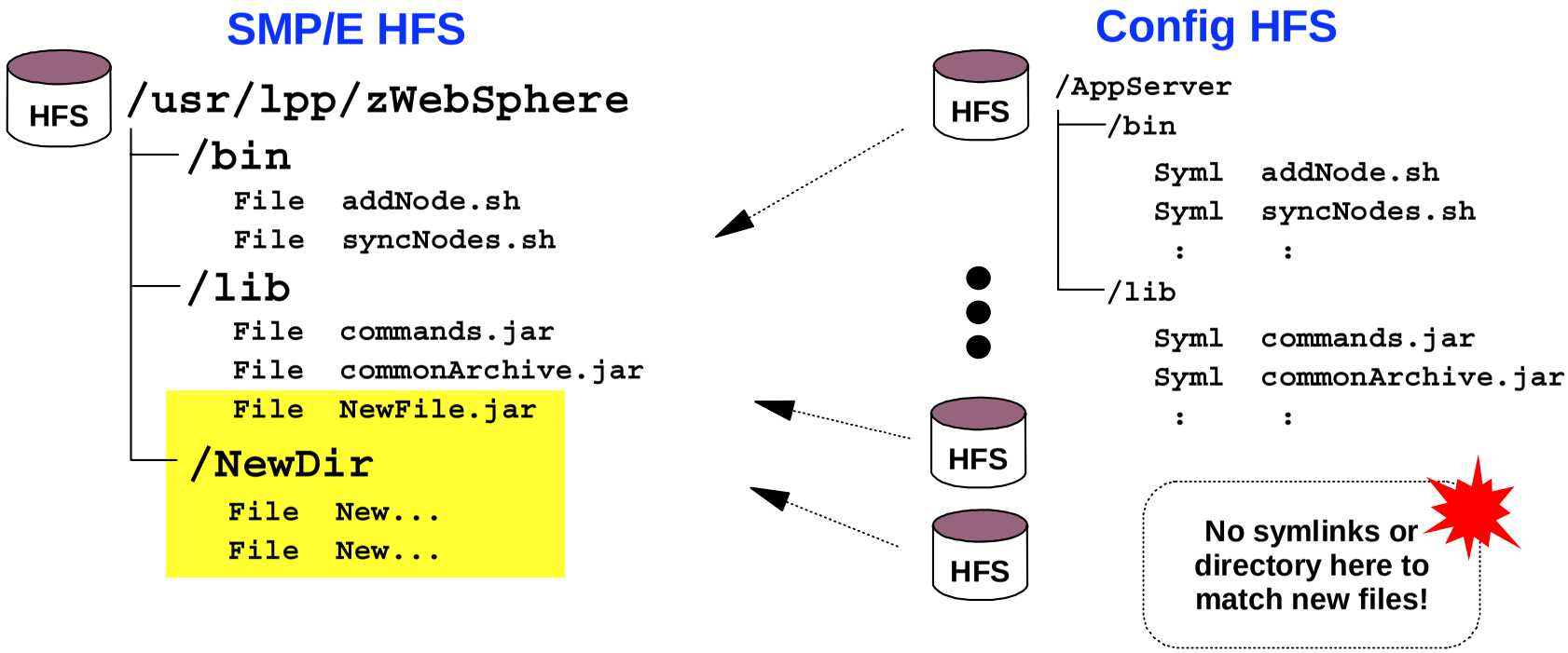
Step 4 -- Restart the servers on SYSB



Maintenance Dilemma

Imagine this scenario:

- Lots of different configurations, each with own Config HFS
- Each Config HFS has symlinks pointing to SMP/E HFS
- IBM maintenance introduces several new files and a directory into SMP/E HFS



SMP/E has no idea about all the different configuration environments. When new files or directories introduced into code HFS, updates to config HFS required complicated, error-prone `++HOLD` processing. Then came `applyPTF.sh` ...

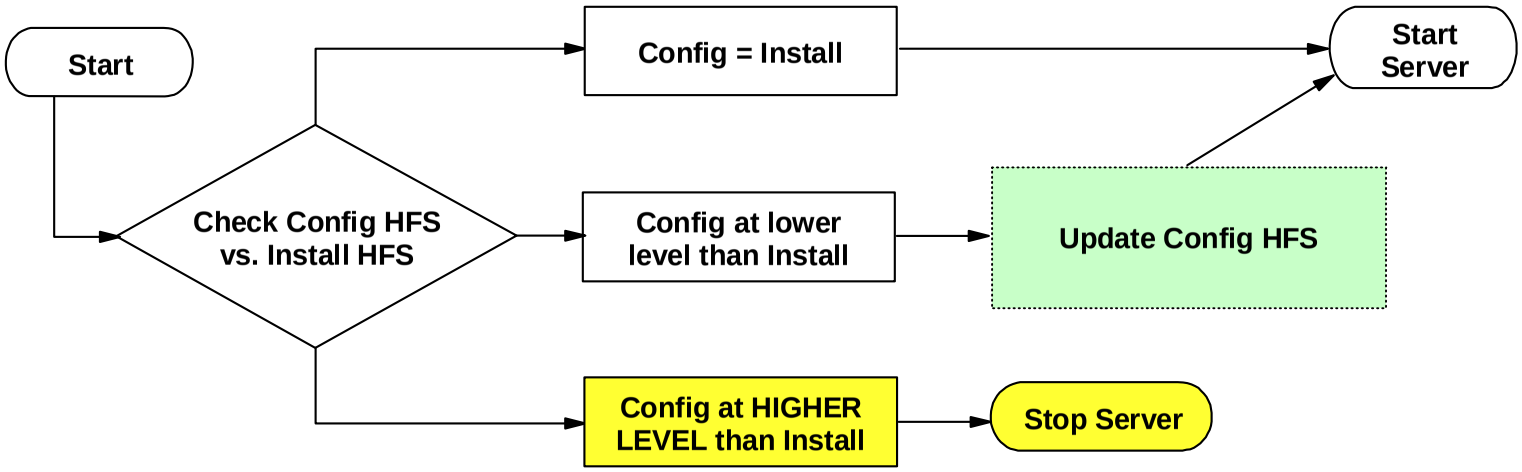
applyPTF.sh Processing

Shell script invoked in controller JCL checks to see if Configuration HFS needs updating:

All controller JCL procedures (DMGR, appserver, Node Agent)

```
//G51ACR  PROC ENV=,PARMS=' ',Z=G51ACRZ
// SET ROOT='/team##/g5cell'
// SET FOUT='properties/service/logs/applyPTF.out'
//APPLY   EXEC PGM=BPXBATCH,REGION=0M,
// PARM='SH &ROOT./&ENV..HOME/bin/applyPTF.sh inline'
```

applyPTF.sh compares Config HFS with Install HFS, then takes this action:



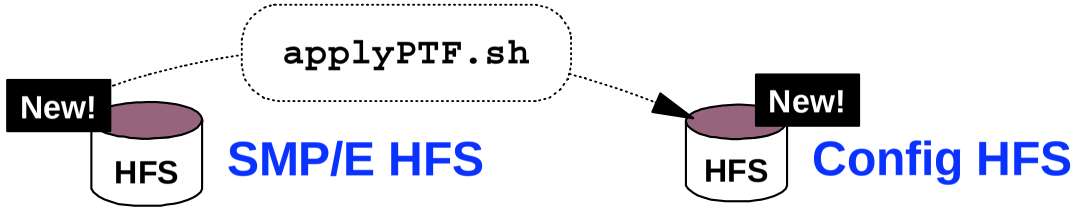
This would be the case if you simply dropped back to lower maintenance after config HFS had been updated

Means *how* you back off maintenance is critical ...

One Way to Back Out Maintenance

Snap a copy of the Config HFS *just before* new maintenance is brought in

Bring new maintenance in ... `applyPTF.sh` updates Config HFS



Bad maintenance! Restore previous SMP/E HFS and copy of Config HFS

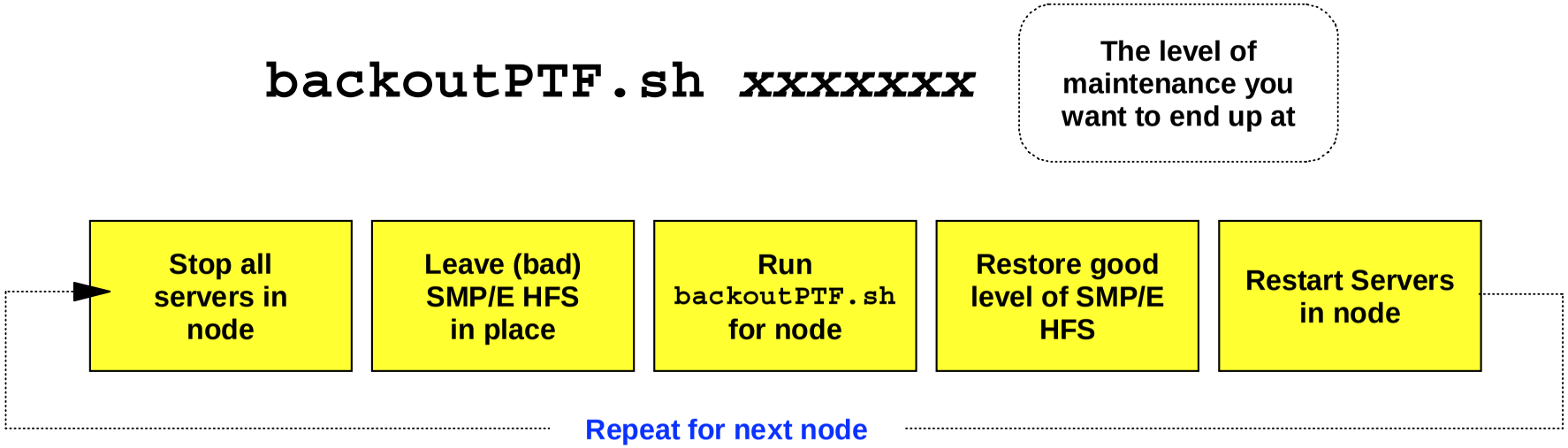
Careful!

- If backup copy isn't recent, falling back may mean you lose configuration changes made since backup taken.
- If after maintenance applied you find you have to fall back, any post-update changes to configuration will be lost.

Another way: `backoutPTF.sh`

Another Way

Another utility -- `backoutPTF.sh` -- is provided. This will restore Config HFS to the specified level. You'll find this in the `/bin` directory of each node.



Notes:

- Process needs "bad" SMP/E HFS left in place so it can know what changes to back out of Config HFS (list of updates for each level of maintenance kept in SMP/E HFS)
- Process can be lengthy ... all the while servers are down
- Remember -- if `applyPTF.sh` not run against node, don't need to back out anything

Message: copy/restore of Config HFS is probably better way to go.