Managing CICS Resources in a Unix File System: Best Practices

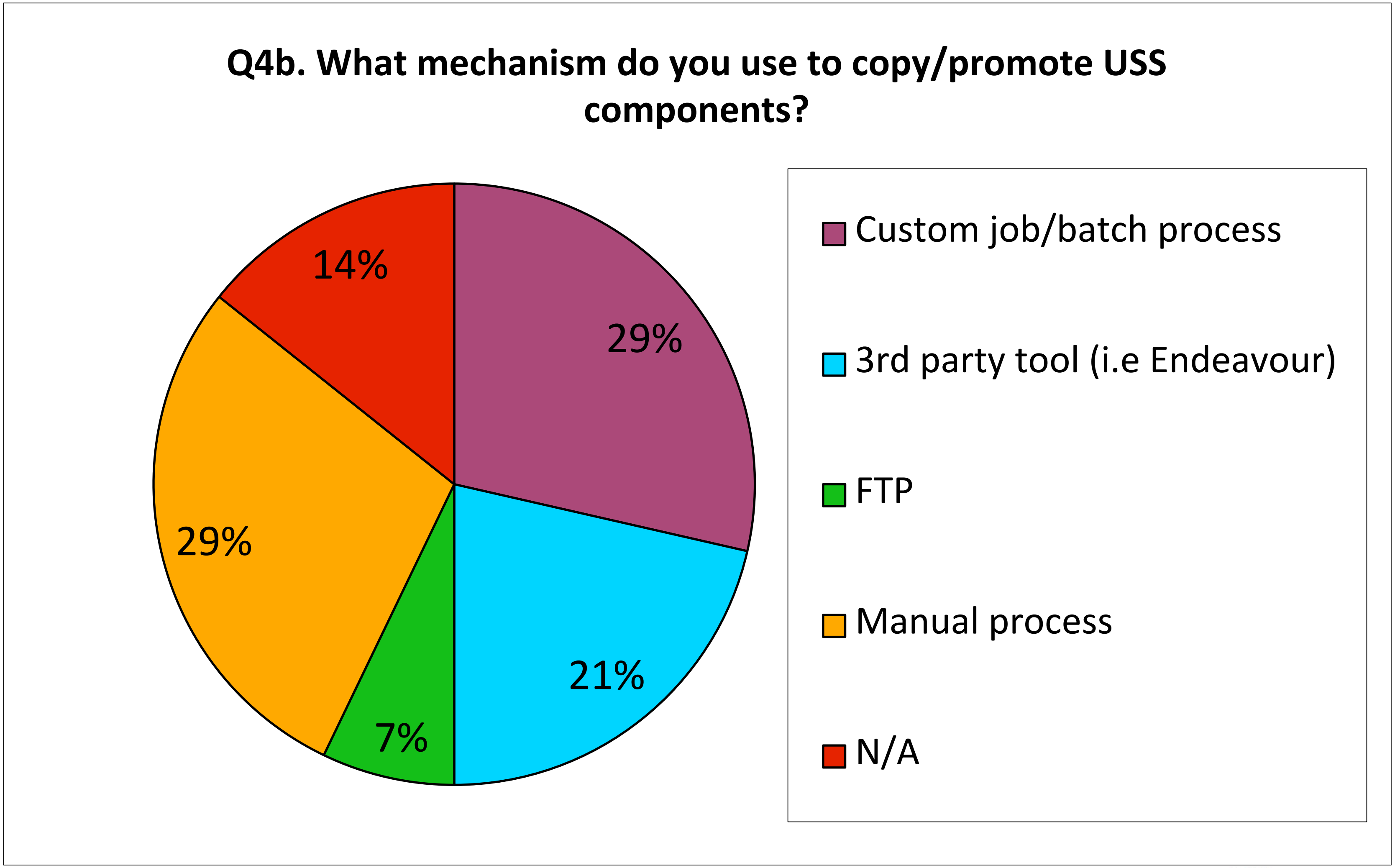
Geoff Pirie
IBM

8 August 2012
Session Number: 11441

Agenda

- Survey of zFS usage
- CICS Resources with zFS components
- Bundles
  - What they are and why we have them
  - What’s in them
  - Their lifecycle
- Best practices
  - Managing resources in development
  - Promoting resources from dev -> test -> production
  - Tips for:
    - Bundles, event adapters/binding, JVMServer
  - zFS setup & security
  - FTP
  - Performance

zFS usage survey :promoting resources



SIT Parameters that point at zFS resources

- The **USSHOME** system initialization parameter specifies the name and path of the root directory for CICS® Transaction Server files on z/OS® UNIX
- The **JVMPROFILEDIR** system initialization parameter specifies the name (up to 240 characters long) of a z/OS® UNIX directory that contains the JVM profiles for CICS®. CICS searches this directory for the profiles it needs to configure JVMs.
    - The default value of **JVMPROFILEDIR** is the same as the default value of the **USSHOME**

Doctemplates

- The **DOCTEMPLATE** resource specifies:
  - **HFSFILE -** When the template resides in a z/OS® UNIX System Services file, this specifies the fully qualified (absolute) or relative name of the z/OS UNIX file. It can be specified as a name relative to the HOME directory of the CICS region userid
- Instructions for setting the userid of the CICS regions can be found here : http://pic.dhe.ibm.com/infocenter/cicsts/v4r2/index.jsp?topic=%2Fcom.ibm.cics.ts.doc%2Fdfha2%2Fparameters%2Fdfha2_usshome.html

Webservices

- The **WEBSERVICE** resource specifies:
  - A **PIPELINE**
  - **WSBIND** Specifies the 1-255 character fully-qualified file name of the Web service binding file on z/OS® UNIX.
  - **WSDLFILE** Specifies the 1-255 character fully-qualified file name of the Web service description (WSDL) file on z/OS UNIX. This file is used when full runtime validation is active.

Pipelines

- The **PIPELINE** resource specifies the name of a z/OS® UNIX file containing an XML description of the nodes and their configuration.
  - **CONFIGFILE**
    - Specifies the name of a z/OS® UNIX file that contains information about the processing nodes that will act on a service request, and on the response.
  - **SHELF**
    - CICS® regions into which the PIPELINE definition is installed must have full permissions to the shelf directory—read, write, and the ability to create subdirectories.
    - A single shelf directory can be shared by multiple CICS regions and by multiple PIPELINE definitions. Within a shelf directory, each CICS region uses a separate subdirectory to keep its files separate from those of other CICS regions. Within each region's directory, each PIPELINE uses a separate subdirectory.
  - **WSDIR**
    - specifies the 1–255 character fully-qualified name of the *Web service binding directory* (also known as the *pickup directory*) on z/OS UNIX.

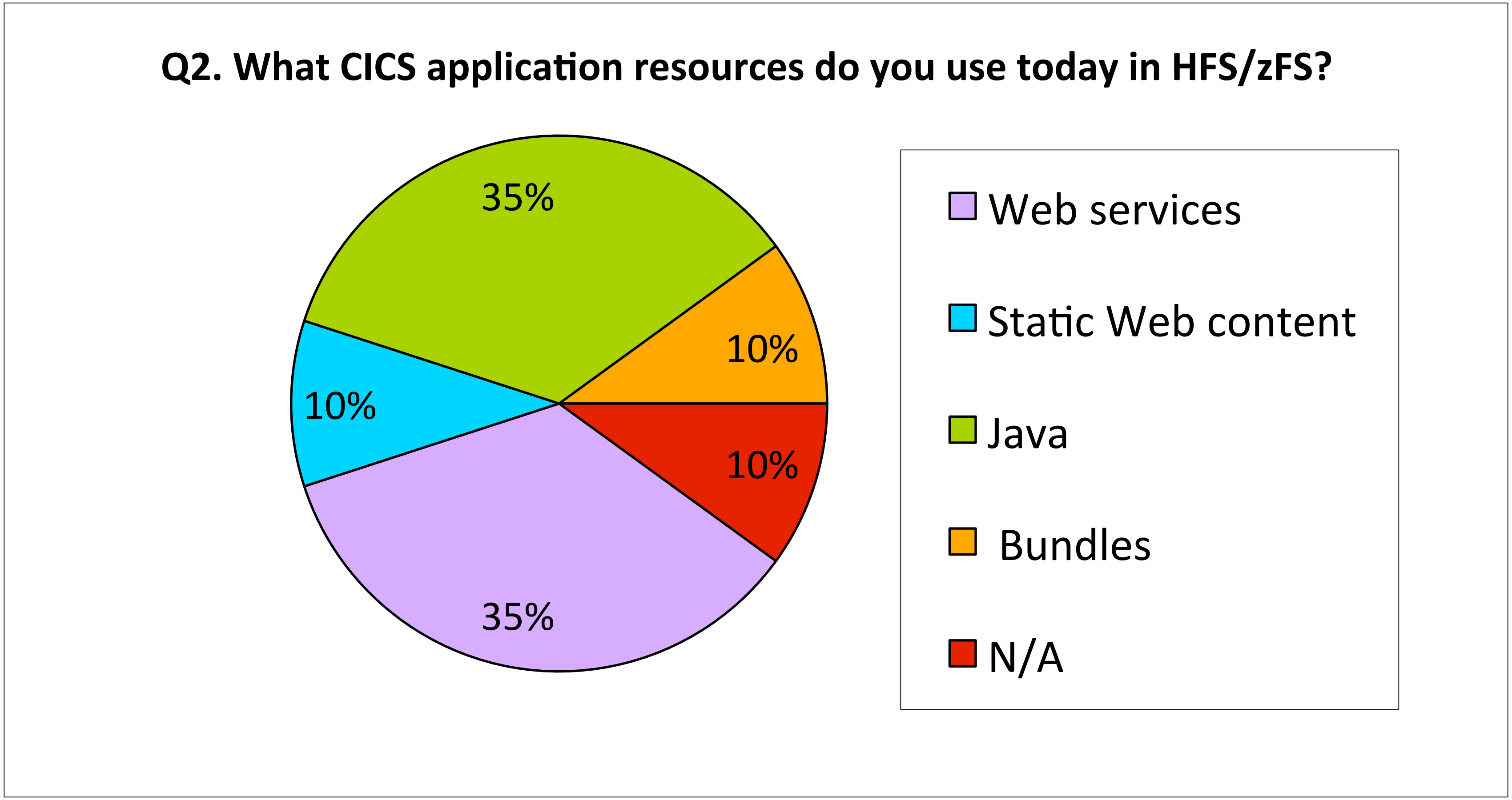
Java

- CLASS files
  - From CICS® Transaction Server for z/OS®, Version 3 Release 2 onwards, the standard class path is constructed in a new way.
  - CICS builds a base standard class path for the JVM using the /lib subdirectories of the directories specified by the **CICS_HOME** and **JAVA_HOME** options in the JVM profile. This standard class path contains the JAR files supplied by CICS and by the JVM. It is not visible in the JVM profile.

Atom & Bundles

- The ATOMSERVICE resource specifies:
  - **BINDFILE** Specifies the fully qualified (absolute) or relative name of an XML binding stored in z/OS® UNIX System Services. For resource types FILE and TSQUEUE, the XML binding is required, for resource type PROGRAM an XML binding is optional.
  - **CONFIGFILE** The Atom configuration file contains XML that specifies metadata and field names for the Atom document
  - ALTHOUGH the Explorer builds and packages all these parts as a bundle for you.
- The **BUNDLE** resource specifies:
  - **BUNDLEDIR** specifies the 1 - 255 character fully qualified name of the root directory for the bundle on z/OS UNIX.

zFS usage survey : types of resource

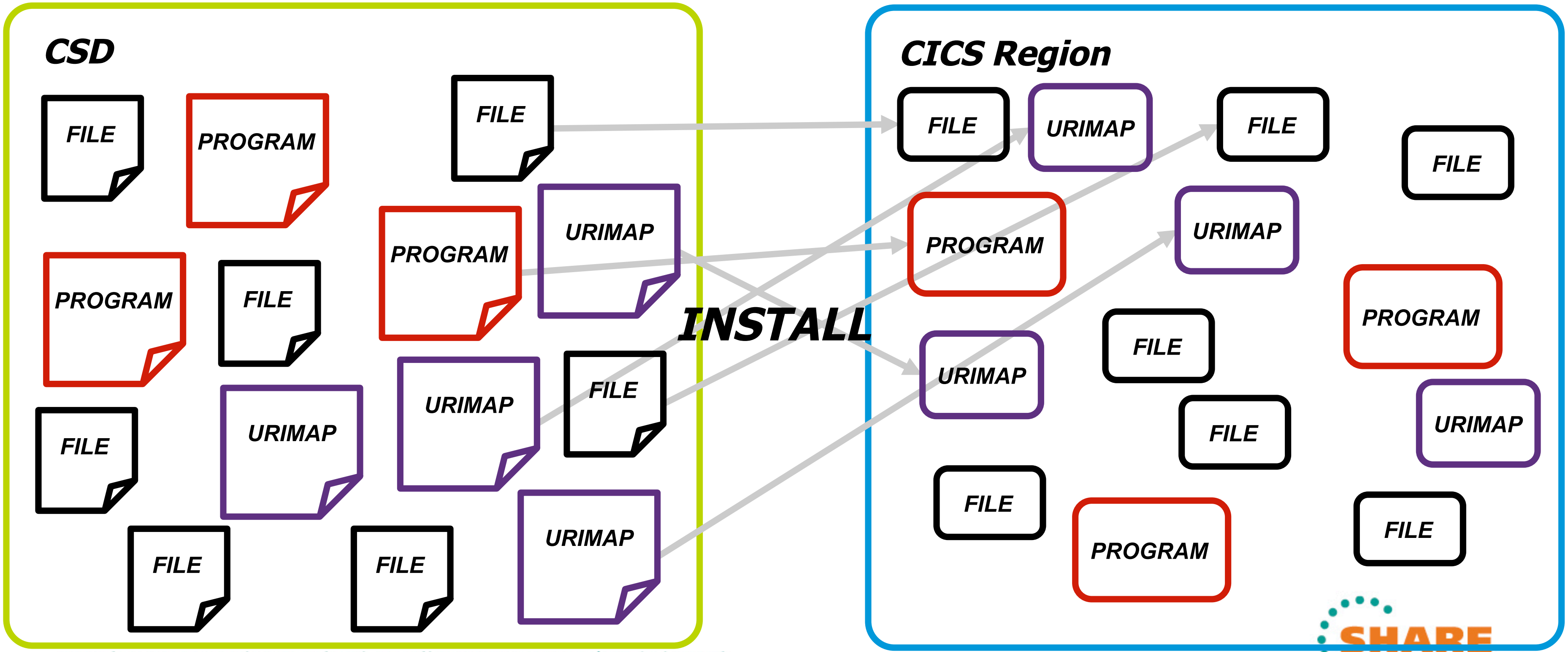


CICS TS 5.1 Open Beta introduces Application and Platform resources which are packaged as bundles…

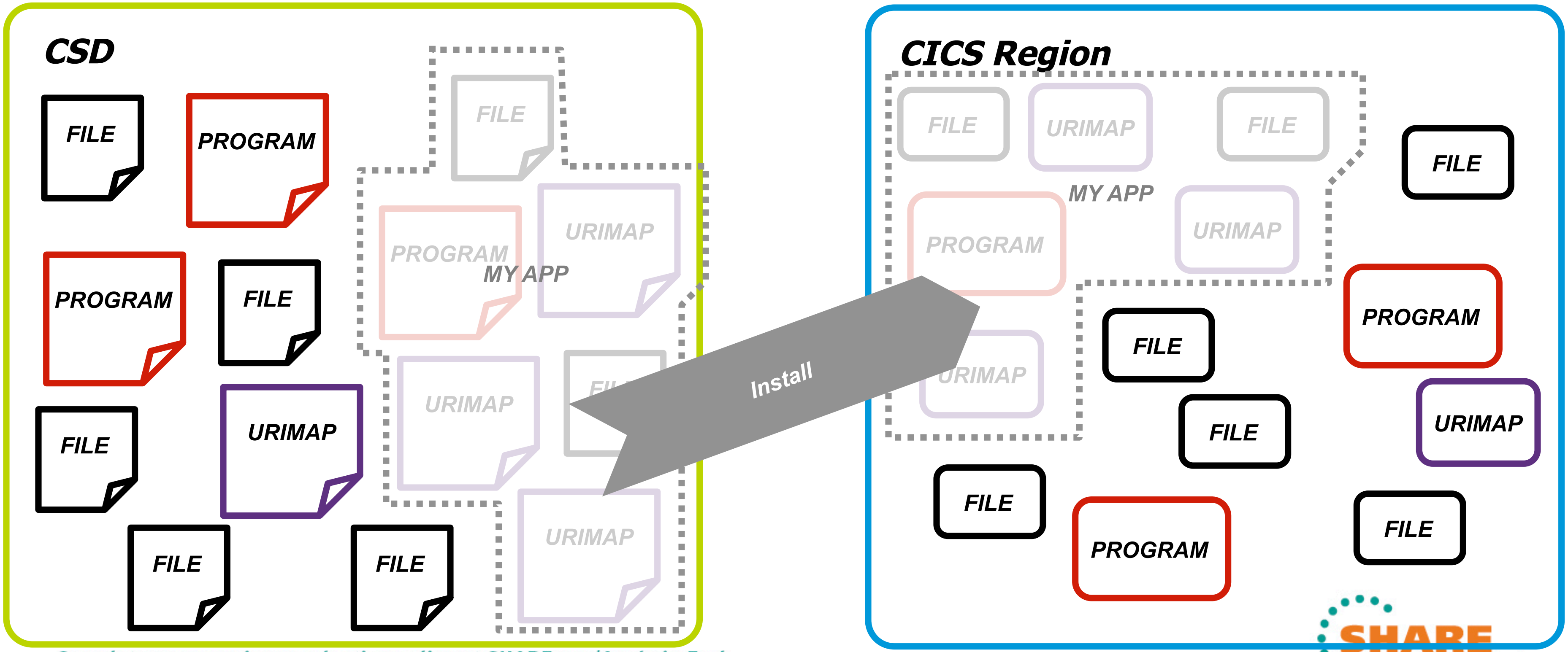
What is a CICS Bundle?

A CICS Bundle is a collection of CICS Resources that can be managed as a logical unit

Need a way to manage an application's CICS Resources as a logical group



Need a way to manage an application's CICS Resources as a logical group



Can't we just use CICS Resource Groups???

Can't we just use CICS Resource Groups???

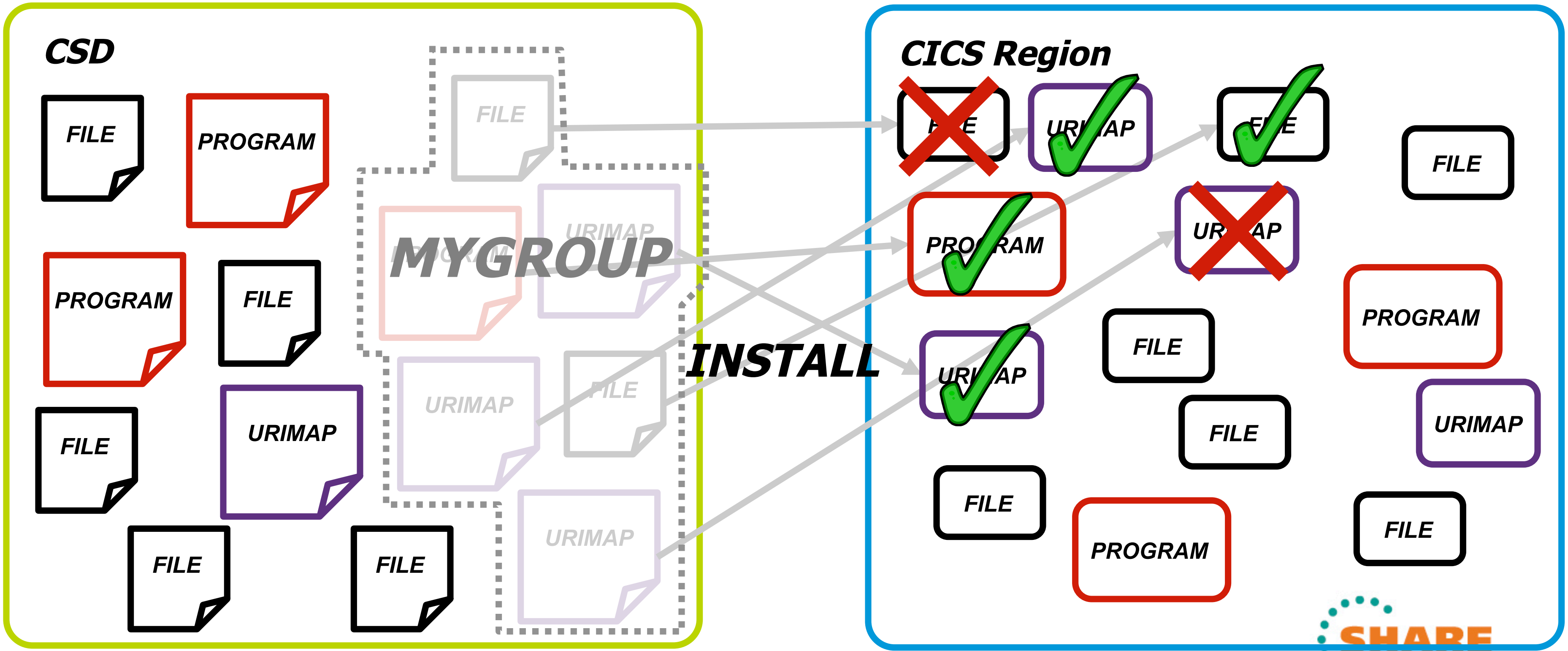
NO!

Well not really...

It's a bit like getting on a flight with your bag packed like this!

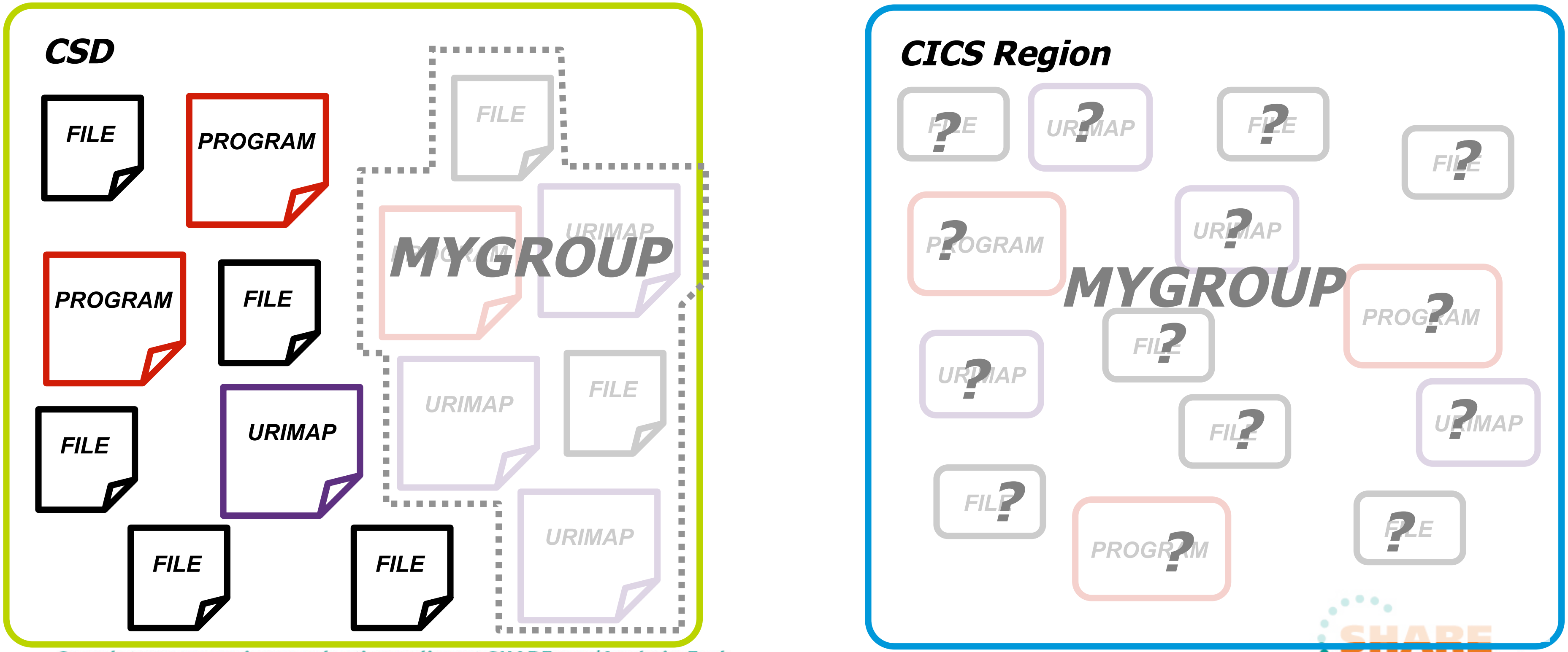
Easy to lose stuff without knowing it

Problem 1: You can accidentally "Part Install" a Resource Group

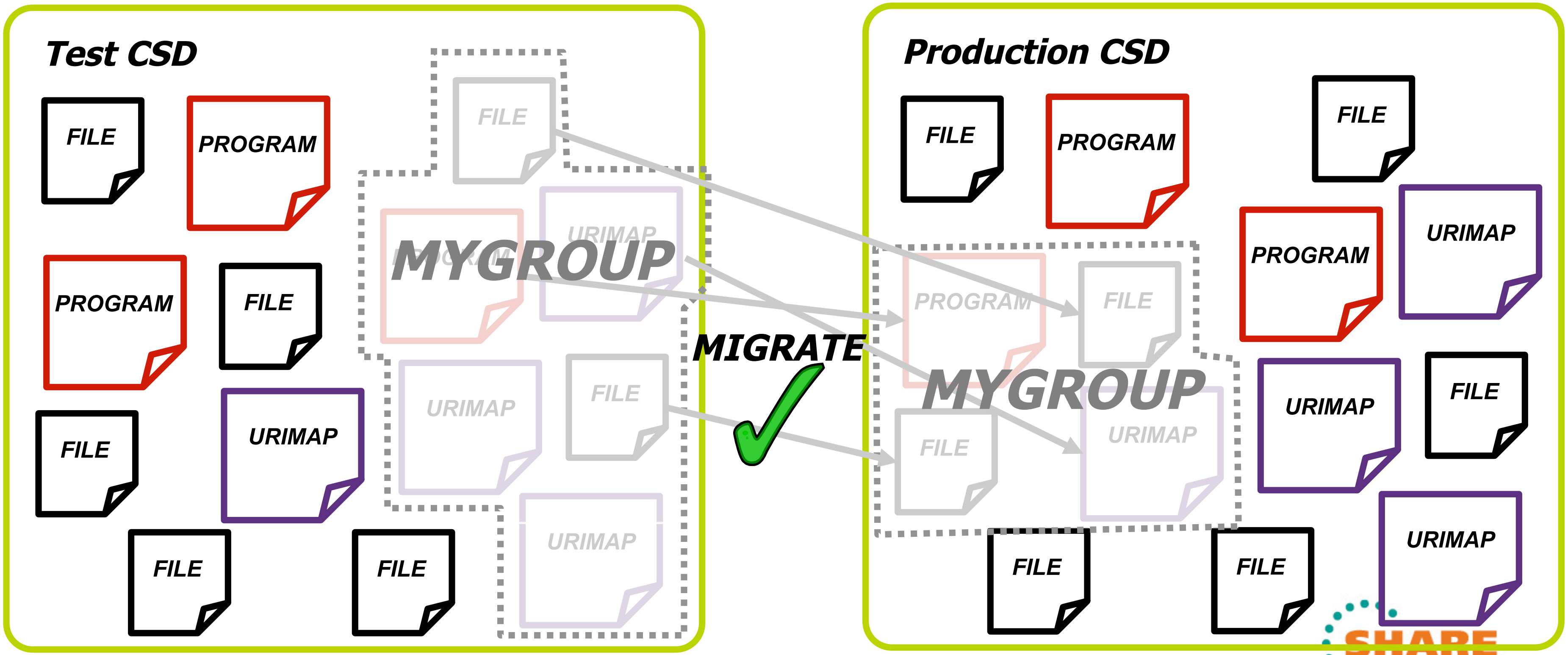


Problem 2: You can't manage the resources as a group once they are installed

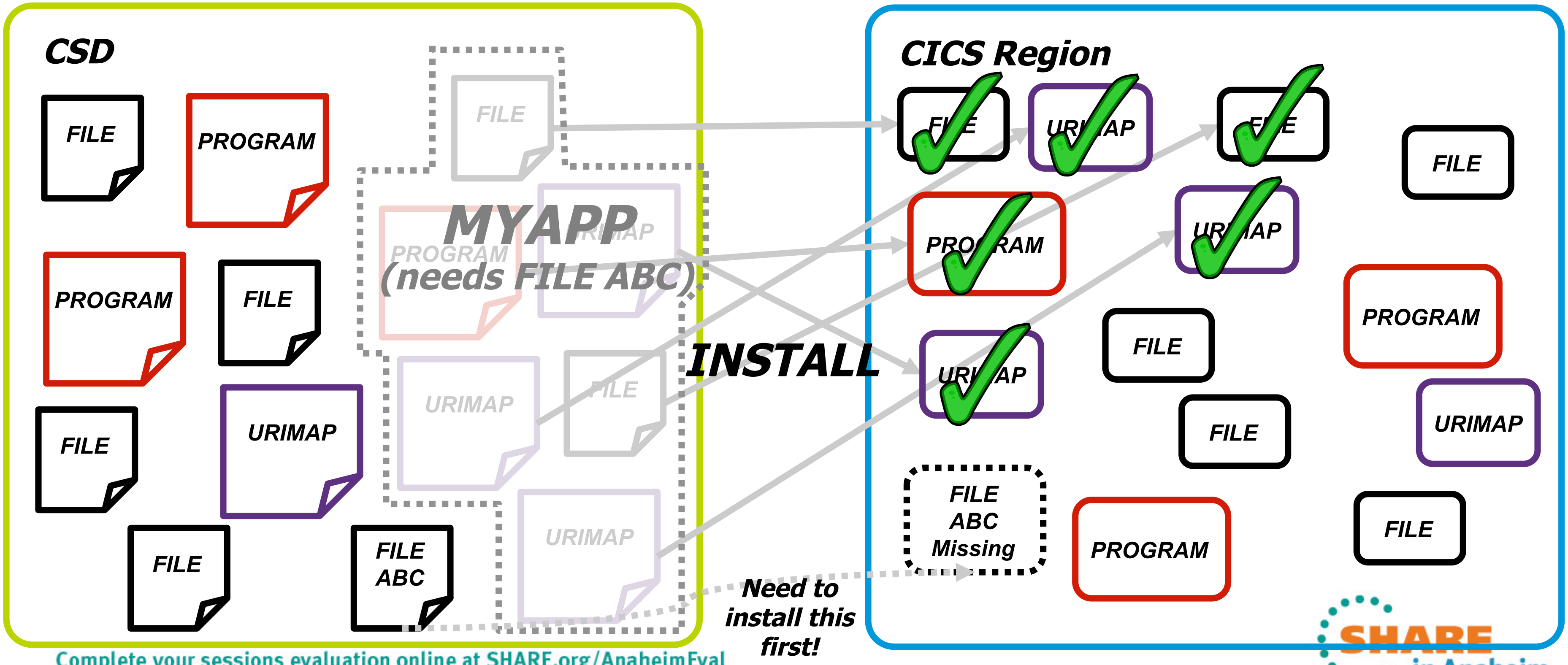
(can't disable or discard as a group)



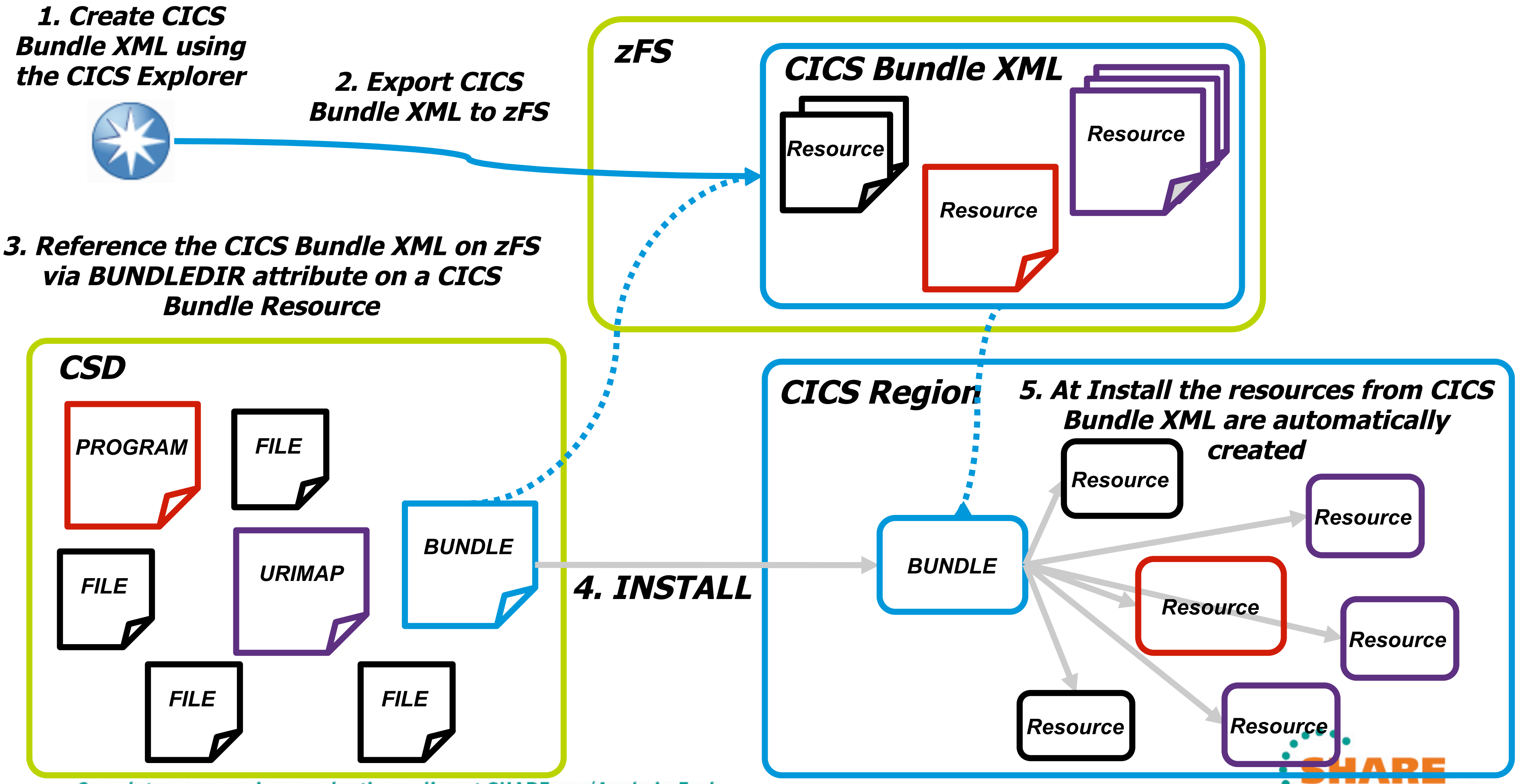
Problem 3: You can unintentionally "Part Migrate" a Resource Group



Problem 4: There is no way to articulate dependencies

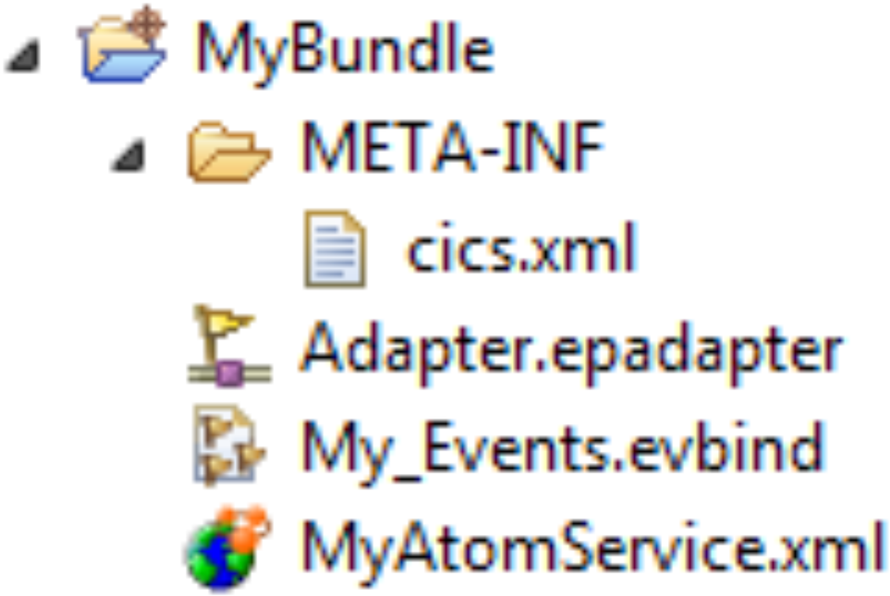


The solution is the CICS Bundle


1. Create CICS Bundle XML using the CICS Explorer
2. Export CICS Bundle XML to zFS
zFS
CICS Bundle XML
Resource
Resource
Resource
3. Reference the CICS Bundle XML on zFS via BUNDLEDIR attribute on a CICS Bundle Resource
CSD
PROGRAM
FILE
FILE
URIMAP
BUNDLE
FILE
FILE
4. INSTALL
CICS Region
5. At Install the resources from CICS Bundle XML are automatically created
BUNDLE
Resource
Resource
Resource
Resource
Resource
Resource

What's in a CICS Bundle?

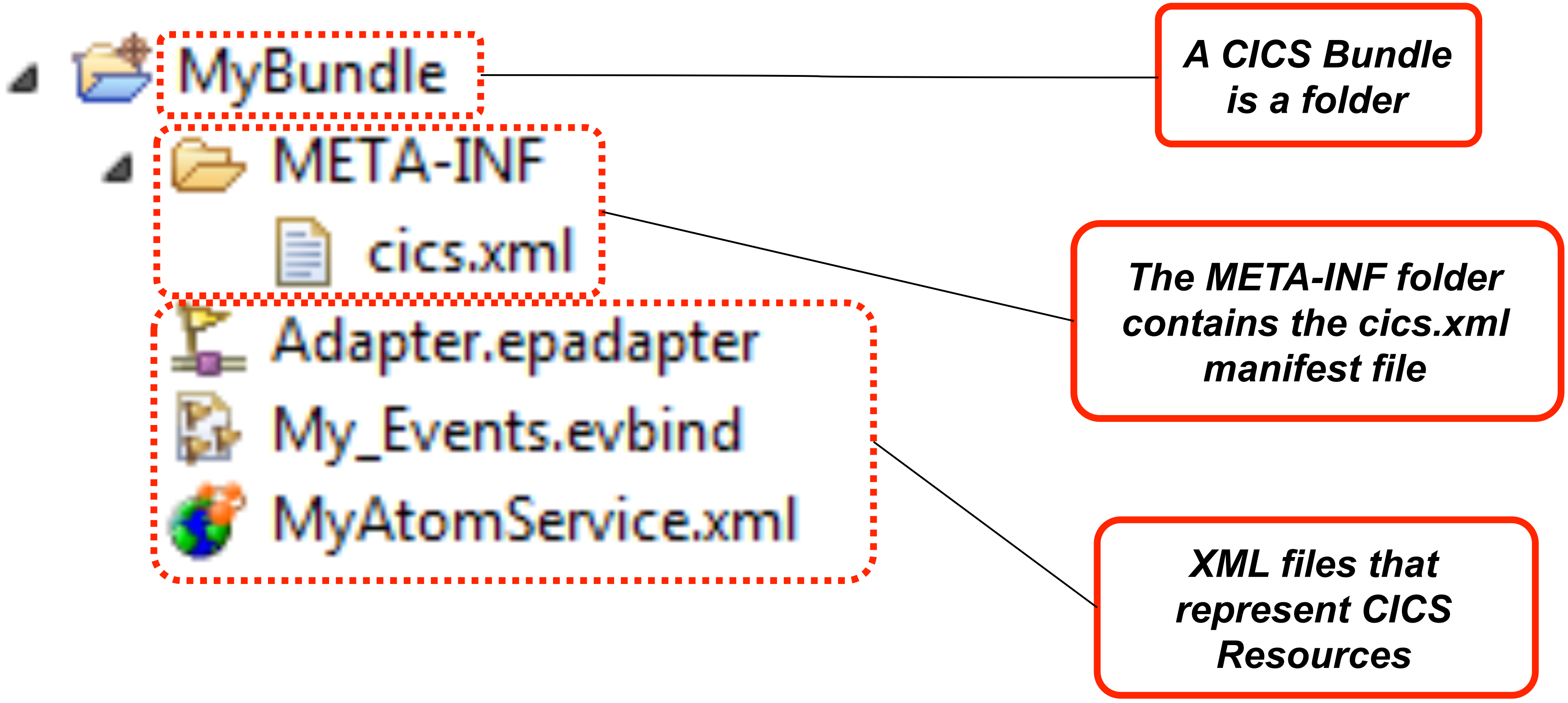
What's in a CICS Bundle?



- MyBundle
  - META-INF
    - cics.xml
  - Adapter.epadapter
  - My_Events.evbind
  - MyAtomService.xml

An example bundle as displayed in the CICS Explorer

What's in a CICS Bundle?



MyBundle

META-INF

cics.xml

Adapter.epadapter

My_Events.evbind

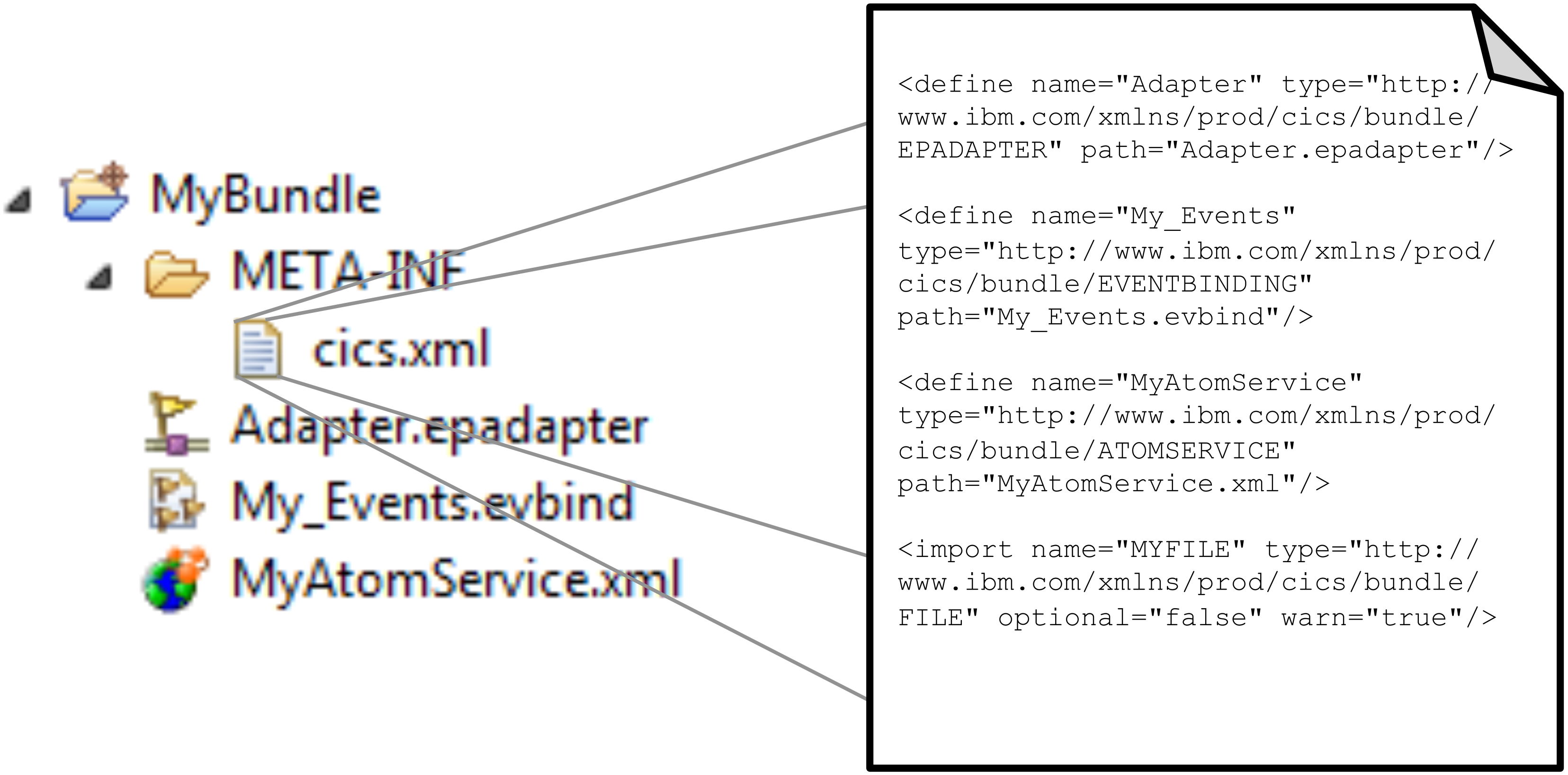
MyAtomService.xml

A CICS Bundle is a folder

The META-INF folder contains the cics.xml manifest file

XML files that represent CICS Resources

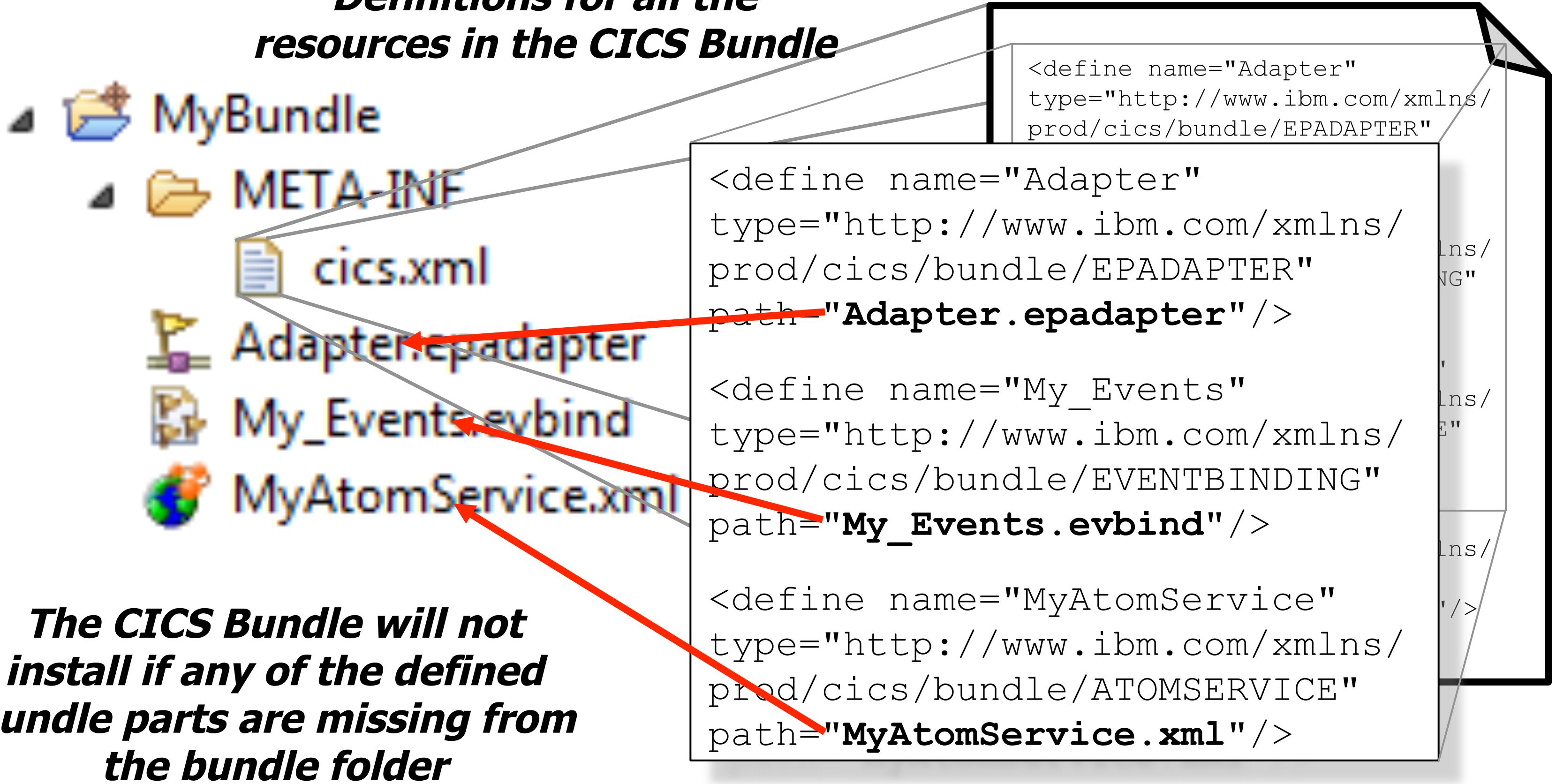
What's in the cics.xml manifest file?



```
<define name="Adapter" type="http://
www.ibm.com/xmlns/prod/cics/bundle/
EPADAPTER" path="Adapter.epadapter"/>

<define name="My_Events"
type="http://www.ibm.com/xmlns/prod/
cics/bundle/EVENTBINDING"
path="My_Events.evbind"/>

<define name="MyAtomService"
type="http://www.ibm.com/xmlns/prod/
cics/bundle/ATOMSERVICE"
path="MyAtomService.xml"/>

<import name="MYFILE" type="http://
www.ibm.com/xmlns/prod/cics/bundle/
FILE" optional="false" warn="true"/>
```

What's in the cics.xml manifest file?

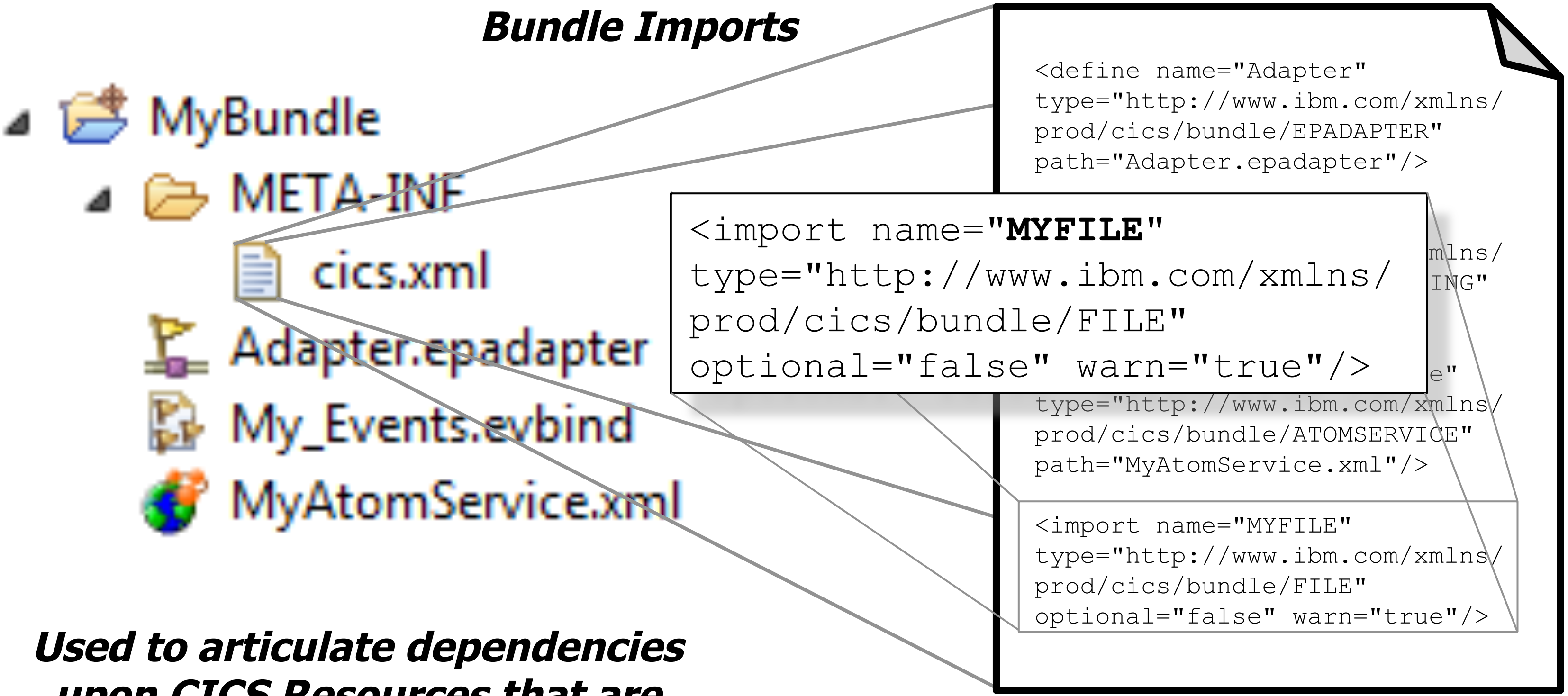
Definitions for all the resources in the CICS Bundle



```
<define name="Adapter"
type="http://www.ibm.com/xmlns/
prod/cics/bundle/EPADAPTER"
path="Adapter.epadapter"/>

<define name="My_Events"
type="http://www.ibm.com/xmlns/
prod/cics/bundle/EVENTBINDING"
path="My_Events.evbind"/>

<define name="MyAtomService"
type="http://www.ibm.com/xmlns/
prod/cics/bundle/ATOMSERVICE"
path="MyAtomService.xml"/>
```

The CICS Bundle will not install if any of the defined bundle parts are missing from the bundle folder

What's in the cics.xml manifest file?



```
<import name="MYFILE"
type="http://www.ibm.com/xmlns/
prod/cics/bundle/FILE"
optional="false" warn="true"/>
```

```
<define name="Adapter"
type="http://www.ibm.com/xmlns/
prod/cics/bundle/EPADAPTER"
path="Adapter.epadapter"/>

mlns/
ING"

e"
type="http://www.ibm.com/xmlns/
prod/cics/bundle/ATOMSERVICE"
path="MyAtomService.xml"/>
<import name="MYFILE"
type="http://www.ibm.com/xmlns/
prod/cics/bundle/FILE"
optional="false" warn="true"/>
```

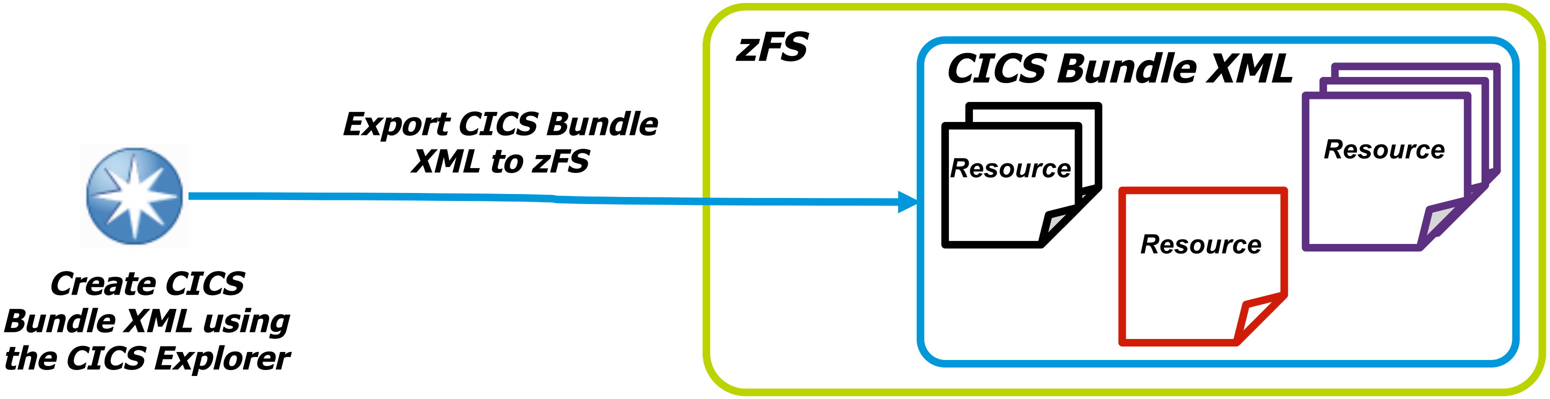
Used to articulate dependencies upon CICS Resources that are not in this bundle

Bundle Lifecycle

- Creating/Editing CICS Bundles and exporting to zFS
- Install CICS Bundles into CICS
- Enabling, Disabling and Discarding

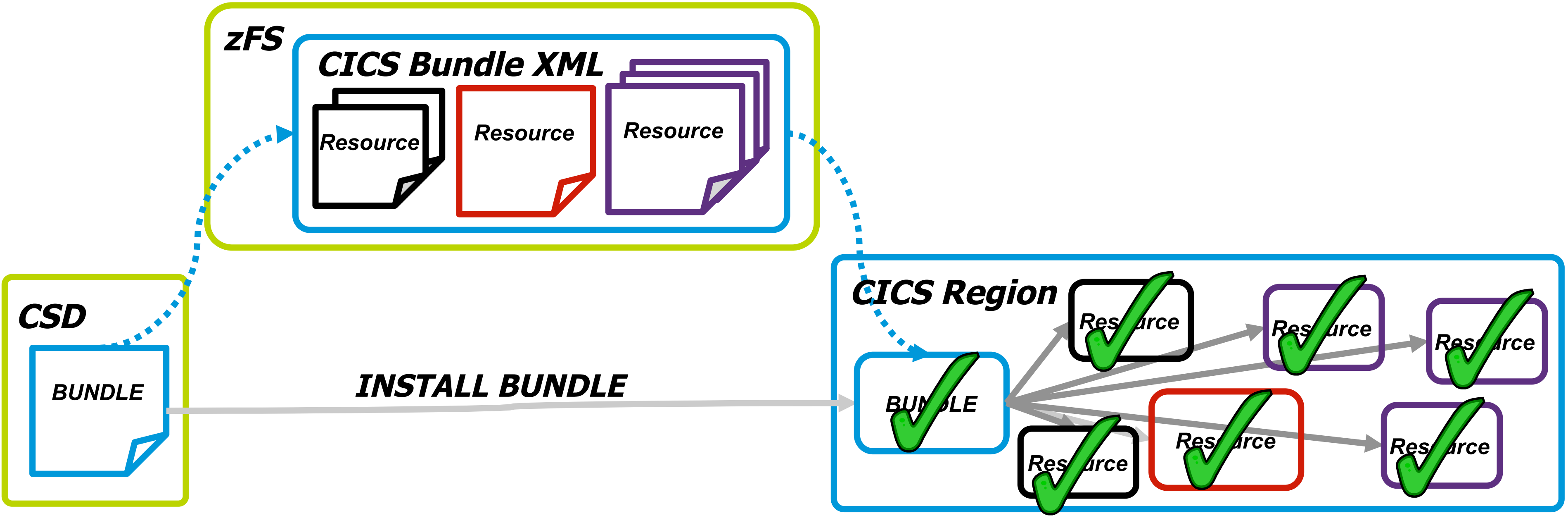
Creating/Editing CICS Bundles and Exporting to zFS

- The CICS Bundle XML is created and edited in the CICS Explorer
- The CICS Explorer **MUST** be used to export the CICS Bundle XML to zFS
- The export to zFS operation is **ONE WAY** (like compiling) you cannot import from zFS
- CICS Bundle XML should **NOT** be edited on zFS directly
- If you need to change the the CICS Bundle XML it **MUST** be re-exported using the CICS Explorer


zFS
CICS Bundle XML
Resource
Resource
Resource
Export CICS Bundle XML to zFS
Create CICS Bundle XML using the CICS Explorer

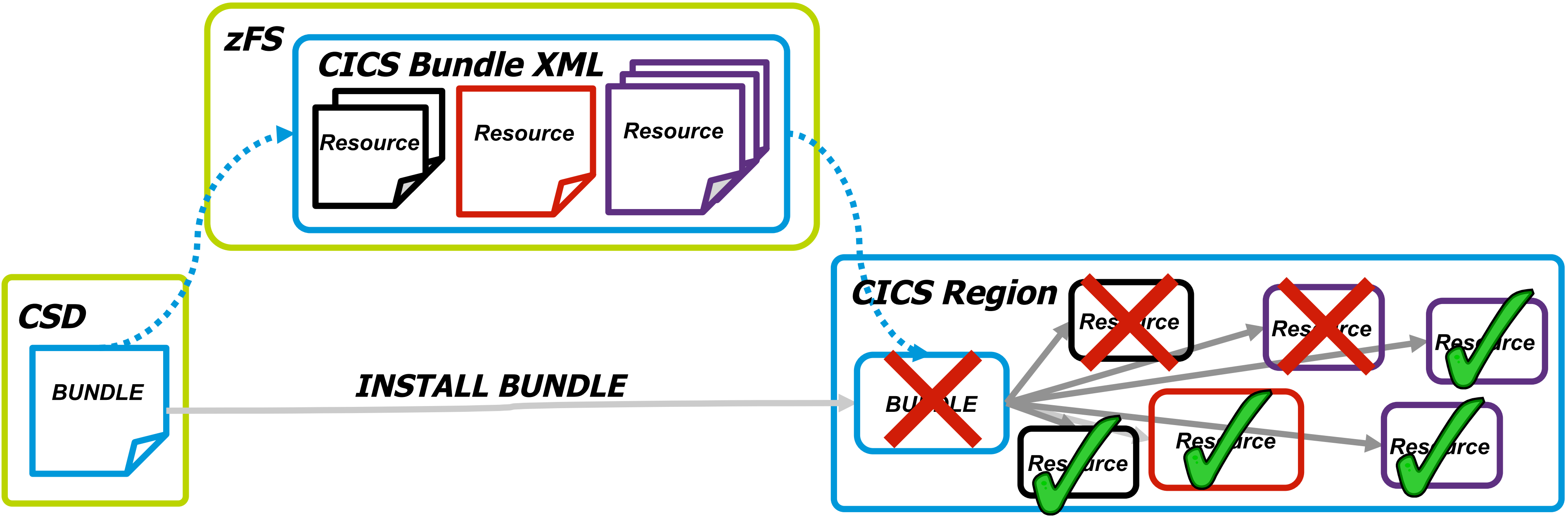
Installing CICS Bundles into CICS

- Create a BUNDLE resource definition which references the CICS Bundle XML in it's BUNDLEDIR attribute
- When the BUNDLE is installed all the resources from CICS Bundle XML are automatically created



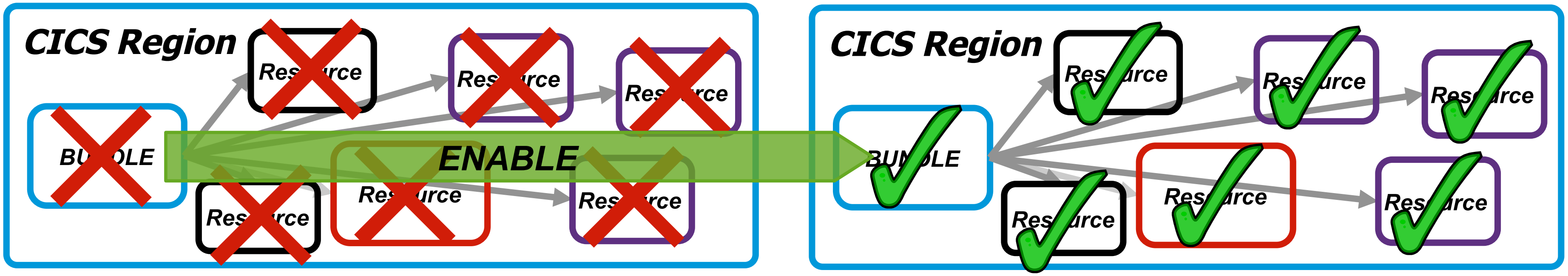
Installing CICS Bundles into CICS

- The CICS Bundle will be implicitly disabled at install if:
  - Any of it's resources fail to install
  - Any of it's resources are installed disabled
  - Any of it's imports are disabled or not present

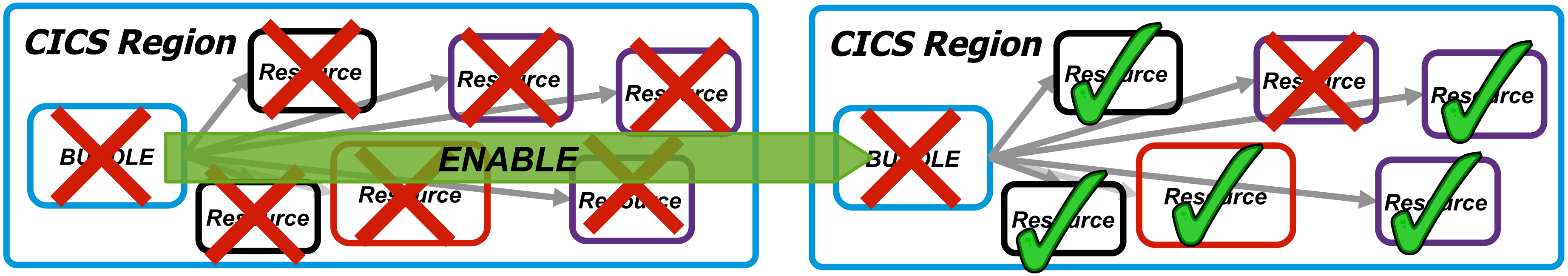


Enabling a CICS Bundle

- When a CICS Bundle is enabled explicitly it will try to enable all it's Resources

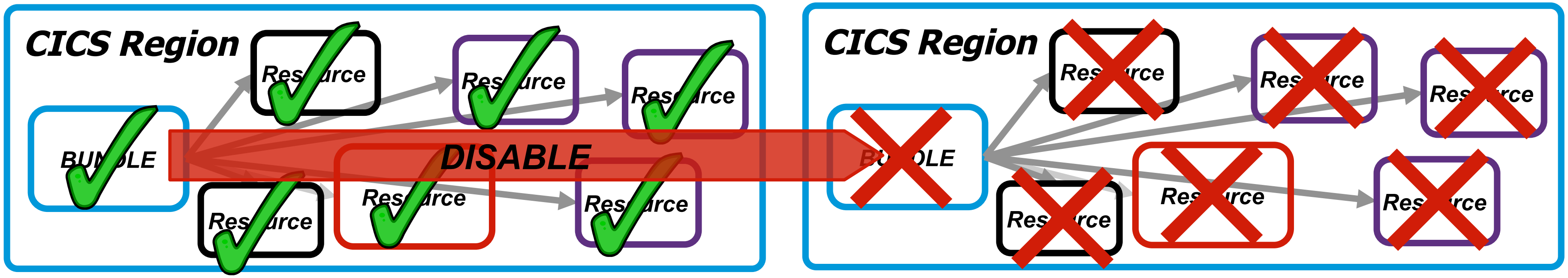


- If there are any resources that cannot be enabled then it will remain disabled

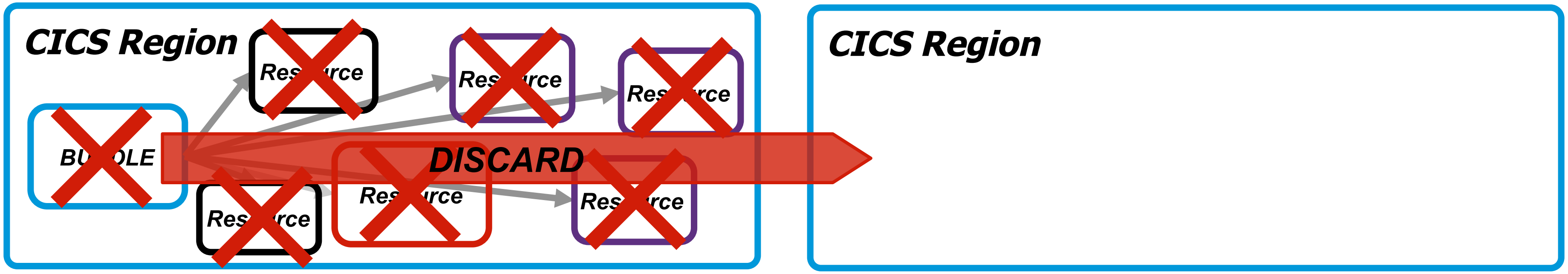


Disabling and Discarding a CICS Bundle

- When a CICS Bundle is disabled explicitly it will disable all it's Resources

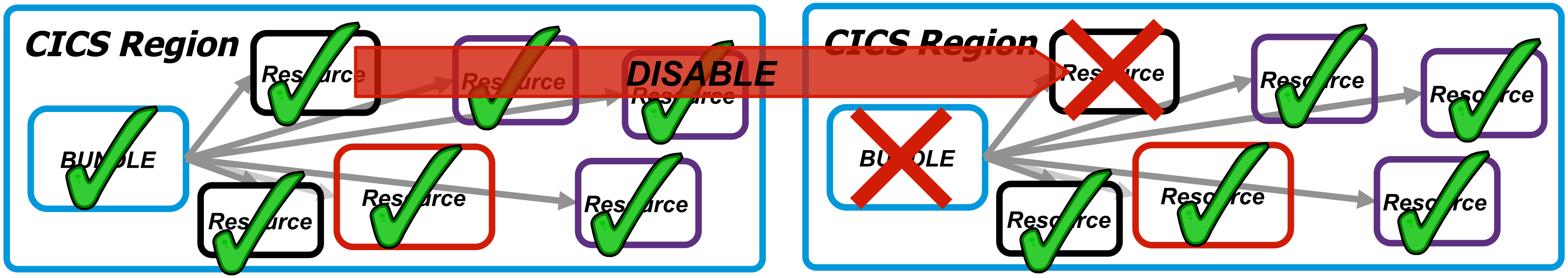


- When a CICS Bundle is discarded it will discard all of it's resources

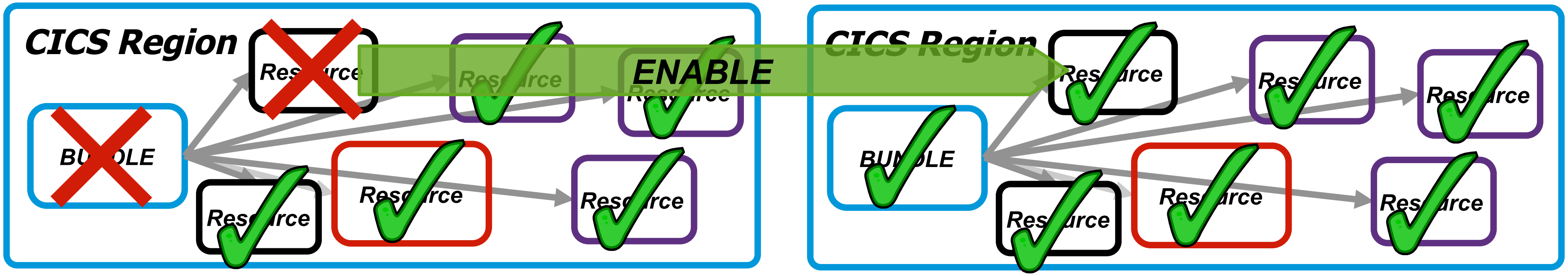


Disabling and Enabling a CICS Resource

- When a CICS Resource is disable, any Bundles which own it or import it will be implicitly disabled



- When a CICS Resource is enabled, any Bundles which own it or import it will be enabled if they were implicitly disabled and this is the last resource stopping it from being enabled



What can you put in a CICS Bundle?

What can you put in a CICS Bundle?

- 2009 - CICS TS V4.1:
  - EVENTBINDING
  - SERVICE via RDz
  - WEBSERVICE via Rdz
  - User resources (create your own!)

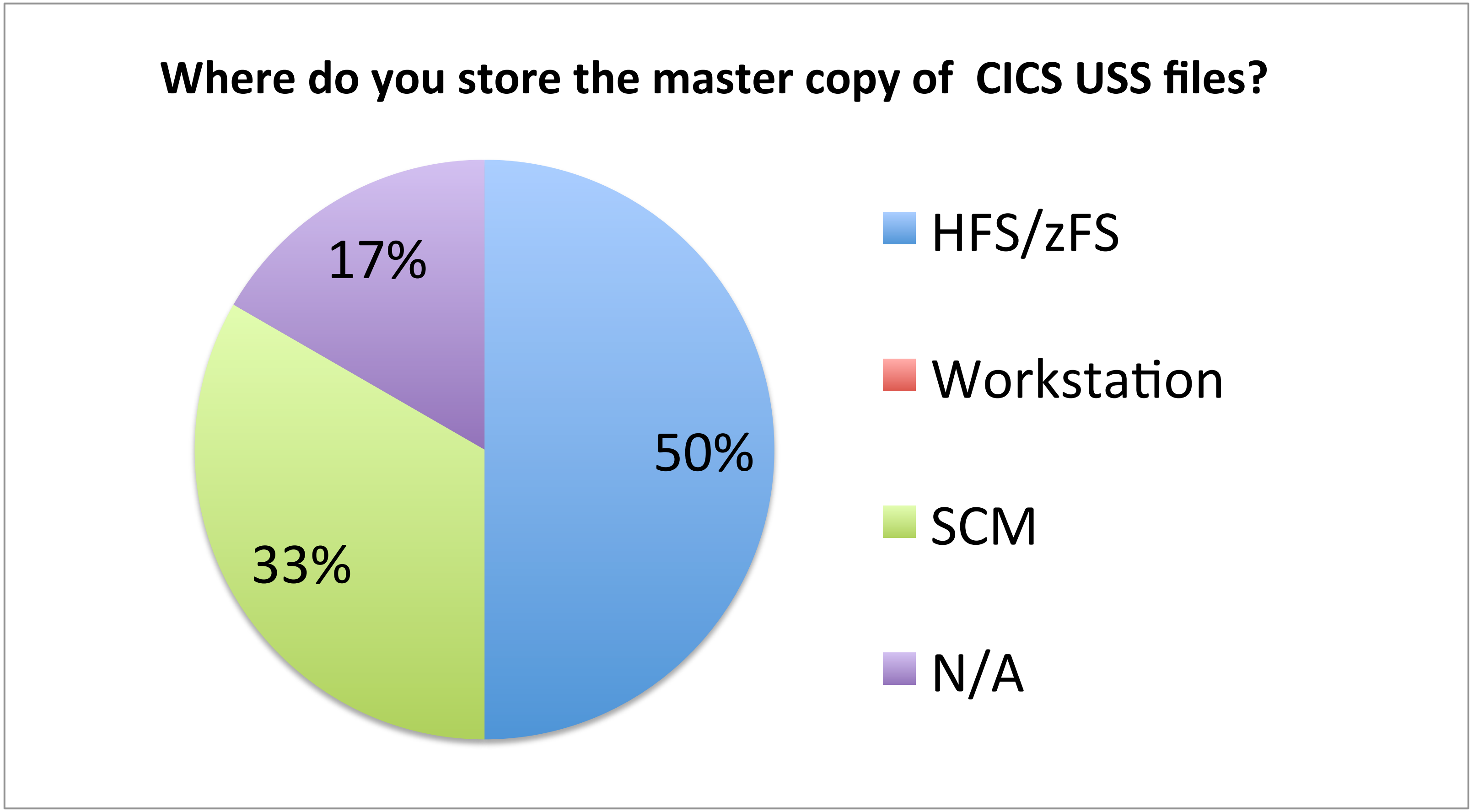
- 2011 - CICS TS V4.2:
  - ATOMSERVICE
  - JVMSERVER
  - Java OSGI Bundles

- 2012 - CICS TS V5.1 Open Beta:
  - PROGRAM
  - ...

Bundles are the future!

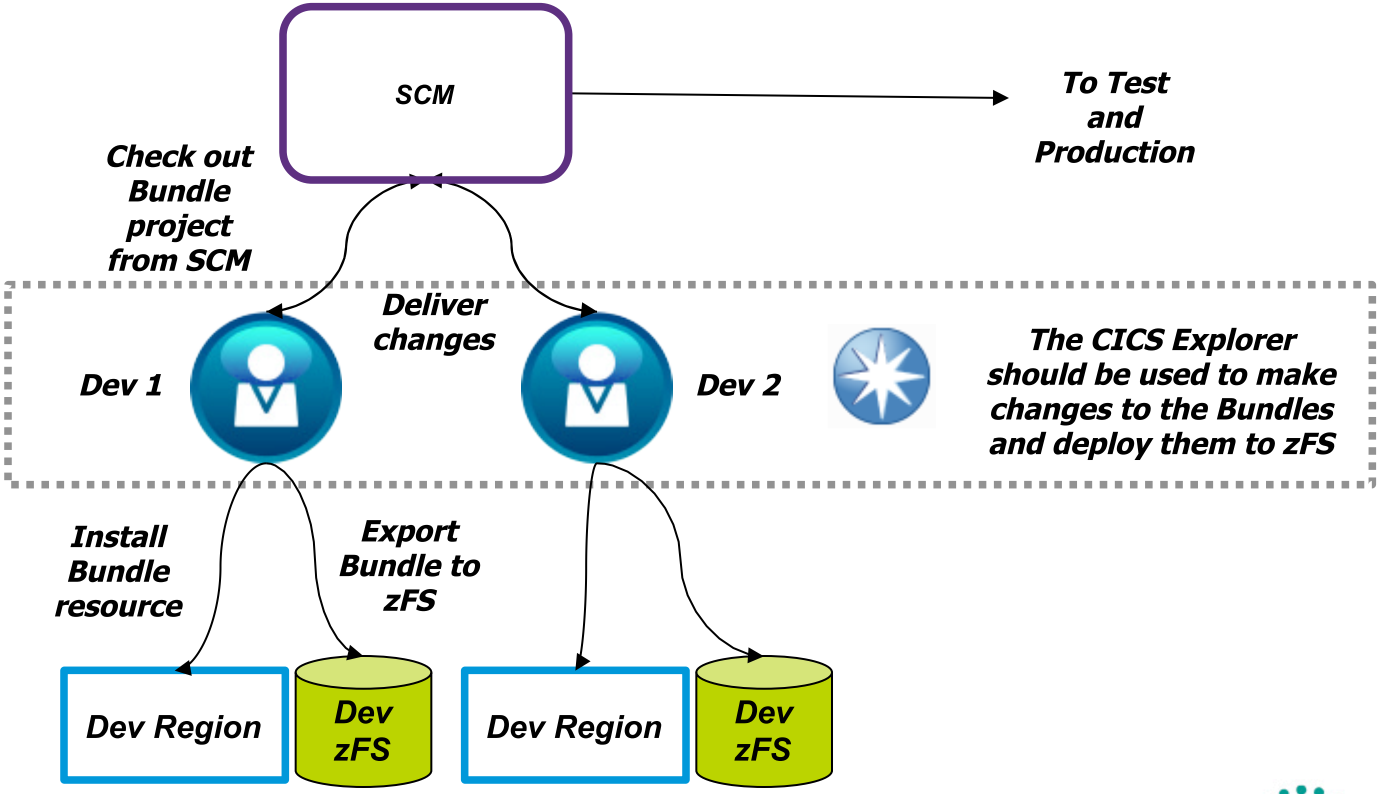
Best practices

zFS usage survey…



Managing changes to CICS Bundles

- CICS Bundle XML should be treated as ***source code***
- Changes should be managed and shared using a source code management (SCM) repository

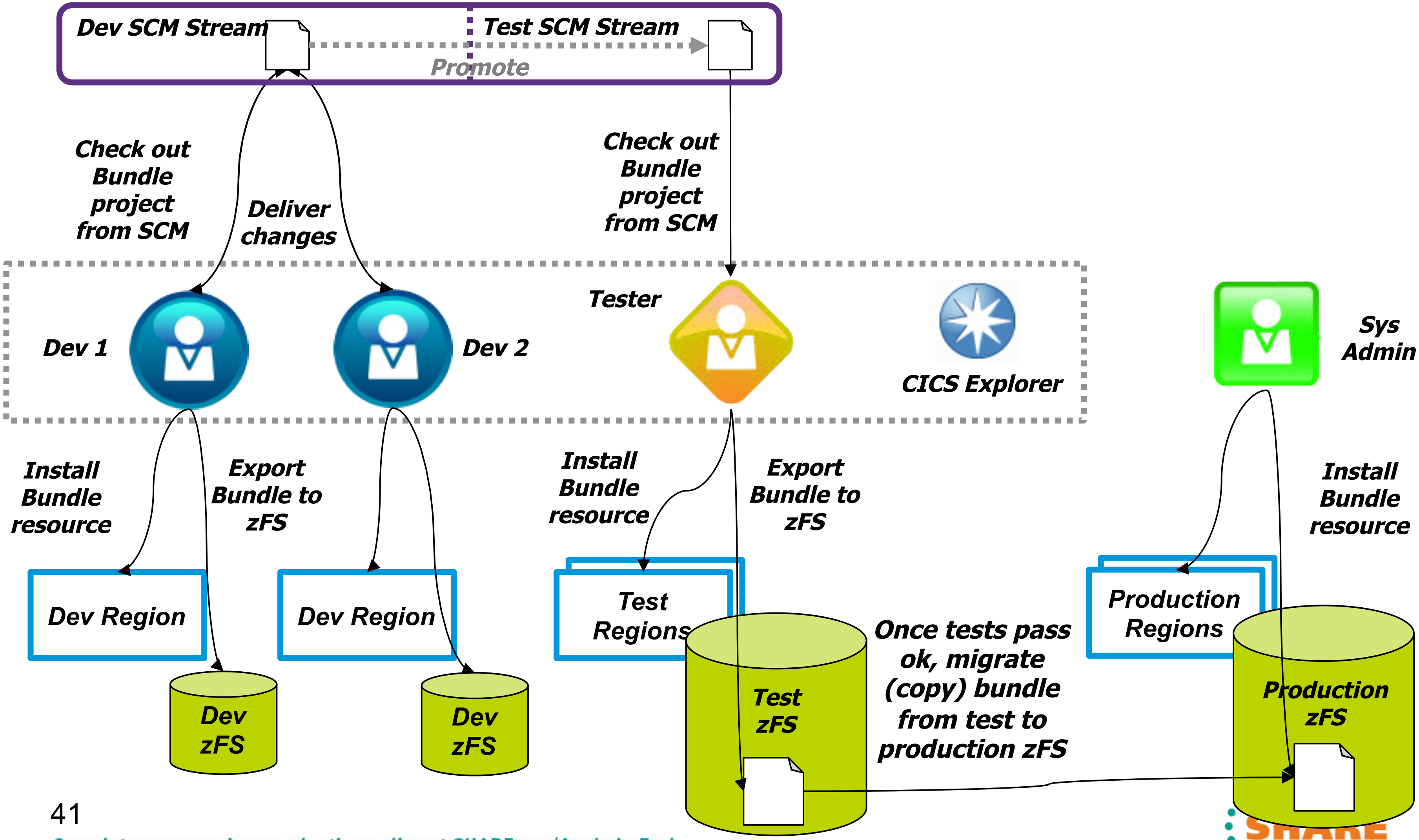


Migrating CICS Bundles from Dev to Test to Production

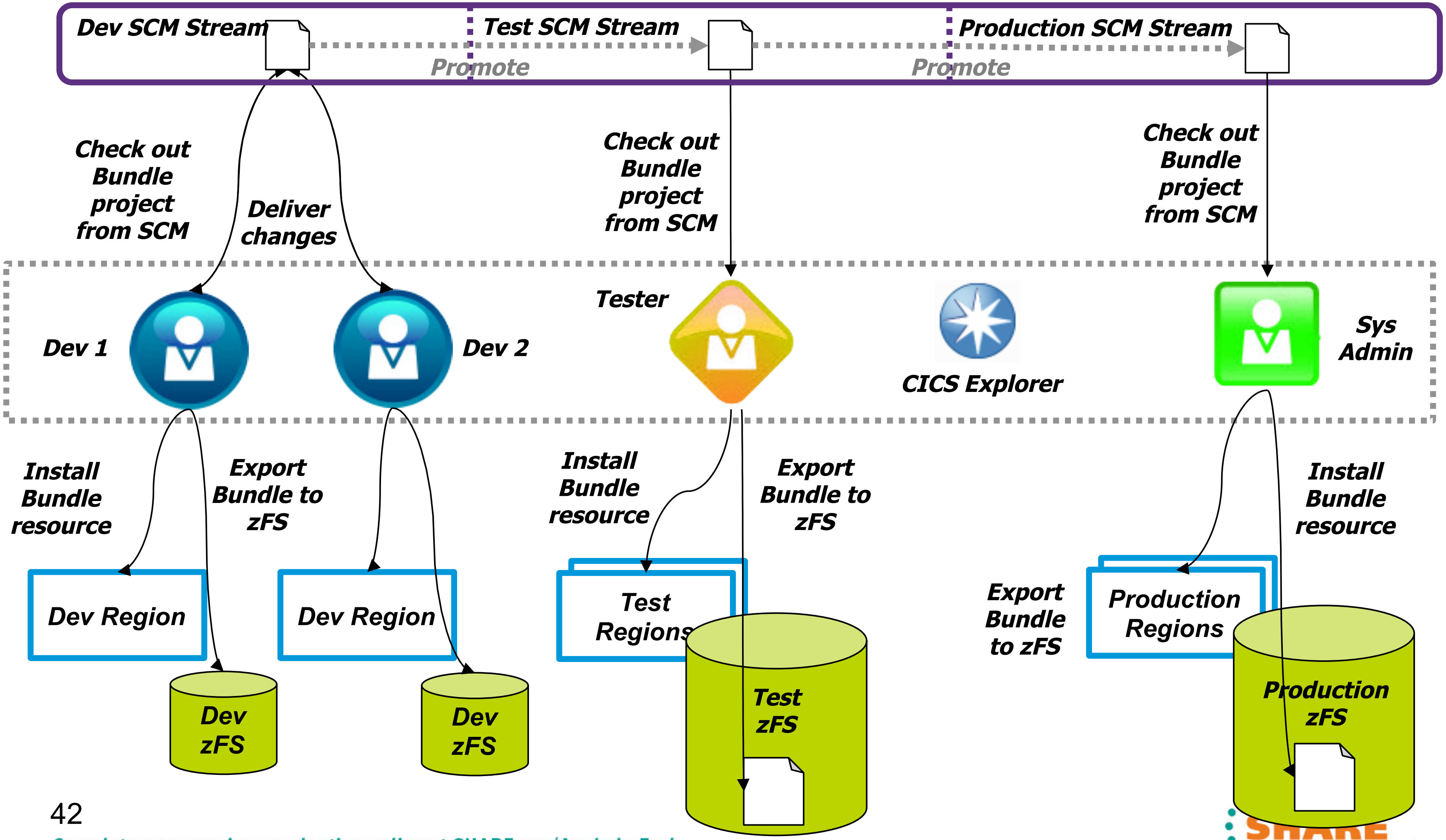
- BUNDLEs should be treated like any other CICS resource that has a reference to an artefact that lies outside the CSD eg:
  - PROGRAMS have load modules/java classes
  - WEBSERVICEs have wsbind files

- You should migrate the CICS Bundle XML ***before*** the BUNDLE resource
  - You wouldn't migrate a new PROGRAM resource before you migrated the load module for it!

Migrating CICS Bundle XML from Dev to Test to Production



Migrating CICS Bundles XML from Dev to Test to Production

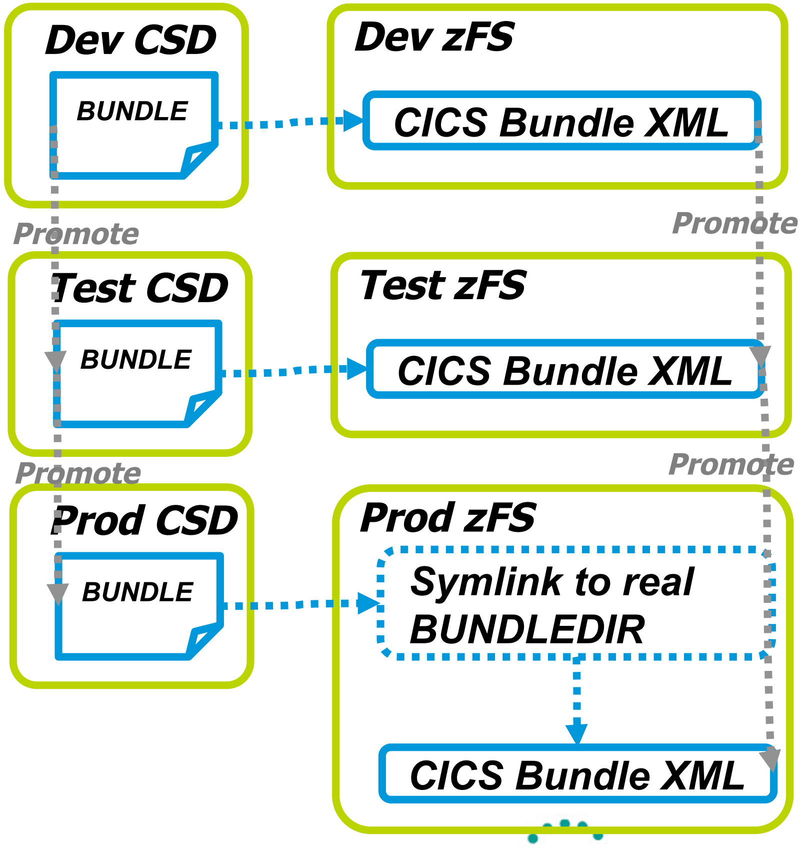


Top Tip 1: Avoid having to change BUNDLEDIR

- Changing the BUNDLE resource's BUNDLEDIR is undesirable because you aren't promoting the same resource that you tested

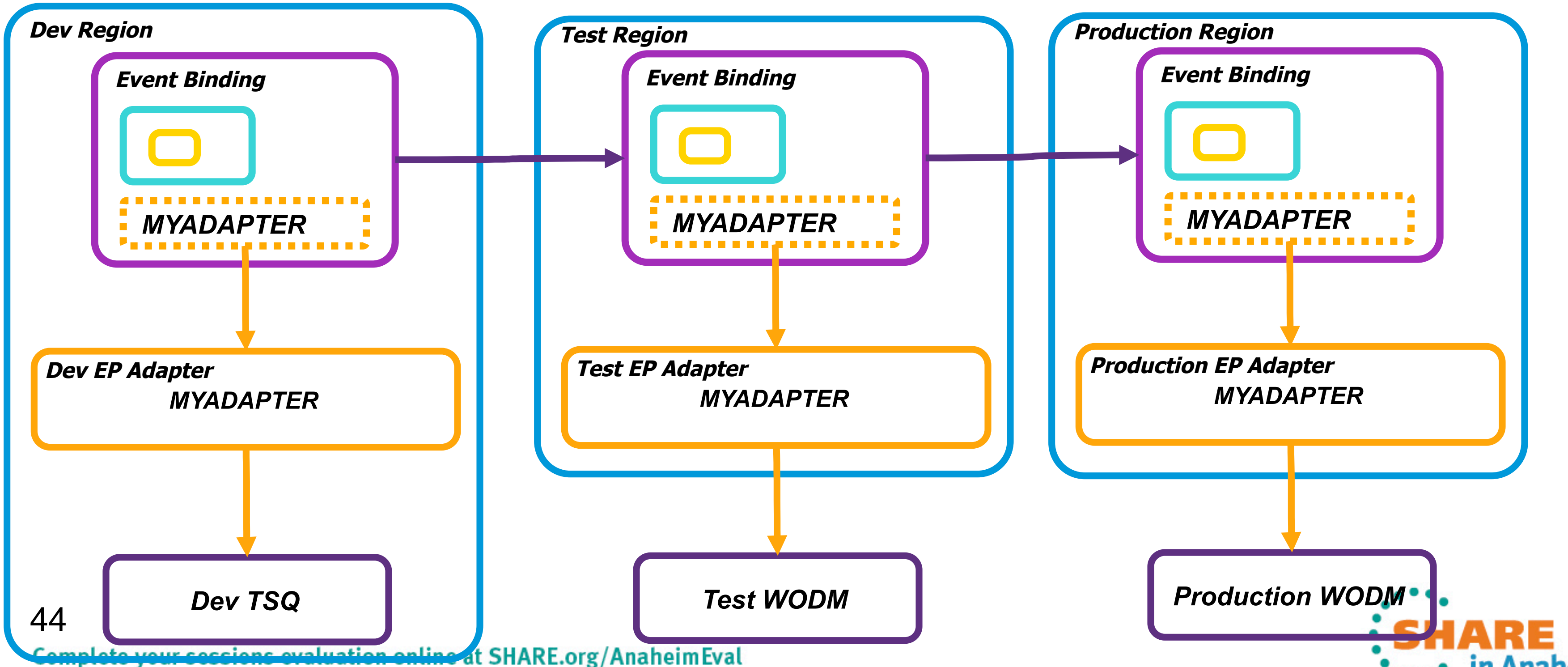
- Option 1: Put your CICS Bundle XML in the same directories on each zFS

- Option 2: Use Symlinks to point to the real bundle location



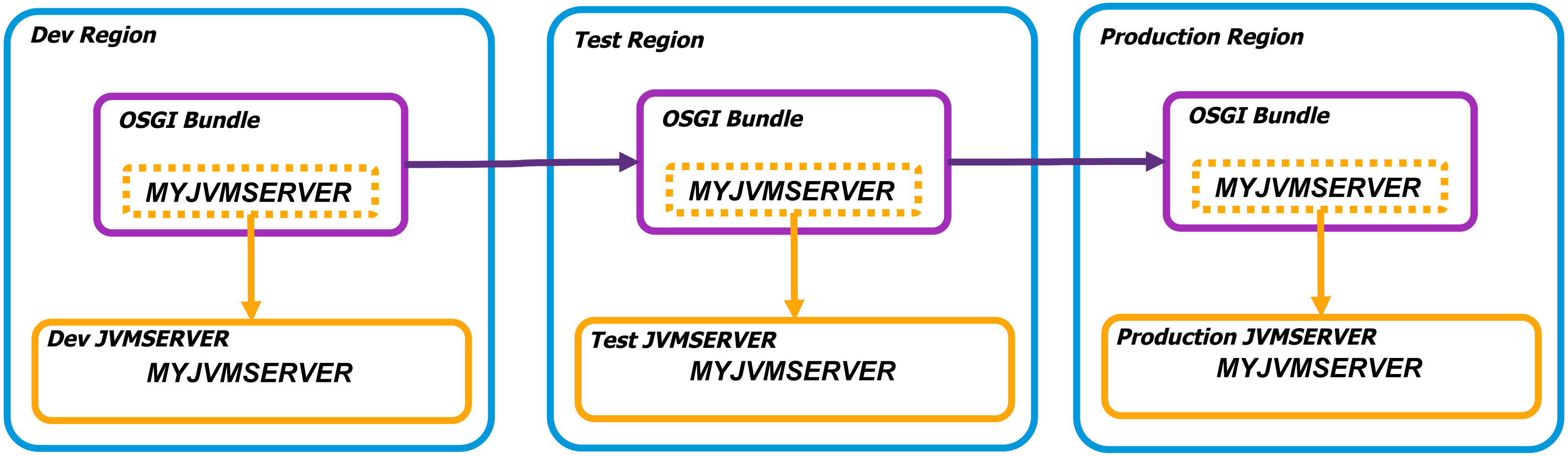
Top Tip 2: Use separate EPADAPTERs on all EVENTBINDINGs

- Use Separate EP Adapters in CICS TS V4.2 to ensure no changes are needed to CICS Bundles containing EVENTBINDINGs during migration



Top Tip 3: Use the same JVMSERVER names in all regions

- For CICS TS V4.2 have the same JVMSERVERs in all regions to ensure no changes are needed to CICS Bundles containing OSGI Bundles during migration



zFS & Security

Security Considerations

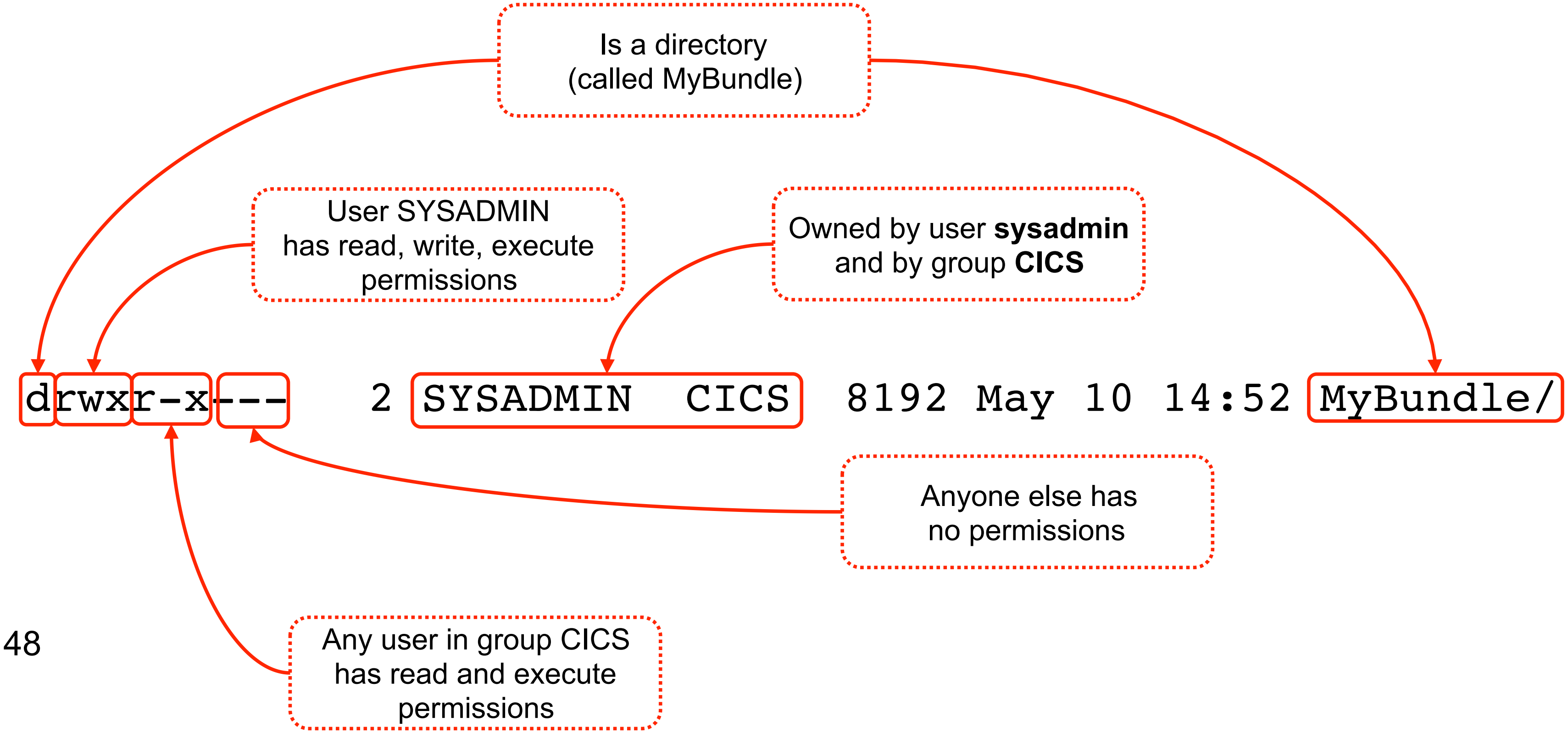
- Use the UNIX permission flags for Owner, Group and All to control access to your CICS resources on zFS

  Here’s an example of an entry you might see if you listed contents of a zFS directory

-
  ```
  drwxr-x---   2 SYSADMIN  CICS  8192 May 10 14:52 MyBundle/
  ```

Security Considerations

- Use the UNIX permission flags for Owner, Group and All to control access to your CICS resources on zFS



-
```
drwxr-x---    2 SYSADMIN   CICS   8192 May 10 14:52 MyBundle/
```

Is a directory (called MyBundle)

User SYSADMIN has read, write, execute permissions

Owned by user **sysadmin** and by group **CICS**

Anyone else has no permissions

Any user in group CICS has read and execute permissions

Security Considerations

- A user can be in many groups, but a file has only one group permission
  - Meaning that if multiple users need to access the file that must be in that group, and will all share the same permissions
  - This means 2 logical groups of users (such as system admins and CICS regions) can not use UNIX group permission bits to be granted access yet be managed as 2 groups

- ACLs provide a solution to this as they allow a more flexible model
  - Multiple groups to a have permissions
  - ACL inheritance to be controlled
  - However, they may only restrict the access permissions that are defined by the UNIX permissions bits

  - ACLs: http://publib.boulder.ibm.com/infocenter/zos/v1r13/index.jsp?topic=/com.ibm.zos.r11.icha700/ichza7a0243.htm

zFS setup - Best Practice

- 1. Create data set for usage as /var/cicsts zFS

- 2a If using shared zFS across a sysplex
  - Mount data set onto root filing system as /cicsts as a r/w filing system
  - On each LPAR create symbolic link to link /var/cicts to /cicsts (/var is always a symlink to /<LPAR>/var)
  - > ln –s /cicsts /var/cicsts

- 2b. If using non-shared zFS,
  - Mount data set onto /var as /var/cicsts

- 3. Set permissions of /var/cicsts to allow access by multiple readers (CICS regions) and a common writer (administrator)
  - 1. Set the owner to have read/write/execute, this will be the userid required by zFS to export files into zFS
  - 2. Set the readers to have read/execute access
  - > chgrp –R <group> /cicsts
  - > chmod –R 750 /cicsts

- 4. Set default file permission for the FTP daemon to give writers(owners) rw and readers(group) r
  - i.e UMASK 027

- 5. If write access is required by multiple groups of writers then you can either
  - 5a Set the group ownership to a common group in which all the writers are members, this will then limit you to running all the readers (CICS regions) under the same uid
  - Set the FTP UMASK to 207, to give write permission to the group.
  - Or 5b. Use ACLs to add additional group permissions. This can be achieved by activating the FSSEC resource class and using the setfacl command

FTP access

- Default FTP file permission are set using a umask (i.e 027 sets 750 ie owner=rwx group=r-x other=---), this is set as a property in the FTP config file
  - see SYS1.TCPPARMS(FTPDATA)

- The USS ftp daemon allows chmod commands to be executed via the site command
- USS FTP does not support chown or chgrp or chmod as commands, however, sftp does support these commands

- The USS guide says you can issue chmod using the site command but not chgrp see http://publib.boulder.ibm.com/infocenter/zos/v1r11/index.jsp?topic=/com.ibm.zos.r11.halu001/site.htm

Performance of zFS

- Performance of shared zFS mounted r/w filesystems has been regarded as an issue (in terms of XCF signalling costs). However:
  - V1R11 provides local read caching – removing overheads for reads
  - V1R13 provides direct I/O for read and write, removing need to function ship these commands to the LPAR owning the file system

Summary