SSM in 15 minutes

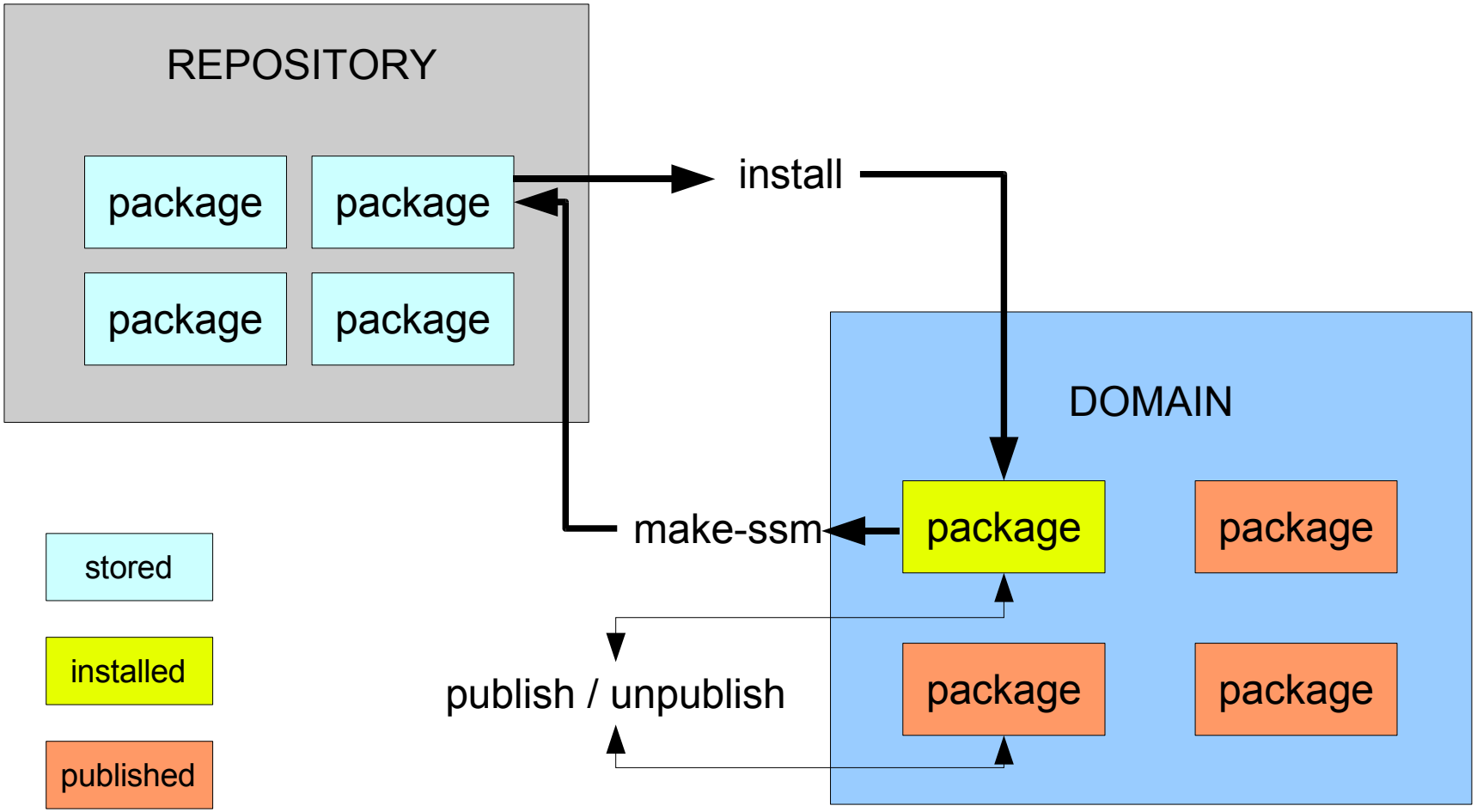
(well, maybe 60)

John Marshall
Michel Valin
HPCS / CIOB

ssm basic element definitions

- Repository
  - A directory tree containing ssm packages for storage (name_version_platform.ssm)
- Domain
  - A directory tree with a special structure containing callable/accessible packages (published/installed)
  - A user who wants to use all the software "published" in a domain (and its subdomains) will subscribe to that domain
- Package
  - A collection of related software pieces (may contain sources, scripts, executables, libraries, data files, …)
  - A package may have two states
    - Installed (it is just present in the domain)
    - Published (it is callable if the user subscribes to the domain)

ssm basic elements



packages

- .ssm files are compressed tar files
- Package name
  - 3 elements separated by an underscore "_"
    name_version_platform (example_1.2.3_multi)
  - name is arbitrary (but MAY NOT contain "_")
  - version is arbitrary (dotted numeric STRONGLY recommended, NO _)
  - platform must be one of the recognized platform names
    - all multi
    - linux2[46]-i[3456]86 linux2[46]-x86-64 linux2[46]-ia64
    - cygwin-i[3456]86 cygwin-x86-64 cygwin-ia64
    - irix65-mips-n32 irix65-mips-64
    - aix5[123]-ppc-32 aix5[123]-ppc-64
      (all means ANY architecture, multi means multiple architectures with architecture management internal to the package)

5 minutes setup

- Make sure that you are using the “standard” environment
- Create your repository
  - either mkdir $HOME/.SsmDepot
  - or mkdir some_path ; ln -s some_path $HOME/.SsmDepot
- Create a domain
  - s.ssm-creat -d path_to_domain
- Start creating packages
  - cd path_to_domain
  - s.ssm-prep -D pkgname_version_platform
  - s.ssm-install -f pkgname_version_platform
- Add to your repository
  - pkgname_version_platform/maint/make-ssm

create a domain

Create the repository

```
 519% mkdir .SsmDepot
 520% ls -al .SsmDepot/
total 12
drwxr-xr-x  2 asphmfv hpcs 4096 Mar  9 21:12 .
drwxr-xr-x 43 asphmfv cidu 8192 Mar 10 13:37 ..
```

Create a domain

```
 521% s.ssm-creat -d /tmp/yyy
will create Domain /tmp/yyy with Repository /home/ib/asph/mfv/.SsmDepot
```

create a domain

```
averroes 522% cd /tmp/yyy
averroes 523% find .
.
./etc
./etc/ssm.d
./etc/ssm.d/platforms
./etc/ssm.d/platforms/IRIX64
./etc/ssm.d/platforms/Linux
./etc/ssm.d/platforms/CYGWIN_NT-5.1
./etc/ssm.d/platforms/AIX
./etc/ssm.d/login
./etc/ssm.d/sources.list
./etc/ssm.d/installed
./etc/ssm.d/version
./etc/ssm.d/profile
./etc/ssm.d/published
./etc/ssm.d/domainHomes
./etc/ssm.d/domainHomes/0
./lib
./lib/ssm.d
./lib/ssm.d/ssm_generateenvconfig.sh
./lib/ssm.d/ssm_lib.sh
./.SsmDepot
```

create a domain

Get into the domain

```
 522% cd /tmp/yyy

 524% ls -al
total 28
drwxr-xr-x  4 asphmfv hpcs  4096 Mar 10 14:35 .
drwxrwxrwt 18 root    root 12288 Mar 10 14:35 ..
-rw-r--r--  1 asphmfv hpcs    28 Mar 10 14:35 .SsmDepot
drwxr-xr-x  3 asphmfv hpcs  4096 Mar 10 14:35 etc
drwxr-xr-x  3 asphmfv hpcs  4096 Mar 10 14:35 lib
```

Repository marker for make-ssm

```
 525% cat .SsmDepot
/home/ib/asph/mfv/.SsmDepot
```

Repository marker for ssm

```
 526% cat ./etc/ssm.d/sources.list
/home/ib/asph/mfv/.SsmDepot
```

create a package directory tree

- s.ssm-prep -D hello_1.0_all
- chmod 755 hello_1.0_all/maint/make-ssm
- find hello_1.0_all
  - hello_1.0_all/
  - hello_1.0_all/etc
  - hello_1.0_all/etc/profile.d
  - hello_1.0_all/maint
  - hello_1.0_all/maint/excludes
  - hello_1.0_all/maint/make-ssm
  - hello_1.0_all/maint/include
  - hello_1.0_all/.ssm.d
  - hello_1.0_all/.ssm.d/control
  - hello_1.0_all/.ssm.d/pre-publish
  - hello_1.0_all/.ssm.d/post-install

create a package directory tree

- chmod 755 hello_1.0_all/maint/make-ssm
  - `516% cat hello_1.0_all/maint/include`
  - `hello_1.0_all/.`
  - `517% cat hello_1.0_all/maint/excludes`
  - `--exclude do_not_tar`
  - `518% cat hello_1.0_all/maint/make-ssm`
  - `#!/bin/ksh`
  - `test -f .SsmDepot && SsmDepot=$(cat .SsmDepot)`
  - `package=${0#./}`
  - `package=${package%%/*}`
  - `echo package=$package`
  - `TarCmd="echo tar"`
  - `tar --help 1>/dev/null 2>/dev/null && TarCmd=tar`
  - `gtar --help 1>/dev/null 2>/dev/null && TarCmd=gtar`
  - `gnutar --help 1>/dev/null 2>/dev/null && TarCmd=gnutar`
  - `set -x`
  - `${TarCmd} $(cat hello_1.0_all/maint/excludes) -zcf ${SsmDepot}/${package}.ssm $(cat hello_1.0_all/maint/include)`

populate a package

- mkdir hello_1.0_all/bin
- vi hello_1.0_all/bin/my_scripts
  - 530% cat hello_1.0_all/bin/my_scripts
  - #!/bin/ksh
  - echo I am $0
  - echo I am truly $(true_path $0)
- chmod 755 hello_1.0_all/bin
  - 531% ssm listr -d .
  - Package Name Location in Repository
  - ------------ ----------------------
  - 532% ssm listd -d .
  - State Package Name Title Domain
  - ----- ------------ ----- ------

create repository image of a package

- create hello_1.0_all.ssm in repository
  -
  -
```
 527%  hello_1.0_all/maint/make-ssm
package=hello_1.0_all
+ cat hello_1.0_all/maint/excludes
+ cat hello_1.0_all/maint/include
+ tar --exclude do_not_tar -zcf /home/ib/asph/mfv/.SsmDepot/hello_1.0_all.ssm
hello_1.0_all/.
```
  -
  -
```
 538% ssm listr -d .
Package Name                    Location in Repository
------------                    ----------------------
hello_1.0_all                   /home/ib/asph/mfv/.SsmDepot/hello_1.0_all.ssm
```

install / publish a package
(dishonest)

- s.ssm-install -f hello_1.0_all -y (-f for a “fake” install)
  - 542% ssm listd -d .
  - State Package Name Title Domain
  - ----- ----------- ----- ------
  - installed hello_1.0_all .
- s.ssm-publish hello_1.0_all -y
  - 545% ssm listd -d .
  - State Package Name Title Domain
  - ----- ----------- ----- ------
  - published hello_1.0_all .
- . s.ssmuse.dot `pwd`
  - 546% . r.ssmuse.dot /tmp/yyy
  - subscribing to domain /tmp/yyy
- my_scripts
  - I am /tmp/yyy/all/bin/my_scripts
  - I am truly /tmp/yyy/hello_1.0_all/bin/my_scripts

install / publish a package
(more honest)

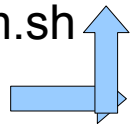
- rm -rf hello_1.0_all
- s.ssm-install hello_1.0_all -y
  - 542% ssm listd -d .
  - State Package Name Title Domain
  - ----- ----------- ----- ------
  - installed hello_1.0_all .
- s.ssm-publish hello_1.0_all -y
  - 545% ssm listd -d .
  - State Package Name Title Domain
  - ----- ----------- ----- ------
  - published hello_1.0_all .
- . s.ssmuse.dot `pwd`
  - 546%. r.ssmuse.dot /tmp/yyy
  - subscribing to domain /tmp/yyy
- my_scripts
  - I am /tmp/yyy/all/bin/my_scripts
  - I am truly /tmp/yyy/hello_1.0_all/bin/my_scripts

publishing a package

- What always gets published
  - bin
  - lib
  - man
  - sbin
  - share
    (if these subdirectories exist in package)
  - the package initialization script
- Where does it get published
  - Under the proper architecture tree
    e.g. demo_1.0_all in domain /tmp/yyy
    gets published under
    /tmp/yyy/all

publishing a package

- What happens when user subscribes to domain
  - environment variables get adjusted
    - PATH
    - MANPATH
    - LD_LIBRARY_PATH (LIBPATH)
    - PYTHONPATH
    - TCL_LIBRARY
  - package initialization scripts are sourced
    - domain/pkg_ver_platform/etc/profile.d/pkg_ver_platform.sh
    - domain/platform/etc/profile.d/pkg_ver_platform.sh (link)



advanced topics

- post-install
  - pkg_ver_plat/.ssm.d/post-install
  - executed after installing
- pre-publish
  - pkg_ver_plat/.ssm.d/pre-publish
  - executed before publishing
- post-publish
  - pkg_ver_plat/.ssm.d/post-publish
  - executed after publishing
- Two arguments are passed
  - $1 = path_to_the_domain
  - $2 = path_to_the_package

advanced topics

- pre-uninstall
  - pkg_ver_plat/.ssm.d/post-install
  - executed before uninstalling
- pre-unpublish
  - pkg_ver_plat/.ssm.d/pre-publish
  - executed before unpublishing
- post-unpublish
  - pkg_ver_plat/.ssm.d/post-publish
  - executed after unpublishing
- Two arguments are passed
  - $1 = path_to_the_domain
  - $2 = path_to_the_package

pre/post scripts from r.ssm-prep

- pre-canned standard recipes
- post-install and pre-publish are created (do nothing templates)
  - `domainHome=$1`
  - `packageHome=$2`
  - `profileDirPath=${packageHome}/etc/profile.d`
  - `packageName=${packageHome##*/}` `(get rid of version and platform)`
  - `profilePath=${profileDirPath}/${packageName}.sh`
  - `mkdir -p ${profileDirPath}`
  - `#echo "#${packageName%%_*}_HOME=${packageHome}" >> ${profilePath}`
  - `(prepare to create the initializer script for the package)`

why pre/post scripts ?

- compile source code after installing it
- compile source code before publishing
- configure initializer script of package with information coming from the actual path to the package
- touchup before package gets published
  - moving data
  - creating internal links
- final touchup after package has been published
- cleanup before uninstalling

pre-publish example

```
domainHome=$1
packageHome=$2
profileDirPath=${packageHome}/etc/profile.d
packageName=${packageHome##*/}
profilePath=${profileDirPath}/${packageName}.sh
mkdir -p ${profileDirPath}
#echo "#${packageName%%_*}_HOME=${packageHome}" >> ${profilePath}
#
Platform=`uname -s` ; [[ $Platform = Linux ]] \
 && Platform=${Platform}-`uname -m`
#
profilePath=${packageHome}/bin/s.set_profile_hostbin.dot
chmod 644 ${profilePath}
echo 'Platform=`uname -s` ; [[ $Platform = Linux ]] \
 && Platform=${Platform}-`uname -m`' >${profilePath}
echo "PATH=${packageHome}/hostbin/\${Platform}:\${PATH}" >> ${profilePath}
chmod 644 ${profilePath}
cd ${packageHome}/src
mkdir -p ../hostbin/${Platform}
EC_ARCH=${Platform} make
```

suggestions ?

Thank you for your attention