

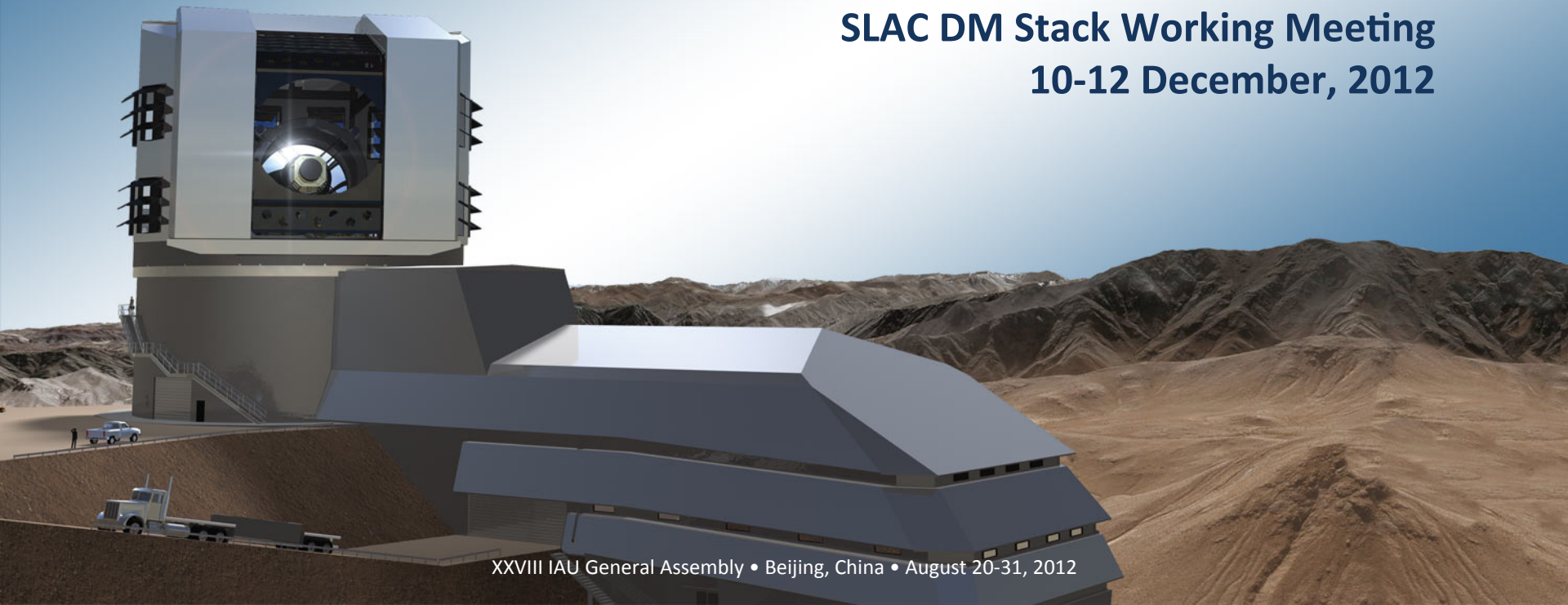


Package Management with EUPS: A Crash Course

Mario Jurić

LSST Data Management Project Scientist

SLAC DM Stack Working Meeting
10-12 December, 2012





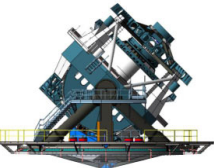
Topics Covered



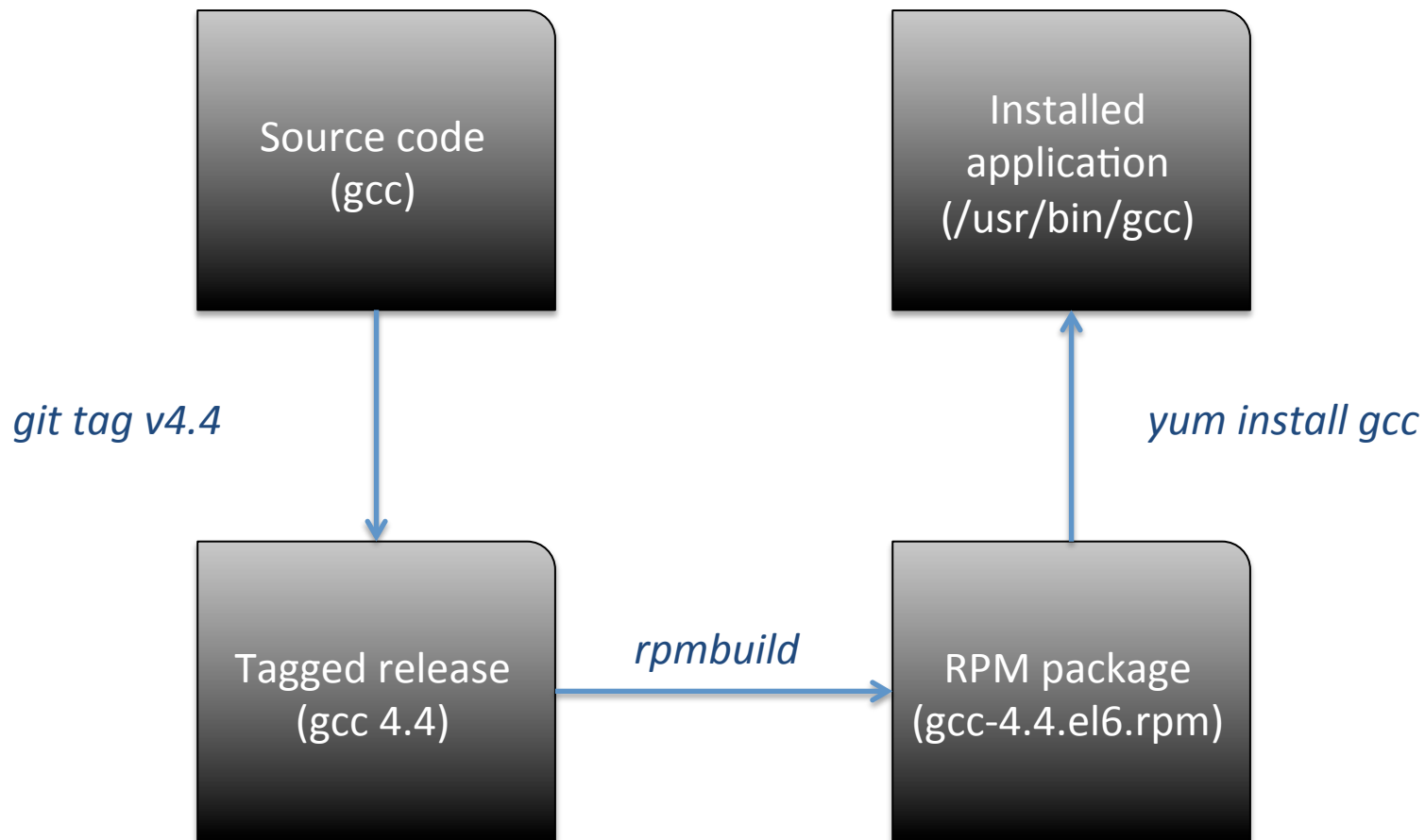
- What is EUPS and its role in DM
- Package management (like `modules`):
 - How to use it to “setup” (select/activate) packages
 - How to work on a package and build it from source
 - What EUPS package versions mean
- Package distribution (like `yum`):
 - How to use it to install DM packages

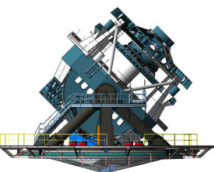


- We use EUPS as the package manager
- Think of EUPS as a combination of `rpmbuild/rpm/yum` and `modules`
- Like `modules`:
 - Multiple versions of a package can be installed at the same time (unlike with `rpm`)
 - A version to work with can be selected at runtime
 - `setup/unsetup` commands
- Like `rpmbuild/rpm/yum`:
 - Allows one to package a version of an application for others to download and install
 - Tracks dependencies and automatically resolves them when installing packages

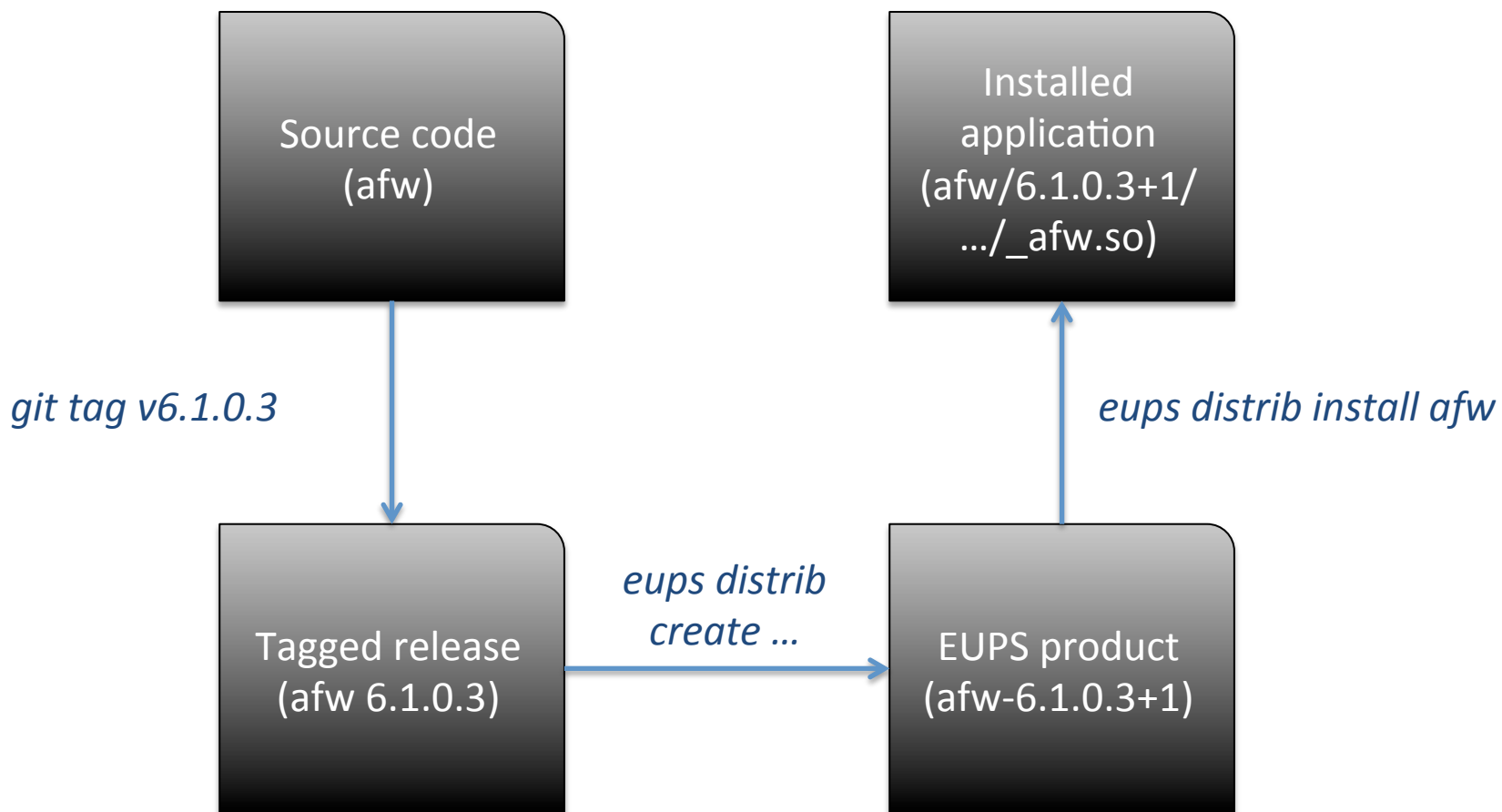


Analogy: RPM-based Linux Distros





Analogy: EUPS-based Distributions



- <http://dev.lsstcorp.org/trac/wiki/Installing/Winter2013>

```
# note: this is for bash users
cd root/directory/where/lsst/stack/will/be/installed (e.g., ~/lsst)
unset LSST_HOME EUPS_PATH LSST_DEVEL

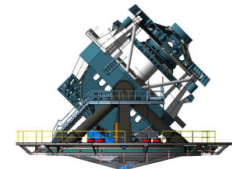
# figure out how many cores are present on the system
export NCORES=$((sysctl -n hw.ncpu || (test -r /proc/cpuinfo && grep
processor /proc/cpuinfo | wc -l) || echo 2) 2>/dev/null)
export MAKEFLAGS="-j $NCORES"
export SCONSFLAGS="-j $NCORES"
```

↙ Installs EUPS and all its prerequisites

```
# download the script that installs the basic environment
curl -O http://dev.lsstcorp.org/pkgs/std/w12/newinstall.sh
bash newinstall.sh
```

Use EUPS to install a top-level DM Stack package. This package will pull its prerequisites

```
# install the stack up to pipe_tasks
source loadLSST.sh
eups distrib install --nolocks -t v6_1 pipe_tasks
```



```
# load the environment  
source loadLSST.sh
```

← Adds EUPS to the path (and sets a few other useful environment variables)

“Activate” the version of pipe_tasks with ‘Summer2012’ tag. Activate obs_sdss. Activate
↙ astrometry_net_data, version ‘sdss-2012-05-01-0’.

```
# activate (‘setup’) the top-level packages and astrometry data  
setup pipe_tasks -t Summer2012  
setup obs_sdss  
setup astrometry_net_data sdss-2012-05-01-0
```

Run a command (made available by pipe_tasks). It will use the
↙ currently setup-ed obs_sdss and astrometry_net_data.

```
# run!  
processCcdSdss.py sdss /lsst7/stripe82/dr7/runs --id run=1033  
camcol=2 field=111 filter=g --output /nfs/lsst7/stripe82/dr7-coadds/  
v1/run2
```



- **setup pipe_tasks 6.0.1.3+1**
 - Find installed package pipe_tasks
 - Find version 6.0.1.3+1
 - If we hadn't specified a version, EUPS would by default look for the one tagged 'current'
 - Get the list of packages that pipe_tasks depends on (e.g., libraries or packages it needs to run)
 - ➔ For each library in list, run setup recursively
 - Set/add value to/remove value from environment variables, as specified by the package's "table file".
 - Typically, add .../bin to PATH, .../lib to LD_LIBRARY_PATH, etc.
 - Set <package_name>_DIR environment variable to point to the package's directory
 - PIPE_TASKS_DIR=.../pipe_tasks

- **unsetup pipe_tasks**
 - Basically, run the above in reverse.

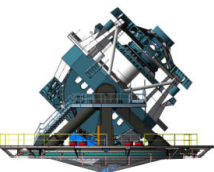


- In each package's directory, there's a subdirectory named 'ups', containing (among other things) a <packagename>.table file
- Example:

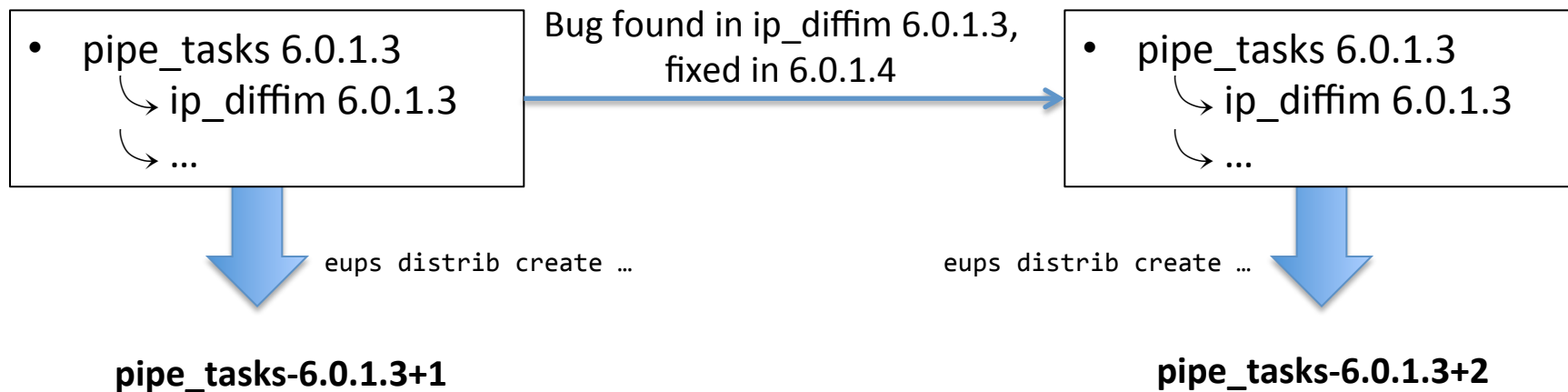
```
mjuric@gamont:pipe_tasks$ cat ups/pipe_tasks.table
setupRequired(pipe_base >= 5.0.0.1)
setupRequired(pex_config)
setupOptional(afw >= 5.0.1.0)
...
setupOptional(matplotlib)

# Extra algorithms
setupOptional(meas_extensions_shapeHSM)
setupOptional(meas_extensions_photometryKron)
setupOptional(meas_extensions_rotAngle)

envPrepend(PYTHONPATH, ${PRODUCT_DIR}/python)
envPrepend(PATH, ${PRODUCT_DIR}/bin)
```



- An EUPS package knows what packages it has been built against. Good for provenance, debugging, and repeatability. Leads to a slight complication in versioning.
- Example:
 - pipe_tasks depends on ip_diffim (among others). How do we differentiate between the same source code of pipe_tasks, built against different sources of ip_diffim?





1. setup some version of the stack (or needed libraries)
2. clone the source code and check out the desired branch of the package you plan to work on
3. setup that source-code package, using the ``setup -j -r <directory>`` syntax
 - Typically, ``setup -r .``, executed from the package's base directory
 - `-j` flag tells EUPS to just set up this package (don't resolve dependencies)
4. Edit code
5. Build the package using `scons`
6. Run (no need to install)



A frequent use case

- Overriding an installed package with a development version (e.g., to test a bugfix)

```
# load the basic environment
source loadLSST.sh

# activate ('setup') the top-level package, and astrometry data
setup pipe_tasks -t Summer2012
setup astrometry_net_data sdss-2012-05-01-0 -k

# get and build the source of obs_sdss
git clone git://dev.lsstcorp.org/LSST/DMS/obs_sdss.git
cd obs_sdss
git checkout tickets/2212
setup -j -r .
scons opt=3 -j 8

# run!
processCcdSdss.py sdss /lsst7/stripe82/dr7/runs --id run=1033
camcol=2 field=111 filter=g --output /nfs/lsst7/stripe82/dr7-coadds/
v1/run2
```



- Locking
 - EUPS currently has a serious bug with locking. The workaround is to disable locking using the `--nolocks` flag.
 - Example:
 - `eups distrib install --nolocks pipe_tasks`
- Developing using an IDE (e.g., Eclipse)
 - Probably could be done with properly crafted Makefiles
 - No one on the DM team uses it so we don't know for sure



- WARNING: There's little up-to-date documentation (including the one included with EUPS source!).
- <http://dev.lsstcorp.org/trac/wiki/EupsTips>
 - Note: there are many other EUPS pages on the wiki, and nearly all of them are out of date! If in doubt, ask us.
- <https://github.com/RobertLuptonTheGood/eups>
 - If in doubt, read the source
- E-mail lsst-dm-stack-users@lsstcorp.org