

# A Question Of Binaries

TIM JENNESS

The logo for the Large Synoptic Survey Telescope (LSST). The letters 'LSST' are rendered in a bold, black, sans-serif font. The letter 'S' is filled with a blue-to-white gradient, giving it a three-dimensional, glowing appearance. The letters are outlined in white.

*Large Synoptic Survey Telescope*

# Why Binary Distributions?



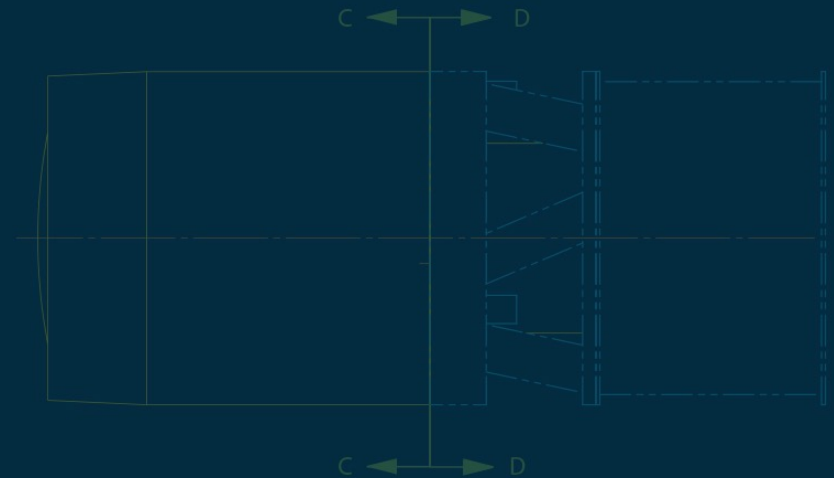
- Can take 1 to 2 hours to build `lsst_apps` if something low down in the stack has been updated (afw can take 20 minutes or more on its own).
- `lsstsw` build system is great for picking up these changes but will always try to rebuild if something has changed upstream.
- “eups distrib install” from source can work for weekly tagged releases but (currently) will likely trigger a long wait for the build and you could still be out of date.
- Binary distributions can significantly simplify this.

# Who Are The Customers?



- DM Developers?
- Calibration team?
- Camera team?
- Sims team?
- Sims users?
- Subaru PFS?
- Interested external users?

How much support do we give Sims?



# Binaries



- Different approaches are available:
  - conda, home-brew, CernVMFS, RPMs, EUPS distrib, PyPI wheels, Docker+EUPS
- Conda is easiest for the general user but has significant cost within DM to get right and support (the conda build environment is a moving target) and the user still has to use EUPS.
- Our non-standard build system (Scons+EUPS) means real development work needed to use RPMs, home-brew or PyPI (possible but who is paying?).
- CernVMFS is an option (IN2P3 distribute the stack this way) but a new system for people to learn.
- Current plan:
  - Attempting to try EUPS binary installs (this may need extra work on macOS).
  - Camera require Docker (currently building their own from Conda releases).
  - Aim to make automated releases from any Jenkins build.

