Fermilab

Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

LArSoft/LArLite Integration Plan

Christopher Jones & Marc Paterno

LArSoft Coordination Meeting

24 November 2015

LArSoft/LArLite Interoperability Project

- Purpose of project
  - Allow interoperability between LArSoft and µBooNE's LArLite
    - allow development of LArSoft algorithms within LArLite
    - allow reading of LArSoft ROOT data files in LArLite
- Original proposal
  - https://cdcvs.fnal.gov/redmine/attachments/download/30799/LArSoftLArLiteInteroperabilityProposal.pdf
- Go ahead for project
  - June 26th LArSoft Steering Group Meeting
- Project website
  - https://cdcvs.fnal.gov/redmine/projects/sketch

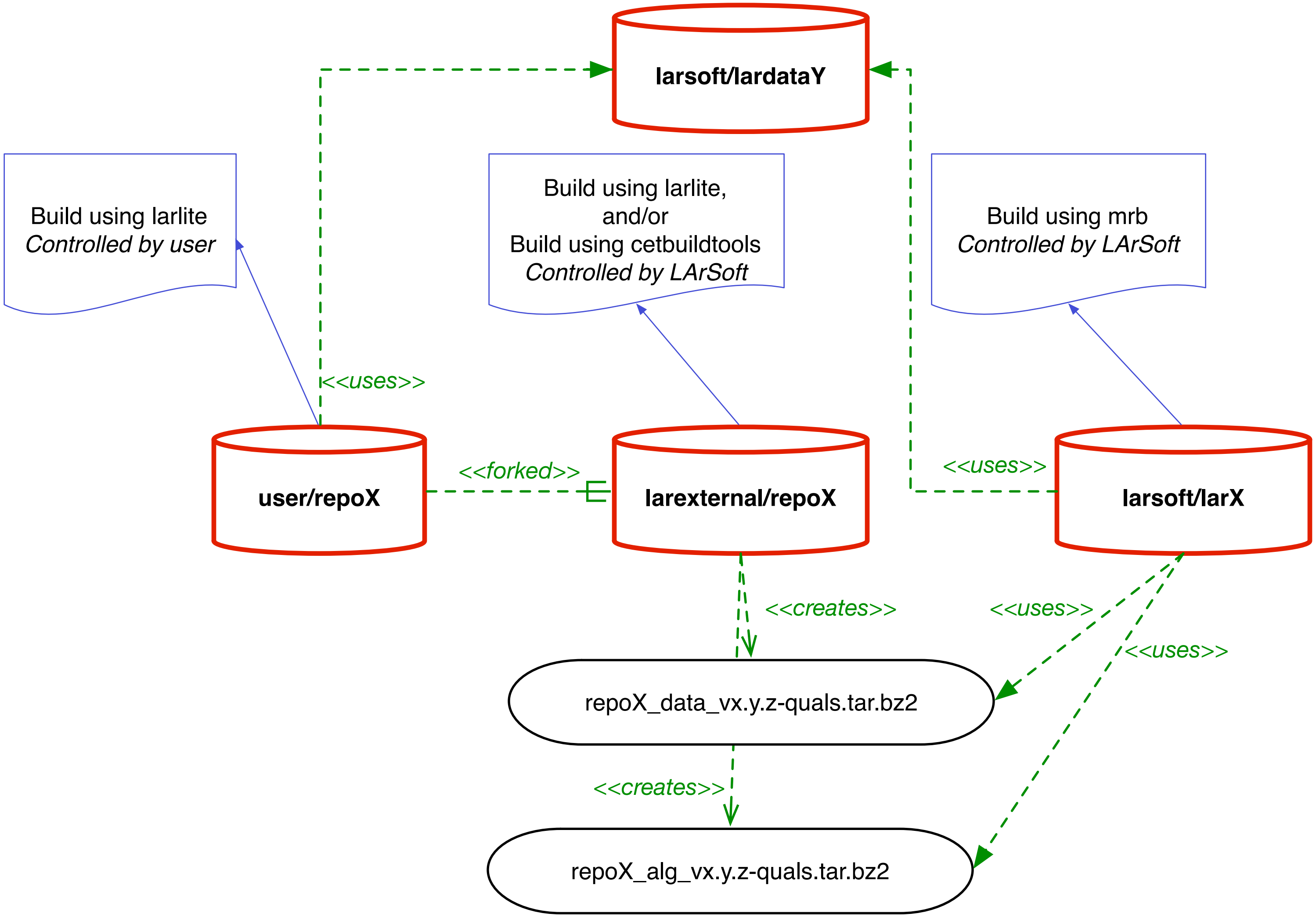
What is LArLite?

- Build Mechanism
  - Creates skeleton of a development area
    - directory structure
    - Makefiles
- Modular event loop driver
  - Provides ‘standard’ data products
    - some are copies of data products from LArSoft
  - Simple framework for driving user written modules over events
  - Has its own data file format

Project Plan

- Split up UPS products in art, nutools and LArSoft
  - Data classes moved to new UPS products
    - includes necessary support code
    - new UPS products have reduced dependencies
  - Original UPS products will depend on the new UPS products
  - This allows LArLite to read the LArSoft data files with minimal dependencies
- Make it possible to read LArSoft *art*/ROOT files outside of *art*
  - LArLite can then use this facility in its own event loop driver
- Create three demonstrators illustrating interoperability
  - Documentation about the demonstrator can then be used by developers on their own projects

Code Repository Relationships



Demonstrators: Three Use Cases

- Developer uses LArSoft data products in LArLite
  - This excludes using LArSoft **algorithms** in LArLite
  - Not all LArSoft data products have to work in LArLite
- Developer uses LArLite to create/maintain LArSoft algorithm
  - Developers can use their original repository for maintenance
  - LArSoft can take changes from developer on your own schedule
    - git repositories handle this gracefully
    - can version code independent of the developer
    - LArSoft can support the code independent of original developer
- Scientist uses LArSoft *art*/ROOT file outside of *art*
  - Plot histograms using ROOT macros or python

Progress

- Initial demonstration of splitting of LArSoft, nutools and art nearly done
- Ability to read LArSoft art/ROOT data files
  - Simple data products all work
  - Work ongoing to support art::Ptr
- Necessary changes to LArLite data classes
  - Demonstrator code given to LArLite developers
  - Changes were accepted by LArLite developers
- Full status of project available at:
  - https://cdcvs.fnal.gov/redmine/projects/sketch/issues