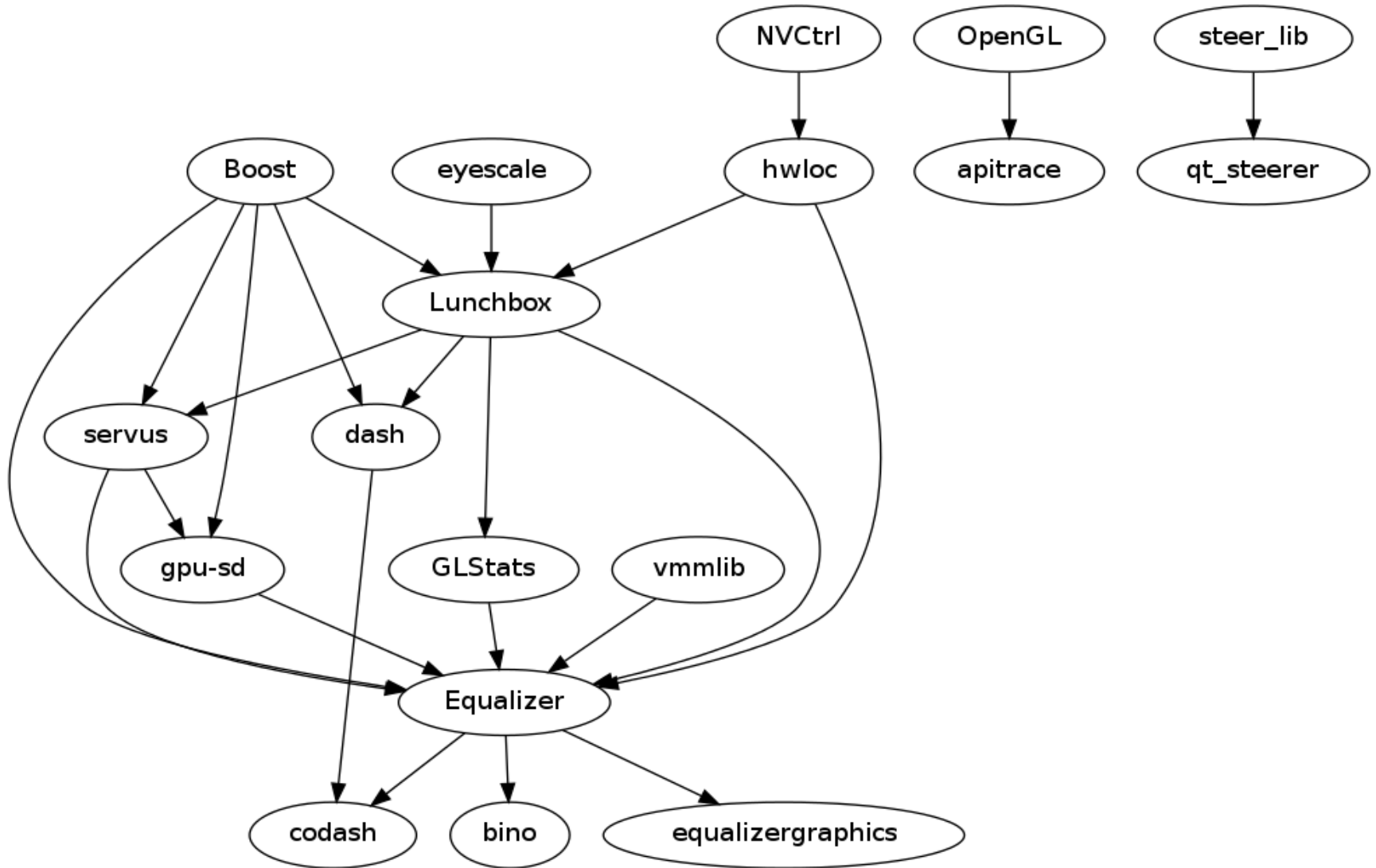


- What is this?
- Motivation
- First example
- How does it work?
- Future

What is Buildyard?

- Build environment based on CMake
- Facilitate build of multiple dependent projects
- Dependencies can be installed packages or compiled from git or svn repositories
- Extensible to user's need
- Buildyard repository:
<https://github.com/BlueBrain/Buildyard>



- Goal: 'make <Project>' to build your code
- Automatic checkout, update, build and install of dependent projects in the correct order
- Comfortable daily development, consolidate code at one location
- Developer environment: implement, test, commit, ...

- Foundation is ExternalProject.cmake, shipped with every CMake installation
- Buildyard enhances this with UseExternal.cmake
- Generates Makefile in project folder: forwards args to Buildyard make
- Reads build configurations from config folders
- config/ default shipped, config.local/ optional for overrides

- config variables
 - `NAME_VERSION`: required version for `find_package`
 - `NAME_REPO_URL`: URL to source code
 - `NAME_REPO_TYPE`: **git**, git-svn, svn
 - `NAME_REPO_TAG`: svn revision, git tag
 - `NAME_DEPENDS`: list of dependencies
 - `NAME_TAIL_REVISION`: oldest svn rev for git-svn clone
- Local overrides (config.local folder always used as last)
 - Any of the above
 - `NAME_USER_URL`: new git remote as origin

- Buildyard make targets
 - clobber: remove build folders
 - clean: clean all projects
 - all: build debug & release
 - {Project}: build & install + dependencies
- make {Project}-target, make targets
 - download: git clone, git-svn clone, svn checkout
 - update: git pull, git-svn rebase, svn update
 - configure: cmake run
 - build: make
 - install, test, package, doxygen

- GNU modules: make {Project}-module
- Release procedure support: tag, branch, package, ...