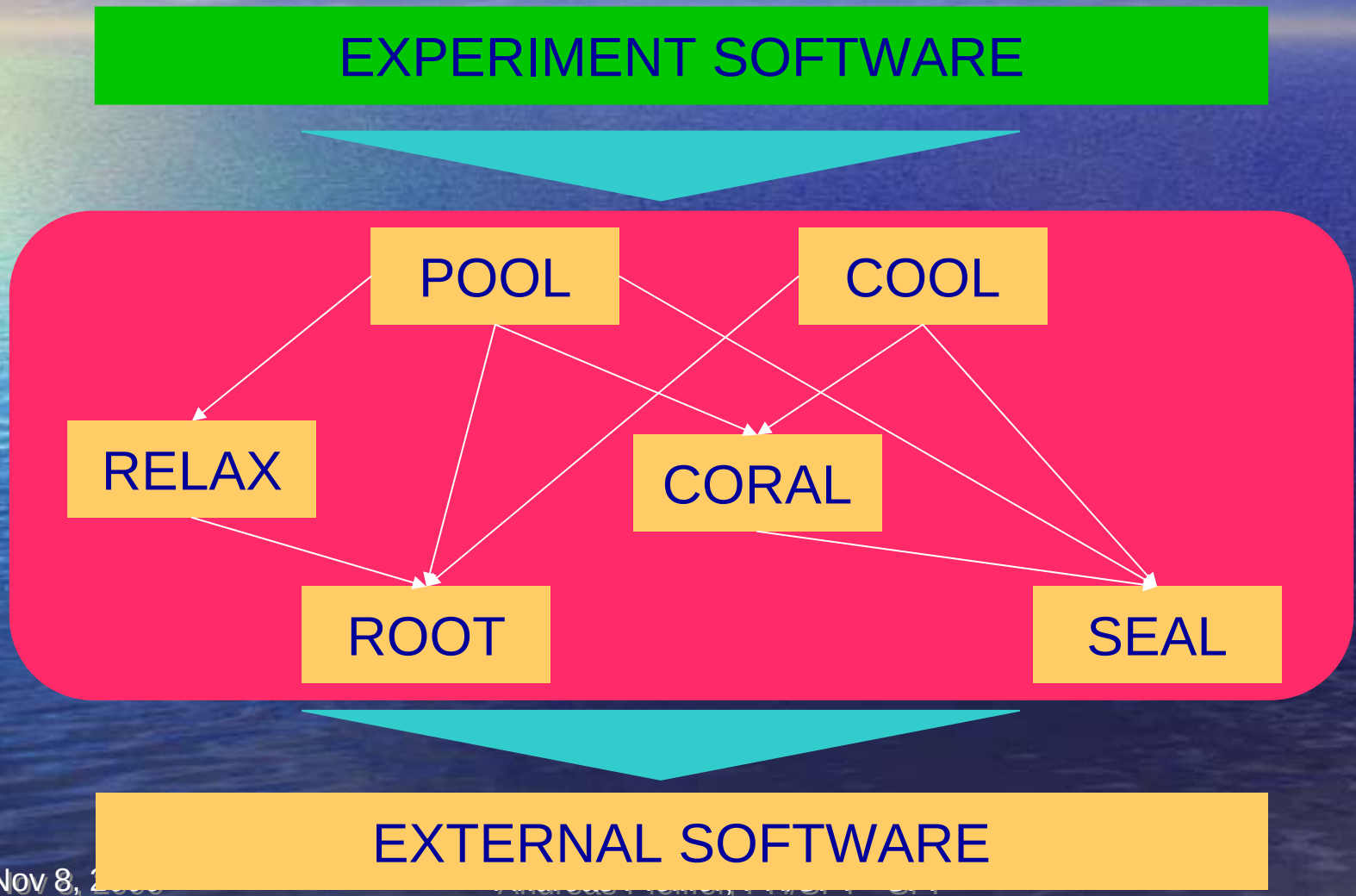
Nightly builds for LCG AA s/w

Andreas Pfeiffer
SPI

Motivation

- Two main goals:
  - Provide prompt feedback of integrations and platform problems to LCG AA developers
  - Provide builds (binary) that the experiments can use directly to make their own tests
    - Validating full stack of LCG AA s/w

LCG Software Stack



Seven Players

Ext SW	ROOT	CORAL	COOL
SEAL	RELAX	POOL	

- (>) 2 Versions each

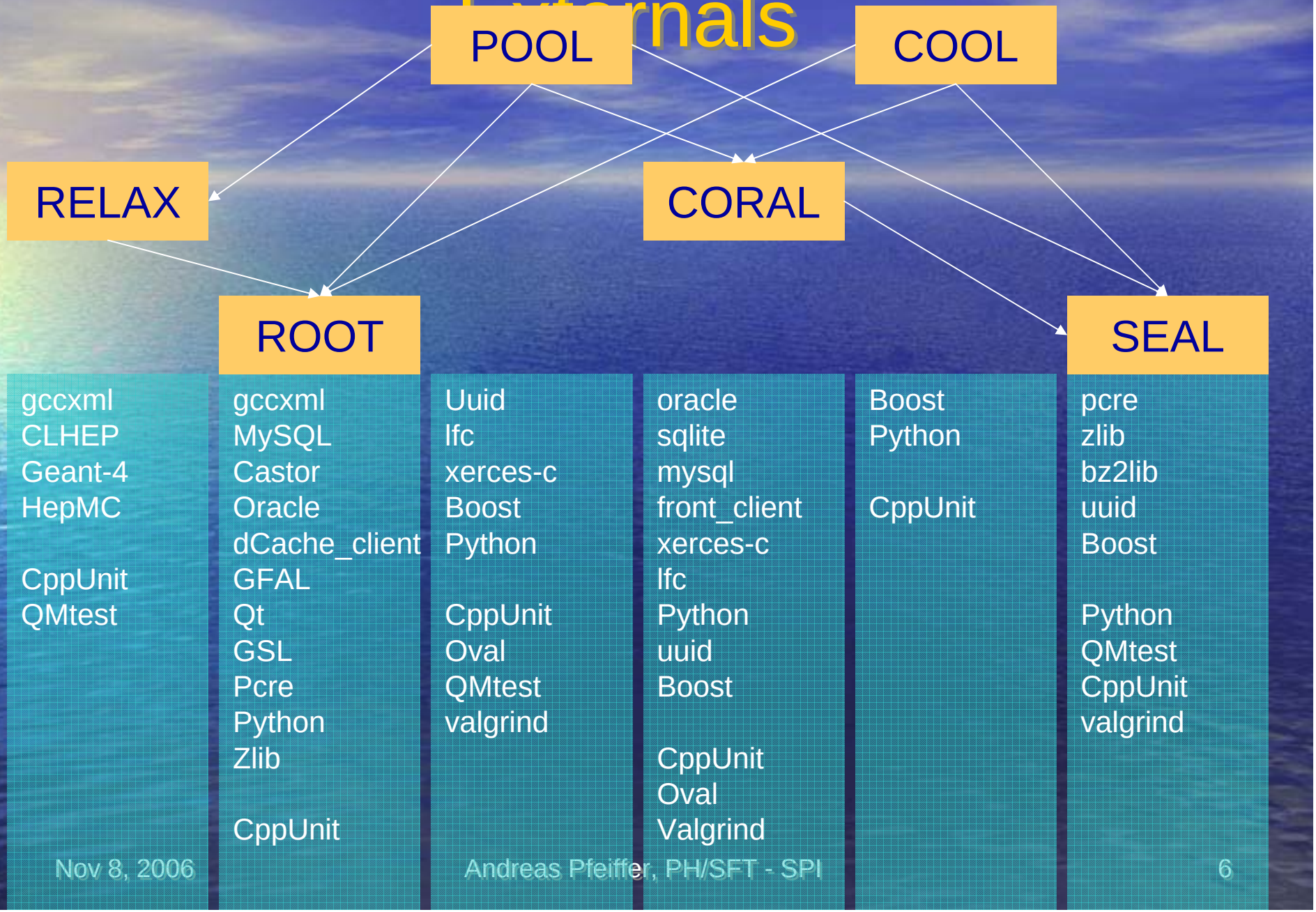
- Latest Working (WORK): Latest known working tag
- Development (DEV): e.g. daily snap or running tag
- ...

2^7 -> 128 possible release combinations

Project Interdependencies

▪ ROOT / reflex ▪ ROOT / math	▪ None	▪ ROOT / meta ▪ ROOT / reflex ▪ CORAL / ▪ SEAL / ▪ RELAX /	▪ SEAL /	▪ ROOT / PyCintex ▪ ROOT / reflex ▪ SEAL / ▪ CORAL /	▪ None

(direct) Dependencies on Externals



Example “Use Cases” - “Slots”

- “Latest Greatest”
  - DEV of all
- “ROOT development”
  - ROOT DEV plus DEV of dependent packages, rest WORK
- “COOL development”
  - COOL DEV, rest WORK
- …

Selection of Use Cases depending on “hot” development areas, decisions taken in AF

Slots and builds

- "Slot" - defined as a set of CVS tags
  - Can/will be opened/closed on demand
  - Each slot defines the weekdays and platforms selected for its builds
    - Via configuration file (config.py from cvs)
- Cronjob builds each night
  - Start determined such that build is finished early morning (08:00)
  - Builds, run tests, install to AFS, analyze logfiles
  - Overwritten every week (Mon, Tue, …)

Nightly build system

- Implemented as a set of Python scripts
- Controlled by a configuration file
  - config.py
- Runs every night on all platforms
  - Via (a)cron on linxu/mac
  - Scheduled job on win (or WinAt)

Output of the process

- Binaries build and installed in AFS
  - …/app/nightlies/slot/day/project/version/platform/
  - LCGCMT is a project in there (slot/day)
  - “stamp-file” to flag build is OK (per platform ?)
- Tags in CVS for reproducable source builds (CMS)
  - Source RPMs in the future (if needed)
- Web page with status of all builds and tests
  - Slot/day/platform views

Known issues

- Builds on non-AFS machines
  - no tokens in “cron”
  - Windows ???
    - Store in DFS ???
  - Mac OS X
    - “polling” data to AFS from linux
- Dealing with missing plug-ins (platform deps)
  - Makes analysis of log files more complex
    - Need to see where exactly the error is
  - No easy algorithm to decide if build is OK (stampfile)
    - Needs table of what should work on which platform

Present status

- Presently in set-up phase
  - Scripts for builds are basically working (Linux/Mac)
- Moving projects to build with CMT and QMtest
  - SEAL, RELAX, CORAL build now with CMT
  - Fixes also needed in LCGCMT
- Rudimentary logfile analysis at present
  - “webified” logs (warnings in blue, errors in red)
  - Static “summary page” for now
    http://lcgapp.cern.ch/spi/aaLibrarian/nightlies/index.html

Near term planning

- Plan to have full stack by end next week
  - Builds and installs in AFS for experiments
  - Web pages with logs for developers
    - Static pages on build logs for a start
  - Running tests through QMtest
    - Needs adaption for CORAL, POOL, COOL
    - Complex testing environment !
  - Porting to Windows environment
- Analyze logs from running tests

Future enhancements

- Several (lots ?) of slots in parallel ?
- Build (and run tests) in parallel to speed up
  - More dedicated machines ? Grid ?
- “Dynamic” web pages
  - Colour code status of builds/tests
  - Needs handle on what is expected to build/run on each platform (plug-ins)

Summary

- Nightly build system for LCG AA s/w
  - Provide prompt feedback of integrations and platform problems to LCG AA developers
  - Provide builds (binary) that the experiments can use directly to make their own tests, validating full stack of LCG AA s/w
- Projects need porting to CMT and QMtest
- Prototype being set up
  - Slc3, slc4/amd64/mac for starting
- Two steps for implementation
  - All projects build and test on all platforms
  - Detailed dynamic web-view