Comments to ROOT/Seal/CORE

Shared Environment for Applications at LHC
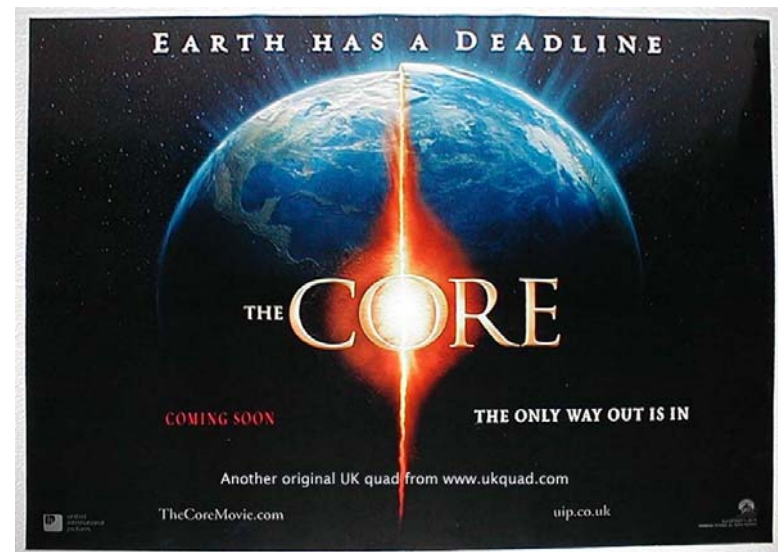
first hit on google images for "CORE"

Reminder of the Charge

◆ Follow-up of the review that took place in October 2003
  – Progress that has been made since the last review
  – Adoption of the recommendations

30/03/05

3

From Pere's talk

Adoption of Recommendations

AA Review Recommendations

- ◆ Foundation Libraries recommendations
  - – Setup tutorials, user-guides and help developers   ➔ partially done
  - – Remove unnecessary dependencies in external packages   ➔ not done
  - – Convergence between the SEAL and ROOT plug-in manager   ➔ not in the Plan
- ◆ Math libraries recommendations
  - – Concerns about the future of Minuit, GSL and CLHEP   ➔ ongoing work
  - – Careful testing to guarantee reliable physics results   ➔ done
  - – Provision of a coherent set of libraries including dictionaries   ➔ ongoing work
- ◆ Dictionary recommendations
  - – Concerns about the size of dictionaries   ➔ already done
  - – Encourage unifying the dictionary with ROOT/CINT   ➔ ongoing work
- ◆ Framework recommendations
  - – Discussion with the experiments to evaluate interest in framework   ➔ in the Plan
- ◆ Scripting recommendations
  - – Evaluate the interest in the interoperability between Boost and Swig before any work is committed   ➔ already done
  - – Continue with the PyLCGDict automatic binding   ➔ done
  - – Seek feedback from the experiment physics community in usability of python in interactive analysis   ➔ ongoing work

From Pere's talk

Progress that has been made since the last review

- "Success should be measured by how widespread the usage is in other LCG components"
  - At the time of the previous review, "only reported user it the POOL project"
- Since then widespread adoption by the experiments
  - Clear from the experiment feedback as reported yesterday
    - Dictionary, Scripting, Foundation Libraries
  - In some cases/components indirect through eg. POOL
    - Plugin mgr

- Clearly major progress has been made!

Reminder of the Charge

◆ Feedback on the program of work for the second phase of the LCG project
- About the new project organization
- The objectives to serve best the LHC experiments
- The technical choices

30/03/05

3

From Pere's talk

Project Organization

- ***ALL*** experiments welcome the proposed ROOT/Seal merger
  - There seems to be general agreement on the *strategy*
- Next step will be a detailed 'tactical' planning
- *LHC experiments should set schedule and priorities*
  - Dictionary, Mathlib, CLHEP replacement, Plugin Mgr…
  - Focus of LCG manpower on the high(est) priority items
  - keep it transparent for end user, simple for core developers
    - minimize new dependencies and req's on client objects
  - Provide fallback solutions in case of delays
  - Architects Forum to supervise the process

The Technical Choices (i)

- Commend the effort to merge and remove duplications for SEAL/ROOT
  - plugin management
  - dictionary
  - mathlibs
- Not enough to "add missing features to ROOT"
  - Should preserve SEAL architectural strengths: component model, limited requirements on client objs
- Lightweight Packaging is **crucial**
  - Especially for math libs and plugin scheme
  - Applications (e.g. trigger) should be able to pick core components (e.g. mathlibs) without buying into the entire (SEAL/ROOT/CORE) framework
    - Minimize/Avoid new dependencies,
    - Reduce where possible!

The Technical Choices (ii)

- We hope that the ROOT-CORE team will seize the opportunity to remove the inheritance from TObject where appropriate to decouple basic classes & components
  - Possible thanks to all the work to support 'foreign' class I/O in ROOT4
  - Esp. important in cases like (Lorentz)Vector, ValidityIntervals, RandomNumberGen, MathLib, etc.
    - Basically, the 'stuff' in SealBase, SealUtils, CLHEP and more…

The Technical Choices (iii)

- plugin management:
  - substantially different approach:
    - Factory (SEAL) vs. Interpreter (ROOT)
  - carefully evaluate the impact on existing experiment schemes
    - e.g. Gaudi component-model
    - Esp. when visible to end-users

The Technical Choices (iv)

- Broad agreement on the need of a Common Dictionary
    - clean implementation
    - unify dictionary clients for experiments using lcgdict/reflex
    - unify automatic generation of glue code (e.g. python bindings)

- Need to go from roadmap to a route map
    - Scheduling, Timescale
    - Subject of May Workshop

Dictionaries : situation in the future

From (one of) Rene's talk(s)

ROOT-CORE Team   4
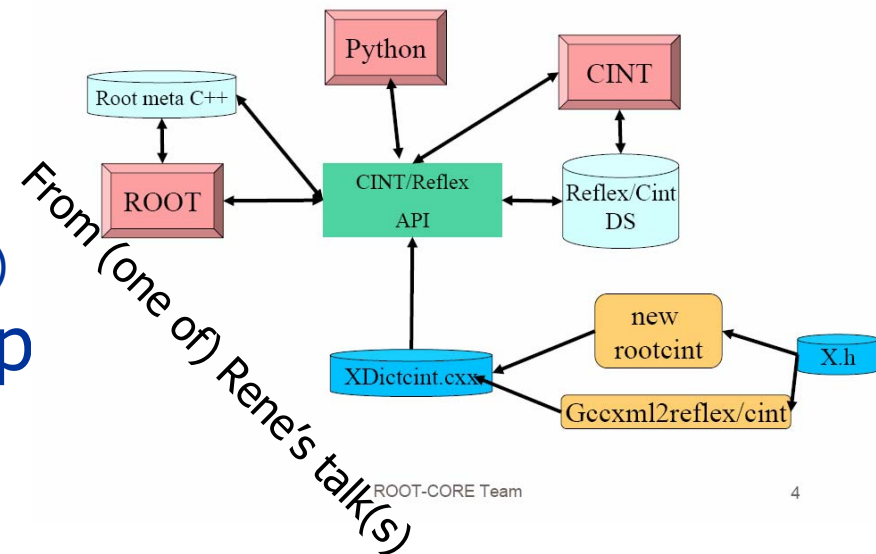
- Mathlib integration work is the most advanced
    - Experiments are neutral about licensing, but concerned about duplication
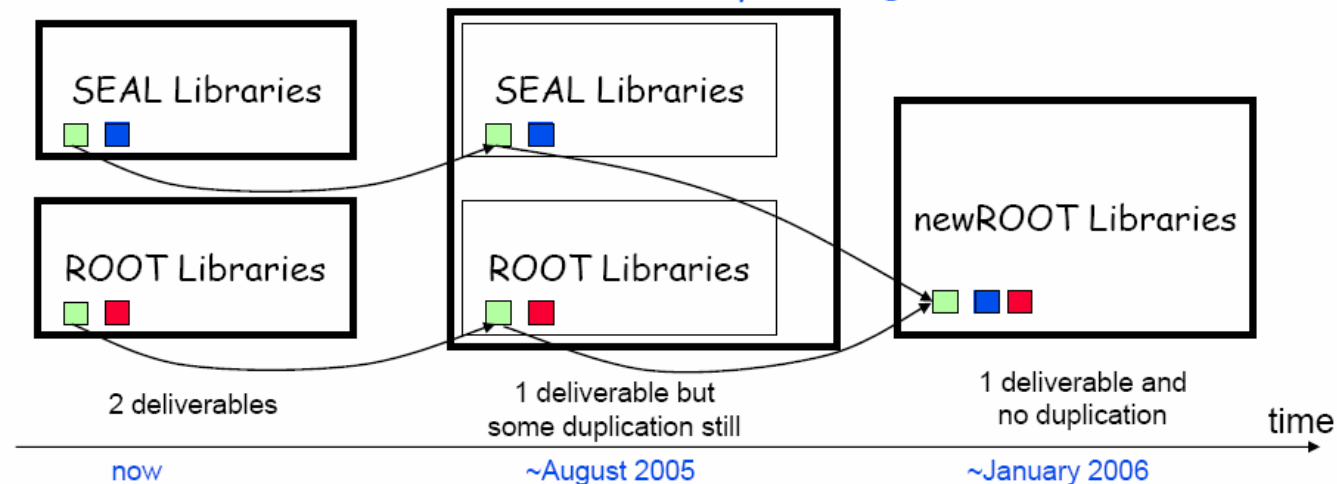   - previous remarks about packaging, dependencies apply here!

Time Scales

SEAL + ROOT Migration

◆ Adiabatic changes towards experiments
 – Experiments need to see libraries they use currently will evolve from current usage today towards a unique set
◆ Some details given during this review
 – More will be needed in the Programme of Work document
 – Will be extra tasks in order to complete migration

SEAL Libraries

ROOT Libraries

SEAL Libraries

ROOT Libraries

newROOT Libraries

2 deliverables

1 deliverable but some duplication still

1 deliverable and no duplication

time

now

~August 2005

~January 2006

(From René's & Pere's talk)

● Support proposed schedule