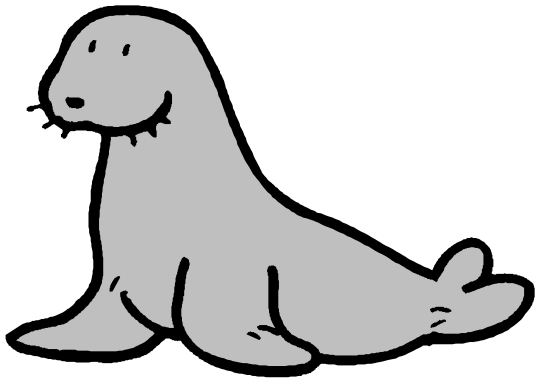
SEAL Project
Core Libraries and Services

18 December 2002

P. Mato / CERN



Shared Environment for Applications at LHC

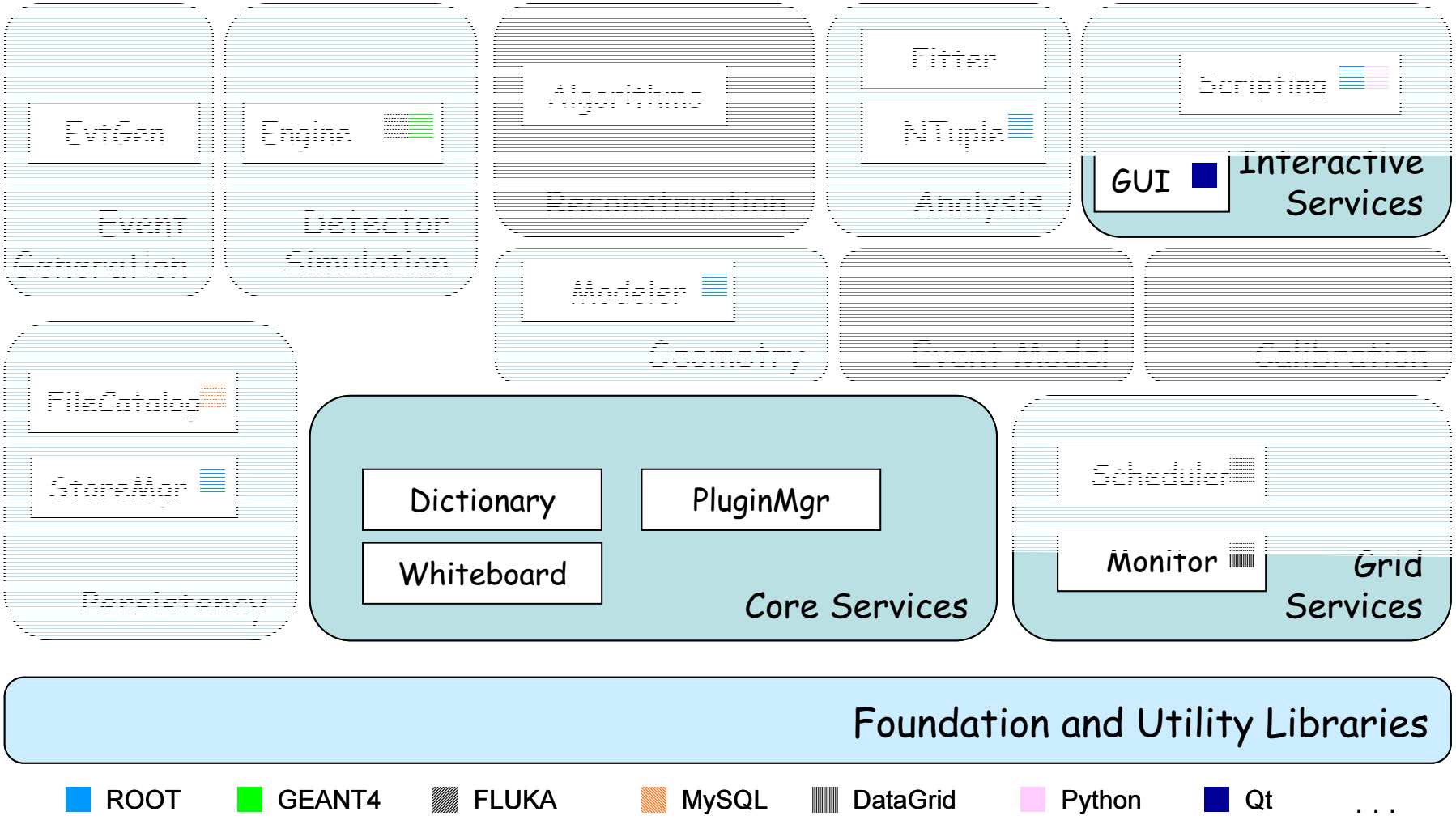
Contents

- Project Scope
- Main Goals
- Work Packages
- Resources
- Main Milestones
- Current Activities
- Summary

Project Scope

- Foundation Class Libraries
  - Basic types (STL, Boost, CLHEP, …)
  - Utility libraries
  - System libraries
  - Domain specific foundation libraries
- Basic Framework Services
  - Component model
  - Reflection
  - Plugging management
  - Incident (Event) management
  - Distribution, Grid
  - Scripting

Domain Coverage



Main Goals

- Provide a **coherent** and as complete as possible set of core classes and services in conformance with overall architectural vision (Blueprint RTAG)
- Facilitate the **integration** of LCG and non-LCG software to build coherent applications
- Avoid duplication of software within the LCG projects

Who the Users are?

- Other LCG application area projects
  - Persistency (POOL), Physicist Interface (PI), Math Libraries, ...
- LHC Experiment Frameworks and Applications
  - ATHENA (ATLAS), COBRA (CMS), GAUDI (LHCb)
- Other HEP projects
  - GEANT4 ?, ...

By When?

- Initial work plan to be presented to SC2 on **January 10th** including detail contents of version v1 alpha
- V1 alpha. Essential functionality sufficient for the other existing LCG projects by **end March 2003**
  - Frequent internal releases (monthly?)
- V1 beta. Essential functionality sufficient to be adopted by experiments by **end June 2003**

Proposed Work Packages

1. Foundation and Utility libraries
2. Component Model and Plug-in Manager
3. LCG Object Dictionary
4. Basic Framework Services
5. Scripting Services
6. Grid Services
7. Education and Documentation

1. Foundation and Utility libraries

- Tasks
  - Inventory of existing utility classes
  - Provide support for *Boost* library, (Loki?)
    - » Boost is a strong candidate to standardize on
    - » Intended to become part of Standard Library (STL)
  - Participation to CLHEP project. Prepare proposal for its evolution
    - » CLHEP workshop Jan 27-31
  - Develop SEAL *utility* and *system library* complementary to Boost and STL from existing code in ClassLib, Gaudi, HepUtilities, etc.
  - Establish guidelines for selecting external libraries

1. Foundation and Utility libraries (2)

- Proposed v1 deliverables
  - SEAL utility candidates inventory (http://cern.ch/seal/components.html)
  - Support Boost library (installation, documentation, etc.)
  - Initial version of SEAL system abstraction library
  - Initial version of SEAL utility library
  - Proposal for external software decision process
- Later deliverables
  - Incorporation of CLHEP evolution

2. Component Model and Plug-in Manager

- Tasks
  - Define component and interface model following the blueprint report guidance
    - » Interfaces, abstract factories, etc.
  - Develop plug-in Manager
    - » Service in charge of managing, querying, [un]loading plug-ins
    - » Application bootstrapping (initialization)
  - Define "Object management protocol"
    - » Object lifetime strategy
  - Document Component Model

2. Component Model and Plug-in Manager (2)

- Proposed v1 deliverables
  - Basic set of interfaces and base classes to support the *Component Model*
  - Initial version of Plug-in Manager. Sufficient for POOL
  - Description of the *Component Model* and *Object Management Protocol*
- Later deliverables
  - Plug-in Manager with sufficient functionality to be used by experiment frameworks

3. LCG Object Dictionary

- Tasks
  - Reflection packages (imported from POOL)
    - » Reflection and ReflectionBuilder
  - Develop tools for populating dictionary from C++ header files (initiated in POOL)
    - » Required by CMS and ATLAS
    - » Investigate gcc-xml technology
  - Develop gateway to Python (Python binding)
    - » Completeness and usability exercise
  - Develop gateway from ROOT
    - » Populate dictionary from CINT (inverse direction to the one developed in POOL)
    - » Should allow to interact to any ROOT object as if it was defined in the LCG dictionary

3. LCG Object Dictionary (2)

- Proposed v1 deliverables
  - Reflection packages with small improvements
    - » Replace static *stub functions* by function objects
    - » Exploit templates for generation of *stub functions*
  - Generation of dictionary from header files (partial C++ support)
    - » Sufficient for CMS and ATLAS event model
  - Python binding
    - » Using Boost.Python
- Later deliverables
  - Full C++ support for the generation of dictionary
  - Gateway from ROOT

4. Basic Framework Services

- Tasks
  - Develop set of basic services for message reporting, exception handling, component configuration, "event" management, etc.
    - » More services will be identified in other projects
  - Develop object "whiteboard"
    - » Study interaction with persistency, visualization and other services

4. Basic Framework Services (2)

- Proposed v1 deliverables
  - Minimal set of basic services sufficient for POOL: message reporting, exception handling, component configuration
- Later deliverables
  - Complete the list of them

5. Scripting Services

- Tasks
  - Define guidelines for developing Python bindings
    - » Evaluate existing options: SWIG, Boost.Python, SIP,...
    - » Study Python extension modules inter-dependencies
  - Develop Python bindings for standard services and utility libraries developed in SEAL
    - » Enable scripting for application configuration
  - Upgrade Python bindings for ROOT (former RootPython)

5. Scripting Services (2)

- Proposed v1 deliverables
  - Evaluation report. Python bindings guidelines
  - ROOT python bindings (PyROOT) following guidelines
- Later deliverables
  - Bindings to SEAL provided services and libraries

6. GRID Services

- Tasks
  - Gather requirements from POOL, PI for GRID-enabled services
  - Provide common interface to various Grid middleware
- Proposed v1 deliverables
  - none

7. Education/Documentation

- Tasks
  - Documentation, Tutorials, …
  - Help incorporate SEAL components into LCG projects and experiment frameworks
- Proposed v1 deliverables
  - Documentation

Resources

- Started with a small team (~3 FTE):
  - Lassi Tuura (CMS), Massimo Marino (ATLAS), Stefan Roiser (LHCb), Lorenzo Moneta (IT/API), Jacek Generowicz (G4, IT/API), Pere Mato (EP/SFT)
- Expected to ramp to ~8 FTE by summer '03
- Not yet assigned people to work packages
  - As soon as work packages are becoming better defined new people can be integrated and assigned to tasks
- People interested to participate in SEAL should contact us

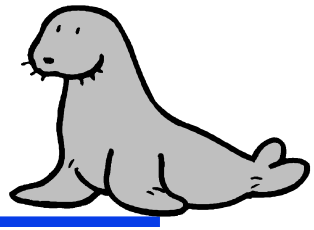
Main Milestones

- 2002/10/30 Establish core libraries and services (SEAL) project
- 2002/11/30 Define the V1 SEAL software
- 2002/12/1 Prototype object dictionary service
- 2003/1/15 Establish external software decision process
  - Establish the process and policies by which decisions are made on what external software is to be used by the LCG applications area.
- 2003/1/31 Complete the initial SEAL workplan
  - Complete the initial SEAL workplan for submission to the SC2. Should cover (at least) the content and implementation plan for SEAL V1.
- 2003/3/31 SEAL V1 essentials in alpha
  - The most essential elements of the V1 SEAL suite (as requested by projects needing to use them) are available in alpha.
- 2003/5/31 Grid enabled services defined
  - The SEAL services which must be grid-enabled are defined and their implementation prioritized.

Current Activities

- Daily meetings
  - Basic organization, know each other, brainstorming, decisions, to-do list, …
- Initial activities
  - Review existing libraries and services
  - Establishing initial plan
  - Building the initial project infrastructure
  - Agree on naming/coding/style conventions
  - Populate CVS repository with software from various sources

Project Information



- Web
  - http://cern.ch/seal
- Mailing lists with archive
  - General Discussion: seal@cern.ch
  - Developers: seal-developers@cern.ch
- Other project infrastructure provided by SPI coming soon
  - Code repository, Project portal and bug tracking, etc.

Summary

- This is the first public presentation of the SEAL project
  - Feedback is welcome. Now it is the moment
  - Initial detailed work plan to be presented in January
- Proposal of work breakdown structure (WBS)
- Starting to define contents of v1 release
  - Driven by the needs of other LCG projects (POOL)
- Building up a development team
  - Started small
  - Open to anybody willing to participate