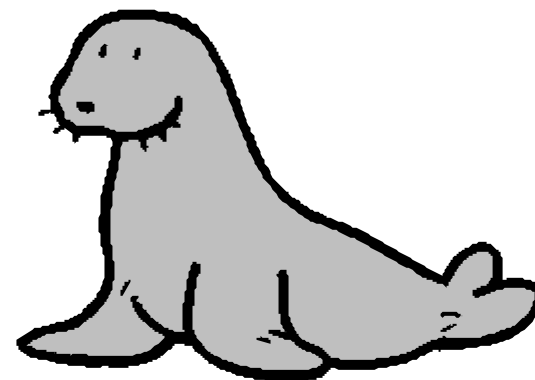


---

# Project Work Plan

## SEAL: Core Libraries and Services

SC2 Meeting  
10 January 2003  
P. Mato / CERN



*Shared Environment  
for Applications at LHC*



# Contents

---

- ✍ Project Overview
  - Purpose, Scope and Objectives
  - Deliverables and Schedule
- ✍ Project Organization
- ✍ Work Packages
- ✍ Resources
- ✍ Main Milestones
- ✍ Current Activities
- ✍ Summary

# Project Overview: Purpose

---

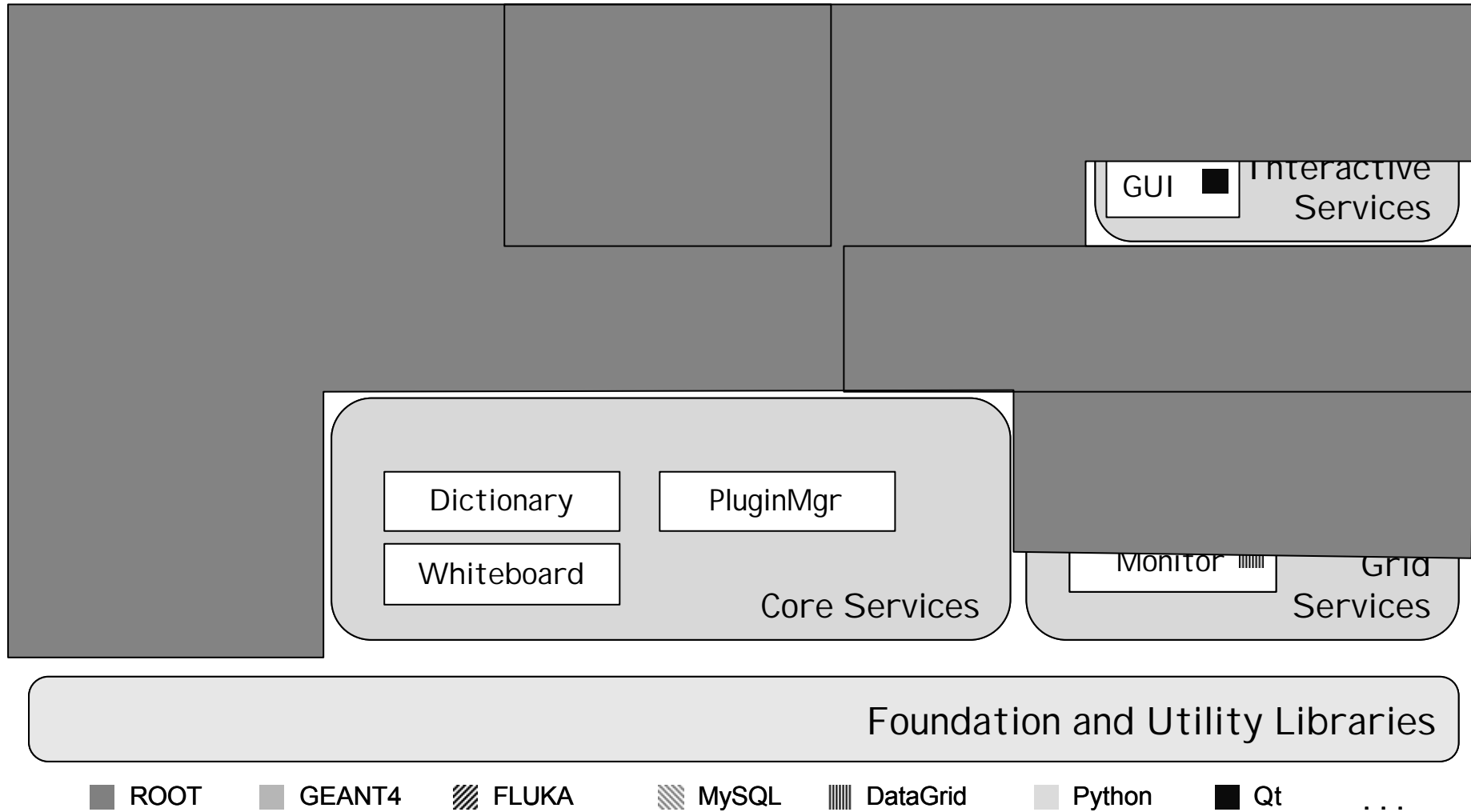
- ✍ Provide a **coherent** and as complete as possible **set of core classes and services** in conformance with overall architectural vision (Blueprint RTAG)
- ✍ Facilitate the **integration** of LCG and non-LCG software to build coherent applications
- ✍ Avoid duplication of software within the LCG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

# Project Overview: 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



# Customers

---

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

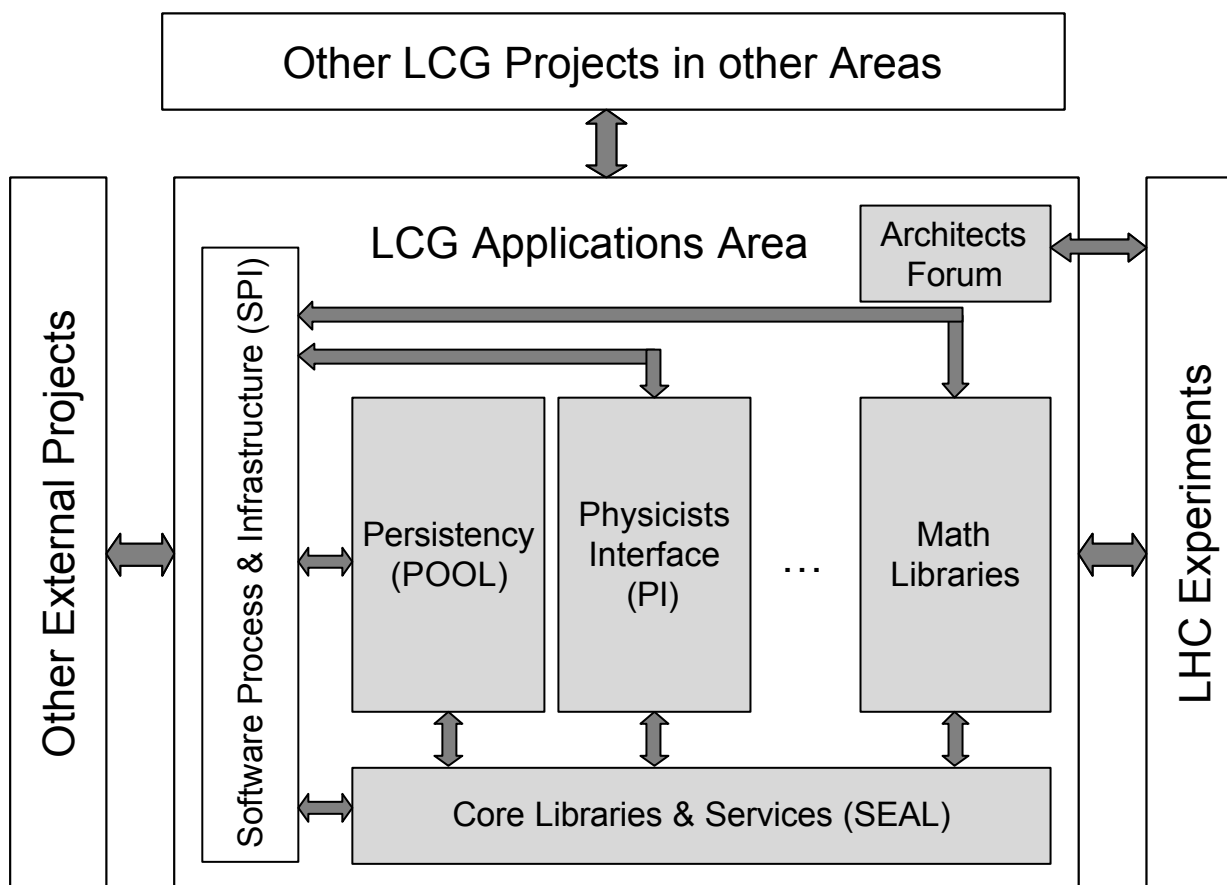
# Project Overview: Schedule

---

- ✍ Initial work plan being presented to SC2 on **January 10<sup>th</sup>** including detail contents of version v1 alpha
- ✍ March 2003 - V1 alpha
  - Essential elements with sufficient functionality for the other existing LCG projects (POOL, ...)
  - Frequent internal releases (monthly?)
- ✍ June 2003 - V1 beta
  - Complete list of the currently proposed elements with sufficient functionality to be adopted by experiments

# Project Organization: External

- ✂ The Project reports
  - LCG Applications Area Manager
  - LCG Project Leader
  - SC2 Committee
- ✂ The staff reports
  - Line management
  - CERN staff to their group leaders
- ✂ Relations to other external projects
  - CLHEP, ROOT, ...
- ✂ Architects Forum



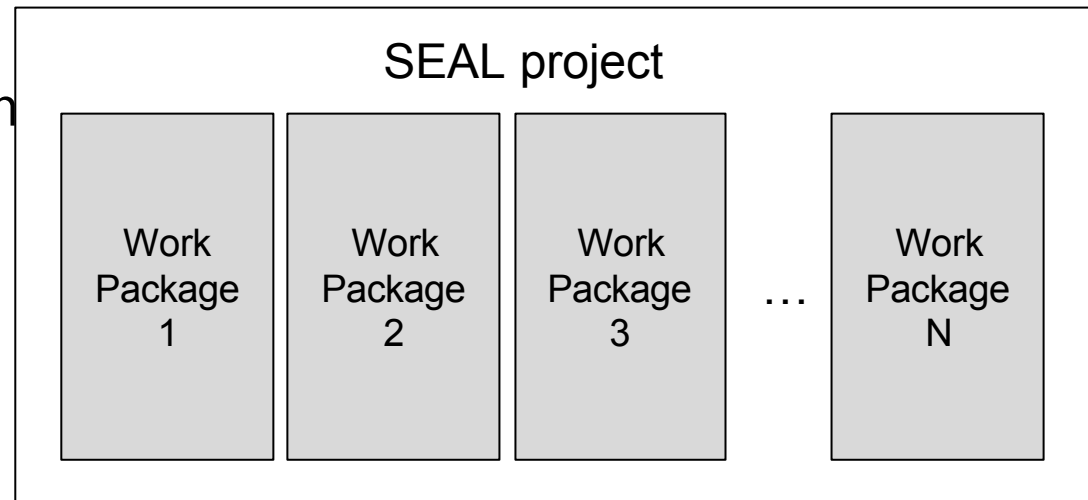


# Project Organization: Internal

---

## ✍ Structure

- No need for complex structure
- A number of Work Packages (WP) has been defined
- 1-3 developers on each WP



## ✍ Roles

- Project Leader
- Work Package Manager
- Developer

# 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

---

## ✍ Main activities and tasks

- Inventory of existing utility classes
- Provide support for *Boost* library
  - » 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

---

- ✍ Main activities and 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

---

## ✍ Main activities and 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* for the time being

## ✍ Later deliverables

- Full C++ support for the generation of dictionary
- Gateway from ROOT



# 4. Basic Framework Services

---

## Main activities and 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

---

## ✍ Main activities and 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

---

## ✍ Main activities and tasks

- Gather requirements from POOL, PI for GRID-enabled services
- Provide common interface to various Grid middleware

## ✍ Proposed v1 deliverables

- none

# 7. Education/Documentation

---

## Main activities and tasks

- Produce documentation
- Produce training material (tutorials)
- Help incorporating SEAL components into LCG projects and experiment frameworks

## Proposed v1 deliverables

- Documentation for the delivered components

# 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
- ✍ Staff skills
  - Several years of experience in C++ and core software required.
- ✍ 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
  - Open to other people interested to participate in SEAL

# Resource Allocation

---

<b>WBS</b>	<b>Name</b>	<b>FTE (available/required)</b>
1	Foundation and Utility libraries	0.5 / 1.0
2	Component Model and Plug-in Manager	0.5 / 0.5
3	LCG Object Dictionary	0.5 / 2.0
4	Basic Framework Services	0.5 / 1.0
5	Scripting Services	0.5 / 1.0
6	Grid Services	0.0 / 1.5
7	Education and Documentation	0.5 / 1.0
<b>total</b>		<b>3.0 / 8.0</b>

✍ Available resources should be sufficient for v1 alpha (March)





# Project Tracking Plan

---

- ✍ Requirements management
  - After each major release (3 per year), changes or new requirements can be proposed
- ✍ Schedule control
  - The project has defined a number of major and minor milestones
  - The project plan will be controlled and tracked quarterly (Application Area Manager, PEB, SC2,...)
- ✍ Quality Control
  - Make use of the bug tracking system
  - Quality control process defined/supported by SPI to be applied

# 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 work plan
  - Complete the initial SEAL work plan 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.

# Software Process Plan

---

## Process Model

- Adopt guidelines described in *Thoughts on Software Process* by T. Wenaus.
- Combination of applicable elements from *Extreme Programming* and *Rational Unified Process*
- Work in tight, iterative development cycles: release early and release often

## Methods, Tools and Techniques

- Use the ones provided by SPI

## Infrastructure

- Provided by SPI



# 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](mailto:seal@cern.ch)
- Developers: [seal-developers@cern.ch](mailto:seal-developers@cern.ch)

## Project Plan (1.0)

- [http://cern.ch/seal/documents/SEAL\\_Project\\_Plan\\_1\\_0.pdf](http://cern.ch/seal/documents/SEAL_Project_Plan_1_0.pdf)

# Summary

---

- ✍ Reviewed the project purpose, scope and objectives
- ✍ Very simple project organization
- ✍ Proposal of work breakdown structure (WBS)
  - Seven work packages defined
- ✍ First project deliveries defined
  - March 2003 first public release (sufficient for POOL et al.)
  - June 2003 v1 release (sufficient for experiments frameworks)
- ✍ Resources
  - Started small (~3 FTE) expected to ramp to (~8 FTE) by summer
  - Acquisition mainly on voluntary basis
- ✍ Building up the development team