
SEAL Project Overview

LCG-AA Internal Review

20-23 October 2003

P. Mato / CERN



Contents

- ◆ Project Overview
 - History
 - Scope and Goals
 - Current Status
- ◆ Work Packages
- ◆ Milestones
- ◆ Software Process, Documentation, QA, ...

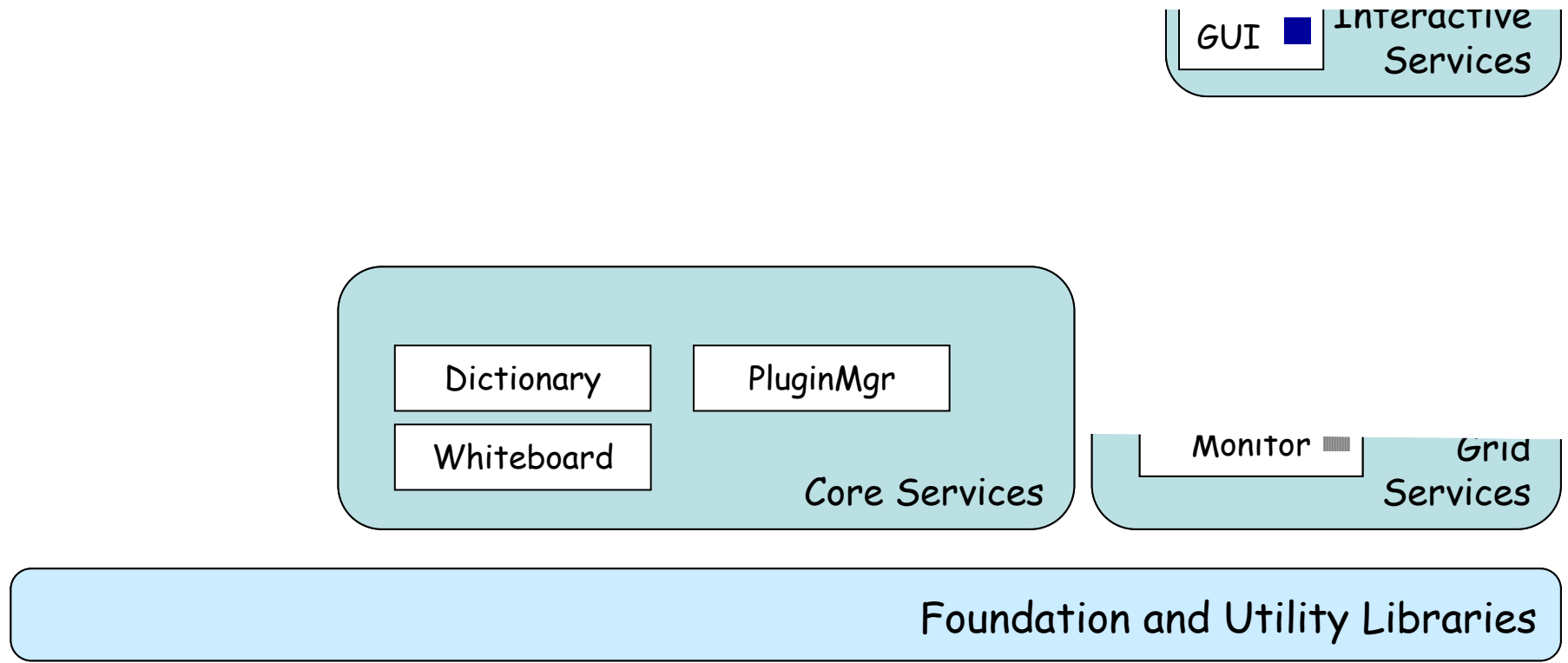
Project History

- ◆ The Math Libraries RTAG recommended to start a project on support and coordination for math libraries
 - Initial math libraries project established in June 2002
- ◆ The Blueprint RTAG recommended to start a common project on core tools and services
 - SEAL: Shared Environment for Applications at LHC
- ◆ SEAL project established end of October 2002
- ◆ Project Plan presented to SC2 in January 2003
- ◆ Math libraries activities incorporated into SEAL project in February 2003

SEAL Overview

- ◆ SEAL aims to
 - Provide the software infrastructure, basic frameworks, libraries and tools that are common among the LHC experiments
 - Select, integrate, develop and support foundation and utility class libraries
 - Develop a coherent set of basic framework services to facilitate the integration of LCG and non - LCG software
- ◆ Scope
 - Foundation Class Libraries
 - » Basic types (STL, Boost, CLHEP, ...), utility libraries, system isolation libraries, domain specific foundation libraries
 - Math Libs
 - Basic Framework Services
 - » Component model, reflection, plugin management, incident (event) management, distributed computing, grid services, scripting

Domain Coverage



■ ROOT ■ GEANT4 ■ FLUKA ■ MySQL ■ DataGrid ■ Python ■ Qt ...



Assumptions, constraints, risks

◆ Do not re-invent the wheel

- Most of the core software to be delivered by SEAL exists - more or less - in experiments' core software
 - » We will re-use as much as possible existing software
 - » Most of the work will be in re- packaging existing pieces of software

◆ If wheel squeaks...

- Develop / adapt / generalize in order to achieve the necessary level of coherency and conformance to the architectural vision already established

◆ Adopt a Seal

- In order to use SEAL, projects will need to replace their own software elements with SEAL functionally equivalent ones. This will certainly imply some period of instability for the experiment applications

Customers

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

Project Work Packages

Foundation	Foundation and Utility Libraries and Plug-in Manager
MathLibs	Math Libraries Support and Coordination
Dictionary	LCG Object Dictionary
Framework	Component Model and Basic Framework services
Scripting	Scripting Services
Grid	Grid Services (not yet active)
Documentation	Education and Documentation

People

Foundation	<i>Lassi Tuura, Lorenzo Moneta, Massimo Marino, Radovan Chytracek</i>
MathLibs	<i>Fred James, Matthias Winkler</i>
Dictionary	<i>Stefan Roiser, Christian Arnault, RD Schaffer, Zhen Xie, Pere Mato</i>
Framework	<i>Radovan Chytracek, Lassi Tuura, Pere Mato, Massimo Marino, Lorenzo Moneta</i>
Scripting	<i>Jacek Generowicz, Pere Mato, Wim Lavrijsen, Massimo Marino</i>
Grid	
Documentation	<i>Jacek Generowicz</i>

SEAL Versions Road Map

Release	Date	Status	Description (goals)
V 0.1.0	Released 14&26/02/03	internal	<ul style="list-style-type: none"> ◆ Establish dependency between POOL and SEAL ◆ Dictionary generation from header files
V 0.2.0	Released 04/04/03	public	<ul style="list-style-type: none"> ◆ Essential functionality sufficient for the other existing LCG projects (POOL) ◆ Foundation library, system abstraction, etc. ◆ Plugin management
V 0.3.0	Released 23/05/03	internal	<ul style="list-style-type: none"> ◆ Improve functionality required by POOL ◆ Basic framework base classes
V 1.0.0	Released 18/07/03	public	<ul style="list-style-type: none"> ◆ Essential functionality sufficient to be adopted by experiments ◆ Collection of basic framework services ◆ Scripting support
V 1.1.0	Released 03/09/03	public	<ul style="list-style-type: none"> ◆ Corrections and improvements of Framework
V 1.2.0	Released 16/10/03	public	<ul style="list-style-type: none"> ◆ Support for ICC and VC++ compilers



Milestones

2002/10/30	Done	Establish core libraries and services (SEAL) project
2002/11/30	Done	Define the V1 SEAL software suite
2002/12/1	Done v=17	Prototype object dictionary service released
2003/1/10	Done v=0	Present the initial SEAL work plan to SC2
2003/3/31	Done v=7	SEAL V1 essentials in alpha (V0.2)
2003/5/16	Done v=8	SEAL V0.3 internal release
2003/5/30	Done	Delivery of first round of GSL enhancements
2003/6/30	Done v=10	Nightly builds deployed in SEAL
2003/6/30	Done v=18	SEAL V1 release
2003/7/31	Late	Math library workplan in place
2003/8/30	Done v=44	SEAL icc test build support
2003/9/15	Late	SEAL ecc test build support
2003/9/15	Done v=24	SEAL support for Windows binaries
2003/9/30	Late	Statement on GSL and NAG usage for math library



Software Process

◆ Design

- Team design sessions (sometimes very lengthy discussions)

◆ Python prototypes

- To illustrate use cases and functionality
- To test design choices

◆ Configuration and Build system

- *SCRAM* is used to configure and build the software (*CMT* used to build the Win32 binaries)

Quality Assurance

◆ Code Review

- No formal code reviews
- Coding done very often in pairs (XP style)
- More than one developer knowledgeable for each package

◆ Testing

- Most of the SEAL tests are unit tests based on *CppUnit*
- 217 tests driven by *QmTest* (small tunings still needed)

◆ Bug reporting and tracking

- Savannah Portal
- Internal SEAL "problems" also reported as bugs

Documentation

- ◆ Code Reference

- Generated with Doxygen

- ◆ HowTo's

- A set of HowTo's pages to teach specific aspects of SEAL
- Being incorporated into the SEAL Workbook

- ◆ Release Notes

- Detailed release notes for each release

- ◆ Design documents

- Partial design documents exists in SEAL web