THE OFFLINE SOFTWARE FRAMEWORK OF THE NA61/SHINE EXPERIMENT

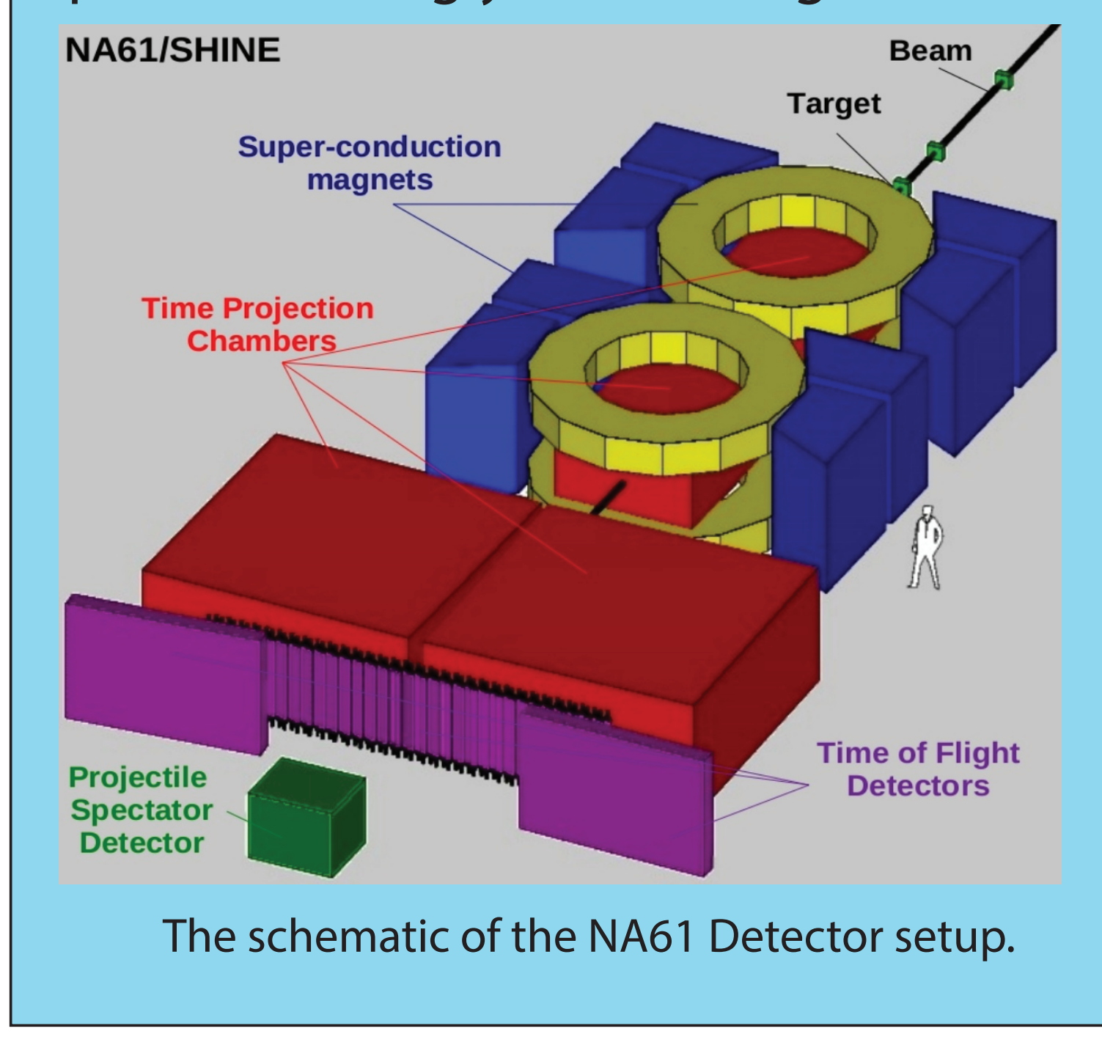
Roland Sipos, for the NA61 Collaboration

E-mail: Roland.Sipos@cern.ch

Co-authors: Andras Laszlo, Antoni Jerzy Marcinek, Tom Paul, Marek Szuba, Michael Unger, Darko Veberic, Oskar Wyszynski

Introduction

NA61/SHINE (SPS Heavy Ion and Neutrino Experiment) is an experiment at the CERN SPS using the upgraded NA49 hadron spectrometer. Moreover, p+p, p+Pb and nucleus+nucleus collisions will be studied extensively to allow for a study of properties of the onset of deconfinement and search for the critical point of strongly interacting matter.



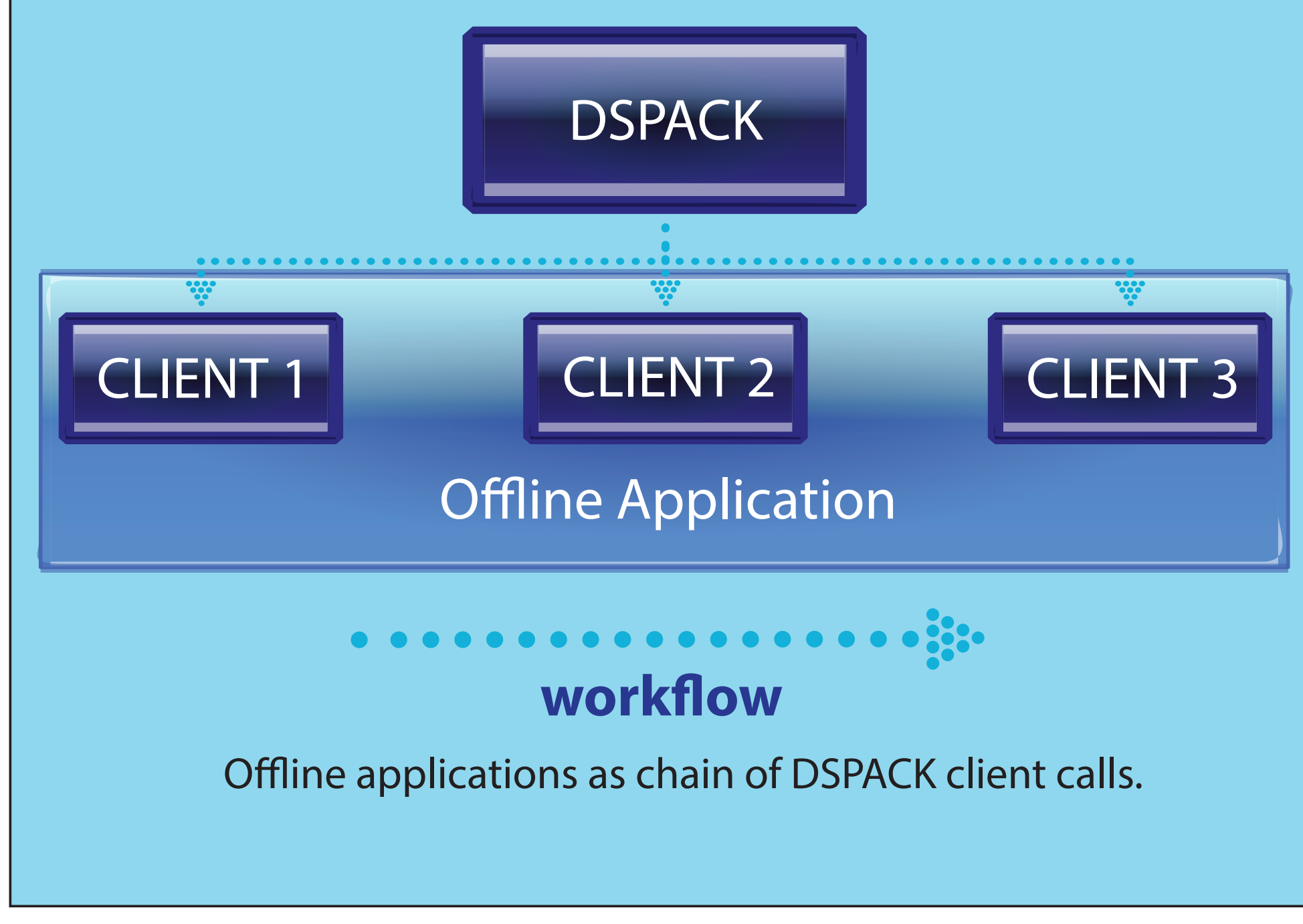
The schematic of the NA61 Detector setup.

Currently NA61/SHINE uses the old NA49 software framework for offline related tasks. The core of this legacy framework was developed in the early 1990s and the collaboration made huge efforts to maintain the rigid software environment, without great success.

In this contribution we will introduce the new software framework, called Shine, a tool for data reconstruction, simulation and analysis.

Legacy software

DSPACK is an implementation of a client-server software architecture with extended capabilities, such as combinations of early object oriented principles. There are multiple choices to structure and store a given data set and provide as a resource for requesters. The ideology was that a kind of server application should handle all data requests and store the description of physics event. Basically every standalone program that attempts to connect to a DSPACK server instance and manipulates stored data, are called clients. On can create standalone programs that use this memory manager to solve sub-tasks of an application. The application itself will be the chain of client calls and DSPACK instructions in a script. In the legacy software several offline related complex tasks were represented with the chain ideology.



Offline applications as chain of DSPACK client calls.

There are multiple services provided by CERN, on which the NA61 software depends on. A complicated environmental setup is required in order to start processing, such as setting up the directory of binaries or paths to dependencies. The environment of the legacy heavily relies on the fixed paths provided by CERN IT infrastructure.

Software upgrade proposal

The experiment's data taking period started in 2007 and will continue to collect data until 2014. An agile and portable software environment is rather a need than an option to aid the continuous productive work for the Collaboration. A decision on a software upgrade strategy was taken, taking into account the known issues. These and the requirements from the new software point out the essential pillars, structural design and necessary key features.

Problems	Solutions
Bound to infrastructure.	Unified language and data format.
Concurrent data formats.	New, framework design, modularity.
Mixed programming languages.	Integration of legacy software.
Obsolete software architecture.	New offline algorithms.
Out-dated production tools.	Portability and support.
Lack of support and documentation.	

Pattern and skeleton

The **Auger Offline Framework** was used as the skeleton for Shine. The design contains distinguished key features, such as extensibility, and there are three principal pillar describing the planned framework.

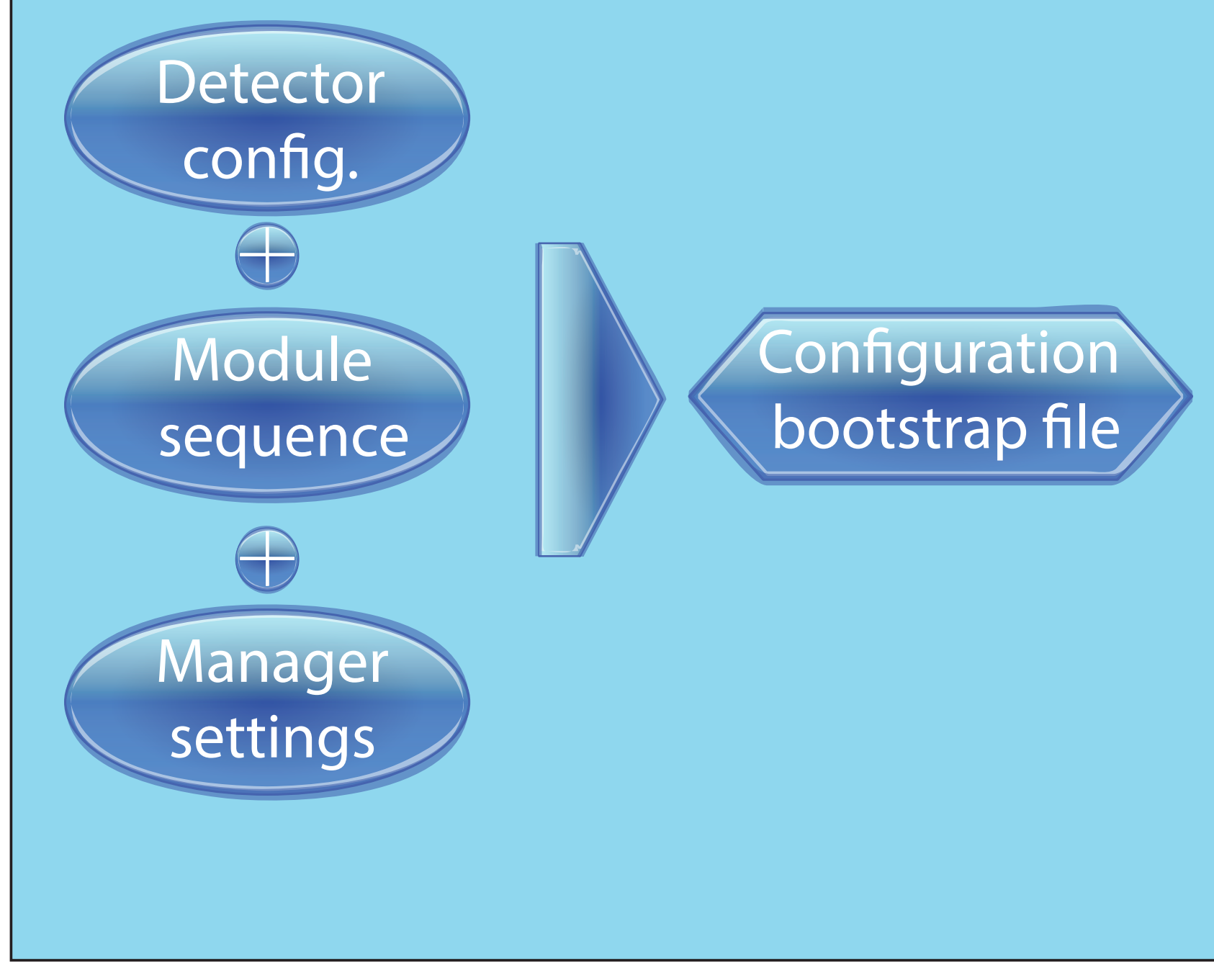
Modules — Basically modules stand for processing elements, and as the user interface for the framework. They should be assembled and sequenced through XML configuration files. This will stand as the application programming interface (API).

Event — The global data model which stores offline related information. Modules can acquire data from the model, and only if needed, they can communicate through this structure.

Detector description — The experimental facility need to be modeled and implemented in the framework and stands as a provider of the updated and correct data what corresponds to detector configuration.

Configuration

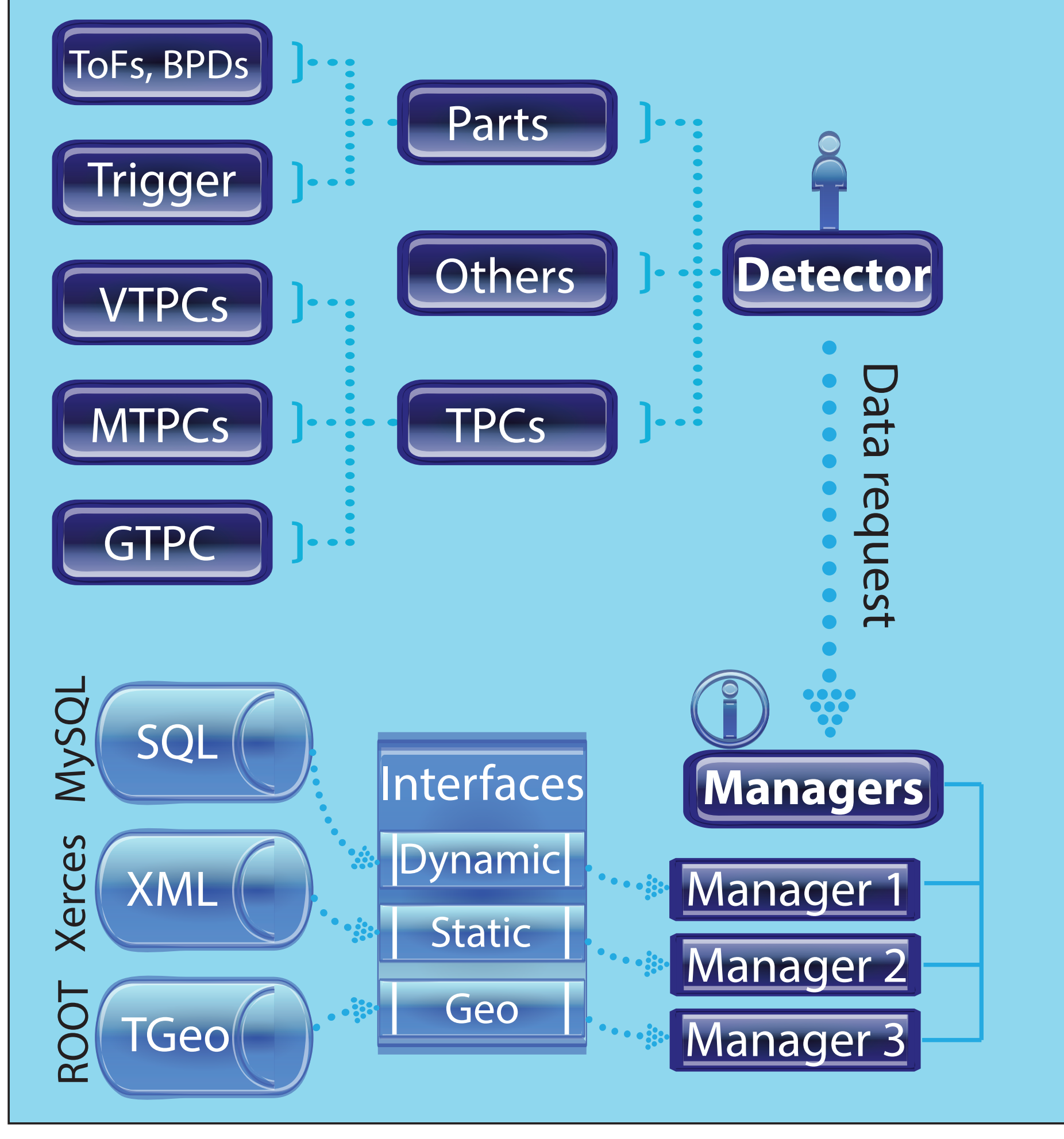
Central configuration stands as the interface to every configuration related task. The goal is to provide access for users to configuration files via the **CentralConfig**, and give option for read, write and modify the settings.



- XML
- Run-time XML Schema validation
- Central configuration for: Detector, Data, Modules
- MD5 Check-summed
- Configuration saved with data

Detector description

One of the principal parts in experimental physics software is that how the detector will be modeled and how will be the interface working to access instrument related data. This need to be separated by two important sections, to the **Detector User Interface** and the **Detector Backend**.



User Interface

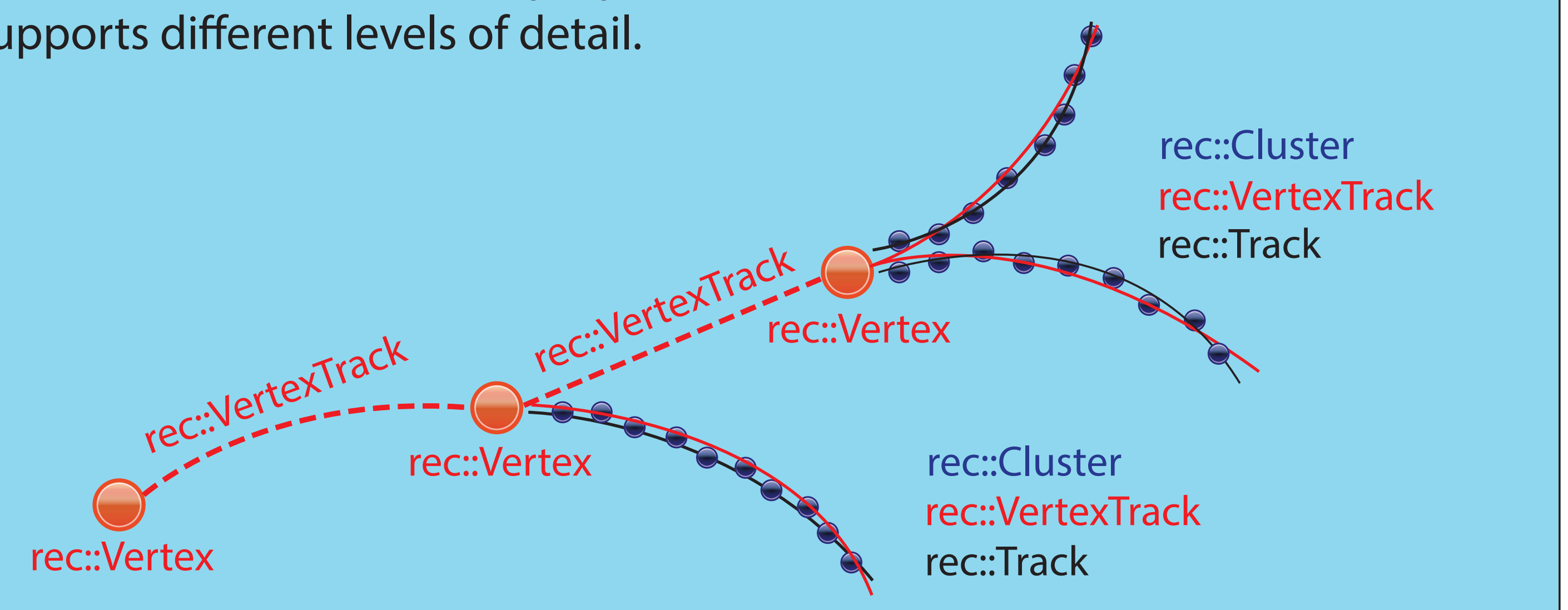
- User deals with a single UI
- Extensible
- Hierarchical structure
- Constant references to data

Backend

- Manager ideology
- Sought data can come from different sources
- Static info in XML Wrappers for MySQL, Xerces

SHOE - The Shine Offline Event

The aim of the SHOE is to replace the legacy DS, T61 and NDST structures by one unified data format. It is based on ROOT and made of streamable collection of classes for offline purposes. The content of information is scalable so supports different levels of detail.

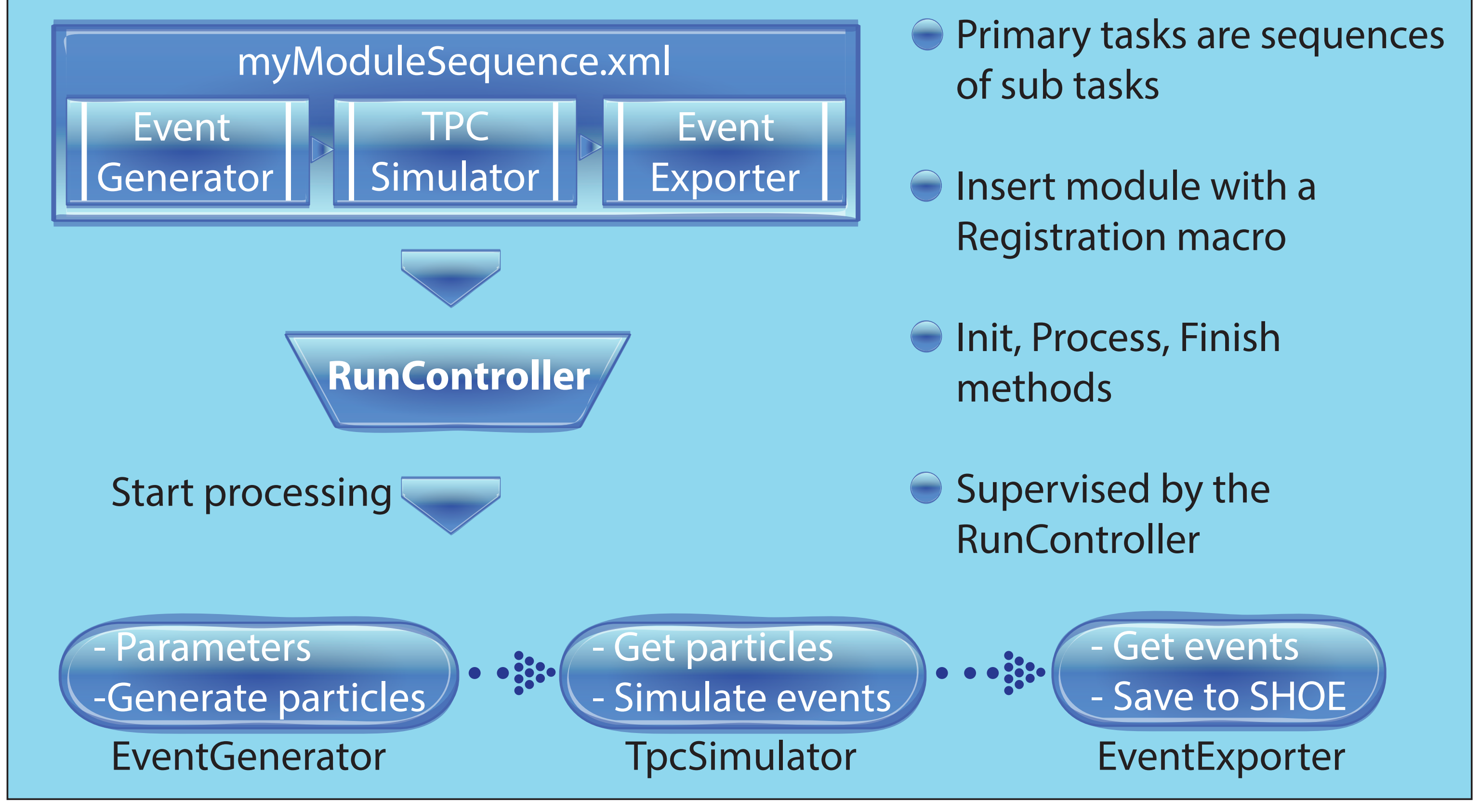


One can access parent- and child-objects using simple indices (SHOE-Laces), and the connection between event objects made by standard lists. With this method the navigation through the data is easy and allows fast random access and removal of objects.

Modules

The elements where users can insert unique processing elements inside the framework and manipulate the event structure, is the Module.

The interface called VModule defines three virtual functions that users need to implement. Each method returns a ResultFlag in order to report success, failure or even instructions for the **RunController** that traces and monitors the work-flow of the offline application.



- Primary tasks are sequences of sub tasks
- Insert module with a Registration macro
- Init, Process, Finish methods
- Supervised by the RunController

Support

The framework comes with many additional feature and support.

Wide range of **utilities** from the fields of math, physics and computer science. Used in user modules (e.g.: ODE Runge-Kutta Integrator) and also in the core framework (e.g.: XML Reader, custom exceptions, shadow pointer).

A tool called **Shape** to download and install every external dependencies with ease and to set up the required running environment for Shine.

A **buildbot** was set up for automatizing compilation, testing and validation after each commit.

Doxygen **documentation** is generated in order to provide a clear picture about the framework and give information for Shine developers and power-users.

Achievements

- New, user friendly offline framework for the whole Collaboration.
- Stable and flexible system, with state-of-art techniques of computer science.
- Huge efforts to revive and integrate the legacy software within Shine.
- Collaboration members are migrating to the Shine Framework.

References

S. Argiro et al. The Offline Software Framework of the Pierre Auger Observatory; Nucl. Instrum. Meth. A506 (2003) 250.

Review of NA61 Software Upgrade Proposal [http://indico.cern.ch/conferenceDisplay.py?confId=125760]