



J. Adamczewski
M. Al-Turany
D. Bertini
H.G.Essel
S.Linev



Analysis Organization: Go4 Analysis Steps and TTask

ROOT workshop February 2004



Contents

- The problem: organization of analysis
- ROOT TTask mechanism
- The Go4 analysis framework
- The Go4 analysis steps
- Upgrade of Go4 framework

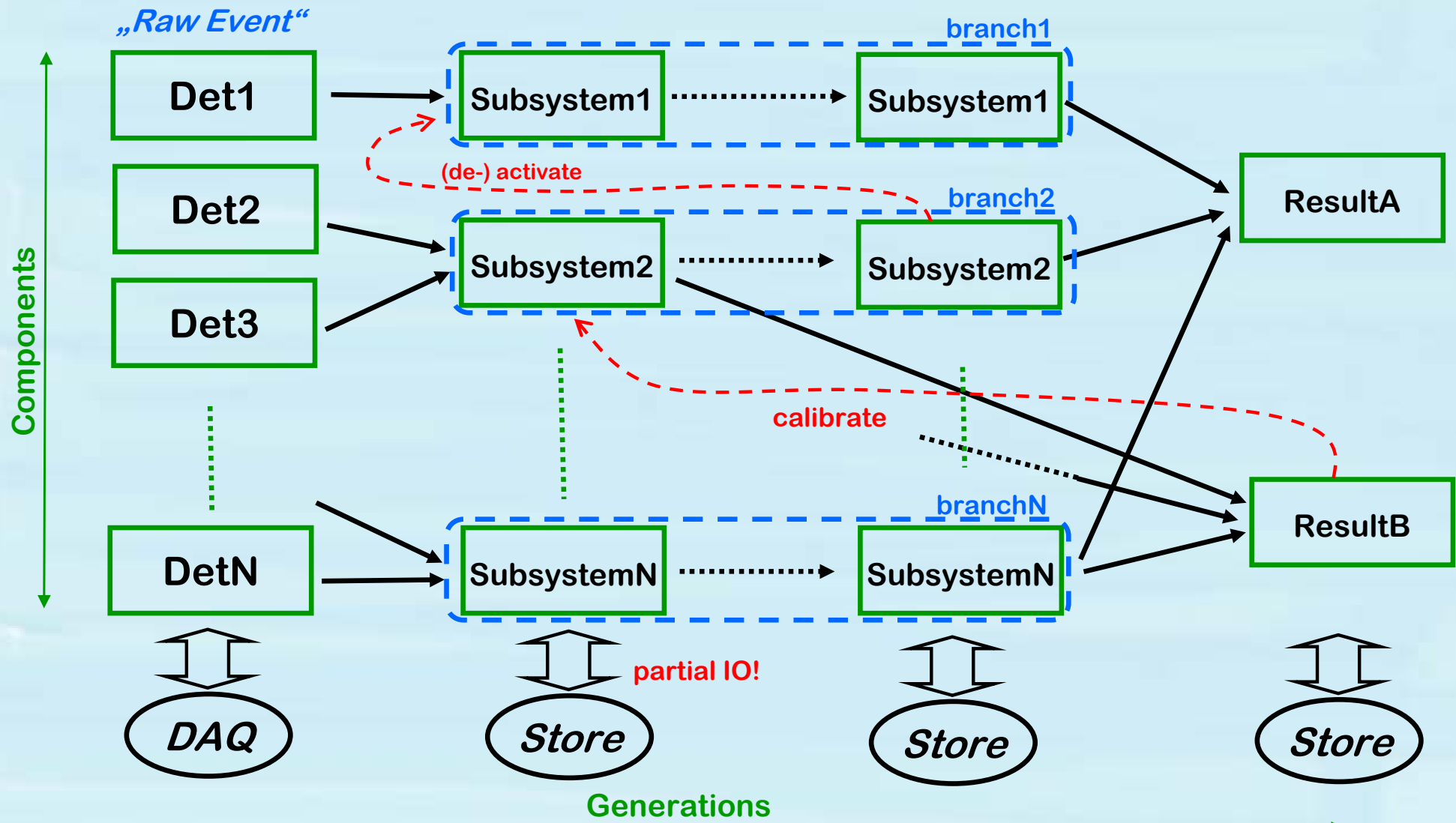


The problem

- **Modular** analysis
- Modules must be **set up** and **controlled**
(IO, processing, interactive, GUI)
- **General purpose framework for different experiments**



The problem: General analysis organization



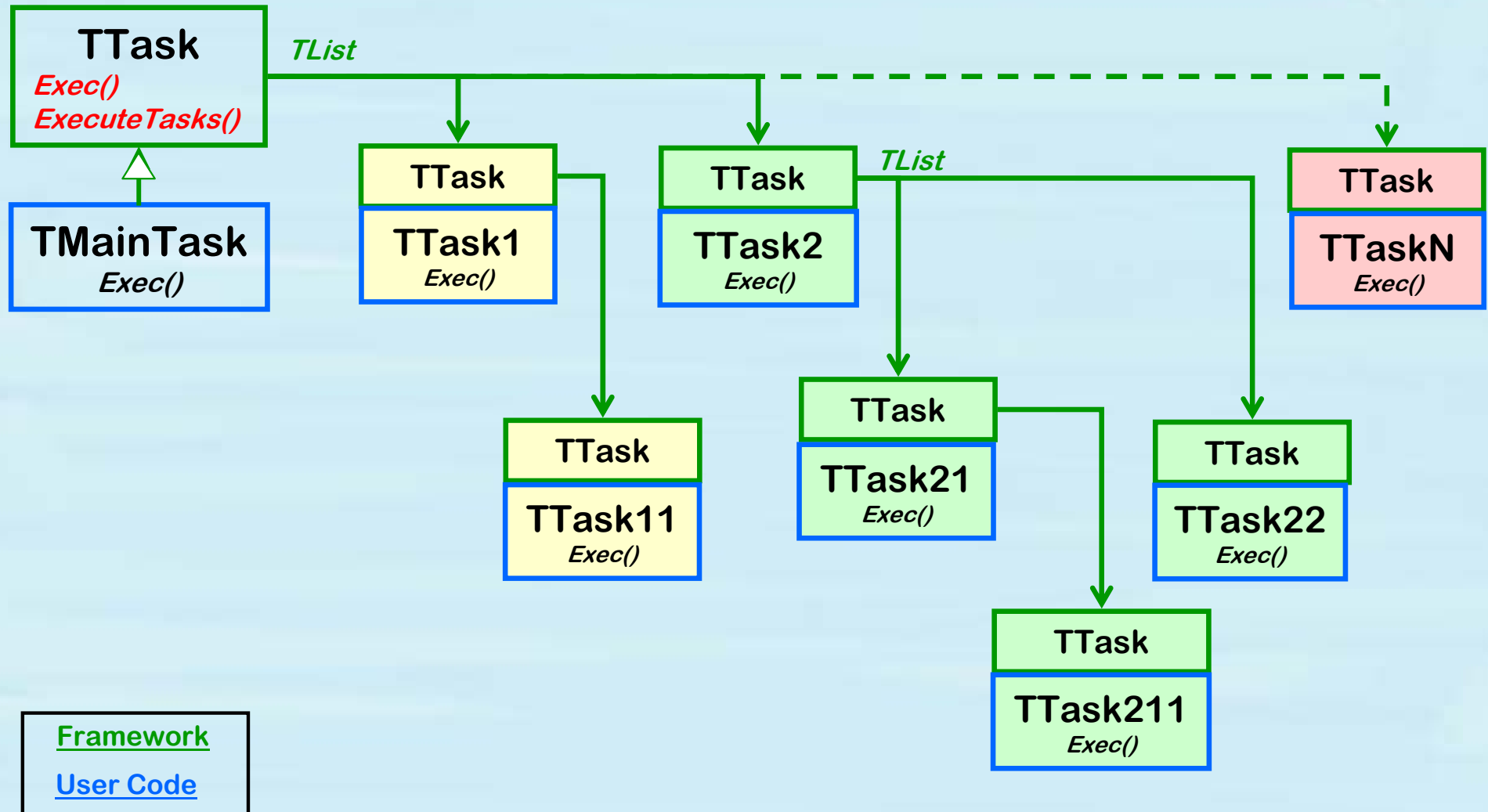


General analysis organization: use cases

- **Components** analysis may run as **independent branches**
each branch may be switched off
- **Store and retrieve** events / data structures of **each generation/each component**
partial IO for components
- **Set-up** (active components, range of generations, „calibration“ data)
should be easy *config script, GUI*
=> Framework with basic definitions required!

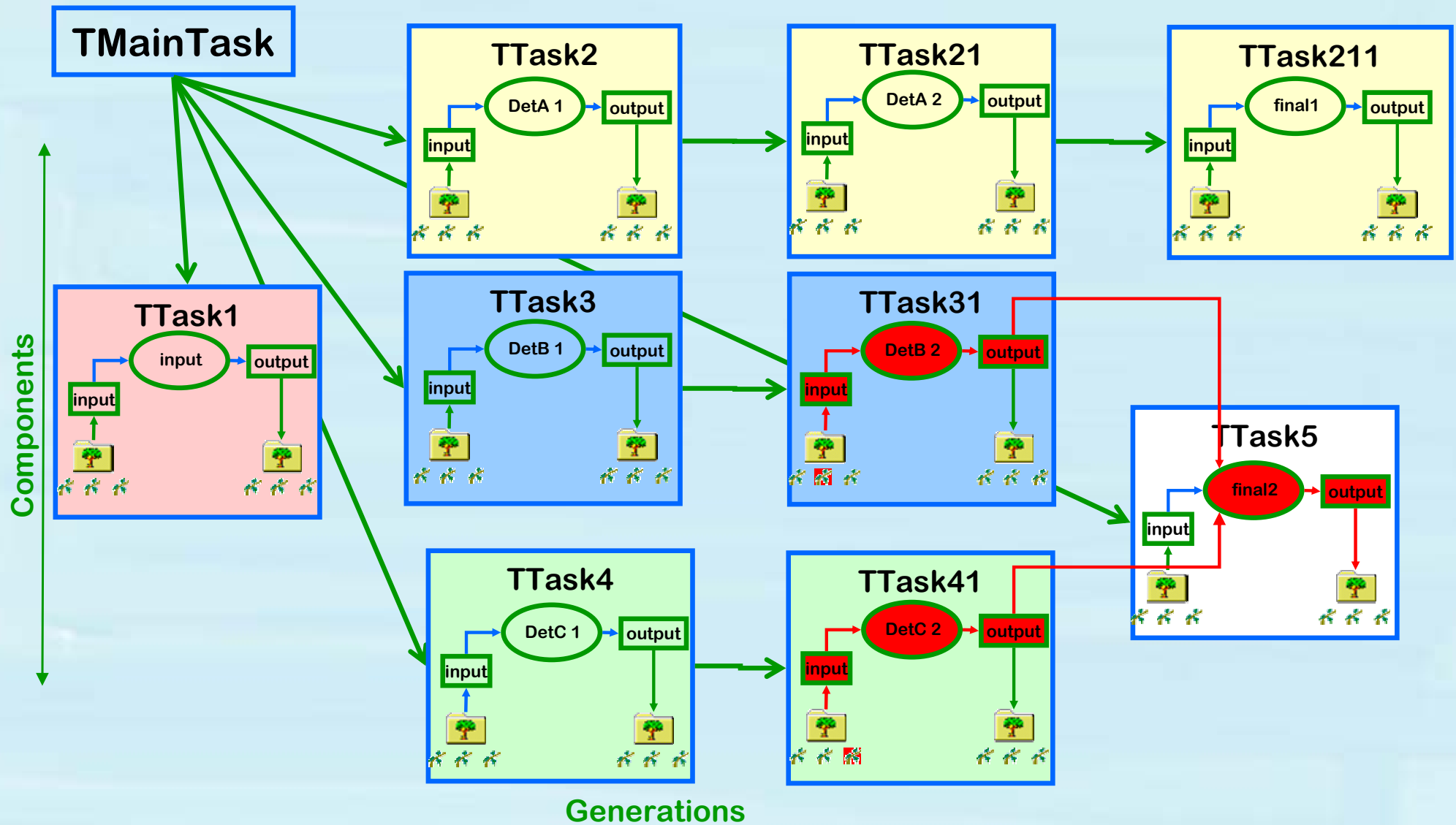


ROOT TTask mechanism



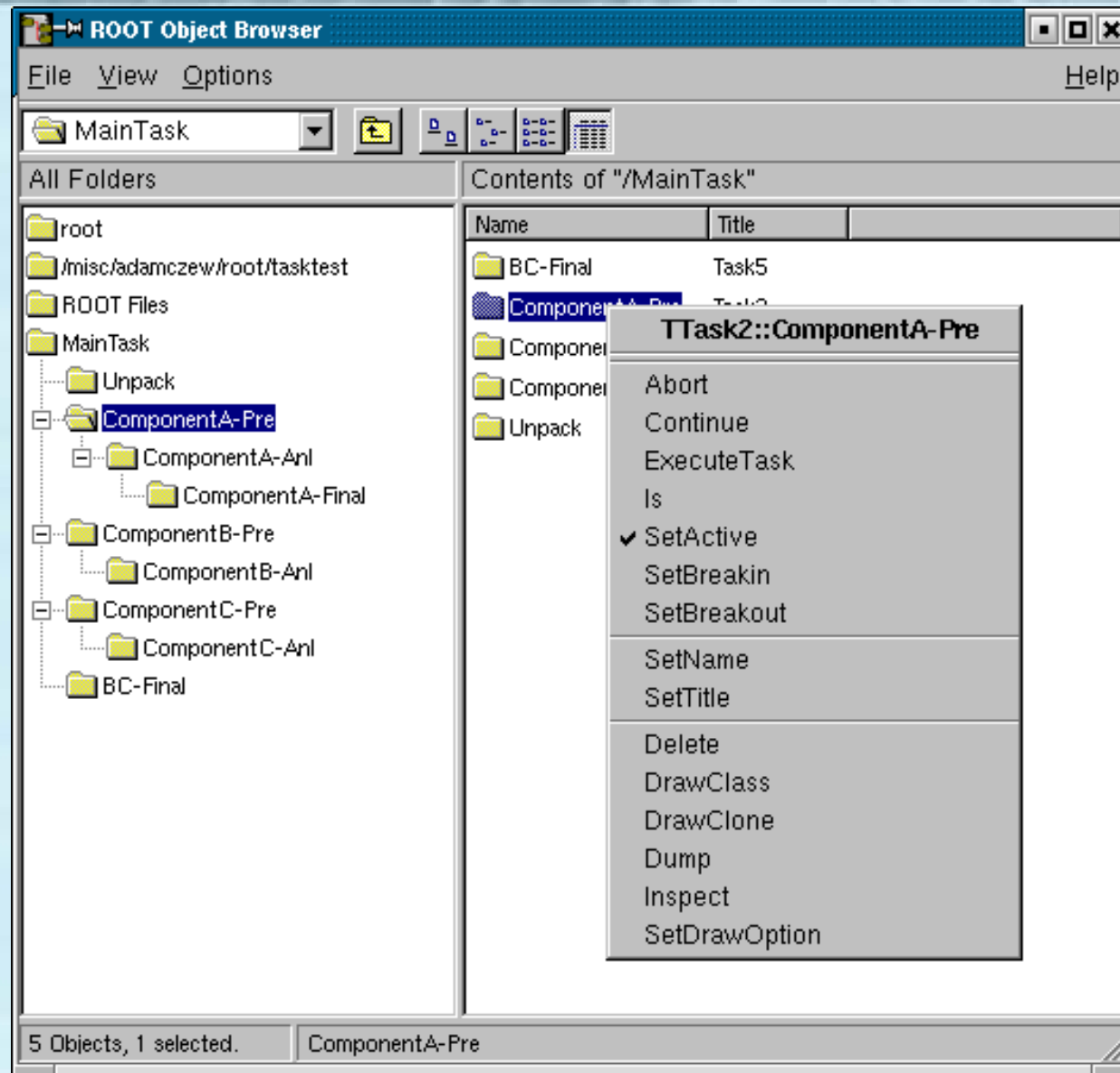


Logical mesh of TTasks





TTask control with TBrowser





ROOT TTask mechanism

- **Lightweight and flexible** framework: Inheritance of **one class** TTask
- **independent branches** and subbranches of tasks
- **GUI control** via TBrowser (branch view, activation, breakpoints)
- **data interface** between **dependend tasks** not given
*user resonsibility! **subframework?***
- **IO interface** of intermediate results not given
*user TTree, GUI controlled enable/disable? **subframework?***

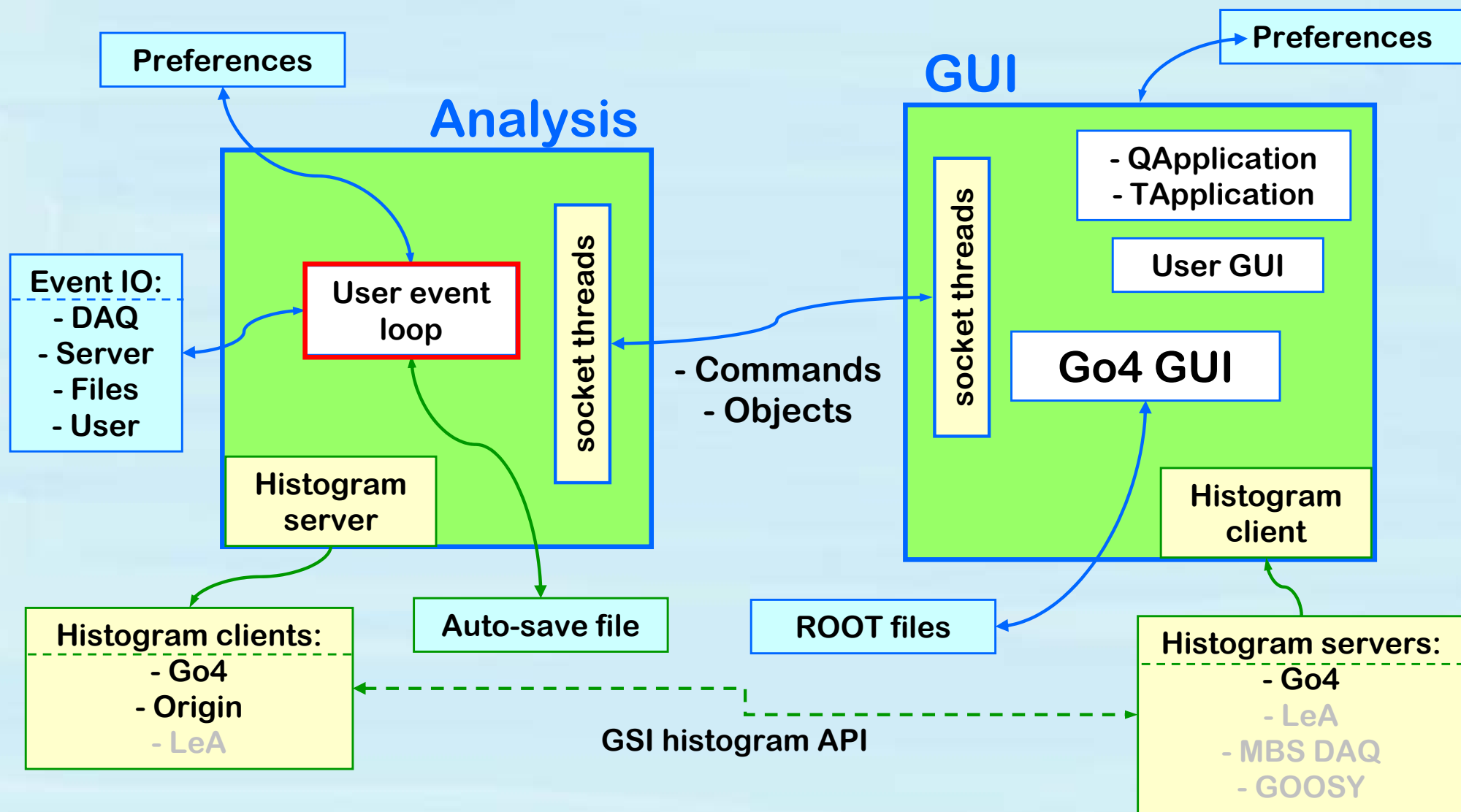


The Go4 (GSI Object Oriented Online Offline)

- **Framework** for many kinds of experiments (Atomic & Nuclear Physics)
- The analysis is written by the user (**unlimited ROOT**)
- **Services** (GSI DAQ, *analysis organization*, IO, ...) **are provided**
- **Batch mode** (CINT or compiled, off-line)
- **Interactive mode** (on-line or off-line):
 - A **non blocking GUI controls and steers the analysis**
 - Analysis may **run permanently** and can **update graphics asynchronously**
 - ROOT object transport between **analysis and GUI process**, multithreaded, sockets
 - GUI interfaces **ROOT and Qt graphics**
 - User defined GUI possible (Qt designer)

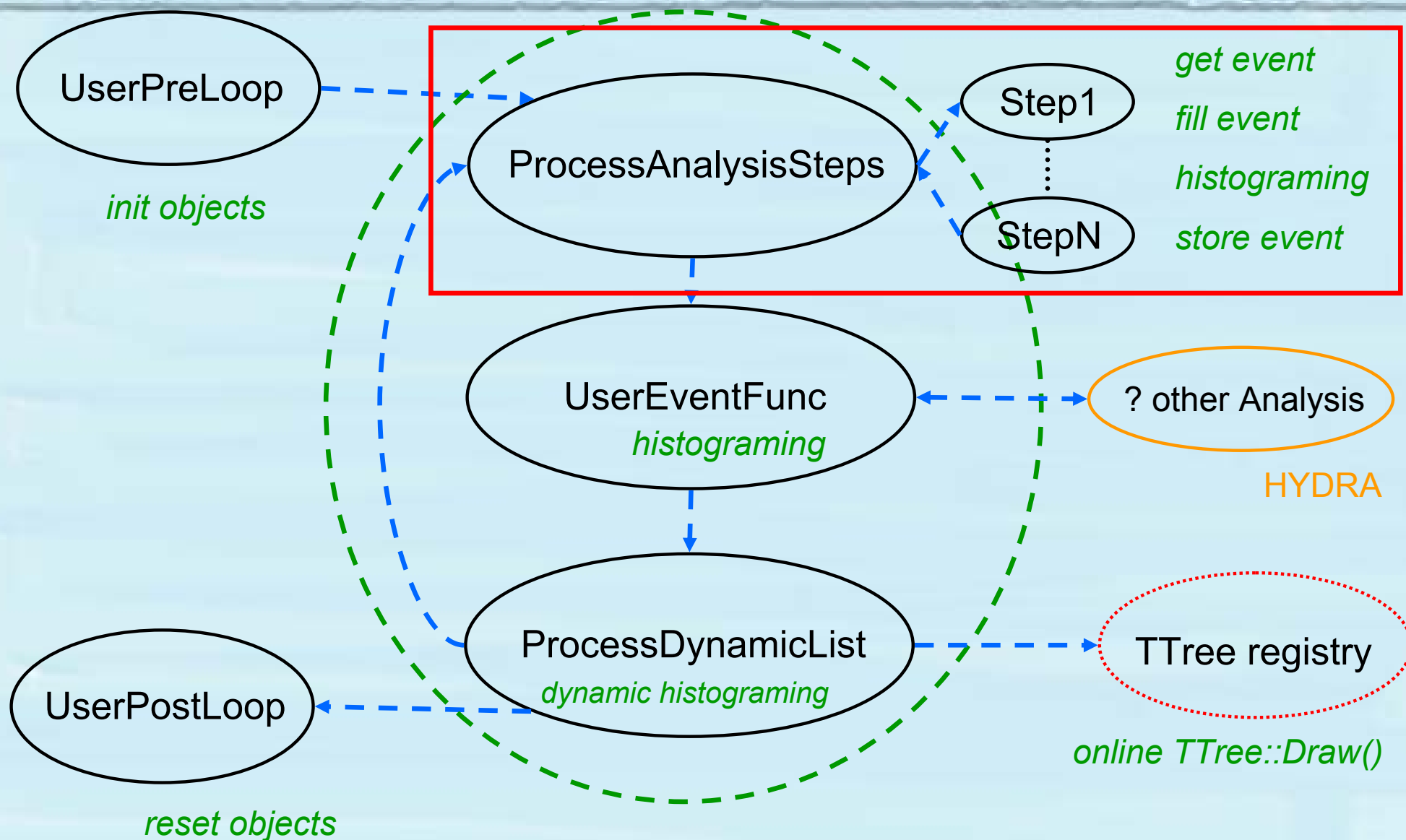


Go4 Processes: GUI & Analysis



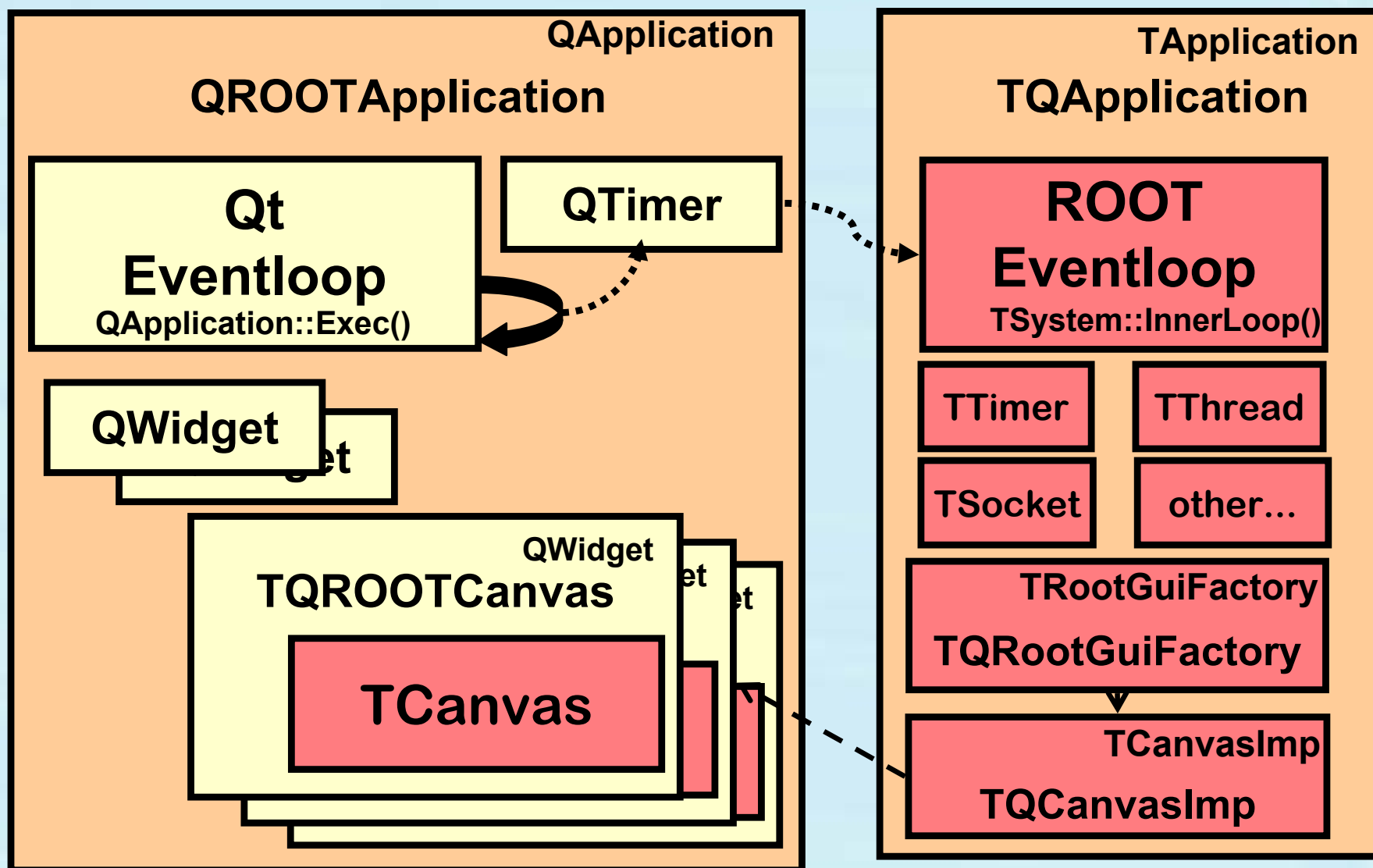


Eventloop Actions





Go4 GUI: ROOT Qt interface to ROOT





Go4 Releases

- Go4 v1.x GSI internal release (May 2002)
- Go4 v2.0 public release (Nov 2002)
- Go4 v2.1 public release (Jan 2003)
- Go4 v2.2 public release (Mar 2003)
- Go4 v2.3 public release (May 2003)
- Go4 v2.4 public release (Aug 2003)
- Go4 v2.5 public release (Dec 2003)

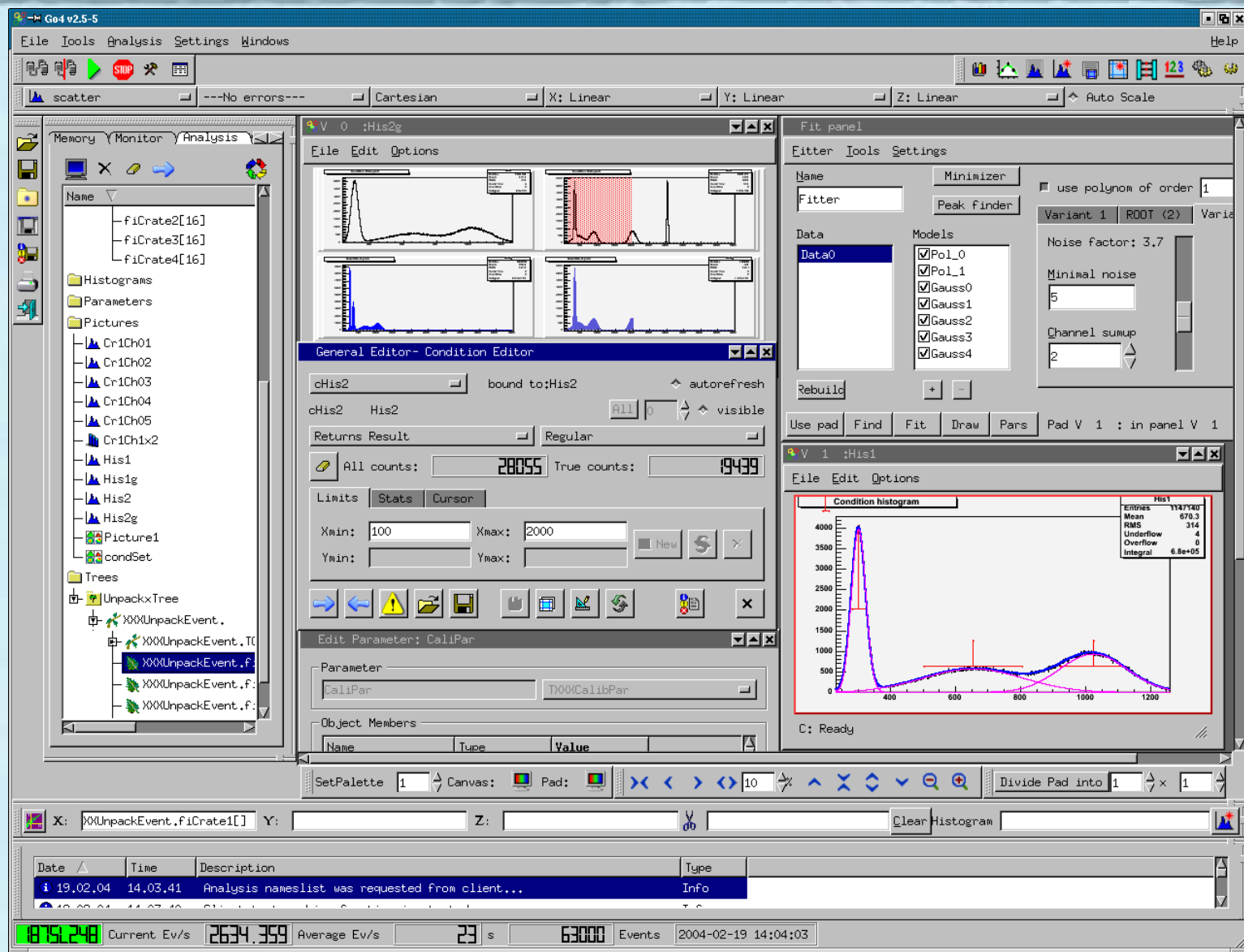
Linux: Debian 3.0, Suse 8.1, Suse 8.2, RedHat 7.3, RedHat 9.0

Compiler: gcc 2.95, gcc 3.2, gcc 3.3

Users: GSI-FRS, SHIP, Euroball/RISING,
HADES (online monitor), Atomic physics,...



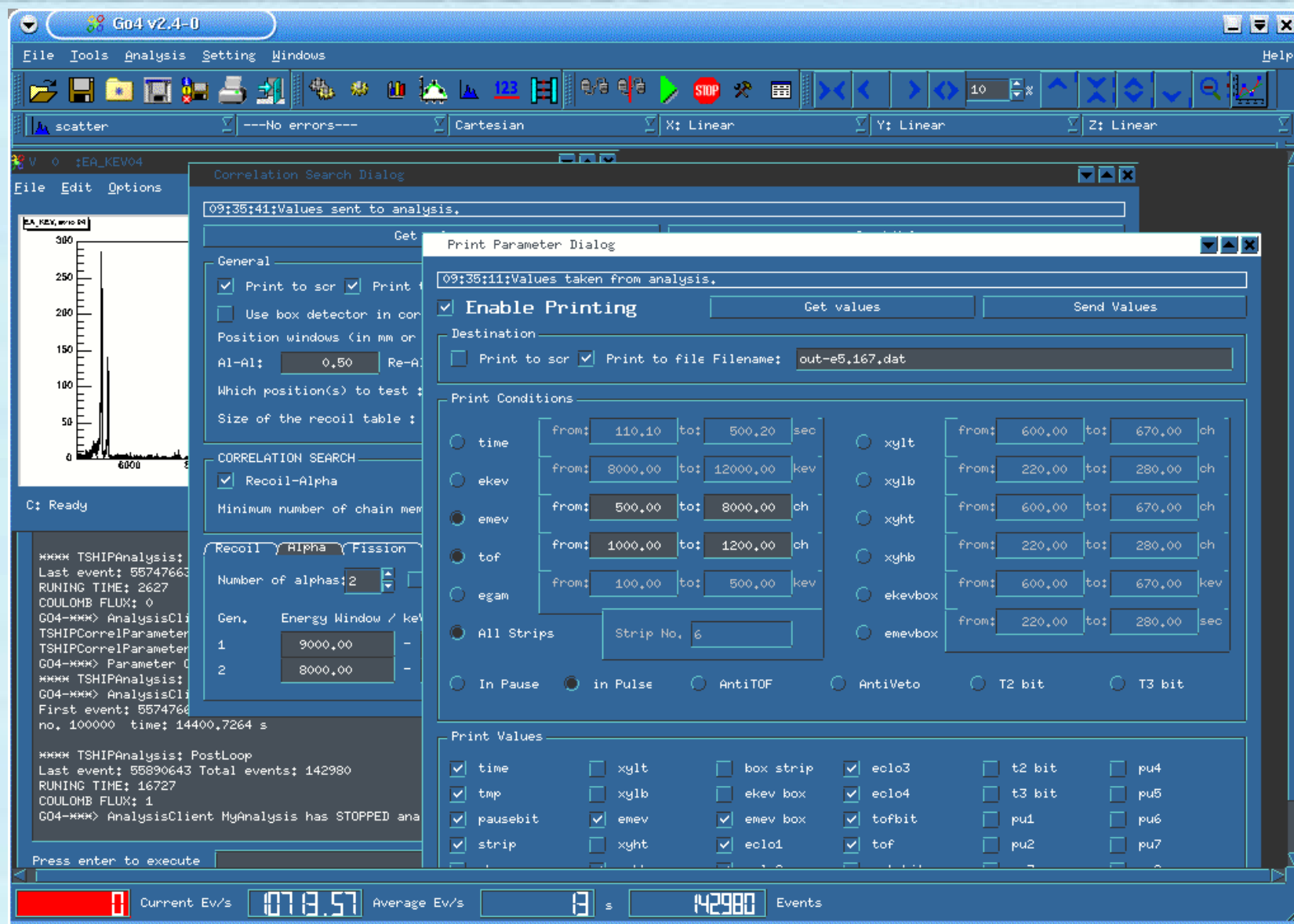
Standard GUI





User GUI (Qt)

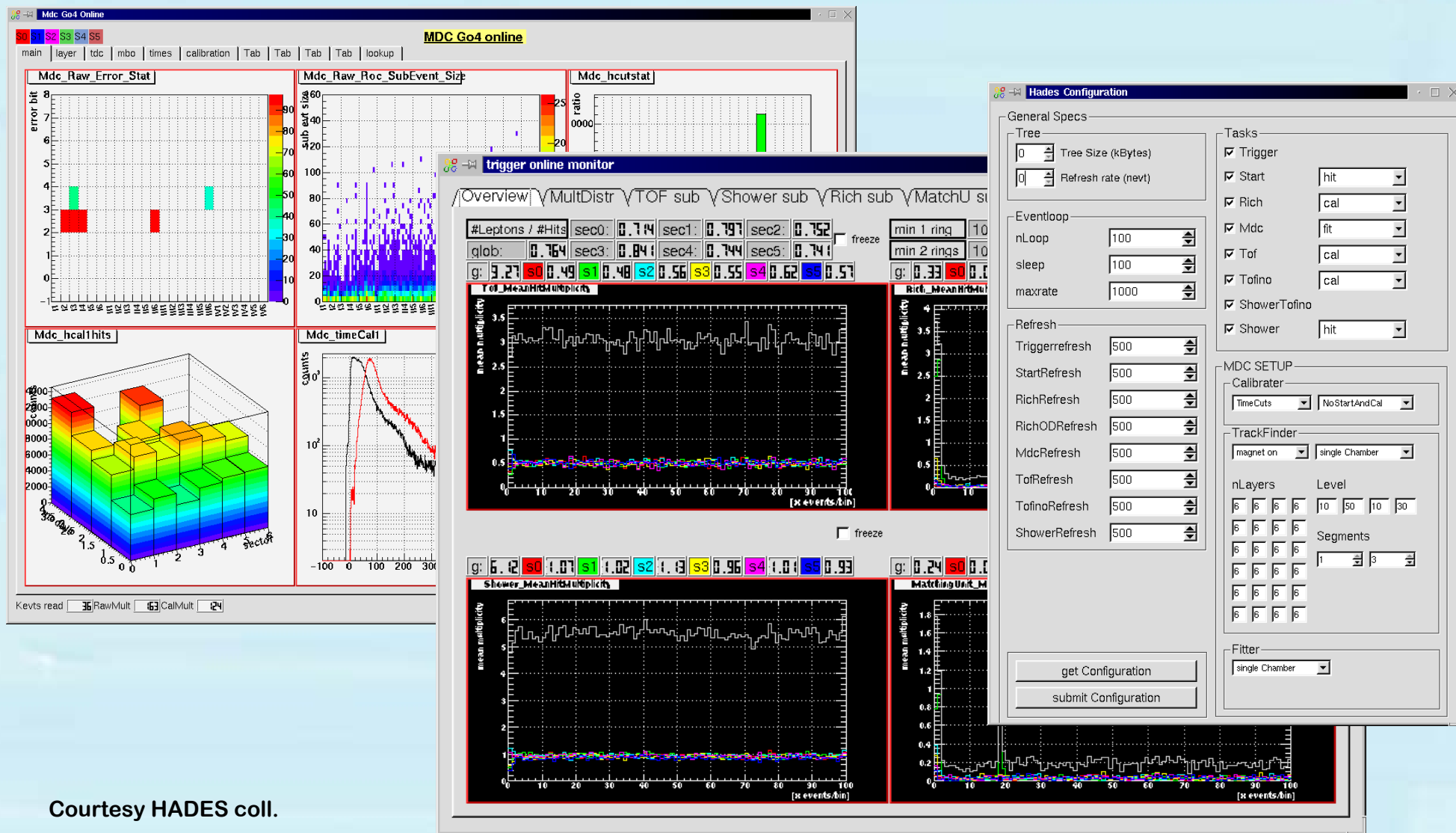
Parameter Setup for SHIP





User GUI (Qt)

On-line monitoring of HADES



Courtesy HADES coll.

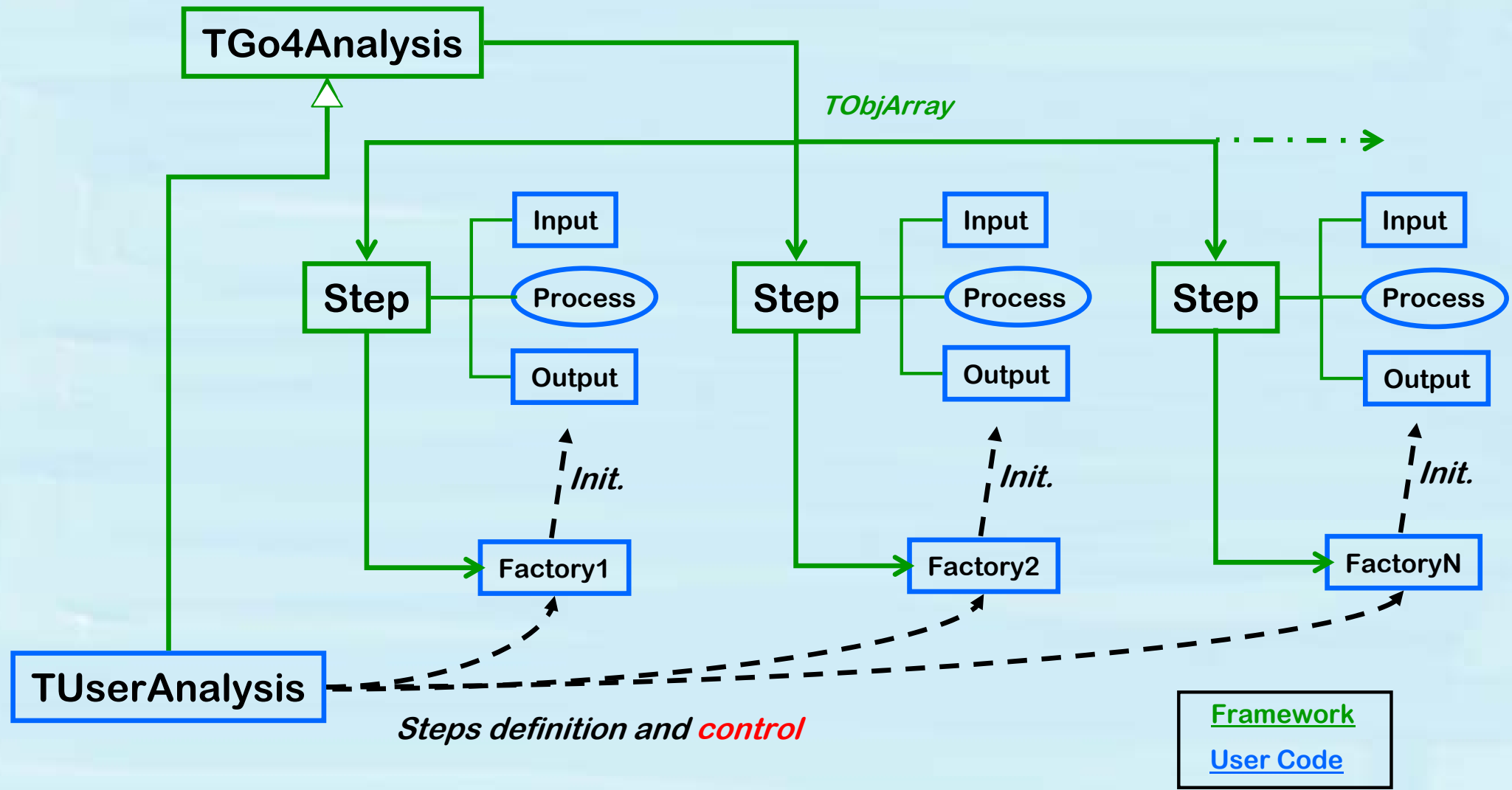
Feb-2004

J.Adamczewski - Go4 -
<http://go4.gsi.de>

17

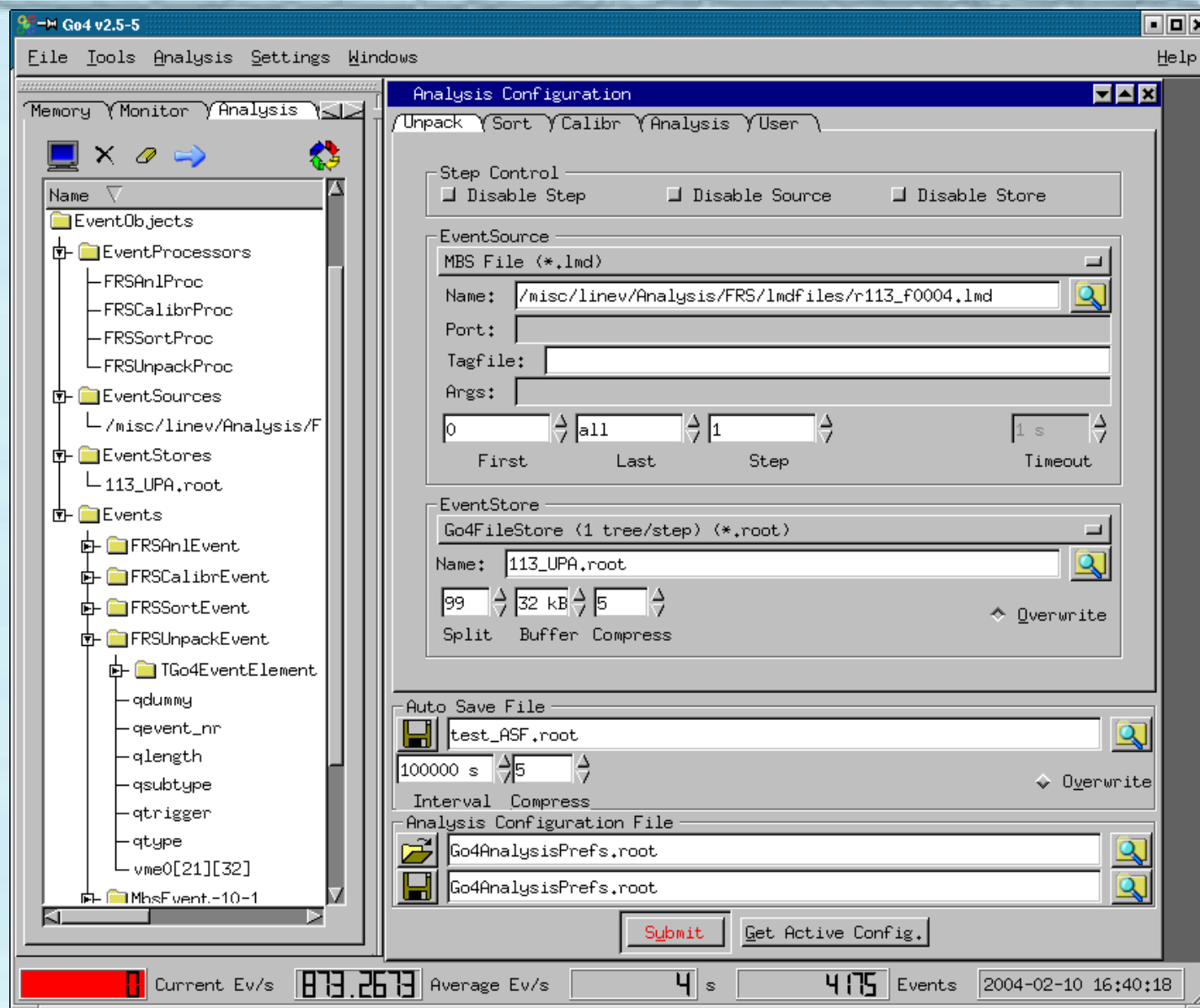


Go4 Analysis steps





GUI control of steps





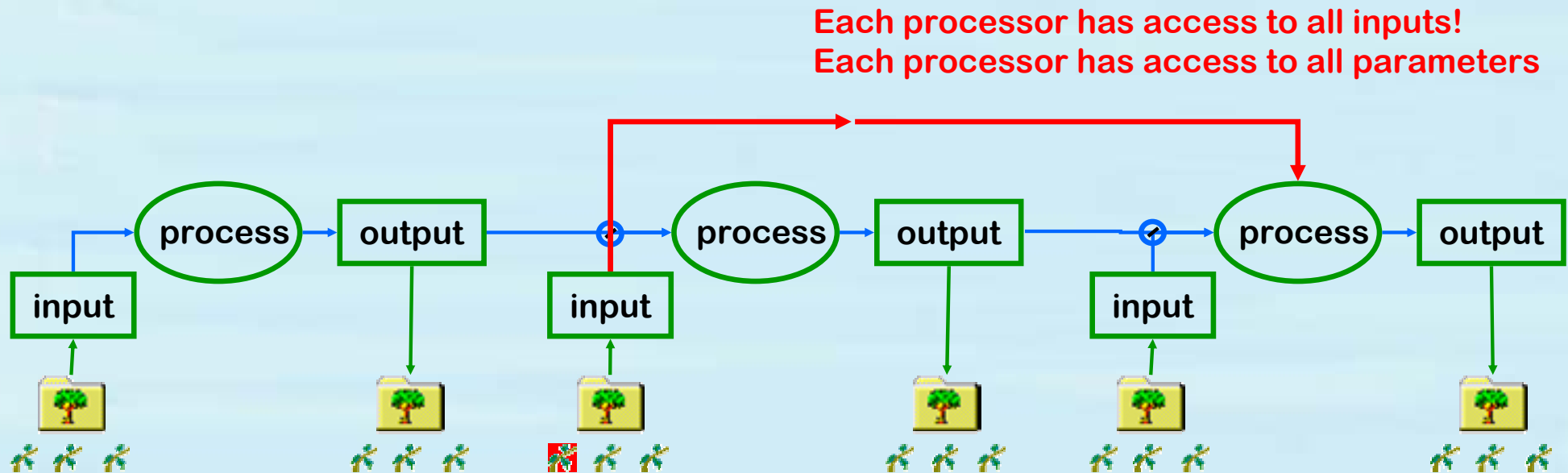
Go4 Analysis steps

Chain of analysis steps processed **sequentially**

Each step can be **en/disabled** (framework)

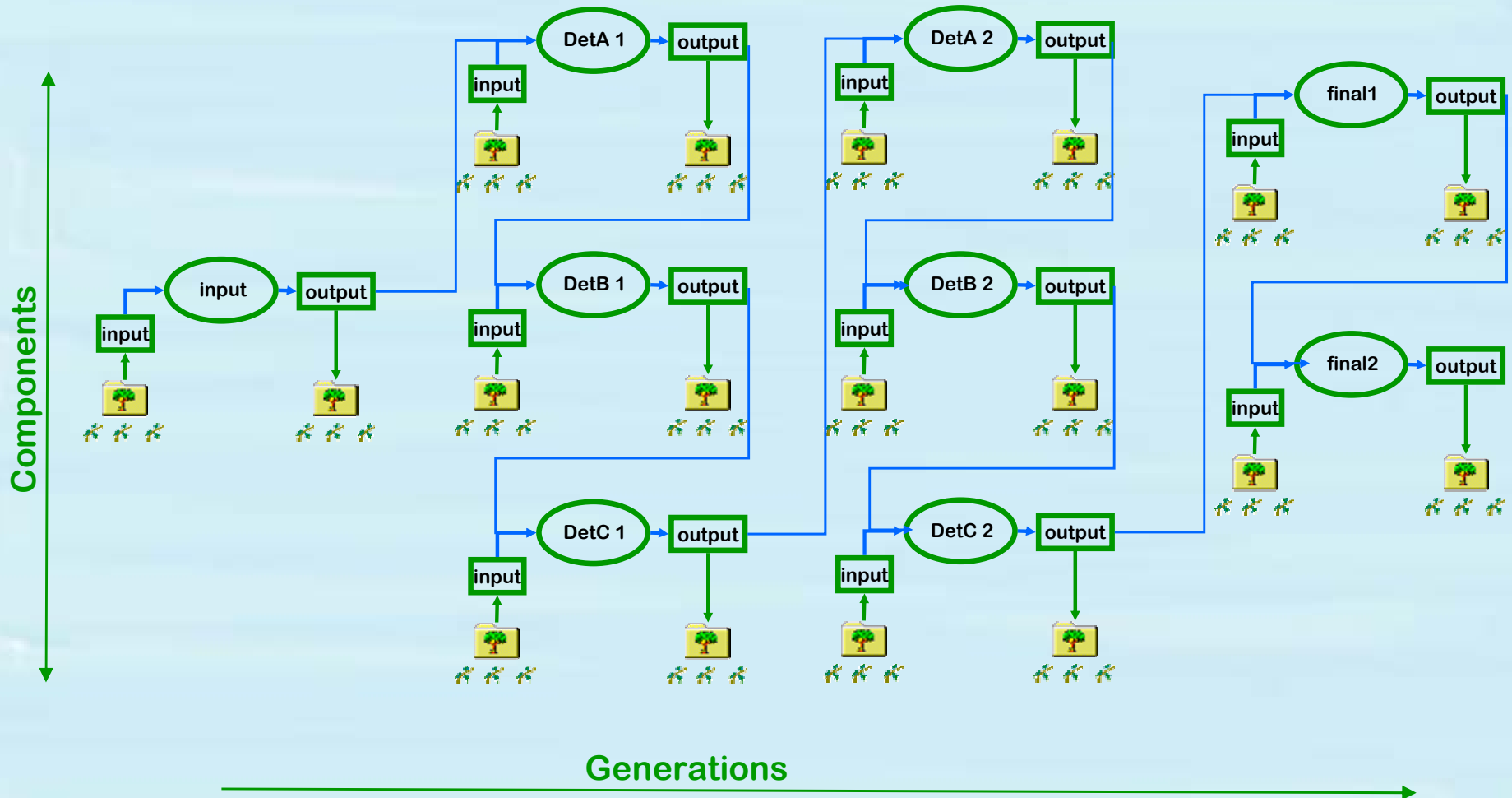
Input/output can be switched (framework)

Partial IO (steered by **application**)



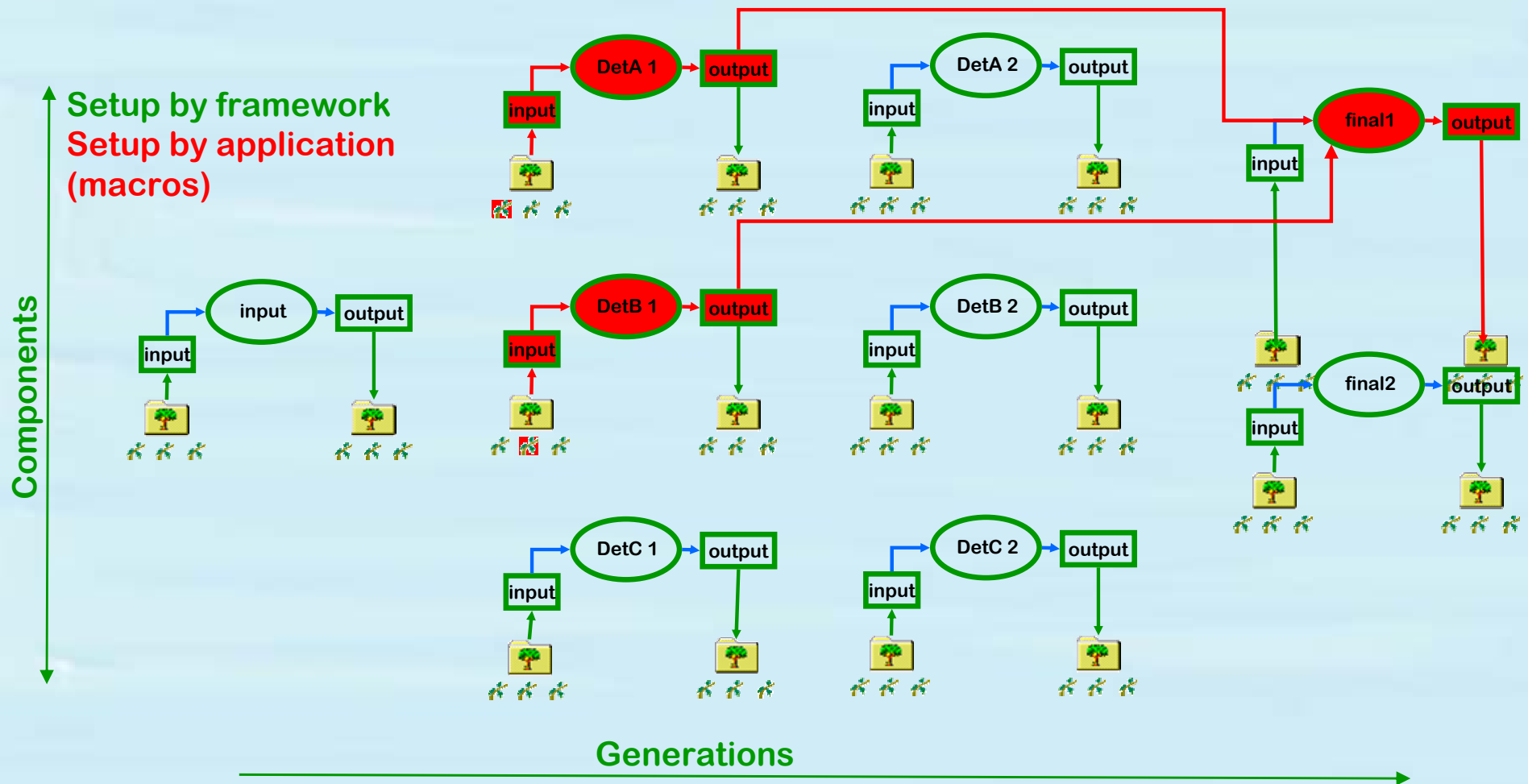


Logical mesh of steps





Logical mesh of steps





Setup by framework
Setup by application (macros)

Components

Generations



Go4 analysis organisation

- Designed for linear flow of analysis, generation oriented
- Abstract Interfaces for IO, data structures, processing
- User defined factory for each step
- Fully controlled by framework (GUI, macros)
- No hierarchy of substeps (no execution branches)
- Control of multiple inputs for one step not supported by framework
- Logical mesh setup possible, but not yet controlled by framework

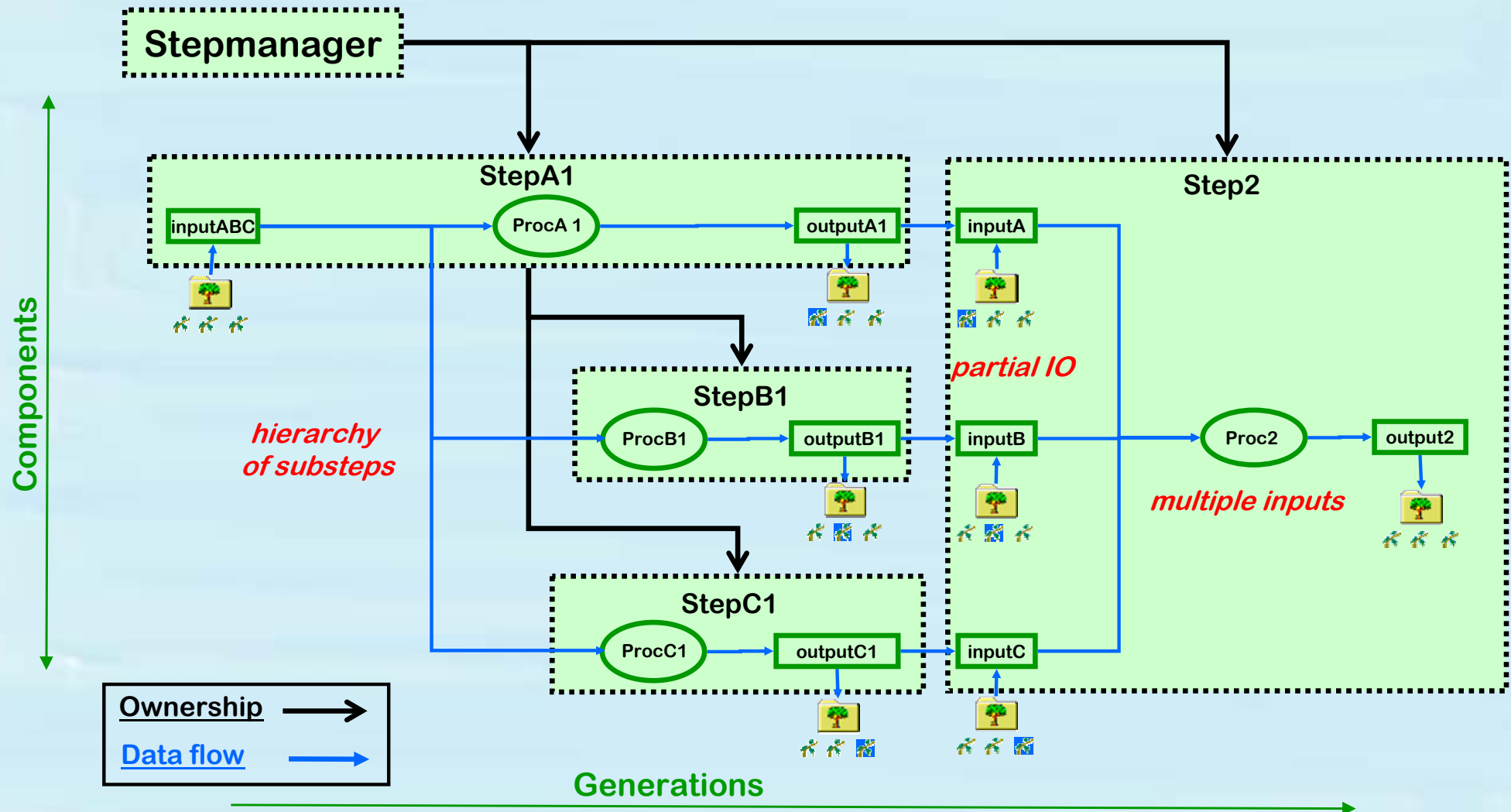


Further Go4 developments

- **Advanced framework interfaces:**
 - Control of **multiple IO** for each step
 - Control of data flow in **analysis mesh**
- Analysis steps redesign for **hierarchical structure** (TTask subclass!)
- Extend standard **analysis configuration GUI** for above cases



Further Go4 framework developments





The End.
