



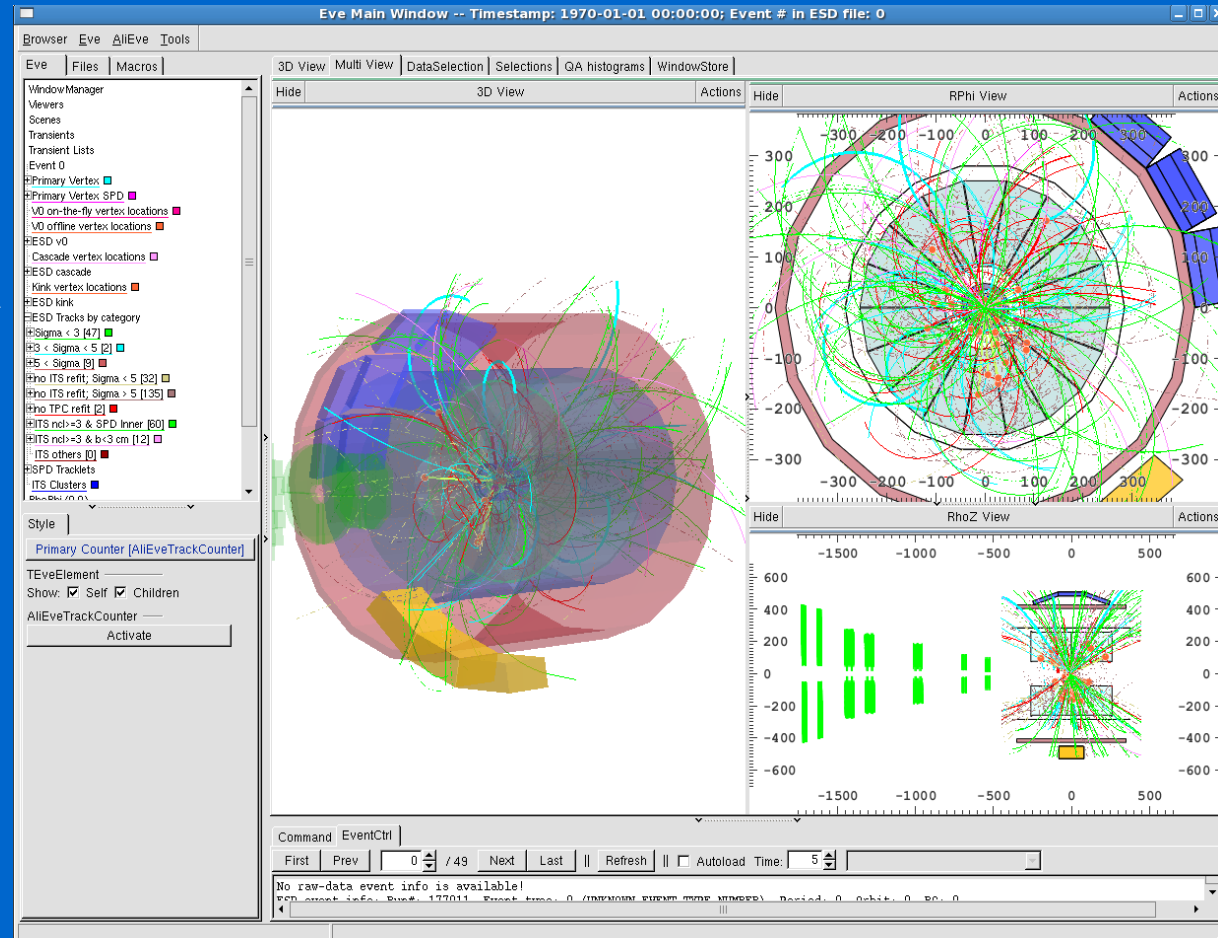
ALICE

Improvement of the ALICE Online Event Display using OO patterns and parallelization techniques

Mihai Niculescu
(Institute of Space Science Romania)
For ALICE Collaboration

A bit of History (1/2)

- ALICE Event Display (alieve) was mainly developed by Matevz Tadel and Alja Mrak-Tadel
- Experiment *dependent*: data loading and processing algorithms:
 - Detector geometry, simulation & reconstructed data
 - Raw and digits data
 - High-level trigger events
 - Visualization scripts
 - Online source



A bit of History (2/2)

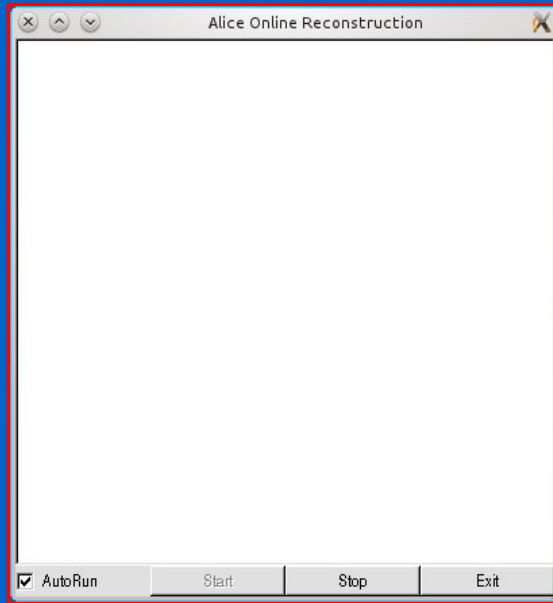
- Experiment *independent* code was ported to ROOT (REVE) – GUI and OpenGL – now part as a ROOT module
 - Application core
 - Management: obj-browsers, 3D scenes and viewers, events and event navigation
 - Environment for CINT scripts
 - Visualization classes (geometry, extracted shape data, hits, clusters, tracks)

Used by: ALICE, CMS, FAIR, K2K, NA62

The Online Event Display in ALICE



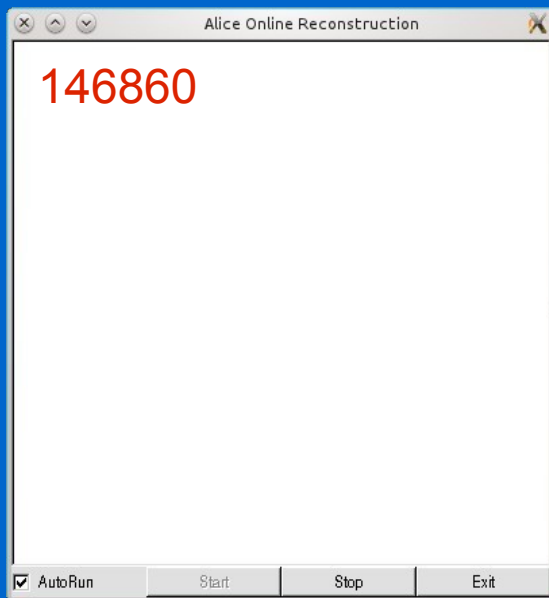
DIM
(Distributed
Information
Management)



The Online Event Display in ALICE



DIM
(Distributed
Information
Management)

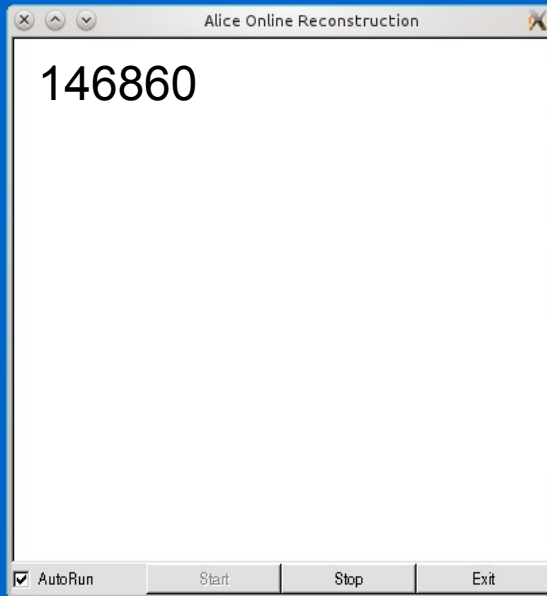


Forked process

The Online Event Display in ALICE



DIM
(Distributed
Information
Management)



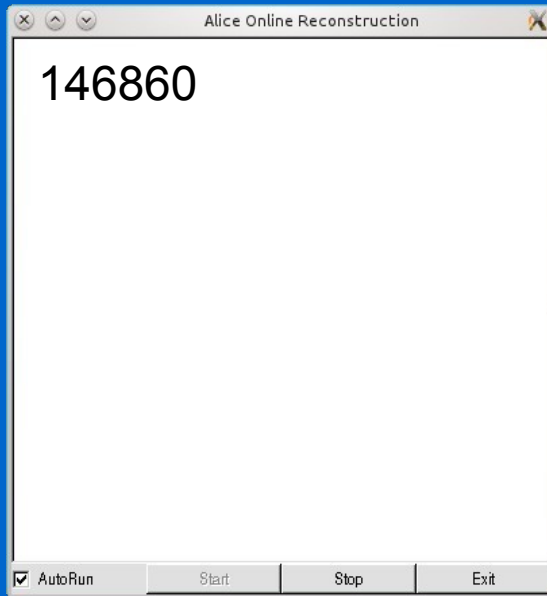
Forked process

TRint application

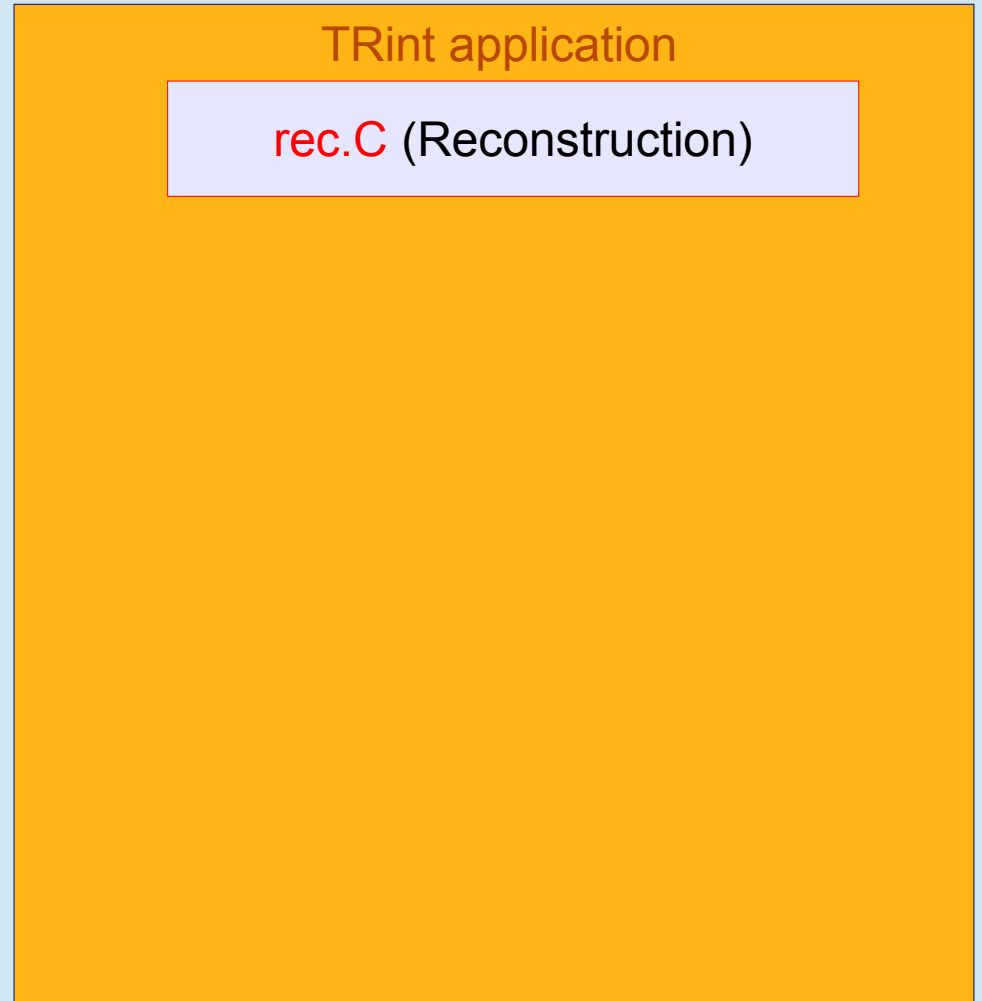
The Online Event Display in ALICE



DIM
(Distributed
Information
Management)



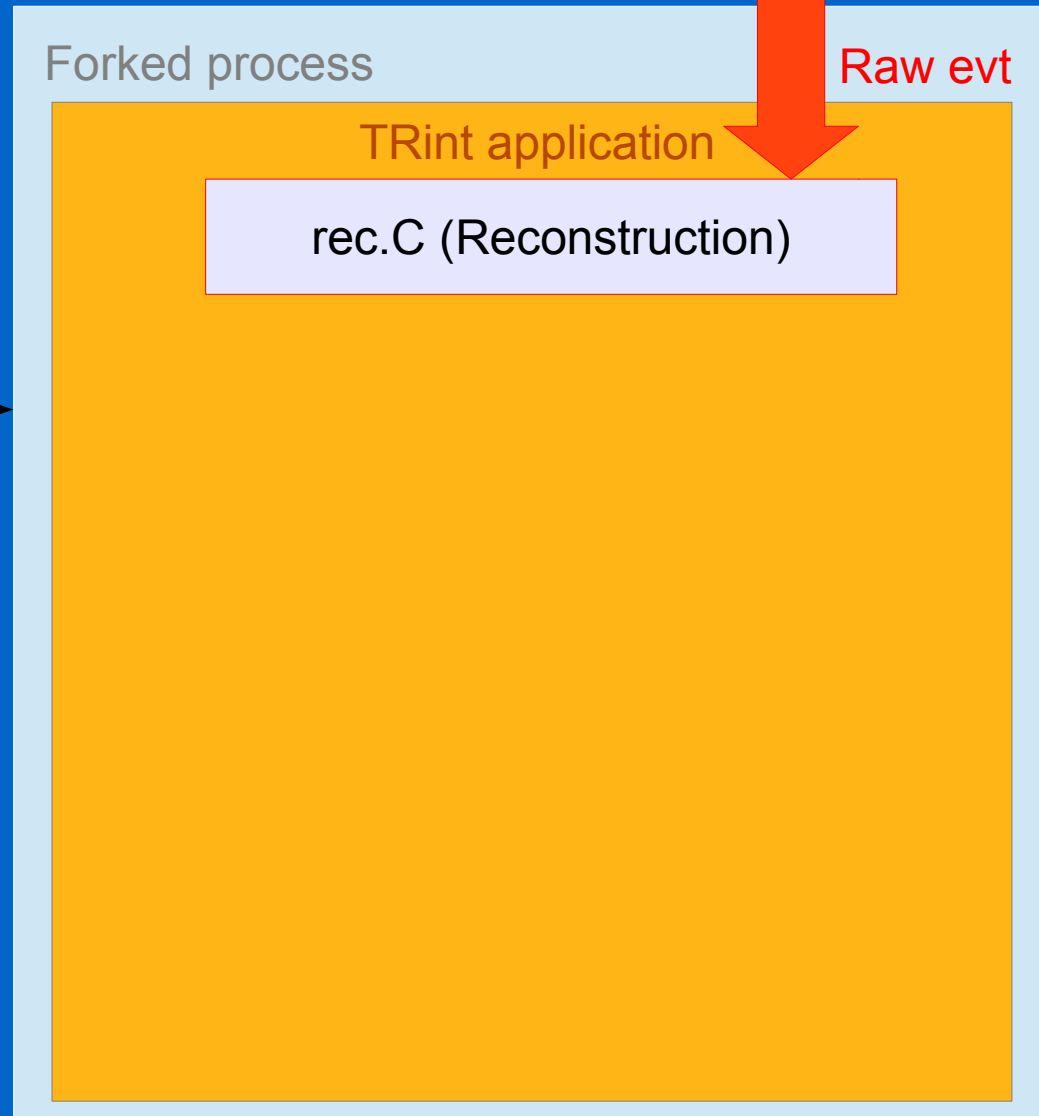
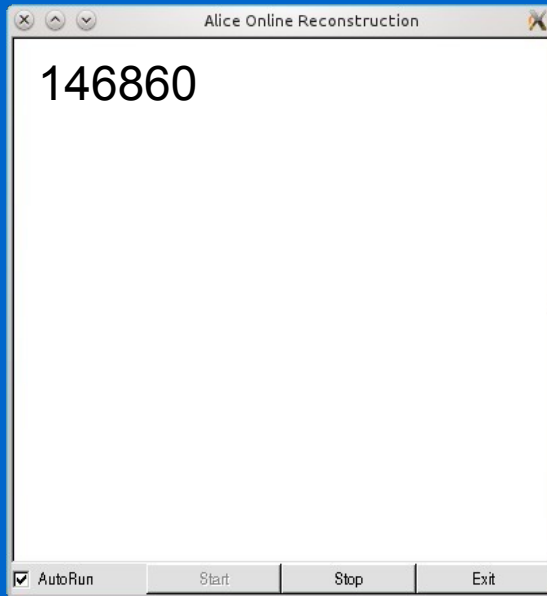
Forked process



The Online Event Display in ALICE



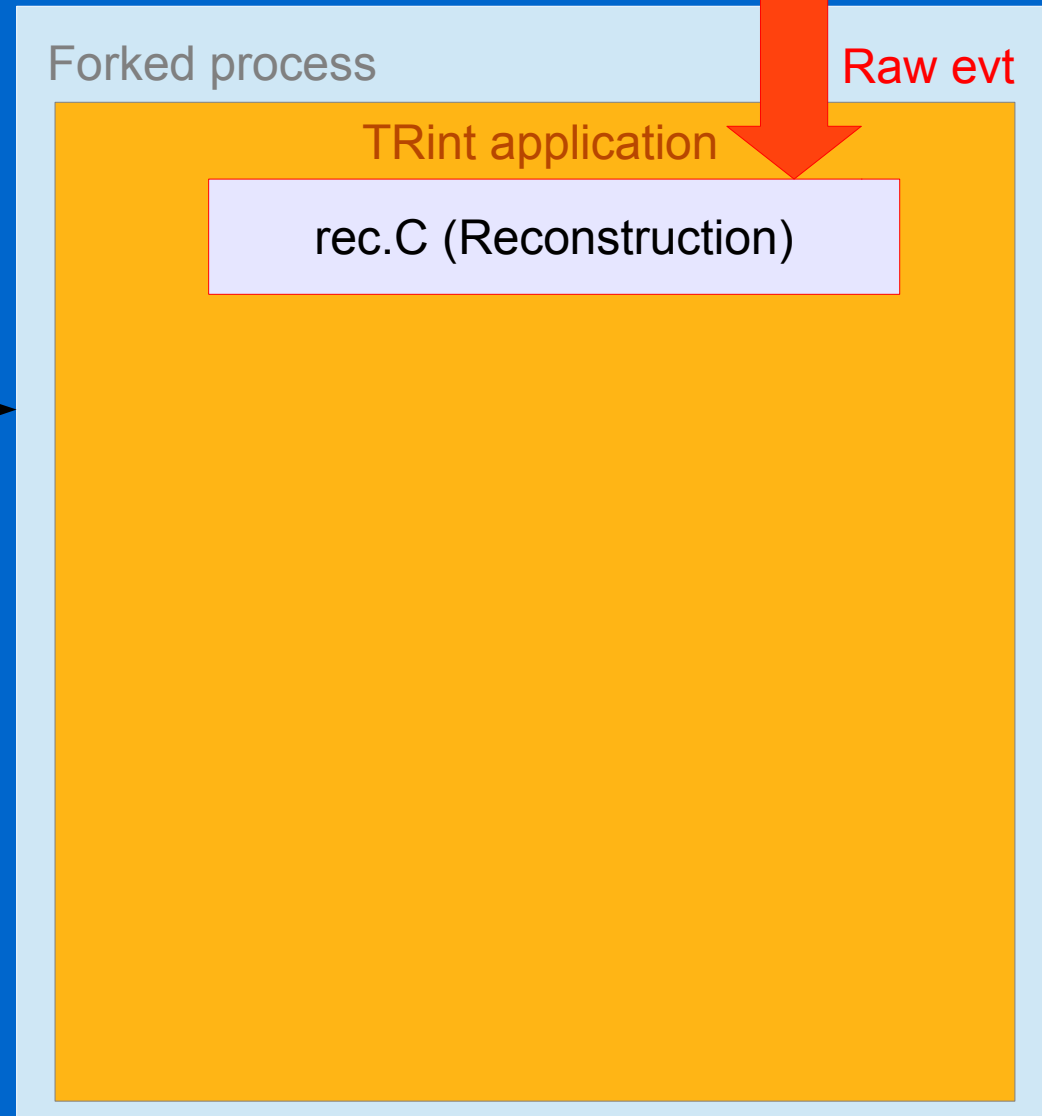
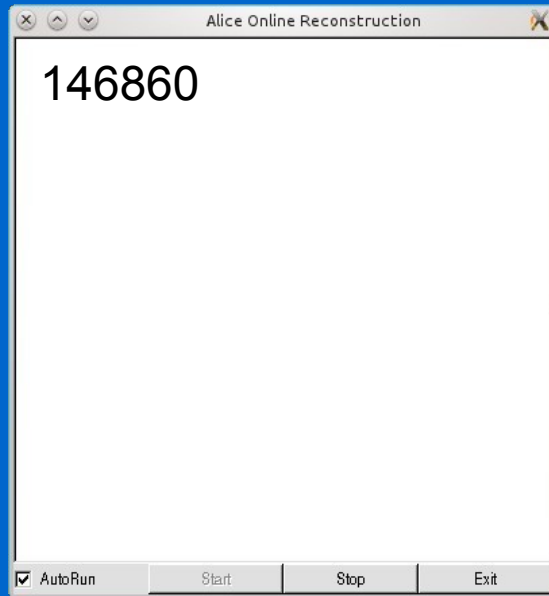
DIM
(Distributed
Information
Management)



The Online Event Display in ALICE



DIM
(Distributed
Information
Management)



The Online Event Display in ALICE

DAQ

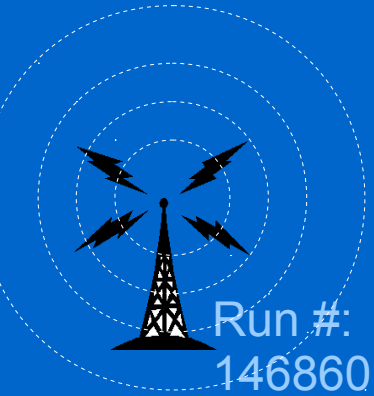
Forked process

Raw evt

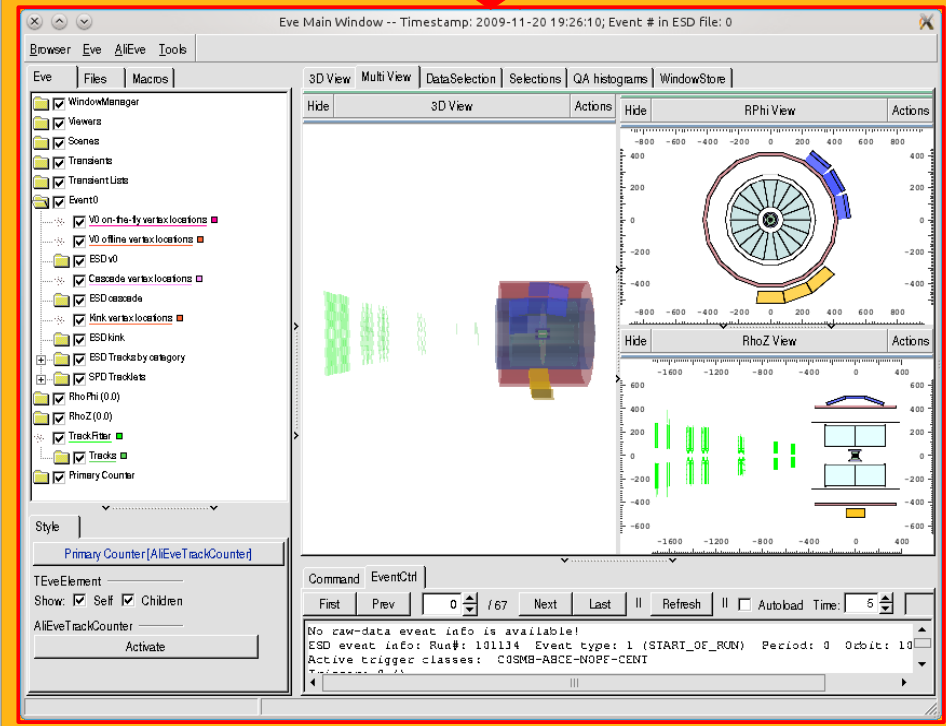
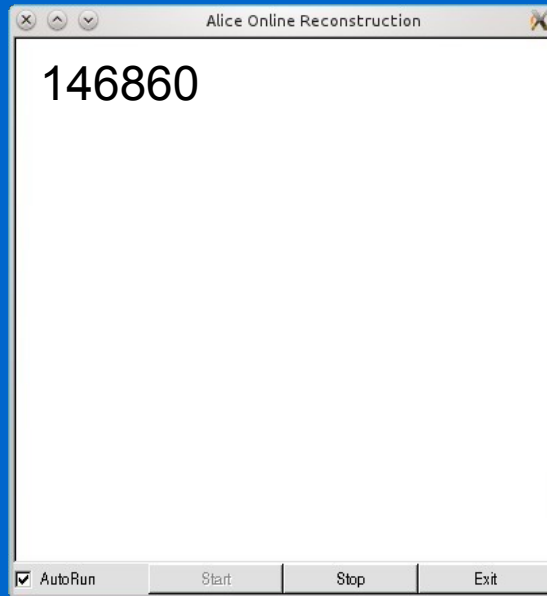
TRint application

rec.C (Reconstruction)

Reco. Evt.



DIM
(Distributed
Information
Management)



The Online Event Display in ALICE

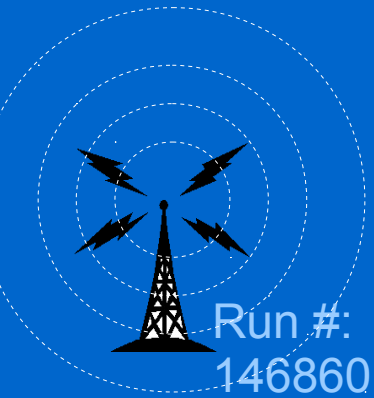
DAQ

Forked process

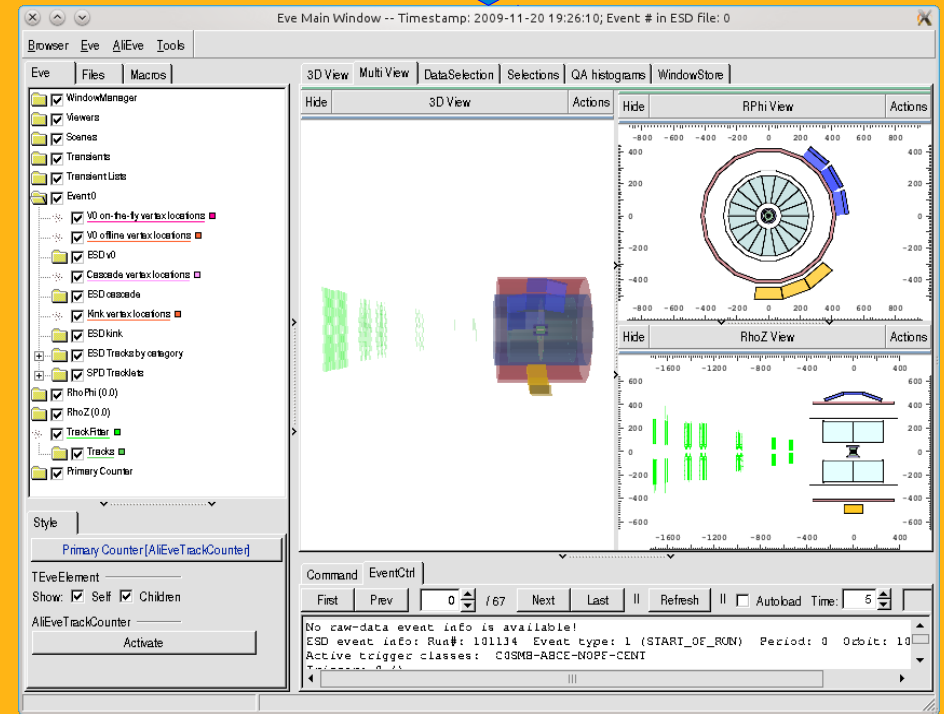
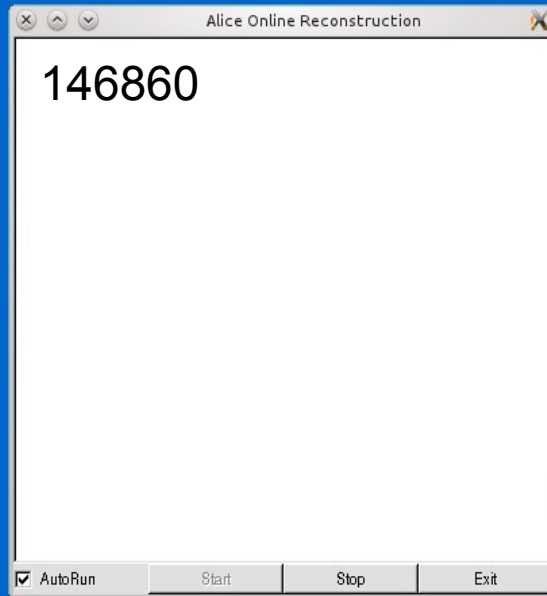
Raw evt

TRint application

rec.C (Reconstruction)



DIM
(Distributed
Information
Management)



Problems with the Current Model

- AliEve GUI **freezes** while reconstruction runs
 - Can't change viz. Options/params
 - Can't take screenshots
- **If reconstruction crashes, so does the visualization**
- Only one **single client** (alieve) to visualize the online reconstructed data

Requirements

- **Decouple** reconstruction & visualization
- Allow **multiple clients** to access reconstructed data

Requirements

- **Decouple** reconstruction & visualization
 - Communication?:
 - **Inter-process**
 - **Inter-threads**
 - **Sockets**
 - **Network file-sharing**
- Allow **multiple clients** to access reconstructed data

Requirements

- **Decouple** reconstruction & visualization
 - **Communication?:**
 - Inter-process
 - Inter-threads
 - **Sockets**
 - **Network file-sharing**
- Allow **multiple clients** to access reconstructed data

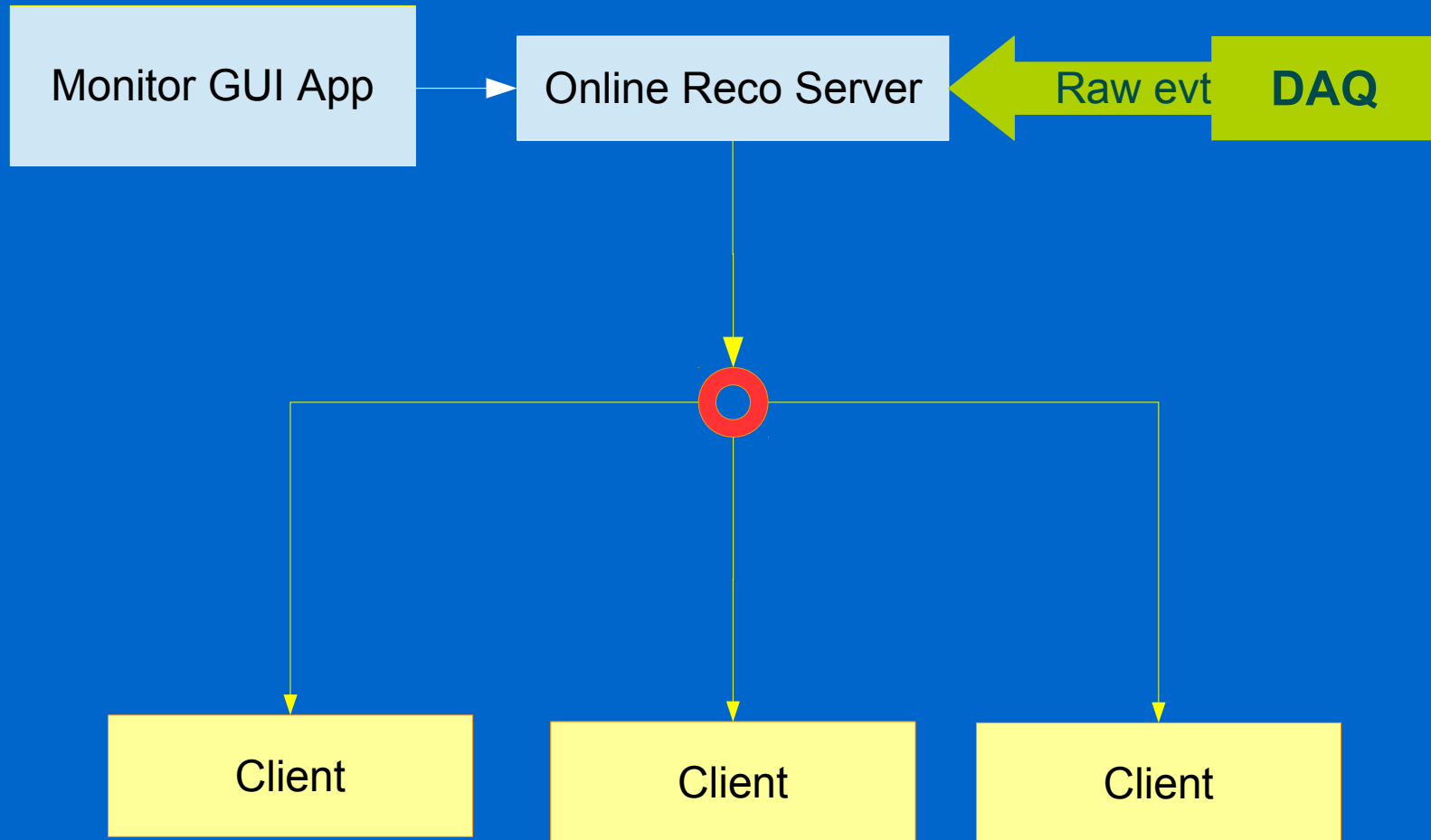
Requirements

- **Decouple** reconstruction & visualization
 - **Communication?:**
 - Inter-process
 - Inter-threads
 - **Sockets**
 - Network file-sharing
- Allow **multiple clients** to access reconstructed data

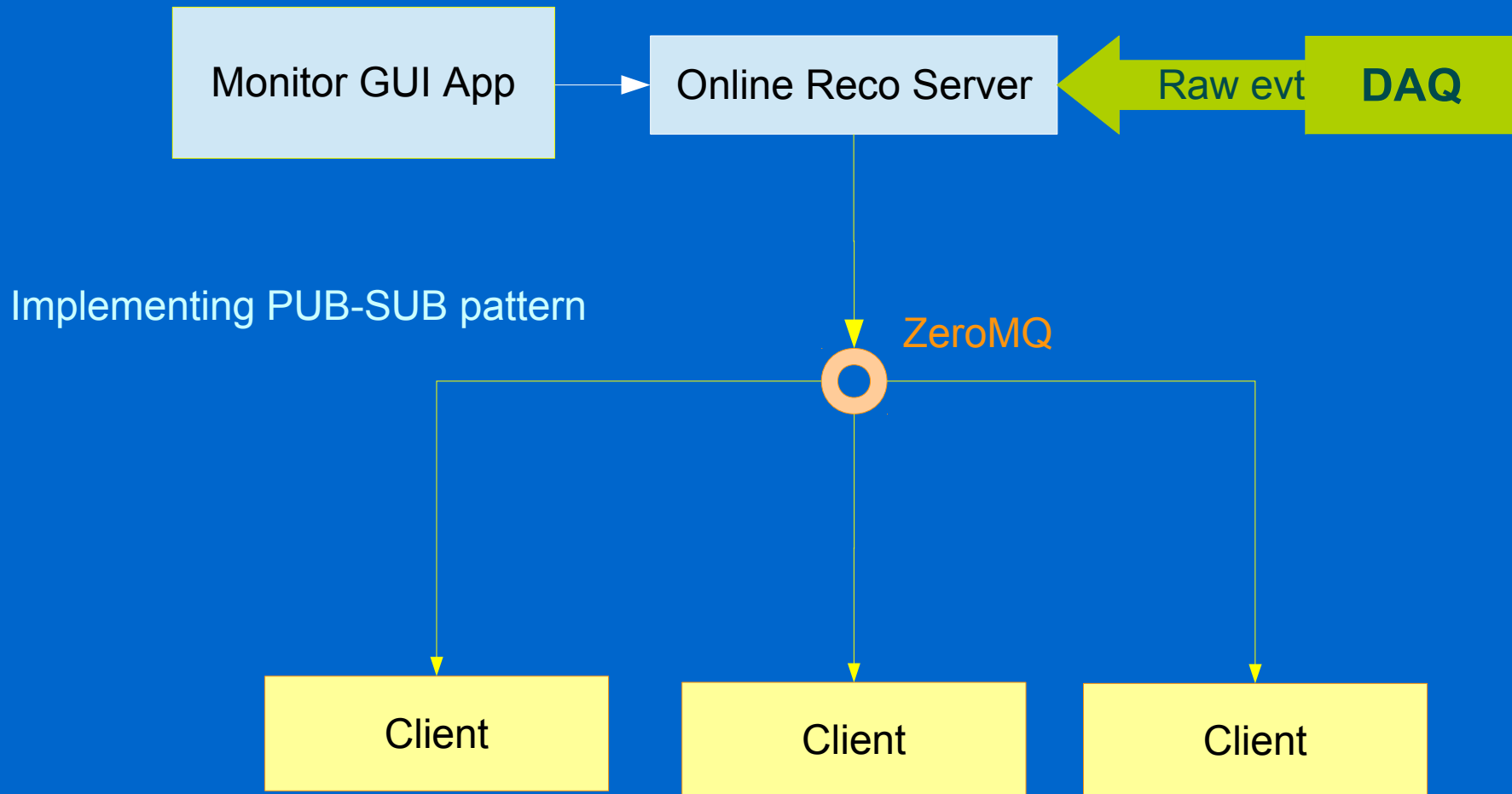
Requirements

- **Decouple** reconstruction & visualization
 - Communication => **Sockets**
- Allow **multiple clients** to access reconstructed data => **Event Server**

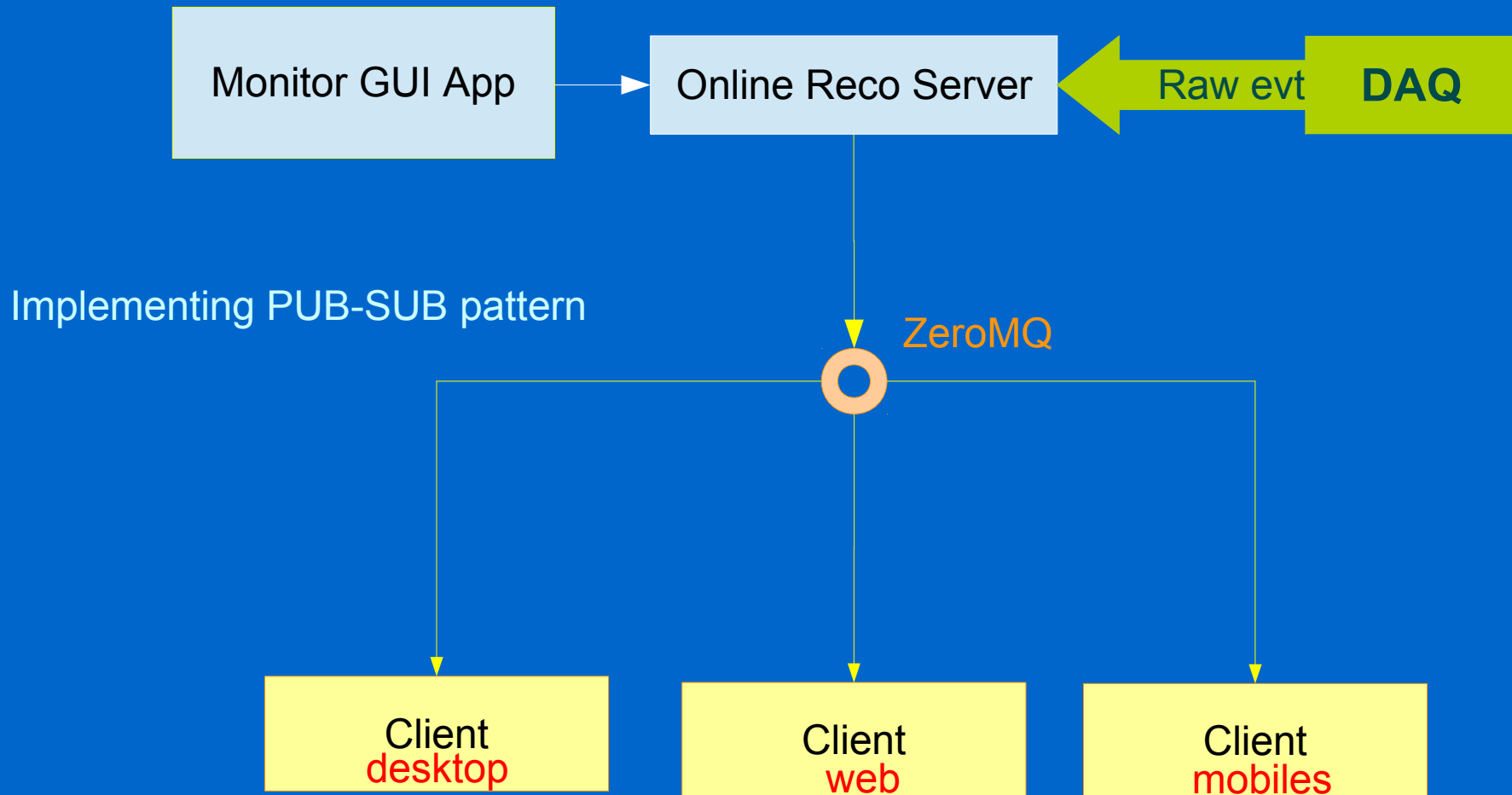
A new Online Model



A new Online Model



A new Online Model




AliRecoServer

Start Of Run

Monitor GUI App.

AliRecoServer



AliRecoServer

Start Of Run



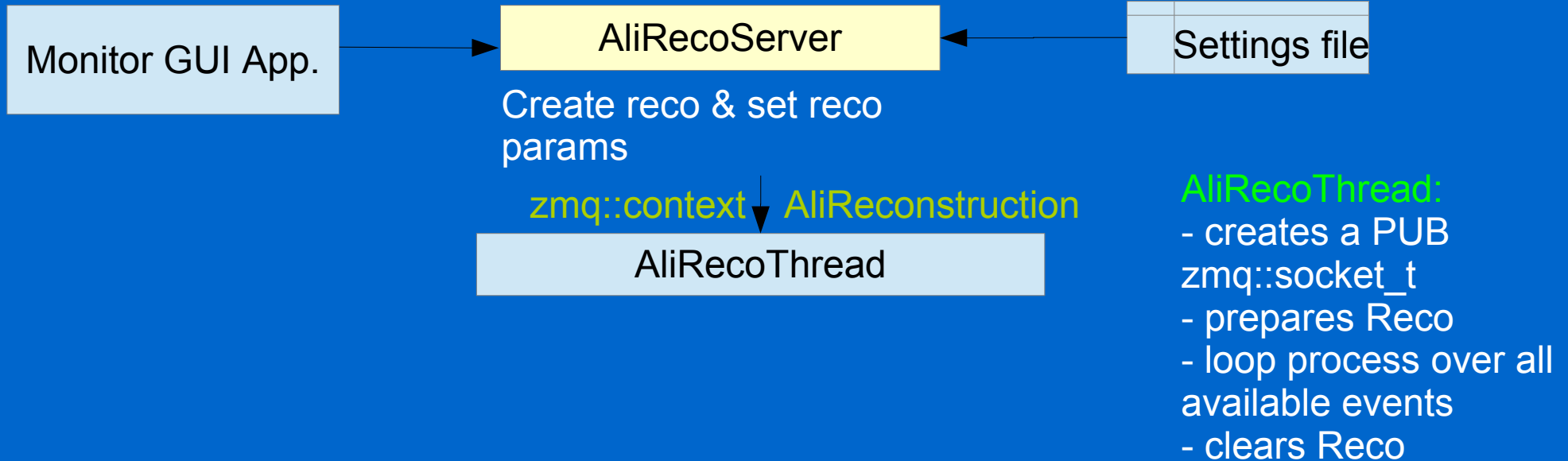
AliRecoServer

Start Of Run



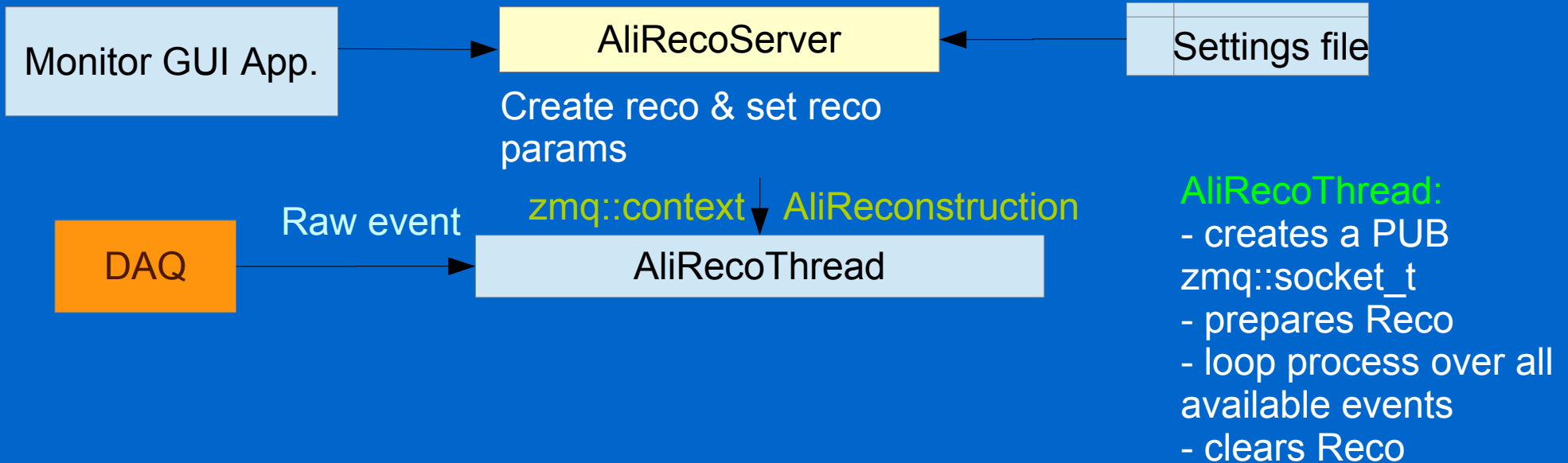
AliRecoServer

Start Of Run



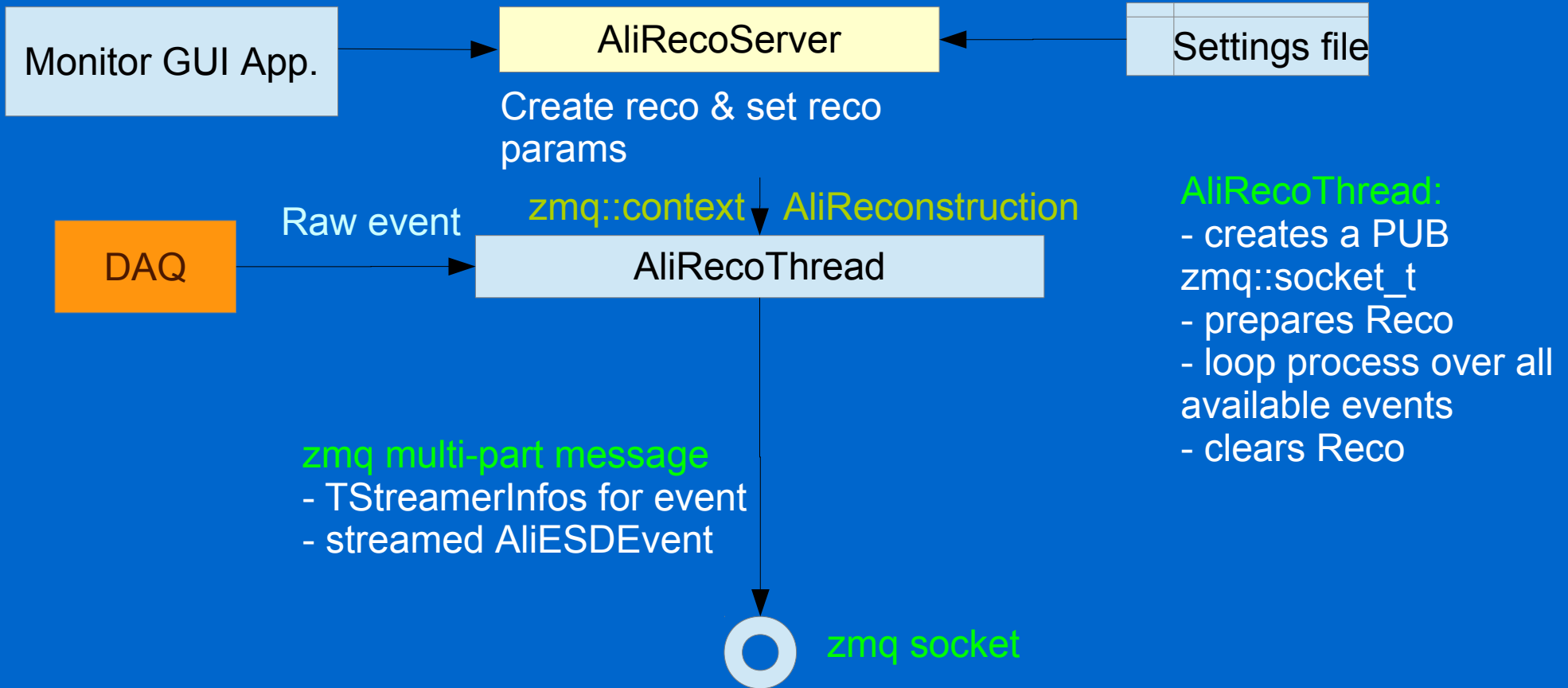
AliRecoServer

Start Of Run

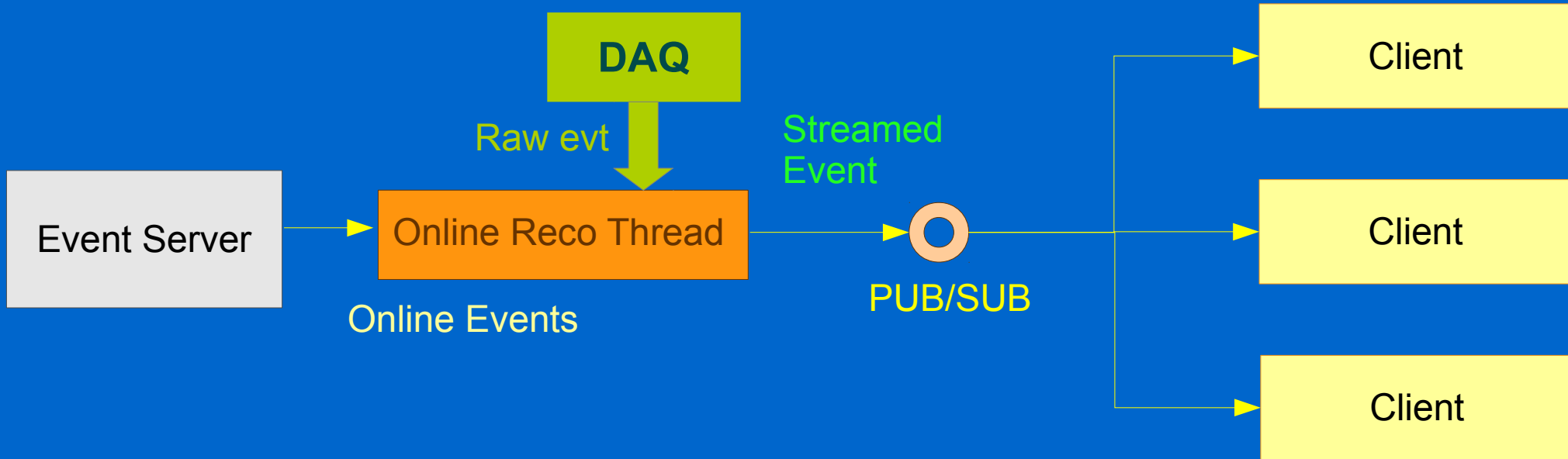


AliRecoServer

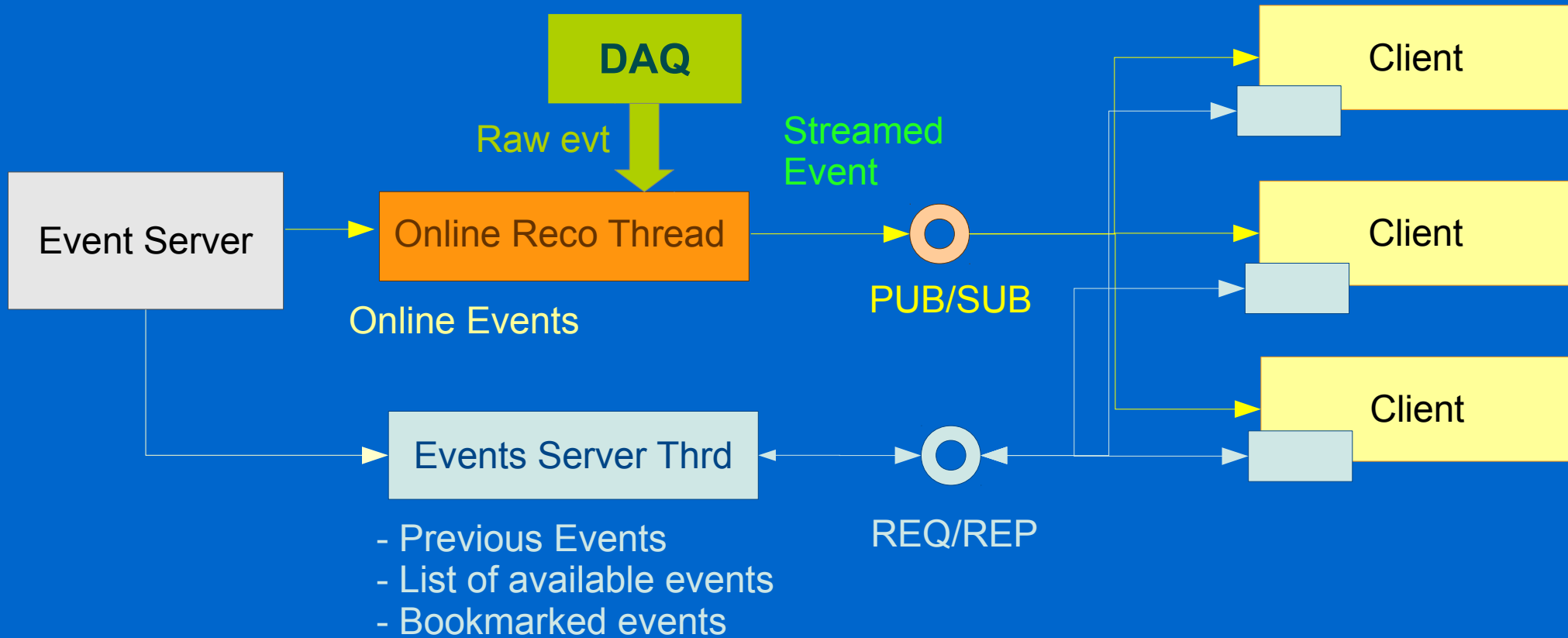
Start Of Run



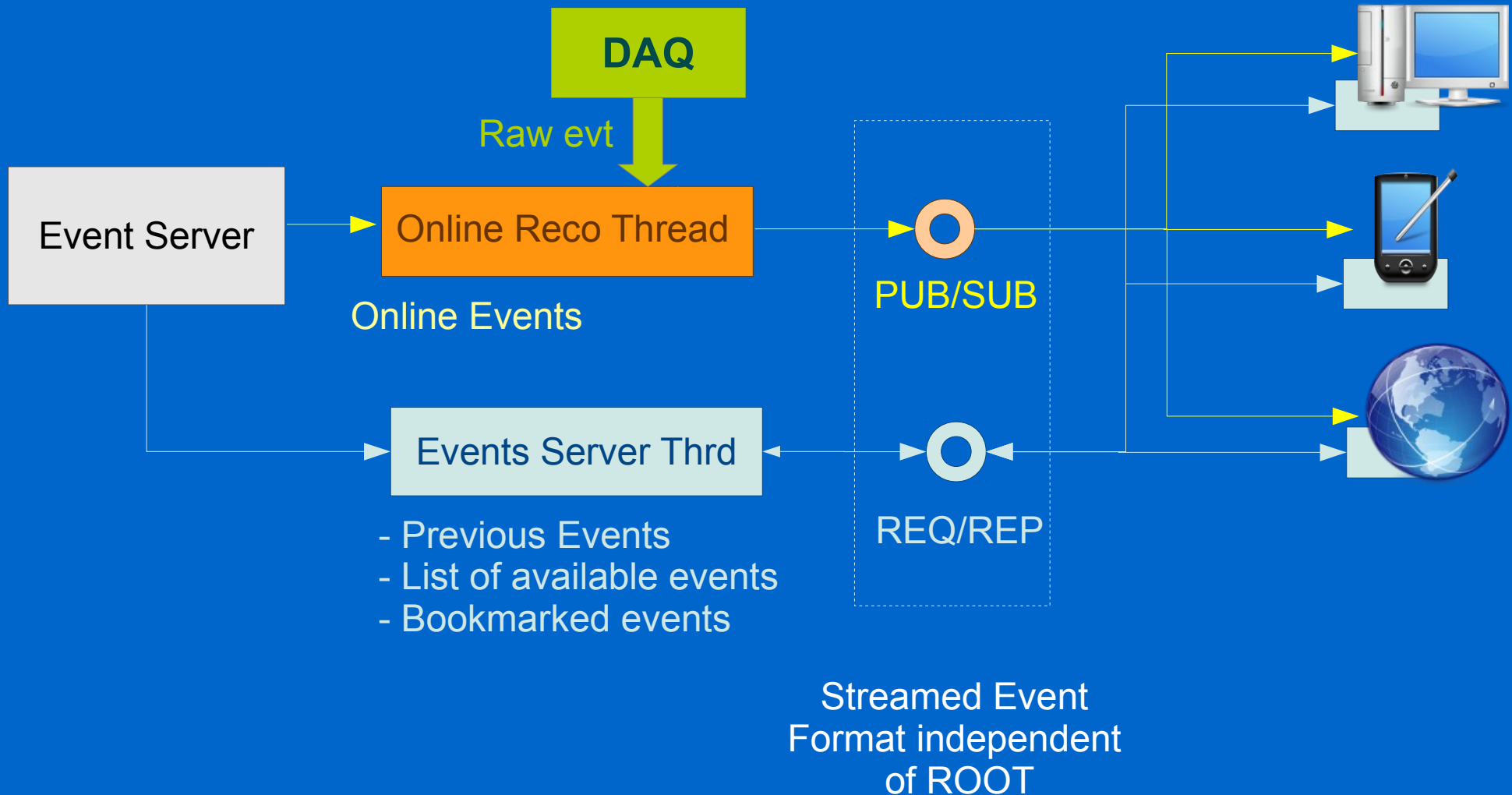
Future plans



Future plans



Future plans



Summary

- usage of **ROOT** (GUI, TThreads) makes the server cross-platform and its **streamer technology** makes **easy to stream** AliRoot objects (AliESDEvent,..)
- This model allows one to **view unhindered** the 3D event while the online reconstruction continues in parallel on the server
- The online reconstruction can run on **multiple processors** using **PROOF lite**
- Developed a **working event server**:
 - **Serve** online reconstructed **events**
- The server provides **events as a service**
- The client (visualization) is not fully modified to access the server
- This model does not change much our reco/viz code, but it **opens the door** for different **clients** on possible **different platforms/environments**

Thank you for your attention!