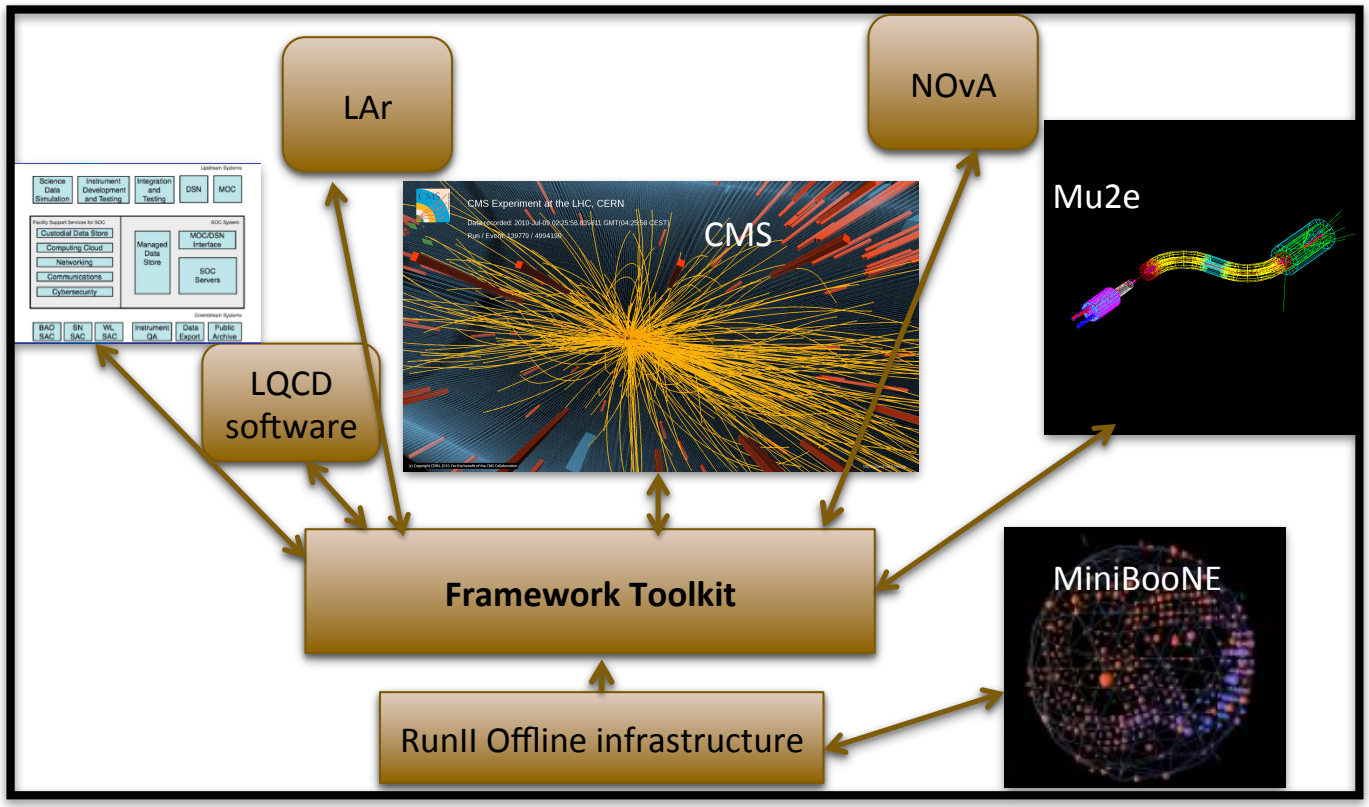
SSI workshop

P. Spentzouris

Event Processing Frameworks as a “Service”

- Provide software infrastructure (tools, development environment, …) that could be used to build event processing frameworks for “any” FNAL user
- Strategy is appealing since, *in principle*, it allows leveraging of resources, both internally (our developers) and externally (any specific capability developed for particular experiment is available for the rest of the users)
  - essentially the approach we take with IF experiments
    - BTW, a change of paradigm: we own the “framework”, experiments own its individual instantiations.
      - I believe this will be forced on HEP experiments anyway, as technologies advance and the computing paradigm changes
- It is also appealing from the point of view of increasing efficiency of our users
  - reduces the overhead of “learning” tools and development environment when switching experiments

So, in an “ideal” world (as per the previous statements)



OK, what is wrong in this picture?

Is it too late to re-sync?

- The real question, what would it take to re-sync the two major tool-sets that SSI develops?
  - Identify differences and (hopefully) similarities
  - Estimate effort for eliminating differences
    - to provide a minimum (useful) set of common tools
    - without interrupting “operations” or deployment of our tools for near-future operations

The charge

- One of the long-term strategic goals for your SSI is to work toward evolving the cms framework and art so they can share common infrastructure. I think that in order to jump start the process of clearly defining the deliverables of the project and better understanding the effort, planning, etc, it would be useful to hold an informal SSI workshop on this topic. In this workshop we should review the current status and plans for the two frameworks, identify design and implementation differences and similarities, identify functionality that could be pulled out to form a common layer, discuss timeline and evaluate the required effort.

- So, in random order, from a non-expert point of view, I would hope that we discuss framework component interactions, data model, persistency, configuration, geometry and conditions data management, interaction with detector simulation, build and release management, work-flow management, error-logger, monitoring,…
  - the outcome of this discussion should be a summary of your findings and a recommendation on how to proceed
    - then we will have to form a plan on how to act on the recommendation…