

# **Status & plans for the MonitorSvc**

**Eric van Herwijnen**  
**Tuesday 8 Sep 2009**

# Plan

---

- ◆ **Current MonitorSvc will be consolidated**
  - Stability issues are being addressed
  - No substantial rewrite for the moment
  - Gain more experience during first data taking
- ◆ **Possible new version targeted for may 2010**
  - Experience gathered so far suggests the design could be improved (new requirements were added and implemented ad hoc)
  - Discussions are taking place
- ◆ **Requirements document being worked on (see agenda)**
  - Also implementation details and suggestions for improvements
- ◆ **Some prototyping of new ideas is going on**
- ◆ **Things to fix immediately (next slide)**

# To fix immediately

---

- ◆ **MF: reset histograms after save, EFF: no reset after save**
  - Behaviour should to be consistent
  - Thus we should reset histograms after save in EFF
  - Resetting easy, but how to guarantee that histograms were saved? (published, added and saved before reset)
- ◆ **Histograms are serialized using the Boost binary archive**
  - Should be portable, but is not (can not deserialize on Windows)
  - Check out boost text-archive so we can run the presenter on Windows
- ◆ **Rates are calculated by the top level Adder using a “MonRate”**
  - It is a profile histogram
  - Avoid remaking the MonRate profile for each monitoring cycle
- ◆ **Behaviour at end of run:**
  - Check no events are lost in Savesets (jobs may terminate before their histograms are published, added or saved)
  - Check synchronization of Adders on fast run changes (events could be mixed between runs)

# To fix immediately, ctd

---

- ◆ **Need rates in MF for e.g. Luminosity monitoring**
  - Rates calculated by Adders (from MonRate)
  - In EFF, Adder only adds one task
  - In MF, need one Adder per task?
- ◆ **Slc5 -> check if current LCG gives the right version of Boost**

# Long term issues

---

- ◆ **One Dim service per published object**
  - Large overhead
  - More efficient: put all published objects inside one dim service
  - Bookkeeping for Adders simplified
  - But: incompatible with Presenters; intermediate solution could be that top level Adder decodes the grouped histograms and republishes them using individual services. Also unclear how it would work with large numbers of histograms. Beat is prototyping.
- ◆ **Boost makes code simpler but has disadvantages:**
  - Serialization module is unsupported
  - Only used for histograms and MonRates anyhow
- ◆ **Implement histograms as Dim rpc's**
  - Only get histograms when required

# Long term issues, ctd

---

- ◆ **Counters are added in PVSS**
  - Requires frequent reconfiguration of the EFF
  - High overhead
  - So, change Adders to add also counters
- ◆ **Adders & Savers have single code base**
  - Use the UTGID to determine where they are running (EFF, MF, top level, subfarm, node level Adders)
  - Redesign this to remove hard coded dependency on UTGID structure (number of “\_”s)