

HLT supervisor status and plans

■ What the HLTS should do

- acts as the Filter Unit (FU) Function Manager (FM)
 - drives FU state transitions commands originating from run control (RCMS)
- read and distribute configuration to FUs at start of run
 - trigger config from confDB
- collect trigger statistics from FUs and store in condDB
 - Prescaler factors and trigger counters per Lumi Section (LS)
- calculate and monitor prescalers settings.
 - Probably performed by the Trigger Manager FM
- distribute prescaler updates to FUs
- etc.

■ DESY/CERN coordination

- CERN = Emilio Meschi our contact
- DESY = Ulf Behrens, Derek Hatton, Andreas Meyer (now DQM) and self

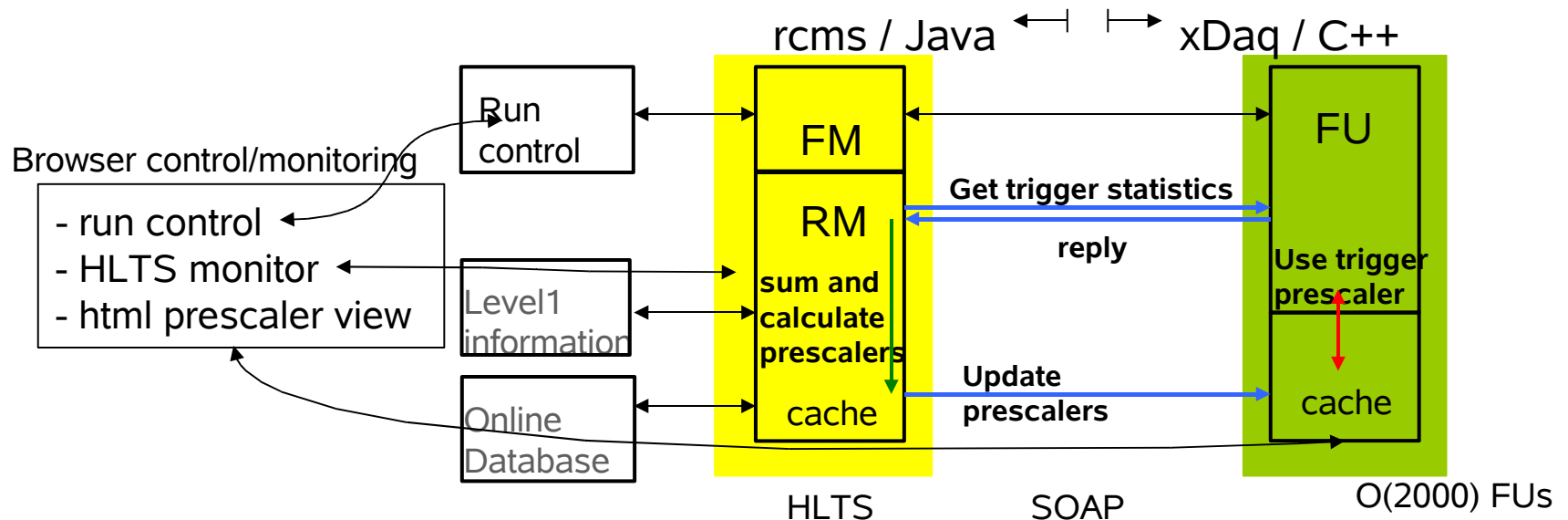
HLTS test system

- Test system
 - 5 node dual CPU CERN SL3 cluster
 - CMS software releases
 - Run control rcms 2-2-0
 - CMSSW-1-2-0-pre5 framework
 - xDaq 3.5 software
 - System consists of
 - run control
 - job control
 - HLTS function manager
 - FUEventProcessor (self generated data/trigger flow.)



Allows local testing before moving to CERN

HLTS prescaler handling



- SOAP messages transport information
- FM - Function Manager implements run control finite state machine
- RM - Rate Monitor; polls trigger statistics, calculates prescalers, updates its cache, and pushes updates to FUs
- Filter software accesses prescaler cache through service class (PrescaleService).
- Prescalers fixed for period of luminosity sections, typically 5-10 minutes.
- Prescalers used will be written to condDB
- Information will be exchanged between L1 and HLTS to coordinate changes

Prescaler handling tested at DESY & CERN – currently being reorganized see later

HLT prescaler status: PrescalerService class

```
...
#include "boost/thread/mutex.hpp"
...

class PrescalerService
{
private:
    Cache store;

public:
    PrescalerService(const ParameterSet&, ActivityRegistry&);
    ~PrescalerService();

    ...

    int getPrescale(unsigned int ls, string path, string module);
    int getPrescale(unsigned int ls, string module);

    int putPrescale(string s);

    int sizePrescale();
    void listPrescale(xgi::Input *in, xgi::Output *out);
};
```

■ methods

- putPrescale
 - inserts prescaler information into cache.
- getPrescale
 - Retrieves prescaler for given LS#, path and module name.

■ data

- store contains last N LS# prescalers
- Access to data synchronized using scope mutex lock

Class to be used by Event Processor framework to get HLT prescalers.

HLT prescaler status: monitoring

modulePs.html - Konqueror

Location: http://zcms3:1972/urn:xdaq-application:lid=51/modulePs

EventProcessor modulePs Lumi sections cached: 5

section 14 touched 5

path	module	prescale
zmumu	prescale1	1
zelel	prescale2	2
jetjet	prescale3	0

section 15 touched 4

path	module	prescale
zmumu	prescale1	1
zelel	prescale2	2
jetjet	prescale3	0

...
...
section 18 touched 1

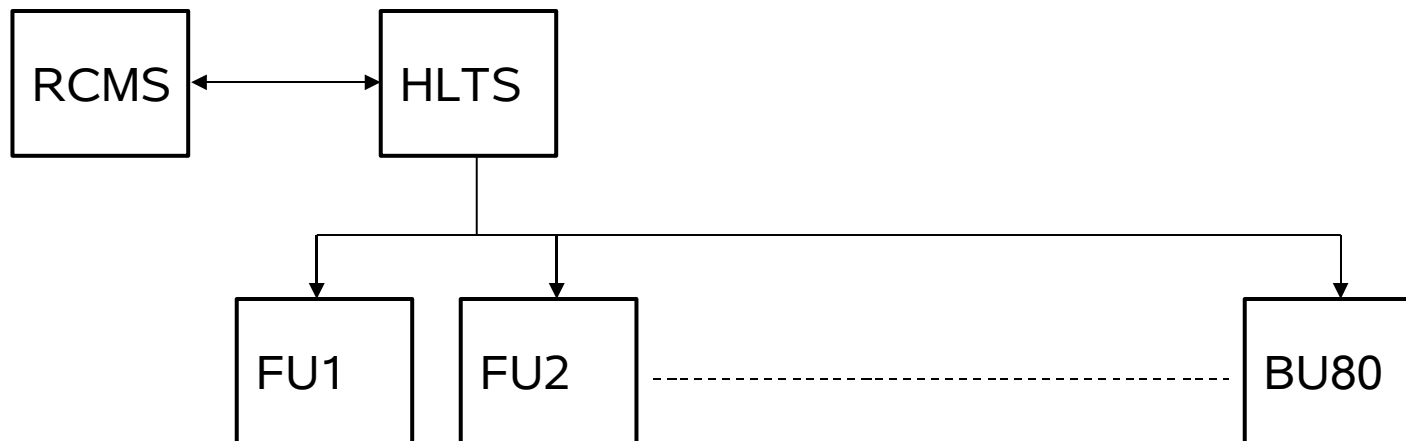
- FUEventProcessor instance specific xDAQ html page

- HLTS Tomcat web services:
 - Javascript using Ajax / DWR / Web2.0 monitoring tool
 - Tool makes prototype activity visible
 - Underlying webtool useful for monitor displays



HLTS status

- **Scaling test at CERN performed successfully**
 - Test performed on cmsdaq test cluster during 1st week Jan. '07
 - 80 FUs was the easiest configurable (1/FU per node) setup
 - test ran successfully for 3 days gathering and distributing prescaler information.
 - Prescaler getTrigger report and caching worked correctly.



HLT prescaler status

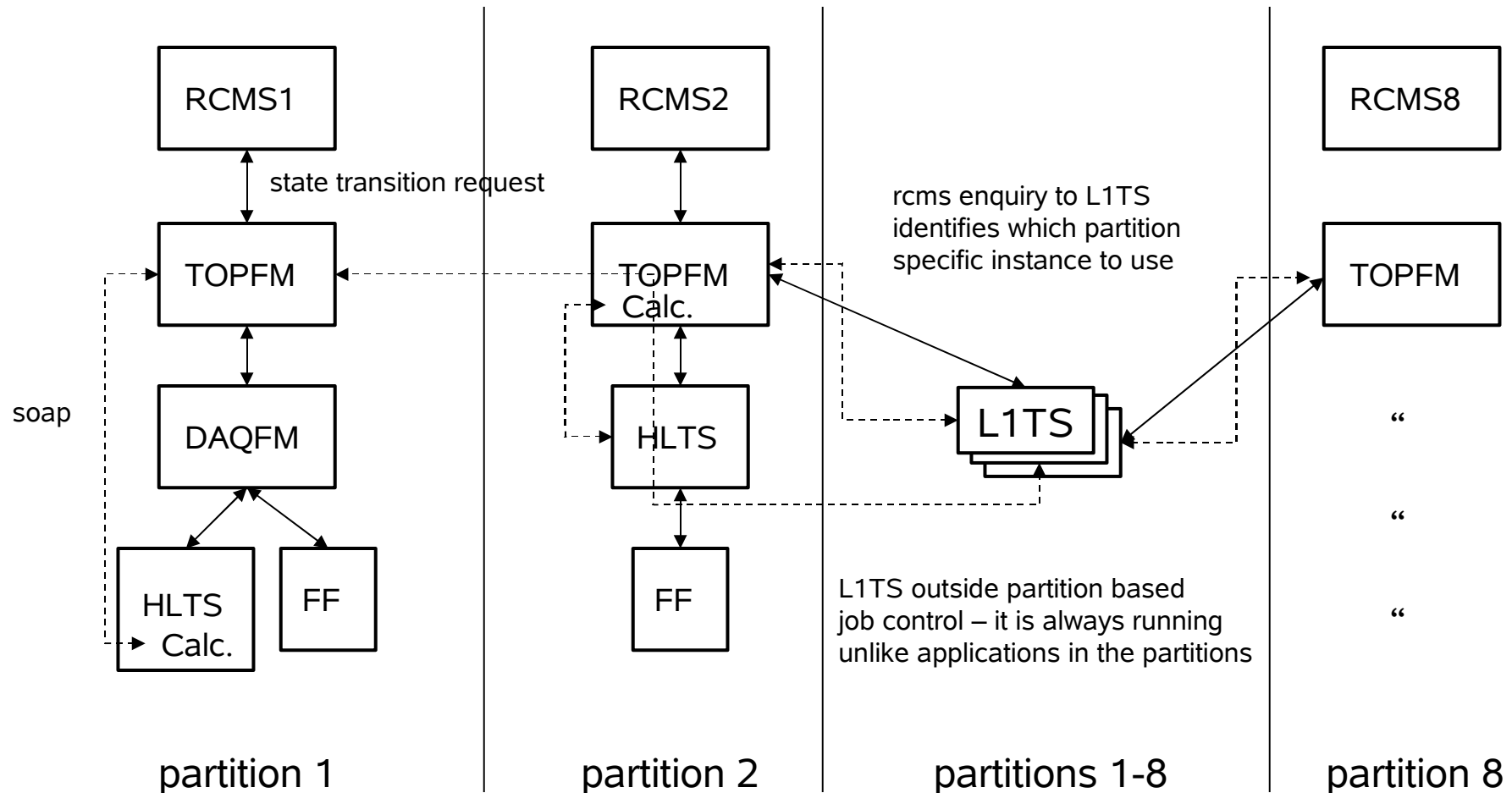
■ Has been implemented

- ❑ SOAP messages including en- and de-coding
- ❑ Filter Unit PrescalerService class
- ❑ Rate Monitor caching of prescalers (unity calculation of new values)
- ❑ Simple viewing html page for FU cache debugging
- ❑ HLTS tomcat web services monitor using Ajax /DWR / Web2.0

■ Test system functionally complete, but

- ❑ Major reorganization is planned to insert a Trigger Manager FM (TMFM) above HLTS and L1TS.
- ❑ The TMFM initially synchronizes L1 and HLT prescaler changes.
- ❑ The design is under discussion – consequences for the HLTS see next slide.

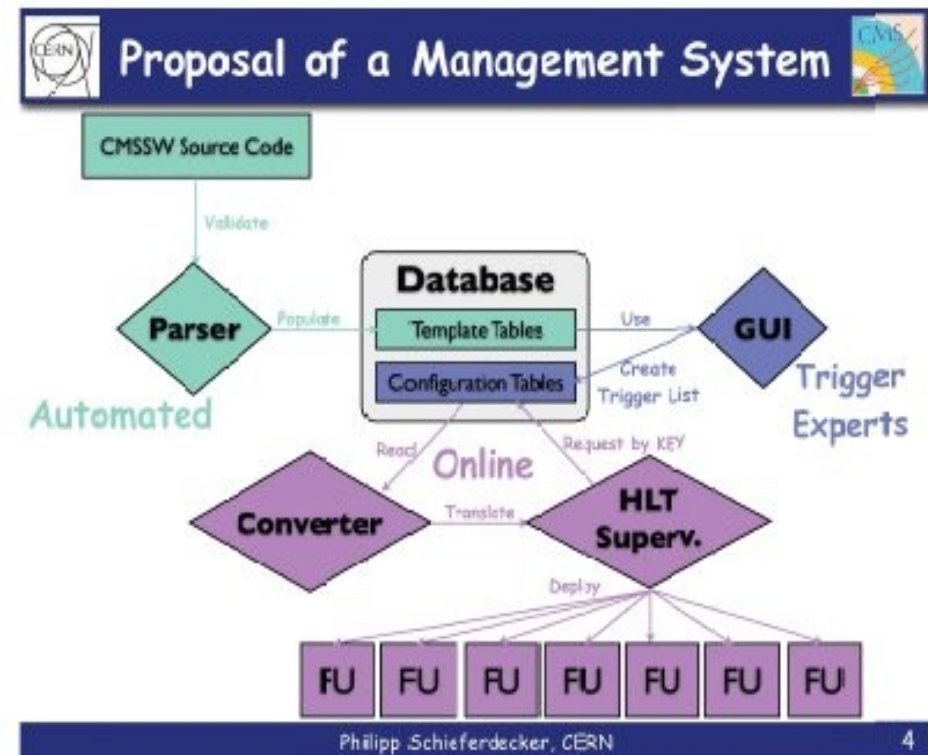
Possible trigger manager (TMFM) partition connectivity



HLT configuration database work

■ Aim

- Store HLT configuration in confDB from where the HLTS can access it. RCMS specifies the configuration via a key.
- Currently RCMS provides a file path and the FUs must parse the contents.
- Work split:
 - Jonathan Hollar
 - CMSSW source parser to get default parameter settings
 - Phillip Schieferdecker
 - GUI to create trigger config
 - DB schema
 - Ulf Behrens
 - DB to configuration string converter



HLT configuration database work

■ Converter implementation issues

- Use Philipp Schieferdecker's modified GUI library
 - Eases implementation, and
 - Ensures no inconsistencies between implementations
- GUI class has been imported into eclipse
 - Integration into tomcat in progress = eases testing

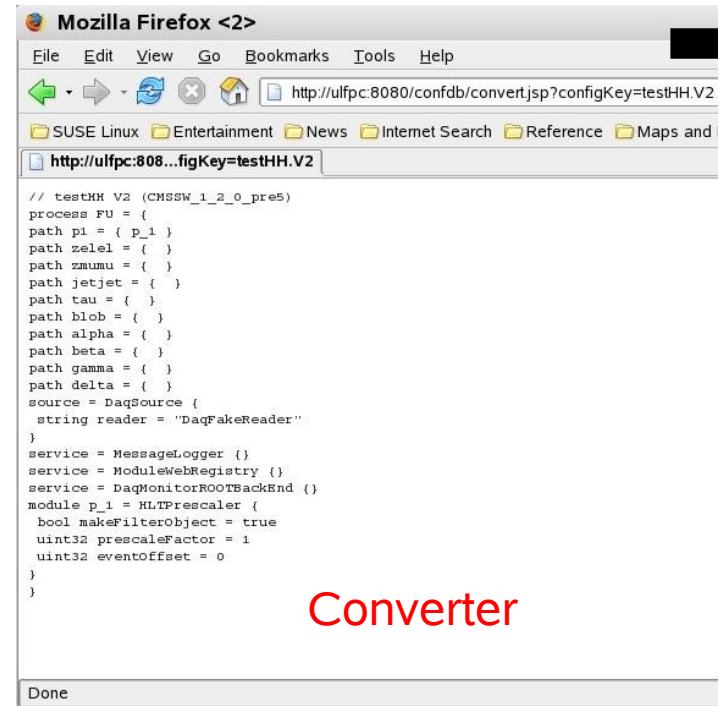
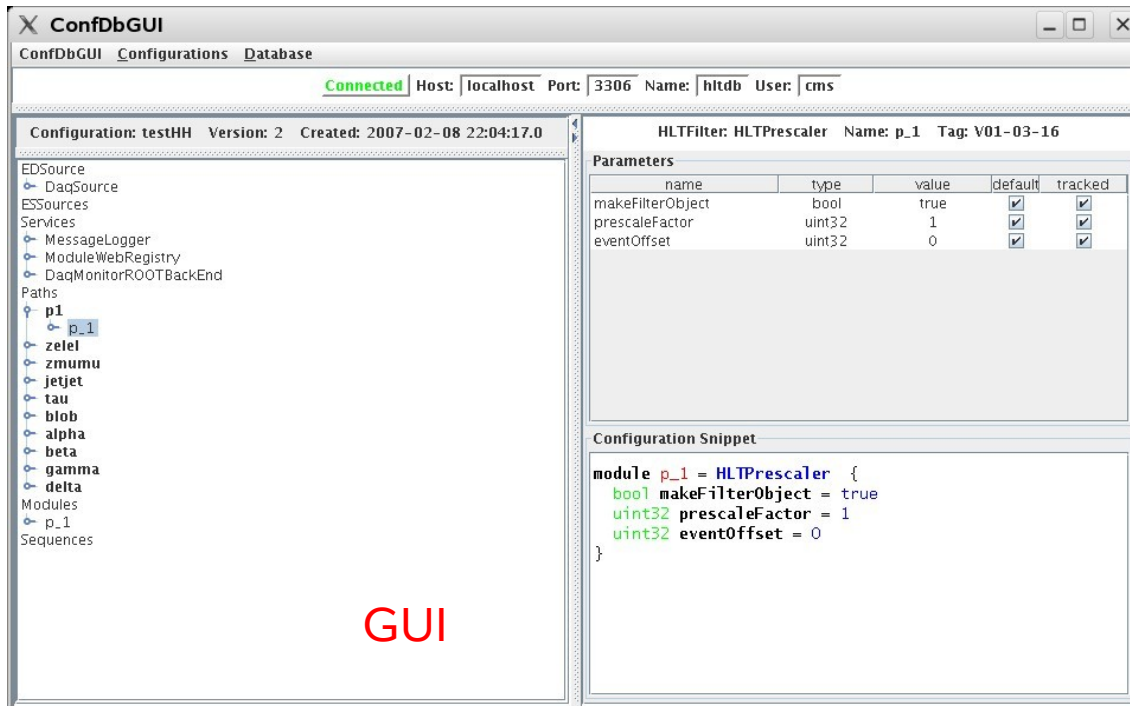
■ Work remaining

- Need to define what node information goes where in the string
 - Currently expect the FUs to use the currently defined grammar
- Key from RCMS
- Interface to HLTS
- Distribute configuration string to FUs

HLT configuration database work

■ Status

- ❑ Parser, schema and GUI essentially complete.
- ❑ 1st version of converter released last Friday.
- ❑ GUI and Converter output for DESY test config:



HLTS data to condDB

- HLTS data which needs writing to condDB to calculate lumi
 - Prescale factors per LS
 - Trigger counters per LS
- Interface to online DB
 - HLTS (or TMFM) is written in Java = Tstore not recommended
 - RCMS use JDBC
 - HLTS will use a JDBC based solution, must be validated with DB group
 - Work has started

HLTS plans

- During next CERN visit (19-23 Feb)
 - Increase the number of FUs to ~300 for further tests of caching
 - Clarify TMFM implementation
 - Continue work on inserting prescaler definitions per LS into conditions DB
 - Insert DESY code as soon as possible into next release
- Ongoing
 - Insert configuration converter output into test setup
 - Probably needs upgrade of cluster (SLC4, etc.)

Cannot afford to lose manpower – must concentrate on CMS