



Computing in High Energy
and Nuclear Physics

13-17 February 2006,
T.I.F.R. Mumbai, India



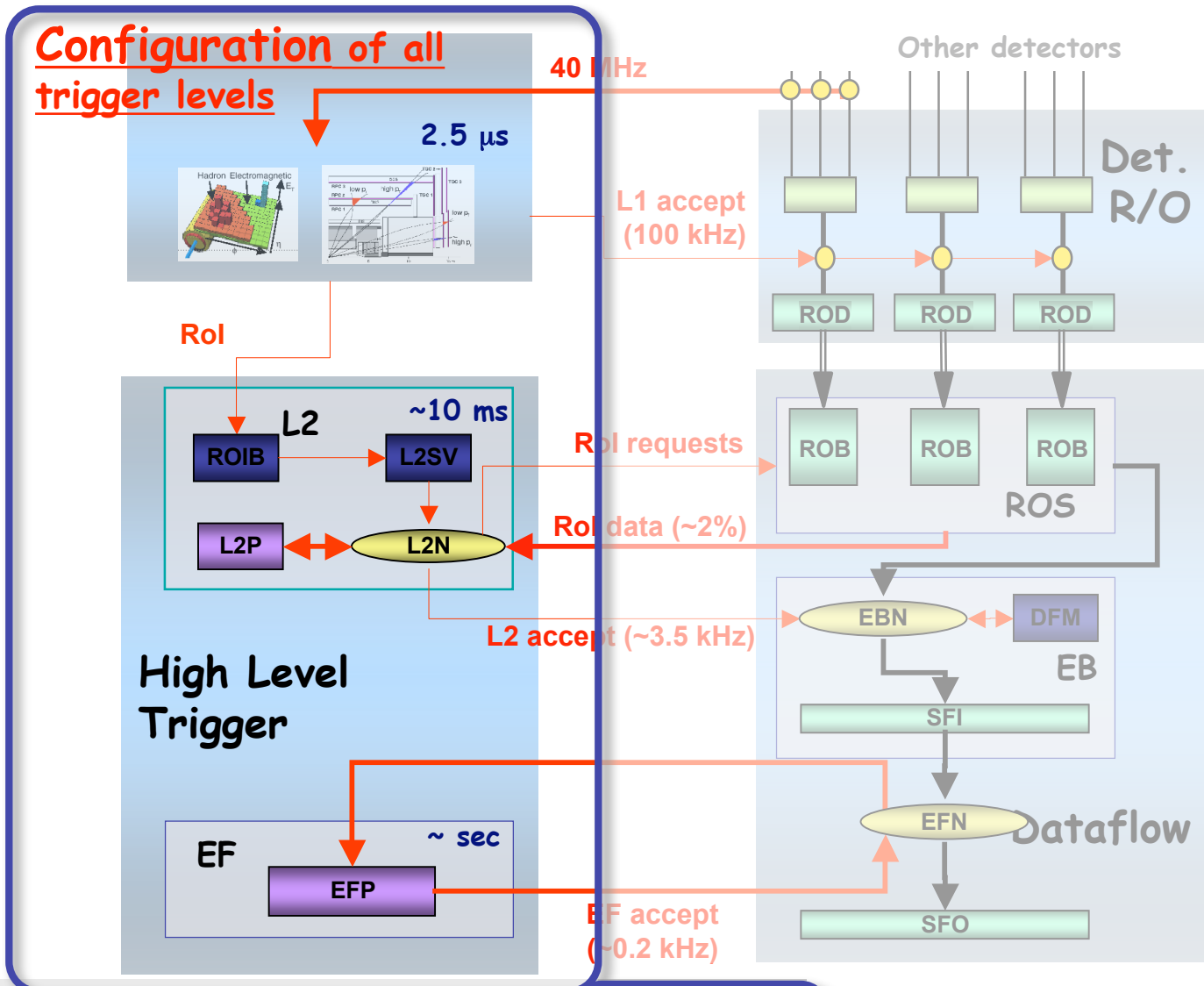
A configuration system for the ATLAS trigger

A. dos Anjos, N. Ellis, J. Haller, A. Höcker,
T. Kohno, M. Landon, H. von der Schmitt,
R. Spiwoks, T. Wengler, W. Wiedenmann, H. Zobernig

Outline

- **ATLAS Trigger**
 - System overview
 - LVL1 selection and configuration
 - HLT selection and configuration
- **ATLAS Trigger Configuration System**
 - Requirements
 - System overview
 - Trigger Data Base
 - Tools for DB population
 - Tools for data retrieval
- **Results of first Tests**
- **Status and Outlook**

TDAQ context



Infrastructure

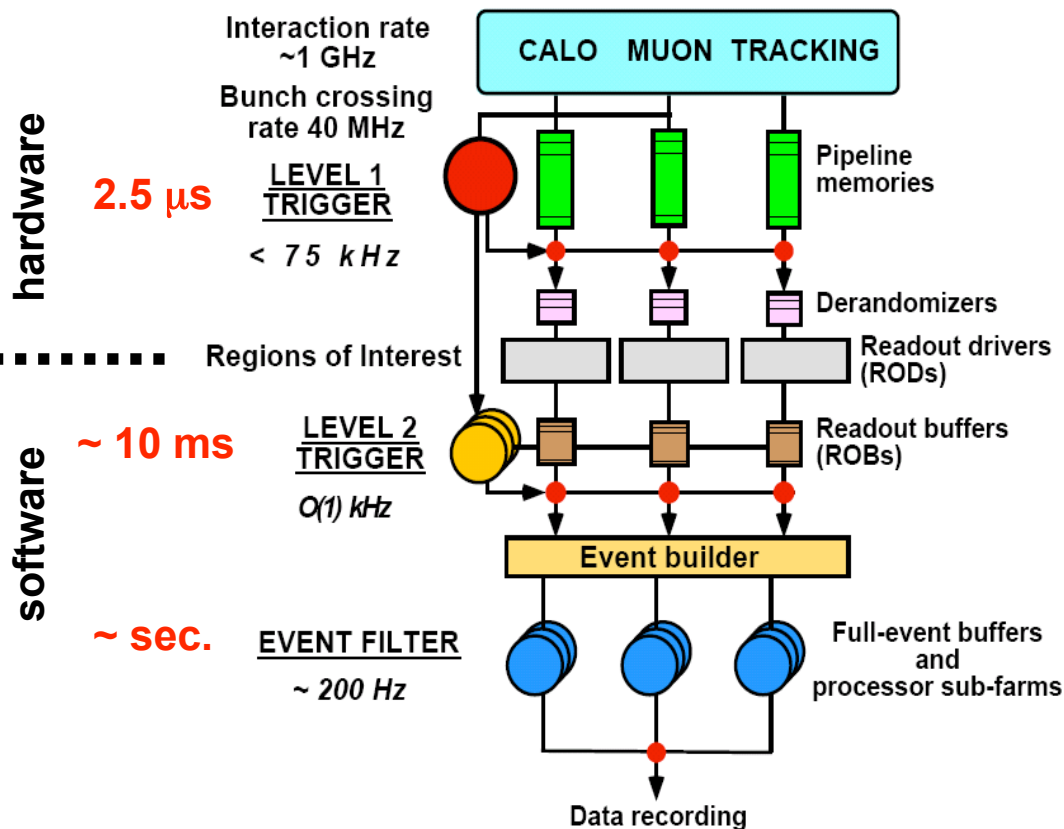
Control

Communication

Databases

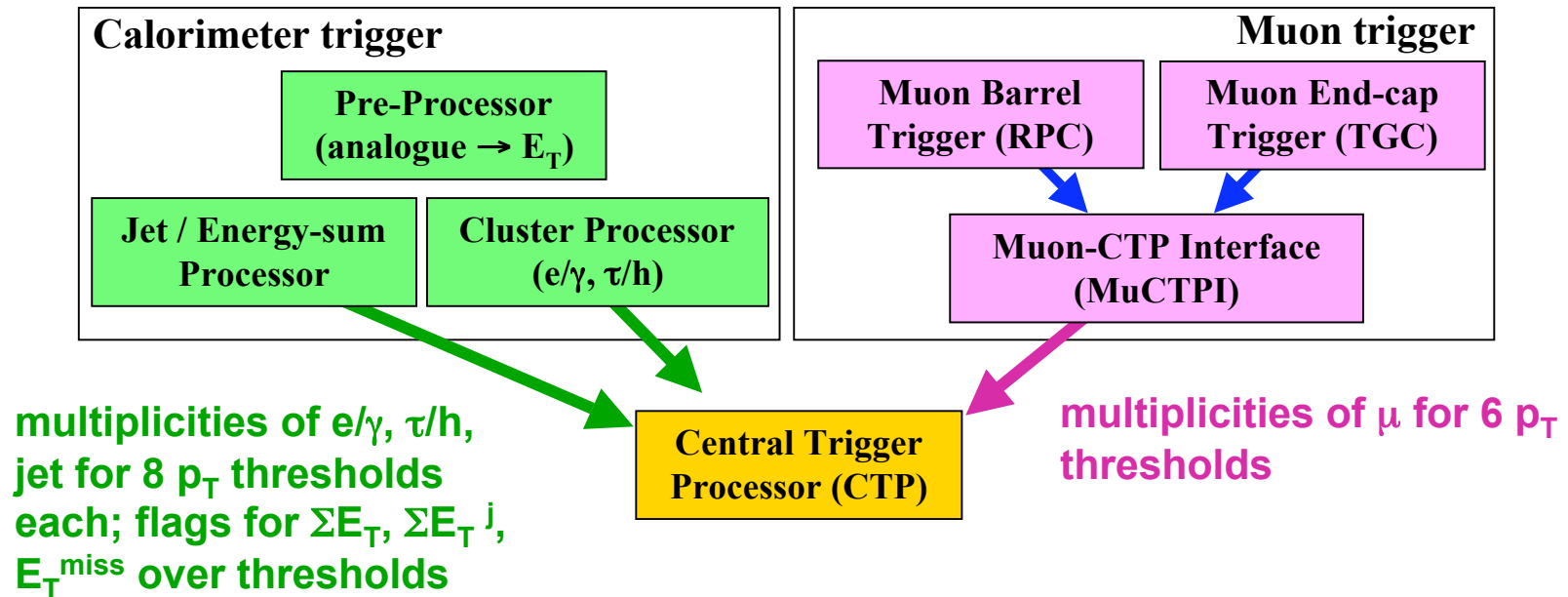
The ATLAS Trigger System

3-Level Trigger System:



- 1) **LVL1** decision based on data from **calorimeters** and **muon trigger chambers**; synchronous at 40 MHz; **bunch crossing identification**
- 2) **LVL2** uses **Regions of Interest** (identified by LVL1 data (ca. 2%) with full granularity from all detectors
- 3) **Event Filter** has access to full event and can perform more refined event reconstruction

LVL1 selection



CTP: trigger decision based on inputs, deadtime, random triggers, bunch groups

LVL1 trigger configuration

LVL1 trigger menu information for **all** LVL1 subsystems: muon trigger chambers, central muon trigger, calorimeter trigger, Central Trigger Processor (CTP)

A LVL1 trigger menu has a hierarchical structure:

- A **Trigger Menu** (e.g. “lumi_01”) is composed of many **Items** (e.g. “2J25+XE45”)
- A **Trigger Item** is composed of some **Thresholds** (e.g. “J25”)
- A **Trigger Threshold** is composed of many **Threshold Values** (e.g. “ $E_T=25$, cone=4, $\eta_{\min}=1.2$, $\eta_{\max}=1.4$, $\varphi_{\min}=0.$, $\varphi_{\max}=0.2$ ”)

plus extra information like: dead-time parameters, bunch-group definition, random trigger rates, prescaled clocks, prescales, trigger type definition, jet input thresholds,...

example trigger menu:

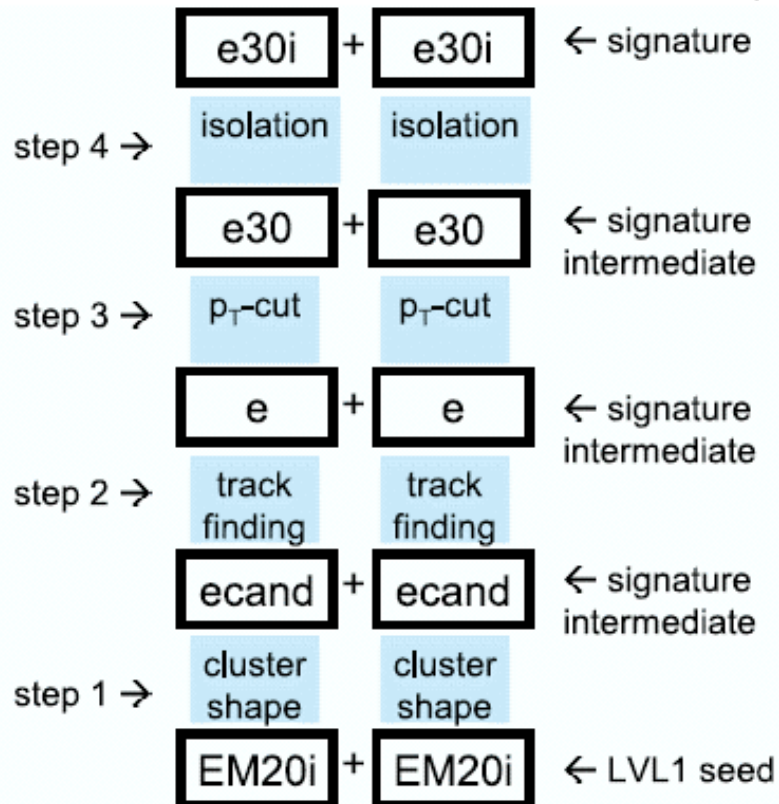
LVL1 Menu	$2 \times 10^{33} \text{cm}^{-2} \text{s}^{-1}$
MU20	0.8
2MU6	0.2
EM25i	12.0
2EM15i	4.0
J200	0.2
3J90	0.2
4J65	0.2
J60+xE60	0.4
TAU25+xE30	2.0
MU10+EM15i	0.1
Others	5.0
Total rate (kHz)	~ 25

Configuration must fulfill hardware limitations, e.g. available thresholds, number of allowed inputs, LUT configuration in CTP, etc.

HLT selection and configuration

HLT strategy: refinement of TriggerElements (seeded from LVL1) in stepwise processing, perform stepwise decisions

example of step-wise processing:



HLT uses the offline reconstruction SW framework ATHENA on ~3000 CPUs

parts to be configured:

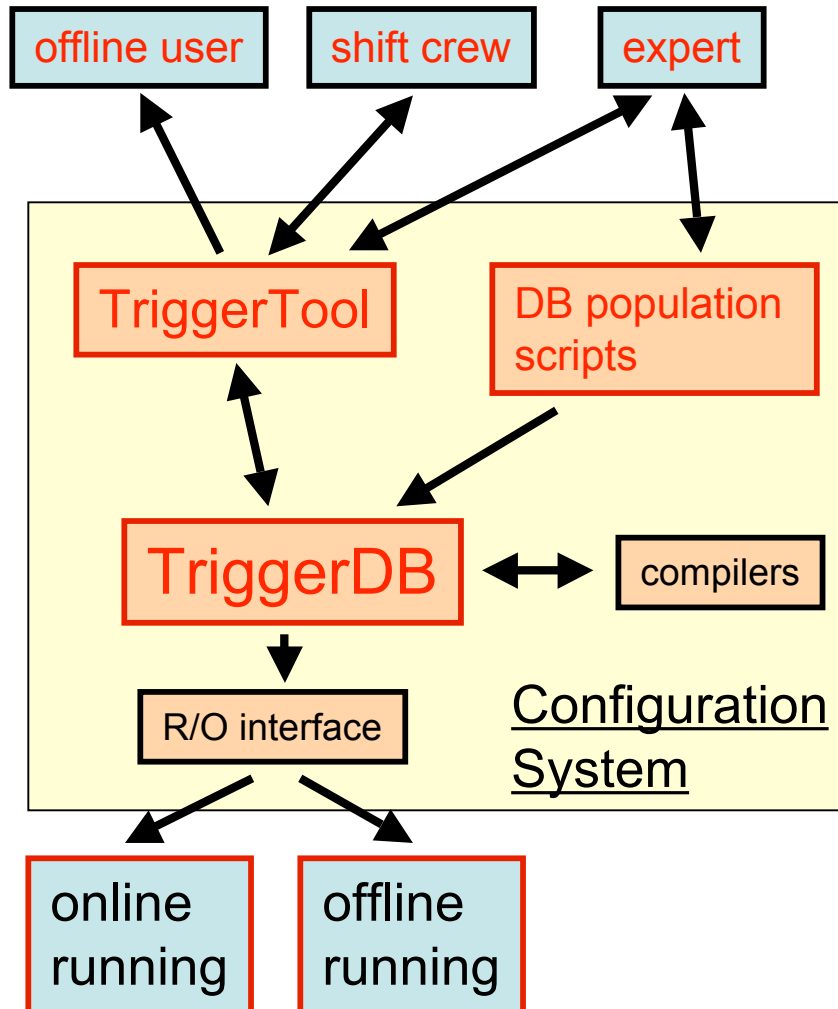
- 1) **HLT Menu:** determines which algorithms are called at which step and which signatures need to be fulfilled for accepted events
 - 2) all **configuration parameters of the algorithms and services**, called JobOptions (JO), compatibility with offline important
 - 3) **release information**
- ← **Consistency (with LVL1) important**

Requirements on the configuration system

- must store **all information** to configure the trigger
- selection will change with time (some very often)
 - **many versions** of configurations
- must store all versions used and protect them
- must allow **consistency checks** (guidance) when preparing new configurations
- must allow **easy trigger operation** for experts, shift crews, offline users
- will be used for **online** trigger **and offline** trigger simulation
- parts must be **compatible with offline SW** (Athena)
- must be **available at the detector, on the GRID, ...**

Trigger Configuration System (schematic)

Data Flow:



TriggerTool:

- GUI for DB population
- easy and consistent menu changes for experts (LVL1 and HLT)

TriggerDB:

- stores all information to configure the trigger: LVL1 menu, HLT menu, HLT algorithm parameters (JO), HLT release information
- stores all versions used, with a key
 - Configuration and Conditions DB
 - DB available at Point and replicated to external sites

Retrieval of information for running:

get information by key via two paths:

- extraction of data in XML/JO files
- direct read-out

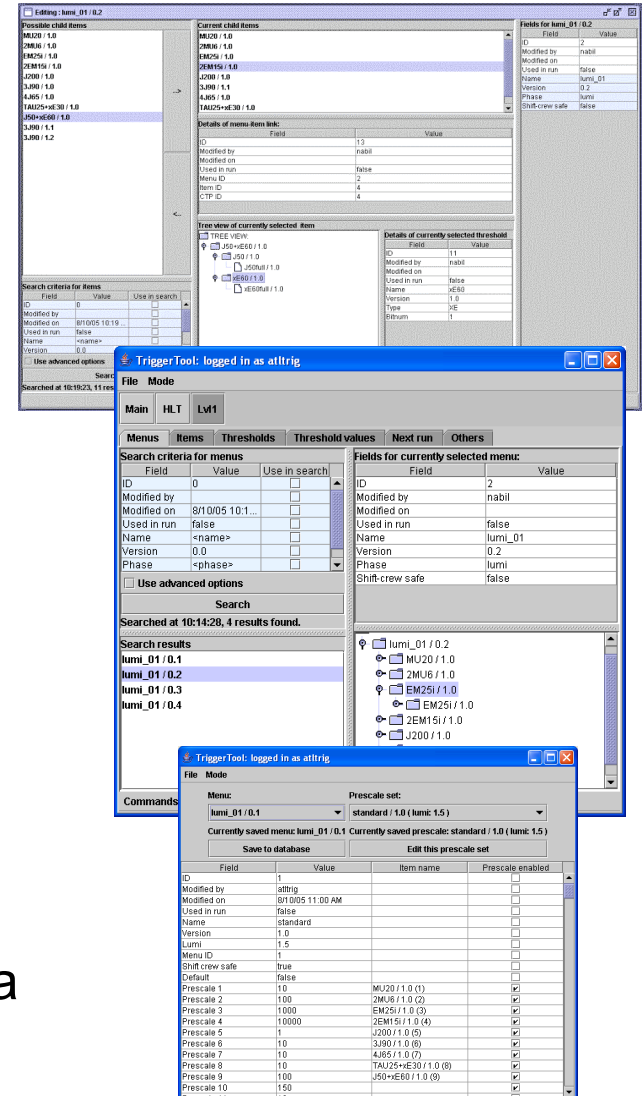
for both online + offline running

DB Population: TriggerTool

- TriggerTool is an interface to TriggerDB.
- central tool for:
 - experts to prepare and change (LVL1 + HLT) menus consistently (queries, automatic checks, ...)
 - restricted menu changes online (by shift crew)
 - offline users to browse the DB (the menus) and export information to files for standalone tests
- covers online and offline use cases

Implementation :

- JAVA based stand-alone GUI
 - can be used on all platforms
- implemented for ORACLE and MySQL
- can access the TriggerDB from everywhere via internet; password protected



CTP Menu Compiler

- Need to translate
 - from high-level definition of the LVL1-menu in the **database**
 - to settings e.g. for LUT, CAM, in the **electronics**

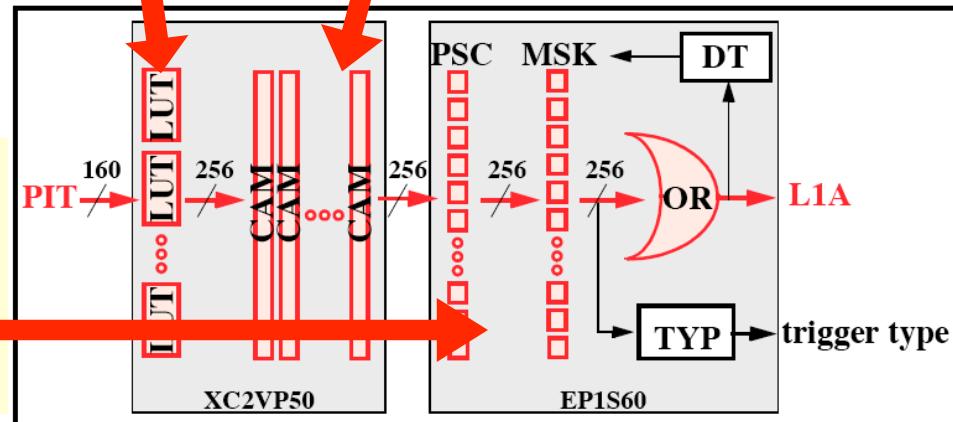
```
...
<TriggerItem TI_ID="i1EM10" mask="on" priority="low" prescale="1">
  <TriggerCondition triggerthreshold="EM10" mult="1" />
</TriggerItem>
...
<TriggerThreshold name="EM10" type="EM" bitnum="3">
  ...
  <TriggerThresholdValue thresholdval= "10"
    emisolation= "1"
    haisolation1= "1"
    haisolation2= "2"
    phimin= "0"
    phimax= "0"
    etamin= "-2.5"
    etamax= "0.0" />
  <Cable name="Cable02">
    <Signal range_begin="15" range_end="17"/>
  </Cable>
</TriggerThreshold>
...
```

LUT File CAM File

```
...
0x00000000
0x00000000
0x00000000
0x00000000
0x00000001
0x00000001
0x00000001
0x00000001
...

...
0xaaafffff
0xffffffff
0x00000000
0x00000000
0xffffffff
0xffffffff
0x00000000
0x00000000
...
```

Files stored as CLOBs in the DB



(e.g. XML extracted from the TrigConf DB)

Registers can be set directly from DB values

Population of the TriggerDB

HLT JobOptions and release part:

- important: get all JobOptions of all algorithms, services etc. for a certain release (offline uses Python files, difficult to map to DB)
- Tool to scan the release and populate the TriggerDB automatically with default values
- foreseen to be done once per release with checking of changes
- after population, interactive changes of algorithm parameters
- overlap with configuration of offline reconstruction jobs

Analysis for 10.0.1

Summary

- 1000 Packages
 - 99 Packages with inbound properties
 - 80 Packages with warnings
 - 1000 Packages with comments
- 4284 Properties declared
 - 4043 Strongly typed properties
 - 164 Weakly typed properties
- 81 Typeton properties
 - 291 Strongly typed
 - 17 Weakly typed
- 4207 Properties with...
 - 4043 Strongly typed
 - 17 Weakly typed
- 85 Services
 - 888 Algorithms
 - 428 Tools
 - 211 Comments

Schema

- + AGDD
- + AGDD_DTD
- + AGDD_Model
- + AGDD_Parser
- + AGDD_Svc
- + AGDD_XML
- + AMI

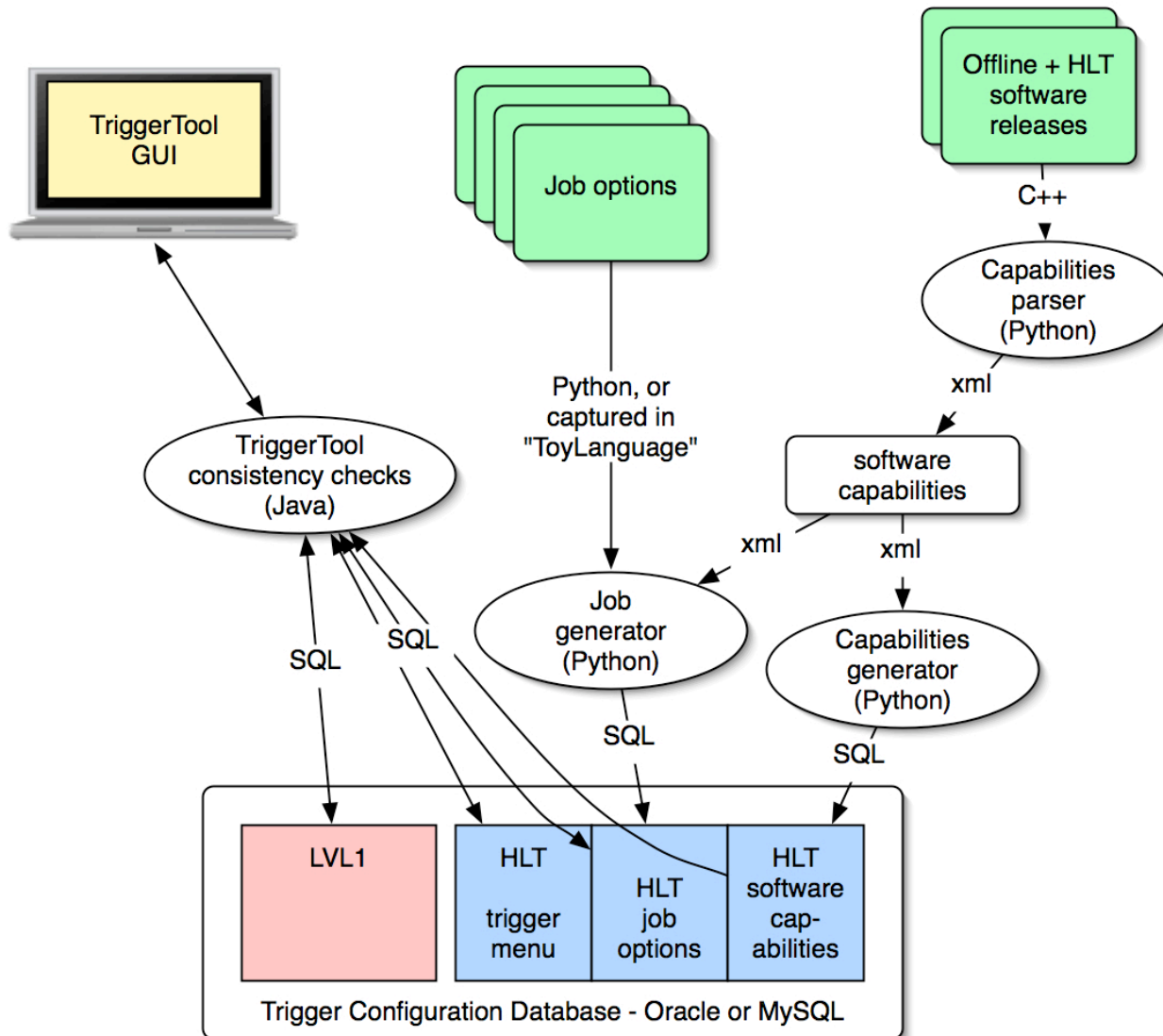
Name	Type	Parent C++ class	Default	C++ binding	Declared in	Type strength
MaxIterate	int	TileRawChannelBuilderOpFilter	0	in_maxIterate	src/TileRawChannelBuilderOpFilter.cxx	strong
RMSChannelNoise	int	TileRawChannelBuilderOpFilter	2	in_RMSChannelNoise	src/TileRawChannelBuilderOpFilter.cxx	strong
ActiveDgDiffThreshold	bool	TileRawChannelBuilderOpFilter	true	in_activeDgDiffThreshold	src/TileRawChannelBuilderOpFilter.cxx	strong
CoordTB	bool	TileRawChannelBuilderOpFilter	false	in_CoordTB	src/TileRawChannelBuilderOpFilter.cxx	strong
DgDiffThresholdLevel	double	TileRawChannelBuilderOpFilter	10	in_dgDiffThresholdLevel	src/TileRawChannelBuilderOpFilter.cxx	strong
MaxIterations	int	TileRawChannelBuilderOpFilter	5	in_maxIterations	src/TileRawChannelBuilderOpFilter.cxx	strong
MaxSampleRange	int	TileRawChannelBuilderOpFilter	1	in_maxSampleRange	src/TileRawChannelBuilderOpFilter.cxx	strong
PresatMode	int	TileRawChannelBuilderOpFilter	0	in_presatMode	src/TileRawChannelBuilderOpFilter.cxx	strong
PrecalculatedWeights	bool	TileRawChannelBuilderOpFilter	true	in_preWeights	src/TileRawChannelBuilderOpFilter.cxx	strong
TimeForConvergence	double	TileRawChannelBuilderOpFilter	1	in_timeForConvergence	src/TileRawChannelBuilderOpFilter.cxx	strong

LVL1 part and HLT menus part:

- can (must) be filled interactively
- use of GUI to ease the operation:

TriggerTool

Population of the DB - synopsis



Data Retrieval from DB for running

reminder: two data paths for both HLT and LVL1, online and offline

I. extraction of **intermediate files** (XML, flat Python) from TriggerDB

- allows standalone tests for e.g. development of new menus
- tests of online trigger without interference with the DB
- using formats compatible with current simulation input files
- different formats for LVL1 (XML), HLT menu (XML), and HLT JO (Python).

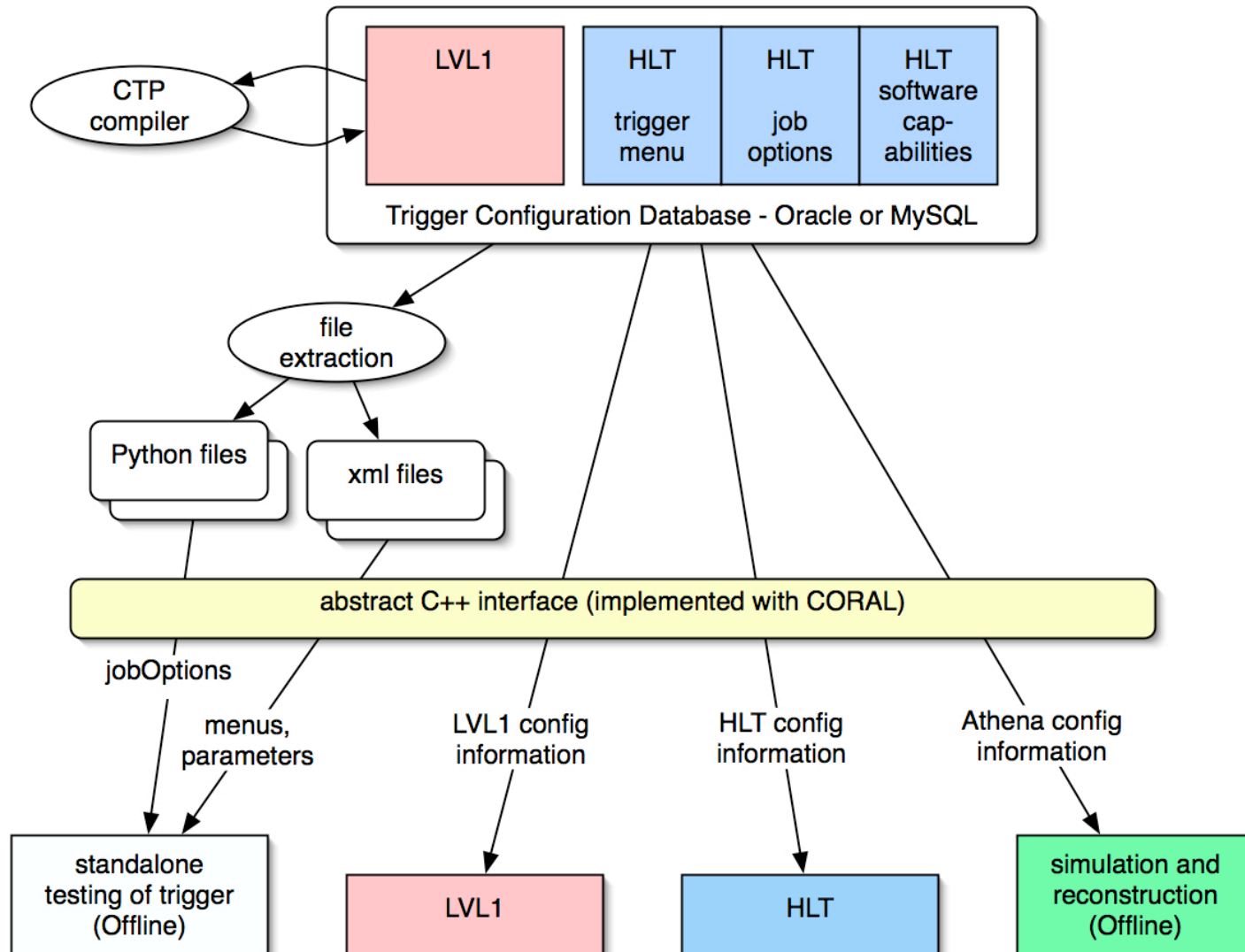
II. **direct read-out of configuration objects** from TriggerDB

- the various clients contact the TriggerDB directly to get their configuration objects (based on the CORAL package)
- to be used for the online running
- for production jobs on the grid (simulation, reconstruction)



one C++ abstract interface for both paths to be used online and offline

Data Retrieval for running - synopsis



Results of first tests

- Software: read options of a trigger job from the **DB** and materialize these into a **Python set**
 - This resembles normal Athena library configuration
 - Makes a few queries to the DB
- Used small testbed of **six nodes** running the HLT software
- Set up trigger job for **LVL2 Muon Identification** on these nodes
- The six nodes **configure correctly** from the DB
 - ~1 second total to access the MySQL configuration data
- Further investigations:
 - **Integration** with HLT menus and LVL1
 - **Scalability** (to be tested in the Large Scale Tests - end of 2006)

Status and outlook

- Database schema
 - ✓ completed, minor changes ongoing in HLT part
- Population of the TriggerDB
 - ✓ Scanning of release capabilities (C++ sources) in place, including restriction of Offline SW to subset needed for HLT
 - ✓ jobOptions for basic HLT menus in place (via manual process)
 - More automation desirable for later (catch properties at exec time)
- TriggerTool
 - ✓ LVL1 part in place
 - HLT part partly in place, under further development
- Data retrieval interfaces
 - ✓ Path via file extraction in place
 - ✓ First tests of direct path for HLT successful
 - Direct path under further development