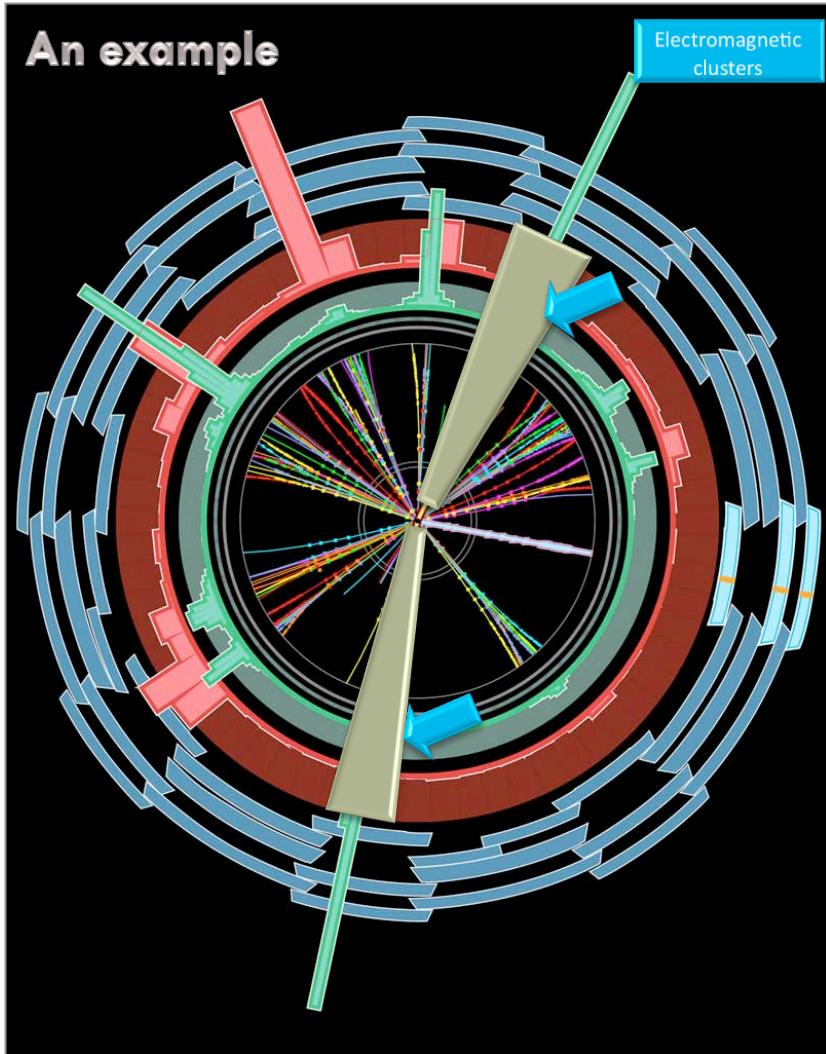


TRIGGER DATA FOR PHYSICS ANALYSIS

Ricardo Gonçalo – Royal Holloway

ATLAS Software Tutorial – 22nd to 24th April 2009

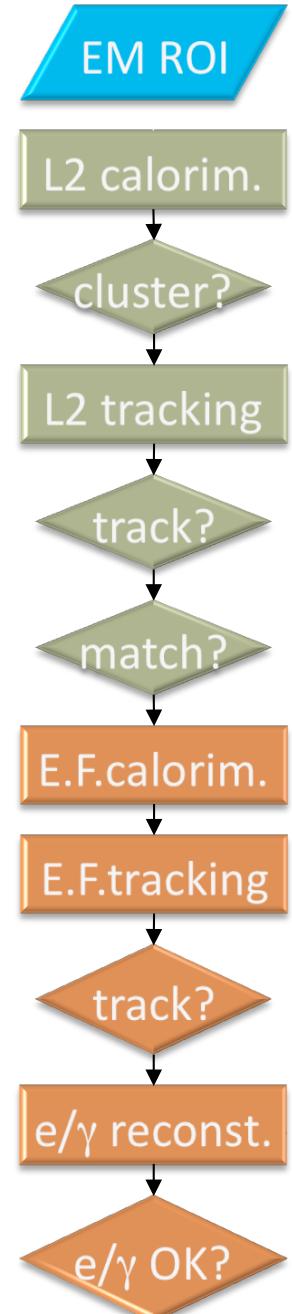
The Trigger



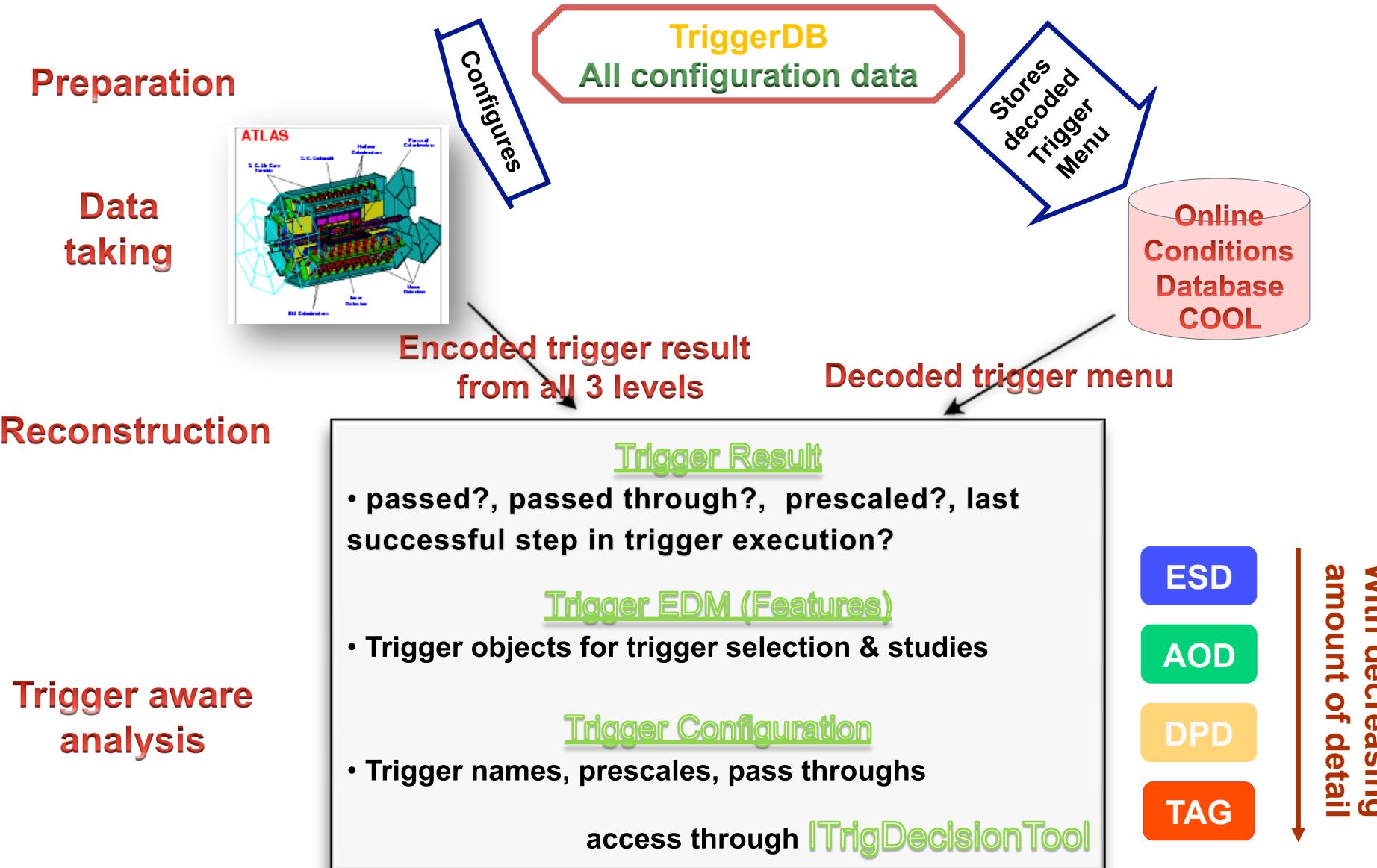
Level1:
Region of Interest is found and position in EM calorimeter is passed to Level 2

Level 2 seeded by Level 1
• Fast reconstruction algorithms
• Reconstruction within RoI

Ev.Filter seeded by Level 2
• Offline reconstruction algorithms
• Refined alignment and calibration



Trigger Configuration + Data



- L1 Items: name, version, CTP-Id, prescale
 - To come: bunch-group for each item
 - Also to come: Definition of L1Items in terms of thresholds
- HLT Chains: name, version, level, counter, prescale, trigger elements
- Streams: chains feeding into each stream
- Chain-groups: chains belonging to each group
- Bunch-groups: name of each of the 8 BG

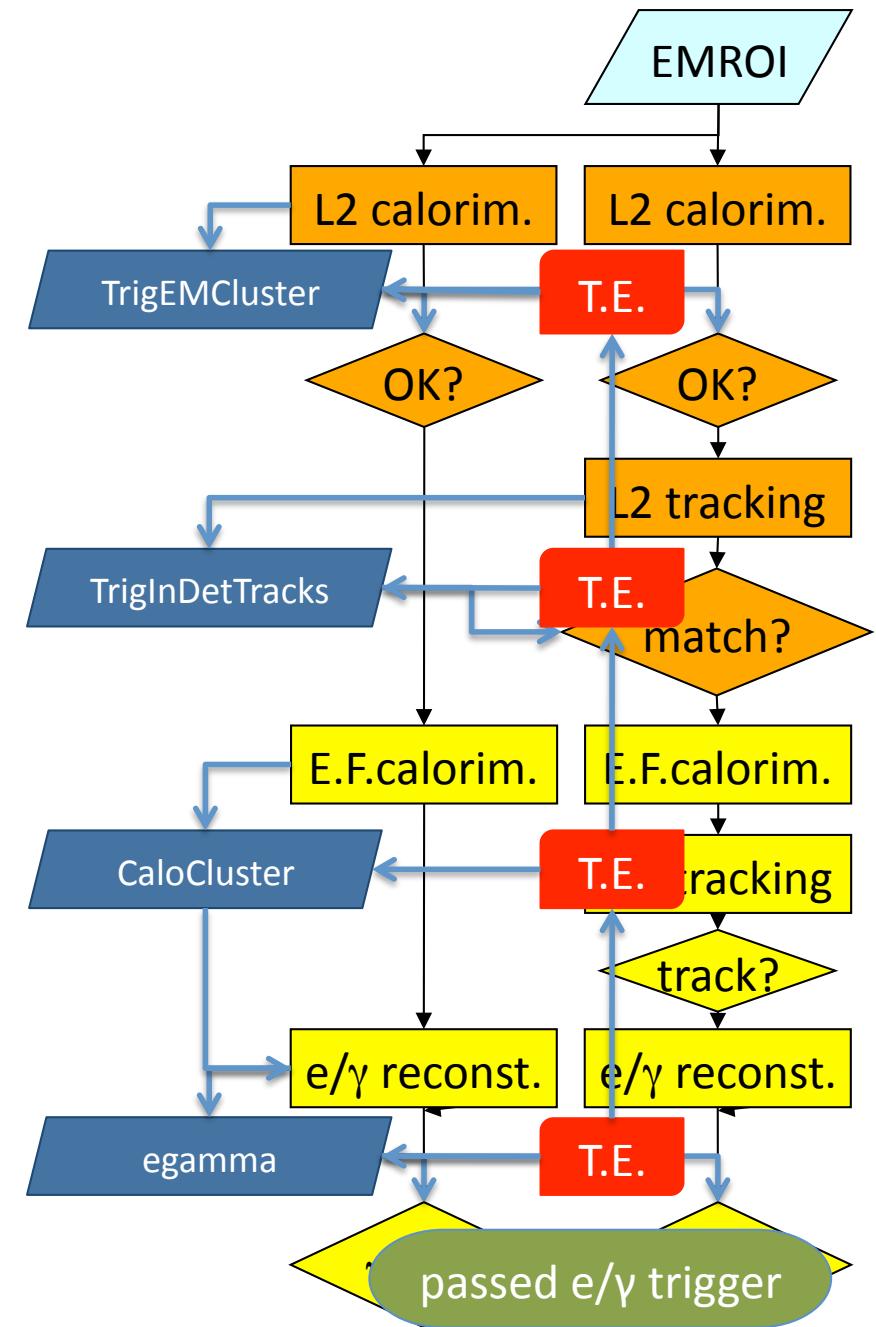
FEX

features

HYPO

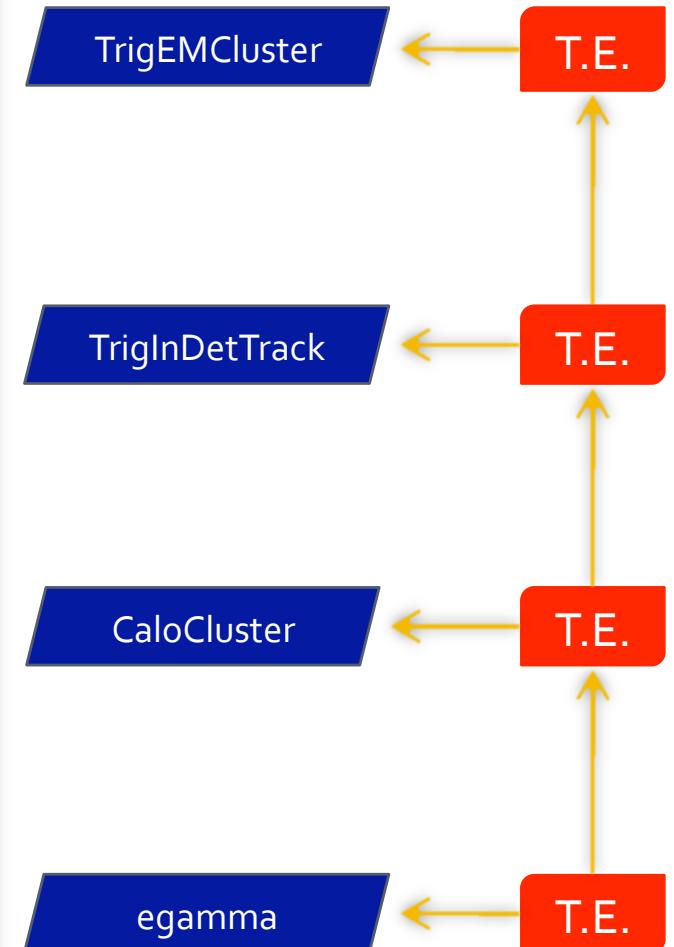
FEature eXtraction algos produce
Features on which selection in
HYPothesis algos is based

- Chain:
 - Started, if seed has fired and chain is not **PRESCALED**
 - Stopped **AT STEP**, if a HYPO is not passed
 - Last HYPO passed → **CHAIN PASSED**
- Event:
 - Passed, if at least one EF chain is passed
 - Put into all streams that are associated with any passed EF chain
- Trigger information in
 - **TRIGGER DECISION** + **ESD** **AOD** **DPD** **TAG**
 - **TRIGGER FEATURES** + **decision** **features**
 - **TRIGGER NAVIGATION** + **T.E.**
 - **CONFIGURATION**



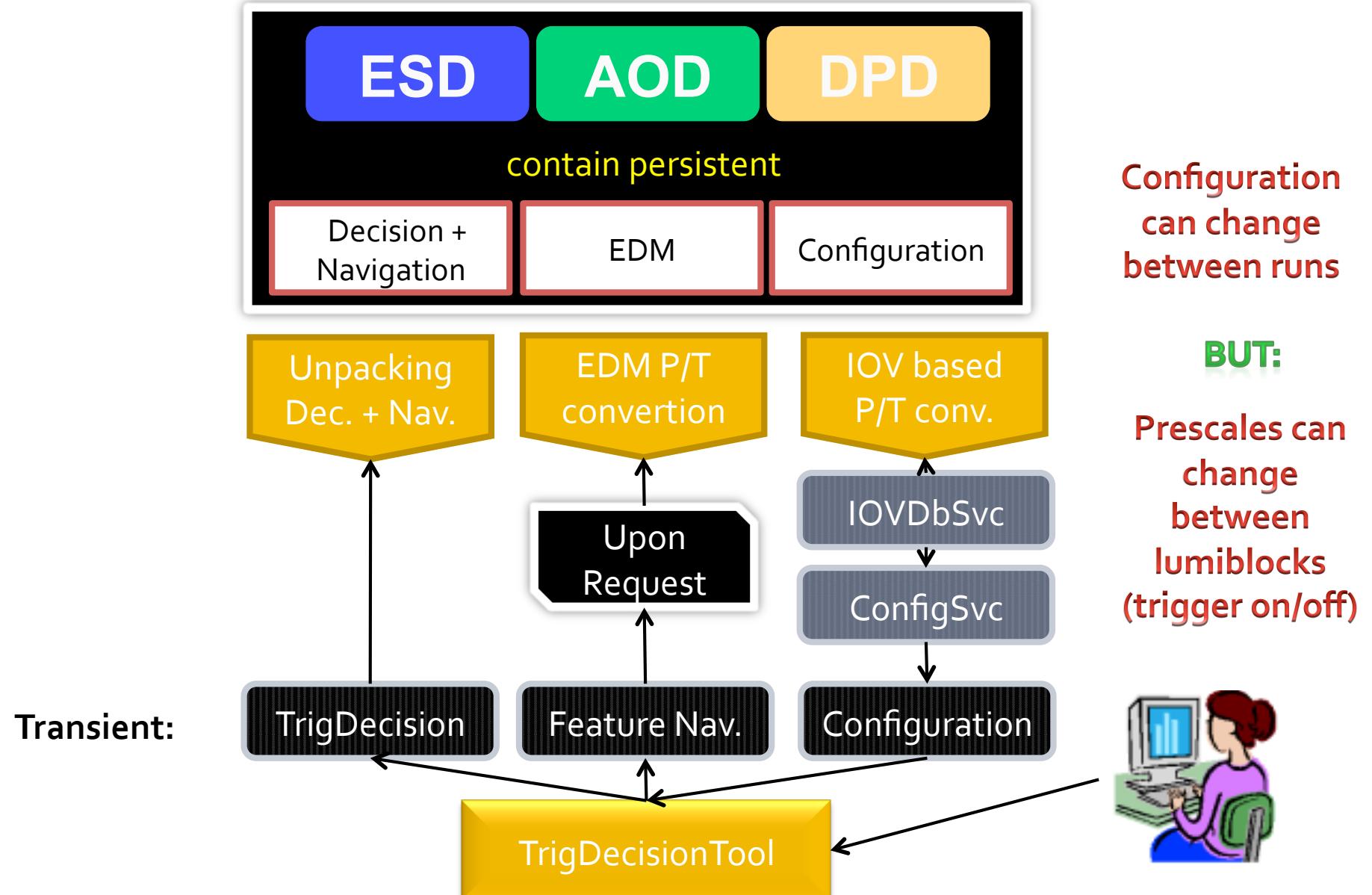
What's there? – Features

- Features can be retrieved through the “TriggerNavigation” using the TrigDecisionTool
- Features are created by FEX algorithms. They appear in StoreGate in containers according to FEX name. A FEX also creates a “TriggerElement” (TE)
 - A TE is used as handle to the feature
 - A TE has a pass/fail state set by the HYPO corresponding to the FEX
- So the navigation can give you the TE's for all the FEX that run in a chain
 - Or just those that passed the last step in the chain
 - From there you get the features (type templated)
 - This is the correct way to retrieve the features for each RoI



- Analysis based on single trigger chain or an 'OR' of a few chains
- **Chain definition** – algorithms, cuts, multiplicities – do not change during a run, but can change between runs
 - Important for analysis on DPD, where multiple runs are merged
- **Prescales at LVL1 or at HLT** can change between luminosity blocks
 - A negative prescale means that this trigger is off. This is important for calculating the integrated luminosity

Bringing it Together – TrigDecisionTool



Migration to new TrigDecisionTool in rel. 15.2.0!!!

- The standard user interface to trigger data for physics analysis
- Major change from release 15.2.0 to:
 - Provide new functionality
 - Simplify the API
 - Increase robustness
 - Allow easy use in both Athena and ARA
- See last tutorial for old version (<15.2.0)
 - <http://indico.cern.ch/conferenceDisplay.py?confId=55063>
- The two versions coexist in Athena for now (old one won't be supported for long)
 - In different namespaces: **old** TrigDec::/ **new** Trig:: and TrigDefs::
- Documentation:
 - Wiki: <https://twiki.cern.ch/twiki/bin/view/Atlas/TrigDecisionTool15>
 - Examples: [Trigger/TrigAnalysis/TrigAnalysisExamples](#)

Usage: Just like any other Tool

1. ToolHandle to a TrigDecisionTool in your algorithm header
2. Initialize ToolHandle in constructor as public tool
3. Retrieve tool in initialization
4. Use tool in execute()

```
private:  
    ToolHandle<Trig::TrigDecisionTool> m_trigDec;
```

```
MyAlgo::MyAlgo(const std::string &name, ...):  
    m_trigDec("Trig::TrigDecisionTool/TrigDecisionTool")  
{  
    declareProperty("TrigDecisionTool", m_trigDec, \  
    "The tool to access TrigDecision");  
}
```

```
StatusCode sc = m_trigDec.retrieve();
```

```
(*m_log) << MSG::INFO  
<< "L2_mu.* passed/failed: "  
<< m_trigDec->isPassed("L2_mu.*")  
<< endreq;
```

- Important concept in the new tool: ChainGroup

- They work as an OR of triggers
- Created from a vector of trigger names, comma-separated list or regular expressions:
- `const ChainGroup* getChainGroup(const string& names)`

```
m_all    = m_trigDec->getChainGroup(".*");
m_allL1 = m_trigDec->getChainGroup("L1_.*");
m_allL2 = m_trigDec->getChainGroup("L2_.*");
m_allEF = m_trigDec->getChainGroup("EF_.*");
```

- Can be used e.g. to get pass/fail information:

```
(*m_log) << MSG::INFO << "L1: " << m_allL1->isPassed() << endreq;
(*m_log) << MSG::INFO << "L2: " << m_allL2->isPassed() << endreq;
(*m_log) << MSG::INFO << "EF: " << m_allEF->isPassed() << endreq;
```

- Can be created on the fly:

```
// this creates on the fly soem chain groups and check if they passed
(*m_log) << MSG::INFO << "L2_mu.*" << m_trigDec->isPassed("L2_mu.*") << endreq;
(*m_log) << MSG::INFO << "L2_j.*" << m_trigDec->isPassed("L2_j.*") << endreq;
```

Accessing features created by a chain

- Use chains or chain groups to access trigger features produced by each trigger/group
 - FeatureContainer gives access to Trigger Elements and trigger objects for simple or complex triggers

```
FeatureContainer f = m_trigDec->features("L2_e15");
FeatureContainer::combination_const_iterator it;

for (it = f.getCombinations().begin(); it !=  

     f.getCombinations().end(); ++it) {  

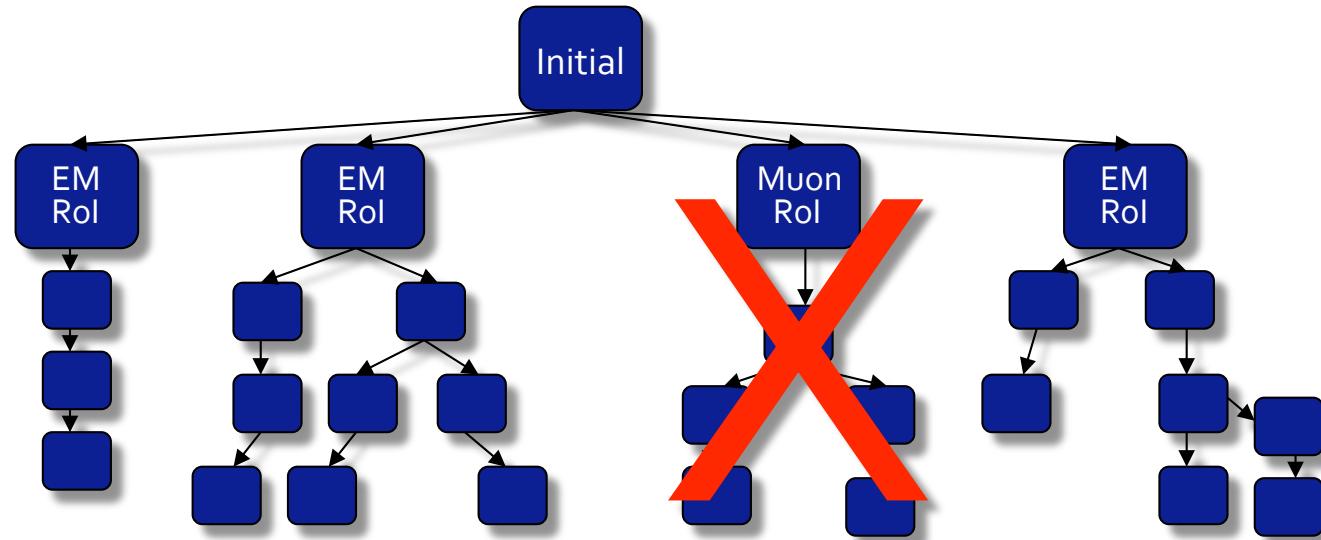
    std::vector< Feature<TrigRoiDescriptor> > initRois =  

        cIt->get<TrigRoiDescriptor>("initialRoI");  

}
```

- Combinations allows to retrieve e.g. combinations of objects satisfying combined triggers like EF_tau16_2j23

Trigger information slimming



- Needed for producing custom DPDs (e.g. Physics DPDs or user DPDs)
- **Feature Removal** – remove objects according to type and FEX
- **Pruning** – remove entire Rols (see example above)
- **Squeezing** – compress navigation information (no information loss)
- **Slimming** – remove chains belonging to groups or streams
- More info in:
<https://twiki.cern.ch/twiki/bin/view/Atlas/HLTTrigNavigationSlimming>

- Run-summary page: trigger chains, prescales, rates – web based
 - Query single run
- AtlCoolTrigger.py: same, but command line
 - Views many runs
- TriggerTool: Java based GUI to browse the TriggerDB (replica) for all information
- <http://trigconf.cern.ch> as web front-end to TriggerTool and AtlCoolTrigger.py

- Trigger Menu and L1 rates stored in COOL, HLT rates coming. Quick access via
 - Run summary pages (WEB based)
 - <http://atlas-service-db-runlist.web.cern.ch/atlas-service-db-runlist/query.html>
 - Trigger names, rates
 - AtlCoolTrigger.py (command line tool)
 - AtlCoolTrigger –r 91000-99000 (many run summary)
 - AtlCoolTrigger –v –m –r 90272 (single run menu)
 - Prints keys, trigger menus, streams, allows diff-ing of menus in different runs

TriggerTool

- Java based front end to TriggerDB, launch from the web (Java web-start):

<http://www.cern.ch/triggertool>

- Overview of all trigger configurations
- Detailed and convenient investigation of trigger menus
 - Trigger definition L₁->L₂->EF: prescales, threshold algorithms, selection criteria, streaming information, etc.
- Possibility to compare different trigger configurations

The TriggerTool interface consists of two main windows. The top window is a list view showing a table of trigger configurations. The columns include ID, Name, Version, Comment, Origin, and Creator. The bottom window is a detailed configuration editor for a specific trigger, showing tabs for L1 CTP Cables, L1 Monitoring, L2 Setup, EF Setup, Histograms, Consistency Check, and Search. The L2 Setup tab is currently active, displaying a tree structure of trigger components and their properties.

Web Interface to COOL and the TriggerDB

- Web interface <http://trigconf.cern.ch>
 - Runs TriggerTool on the server, result presented as dynamic html pages

Listing of Trigger Keys by Run
91000-91500 Example: 91000-92000,90275,93500-

Listing trigger configurations
run OR smk L1 psk and HLT psk Show Configuration
Specify either run number or set of configuration keys to display the trigger configuration

Comparing trigger configurations
run OR smk L1 psk and HLT psk Compare Configurations
Specify either run number or set of configuration keys for comparison with the trigger configuration above

1. Search run-range

2. Run list

Also with simple comparison functionality

Click on L1 prescale to get to full trigger menu display. Mark two menus and click

run	Start Time	SMK	HLT PSK	LVL1 PSKs
91001	Wed Oct 8 15:28:35 2008	351	368	520 (1) <input type="checkbox"/>
91003	Wed Oct 8 15:36:43 2008	351	368	520 (1) <input type="checkbox"/>
91007	Wed Oct 8 15:54:39 2008	351	368	520 (1) <input checked="" type="checkbox"/>
91043	Wed Oct 8 21:03:23 2008	355	373	520 (1) <input checked="" type="checkbox"/>
91044	Wed Oct 8 21:27:12 2008	351	369	520 (1) <input type="checkbox"/>
91045	Wed Oct 8 21:37:16 2008	351	369	520 (1) <input type="checkbox"/>
91047	Wed Oct 8 22:04:03 2008	351	369	520 (1) <input type="checkbox"/>
91056	Wed Oct 8 22:42:37 2008	351	369	520 (1) <input type="checkbox"/>
91059	Wed Oct 8 23:57:22 2008	351	368	520 (1) <input type="checkbox"/> 520 (2) <input type="checkbox"/> 521 (3) <input type="checkbox"/>
91060	Thu Oct 9 01:28:35 2008	351	368	520 (1) <input type="checkbox"/> 522 (2-5) <input type="checkbox"/> 520 (6) <input type="checkbox"/>
91077	Thu Oct 9 12:25:35 2008	351	368	515 (1) <input type="checkbox"/>
91086	Thu Oct 9 13:12:05 2008	351	368	515 (1-2) <input type="checkbox"/> 9 (3) <input type="checkbox"/>
91112	Thu Oct 9 14:13:32 2008	357	356	519 (1) <input type="checkbox"/>
91120	Thu Oct 9 15:20:10 2008	357	356	520 (1) <input type="checkbox"/>

3. Trigger configuration (browsable)
(definition, algorithms, selection cuts)

SMK 369 HLT Prescales Key: 390 Lvl1 Prescales Key: 541
If more advanced browsing is needed please launch TriggerTool [Follow the link if you have trouble to launch it](#)

Streams
[L1Calo](#) | [RNDM](#) | [TGCwBeam](#) | [MBTS_BCM_LUCID](#) | [RPCwBeam](#) | [CosmicMuons](#) | [DCosmic](#) | [DTracks](#) | [Express](#) | [BPTX](#) | [L1GlobalEM](#) | [CosmicDownwardMuons](#) | [Tile](#) | [LArCells](#)

EF chain	PS	PT	STP	L2 chain	PS	PT	L1 item	LL prescale
e10_low	1	1	1	e10_low	1	1	EM1	1
e10_low_pass1_2	1	1	1	e10_low_pass1_2	1	1	EM7	1
e10_low_passEE	1	1	1	e10_low_passEE	1	1	EM7	1
e10_medium	1	1	1	e10_medium	1	1	EM7	1
e10	1	1	1	e10	1	1	EM7	1
tau10_low	1	1	1	tau10_low	1	1	TAU10	1
tau10_low_pass	1	1	1	tau10_low_pass	1	1	TAUS	1
tauNoC_d	1	1	1	tauNoC_d	1	1	T	1
t5	1	1	1	t5	1	1	J10	1
t10	1	1	1	t10	1	1	J20	1
t20	1	1	1	t20	1	1	J110	1
B110	1	1	1	B110	1	1	J118	1
P118	1	1	1	P118	1	1	2P118	1
Z118	1	1	1	Z118	1	1		
ta150	1	1	1	ta150	1	1	TAU150	1
tae20	1	1	1	tae20	1	1	TAUS	1
tau10_low_loose	1	1	1	tau10_low_loose	1	1	TAUH	1
t19	1	1	1	t19	1	1	J23	1
tae_debug_EFonly	1	1	1	tae_debug_EFonly	1	1	J19	1
trk10	1	1	1	trk10	1	1	TAUE	1
trk15	1	1	1	trk15	1	1	TAUH	1
trk9_id	1	1	1	trk9_id	1	1	TAUS	1
tauNoC_d_TauRecNoTopo	1	1	1	tauNoC_d_TauRecNoTopo	1	1	TAUS	1
tauNoC_d_Calo	1	1	1	tauNoC_d_Calo	1	1	EM3	1
t5_low	1	1	1	t5_low	1	1	EM3	1
s5_Noc_d_05can	1	1	1	s5_Noc_d_05can	1	1	EM3	1
s5_Noc_d_TRT	1	1	1	s5_Noc_d_TRT	1	1	EM3	1
s5_Noc_d_ST1s	1	1	1	s5_Noc_d_ST1s	1	1	EM3	1
MBTS_lowBackTrk	1	1	1	MBTS_lowBackTrk	1	1	EM3	1
re20_noMu	1	1	1	re20_noMu	1	1	EM20	1
re10_calib	1	1	1	re10_calib	1	1	EM7	1
re20_noiseSupp	1	1	1	re20_noiseSupp	1	1	EM20	1
re20_FEB	1	1	1	re20_FEB	1	1	EM20	1
re20_Mu	1	1	1	re20_Mu	1	1	EM3	1
re150_noMu	1	1	1	re150_noMu	1	1	TAU150	1
SingleBeam_RNDM	1	1	1	SingleBeam_RNDM	1	1		

RNDM

EF chain	PS	PT	STP	L2 chain	PS	PT	L1 item	LL prescale
Mt5dTrk	1	1	1	Mt5dTrk	1	1	RNDM_FILLED	-1
Mt10dTrk	1	1	1	Mt10dTrk	1	1	RNDM_FILLED	-1
Mt15dTrk	1	1	1	Mt15dTrk	1	1	RNDM_FILLED	-1
SingleBeam_RNDM	1	1	1	SingleBeam_RNDM	1	1		

TGCwBeam

EF chain	PS	PT	STP	L2 chain	PS	PT	L1 item	LL prescale
MbTGC_SoFed_pisect	1	1	1	MbTGC_SoFed_pisect	1	1	MU0_TGC_HALO	-1
MbTGC_Mes	1	1	1	MbTGC_Mes	1	1	MU0_TGC_HALO	-1
MbTGC_Mhs_2	1	1	1	MbTGC_Mhs_2	1	1	MU0_TGC_HALO	-1
MbTGC_SoFed_sct	1	1	1	MbTGC_SoFed_sct	1	1	MU0_TGC_HALO	-1

- `checkTrigger.py AOD.pool.root`
 - Runs over ESD/AOD/DPD and presents detailed (chain-wise) counts of the trigger decision
- `checkTriggerConfig.py -d AOD.pool.root`
 - Runs over ESD/AOD/DPD and presents detailed trigger configuration(s) in the file
 - Shows multiple configurations, in DPD possible/likely

Additional Information

General Trigger info: <https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerUserPages>

Tutorials: <https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerSoftwareTutorialPage>

Trigger Event Data Model (EDM) : <https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerEDM>

TrigDecisionTool – all releases/versions:

Twiki: <https://twiki.cern.ch/twiki/bin/view/Atlas/TrigDecisionTool>

Athena examples:

TrigAnalysisExamples package:

http://atlas-computing.web.cern.ch/atlas-computing/links/nightlyDevDirectory/AtlasOffline/latest_doxxygen/InstallArea/doc//TrigAnalysisExamples/html/

TrigDecisionChecker (pre-15.X.0 for the moment):

<http://atlas-sw.cern.ch/cgi-bin/viewcvs-atlas.cgi/offline/Trigger/TrigValidation/TrigValAlgs/TrigValAlgs/TrigDecisionChecker.h?revision=1.2&view=markup>

ARA example:

ARA: <https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerARA14>

Trigger Configuration: <https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerConfiguration?topic=TrigDecisionTool>

Run summary pages: <http://atlas-service-db-runlist.web.cern.ch/atlas-service-db-runlist/query.html>

(include link to start up the TriggerTool)

Trigger information slimming: <https://twiki.cern.ch/twiki/bin/view/Atlas/HLTTrigNavigationSlimming>

TriggerMenu working group: <https://twiki.cern.ch/twiki/bin/view/Atlas/TriggerPhysicsMenu>

Help! Hypernews forum hn-atlas-TriggerHelp@cern.ch