CMS Offline Software

Liz Sexton-Kennedy

Jterm IV or Aterm

August 4, 2009 – Fermilab

Introduction

The goal of this talk is not to teach you how to use the software. That is what today's tutorials are for. Instead:

- ➡ Introduce the offline software concepts and organization - CMSSW
- ➡ I will speak about the offline software as a project. How is the software project broken down into subprojects and who are the current leaders. These are the people to contact if you want to become involved.

What is it that the offline does from day to day.

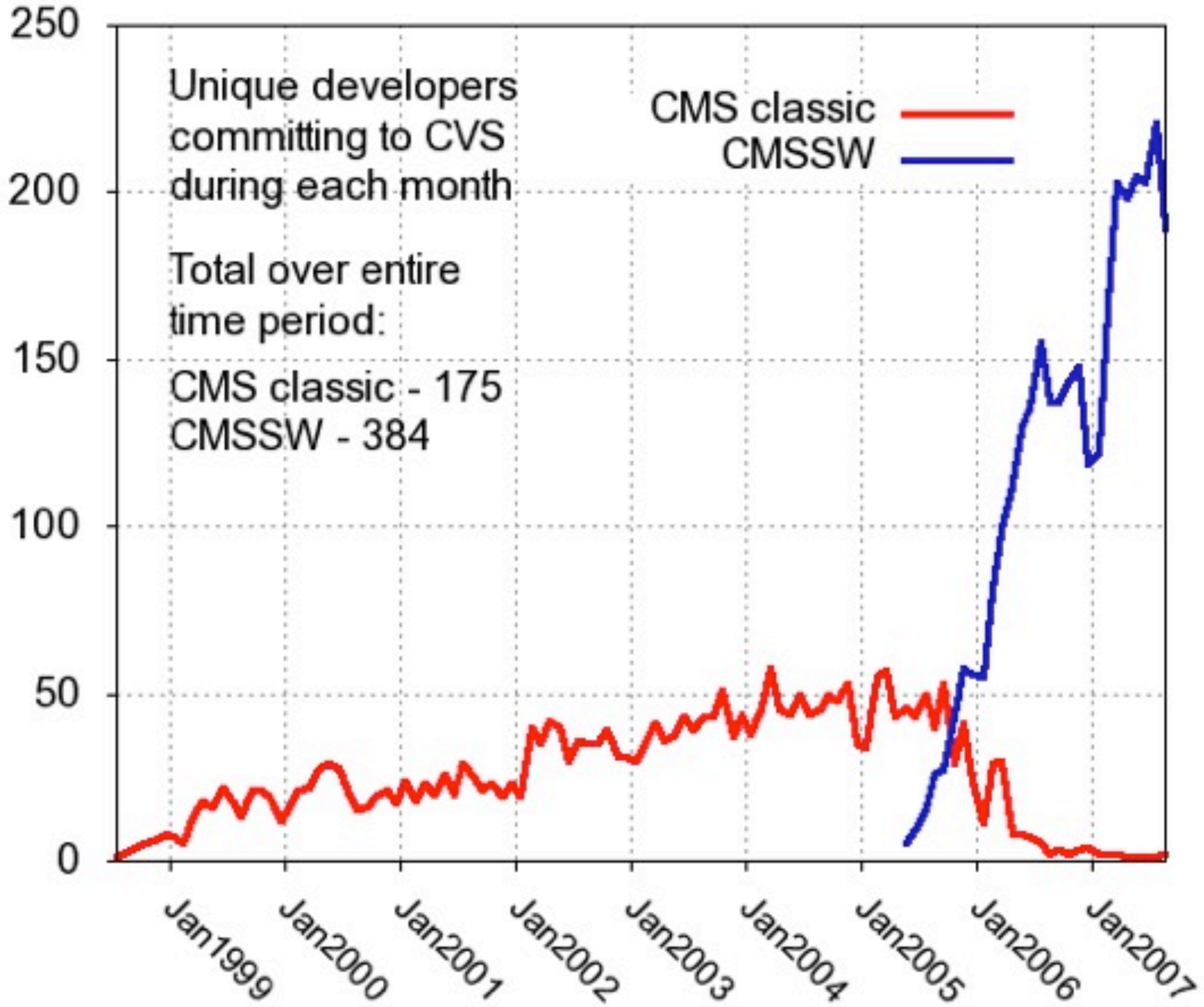
How can you take advantage of the work being done for offline.

- ➡ Release Integration
- ➡ cmsRun, cmsDriver
- ➡ FWLite

Summary, and concluding remarks

CMSSW Introduction

CMS is now on it's second generation of offline software. Use of the first generation software systems is long gone now, but much of the work that went into these projects (ORCA, OSCAR, COBRA, IGUANA) was reused by "porting" it to CMSSW.

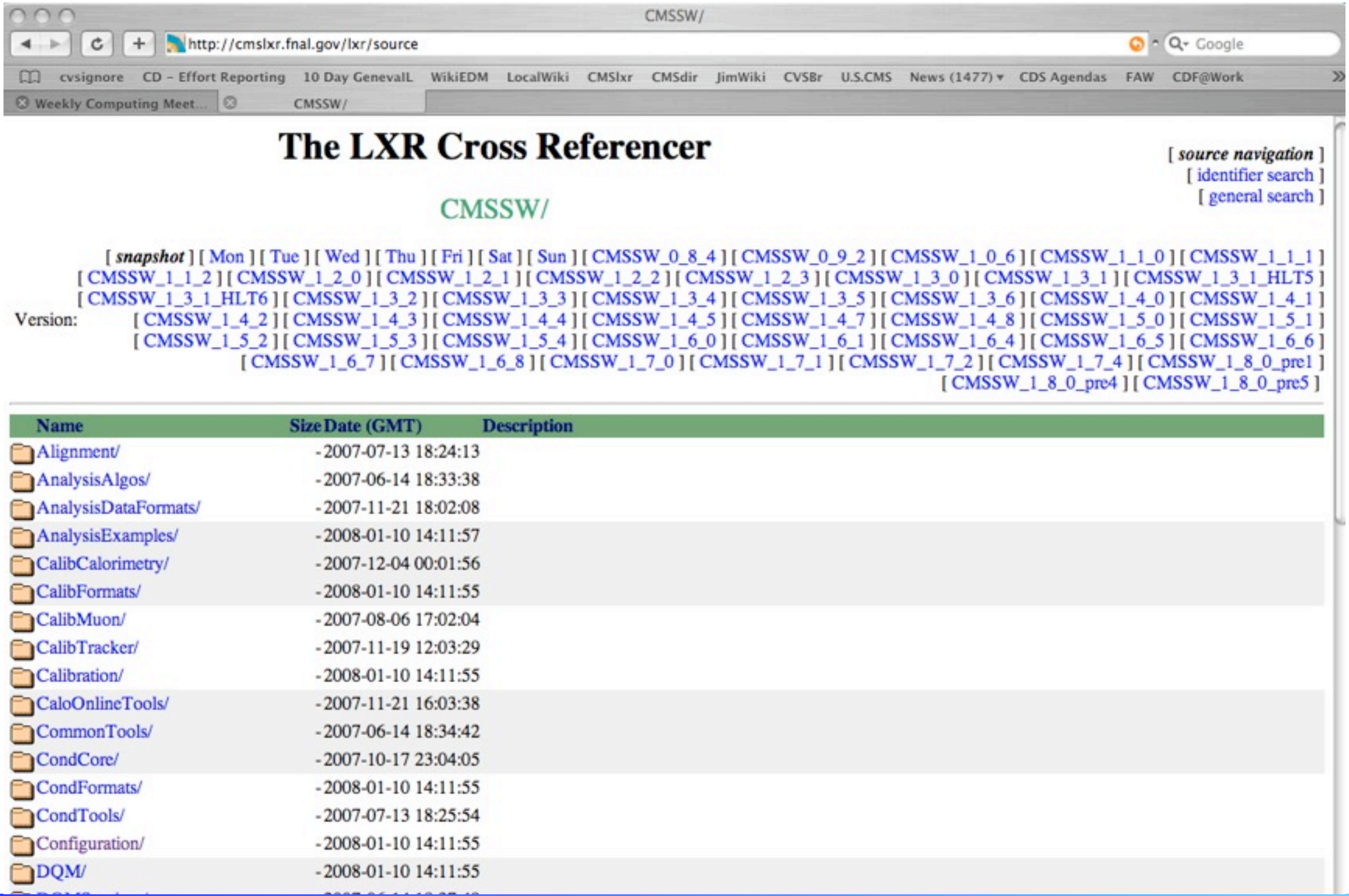


CMSSW is the full suite of offline software in the new architecture.

Code for online data taking Simulation, Primary reconstruction, and Physics analysis.

Code Organization

The thousand (or so) packages that make up the project are organized into sub-systems with names that should be suggestive of their purpose:



The LXR Cross Referencer

[***source navigation***]
[identifier search]
[general search]

CMSSW/

Version: [***snapshot***] [Mon] [Tue] [Wed] [Thu] [Fri] [Sat] [Sun] [CMSSW_0_8_4] [CMSSW_0_9_2] [CMSSW_1_0_6] [CMSSW_1_1_0] [CMSSW_1_1_1] [CMSSW_1_1_2] [CMSSW_1_2_0] [CMSSW_1_2_1] [CMSSW_1_2_2] [CMSSW_1_2_3] [CMSSW_1_3_0] [CMSSW_1_3_1] [CMSSW_1_3_1_HLT5] [CMSSW_1_3_1_HLT6] [CMSSW_1_3_2] [CMSSW_1_3_3] [CMSSW_1_3_4] [CMSSW_1_3_5] [CMSSW_1_3_6] [CMSSW_1_4_0] [CMSSW_1_4_1] [CMSSW_1_4_2] [CMSSW_1_4_3] [CMSSW_1_4_4] [CMSSW_1_4_5] [CMSSW_1_4_7] [CMSSW_1_4_8] [CMSSW_1_5_0] [CMSSW_1_5_1] [CMSSW_1_5_2] [CMSSW_1_5_3] [CMSSW_1_5_4] [CMSSW_1_6_0] [CMSSW_1_6_1] [CMSSW_1_6_4] [CMSSW_1_6_5] [CMSSW_1_6_6] [CMSSW_1_6_7] [CMSSW_1_6_8] [CMSSW_1_7_0] [CMSSW_1_7_1] [CMSSW_1_7_2] [CMSSW_1_7_4] [CMSSW_1_8_0_pre1] [CMSSW_1_8_0_pre4] [CMSSW_1_8_0_pre5]

Name	Size	Date (GMT)	Description
Alignment/	-	2007-07-13 18:24:13	
AnalysisAlgos/	-	2007-06-14 18:33:38	
AnalysisDataFormats/	-	2007-11-21 18:02:08	
AnalysisExamples/	-	2008-01-10 14:11:57	
CalibCalorimetry/	-	2007-12-04 00:01:56	
CalibFormats/	-	2008-01-10 14:11:55	
CalibMuon/	-	2007-08-06 17:02:04	
CalibTracker/	-	2007-11-19 12:03:29	
Calibration/	-	2008-01-10 14:11:55	
CaloOnlineTools/	-	2007-11-21 16:03:38	
CommonTools/	-	2007-06-14 18:34:42	
CondCore/	-	2007-10-17 23:04:05	
CondFormats/	-	2008-01-10 14:11:55	
CondTools/	-	2007-07-13 18:25:54	
Configuration/	-	2008-01-10 14:11:55	
DQM/	-	2008-01-10 14:11:55	

Documentation

There are now much entry paths into learning the software system:

- ➨ WorkBook - This is the initial starting point for people new to CMS software and computing. It is really intended for people to read through it, and work out the examples and tutorials it contains. The home of all of the tutorials you will see today is here.
- ➨ SWGuide - This is a full description of the CMS software including
  - ✓ The software architecture
  - ✓ Detailed descriptions of the algorithms
  - ✓ Instructions for analysis and validation
- ➨ Reference Manual - mixture of auto-generated Doxygen and hand written pages always check the last edit date...

Top Level Example

Take a top level view and start from the top level "Workflow STEPs" which are tested at the 10K level for each release. They should all work "out of the box" See:

http://cmslxr.fnal.gov/lxr/source/Configuration/PyReleaseValidation/data/cmsDriver_standard_hlt.txt?raw=1

and look for cmsDriver commands like this (step1 workflow):

```
cmsDriver.py  TTbar_Tauola_cfi.py
-s
GEN:ProductionFilterSequence,SIM,DIGI,L1,DIGI2RAW,HLT,RAW2DIGI,L1Reco
-n 10  --conditions FrontierConditions_GlobalTag,STARTUP31X_V2::All
--datatier 'GEN-SIM-DIGI-RAW-HLTDEBUG'
--eventcontent FEVTDEBUGHLT
```

or pick your own favorite process to generate from here:

http://cmslxr.fnal.gov/lxr/source/Configuration/Generator/python/

to reconstruct what you produced in step 1 you run a step 2 workflow command:

```
cmsDriver.py step2 -s RAW2DIGI,L1Reco,RECO,VALIDATION
--datatier GEN-SIM-RECO
--eventcontent RECOSIM
--conditions FrontierConditions_GlobalTag,STARTUP31X_V2::All
--no_exec --datatier ALCARECO  --oneoutput --eventcontent ALCARECO
```

Glossary T(more/less)LA's

CMSSW = Compact Muon Solenoid SoftWare

EDM = Event Data Model

FW = Application Framework

AOD = Analysis Object Dataset

TDR = Technical Design Report

DAQ = Data Acquisition

FED = Front End Digitizer

HLT = High Level Trigger

PAT = Physics Analysis Toolkit – software effort

PADA = Processing and Data Access – computing effort

CVS = Code Versioning System – also called the code repository

CmsTC = CMS Tag Collector – integration tool

FW Architecture - Plugins

Applications in CMSSW are built from special shared object libraries called plugins. In practice this means that there is only one command you need to know to run most CMS applications: cmsRun <some-configuration-file>

Configurations are written in the python language.

There are two types of plugins that users define:

- Module Plugins – EDProducers, EDFilters, EDAnalyzers, and ESSources, ESProducers. These are the worker components of the FW. ED* process event data, ES* process event setup data.
- Data Object Plugins – also known as "root dictionaries" because they can also be loaded directly into the root application. These are most of the products of the above work, and form the elements of the EDM.

Basic Concept of the EDM

The data produced in the primary reconstruction farms (Tier 0 = CERN), or reprocessing farms (Tier 1s) must be immediately useful for analysis (at least once we really understand the detector).

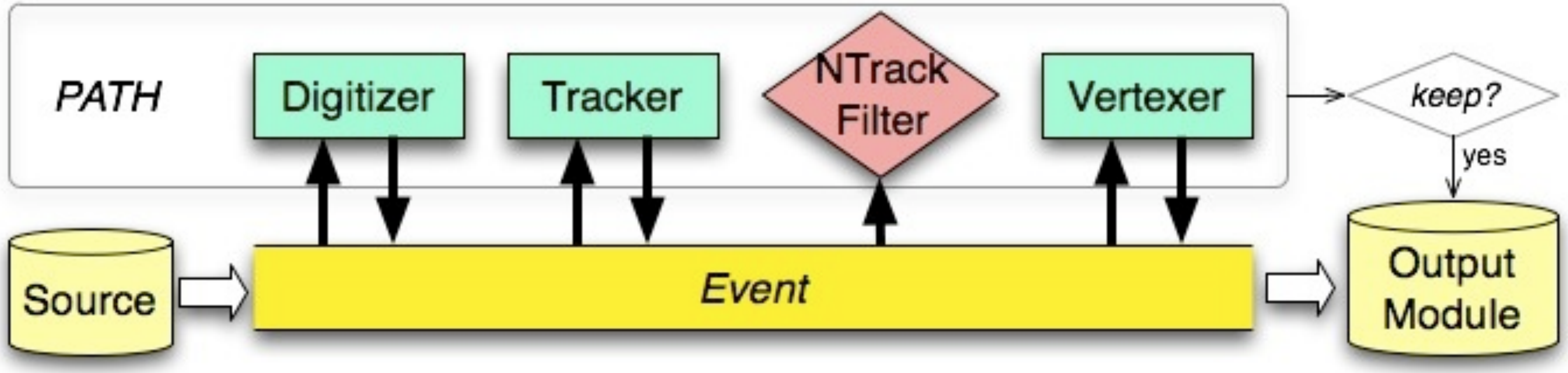
We have designed the system so that the same pool/root data file can be used in three contexts:

- Bare root.exe – treat pool file as a root input file, allows browsing of simple objects (floats, ints, and composites of them) from the root TBrowser GUI is the ONLY recommended use case.
- FWLite – a small set of CMS defined loadable libraries added to root to allow more sophisticated use with root. For the EDM this is a read-only application.
- cmsRun - input for a full framework application, could be used for a reprocessing pass or creation of more refined analysis objects, like PAT objects.

Basic Concept of the FW

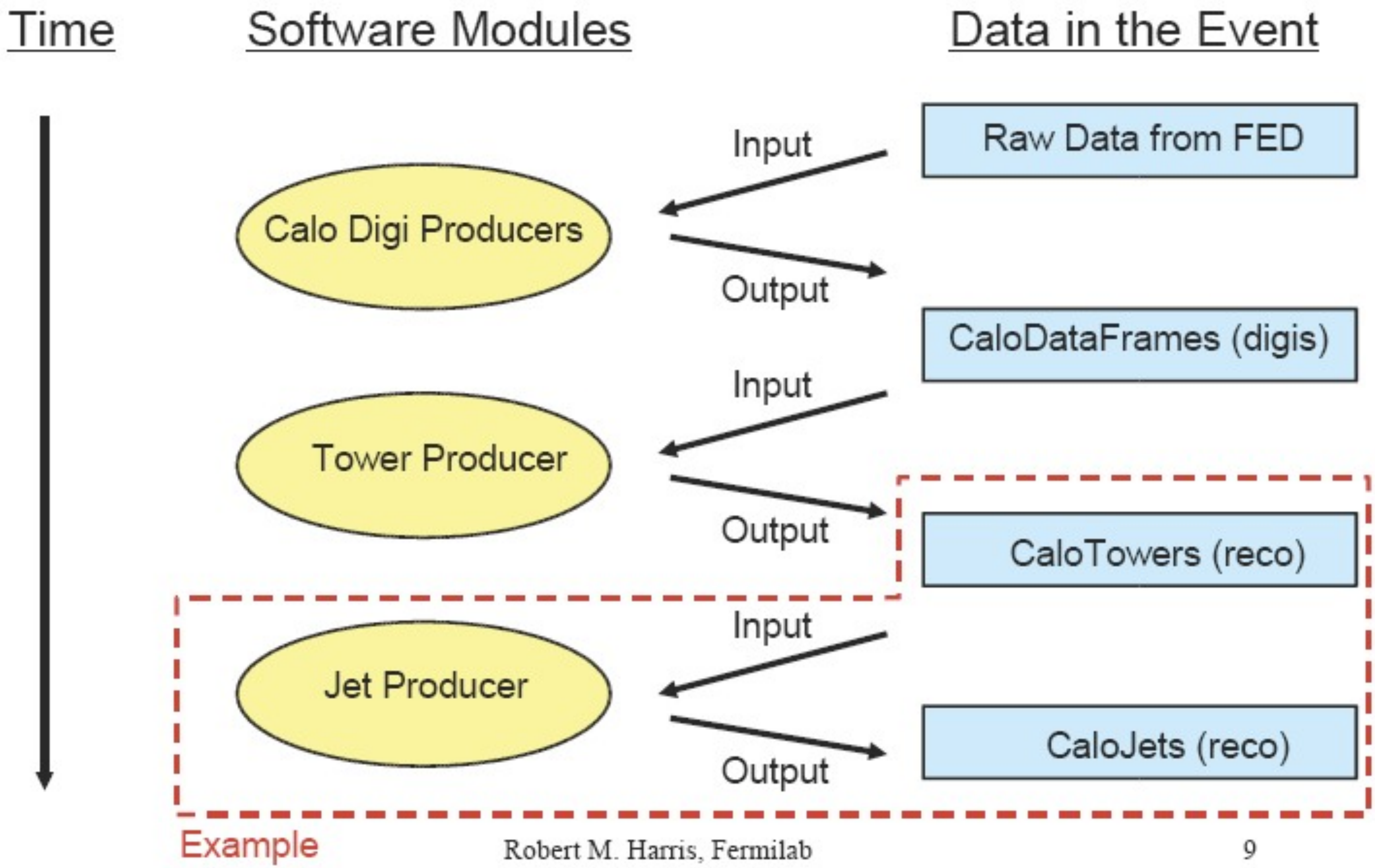
A Software Bus Model

- Start from the raw or generated data, read from a source
- Producers are scheduled to operate on the event data and produce their output which is written into the event
- Because it's modular you can inspect/debug the job at any point in the processing path. The contents of the event can be examined outside of the context of the process that made it.
- The FW automatically tracks the provenance of what is produced.
- Several instances of the same module can be run in the same application and you will still be able to uniquely identify their products, eg. JetProducer with different cone sizes. Identified by C++ type, producer label, instance name, process name



Schematic Picture of Reconstruction

A simplified example take from the calorimeter reconstruction:



Robert M. Harris, Fermilab 9

Workflows

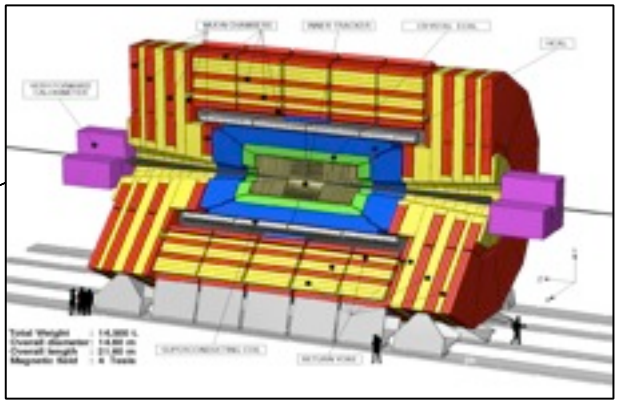
The set of processes that both MC and real data flow through.

SimHits
Pileup Events

SimHits
Signal Event

Simulation of
Electronics

Tracker
Calorimeter
Muon Det.
TriggerSim



Raw Data

STEP 1 = `GEN:ProductionFilterSequence,SIM,DIGI,L1,DIGI2RAW,HLT`

Low-level Reconstruction

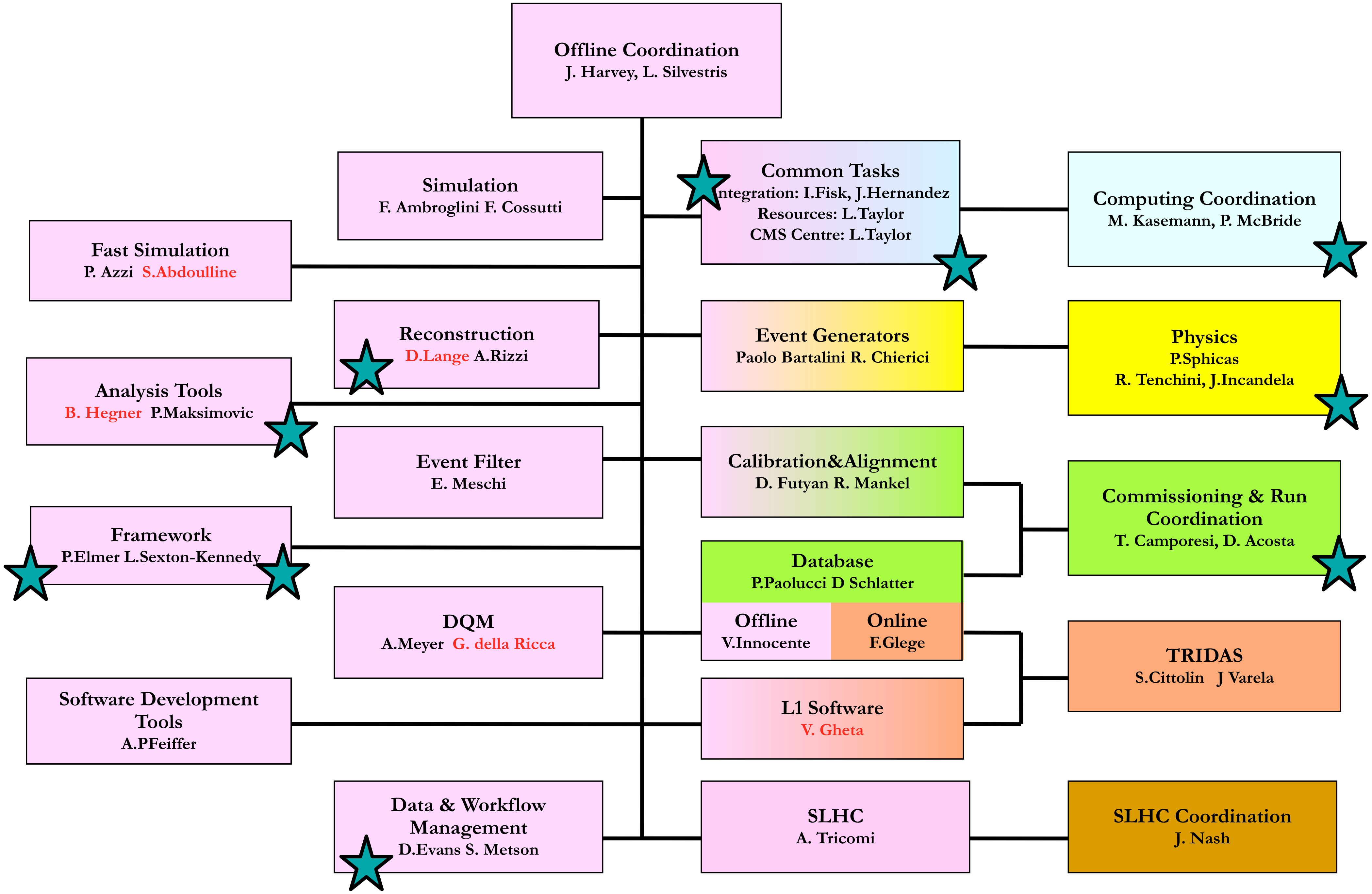
Digi's
RecHits,
Segments

High-level Reconstruction and Analysis

Tracks, Jets,
Muons, Vertices,
MET, Electrons,
Photons, b-tags,
etc...

STEP 2 = `RAW2DIGI,L1Reco,RECO,VALIDATION`

Organization



Offline Coordination
J. Harvey, L. Silvestris

Simulation
F. Ambroglini F. Cossutti

Fast Simulation
P. Azzi S.Abdoulline

Reconstruction
D.Lange A.Rizzi

Analysis Tools
B. Hegner P.Maksimovic

Event Filter
E. Meschi

Framework
P.Elmer L.Sexton-Kennedy

DQM
A.Meyer G. della Ricca

Software Development Tools
A.PFeiffer

Data & Workflow Management
D.Evans S. Metson

Common Tasks
ntegration: I.Fisk, J.Hernandez
Resources: L.Taylor
CMS Centre: L.Taylor

Event Generators
Paolo Bartalini R. Chierici

Calibration&Alignment
D. Futyan R. Mankel

Database
P.Paolucci D Schlatter

Offline
V.Innocente

Online
F.Glege

L1 Software
V. Gheta

SLHC
A. Tricomi

Computing Coordination
M. Kasemann, P. McBride

Physics
P.Sphicas
R. Tenchini, J.Incandela

Commissioning & Run Coordination
T. Camporesi, D. Acosta

TRIDAS
S.Cittolin J Varela

SLHC Coordination
J. Nash

Subprojects:

- Full and Fast Simulation, Generators, SLHC
- L1Software, EventtFilter, Calibration&Alignment, Reconstruction, DQM
- Framework, Databases, SW Dev. Tools, WMDM, Computing Integration

This Year in Releases

CMSSW_2_1_X – First data taking release, used in fall of 2008! Completely deprecated by now....

CMSSW_2_2_X - Still in pre-release series; 6 is the latest. DPG & POG content frozen before CERN Dec. shutdown. The remaining pre-releases are for bug fixes from physics, and upgrades of our highly coupled external dependencies like root and geant4.

CMSSW_3_1_X - Born from the accelerator schedule delay. It contains more then a years worth of DPG and POG developments including all of the knowledge gained by the CRAFT08 run. It has now been re-purposed as the '09 MC production run release. However it was a very painful release to create. It's generally accepted within the offline that the process used to create this, is NOT the way to create releases in the future.

CMSSW_3_2_X - Close to the content of 3_1_X, it's main purpose is to integrate the changes required for the CRAFT09 data taking.

CMSSW_3_3_X - Will be cloned from 3_2_X (probably this week, with X=3). From there it will contain new developments.

Summary & Concluding Remarks

The CMS software is large and complex. However we have a good organization and many tools to deal with this complexity.

There are still many parts of the offline which need more help. If you are a students or post-docs that has some software skills (C++ or python) and still need to fulfill your service requirement, please come see us at the LPC!

Efforts within the software project that the LPC is strong in:

- Generators
- Geant Simulation
- Tracking, Jet/Met, and Tau ID
- Data operations and WMDM development
- Framework and SWDT
- FWLite and Fireworks