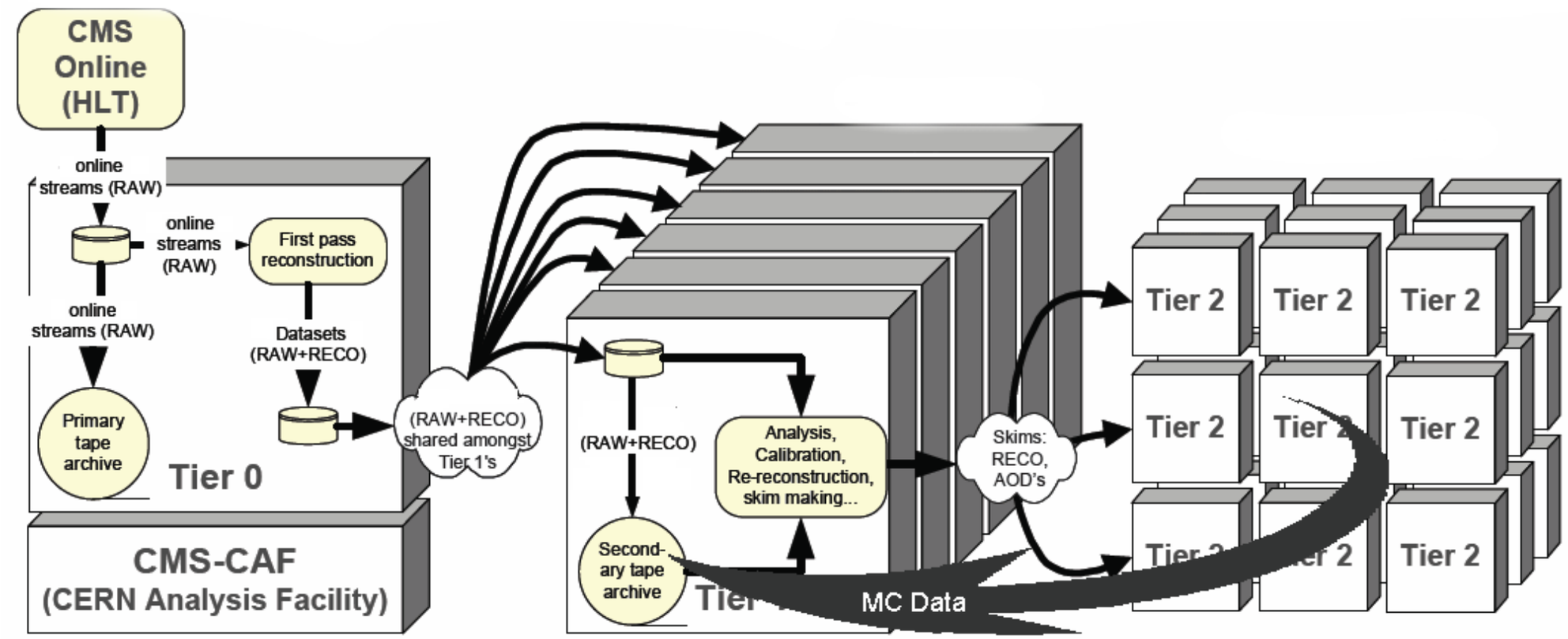
# The CMS workload management system

Stuart Wakefield (On behalf of CMS)

# Contents

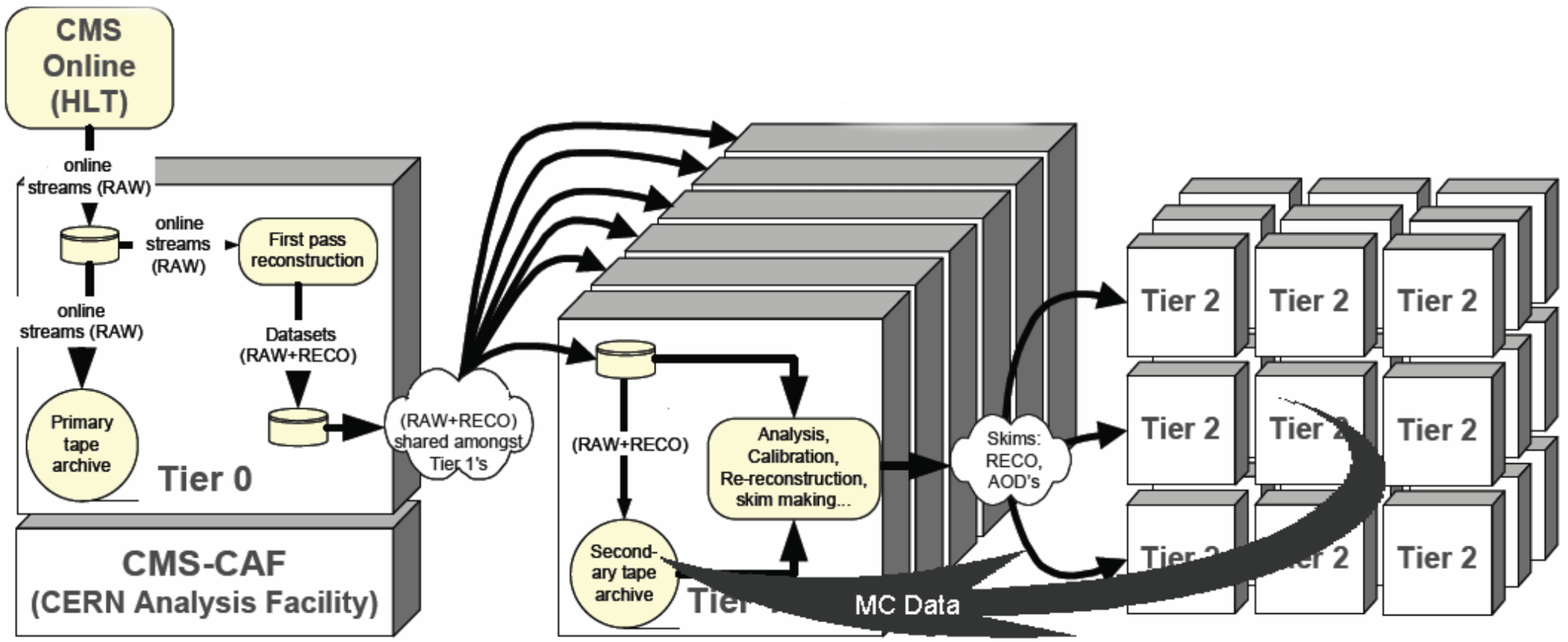
- CMS computing model.
- Previous Workload Management system.
- New WM system.
- New components
  - ReqMgr
  - WorkQueue
  - WMAgents
- Analysis
- Logging & bookkeeping.
- Usage (briefly)

# CMS computing model



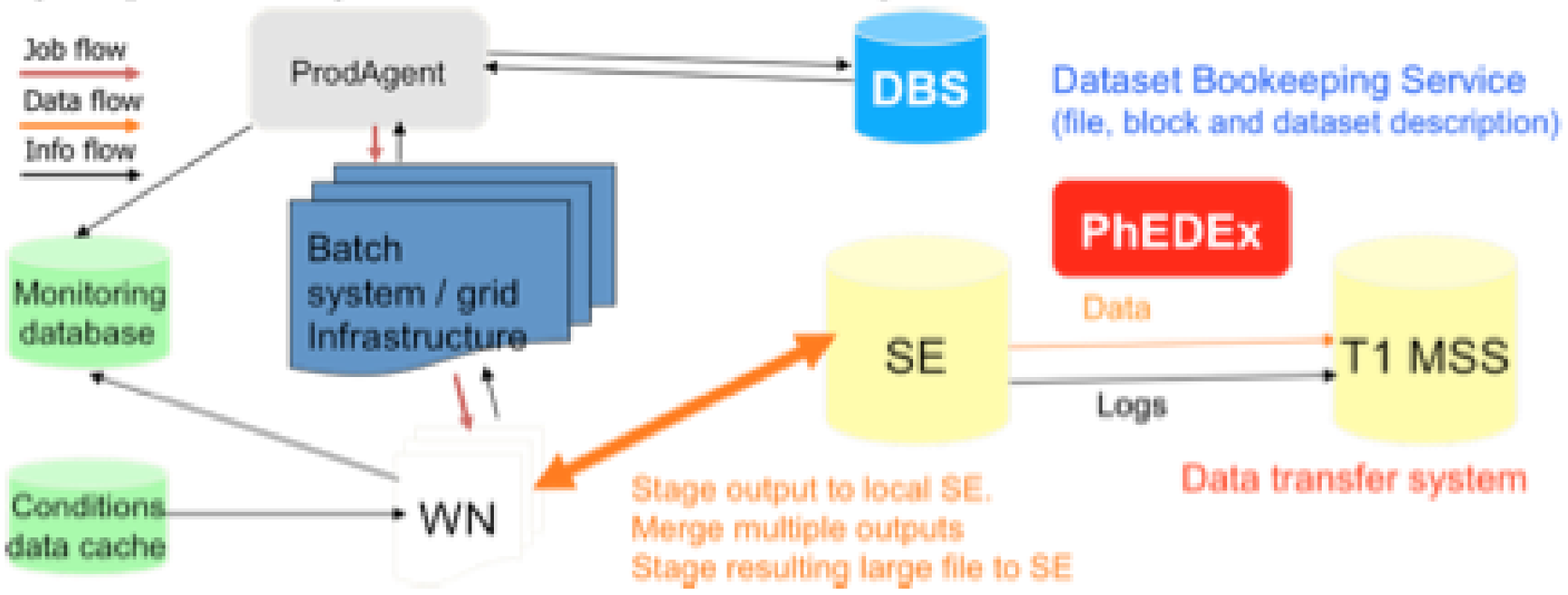

- Standard HEP computing model (i.e. MONARC).
- Large amount of organised computation and data transfer.
  - 23 PB of tape and 4000 cpu's at T0.
  - 45 PB of tape with 22 PB of disk and 14000 cpu's at T1's.
  - 25 PB of disk and 30000 cpu's at T2
- Outbound data rate from CERN ~ 1GB/s.

# CMS computing model



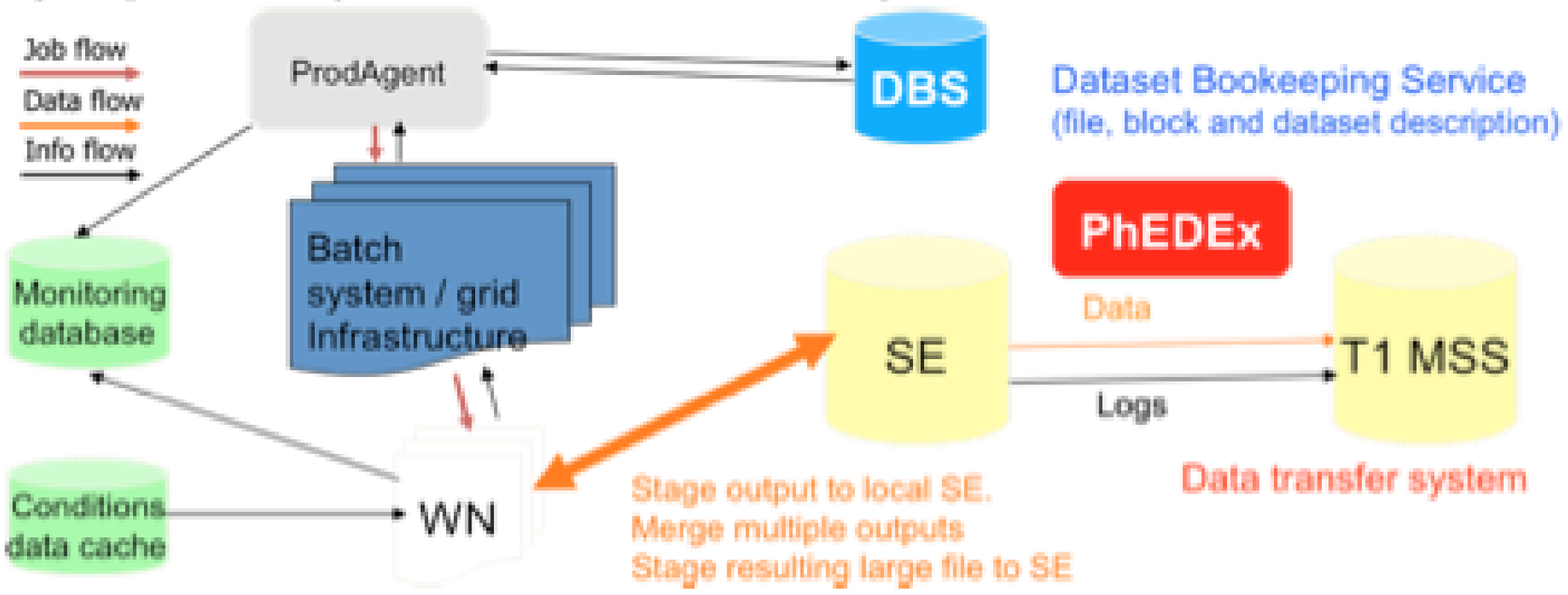

- O(100,000) concurrently running compute jobs.
  - Composed of organized and chaotic (user) activities.
- Tier0 not discussed in this talk.
  - Port to new system currently under development.
  - See “CMS Tier-0: Preparing for the future” (poster 271).
- Extension to this model discussed in “Evolution of the Distributed Computing Model of the CMS experiment at the LHC” (poster 201).

# Workload Management (old)



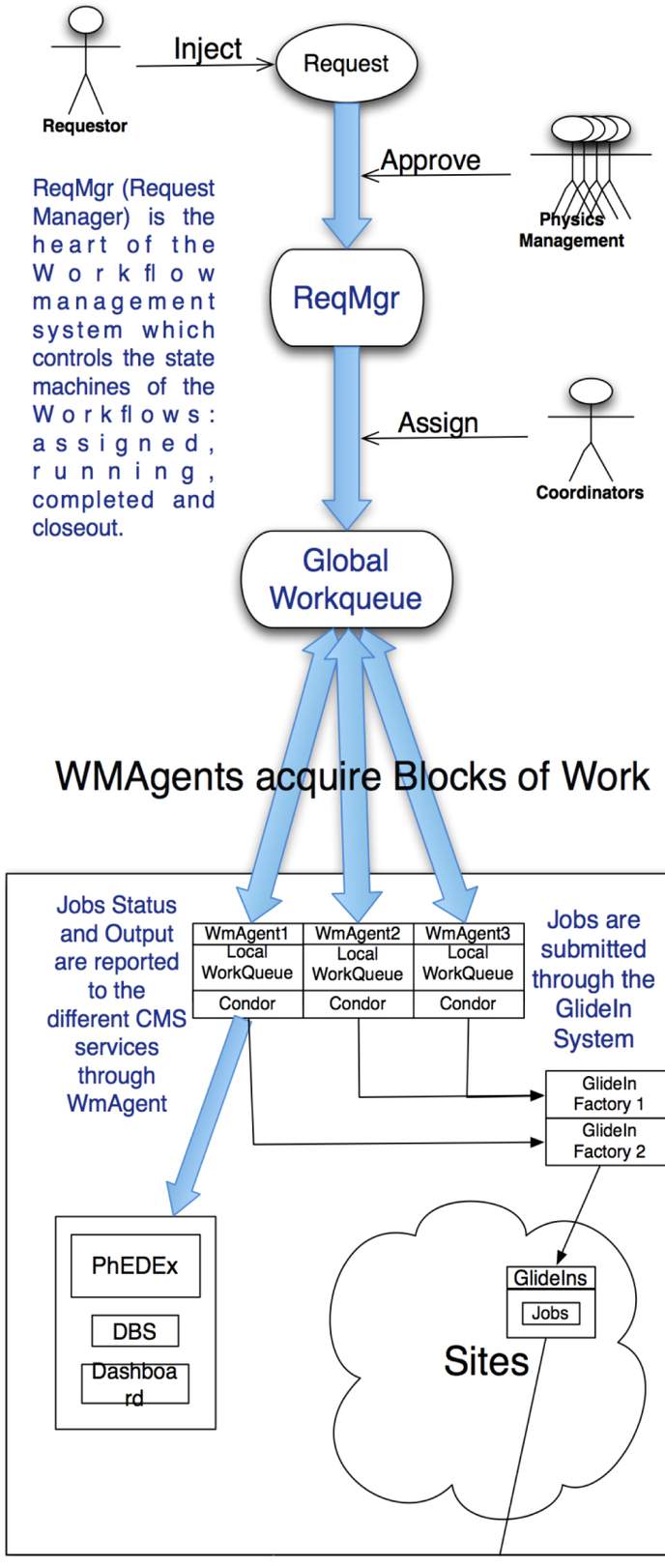

- Many independent agents
- Each responsible for a portion of CMS's needs i.e. T0, a T1, group of T2's.
- Each fed work by an operator manually on the command line.
- No system for handling requests - communicated via email.
- Tough to share work between agents/resources.
- Manually craft workflows to avoid overlaps.

# Workload Management (old)




- No workflow repository or request bookkeeping.
- Agent scalability issues:
- Manpower intensive: feed workflow, monitor output, check for errors etc.
- Limit on number of jobs an instance can handle - speed of submission / tracking.
- Designed for producing simulation data rather than processing real data (loss of few % of events no longer acceptable).
- Analysis (users) used different system.
- Very little shared code / experience.

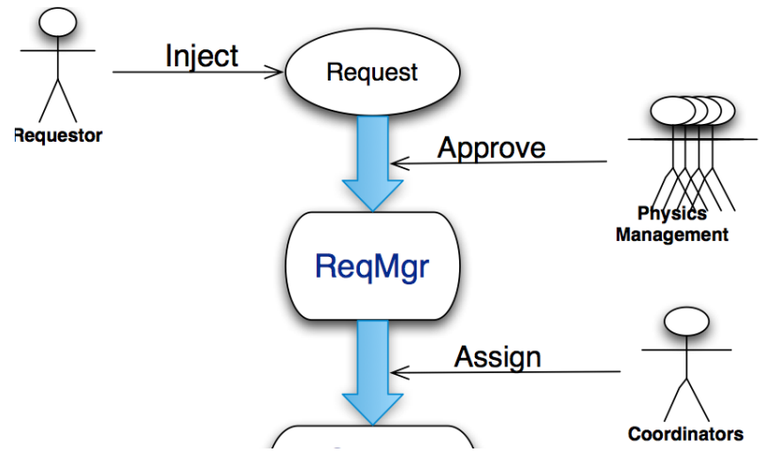
# New WM system




- Consolidate analysis and organized activity
  - Same components but different instances
    - Prevent interference
- Single entry point for requests.
  - Permanently recorded - reproducible
  - Requester can view status.
- Prioritization between requests.
- Approval chain.
- Work distributed automatically & optimally to resources.
- Reduced manpower needs.
- Adapt to new features / requirements:
  - Pilot jobs.
  - Multi-core processes.
- Some use cases require all events to be processed.
  - Cope with intermittent problems.

# ReqMgr

- Web site + REST interface to store & manage requests (workflows).
- Separate out request life cycle stages:
  - Requestor: Define work to be carried out.
  - Management: Approve & prioritize
  - Coordinators: Assign work to (groups of) resources.
- Based on CouchDB (NoSQL) and MySQL.
  - Workflow and CMSSW config in CouchDB.
  - User roles etc. in MySQL
  - For more details see “Life in extra dimensions of database world or penetration of NoSQL in HEP community” (poster 184)
- User & group based prioritization
- Input parameters validation.
  - Large source of errors previously.
- Record of what was run (including progress metrics).
- Classify workflow characteristics
  - “Team” - designates what team of WMAgents and operators will manage it.
  - Memory usage etc.



# ReqMgr II

## Assign Request

**Assign request to teams**

- testbed-dataops
- processing
- relval
- mc
- testbed-production
- testbed-processing
- testbed-dmwm
- integration
- analysis
- t1
- mc_highprio
- production
- dmwm
- testbed-integration
- testbed-analysis
- dataops
- t1_highprio
- testbed-relval

Assign request

**Site White/Black Lists**

Site Whitelist: T1_CH_CERN, T1_DE_KIT, T1_ES_PIC, T1_FR_CCIN2P3, T1_IT_CNAF, T1_TW_ASGC, T1_UK_RAL, T1_UK_RAL_Disk, T1_US_FNAL, T2_AT_Vienna

Site Blacklist: T1_CH_CERN, T1_DE_KIT, T1_ES_PIC, T1_FR_CCIN2P3, T1_IT_CNAF, T1_TW_ASGC, T1_UK_RAL, T1_UK_RAL_Disk, T1_US_FNAL, T2_AT_Vienna

**LFN Bases**

Merged LFN Base: /store/data
Unmerged LFN Base: /store/unmerged

**Other Options**

Min Merge Size 2147483648 Max Merge Size 4294967296 Max Merge Events 50000
Memory limits: RSS 4294967296 VSS 4294967296
Acquisition Era:
ProcessingVersion
Dashboard Activity: reprocessing

Actions: Assign Submit

Select All Select None

| Select | Name | Type | Priority | Input | Splitting and Timeouts |
|---|---|---|---|---|---|
| | spinoso_HIG-Summer12-00739_rq1170_b53_51_LHE_120410_140208_3276 | MonteCarloFromGEN | (1u, 1g) 94998 | /VBF_HToZZTo4L_M-130_8TeV-powheg-pythia6/Summer12-START50_V13-v1/GEN | parameters |
| | etorassa_EWK-Summer12_DR52X-00016_T1_US_FNAL_MSS_batch15_v1__120426_180909_484 | ReDigi | (1u, 1g) 0 | /DYToEE_M_20_TuneZ2star_8TeV_pythia6/Summer12-START50_V13-v2/GEN-SIM | parameters |
| | casarsa_SUS-Summer12-00001_49_v2__120327_235618 | MonteCarloFromGEN | (1u, 1g) 89998 | /ZJetsToNuNu_50_HT_100_TuneZ2Star_8TeV_madgraph/Summer12-START50_V13-v2/GEN | parameters |
| | spinoso_EXO-Summer12-00318_rq986_11_T1_IT_CNAF_LHE_120404_154619_6317 | MonteCarloFromGEN | (1u, 1g) 39998 | /NeutralinoToLLL_lambda123_Msquark-1500_Mgluino-2200_TuneZ2star_8TeV-pythia6/Summer12-START50_V13-v1/GEN | parameters |

ReReco | Monte Carlo | RelVal MC | StoreResults | DataProcessing | Resubmission | ReDigi | MonteCarlo FromGEN

Optional Request ID String:
Optional Prep ID String:
User: swakef Group: DMWM
Request Priority: 1
Software Release: CMSSW_5_0_0 slc5_amd64_gcc434
Include Parents: False
Campaign:
Configuration
Scenario pp
Reco ConfigCache ID

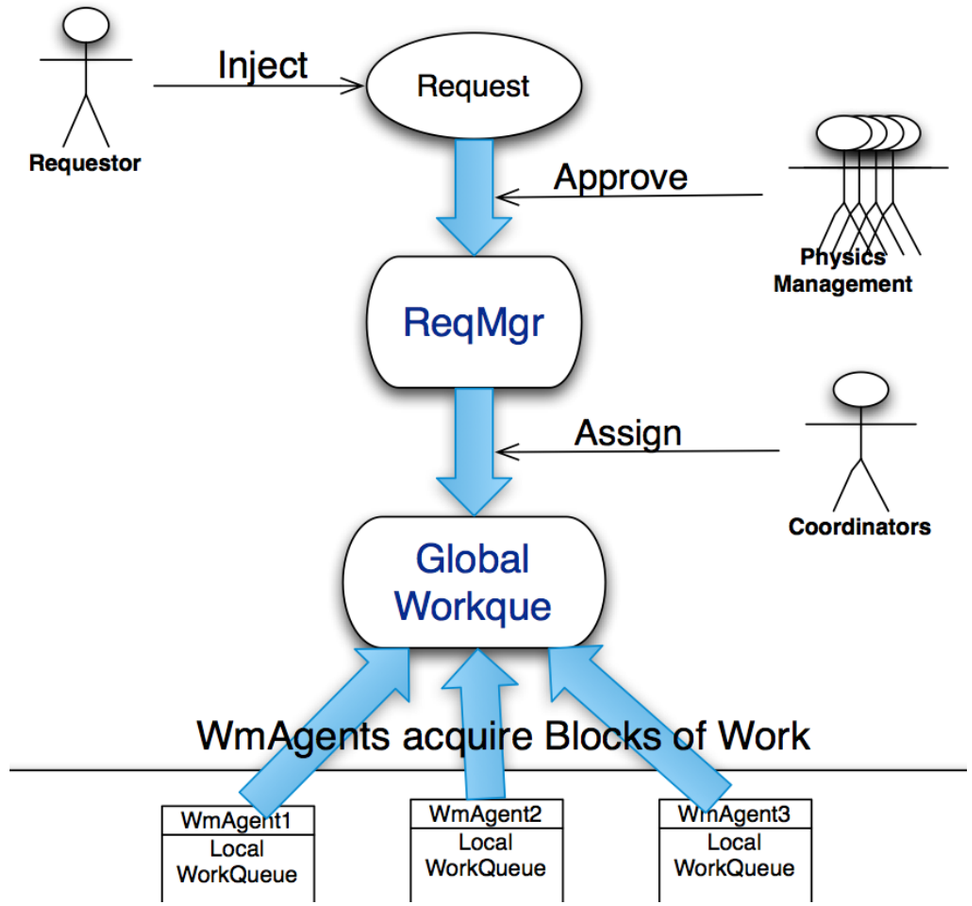
Define work to be done.

Global Tag: Input Dataset:
DBS: http://cmsdbsprod.cern.ch/cms_dbs_prod_global/servlet/DBSServlet
Add Skim
Skim Name: skim1
Skim Input 1:
Skim ConfigCache ID 1:

Run Whitelist (separated by commas):
Run Blacklist (separated by commas):
Block Whitelist (separated by commas):
Block Blocklist (separated by commas):

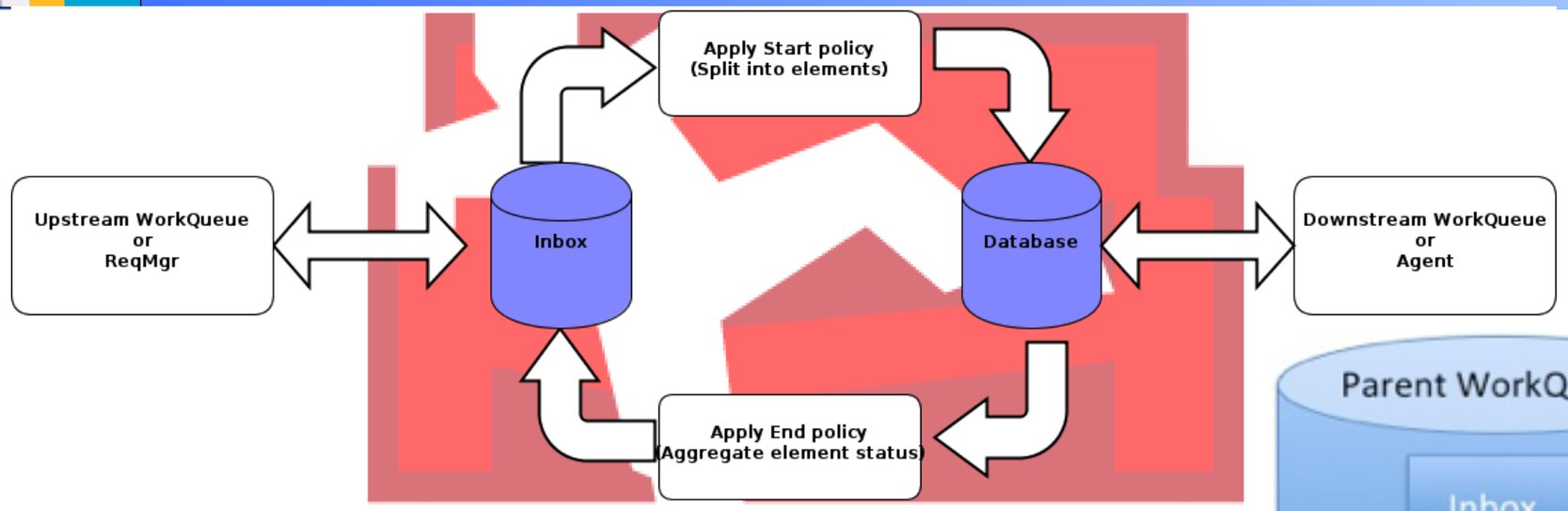
Submit

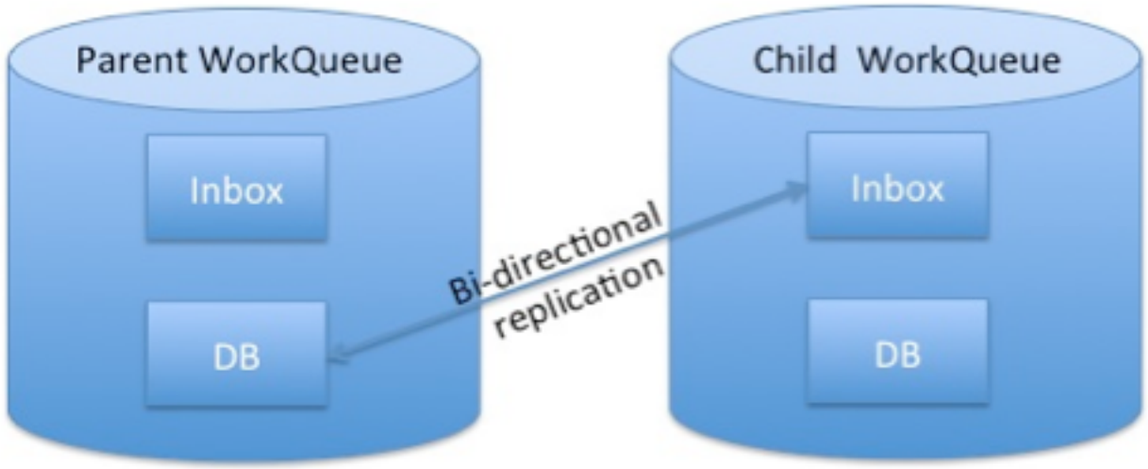
# WorkQueue




- Task queue
- Workflows split into blocks (elements) of work.
- Multi-level hierarchy possible.
  - Currently deployed with global server and local queue next to WMAgents.
- Work obtained by best placed downstream agent or queue.
- Local queue acts as interface to WMAgent and buffers work locally.
- Work released based on priority.
- ReqMgr updated with workflow status (% complete etc.).
- Javascript map/reduce views used widely.

# WorkQueue II



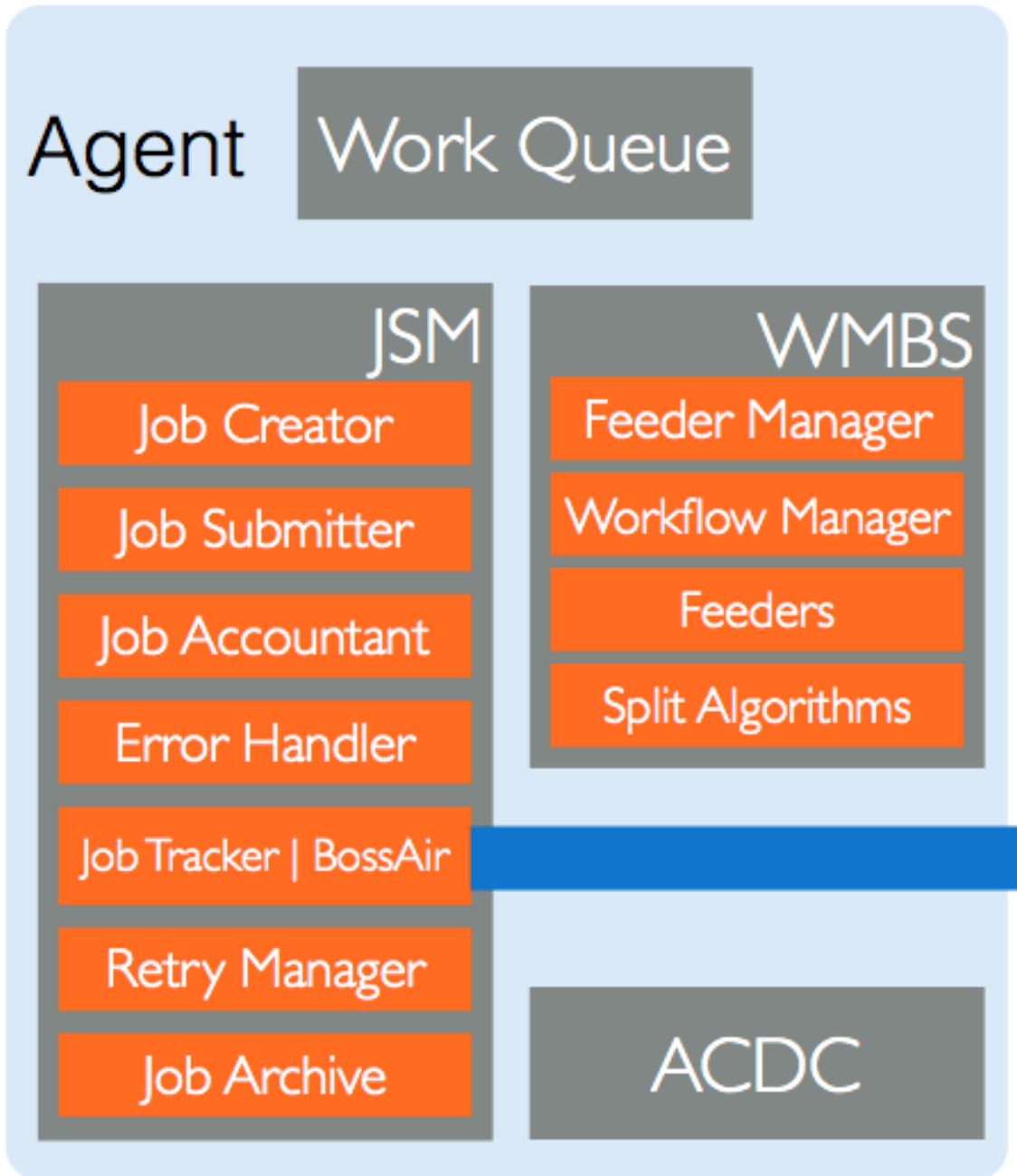




- WorkQueue internal layout:
  - Inbox - contains full request (global queue)
  - Database - contains elements to be acquired by child queue.
- Workflow expanded into coarse units of work (elements)
  - Driven by policy specified in workflow.
  - Advanced features possible such as releasing a small validation sample.
- Elements' aggregated to obtain relevant info (i.e. status, % complete)
  - propagate up to parent queue or ReqMgr.
  - Advanced features include canceling a workflow if a high failure rate encountered.
- CouchDB bi-directional replication keeps agent and WorkQueue's synchronized
- More details in "The WorkQueue project - a task queue for the CMS workload management system" (Poster 227)

# Agent architecture

- ~10 agents each dedicated to a different task and managed by different people.
- Local WorkQueue next to agent
  - Monitor available resources & pull work from global WorkQueue.
  - Inject work into WMBS
- JobCreator looks into WMBS and generates needed jobs.
- JobSubmitter submits jobs.
- Failed jobs logged for bookkeeping and can be used as starting point for rerunning failures.

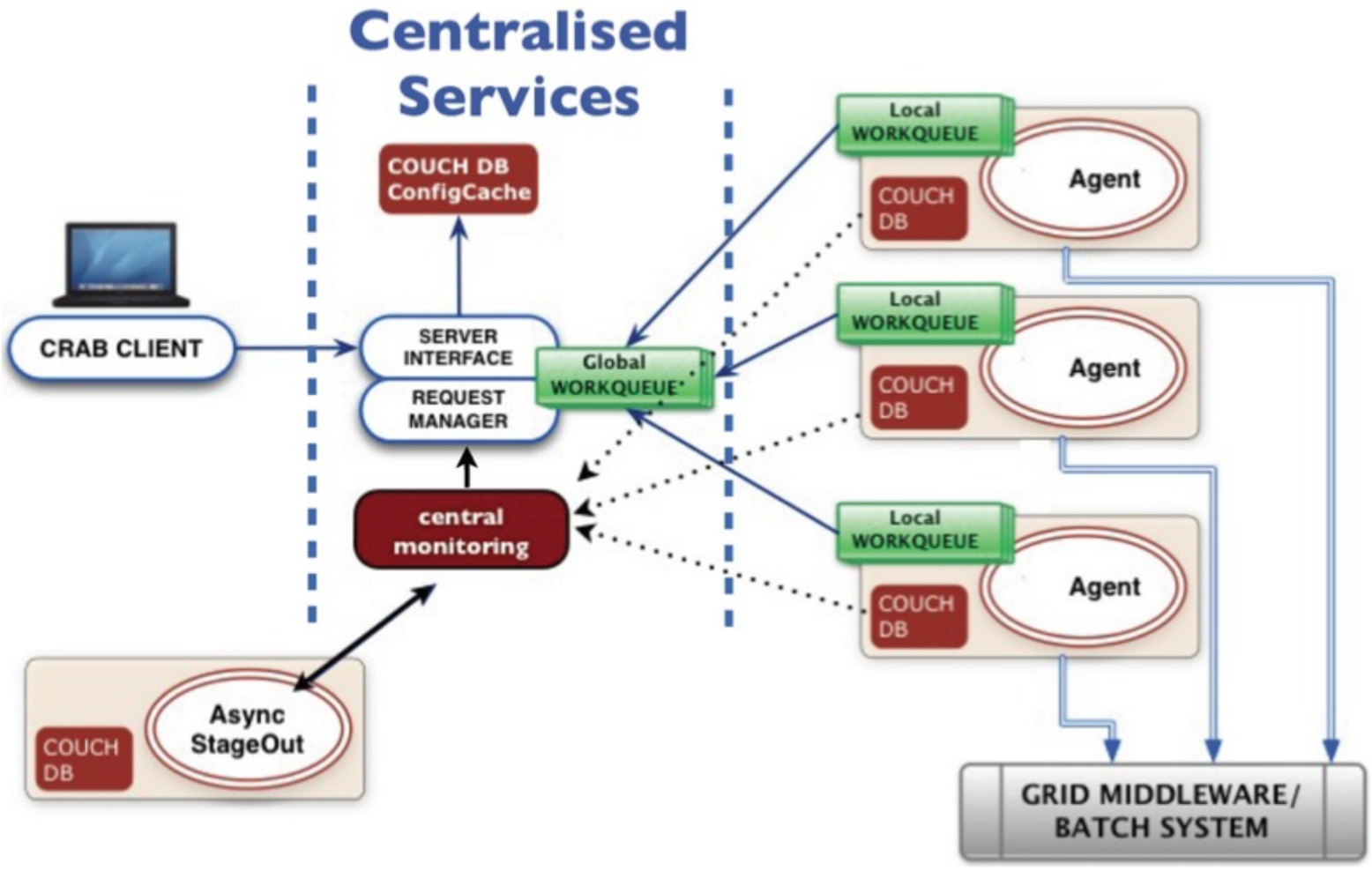
# Agent architecture II




- Workload Management Bookkeeping System (WMBS) defines workflow to data mapping.
- Job State Machine (JSM) components generate, create, submit & track jobs based on WMBS information.
  - ErrorHandler applies policies to failed jobs.
  - After processing summary information / plots generated and uploaded for permanent storage.
- ACDC is a lightweight data collection.
  - Useful for tracking (and resubmitting) failed jobs.

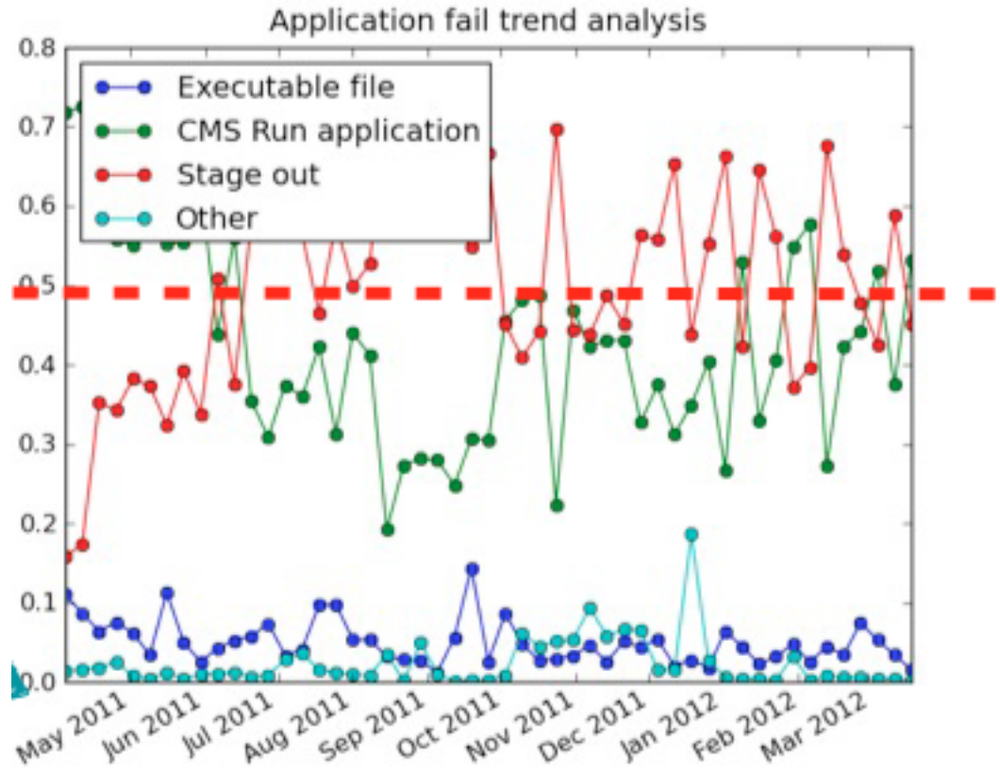
# Analysis

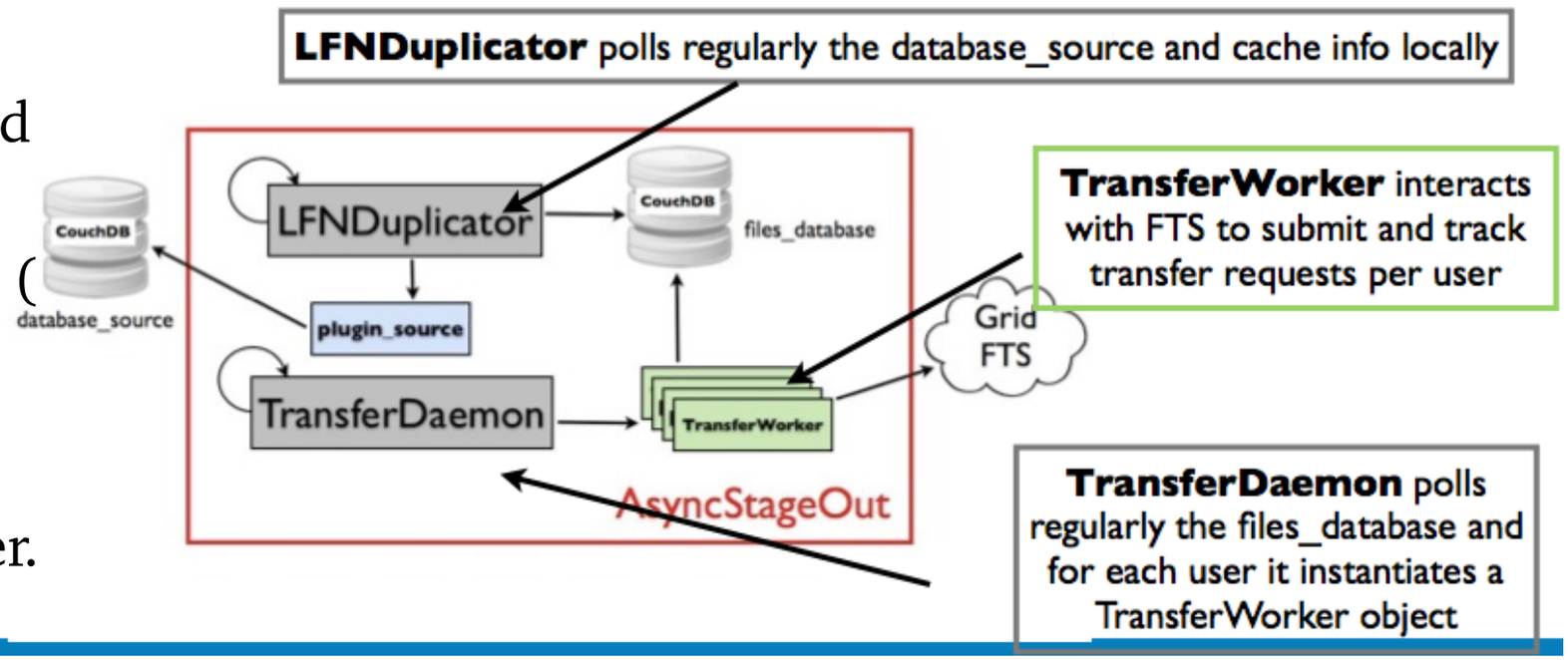
- ~ Production architecture
  - Different instances to avoid interference
  - Custom components
    - Web server holds user files / configs
    - Data transfer (AsyncStageOut)
    - Currently under testing before rollout.
- See “CRAB3: Establishing a new generation of services for distributed analysis at CMS” poster.



# Analysis II

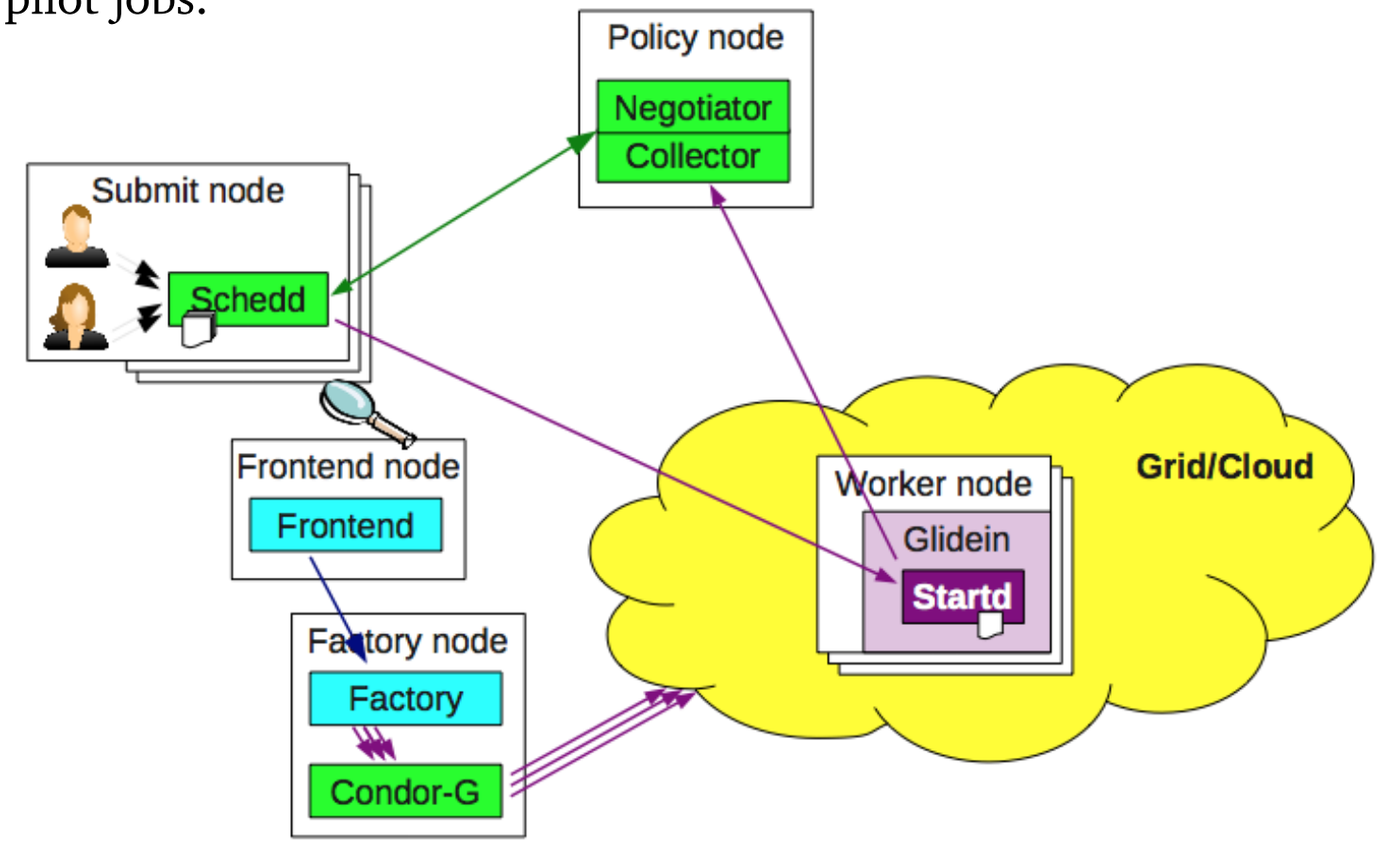
- Current practice is to save job output files to users "home" site.
  - Analysis files not handled by production system
- Generally involves a remote (WAN) copy.
  - Can be large distances (trans-Atlantic)
- Large source of error and inefficiency.
- Move to AsyncStageOut
  - File saved to local site storage.
  - File copied asynchronously to desired output location.
  - Uses gLite file transfer system (FTS) (http://www.eu-emi.eu/)
- See "A gLite FTS based solution for managing user output in CMS" poster.



# Pilot jobs

- Unlike other LHC experiments CMS uses the gLite WMS heavily.
  - Standalone / fully configured jobs.
  - Sent directly to site - resubmit on failure until success.
- Alternative is to send “pilot” jobs which contact a server to obtain a full job.
  - Provides better resilience to failures
  - Allows experiment to exercise finer scheduling control.
- New system includes support for gLite WMS but usage to be phased out
- Transition to only pilot jobs.



# Logging and bookkeeping

- 2 types of logging information archived:
  - Job logs: zipped and archived to tape system
    - Only looked at to track down problems.
  - Summary information for workflows.
    - Track key metrics (and plot) over time e.g. memory usage
    - Useful for monitoring individual workflows and for tracking general trends over time.

**pkonst_DBS3_ReDigiTest_01_120426_182935_3226 Summary**

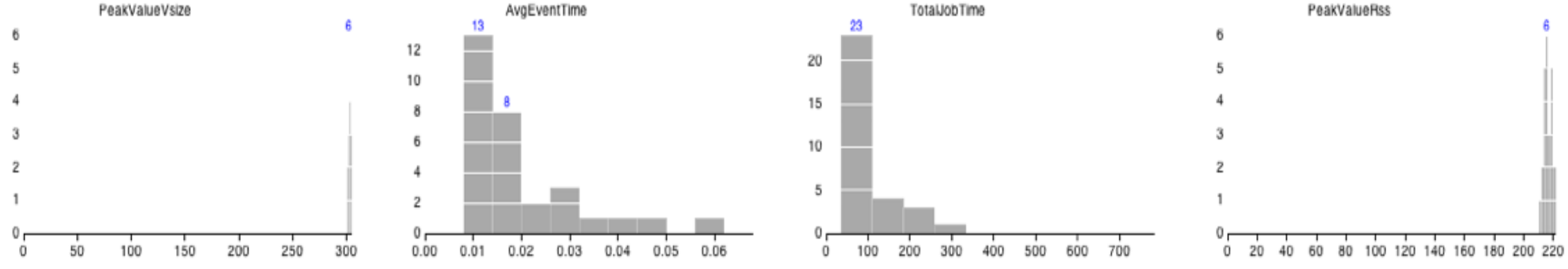
**Output:**

- **/W1Jet_TuneZ2_7TeV-madgraph-tauola/Backfill-MyFilter-120426/GEN-SIM-RAW**
  - **files**: 47
  - **events**: 190136
  - **size**: 136813893463 (bytes)

Histogram :

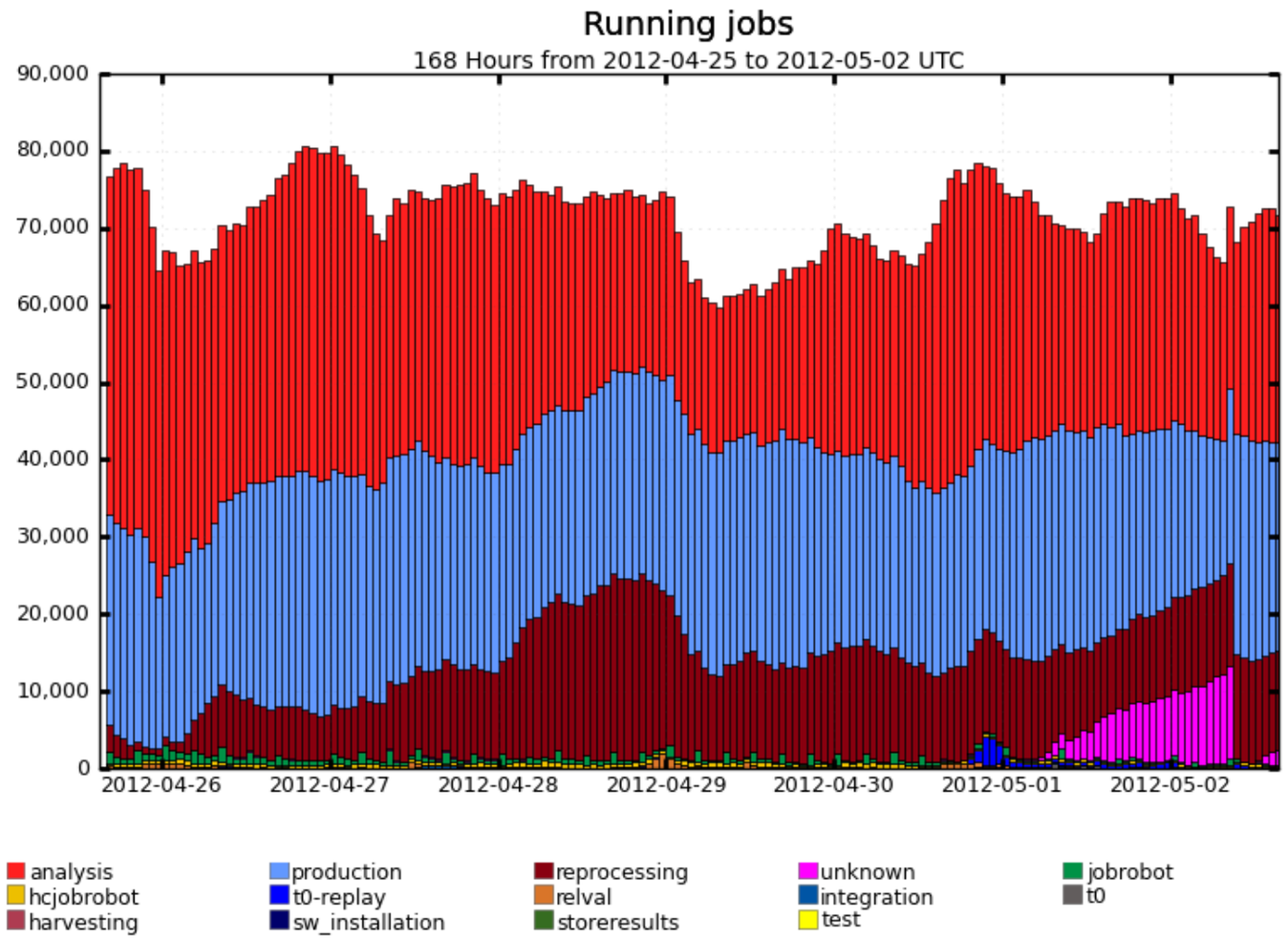
/pkonst_DBS3_ReDigiTest_01_120426_182935_3226/StepOneProc/StepOneProcMergeRAWSIMoutput

cmsRun1



# Usage

- Analysis (in red below) still under old system.
- Production and processing mostly using new system.
  - Some specialized functionality relating to custom generators still to be ported.
- For more details see: “A new era for central processing and production in CMS” (poster 291)

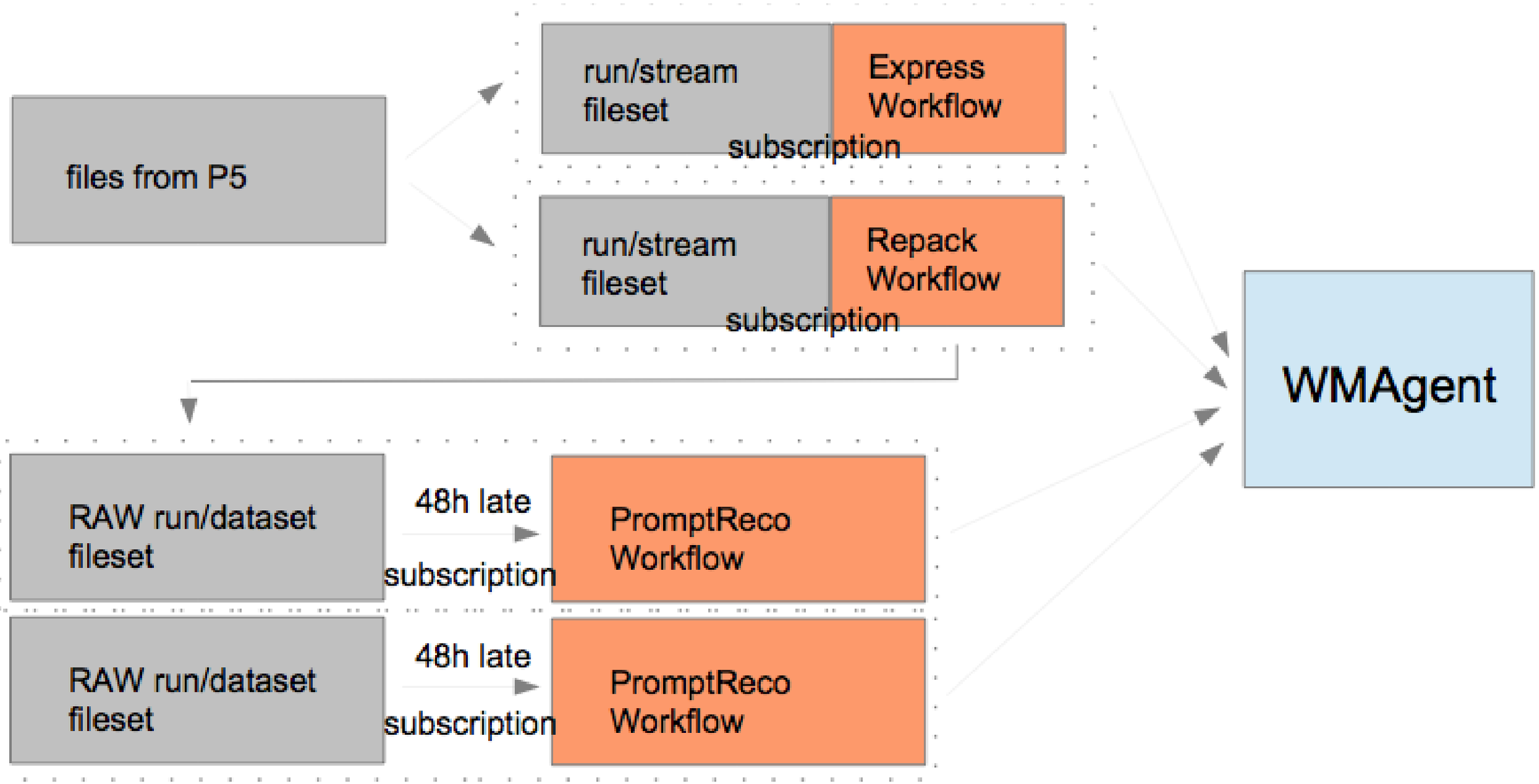


# Summary

- CMS workload management system has been re-written to deal with increasing demands.
- Multiple activity specific implementations have been combined.
- T0 and analysis currently finalizing adoption.
- Hopefully this system will meet CMS's needs for the near future.
- Questions?

# Backups

# Tier 0


files from P5
run/stream fileset
Express Workflow
subscription
run/stream fileset
Repack Workflow
subscription
WMAgent
RAW run/dataset fileset
48h late
subscription
PromptReco Workflow
RAW run/dataset fileset
48h late
subscription
PromptReco Workflow