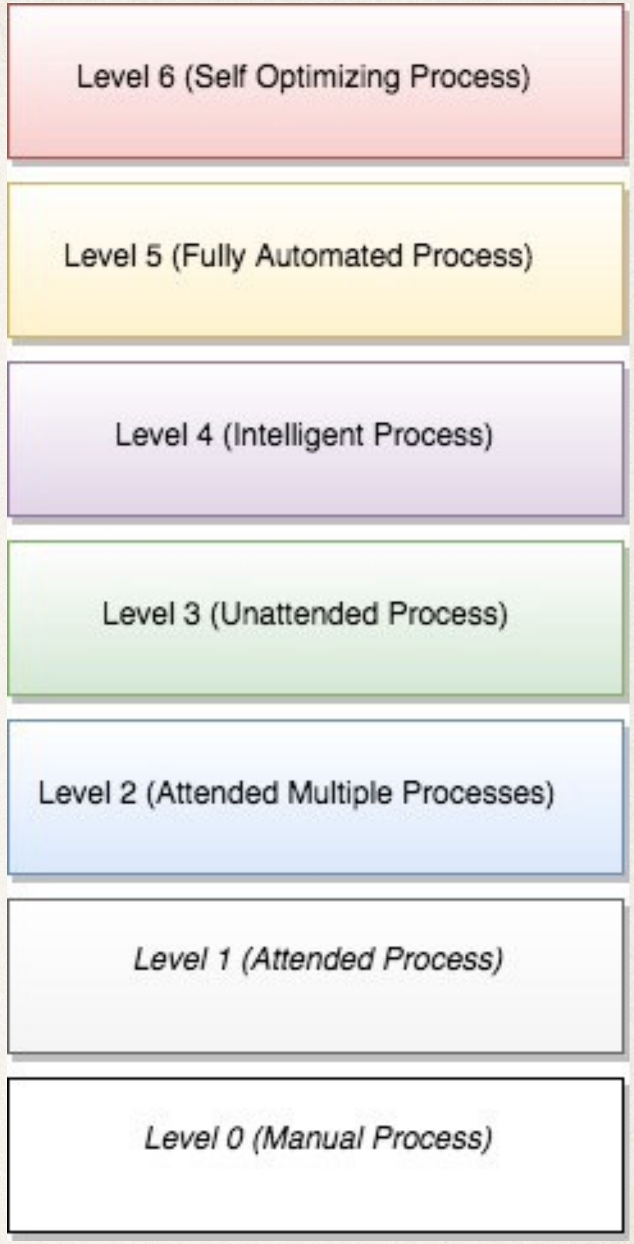


[SAE J3016]

CMS Analytics

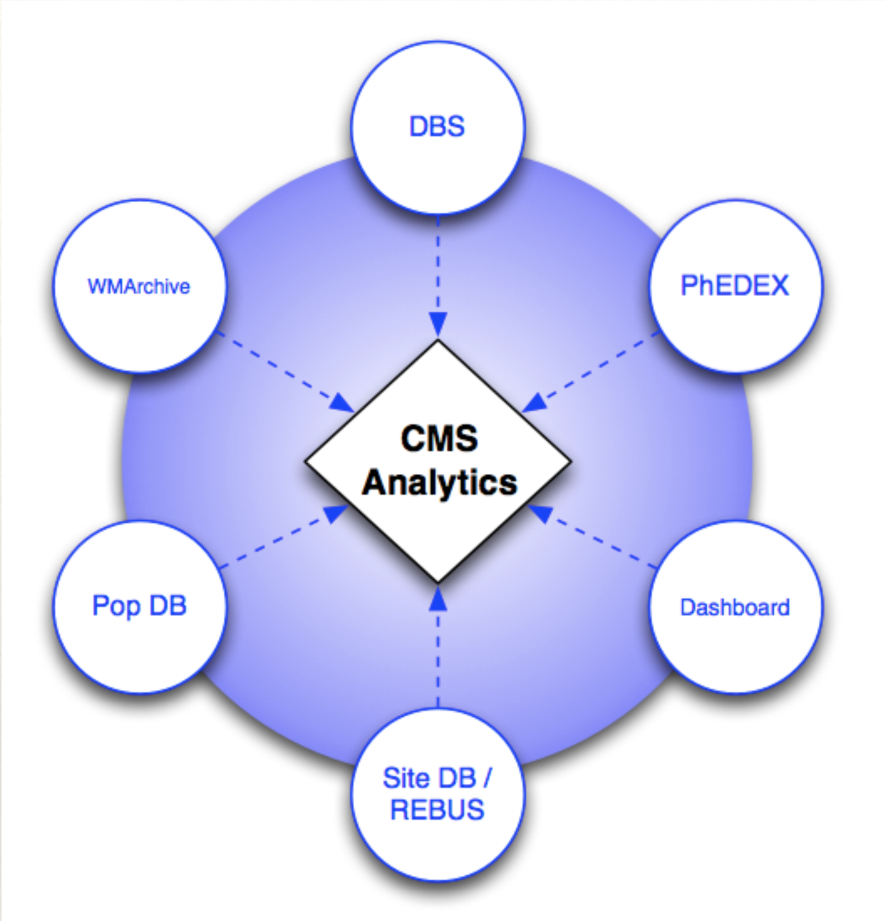
Daniele Bonacorsi, Valentin Kuznetsov, Federica Legger

HOW 2019, JLAB - March 2019

The start

- CMS built a computing system that **worked** in LHC Run-1/2.
  - At which depth do we fully "**understand**" it? Can we perform precise modelling of specific workflows / site behaviours / systems performances? Can we use this modelling to make predictions (e.g. population vs pollution of Tier disks; TierX - Tier-Y data transfer patterns; ..)
- Computing operations (meta-)data is all archived. **Only recently started to be accessed**.
  - e.g. transfers, job submissions, site performances, infrastructure and services behaviours, storage accesses, ..
- For long, we monitored to debug in near-time, not to analyse and learn from the past to design and build what's next. A complementary "data scientist" approach towards an **adaptive modelling of CMS workflows** was felt as necessary to **extract actionable insight**.
- Note that <u>most ideas came out of a first CMS R&D workshop in Bologna, back in June 2014</u>

CMS **Structured** data



(these plots are almost 5 yrs old today..)

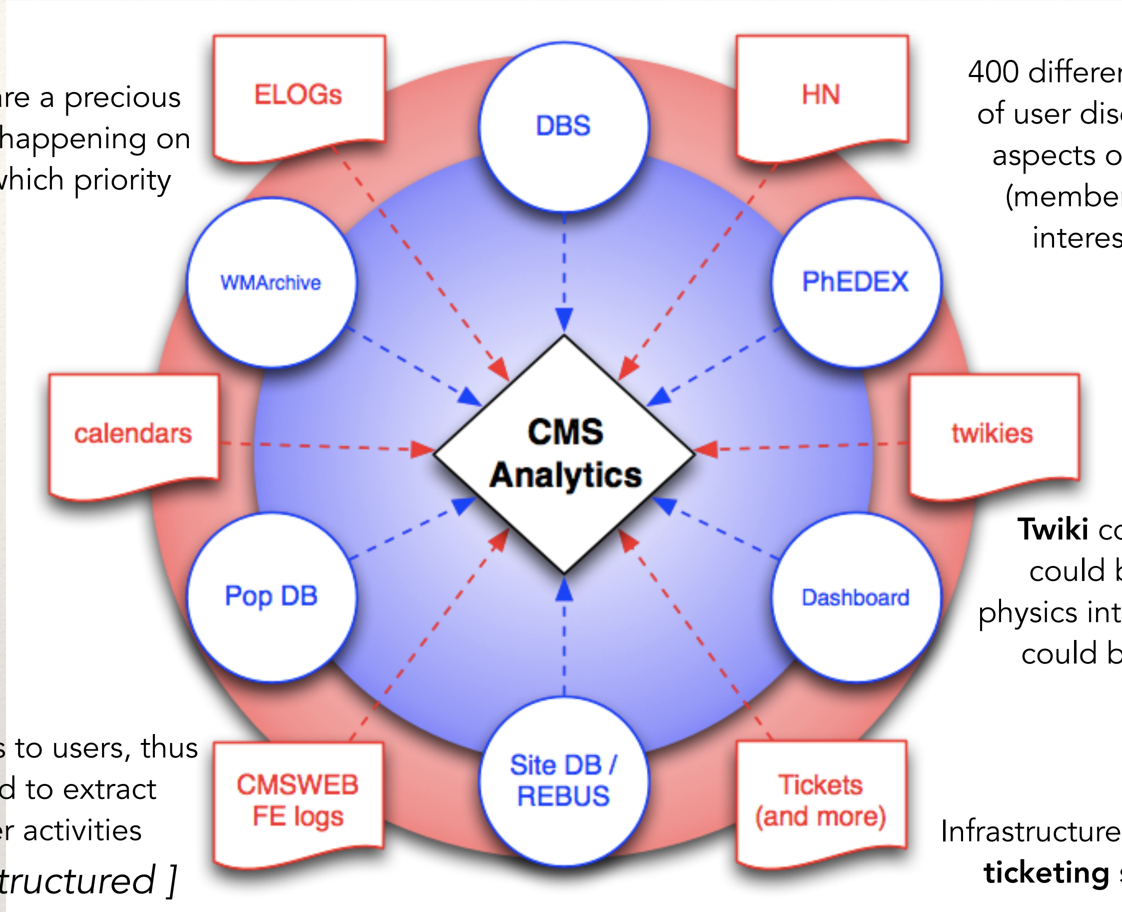
Structured info on a variety of CMS Computing activities are stored across multiple data services

- all info available via CMS data service APIs

CMS **Structured** and **Unstructured** data

Activity-based **ELOGs** are a precious source of info on what's happening on which systems and at which priority

400 different **HyperNews** fora, several yrs of user discussions, "social data-mining" aspects of collaboration-level research (membership changes study, physics interests evolution over time, …)



CMS events **calendar**, activity planning docs, list of major conferences and workshops, … could identify cycles within different physics communities

Twiki content as a knowledge graph that could be mapped to user activities and physics interests, and their evolution over time could be studied with appropriate tools.

It serves all data sources to users, thus its logs may be mined to extract valuable info on user activities

[Warning: semi-structured]

Infrastructure issues reporting/tracking, **ticketing** systems (JIRA, GGUS)..

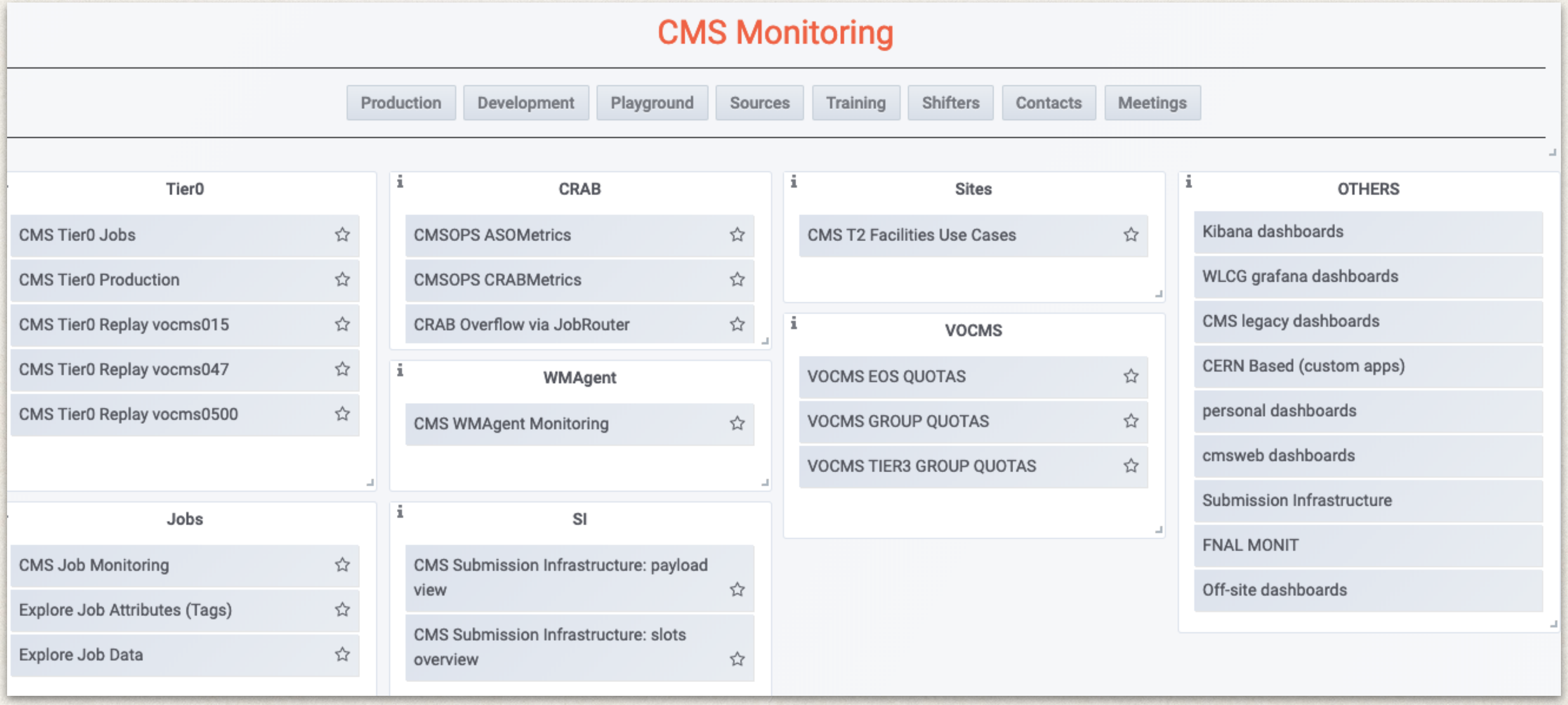
Plenty of **unstructured** information in the CMS Computing (meta-)data ecosystem

- potentially very rich and sensitive predictors of user activities and future needs
- hard to process; manpower shortage; needs careful cost vs gain evaluation

We started by focussing mostly on **structured**

CMS and CERN MONIT

- CMS recently started migration of old dashboards to CERN MONIT infrastructure
- In about 6 months, from scratch, plenty of dashboards were developed



CMS Monitoring

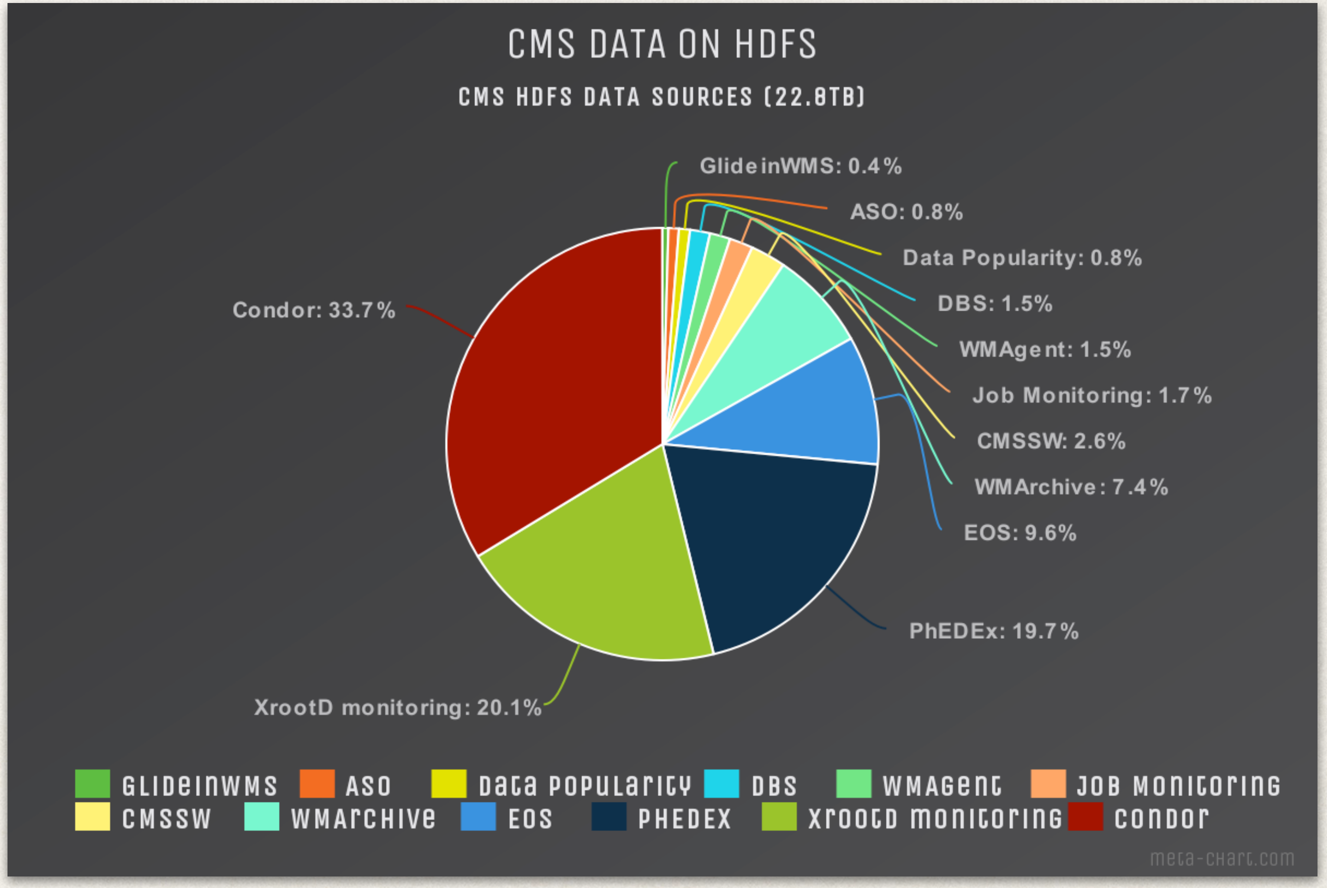
- We successfully adopted "modern" (as the time defines them) solutions
  - Short-term data storage: **Elastic Search**
  - Long-term data storage: **HDFS**, **InfluxDB**
  - Visualization: **Kibana**, **Grafana**
  - Data scraping: **Prometheus**, **Logstash**
  - Analytics: **HDFS+Spark** (PySpark)
  - Alerts: **ES**, **Prometheus**, **Grafana** to **SNOW** tickets, Email, Messengers
- We train our users to use new tools and simplify users workflow
  - e.g. CMSSpark encapsulates all CMS HDFS data sources and provide common framework to run Spark jobs

Training
↓
Data Sources
↓
Playground
↓
Development
↓
Production

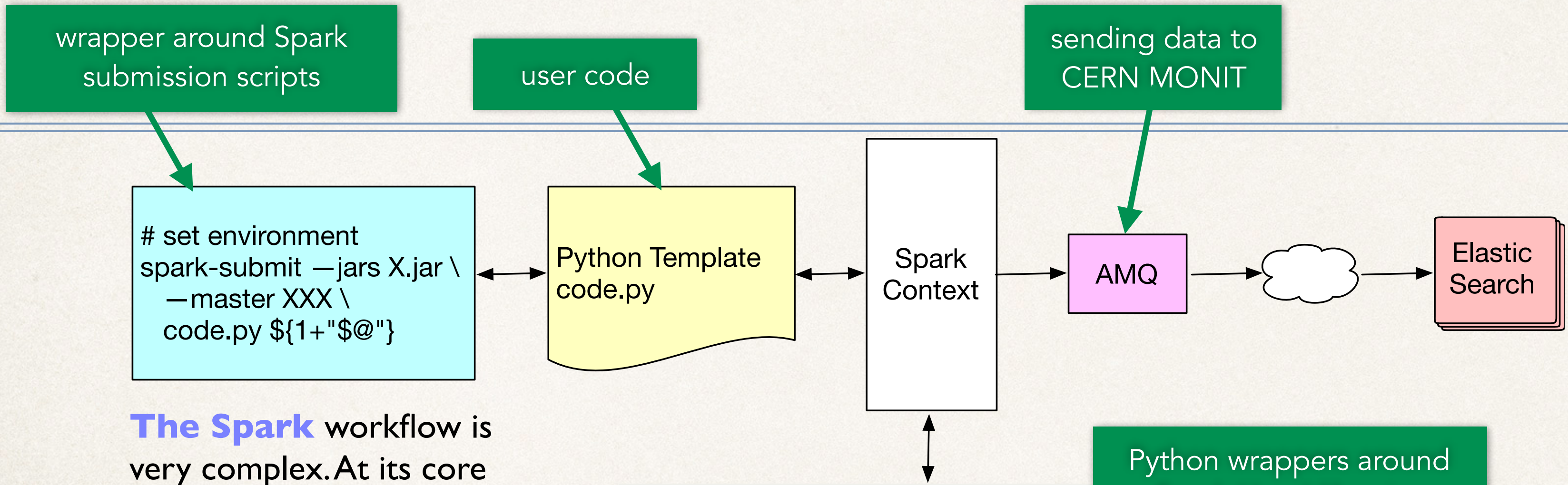
CMS analytics

J.Phys.Conf.Ser. 898 (2017) 092030

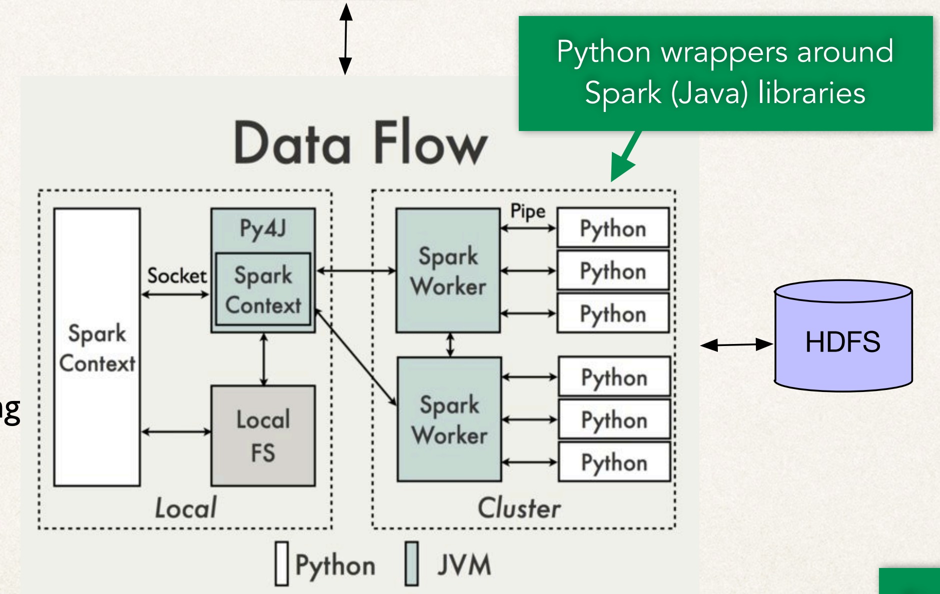
- In recent years CMS started to heavily use data source on HDFS
- So far we have **dozen of data-streams on HDFS** and run **30+ different analytics workflows**
- Majority of user jobs are automated via CMSSpark framework



CMSSpark pipeline



The Spark workflow is very complex. At its core are Java based libraries wrapped around in APIs in various programming languages



We simplify user access to the Spark platform by wrapping user code in framework which takes care of setting up cluster environment, data access and jobs.

Simplified user experience

run_spark	dbs_agg.py	–yarn –date 2018 –fout hfs:///cms/users/vk
shell wrapper	user workflow	user options

Data popularity (and the scrutiny plot)

J.Phys.Conf.Ser. 762 (2016) 012048
J.Phys.Conf.Ser. 664 (2015) 032003
J.Grid Comput. 16 (2018) no.2, 211-228

- Collected yearly stats for DBS datasets, PhEDEx replica sets and HTCondor logs
  - O(week) to process a year of data on Spark in the CERN ‘analytix' cluster
  - O(hour) to merge PhEDEx data-frames
  - O(min) to produce a plot

More: produce site utilisation plots on regular basis using CMSSpark

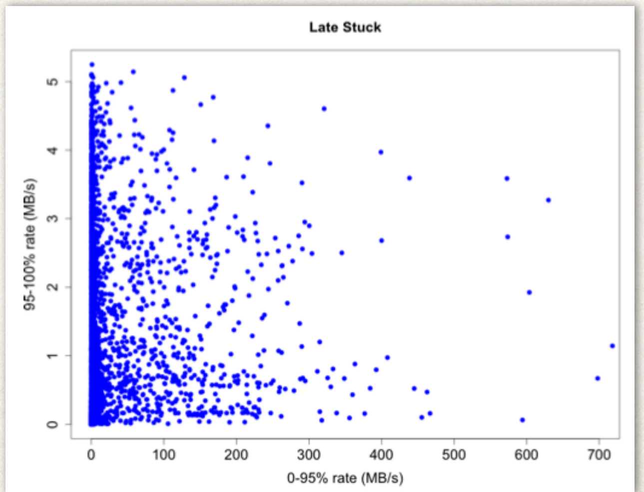
Transfer latencies

J.Phys.Conf.Ser. 664 (2015) 032033
PoS ISGC2017 (2017) 023

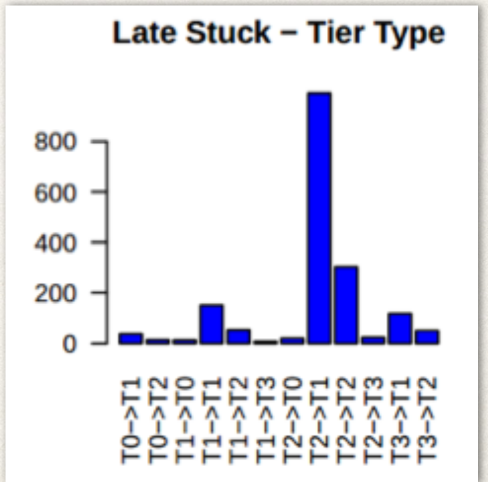
- Classification of transfer latencies (e.g. early/late-stuck), as well as supervised regressions to predict latency figures in Tx-Ty routes at ~80% accuracies
    - predictions to be used in the routing logic of any DM tool.

Example: *late stuck* may occur while transferring large datasets (due to e.g. transient storage issues or corrupted files). Identified cases well in advance may help Ops team to cure, and ultimately automatic procedures to be set up.

95%–100% rate (MB/s)



0-95% rate (MB/s)



Assisted workflows handling

- Automate the job errors handling in Grid/Cloud workflows, i.e. for well defined groups of errors, automatically kill/clone/resubmit/recover workflows as appropriate
    - unsupervised clustering methods before the operator, and learn from the operator choices, or..
    - .. supervised deep learning

One of these tables per **EACH CMS WORKFLOW**

SITES
The color of each site corresponds to its status (white means the site status is not tracked by Dashboard, and it is assumed "good")

	T0_CH_CERN	T1_DE_KIT	T1_ES_PIC	T1_FR_CCIN2P3	T1_IT_CNAF	T1_RU_JINR	T1_UK_RAL	T1_US_FNAL	T2_CH_CERN	T2_CH_CERNBOX	T2_CH_CERN_HLT	T2_DE_DESY	T2_ES_IFCA	T2_FR_GRIF_IRFU	T2_FR_GRIF_LLR	T2_IT_Legnaro	T2_UK_London_Brunel	T2_UK_London_IC	T2_UK_SGrid_RALPP	T2_US_Florida	T2_US_MIT	T2_US_UCSD	T2_US_Wisconsin	T3_US_FNALLPC	null
-1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
84	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	0	2	0	0	0	1	6	0	0	0	8	1	0	0	0	0	0	0	0	0	0	1	0	0	0
92	0	0	1	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0
134	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
139	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
8001	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
50110	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50660	0	0	3	2	4	1	1	1	6	0	63	1	0	1	0	1	0	1	0	0	1	0	1	0	0
50664	0	0	2	7	0	0	2	18	0	0	32	0	0	7	0	0	4	0	2	0	0	0	0	0	0
71304	0	0	0	0	0	0	0	0	0	0	102	0	0	0	0	0	0	0	0	0	0	0	0	0	0
99305	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	96

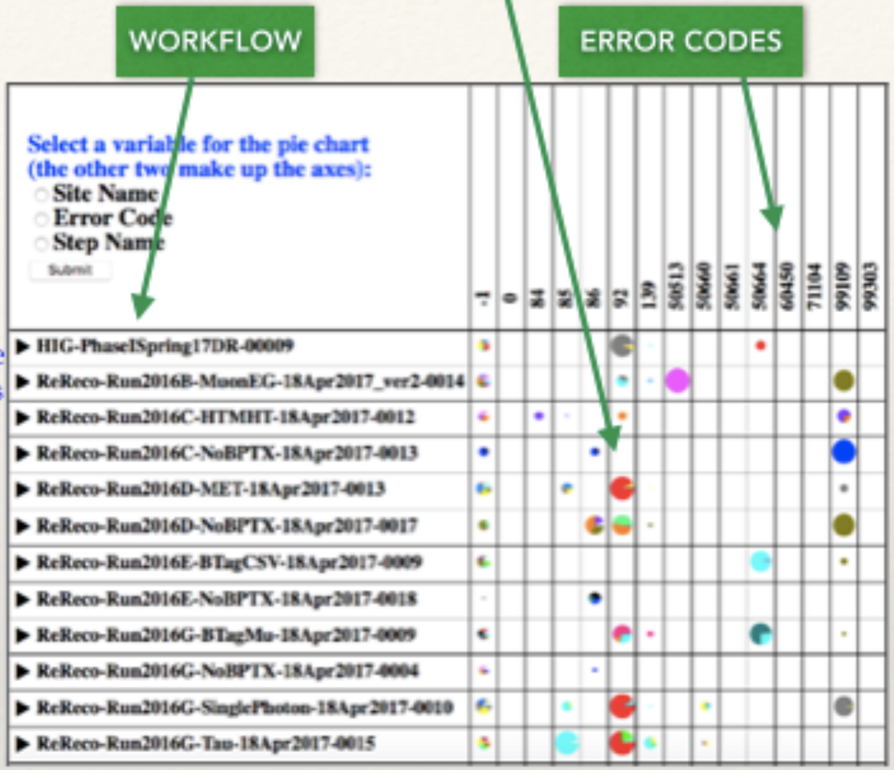
ERROR CODES

A not-obviously similar workflow to the one above, but the same errors at different sites are likely from similar causes

	T0_CH_CERN	T1_US_FNAL	T2_CH_CERN	T2_US_Caltech	T2_US_Florida	T2_US_MIT	T2_US_Purdue	T2_US_UCSD	T2_US_Wisconsin	T3_US_FNALLPC
-1	1	1	1	1	1	1	1	1	1	1
92	0	2	1	0	0	0	0	0	0	0
134	0	0	0	0	8	0	0	0	0	0
139	0	0	0	0	3	0	0	0	0	0
50660	0	28	0	2	1	6	10	4	8	0
50664	0	2	0	1	0	0	0	0	0	0
71304	0	24	0	0	0	0	0	0	0	0
99303	0	3	0	0	0	0	0	0	0	0

Workflows that are similar get clustered together. All features are dumped into a K-means clustering algorithm.

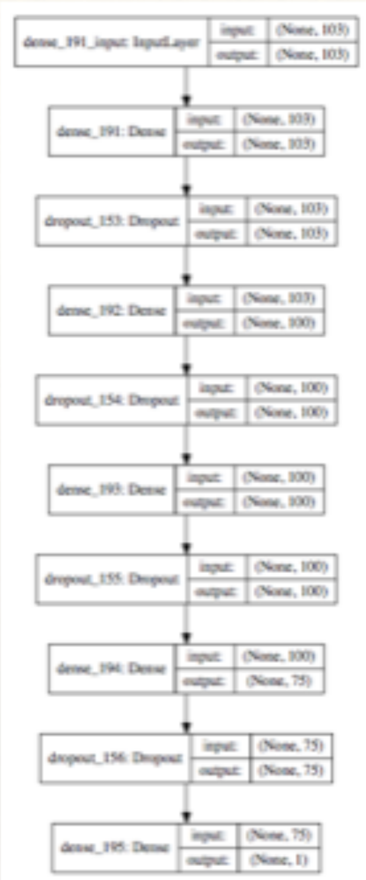
The **size** of the pie charts corresponds to the # errors. Each **color** in the pie chart is a different site.



Prototype of interface for operator **built** and **in use**: capable of clustering workflows with similar errors

Now **recording the actions taken by operators** in a manner that can be trained.

Alternative approach with DNN



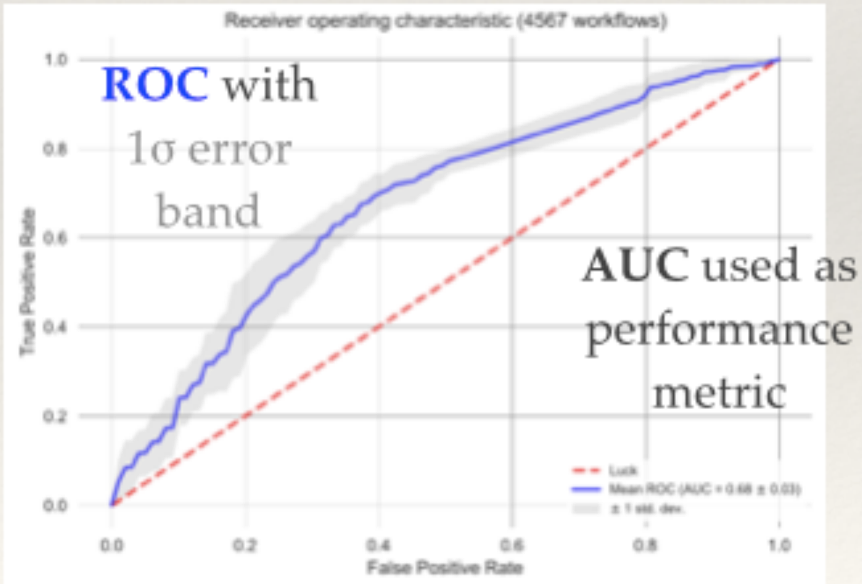
Deep net arch (not optimised yet):
- 5 Layers
- 75 neurons (hidden units) per layer
- RELU as activation function
- Dropout 0.002
- Learning rate 1e-3
- Binary cross-entropy as loss function
- ADAM for GD optimiser

An alternative **supervised** approach with DNN can be adopted

- Goal: predicting *recover* actions versus *resubmit* actions
- using the same "error codes - site status" matrix information as in previous slide (e.g. top left)

Deep network implemented

- *Preliminary*: at first attempt, already (68±3)% accurate (AUC)
- Of course to be further optimised, cured for over-training, etc.: work in progress..



Preemptive maintenance on a WLCG site

[2 talks soon at ISGC 2019]

- Builds on the richness of info in logs of various services and systems, to be exploited towards building human-assisted (ML based) "intelligent" monitoring systems
- e.g. prototyping work at a WLCG Tier-1 (INFN-CNAF)
  - *focus on StoRM and storage services so far (plan to extend at a later stage)*
  - *tough data preparation phase: log investigation, massaging, correlation studies, …*
  - *design and training of first ML models*
    - *(PRELIMINARY) first glances to a demonstrated ability to raise alarms hrs before problems occur (despite very site/system-specific)*
  - *infrastructure-wise, ELK ingestion so far - moving now to fully Spark-based (wip)*

For discussion at JLAB

- CMS active since few years in various areas, and stated in many occasions in the past that:
  - the good: any experiment exploiting WLCG resources would largely benefit from **cross-experiment activities on analytics**
  - the bad: context is large, (meta)data are vast: need to focus on well-formulated problems we do have (investing on **<u>common problems</u>**)
- Time is perhaps mature now to make new steps:
  - Rucio as a natural playground from a cross-experiment exploration on analytics
  - smart caches studies are a great example of a x-experiments work to do
  - more? predictive maintenance at other WLCG sites? a common platform on ML/DL work on LHC computing (meta)data? joined projects that can attract students? ..)
- Discussion.

Society of Automation Engineering (SAE)'s international standard [**SAE J3016**] defining 7 levels of (driving) automation, but an interesting framework to classify automation levels even in other domains.

Human intervention not required, also adapt itself and improves itself autonomously	Level 6 (Self Optimizing Process)	**Fully "adaptive" modelling**
Human intervention not required, unless for designing process optimisations	Level 5 (Fully Automated Process)	
Responsibility to define the end goal(s) and context of automation, then all processes execution and compensating actions are automatic	Level 4 (Intelligent Process)	**OpsIntel** as of this meeting
Notification only in case of exceptional states, actions required depending on severity	Level 3 (Unattended Process)	
Awareness of multiples processes start/stop, no sequencing responsibility, more complex to undo	Level 2 (Attended Multiple Processes)	**Run-2** around here?
Awareness of process start/stop, overview sequencing, ability to undo	*Level 1 (Attended Process)*	**Run-1** around here?
No automation at all	*Level 0 (Manual Process)*	