

# Data Knowledge Base evolution for a stable usage and operation

**V. Kotliar** NRC “Kurchatov institute” - IHEP, RU-142281, Protvino,  
Moscow region, Russia

E-mail: [Viktor.Kotliar@ihep.ru](mailto:Viktor.Kotliar@ihep.ru)

with support from NRC “Kurchatov institute” and from Marina  
Golosova @yandex, Mikhail Borodin @CERN



The Data Knowledge Base (DKB) project is aimed at knowledge acquisition and metadata integration which consists of **DKB python library** and **API service**. Due to its **ETL workflow engine** it allows to build a single system with integrated and pre-processed information behind **distributed metadata infrastructure**. Such system provides fast and efficient response for summary reports, monitoring tasks (aggregation queries), multisystem join queries and others. In this work the recent improvements and a **current status** of the project are shown as well as **its usage** in WLCG for operation metadata analysis.



# Introduction and history

The Data Knowledge Base (DKB) project is aimed at knowledge acquisition and metadata integration

Started at 2016 v1 main purpose

- Integrate and link pieces of information from independent sources (pdf, indico, wiki pages,...)
- Reconstruct connections between research results and data samples
- Provide fast and flexible access to everything people might want to know about some process or object

2018 v2

- Universal tool for multi-source queries
  - Library pyDKB
  - ATLAS dataflow system
    - ETL flow based on scripts and library
    - System to run and check flow
    - Database to store results
    - REST API to access system
    - Frontend UI for users



# Introduction and history

2022

- ATLAS dataflow system
  - Migration to Elastic Search 7.1 central CERN service
  - Frontend UI for users in production

2023 v3

- DKB python library
  - Moving into python environment (Miniconda)
  - Migration to python3
- ATLAS dataflow system
  - New rucio, ES, Oracle python libraries
  - More UI for Atlas metadata analysis for LHC Run3

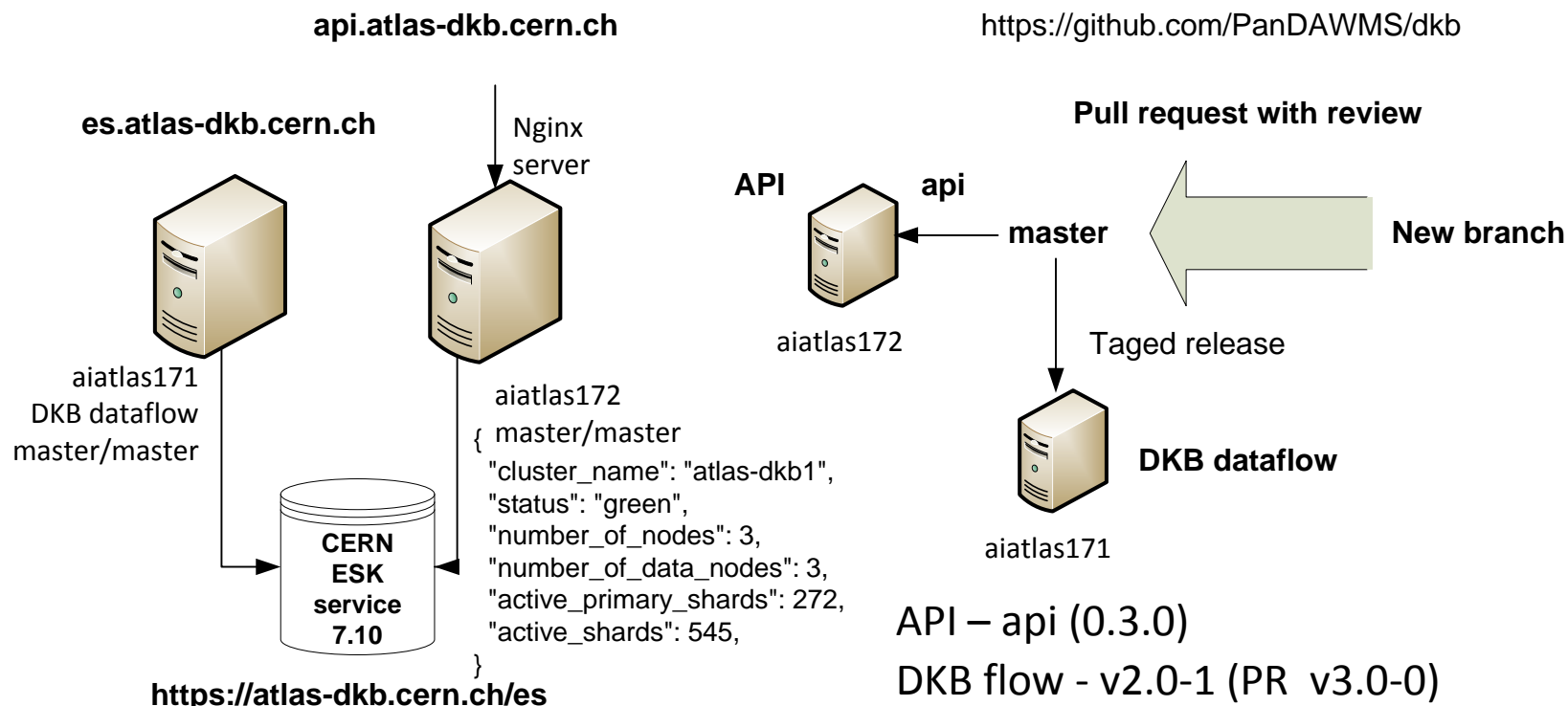


# Current environment

host	OS	where	Alias	Puppet
aiatlas172	CC7-x86_64	cern-geneva-a	api.atlas-dkb.cern.ch	master
aiatlas171	CC7-x86_64	cern-geneva-a	es.atlas-dkb.cern.ch	master
aiatlas170	CC7-x86_64	cern-geneva-a	-	master/reserved
aiatlas173	CC7-x86_64	cern-geneva-a	-	master/reserved
dkb-dev-cc7	CC7-x86_64	cern-geneva-b	-	-

gitlab.cern.ch it-puppet-hostgroup-voatlasdkb  
<https://github.com/PanDAWMS/dkb>

# Current environment





# Current environment: numbers

```
curl -sS -X 'https://atlas-dkb.cern.ch:443/es/_cat/indices?v' |grep -e 'production ' -e docs.count
```

health	status	index	uuid	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
green	open	progress_production	6t3_ubPyTuikfG1Y0FMSRQ	4	1	4596109	67609	805mb	402mb
green	open	tasks_production	J2GW7puVQJ61a7rMUTh6Pw	4	1	9853207	290281	10.3gb	5.1gb

DKB workflow runs every hour by cron

- ATLAS production 350sec (646 tasks 664 datasets)

Consistency check every day

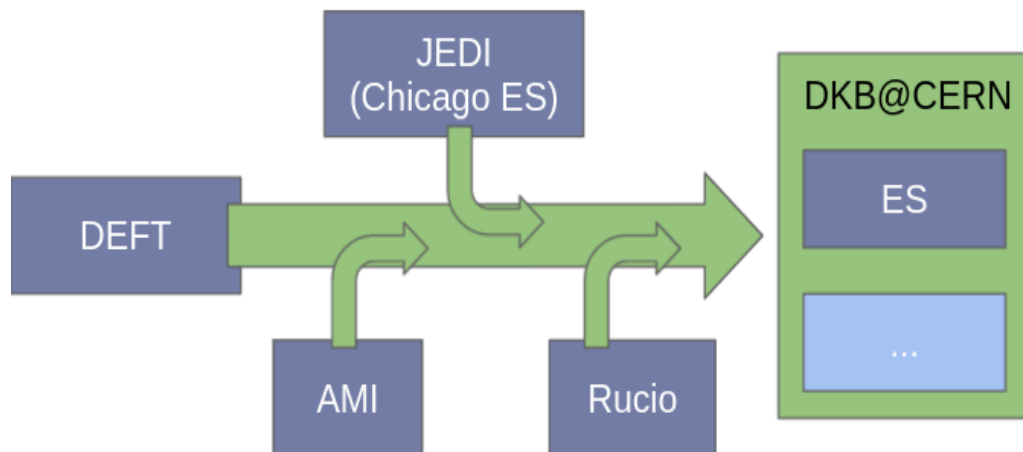
- Check if some records in DB missed in DKB

Access from aipanda[xx].cern.ch directly to ES 7.10 CERN cluster

# Current DKB metadata integration

ATLAS case:

- information update is based on “task timestamp” from ProdSys;
- main information comes from DEFT (Database Engine for Tasks) and is extended with additional metadata from other systems:
  - AMI – ATLAS Metadata Interface
  - JEDI - Job Execution and Definition Interface
  - Rucio - scientific data management system





# Current DKB metadata integration

ATLAS flow:

- ETL (Extract, Transform, Load) process
- Single bash script for data flow
  - Stages flow parts
  - DKB python library to build stages

Utils/Dataflow/data4es-nested/run/data4es-start

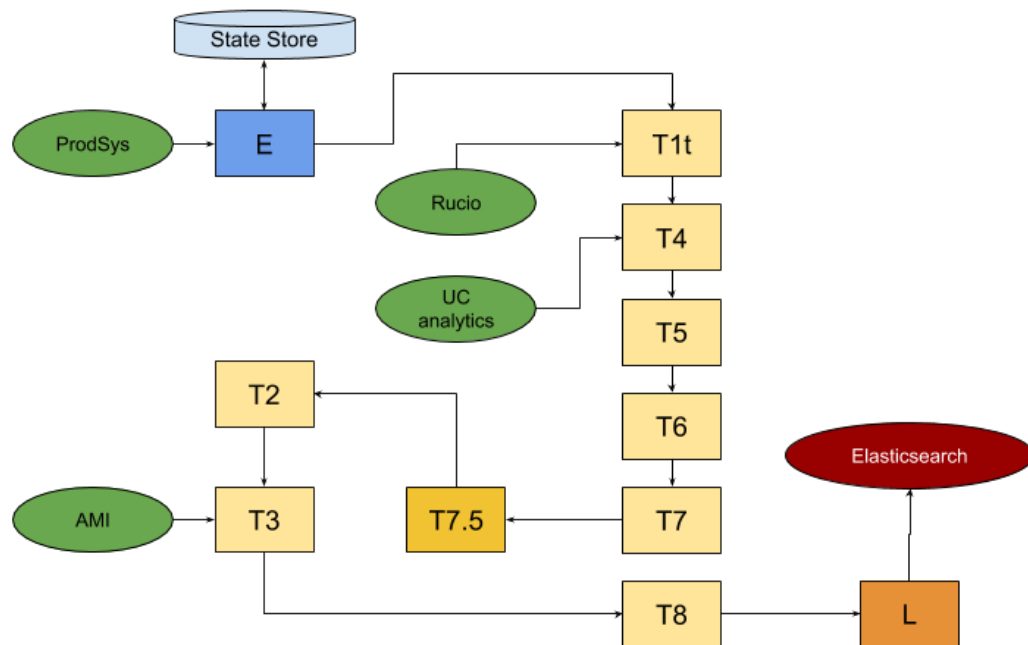
```
....
source_chain() {
  run_stage '09' | tee $b_process
}

process_chain() {
  cat $b_process | run_stage '91' | eop_filter \
    | run_stage '25' | eop_filter \
    | run_stage '16' | eop_filter \
    | run_stage '17' | eop_filter \
    | run_stage '40' | eop_filter \
    | run_stage '93' | eop_filter \
    | run_stage '95' | eop_filter
}

# Sink chain
sink_chain() {
  [ -n "$DEBUG" ] \
    && cmd_69="tee"
  tr '$\n' '$\x1e' | run_stage '19' "$1" | tr -d '$\x1e' | eop_filter \
    | mediator "$1" | run_stage '69' "$1" > /dev/null
}

process_chain | tee $out | sink_chain from95
```

Utils/Dataflow/data4es-nested/  
009\_oracleConnector  
025\_chicagoES  
091\_datasetsRucio  
016\_task2es  
040\_progress  
093\_datasetsFormat  
017\_adjustMetadata  
069\_upload2es  
095\_datasetInfoAMI  
019\_esFormat  
071\_esConsistency



04.07.2023

Grid 2023



# ATLAS use cases

There are two main use cases for DKB in ATLAS: "campaign progress monitoring" and Google-like search

- Campaign progress monitoring is heavily used by production coordinators (MC, derivation, reprocessing) and management
  - A campaign often includes thousands of tasks submitted over a long period of time, each with different metadata
  - The unique ability of DKB is to aggregate verified data using hashtags, which are used by production coordinators to track the progress of the campaign and answer questions such as how long it will take to finish
  - Management also utilizes DKB to check the amount of resources (disk space, time) used for future campaign predictions
- The Google-like search feature is extremely useful for tracing issues and is used by shifters and distributed computing experts
  - Specifically, if a problem arises (such as task failures), this feature allows users to find similar productions and observe how a particular dataset was processed in the past



# ATLAS use case 1

Hashtags: |QuickRepro2022|PeriodF2022

total	done	finished	failed	broken	aborted
1236	1138	73	3	13	9

Step Name									
Tasks total (running)	<b>HIST p5450</b> 100.00%	<b>HIST r14190</b> 99.99%	<b>DAOD_IDTIDE r14190</b> 99.99%	<b>AOD r14190</b> 99.99%	<b>DESDM_MCP r14190</b> 99.99%	<b>DRAW_EGZ r14190</b> 99.99%	<b>DRAW_ZMUMU r14190</b> 99.99%	<b>AOD p5449</b> 100.00%	<b>C p5449</b> 100.00%
Input events (running tasks)	366 (366)	122 (122)	122 (122)	122 (122)	122 (122)	122 (122)	122 (122)	122 (122)	1
Processed events	~8,223,173,691	2,741,126,164	2,741,126,164	2,741,126,164	2,741,126,164	2,741,126,164	2,741,126,164	2,741,057,897	3
Output task events	8,223,173,691	2,741,057,897	2,741,057,897	2,741,057,897	2,741,057,897	2,741,057,897	2,741,057,897	2,741,057,897	3
Running/Pending/Not started	8,223,173,691	2,741,067,839	2,741,067,839	2,741,067,839	2,741,067,839	2,741,067,839	2,741,067,839	2,741,057,897	3
Input bytes (running tasks)	0%/0%/0%	0%/0%/0%	0%/0%/0%	0%/0%/0%	0%/0%/0%	0%/0%/0%	0%/0%/0%	0%/0%/0%	0
Input bytes (done tasks)	26.29 TB ( 364 tasks)	3.84 PB ( 122 tasks)	3.84 PB ( 122 tasks)	3.84 PB ( 122 tasks)	3.84 PB ( 122 tasks)	3.84 PB ( 122 tasks)	3.84 PB ( 122 tasks)	834.84 TB ( 122 tasks)	1 1
Output bytes	0 B ( 0 tasks)	0 B ( 0 tasks)	0 B ( 0 tasks)	0 B ( 0 tasks)	0 B ( 0 tasks)	0 B ( 0 tasks)	0 B ( 0 tasks)	0 B ( 0 tasks)	0
Average HS06 per event	1.57 TB ( 364 tasks)	24.91 TB ( 122 tasks)	2.49 TB ( 122 tasks)	834.84 TB ( 122 tasks)	118.30 TB ( 122 tasks)	29.12 TB ( 122 tasks)	31.82 TB ( 122 tasks)	829.85 TB ( 122 tasks)	9 1
Events (finished datasets)	1	277	277	277	277	277	277	5	3
Duration (finished tasks)	8,179,700,816	2,741,057,897	7,513,531	2,741,057,897	323,262,709	20,556,893	22,066,302	2,741,057,897	3
	1.08 days	2.55 days	2.55 days	2.55 days	2.55 days	2.55 days	2.55 days	3.48 days	2

Total done (estimation): 99.99%



# ATLAS use case 2

Search keywords:

s4122,801451

Search

Analysis ☐

**Keywords: s4122 AND 801451**

**Tasks found: 47 Tasks displayed: 47**

Manage tasks

Filter

Default



32761639	mc21_13p6TeV.801451.Py8EG_A3_NNP23LO_minbias_ND.simul.e8486_s4122	0
32767413	mc21_13p6TeV.801451.Py8EG_A3_NNP23LO_minbias_ND.simul.e8486_s4122	100000
32761502	mc21_13p6TeV.801451.Py8EG_A3_NNP23LO_minbias_ND.simul.e8486_s4122	0
32937117	mc21_13p6TeV.801451.Py8EG_A3_NNP23LO_minbias_ND.simul.e8486_s4122	5000000
33059877	mc21_13p6TeV.801451.Py8EG_A3_NNP23LO_minbias_ND.simul.e8486_s4122	0
32769915	mc21_13p6TeV.801451.Py8EG_A3_NNP23LO_minbias_ND.simul.e8486_s4122	100000
32937126	mc21_13p6TeV.801451.Py8EG_A3_NNP23LO_minbias_ND.simul.e8486_s4122	4995000



# Resent changes: move to python env

Before:

- DKB used OS based installation
- Additional setup was done by puppet profiles (CERN depended)

Now:

- Special OS independent python environment created (based on minicoda)
- All python DKB dependency described by environment yaml file

Advantages:

- Possible to use more resent python
- Possible to use combo conda+pip in env
- Locked stable environment by yaml
- Easy deploy on any OS and inside containers



# Resent changes: migration 2to3

For migration:

- Many lines of python code
- Many authors
- Some external python libraries in use
- No more python2 support
- Important: DKB has some tests implemented

How to migrate:

1. Initial migration performed by running 2to3 python helper program:
  - `find /opt/inMem/dkb/ -type f -name *.py -exec 2to3 -n -w -f all -f idioms -f buffer -f set_literal -f ws_comma {} ;`
2. Run DBK test program and do specific fixes for python3 :
  - all strings are Unicode
  - no more unbuffered io for strings
  - sorted dictionary in mem
  - no more encoded strings by default
  - changes in error output for tests
3. Run preprod DKB 3 in parallel with prod for tasks workflow verification
  - changes in Elasticsearch python3 library
  - changes in Oracle python3 library
  - no more decode `string\_escape`



# Plans

- Switch to Python3
  - Everything is ready
- Start looking to AlmaLinux 9
  - More puppet specific for CERN infrastructure
- Check tokens usage instead of Grid certificates
  - Rucio, AMI
- Add more use cases for Atlas workflows



Thank you!

Any questions?