# CERN Central Monitoring Overview

Borja Garrido Bear (On behalf of the monitoring team)

12.09.2024

**INTRODUCTION**

- The MONIT Infrastructure
- Some users

THE THREE USE CASES

FTS

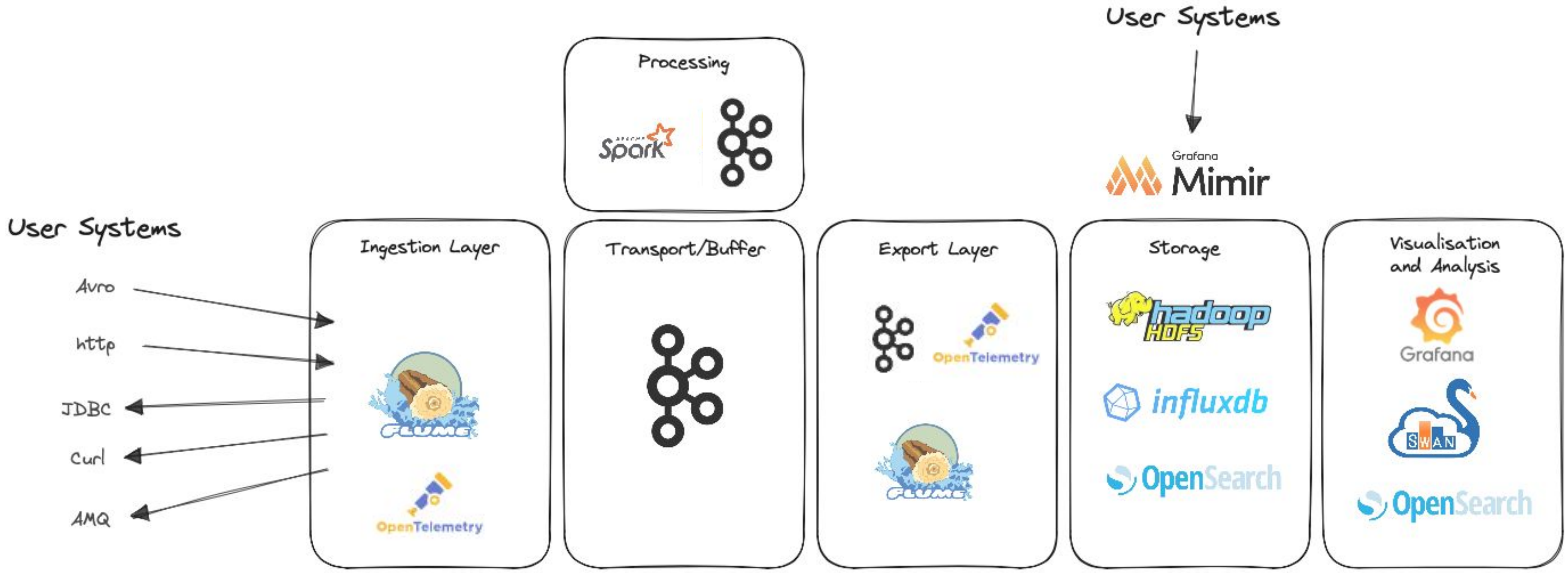
DDM

XROOTD

SUMMARY

# The MONIT infrastructure

**The aim of the monitoring team is to provide an infrastructure allowing the usage of tools and service to ease the load of monitoring CERN data centres (host and services) and WLCG experiments**



# Some users

wmarchive
eoscmsquotas
sso
certmgr
pandaqueuestatus
mig
netstat
accounting
eos_users
frontier
eventindex
cms_rucio
afs
sdr
si-condor
volhcb
pbeast
cms-ms-pileup
cvmfs
old-sso
itdb
rucioacc
acron
cmsaaa
eosc-future
openstack
cernbox
amssoc
network
cms_aaa_subscription
dcache
panda
rucio
lxplus
fts
cms-fts-logsanalysis
cmsunified
landb
sam3
admon
prometheus
opint
voms
cms-cmpwg
cms_eos_mon
cmst0
indico
wmagent
openshift
dbod
cdr
filer
condor
escape_wp2
oracle
git
backupservice
digitalrepos
eventservice
wlcgops
cloud
cmssst
tier0wmagent
linuxsupport
dir
config
mdm
jeedy
hadoop
crab
lb-telemetry
smashboxtests
atlascric
twiki
ceph
cmssw_pop
atlas-ami
slurm-sls
publicscreens
atlasact
tape
drupal
xrootd
certsocmom
conferencerooms
atlasagis
hedison
monitsli
atlasssb
it-elasticsearch
bigpandamon
lpwan
tottc
batch
monitsli-qa
eos
cms_dmops
cms-es-size
cric
ssb
service
cms-xrootd-cache
wininfra
myproxy
loadbalancer
alice-uem
lcgbdii
lhcathome
green-it

INTRODUCTION

## THE THREE USE CASES

- Same set of tools
- Data processing
- CRIC topology document

FTS

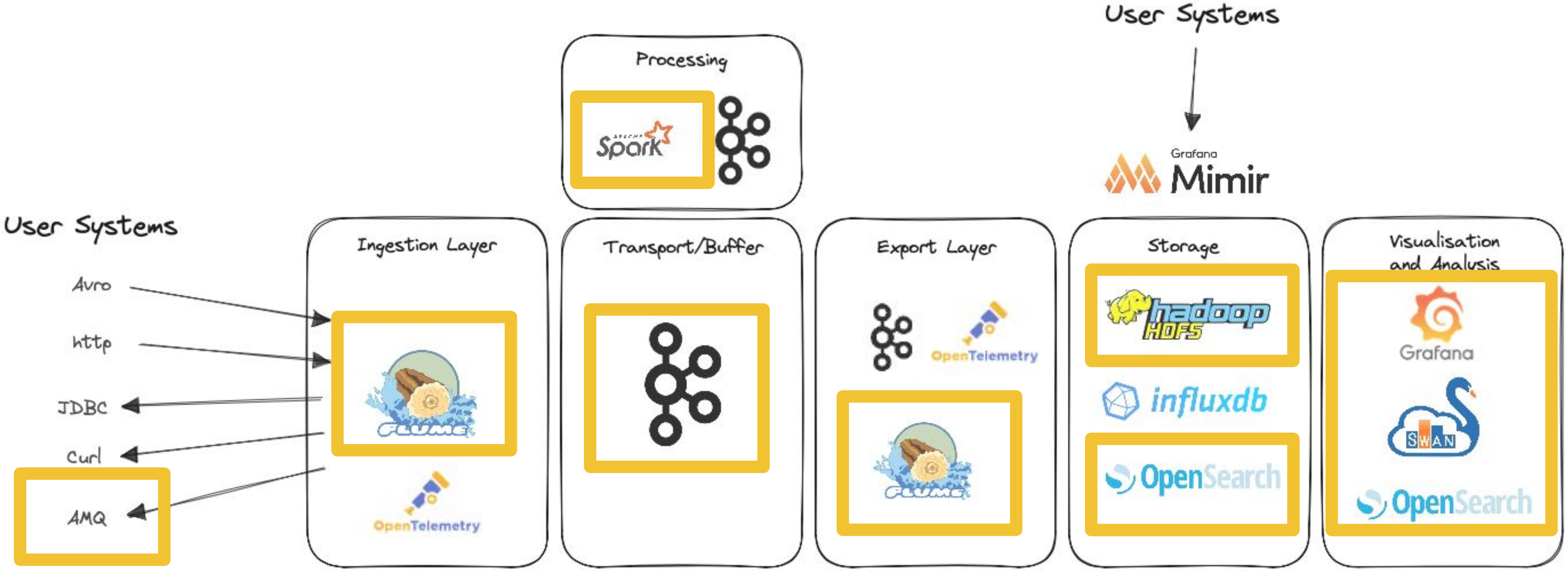
DDM

XROOTD

SUMMARY

# Same set of tools

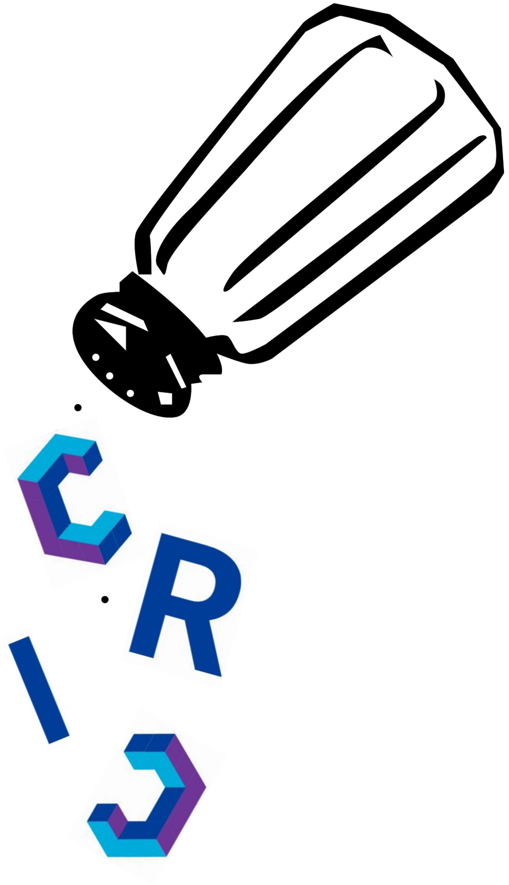
- **FTS, DDM and XRootD flows share a similar processing within MONIT**
  - Ingested via Message broker (AMQ) in the shape of JSON
  - Processed with Apache Spark (enrichment, aggregation)
  - Stored in Opensearch (short term and long term) and HDFS
  - Accessible from Grafana, Opensearch or SWAN



# Data processing

- **Data is processed with two main goals behind**
  - Real time monitoring, requires stream enrichment, stored for short-term (few months)
  - Accounting, requires batch enrichment and aggregation, stored forever
- **Enrichment process**
  - Extend the base document, in general with topology information (CRIC)
    - I.e: site, country, federation…
- **Aggregation process**
  - Create buckets based on time and a selected set of labels, usually 1h

# CRIC Topology Document



- **MONIT does a dump of CRIC endpoints every 6 hours**
- **Enrichment jobs get changes on MONIT CRIC flow every 24 hours**
- **Aggregation jobs get changes on MONIT CRIC flow every execution**
- **The CRIC document contains the following fields**
  - CRIC fields used for general enrichment

| | | | | |
|---|---|---|---|---|
| VO | experiment_site | official_site | federation | state |
| country | country_code | description | tier | institute_name |
| netroutes | endpoint | hostname | flavour | status |

INTRODUCTION

THE THREE USE CASES

## FTS

- Types of data
- Transfer Complete transformations
- Dashboards

DDM

XROOTD

SUMMARY

# Types of data

**TRANSFER START**
(monit_prod_fts_raw_start)

Created only by the transfer agent at start of transfer

**TRANSFER COMPLETE**
(monit_prod_fts_raw_complete)

Created only by the transfer agent at end of transfer

**TRANSFER STATE**
(monit_prod_fts_raw_state)

Created by QoS and Transfer daemons when a file state changes

**OPTIMIZER**
(monit_prod_fts_raw_queue_state)

Created by the Optimizer when adjusting parameters

- **Information taken from this presentation by the FTS team**

# Transfer Complete transformations (I)

- **Initial round of derivation (extract fields from other fields)**
  - Field added by MONIT, Field provided by FTS

| | | | |
|---|---|---|---|
| src/dst_se <- src/dst_url | protocol <- src/dst_url | Remove paths from src/dst_hostname | job_id <- tr_id |
| file_id <- tr_id | log_link <- https://{endpnt}:8449/fts3/ftsmon/#/job/{job_id} | file_size <- f_size | latency <- now() - timestamp_tr_comp |
| operation_time <- (tr_timestamp_complete - tr_timestamp_start) | transfer_time <- (timestamp_tr_comp - timestamp_tr_st) | throughput <- f_size / (transfer_time/1000) | srm_finalization_time <- (time_srm_fin_end - time_srm_fin_st) |
| srm_preparation_time <- (time_srm_prep_end - time_srm_prep_st) | srm_overhead_time <- (srm_preparation_time - srm_finalization_time) | srm_overhead_percentage | timestamp_checksum__diff<- timestamp_checksum__diff - timestamp_checksum__st |
| t_final_transfer_state_flag <- t_final_transfer_state | activity <- file_metadata[activity] | dst/src_rse <- file_metadata[dst/src_rse] | _id <- SHA1(job_id+retry+tr_id) |

Complete format in FTS Messaging documentation

# Transfer Complete transformations (II)

- **Enrichment with different sources**
  - This ends up producing so called FTS Complete Enriched dataset (monit_prod_fts_enr_complete)

### CRIC

- Match by:
  - VO
  - Endpoint
- Added fields (src/dst):
  - Hostname
  - Experiment_site
  - Site
  - Tier
  - Country
  - Federation

### Transfer State

- Match by:
  - Timestamp
  - File_ID
- Added fields:
  - Staging
  - Staging_start
  - Staging_finished
  - User_filesize

### Final derivations

- Remote_access:
  - Src_site != Dst_site
- Transferred_volume:
  - File_size if complete
  - Tr_bt_transferred else

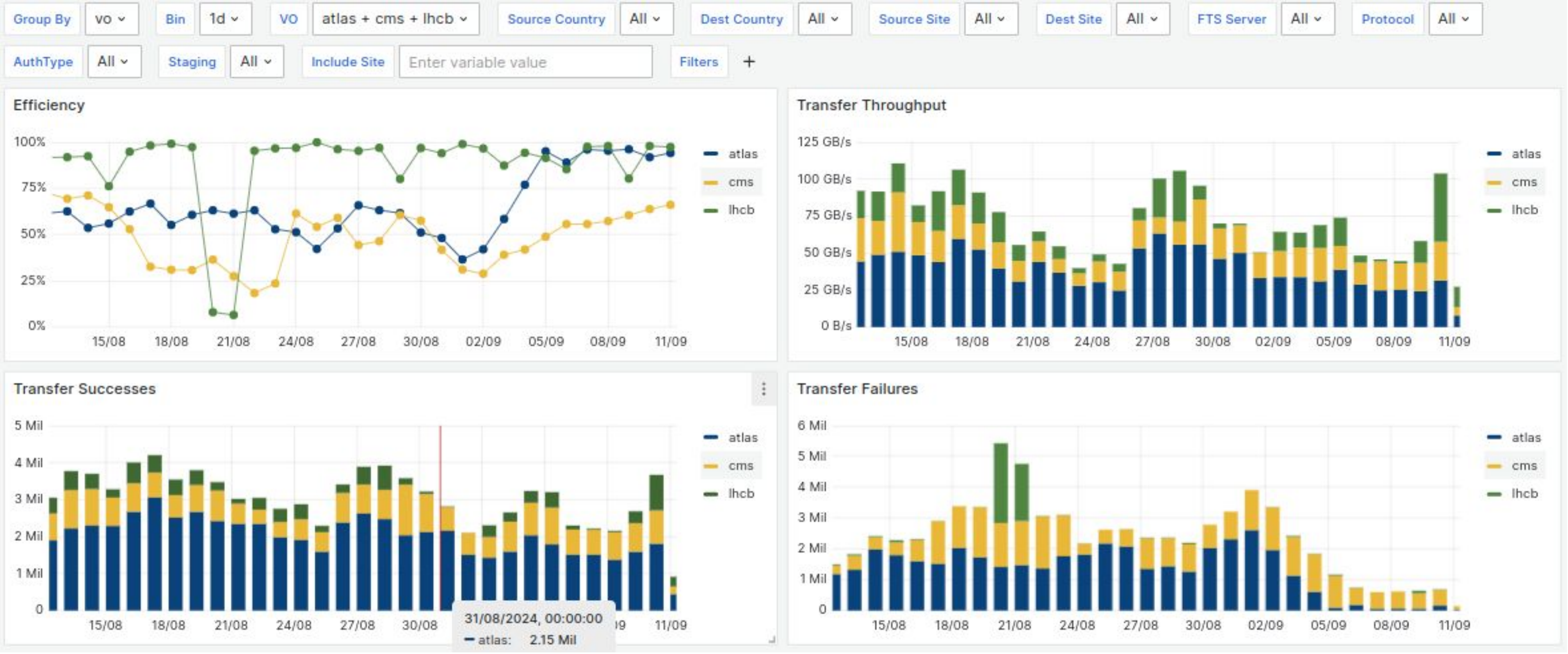
# Transfer Complete transformations (III)

- **Aggregation for accounting**
  - This ends up producing so called FTS Complete Aggregated dataset (monit_prod_fts_agg_complete)
  - Aggregates data by labels every hour and corrects produced data up to 2 days in the past using the labels as primary key, uses the tr_timestamp_complete field for the time window

| | | | | |
|---|---|---|---|---|
| activity | channel_type | country | experiment_site | federation |
| hostname | se | site | srm_v | tier |
| endpnt | ipv6 | vo | is_recoverable | job_state |
| remote_access | srm_space_token | staging | t_failure_phase | technology |
| t_final_transfer_state | tr_error_category | tr_error_scope | rse | protocol |
| ipver | auth_method | transfer_type | count | avg_file_size |
| transferred_volume | avg\|sum operation_time | avg\|sum overhead_time | sum_transfer_state | sum_staging_duration |
| stage_transferred_volume | | sum_transfer_time | | |

# Dashboards

- Initially accounting dashboard was the main goal for WLCG Monitoring
- Data was there, so people started using it to build their own dashboards



Group By: vo · Bin: 1d · VO: atlas + cms + lhcb · Source Country: All · Dest Country: All · Source Site: All · Dest Site: All · FTS Server: All · Protocol: All

AuthType: All · Staging: All · Include Site: Enter variable value · Filters +

INTRODUCTION

THE THREE USE CASES

FTS

## DDM

- Types of data
- Transformations
- Dashboards

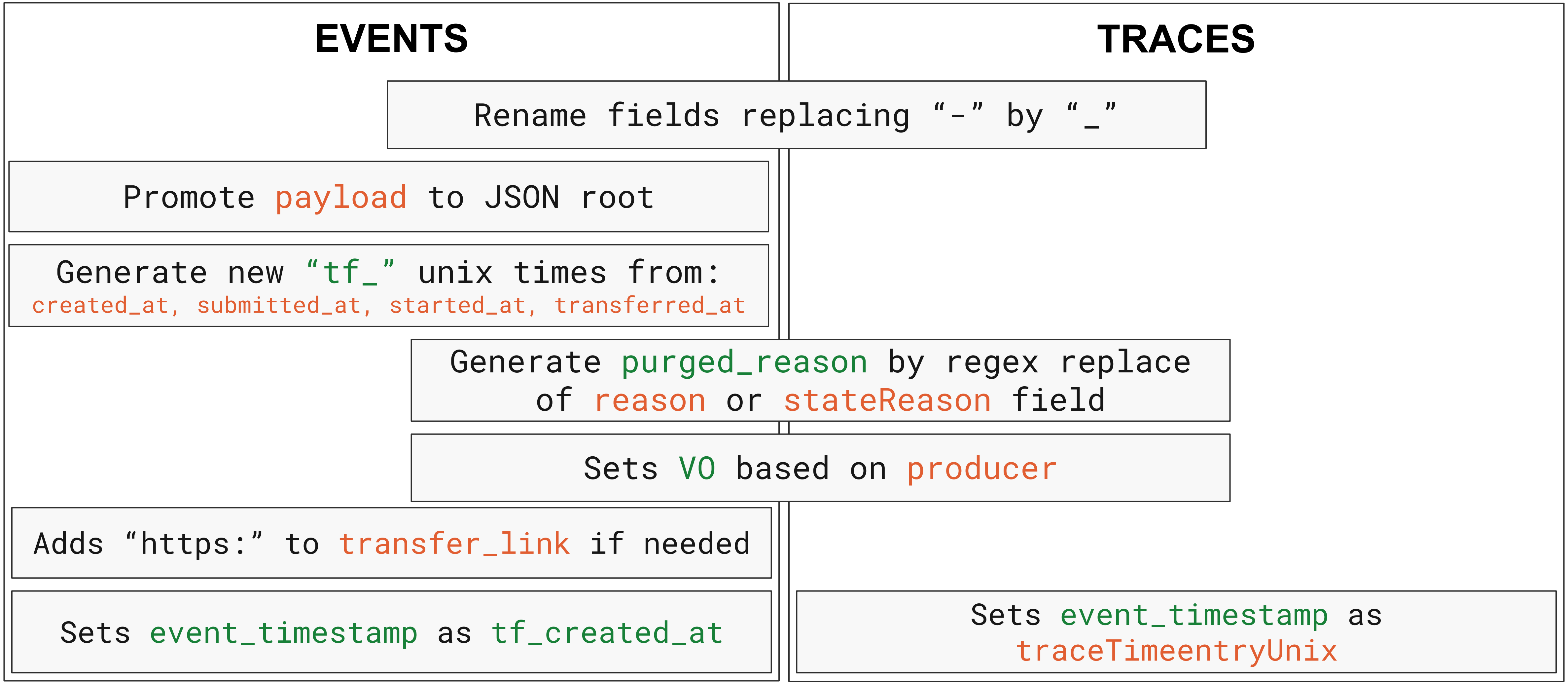
XROOTD

SUMMARY

# Types of data

- **Transfer Monitoring (Rucio events)**
  - Transfers events
  - Deletion events
- **Access Monitoring (Rucio traces)**
  - Traces sent by pilot when a file is downloaded/uploaded

# Transformations (I)



EVENTS

TRACES

Rename fields replacing “-” by “_”

Promote payload to JSON root

Generate new “tf_” unix times from:
created_at, submitted_at, started_at, transferred_at

Generate purged_reason by regex replace of reason or stateReason field

Sets VO based on producer

Adds “https:” to transfer_link if needed

Sets event_timestamp as tf_created_at

Sets event_timestamp as traceTimeentryUnix

# Transformations (II)

- **Enrichment with different sources (Only for ATLAS)**
  - This ends up producing so called DDM transfer Enriched dataset (monit_prod_ddm_enr_transfer)

### CRIC (ATLAS)

- Match by:
  - Endpoint
- Added fields (src/dst):
  - Experiment_site
  - Token
  - Cloud

### CRIC (WLCG)

- Match by:
  - Experiment_site
- Added fields (src/dst):
  - Tier
  - Country
  - Federation

### Final derivations

- Remote_access:
  - Src_site != Dst_site

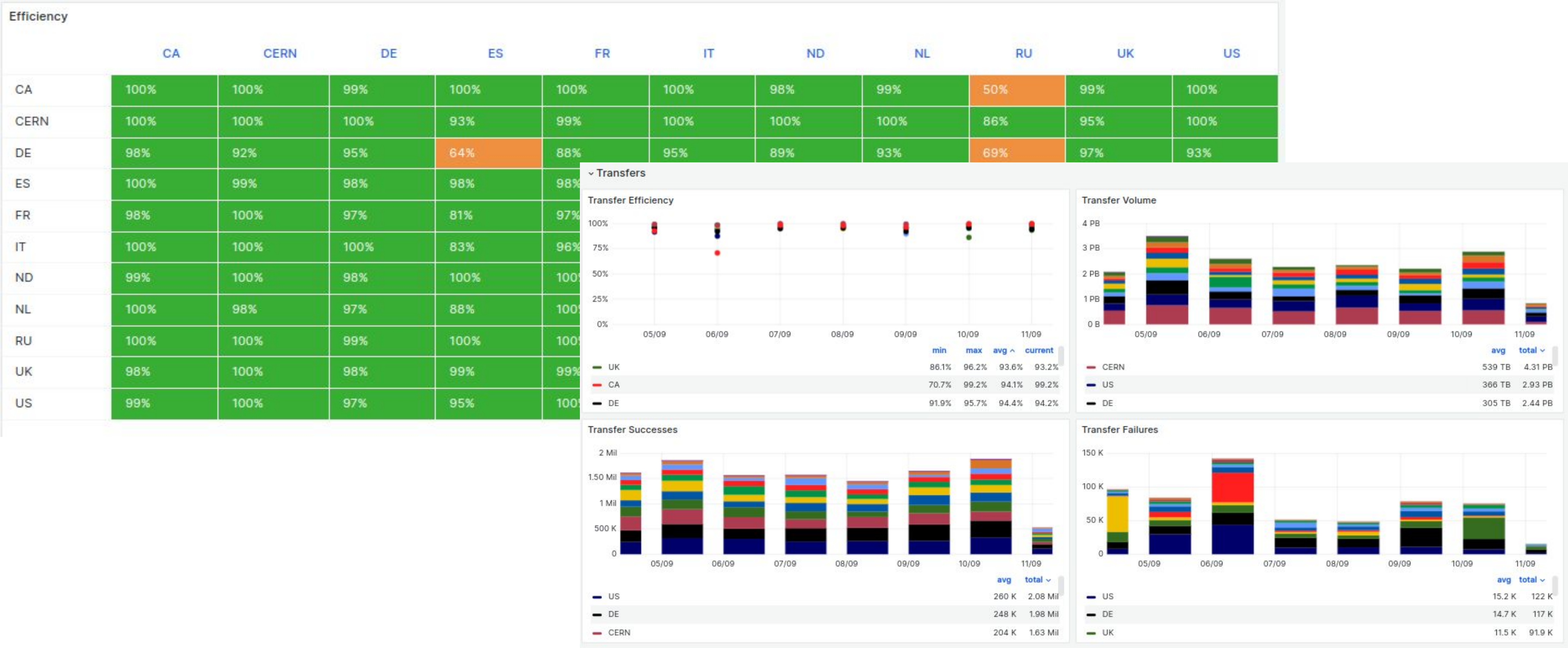
# Transformations (III)

- **Aggregation for accounting**
  - This ends up producing so called DDM Aggregated dataset (monit_prod_ddm_agg_transfer)
  - Aggregates data by labels every hour and corrects produced data up to 1 day in the past using the labels as primary key, uses the event_timestamp field for the time window

| | | | | |
|---|---|---|---|---|
| state | activity | protocol | endpoint | is_staging |
| fts_host | vo | site | tier | federation |
| country | event_type | remote_access | experiment_site | token |
| cloud | | | | |
| bytes_done | bytes_failed | bytes_planned | bytes_total | |
| files_done | files_failed | files_planned | files_total | |
| throughtput_done | throughtput_failed | throughput_planned | throughtput_total | |

# Dashboards

- Done in collaboration with ATLAS for their monitoring



Efficiency

| | CA | CERN | DE | ES | FR | IT | ND | NL | RU | UK | US |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CA | 100% | 100% | 99% | 100% | 100% | 100% | 98% | 99% | 50% | 99% | 100% |
| CERN | 100% | 100% | 100% | 93% | 99% | 100% | 100% | 100% | 86% | 95% | 100% |
| DE | 98% | 92% | 95% | 64% | 88% | 95% | 89% | 93% | 69% | 97% | 93% |
| ES | 100% | 99% | 98% | 98% | 98% | | | | | | |
| FR | 98% | 100% | 97% | 81% | 97% | | | | | | |
| IT | 100% | 100% | 100% | 83% | 96% | | | | | | |
| ND | 99% | 100% | 98% | 100% | 100 | | | | | | |
| NL | 100% | 98% | 97% | 88% | 100 | | | | | | |
| RU | 100% | 100% | 99% | 100% | 100 | | | | | | |
| UK | 98% | 100% | 98% | 99% | 99% | | | | | | |
| US | 99% | 100% | 97% | 95% | 100 | | | | | | |

⌄ Transfers

Transfer Efficiency

| | min | max | avg ^ | current |
|---|---|---|---|---|
| UK | 86.1% | 96.2% | 93.6% | 93.2% |
| CA | 70.7% | 99.2% | 94.1% | 99.2% |
| DE | 91.9% | 95.7% | 94.4% | 94.2% |

Transfer Volume

| | avg | total ⌄ |
|---|---|---|
| CERN | 539 TB | 4.31 PB |
| US | 366 TB | 2.93 PB |
| DE | 305 TB | 2.44 PB |

Transfer Successes

| | avg | total ⌄ |
|---|---|---|
| US | 260 K | 2.08 Mil |
| DE | 248 K | 1.98 Mil |
| CERN | 204 K | 1.63 Mil |

Transfer Failures

| | avg | total ⌄ |
|---|---|---|
| US | 15.2 K | 122 K |
| DE | 14.7 K | 117 K |
| UK | 11.5 K | 91.9 K |

INTRODUCTION

THE THREE USE CASES

FTS

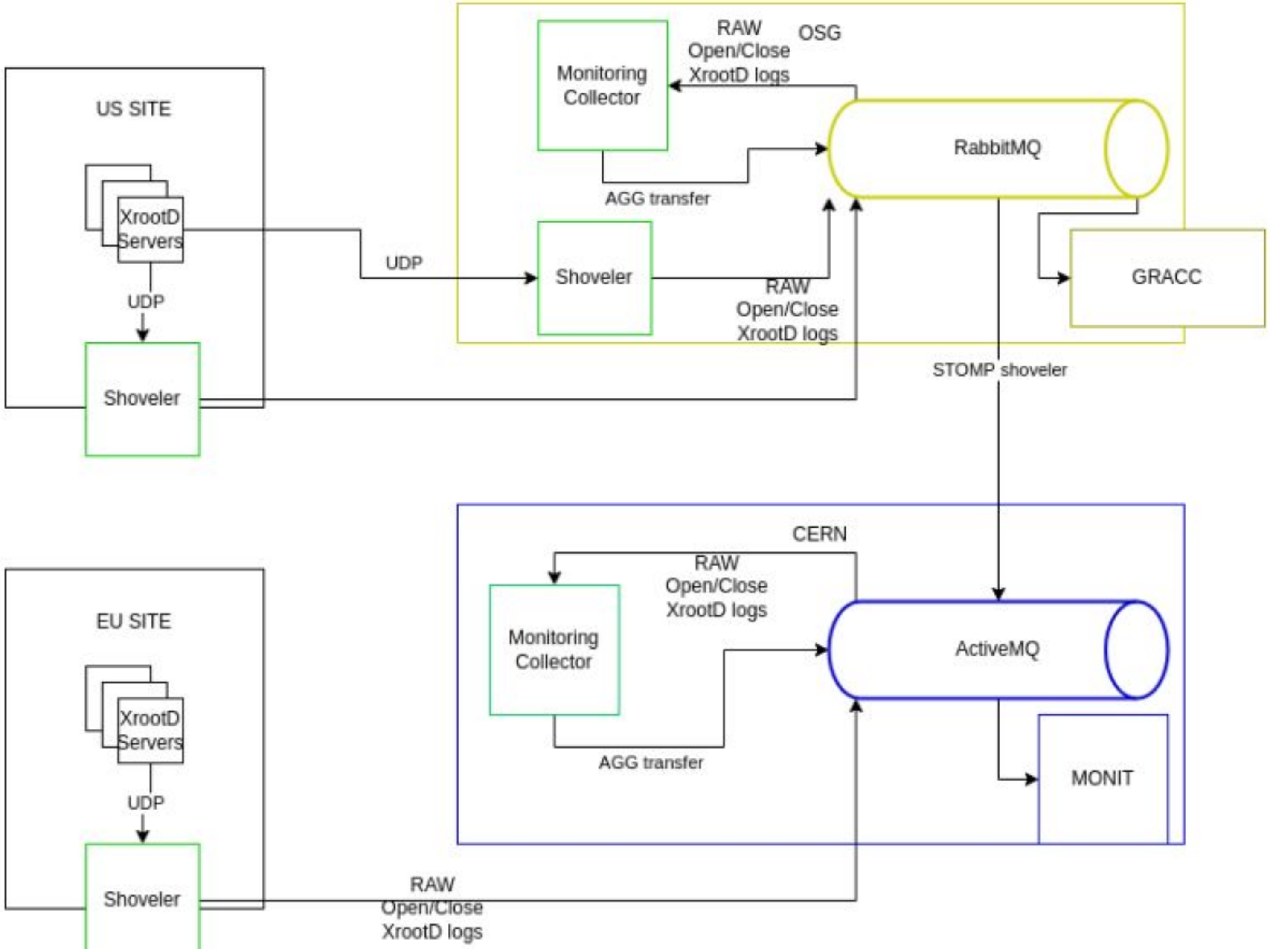
DDM

## XROOTD

- New flow
- Transformations
- Dashboards

SUMMARY

# New Flow


OSG
Monitoring Collector
RAW Open/Close XrootD logs
RabbitMQ
AGG transfer
Shoveler
RAW Open/Close XrootD logs
GRACC
UDP
US SITE
XrootD Servers
UDP
Shoveler
STOMP shoveler
CERN
Monitoring Collector
RAW Open/Close XrootD logs
ActiveMQ
AGG transfer
MONIT
EU SITE
XrootD Servers
UDP
Shoveler
RAW Open/Close XrootD logs

# New Flow (New components)

- **Shoveler**
  - Receives streams (UDP) and persists them to a message queue
    - It also does stream validation and IP translation (configurable map)
  - Should sit close to the XRootD server (run at sites)
    - Avoid UDP fragmentation and general unreliability
- **Collector**
  - One single collector per Message Queue (run centrally)
  - Aggregates streams into a transfer document
    - Needs to keep state due to how XRootD produces streams
  - Does some initial massaging of the transfer document
    - Extracts VOs, resolves hostnames, domains…

# Transformations (I)

- **Much simpler that in the previous cases**
  - Mainly because the collector does most of the required derivations before sending the data

```
remote_access <-
client_domain != server_domain
```

```
is_transfer <-
bytes_read + bytes_write == file_size
```

```
event_timestamp <- end_time
```

# Transformations (II)

- **Join together WLCG and US transfers (currently integrated in different flows)**
- **Enrichment with different sources**
  - This ends up producing so called XRootD transfer enriched dataset (monit_prod_xrootd_enr_transfer)

**Static Topology**

- Match by:
  - Domain
  - VO
- Added fields (src/dst):
  - Site
  - Experiment_site
  - Country

**CRIC (WLCG)**

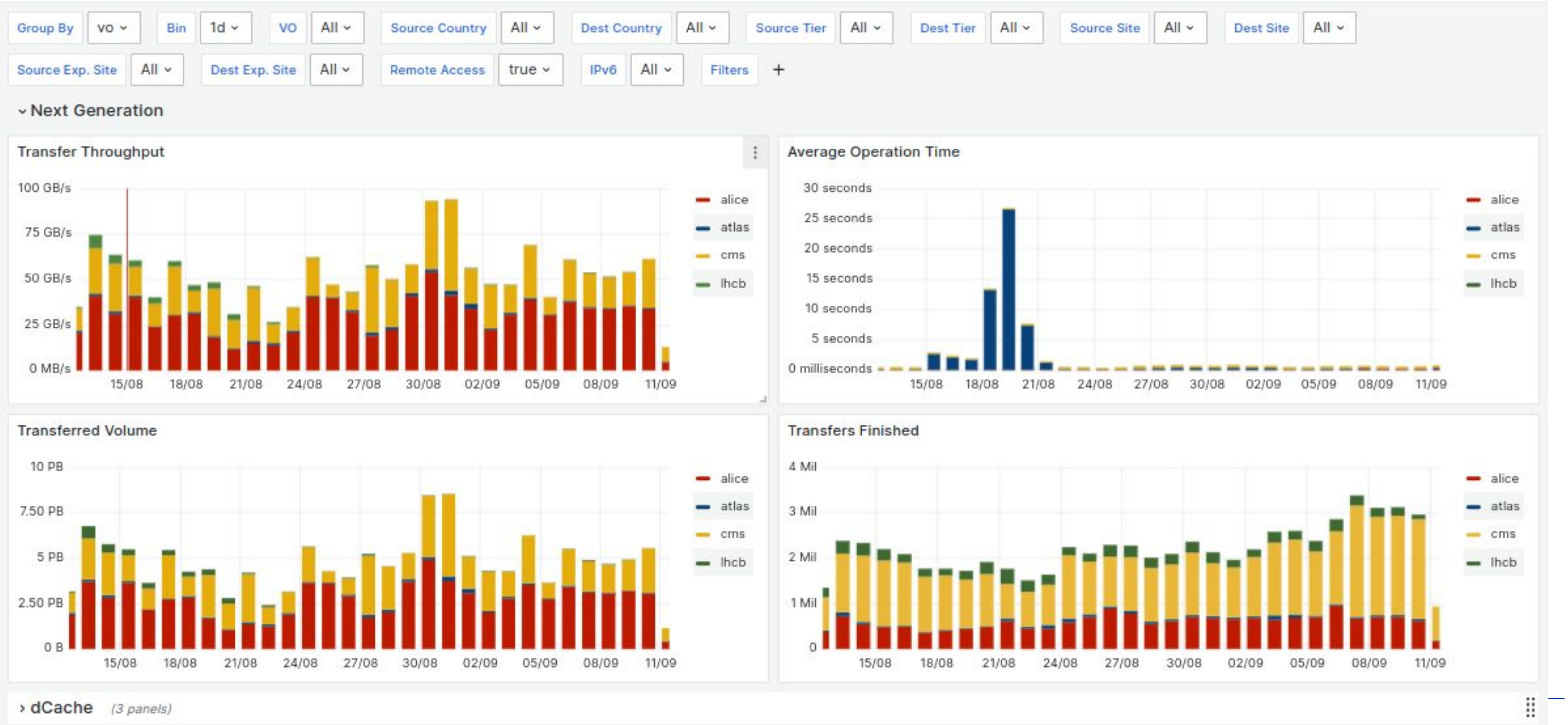
- Match by:
  - Experiment_site
- Added fields (src/dst):
  - Tier
  - Federation

# Transformations (III)

- **Aggregation for accounting**
  - This ends up producing so called XRootD Aggregated dataset (monit_prod_xrootdng_agg_transfer)
  - Aggregates data by labels every hour and corrects produced data up to 6 hours in the past using the labels as primary key, uses the event_timestamp field for the time window

| | | | | |
|---|---|---|---|---|
| vo | site | ipv6 | is_transfer | operation |
| server_site | experiment_site | tier | country | federation |
| hostname* | technology | remote_access | | |
| count | avg_file_size | sum_file_size | transferred_volume | |
| avg_operation_time | sum_operation_time | | | |

# Dashboards



Group By: vo ⌄ | Bin: 1d ⌄ | VO: All ⌄ | Source Country: All ⌄ | Dest Country: All ⌄ | Source Tier: All ⌄ | Dest Tier: All ⌄ | Source Site: All ⌄ | Dest Site: All ⌄

Source Exp. Site: All ⌄ | Dest Exp. Site: All ⌄ | Remote Access: true ⌄ | IPv6: All ⌄ | Filters +

⌄ Next Generation

Transfer Throughput

Average Operation Time

Transferred Volume

Transfers Finished

› dCache *(3 panels)*

INTRODUCTION

THE THREE USE CASES

FTS

DDM

XROOTD

# SUMMARY

# Summary

- **There’s more to the eye from just “simple” integration into MONIT**
  - FTS and DDM flows have been curated with time (10+ years)
  - New fields are added from other datasources as requested
- **Split knowledge of the monitoring (not ideal)**
  - Tools experts know what they produce
  - MONIT experts know all the extra bits happening under stage
- **Be careful when putting different flows against each other**
  - Something like the timestamp concept difference might drive to very different plots!
- **WLCG Goal remains the same**
  - Being able to plot together WLCG transfers data (FTS + XRootD)
- **XRootD new flow still in early days**
  - Will require lots of work to arrive to a “stable” state comparable with FTS or DDM

# Thank you !

Q & A

SNOW: Monitoring Service
Mattermost: MONIT
Docs: https://monit-docs.web.cern.ch/

CERN

home.cern