

Unified Experiment Monitoring

CHEP 2024

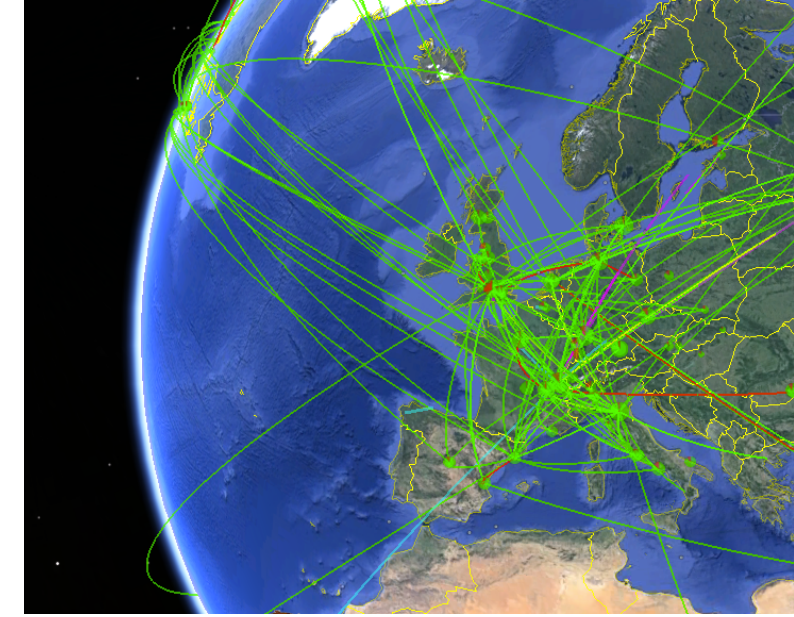
Ewoud Ketele, Domenico Giordano, Maarten Litmaath, Panos Paparrigopoulos | CERN-IT

23.10.2024

Monitoring at WLCG

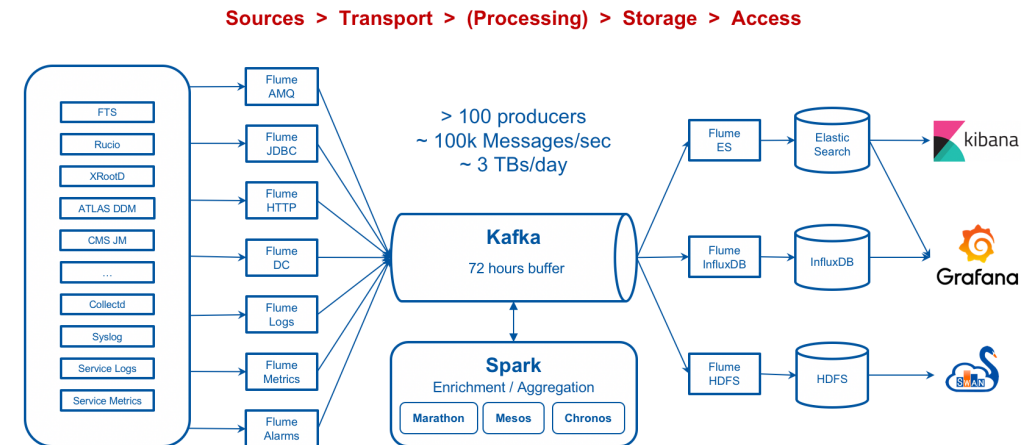
Monitoring is a critical component of the WLCG infrastructure

- EGI and experiment monitoring
- WLCG accounting
 - Overview of the resources available and resources consumed
 - Reporting resource consumption to sites, experiments, WLCG and funding agencies



Monitoring at WLCG is done using MONIT

- The centralized monitoring service at CERN IT
- Provides monitoring tools to several IT services and experiments



monit-docs.web.cern.ch/overview

MONIT ... with some gaps

MONIT Home Dashb

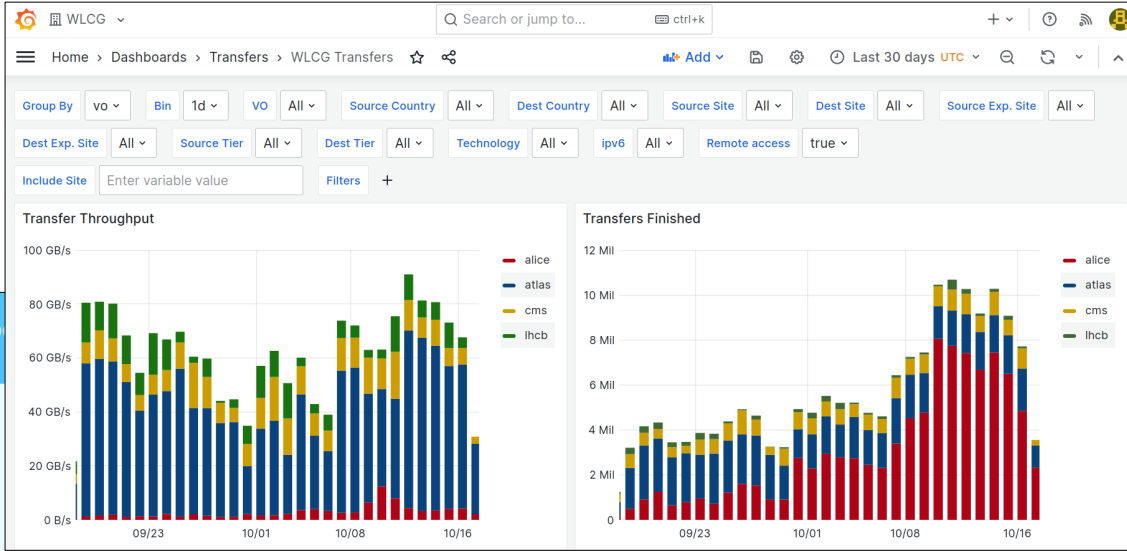
Dashboards

Direct access to general MONIT dashboards

- DATA CENTRE ▾
- LHCOPN ▾

Grafana organizations for experiment monitoring efforts

monit.cern.ch



WLCG transfers

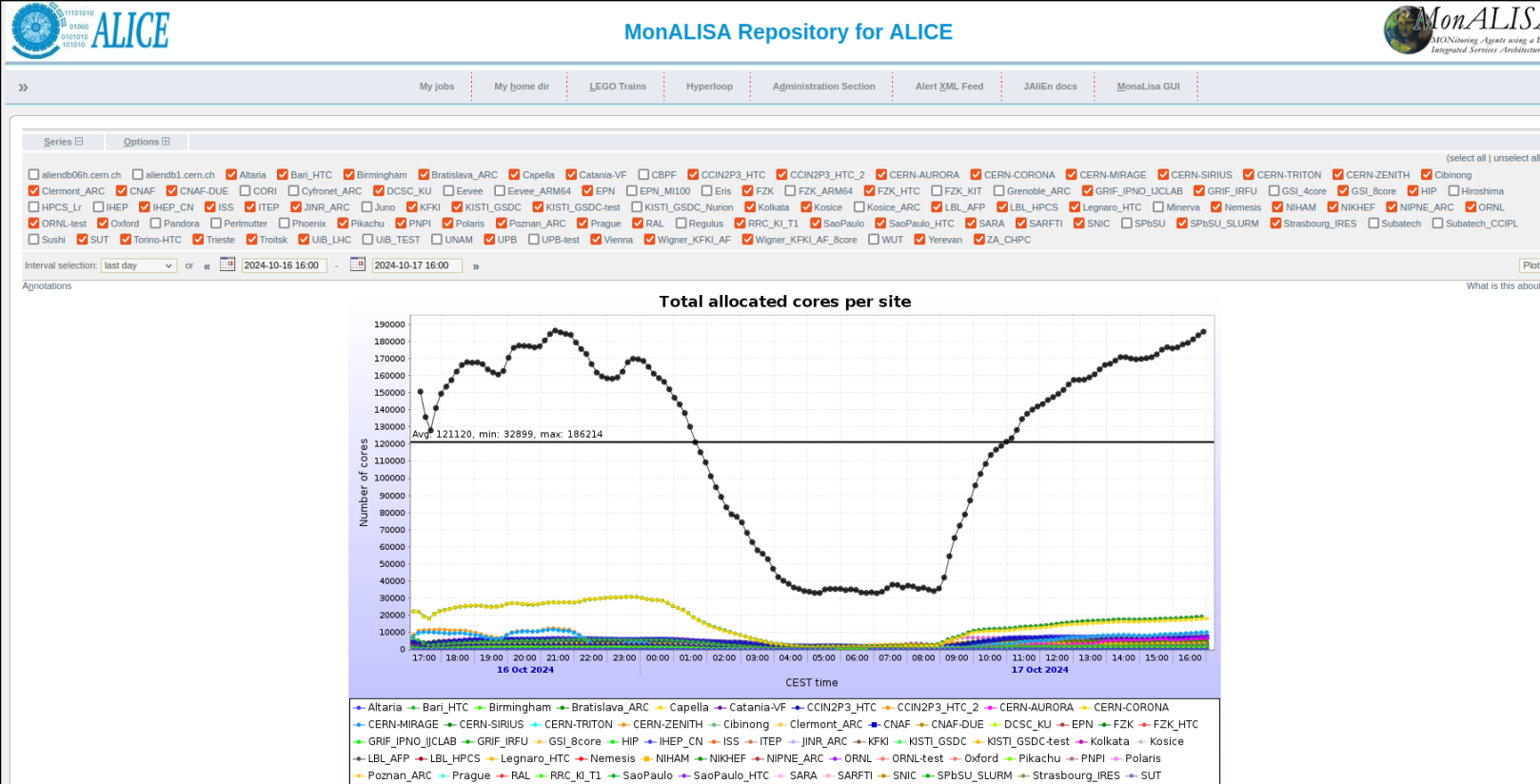
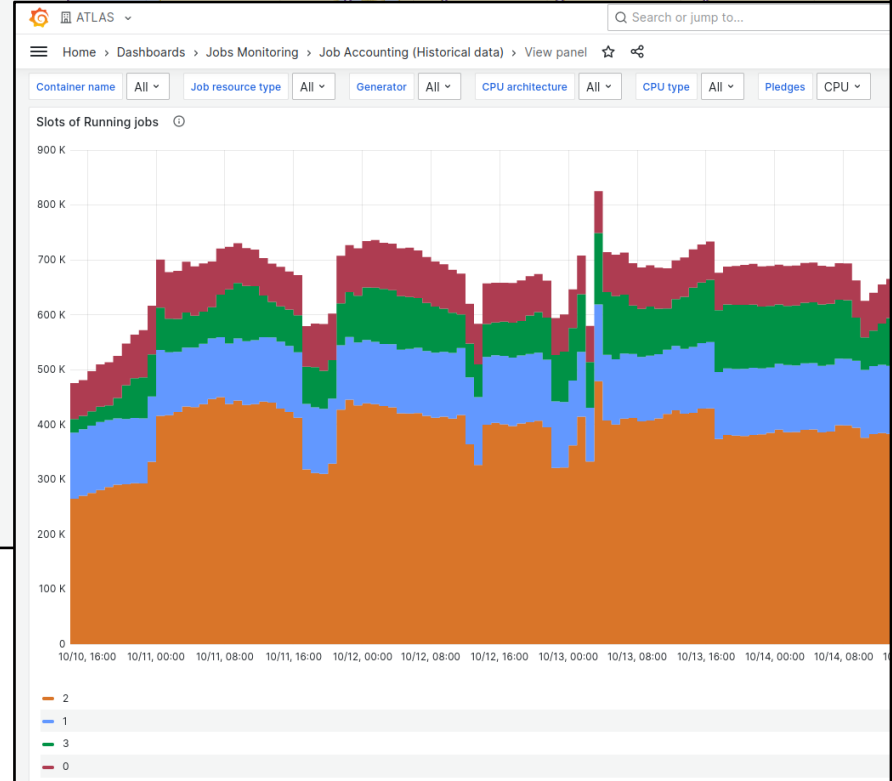
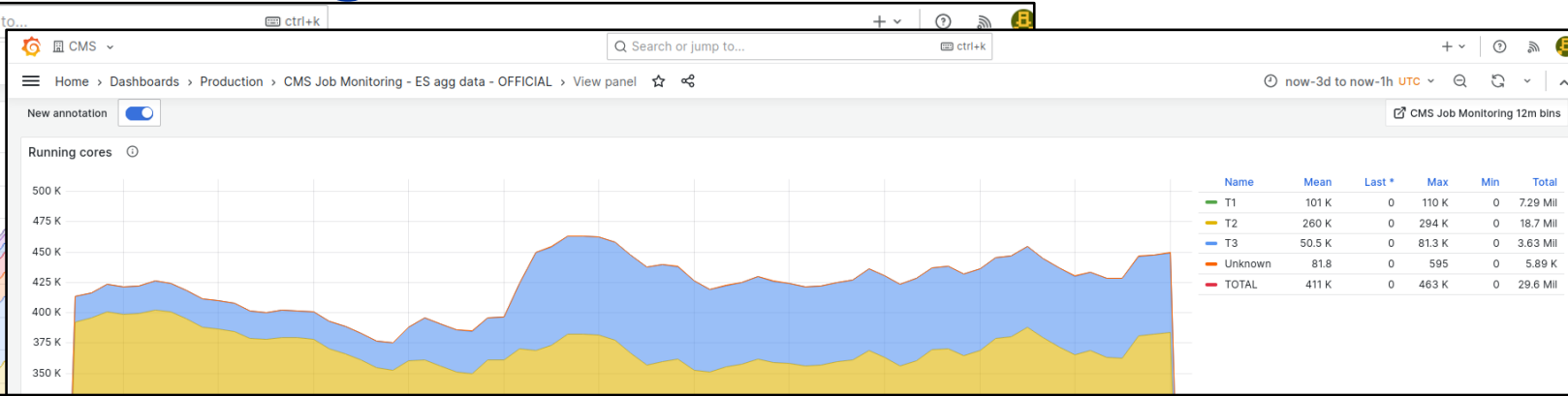
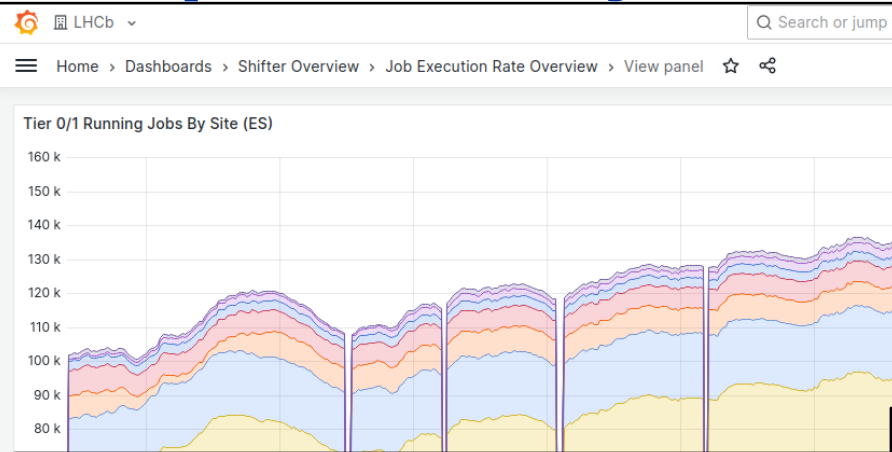
WLCG ▾

- FTS Transfers
- XRootD Transfers
- WLCG Transfers
- WLCG Job Monitoring?**

EXPERIMENTS ▾

- ATLAS Dashboards
- LHCb Dashboards
- CMS Dashboards

Experiment job monitoring



Challenge: WLCG job monitoring

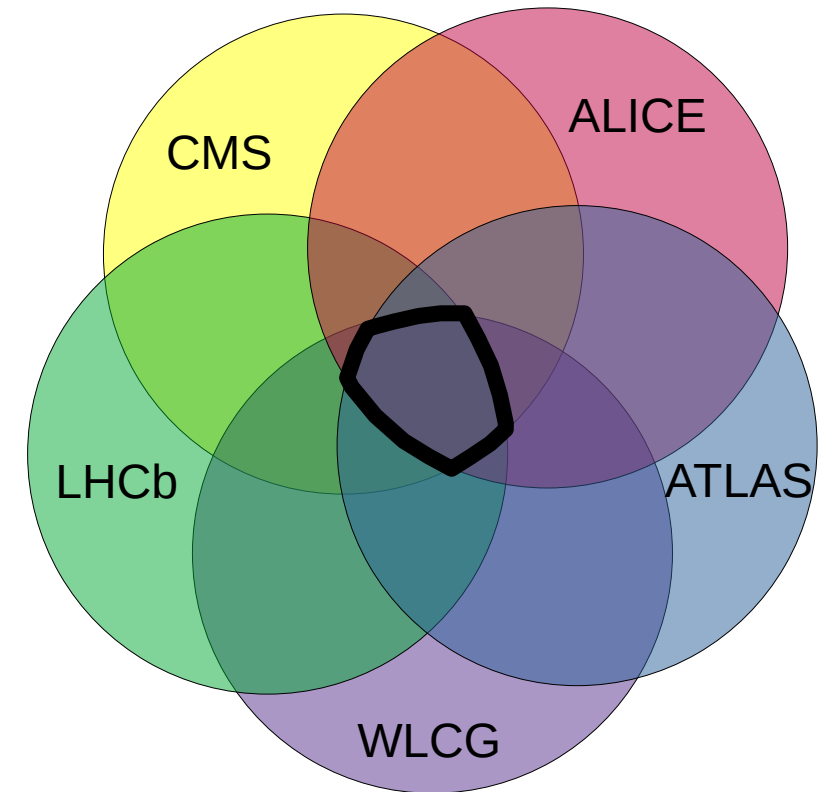
Answering simple questions is not straightforward (and takes time)

- How many CPU cores were used at tier 1 sites during the last year?

The data to answer this question exists!

- Not always available in MONIT
- Comes with a lot of experiment-specific caveats and asterisks
 - Experiments have differing definitions/terms for universal concepts
 - Experiment monitoring follows experiment infrastructure and needs

How to stitch the everything together?



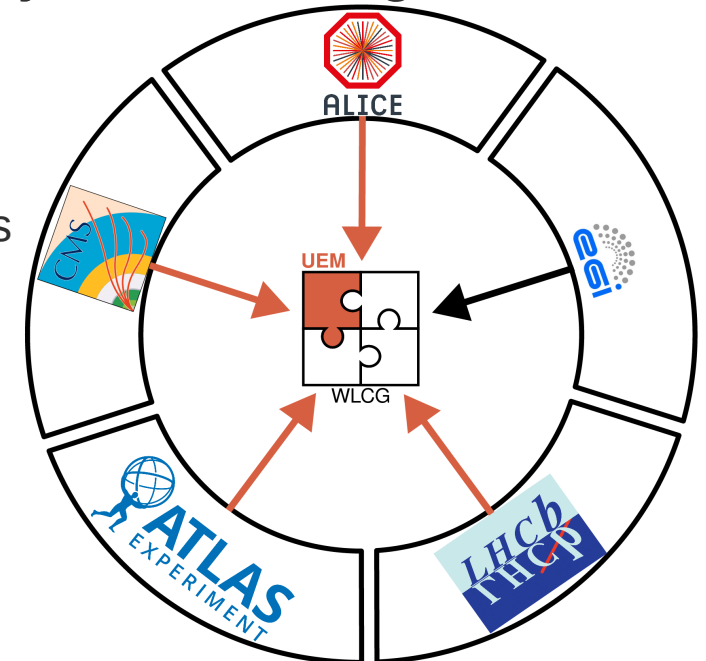
Optimizing WLCG monitoring

A more unified WLCG monitoring approach is necessary

- “A unified WLCG monitoring is a prerequisite for optimization of WLCG operations.” - **LHCC Referee**
- “The interoperability and maintenance of the monitoring tools are seen as critical areas that would benefit of a larger adoption of CERN IT MONIT as unified monitoring infrastructure.” - **WLCG Operations & Coordination**

Unified Experiment Monitoring (UEM) aims to provide unified job monitoring to WLCG

- Adding experiment data to WLCG accounting
 - Help improve data quality of EGI supplemented data
 - Provide overview of the different types of resources used by the experiments
- Not replacing experiment operations monitoring



Goals of the Unified Experiment Monitoring (UEM) project

- **Help experiments migrate/transition to MONIT**
- **Create WLCG job monitoring**
 - 1) Define a list of critical-but-common job monitoring metrics
 - 2) Extract metrics from experiment monitoring infrastructure
 - 3) Publish metrics in unified dashboards
 - 4) Validate dashboards with experiment experts

Job metrics of interest

Metric	Aggregation level
number of running cores	site/tier
number of running jobs	site/tier
wall-clock time	site/tier
wall-clock work	site/tier
number of running cores	activity (Monte Carlo/Analysis/User jobs/...)
number of running jobs	activity
wall-clock time	activity
wall-clock work	activity
number of running cores	resource type (HPC/Grid/public cloud/...)
number of running jobs	resource type
wall-clock time	resource type
wall-clock work	resource type

Migration of LHCb monitoring to MONIT

Connected LHCb data sources to LHCb and WLCG grafana organizations

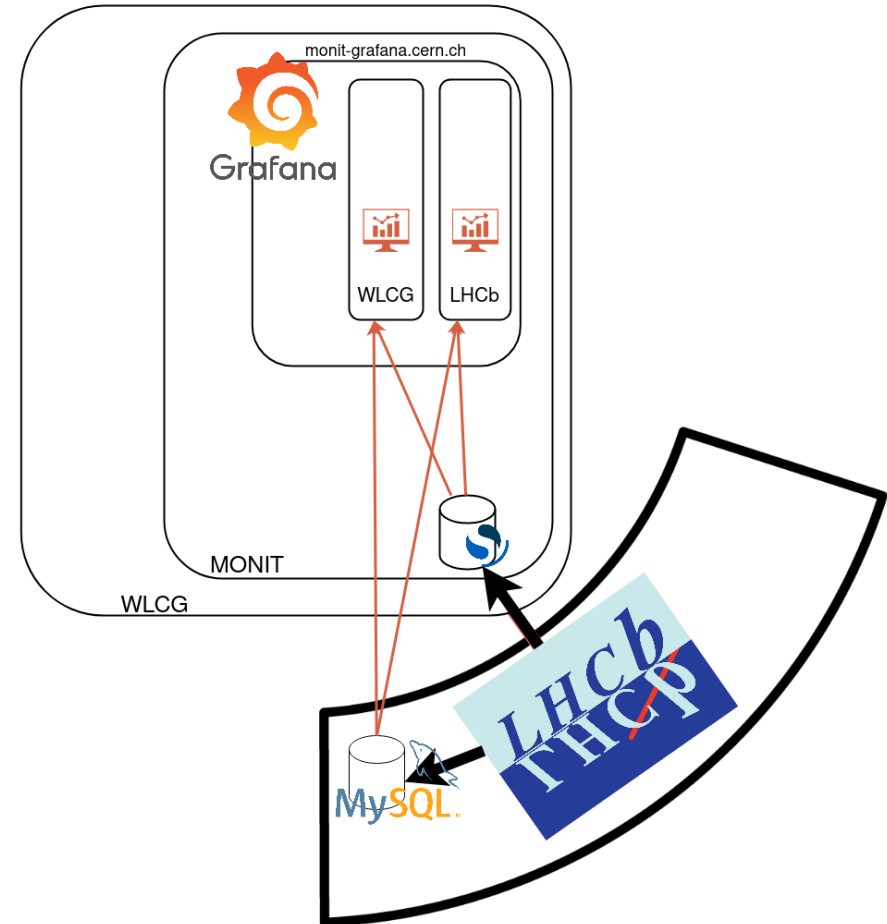
- Opensearch and MySQL instances
- OpenSearch instance managed by MONIT

Migrated LHCb monitoring dashboards

- Dashboards used by shifter rotas

Made LHCb metrics available to WLCG

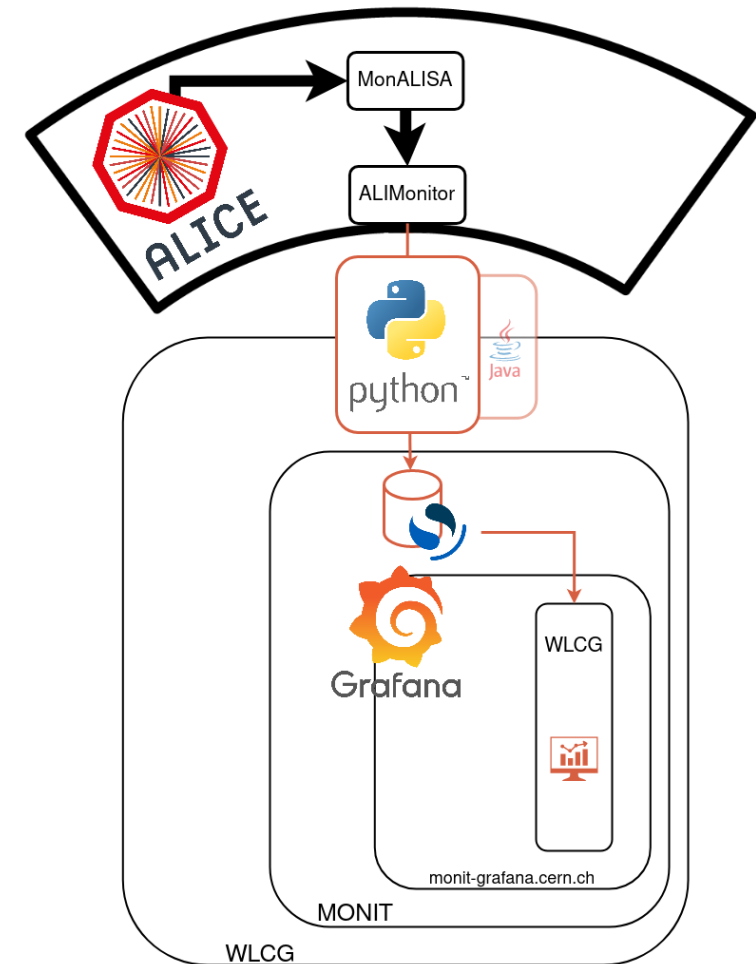
- Already used in Resource Review Board (RRB) reports



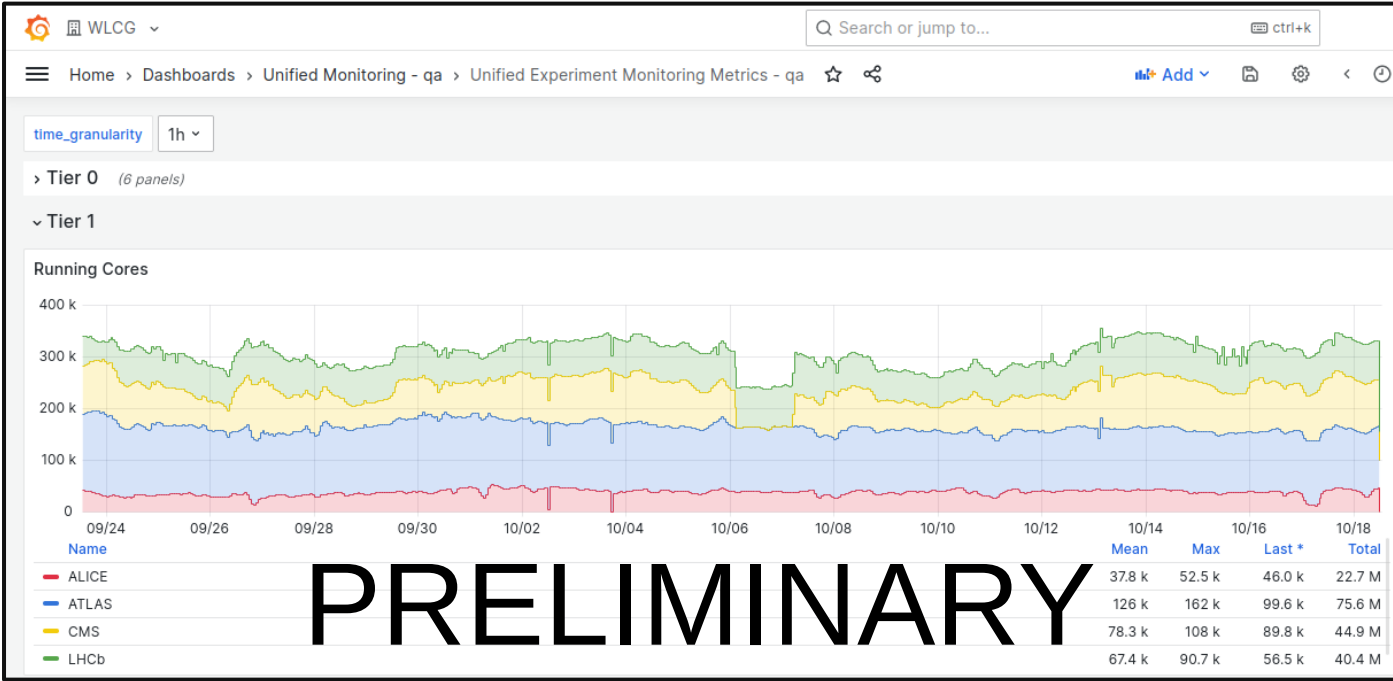
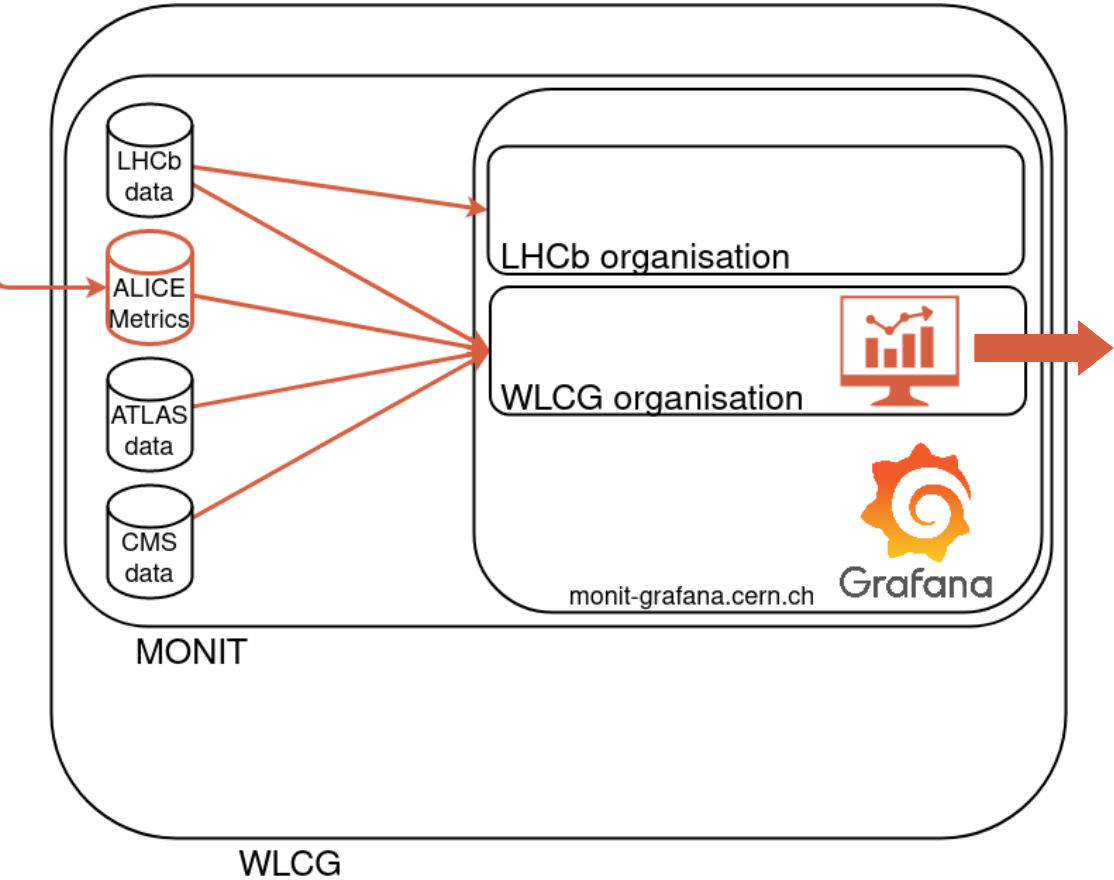
Integration of ALICE monitoring into MONIT

This integration aims to have access to ALICE monitoring data

- Since ALICE will continue using MonALISA
- Navigating constraints from ALICE and WLCG
 - Use existing software, endpoints (ALICE)
 - Keep it as simple as possible (WLCG)
- Solution:
 - Python script to extract metric data from ALIMonitor
 - running every hour as Gitlab pipeline
 - Easy to extend with new metrics
- Data stored in OpenSearch instance managed by MONIT
 - Made available as data source in WLCG Grafana organisation



Unified WLCG job monitoring



Dashboards-as-code?

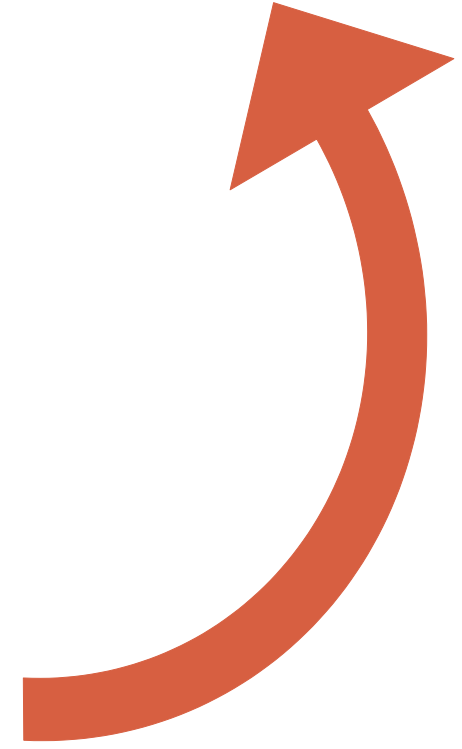
Manual dashboard creation is error-prone (and repetitive)

Version controlling json files is hard

The screenshot shows a dashboard titled 'Running Cores' with a line chart and a data table. The table lists metrics for ALICE, ATLAS, CMS, and LHCb across four time periods. A 'Query options' box is highlighted in red, showing 'MD = auto = 1199' and 'Interval = 1h'. Below the chart, four query cards are shown, each with a red border. The right panel shows configuration options for the chart, with 'Total' selected in the 'Values' section.

	10/11, 00:00	10/11, 12:00	10/12, 00:00	10/12, 12:00	10/13, 00:00	10/13, 12:00	10/14, 00:00	10/14, 12:00	10/15, 00:00	10/15, 12:00	10/16, 00:00	10/16, 12:00	10/17, 00:00	10/17, 12:00
ALICE	38.2 k	45.0 k	43.5 k	6.38 M										
ATLAS	118 k	140 k	20.3 k	19.8 M										
CMS	85.6 k	105 k	99.7 k	14.1 M										
LHCb	70.1 k	90.7 k	53.9 k	11.8 M										

The 'Save dashboard' dialog box shows the title 'Unified Experiment Monitoring Metrics - qa'. The 'Changes' tab is active, showing '109' changes. There are two checkboxes: 'Save current time range as dashboard default' and 'Save current variable values as dashboard default', both of which are unchecked. A text area contains the message 'adding minimum to legend of Running Cores plot for Tier 1.'. At the bottom, there are 'Cancel' and 'Save' buttons.

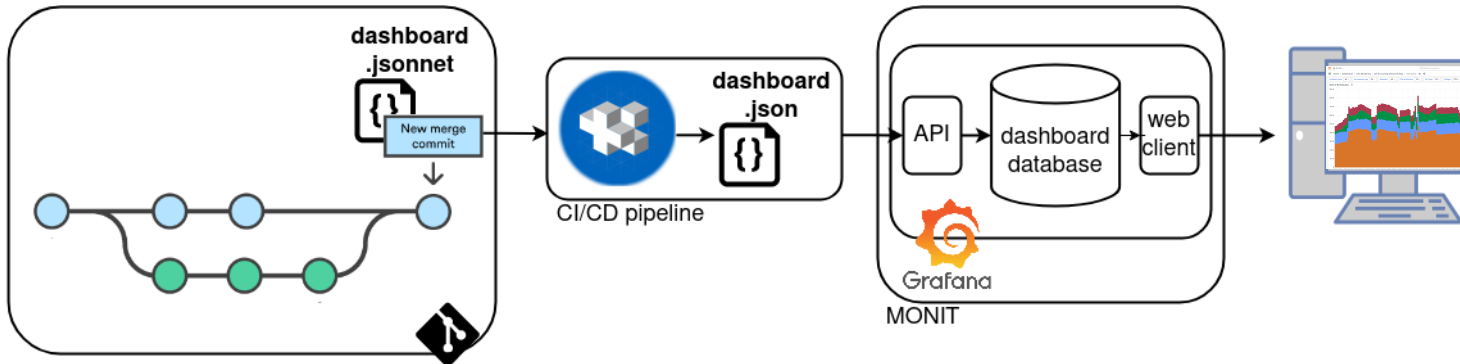


Dashboards-as-code!

Bind CLI-provided arguments to variables that can be used in dashboard

Jsonnet and grafana extension *grafonnet* to the rescue!

- Configuration language and extension of json
 - configuration file to store
 - data sources used by plots
 - frequently used queries
 - Style sheets for dashboards
 - Functions, variables, and OOP concepts
- Output is json



```

local g = import './lib/g.libsonnet';
local row = g.panel.row;

local timeseries = import './lib/panels/timeseries.libsonnet';
local piechart = import './lib/panels/piechart.libsonnet';

local variables = import './lib/variables.libsonnet';

local tag = std.extVar('tag');
local id = std.extVar('id');
local url = std.extVar('url');
local folderUid = std.extVar('folderUid');
local overwrite = std.extVar('overwrite');

{
  "dashboard": g.dashboard.new('Unified Experiment Monitoring Metrics - ' + tag)
    + g.dashboard.withDescription('Description ' + id + ':' + url)
    + g.dashboard.withTags(['grafonnet', 'NOT validated', tag, id])
    + g.dashboard.time.withFrom('now-1y')
    + g.dashboard.graphTooltip.withSharedCrosshair()
    + g.dashboard.withVariables([
      variables.granularity,
      variables.division_factor,
      variables.egi_datasource,
    ])
    + g.dashboard.withPanels([
      row.new('Tier 0')
      + row.withCollapsed(true)
      + row.withPanels(g.util.grid.wrapPanels([
        timeseries.all_cores_0,
        piechart.all_cores_0,
        timeseries.all_jobs_0,
        piechart.all_jobs_0,
        timeseries.all_wallclock_0,
        piechart.all_wallclock_0,
      ])),
      row.new('Tier 1')
      + row.withCollapsed(true)
      + row.withPanels(g.util.grid.wrapPanels([
        timeseries.all_cores_1,
        piechart.all_cores_1,
        timeseries.all_jobs_1,
        piechart.all_jobs_1,
        timeseries.all_wallclock_1,
        piechart.all_wallclock_1,
      ])),
      row.new('Tier 2')
      + row.withCollapsed(true)
      + row.withPanels(g.util.grid.wrapPanels([
        timeseries.all_cores_2,
        piechart.all_cores_2,
        timeseries.all_jobs_2,
        piechart.all_jobs_2,
        timeseries.all_wallclock_2,
        piechart.all_wallclock_2,
      ])),
    ]),
}
    
```

Import modules that define style of time series and pie chart panels

Add list of predefined panels to row 'Tier-0'

Validation and next steps

In contact with experiment-experts to validate our approach

- Correctly interpreting the data
- Creating correct queries

Use Jsonnet for other WLCG dashboards

Add metrics aggregated by activities

- Need to define common 'activities'

Conclusion

UEM is adding unified job monitoring to MONIT

- To help improve WLCG data quality
- To provide an overview of the experiment resources

Validated unified job monitoring dashboards will be published in 2025

Jsonnet/grafonnet are life-savers when creating monitoring dashboards



home.cern