

Cloud-Native Graphical and Programmatic Cross-Database Access to Experiment Metadata with Trino, Superset, and Ibis

Eli Mizrachi (they/them)  1,2

on behalf of the LZ Collaboration

October 20, 2026

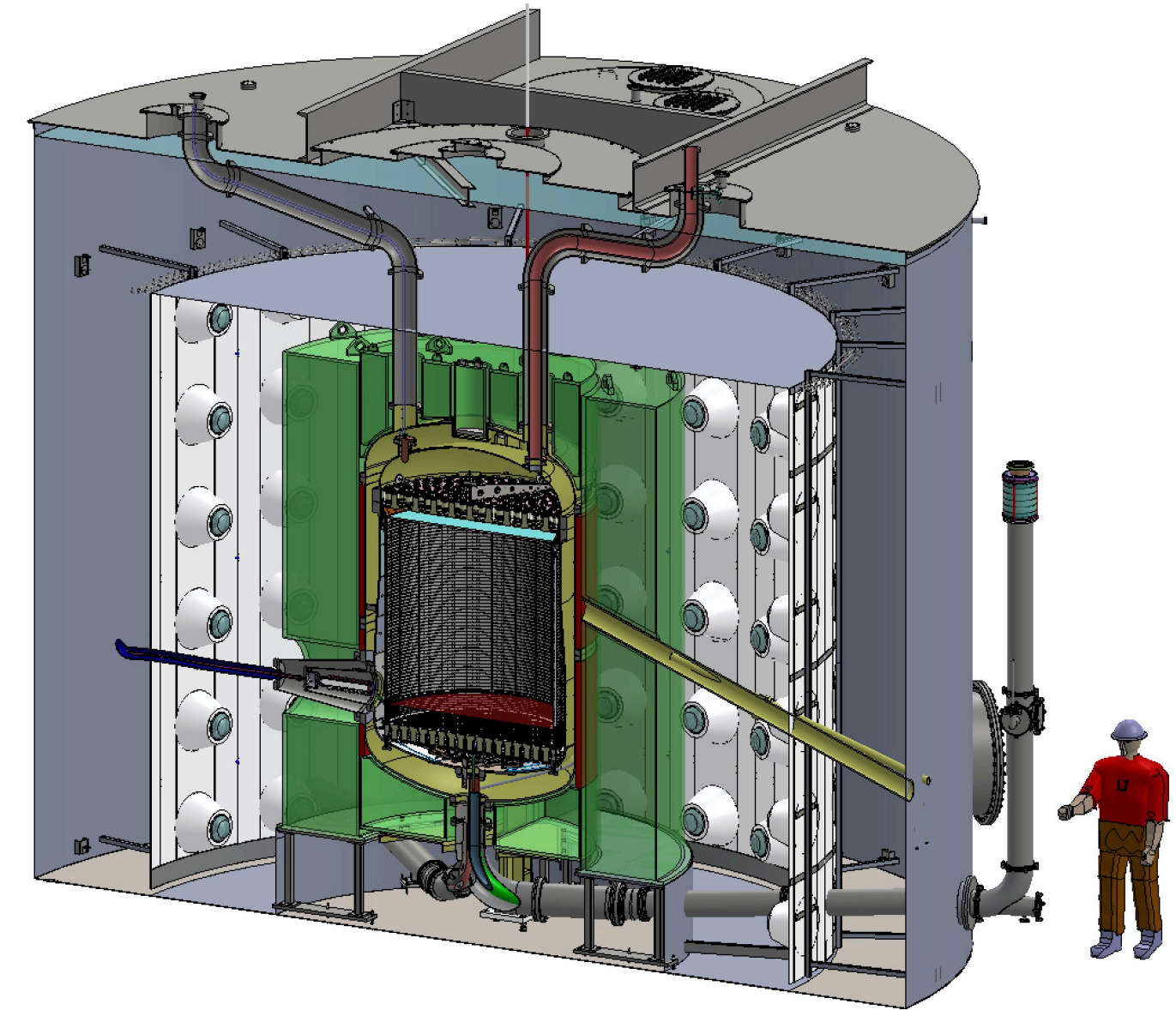
¹ SLAC National Accelerator Laboratory

² Kavli Institute for Particle Astrophysics and Cosmology

The LZ Experiment

- Dark matter search with dual-phase xenon time projection chamber (TPC)
- 250+ scientists, engineers, technicians
- 39 collaborating institutions
- ~10 PB of data produced since 2020
- Few TB of metadata related to detector conditions and data processing environments ([M. Arthurs et al. EPJ Web Conf. 337, 01232](#))¹

1. See also: Maris's talk on [AI-assisted knowledge retrieval](#)!

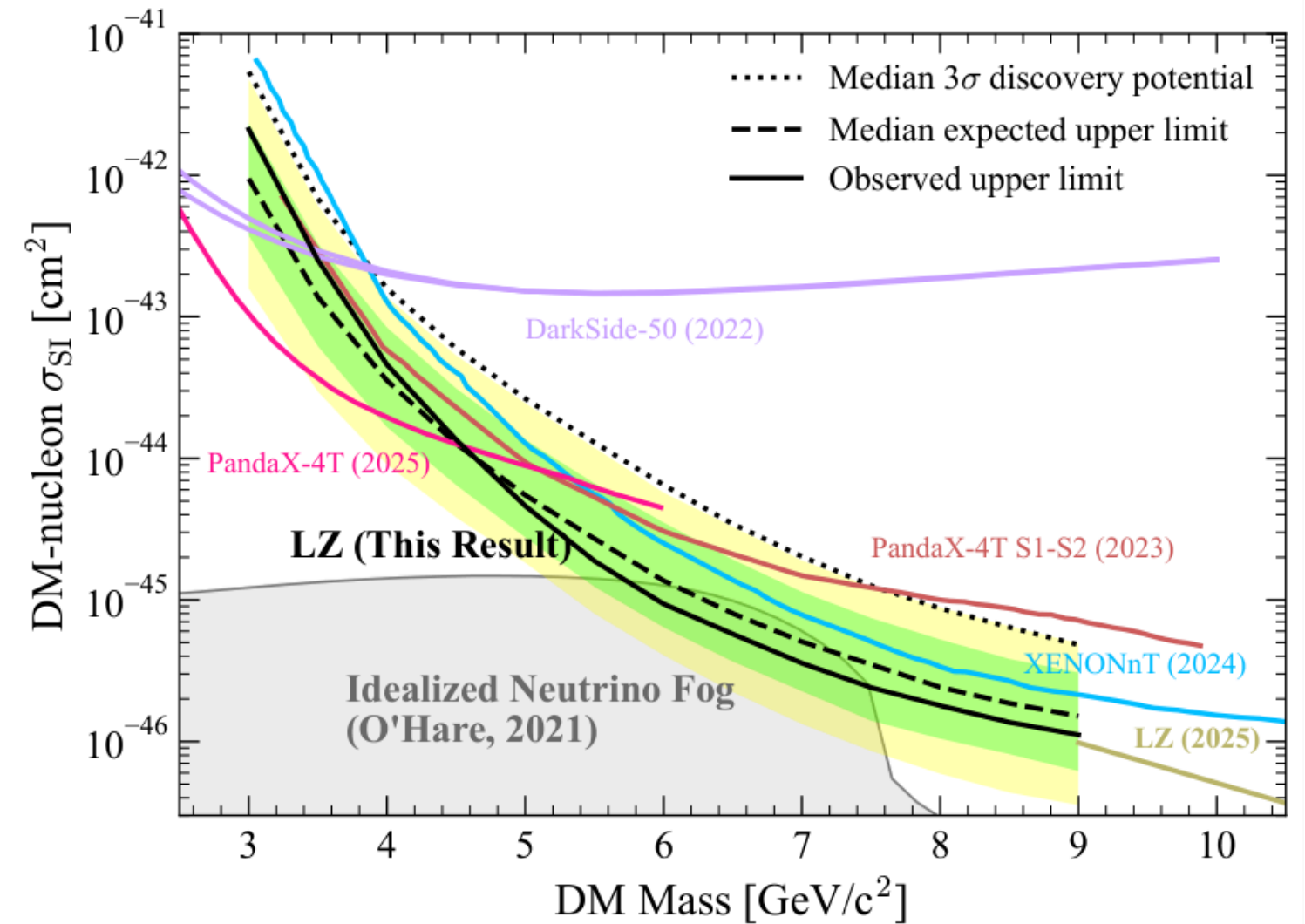


Rendering of major LZ subsystems
[D. S. Akerib et al. NIM-A 953, 163047](#)

The LZ Experiment

- Dark matter search with dual-phase xenon time projection chamber (TPC)
- 250+ Scientists, Engineers, Technicians
- 39 Collaborating institutions
- ~10 PB of data produced since 2020
 - Moved and processed with automated pipelines ([M. Arthurs et al. EPJ Web Conf. 337, 01232](#))¹
- Few TB of metadata related to detector conditions and data processing environments

1. See also: Maris's talk on [AI-assisted knowledge retrieval](#)!



World-leading limits on low-mass WIMPS

[D. S. Akerib et al. arXiv:2512.08065](#)

LZ Databases and Interfaces

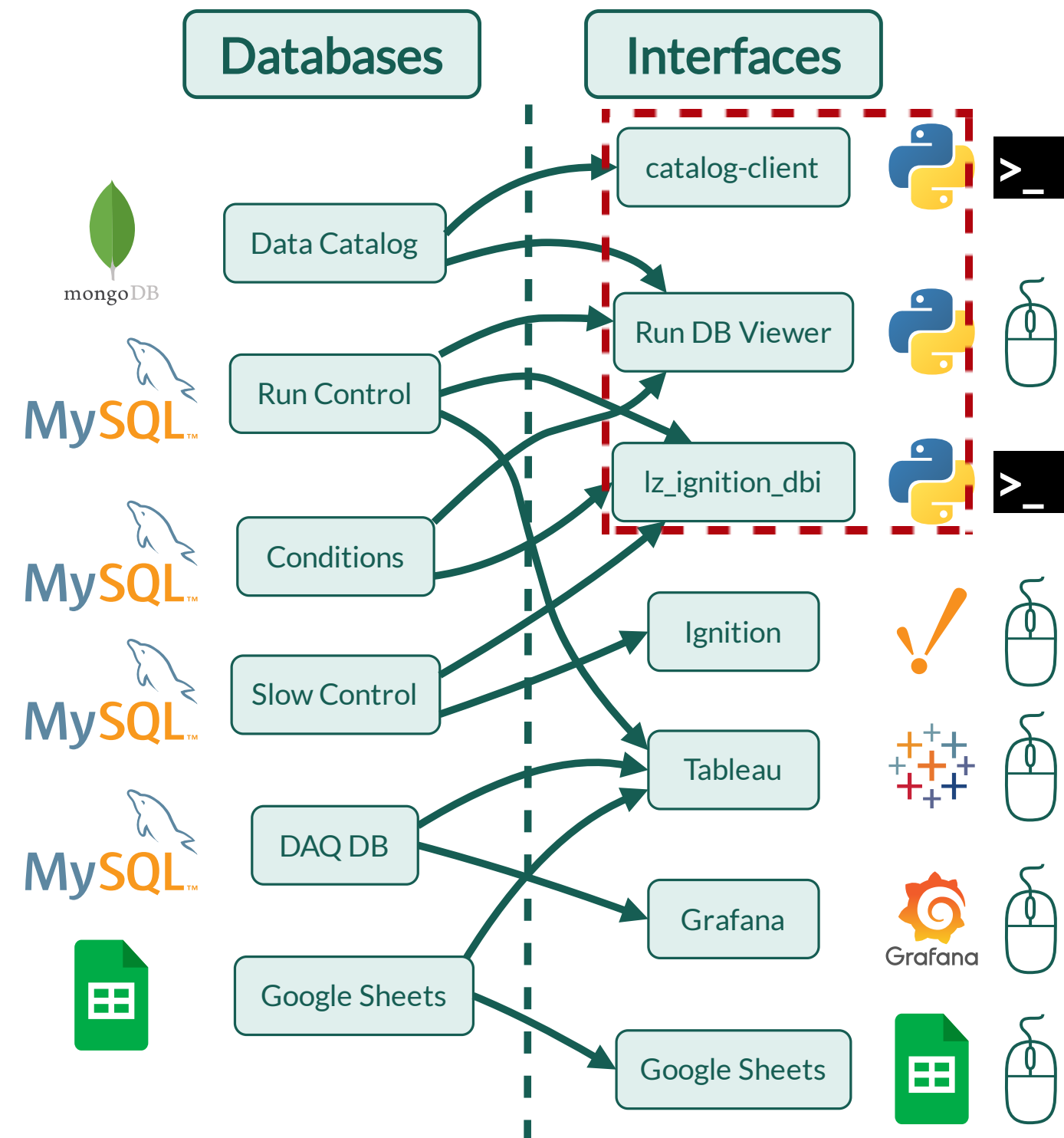
Our existing interfaces are

 **Programmatic** |  **Graphical**

- Our python interfaces are
 - Intensive to deploy and maintain
 - Difficult to extend to other databases

With open-source tools we can:

- Generalize **ACCESS**
- Boost **PERFORMANCE**
- Streamline **OPERATIONS**



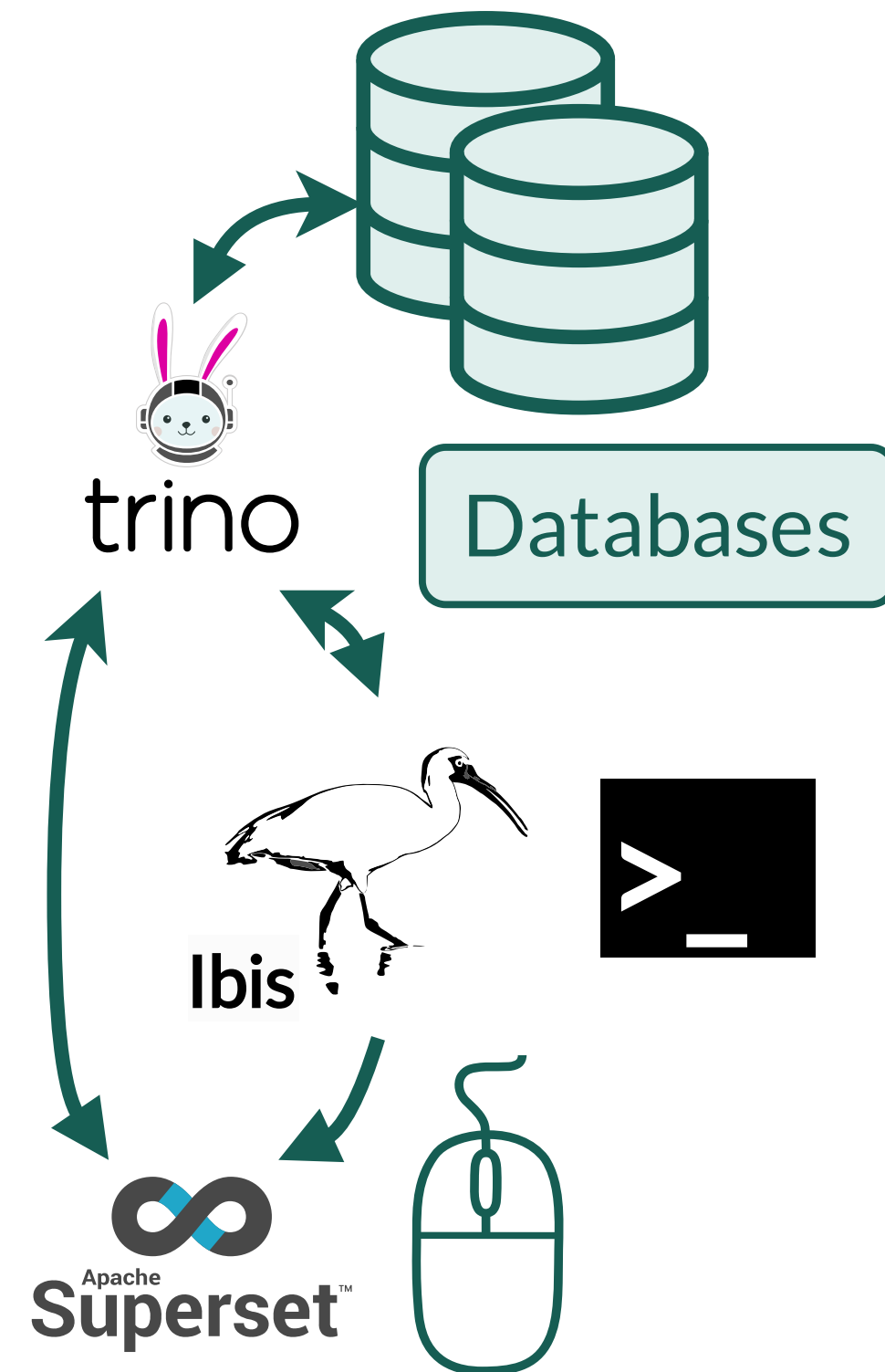
Generalizing Access

Trino, Ibis, and Superset

ACCESS

- **Trino**: Distributed SQL query engine
 - Query, `JOIN`, etc. across databases (DBs)
 - Couple to 30+ types of SQL/NoSQL DBs
- **Ibis**: Compose SQL queries in python with dataframe-like interface
 - Couple to 10+ types of backends
- **Superset**: Interactive dashboard builder
 - Source from DB tables or `ibis` queries
- All three can write back into DBs

Unified python and graphical support
for cross-database queries

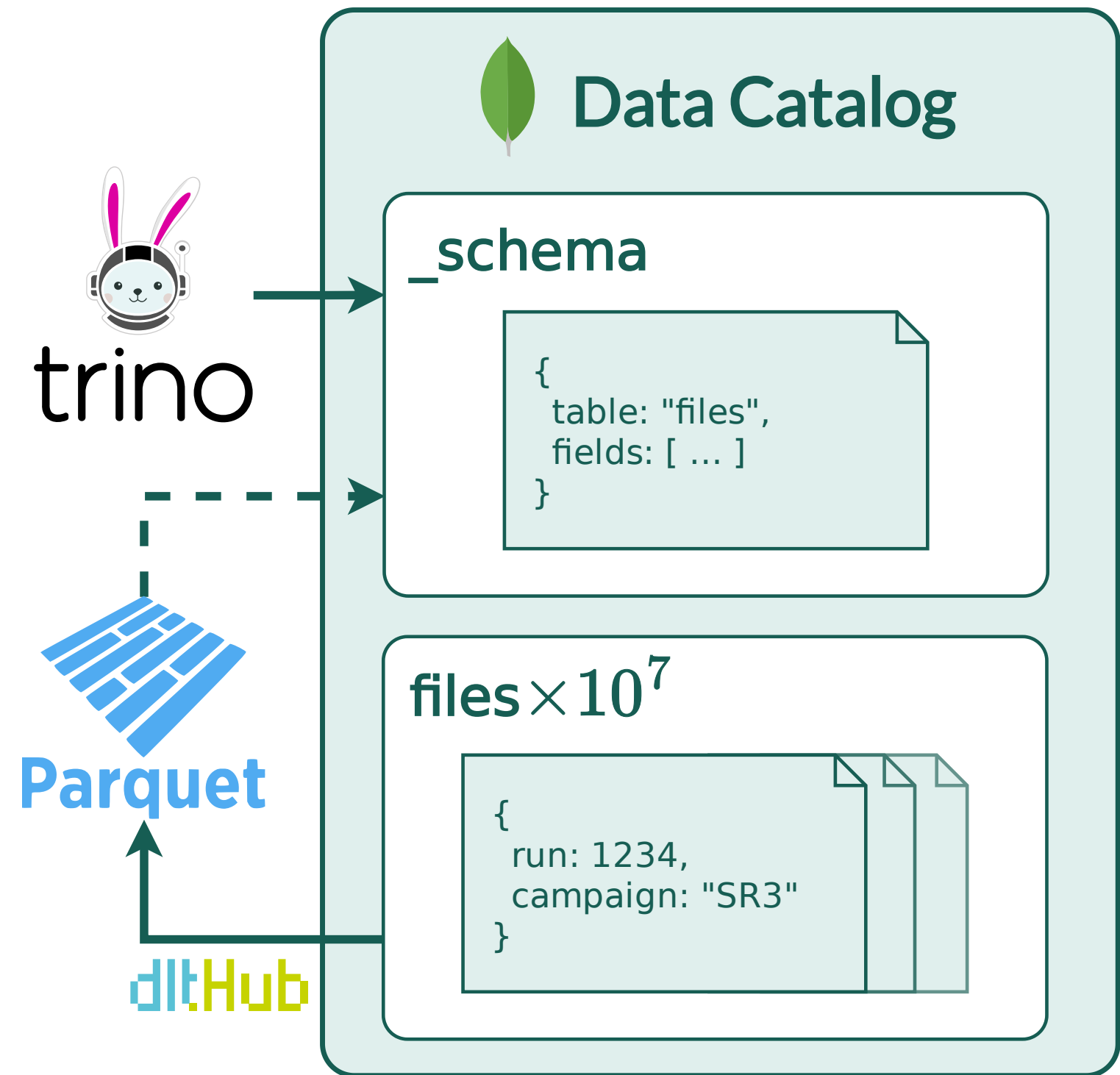


Trino and MongoDB

ACCESS

How does Trino talk to MongoDB?

- Schema inferred from a document *sample* on first connection
 - Then stored as a document in a `_schema` collection
- Extract `files` collection to Parquet with `dlt` to obtain *full* schema
- Manually insert `dlt`-derived schema into `_schema`



Production Status

ACCESS

PERFORMANCE

BEFORE

- Run Control: data acquired
- Data Catalog: data processed
- Manually compare both tables to diagnose data pipeline failures
- Faulty cache → Data Catalog tab *silently* out-of-date

App required daily-to-weekly manual restarts

Run DB Viewer

Run DB Viewer

Default displays the LATEST LZap production. Use search filters to see earlier productions. All times Mountain.

Run	All productions	Production	Cond Ver	Prod Date	Proc. Evts	Tot. Evts	No. files
16337	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T05:09:35	300000	300000	120
16336	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:43:13	300000	300000	120
16335	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:46:54	300000	300000	120
16334	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:48:22	300000	300000	120
16333	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:14:10	300000	300000	120

Old LZ Run DB Viewer

Production Status

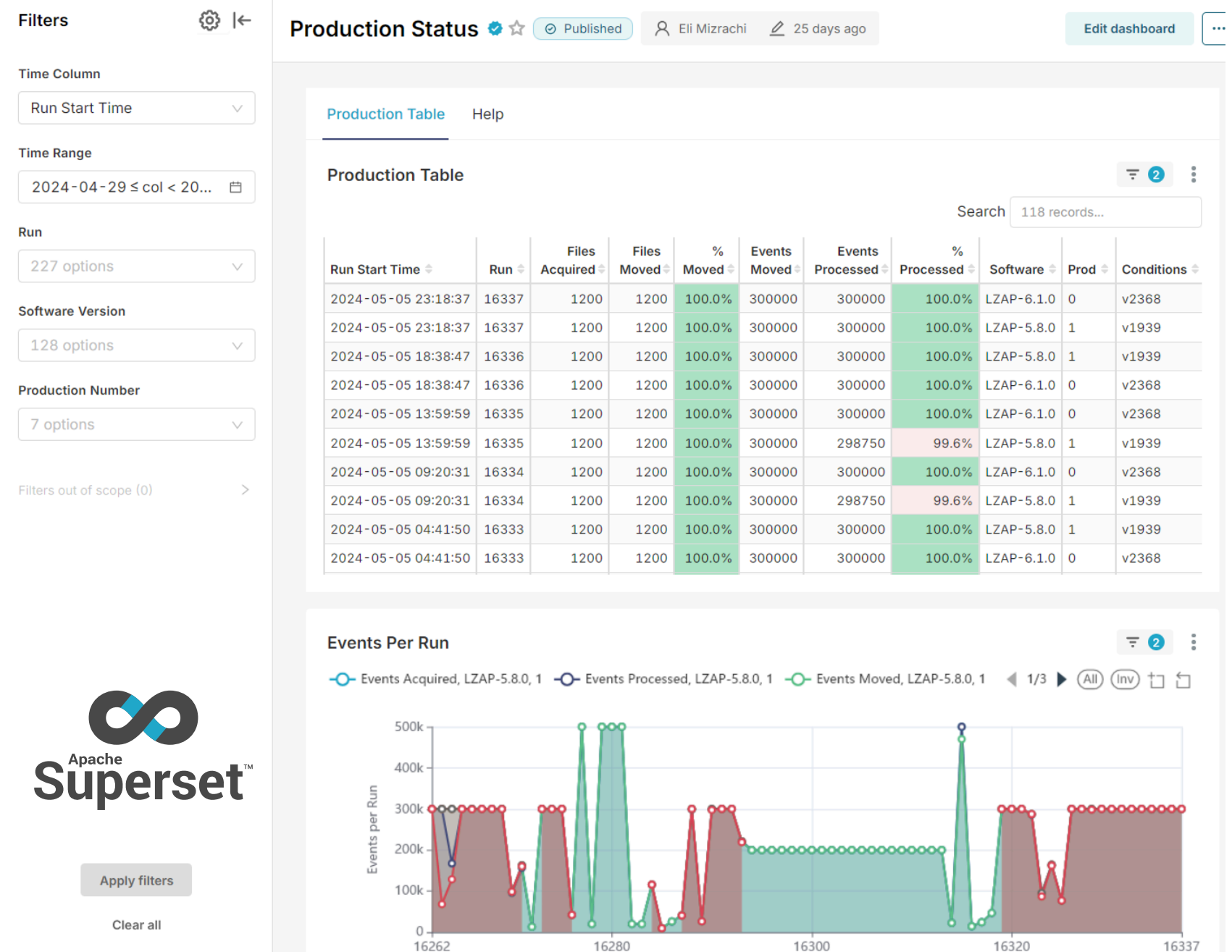
ACCESS

PERFORMANCE

AFTER

- Results cached with Redis
- Dashboard uses single `ibis` SQL query
 - `JOIN` Run Control (MySQL) and Data Catalog (MongoDB) tables with Trino
 - Reuse query in Python reports

- 12s → 4s loading time
- No manual restarts



New Production Status dashboard in Superset

Expressive Queries

[ACCESS](#)

What are the newest reprocessings for a list of runs?

BEFORE

- Custom `catalog-client` package
- User-dependent post-processing

AFTER

- Single `ibis` query on `files` table
 - Group by `(run, software, production)`
 - Rank `(software, production)` per `run`; 0 is newest

```
1 t_files.filter((_.run.isin(run_list)) & (_.usecase == "DataRQ"))
2 .group_by([_.run, _.software_versions, _.production_number])
3 .agg(files=_.run.count())
4 .mutate(
5     rank=ibis.dense_rank().over(
6         group_by=[_.run],
7         order_by=[_.software_versions.desc(),
8                 _.production_number.desc()]
9     ),
10 ).to_pandas() # deferred execution!
```

run	software_versions	production_number	files	rank
18781	LZAP-6.2.1	2	30	0
18781	LZAP-5.8.0	1	60	1
18777	LZAP-6.2.1	2	30	0
18777	LZAP-5.8.0	1	60	1

`ibis` query and result (with aggregation)

Boosting Performance

Data Catalog Tag Index

PERFORMANCE

What sources have we calibrated with?

BEFORE

- Distinct categorical values in Data Catalog required full table scan
- Expensive, 40s query

Lack of feedback indistinguishable from silent failures

 Run DB Viewer

Data

Simulation

Run Control

Data Catalog

Conditions

Refresh

Catalog Details

Default displays the LATEST LZap production. Use search filters to see earlier productions.

All times Mountain.

Search

Run

16262-16337

Populate search parameters

LZap Production

Op Name

Campaign

Source

Submit

Clear Search

	Run	All productions	Production	Cond Ver	Prod Date	Proc. Evts	Tot. Evts	No. files
☐	16337	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T05:09:35	300000	300000	120
☐	16336	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:43:13	300000	300000	120
☐	16335	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:46:54	300000	300000	120
☐	16334	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:48:22	300000	300000	120
☐	16333	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:14:10	300000	300000	120

Data Catalog Tag Index

PERFORMANCE

AFTER

- Pre-aggregated table of categorical values in Data Catalog DB

- Single `ibis` query

```
1 small_table = big_table.groupby(...)
2 db_connection.create_table(small_table)
```

- Rows form distinct data subsets:

*How many files from X campaign
AND with Y source?*

Data Catalog Tag Index

Search 203 records...

Files	Software	Prod	Use Case	Source	Campaign	Conditions	Op Name
15152	LZAP-5.4.7	1	DataRQ	Cf252	SR1	v856	Cf252 OD + TPC Scan
12049	LZAP-5.4.7	1	DataRQ	Rn220	SR1	v857	Rn220 Calib
10947	LZAP-5.4.5	1	DataRQ	Amli	SR1	v852	N/A
7053	LZAP-5.5.1	0	DataRQ	Amli	SR1	v968	N/A
6634	LZAP-5.4.6	0	DataRQ	Amli	SR1	v852	N/A



Materialized daily
with CI/CD

Query time reduced

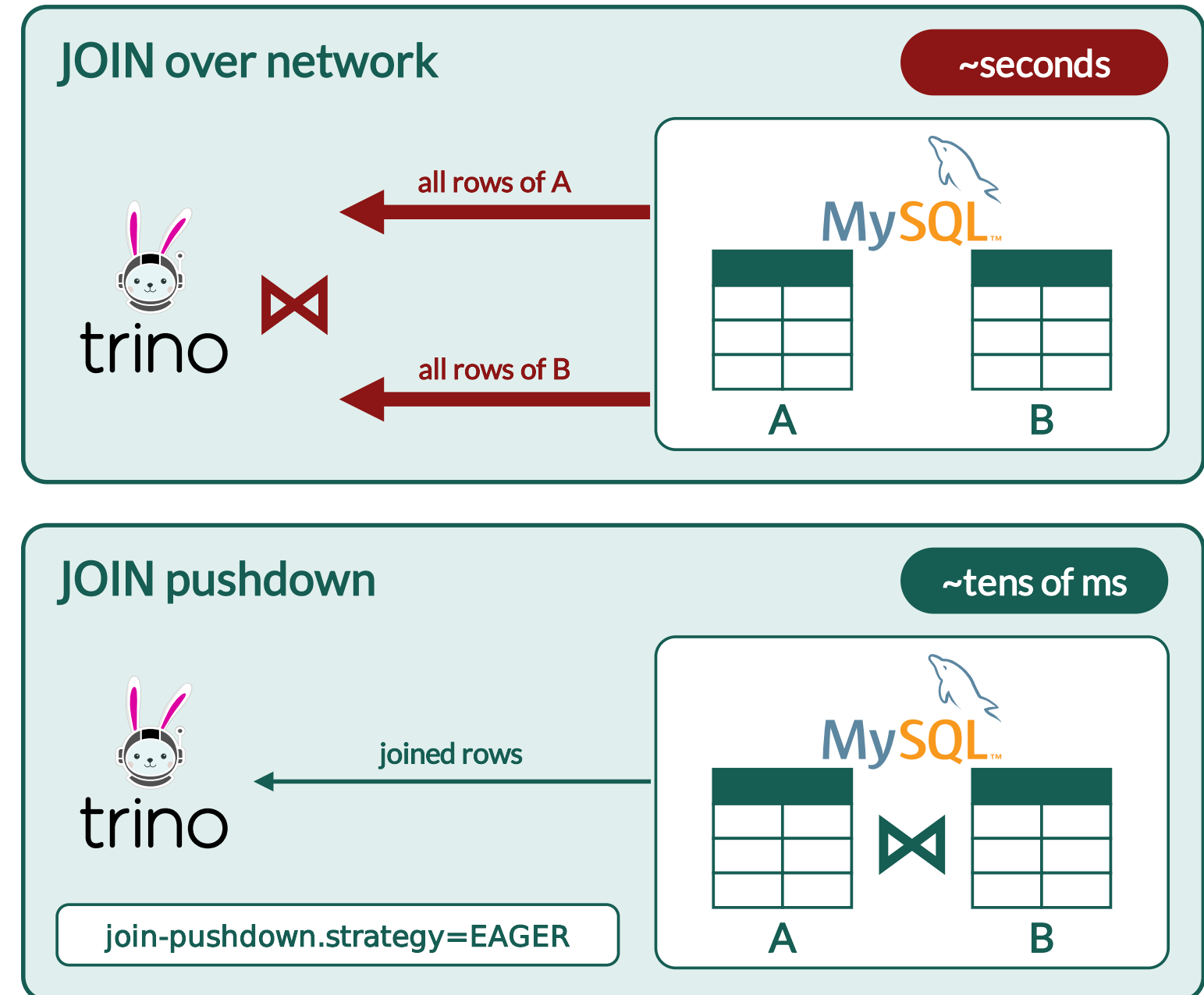
40s → 100ms

Trino Caveats

PERFORMANCE

- Expect query overhead of 50–100 ms
- **JOINS** are not always pushed to DBs
 - Difference of seconds vs tens of ms even for small tables
- Expect incompatibilities between Ibis→Trino→DBs
 - e.g. more chances to skip indexes

Consider a dedicated analytical store for best performance



Streamlined Operations

Application Deployment

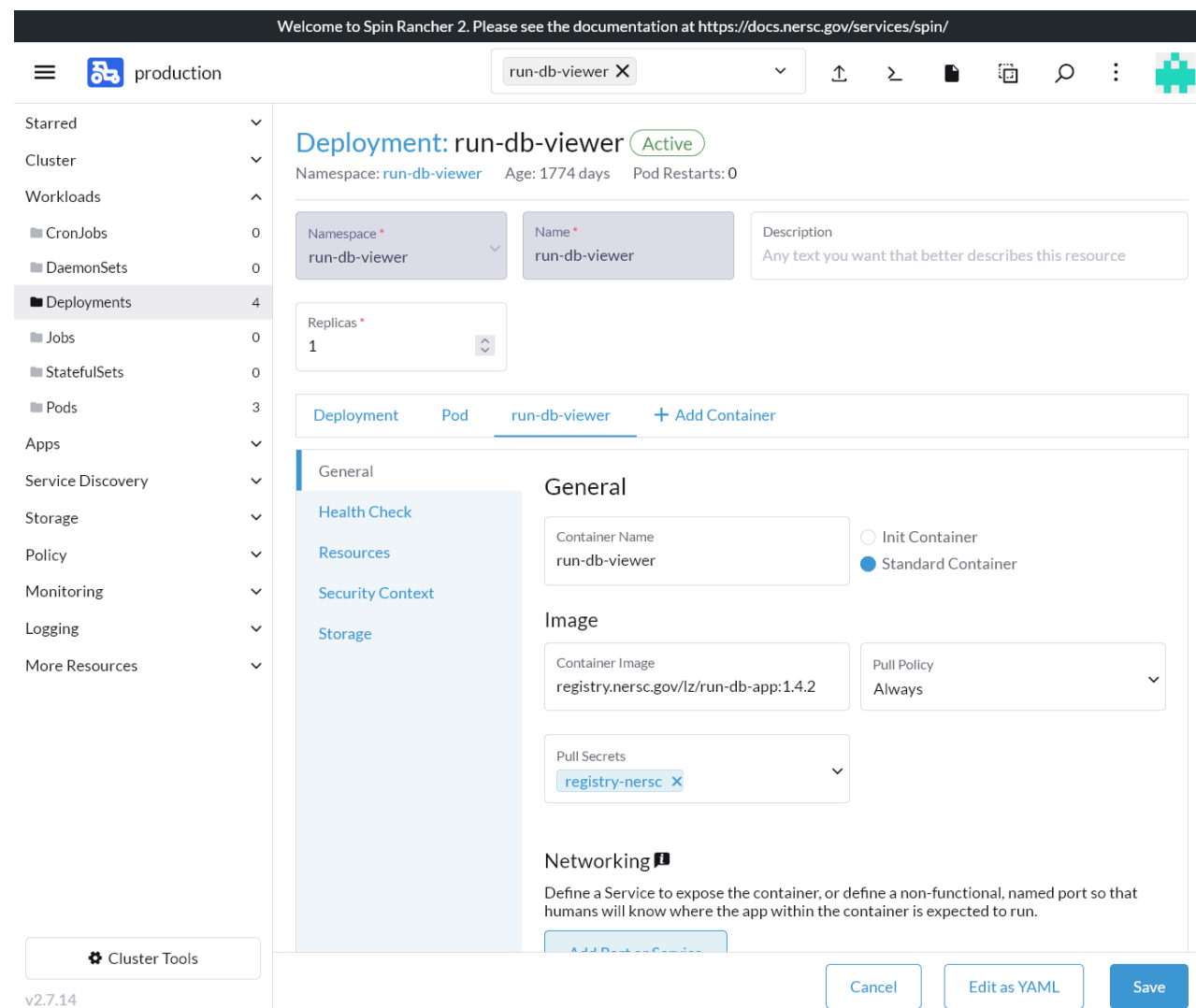
OPS

BEFORE

AFTER

Configured in Rancher (**k8s**) GUI by hand

Version controlled with **Helm**, **Helmfile**



```

1 environments:
2   prod:
3     values:
4       - kubeContext: lz-trino
5         namespace: lz-trino
6         cluster: production
7   ---
8 releases:
9   - name: trino
10    chart: ./trino
11    kubeContext: "{{ .Values.kubeContext }}"
12    namespace: "{{ .Values.namespace }}"
13    labels:
14      env: "{{ .Environment.Name }}"
15    values:
16      - values/lz-trino.yaml.gotmpl
17    secrets:
18      - secrets/lz-trino.vaml

```

Application Deployment

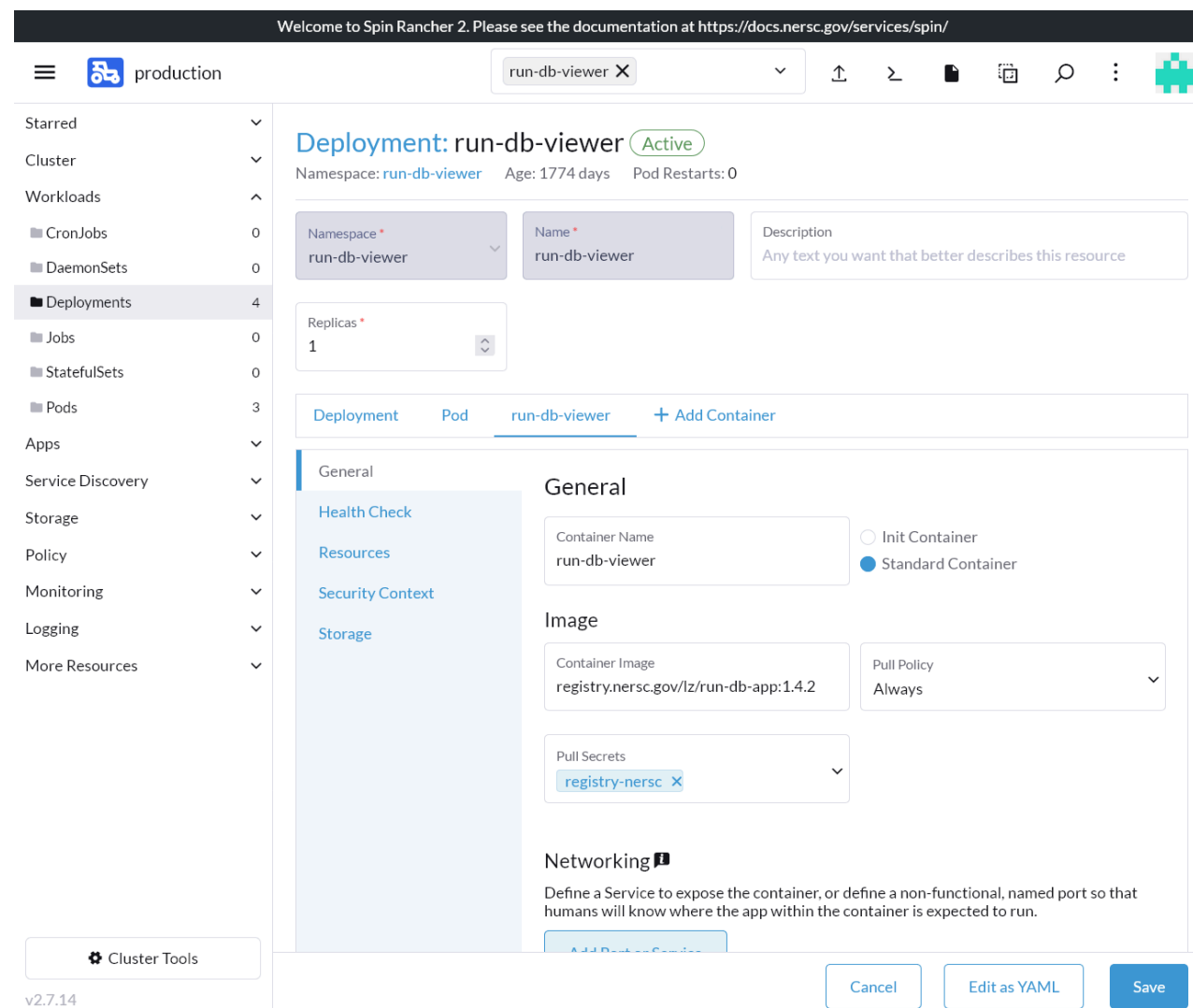
OPS

BEFORE

AFTER

Configured in Rancher ([k8s](#)) GUI by hand

Version controlled with [Helm](#), [Helmfile](#)



```
1 helmfile apply \
2   --file lz-trino/helmfile.yaml.gotmpl \
3   -e prod
```

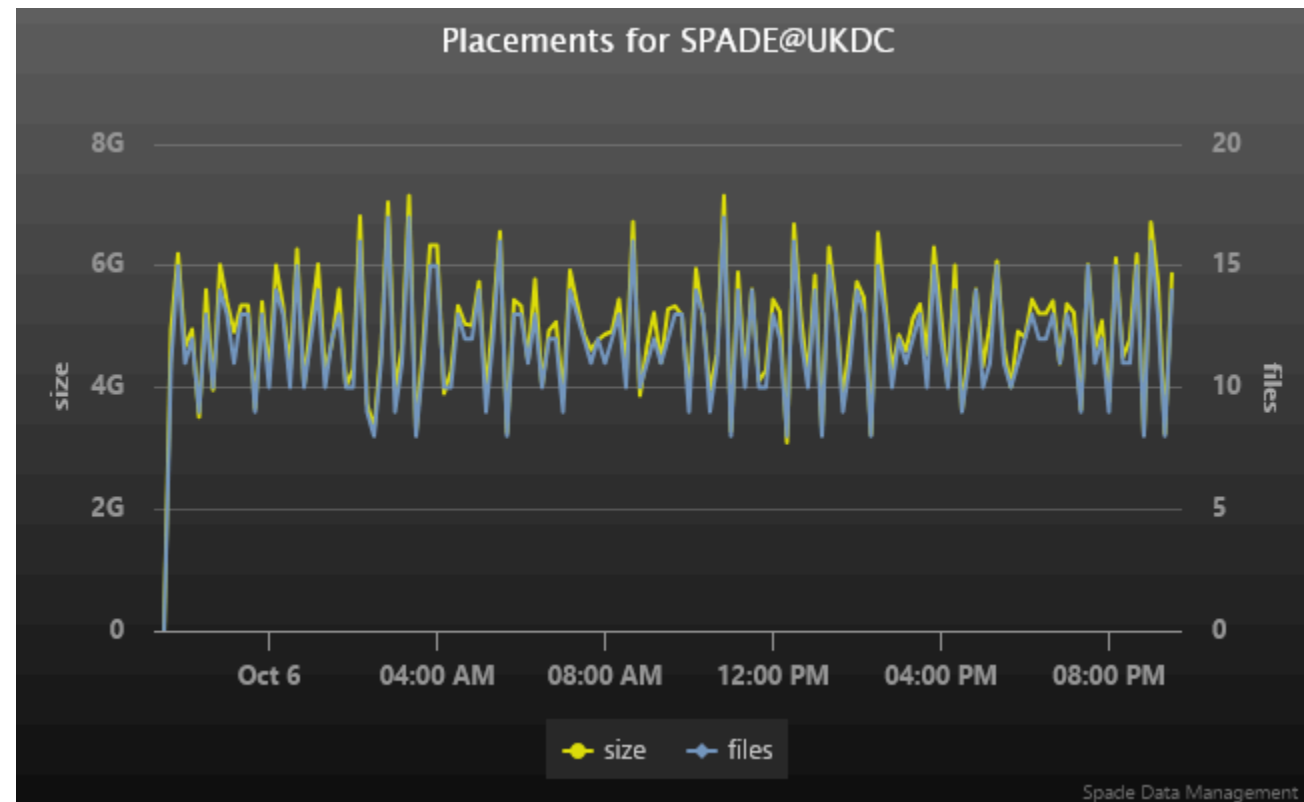
- Similar initial time investment
- Secrets encrypted *in* git with [SOPS](#)
- Redeploy in minutes
- Reproducible and portable

Dashboard Deployment

ACCESS

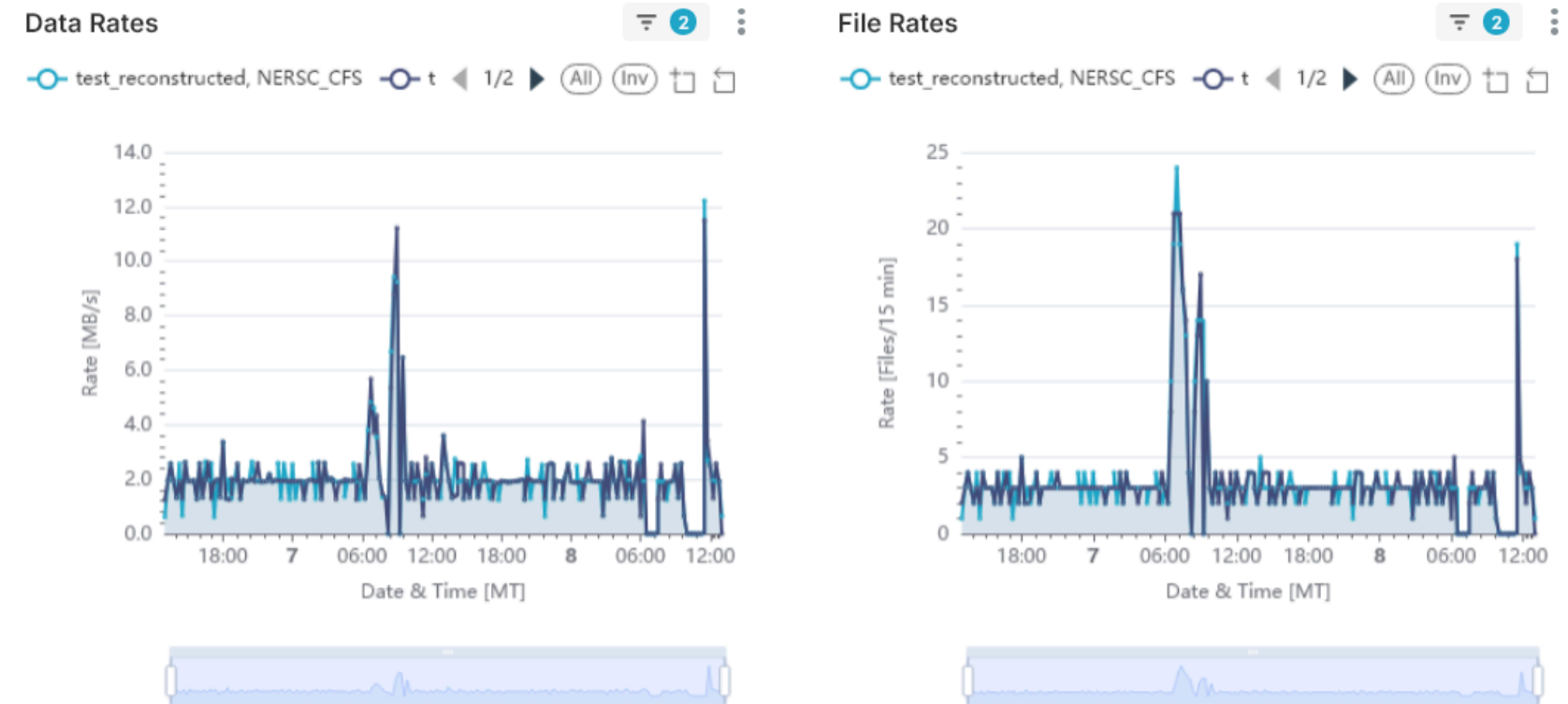
OPS

BEFORE



Old **SPADE** Monitor Python app

AFTER



New **Rucio** dashboard in Superset

No new app for new data movement system
Just a new dashboard

Conclusion

Future Work

Query Repository

- Migrate queries from `lz_ignition_dbi` to generic package based on `ibis`
- Derived tables refreshed by CI

Conditions Dashboards

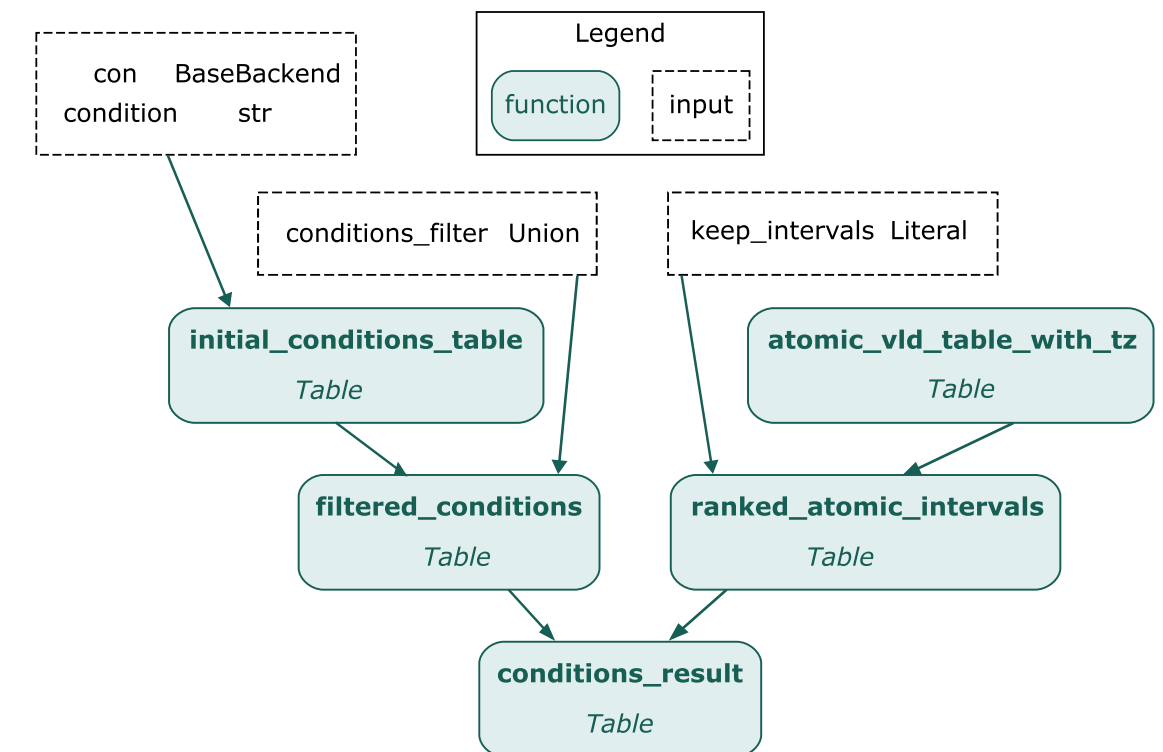
Extend Superset to cover detector conditions

Spreadsheet Migration

Upload critical spreadsheets into databases

Consolidated Run Information

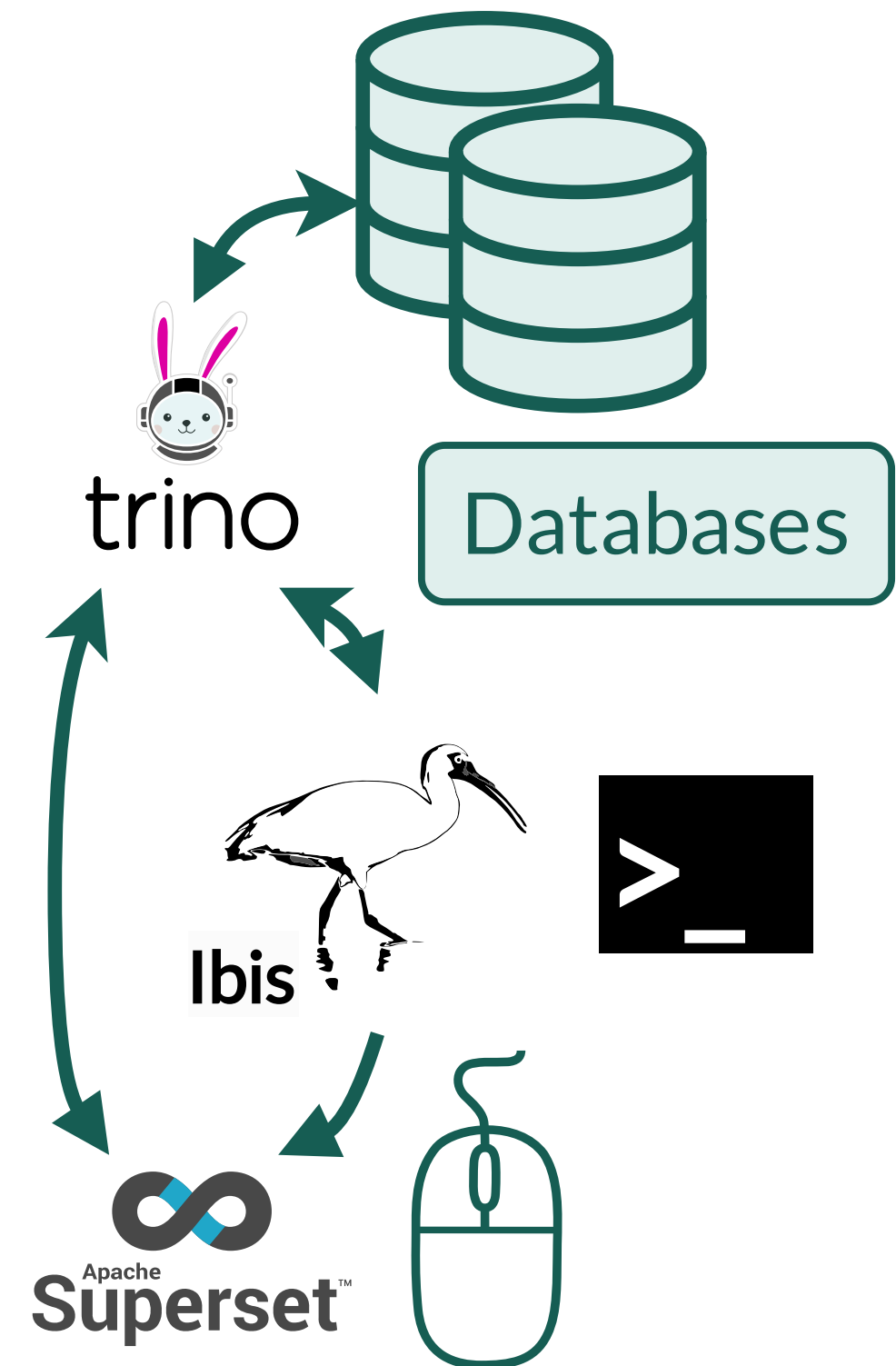
- Deploy dedicated analytical store (Clickhouse?)
- Create denormalized tables of run-level information



Trialing `Hamilton` for lineage: partial conditions query as a dataflow

Summary

- Generalized DB access with Trino, Superset, `ibis`
- Suitable for generic queries, but specialized features in databases might not work
- Deprecated Run DB Viewer and SPADE Monitor applications
- More informative Production Status dashboard
 - Loads ~3× faster than old app
- >100× speedup from pre-aggregated Tag Index
- Version-controlled deployments with `helmfile`, `sops`



Summary

One of the grand challenges of data-intensive science...is to improve knowledge discovery through assisting both humans, and their computational agents in the access to...scientific data.

2016 paper on Findability Accessibility, Interoperability, Reusability (FAIR) principles ([M. D. Wilkinson et al. Sci Data 3, 160018](#))

Acknowledgments

UNITED STATES

Black Hills State University	University at Albany, SUNY
Brookhaven National Laboratory	University of Alabama
Brown University	University of California Berkeley
Fermi National Accelerator Laboratory	University of California Davis
Lawrence Berkeley National Laboratory	University of California Los Angeles
Lawrence Livermore National Laboratory	University of California Santa Barbara
Northwestern University	University of Maryland
Pennsylvania State University	University of Massachusetts, Amherst
Purdue University	University of Michigan
SLAC National Accelerator Laboratory	University of Rochester
South Dakota School of Mines & Technology	University of Texas at Austin
South Dakota Science & Technology Authority	University of Wisconsin, Madison
Texas A&M University	

EUROPE

Imperial College London	University of Bristol
King's College London	University of Edinburgh
LIP Coimbra	University of Liverpool
Royal Holloway University of London	University of Oxford
STFC Rutherford Appleton Laboratory	University of Sheffield
University College London	University of Zürich

ASIA

Center for Underground Physics

OCEANIA

University of Sydney



LZ Collaboration Meeting at UCLA, March 2025



Science and
Technology
Facilities Council



Fundação
para a Ciência
e a Tecnologia



SANFORD
UNDERGROUND
RESEARCH
FACILITY



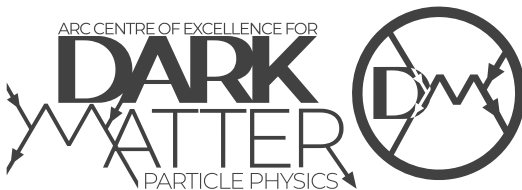
U.S. DEPARTMENT
of ENERGY



Swiss National
Science Foundation



Institute for
Basic Science



References

- [1] M. Arthurs et al., *The Real-Time Data Workflow of The LUX-ZEPLIN Dark Matter Experiment at NERSC*, EPJ Web Conf. **337**, 01232 (2025).
- [2] D. S. Akerib et al., *The LUX-ZEPLIN (LZ) Experiment*, NIM-A **953**, 163047 (2020).
- [3] D. S. Akerib et al., *Searches for Light Dark Matter and Evidence of Coherent Elastic Neutrino-Nucleus Scattering of Solar Neutrinos with the LUX-ZEPLIN (LZ) Experiment*, arXiv:2512.08065, (2025).
- [4] *Trinodb/Trino*, (2025).
- [5] The Ibis Project Authors, *The Ibis Project*, (2025).
- [6] *Apache/Superset*, (2025).
- [7] *Dlt-Hub/Dlt*, (2026).
- [8] *Helm/Helm*, (2025).
- [9] *Helmfile/Helmfile*, (2026).
- [10] SOPS Authors, *Getsops/Sops*, (2026).
- [11] S. Patton, T. Samak, C. E. Tull, and C. Mackenzie, *SPADE: Decentralized Orchestration of Data Movement and Warehousing for Physics Experiments*, in (IEEE, Ottawa, ON, Canada, 2015), pp. 1014–1019.
- [12] M. Barisits et al., *Rucio: Scientific Data Management*, Comput. Softw. Big Sci. **3**, 11 (2019).

References (cont.)

- [13] S. Krawczyk, E. ben Izzy, and D. Quinn, *Hamilton: Enabling Software Engineering Best Practices for Data Transformations via Generalized Dataflow Graphs*, in *1st International Workshop on Data Ecosystems Co-Located with 48th International Conference on Very Large Databases (VLDB 2022)*, edited by C. Cappiello, S. Geisler, and M.-E. Vidal (2022), pp. 41–50.
- [14] M. D. Wilkinson et al., *The FAIR Guiding Principles for Scientific Data Management and Stewardship*, *Sci Data* **3**, 160018 (2016).
- [15] LUX-ZEPLIN Collaboration et al., *Study of Few-Electron Backgrounds in the LUX-ZEPLIN Detector*, *Phys. Rev. D* **113**, 072018 (2026).
- [16] *Introduction to the American Science Cloud (AmSC): Part 2*, https://american-science-cloud.github.io/amsc-site/events/AmSC_January_Webinar_Part2_Slides.pdf.
- [17] T. Anderson, *Ensuring Long-Term Data Sustainability: Evaluating a Migration to Rucio for the LZ Dark Matter Experiment*, <https://indico.cern.ch/event/1471803/contributions/6967401/>.
- [18] C. Snaveley, G. Alvarez, V. Hendrix, S. Cholia, and S. Lasiewski, *Spin: A Docker-Based Platform for Deploying Science Gateways at NERSC*, (2018).

Backup

Motivating A Unified Metadata Interface

One of the grand challenges of data-intensive science... is to improve knowledge discovery through assisting both humans, and their computational agents in the access to...scientific data.

2016 paper on Findability Accessibility, Interoperability, Reusability (FAIR) principles (Collaboration et al, Phys. Rev. D **113**, 072018)
 • In the Open Access, Reusability (FAIR) principles (Collaboration et al, Phys. Rev. D **113**, 072018), favor open source data engineering tooling
 Wilkinson et al, Sci Data **3**, 160018



Recent update on LZ's [Ruico Migration](#)



Choosing a Dashboard Tool

CRITERIA

- Licensing & active maintenance
- Works with reverse proxy & SSO
- User-level authoring controls
- Deployable on `k8s` via `Helm`
- Table views with drill-down

Application	Evaluation
Lightdash	No Trino connector
Redash	No active maintainers
Metabase	SSO customization is a paid tier
Tableau	SSO customization is a paid tier
Grafana	Weaker table/drill-down views
Superset	Met every requirement

Applications were evaluated in early 2025. Grafana still used elsewhere in LZ for high-volume time series data.

Conditions

ACCESS

BEFORE

- Select specific version of conditions
- Display data from only one validity interval at a time
- No option to download data

ElectronLifetime	ElectronLifetimeError
16983	748

Electron lifetime data in Run DB Viewer

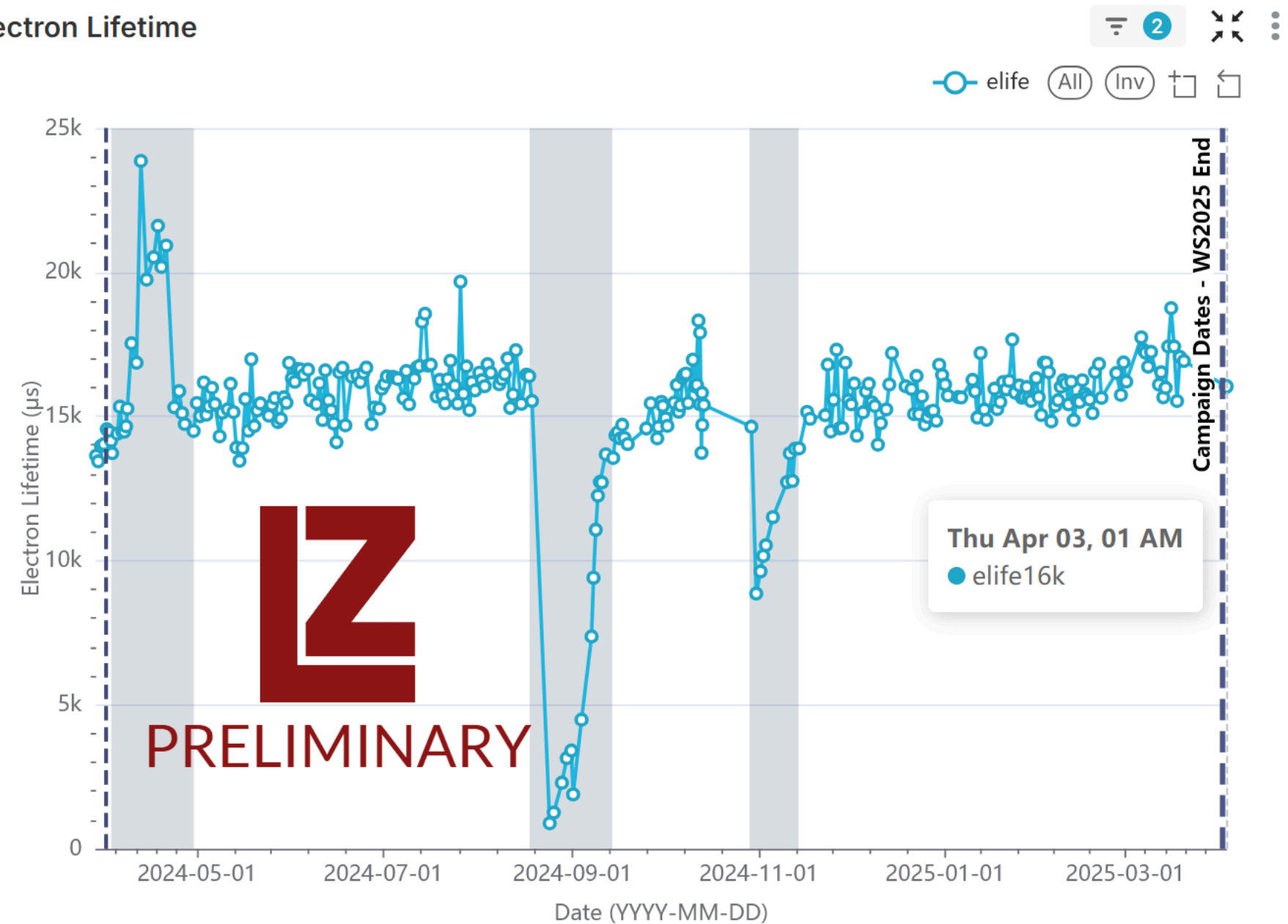
Conditions

[ACCESS](#)

AFTER

- Plots of data over time
- Can display data from prior validity intervals to examine changes between conditions versions
- Option to download data from accompanying table view (not shown)
 - Or get data directly in python with `ibis` 😊

Electron Lifetime



Plot of electron lifetime data in Superset

Run DB Viewer Interface

- Custom **dash** application for browsing key databases:
 - Data Catalog: finding raw and processed (RQ) data
 - Run Control: data acquisition status, DAQ parameters
 - Conditions: PMT gains, electron lifetime, other “versioned” detector parameters

 Run DB Viewer

Data

Run Control

Simulation

Conditions

Default displays the LATEST LZap production. Use search filters to see earlier productions.

All times Mountain.

Refresh Catalog Details

Search

Run

16262-16337

Populate search parameters

LZap Production

Op Name

Campaign

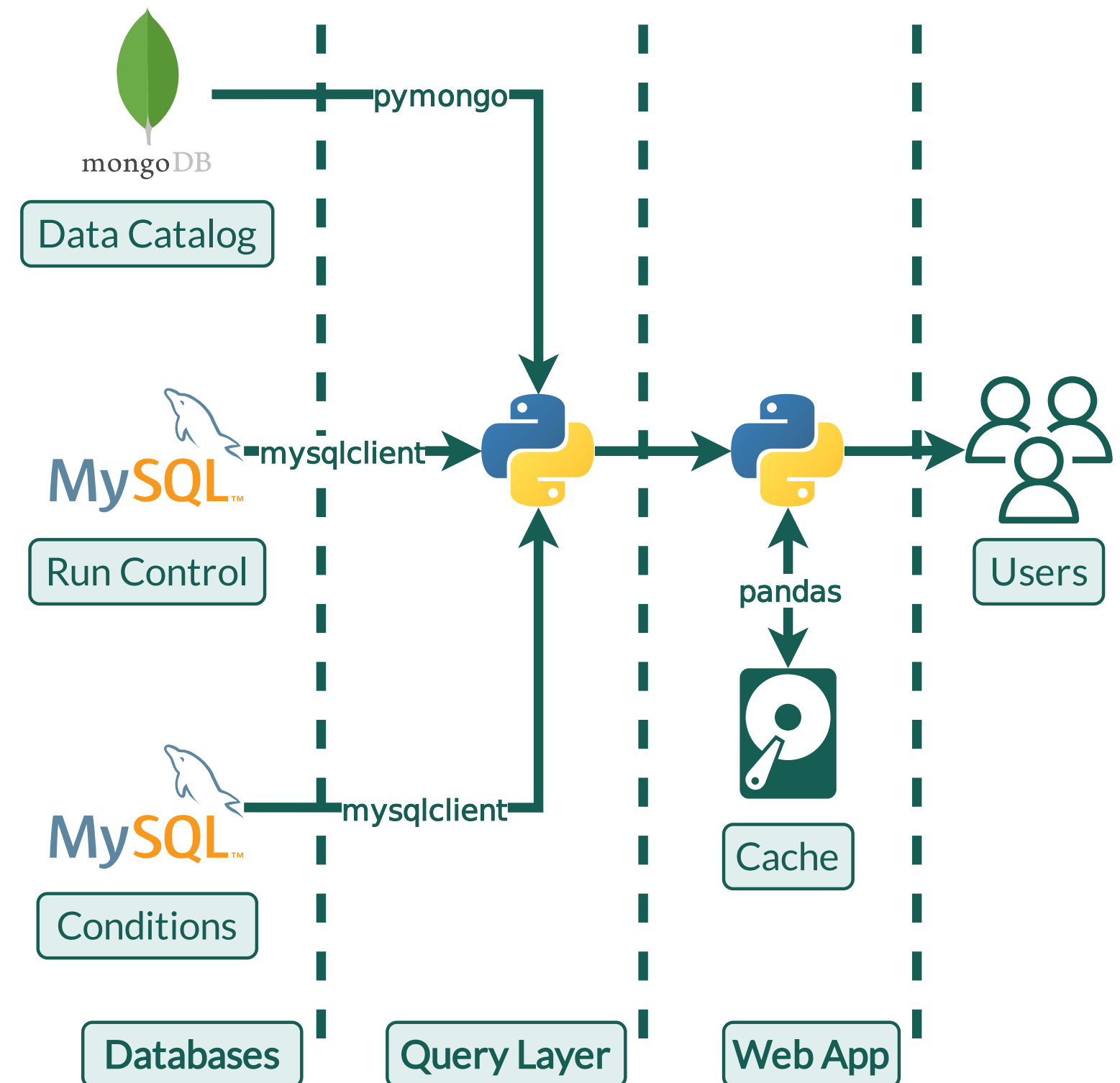
Source

Submit Clear Search

	Run	All productions	Production	Cond Ver	Prod Date	Proc. Evts	Tot. Evts	No. files
☐	16337	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T05:09:35	300000	300000	120
☐	16336	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:43:13	300000	300000	120
☐	16335	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:46:54	300000	300000	120
☐	16334	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:48:22	300000	300000	120
☐	16333	LZAP-6.1.0_PROD-0, LZAP-5.8.0_PROD-1	LZAP-6.1.0_PROD-0	v2368	2024-12-22T04:14:10	300000	300000	120

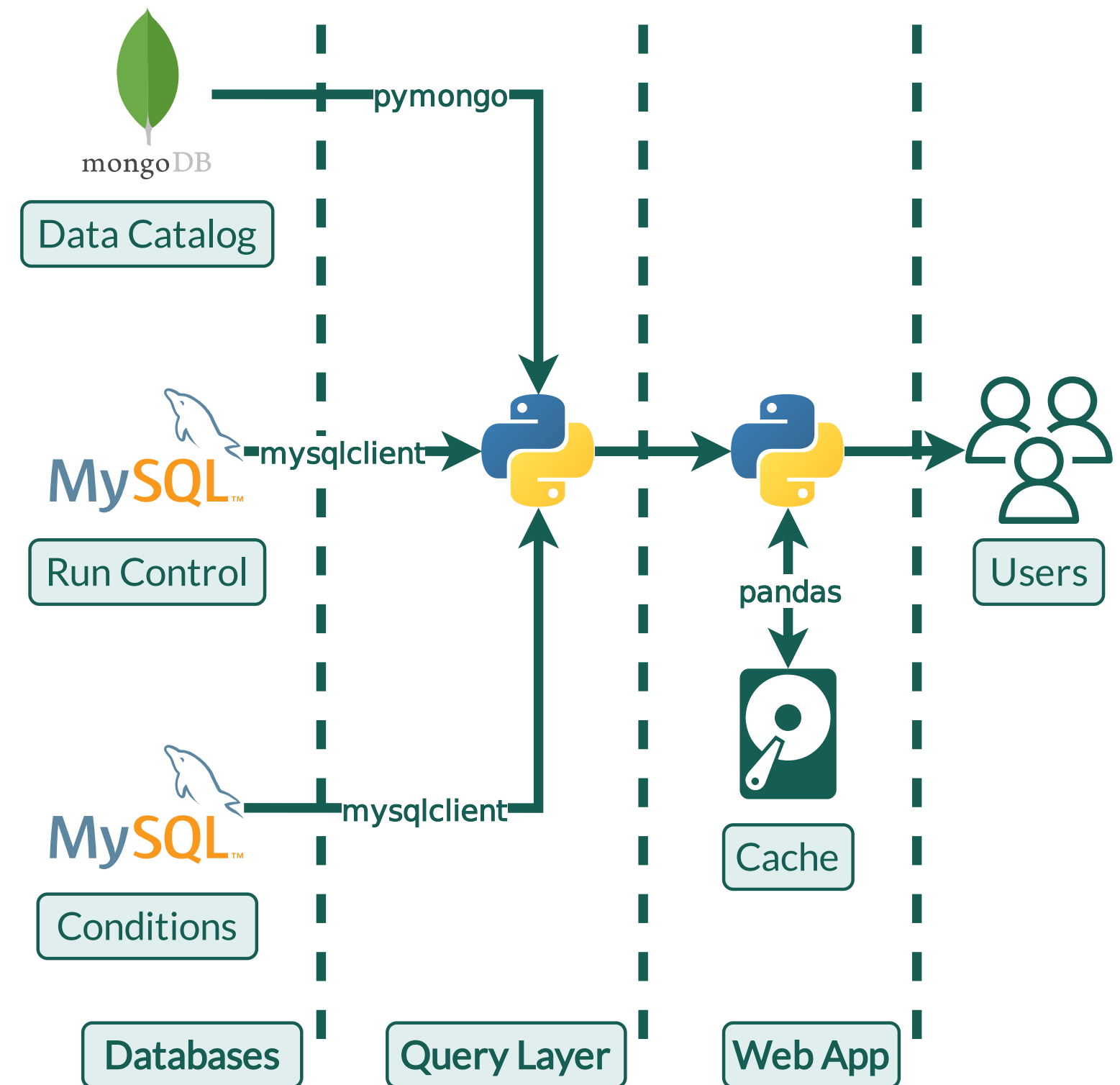
Run DB Viewer Issues

- Custom query layer did not re-use existing **programmatic interfaces**
- Manual restarts to reset cache
- Slow (>10s) response times
 - Indistinguishable from silent failures or poorly optimized queries/DBs
- > 5000 lines of Python



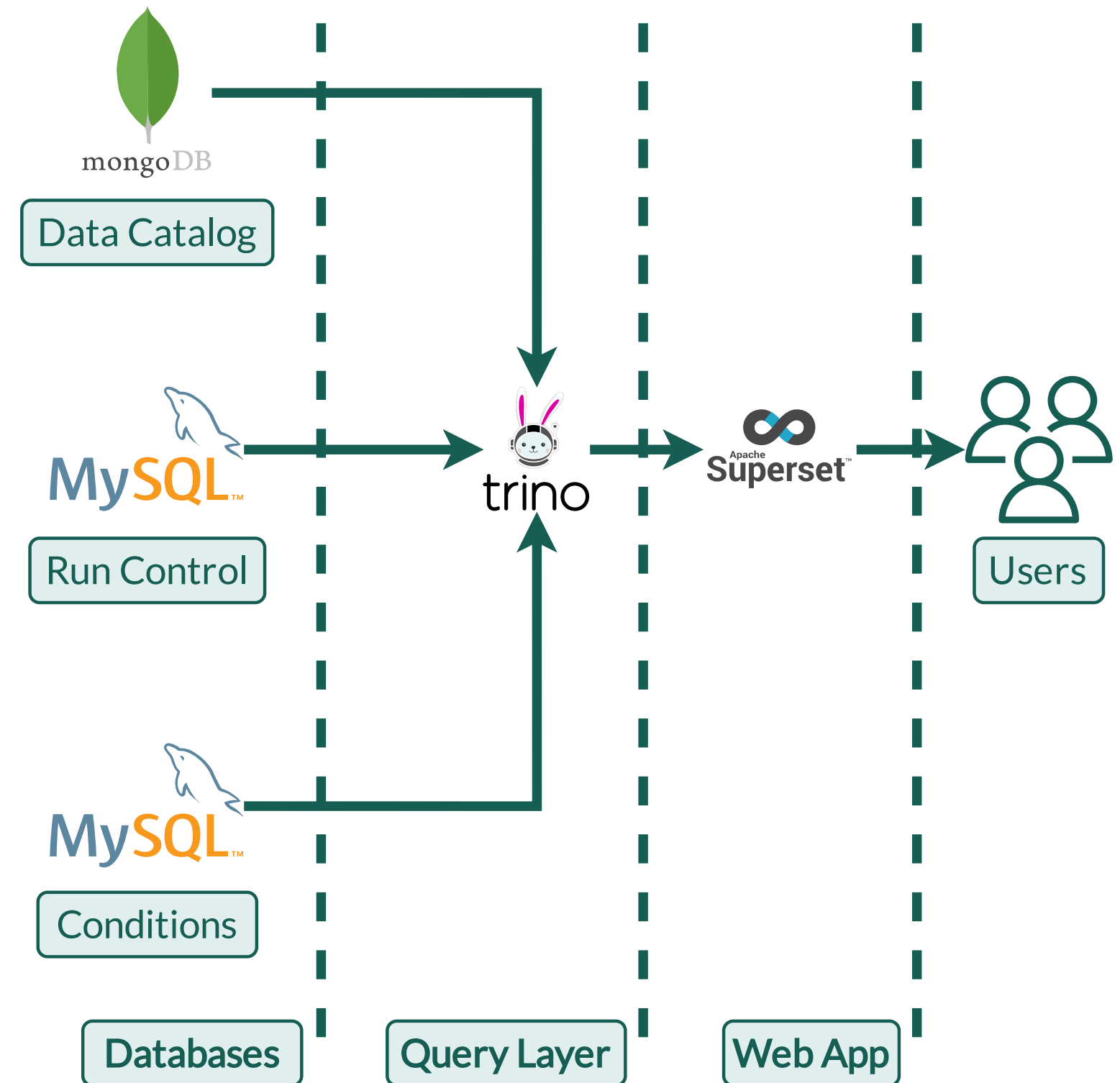
Run DB Viewer Architecture

- User logs into app
- App makes initial queries to custom query server
- Server runs initial queries metadata databases
- Initial queries cached as `pandas` dataframe for subsequent users



Run DB Viewer Replacement

Better performance, features, reliability
Same time investment



Run DB Viewer Deployment

- Hosted on NERSC Spin (based on Rancher)
 - Platform for managing Kubernetes (k8s) clusters
- **Right:** Rancher interface used to manage deployment configuration by hand

Good for learning and debugging, bad for reproducibility/versioning

Welcome to Spin Rancher 2. Please see the documentation at <https://docs.nersc.gov/services/spin/>

production run-db-viewer X

Starred Cluster Workloads CronJobs 0 DaemonSets 0 Deployments 4 Jobs 0 StatefulSets 0 Pods 3 Apps Service Discovery Storage Policy Monitoring Logging More Resources

Deployment: run-db-viewer Active

Namespace: run-db-viewer Age: 1774 days Pod Restarts: 0

Namespace * run-db-viewer Name * run-db-viewer Description Any text you want that better describes this resource

Replicas * 1

Deployment Pod run-db-viewer + Add Container

General

Health Check Resources Security Context Storage

General

Container Name run-db-viewer ☐ Init Container ☒ Standard Container

Image

Container Image registry.nersc.gov/lz/run-db-app:1.4.2 Pull Policy Always

Pull Secrets registry-nersc X

Networking

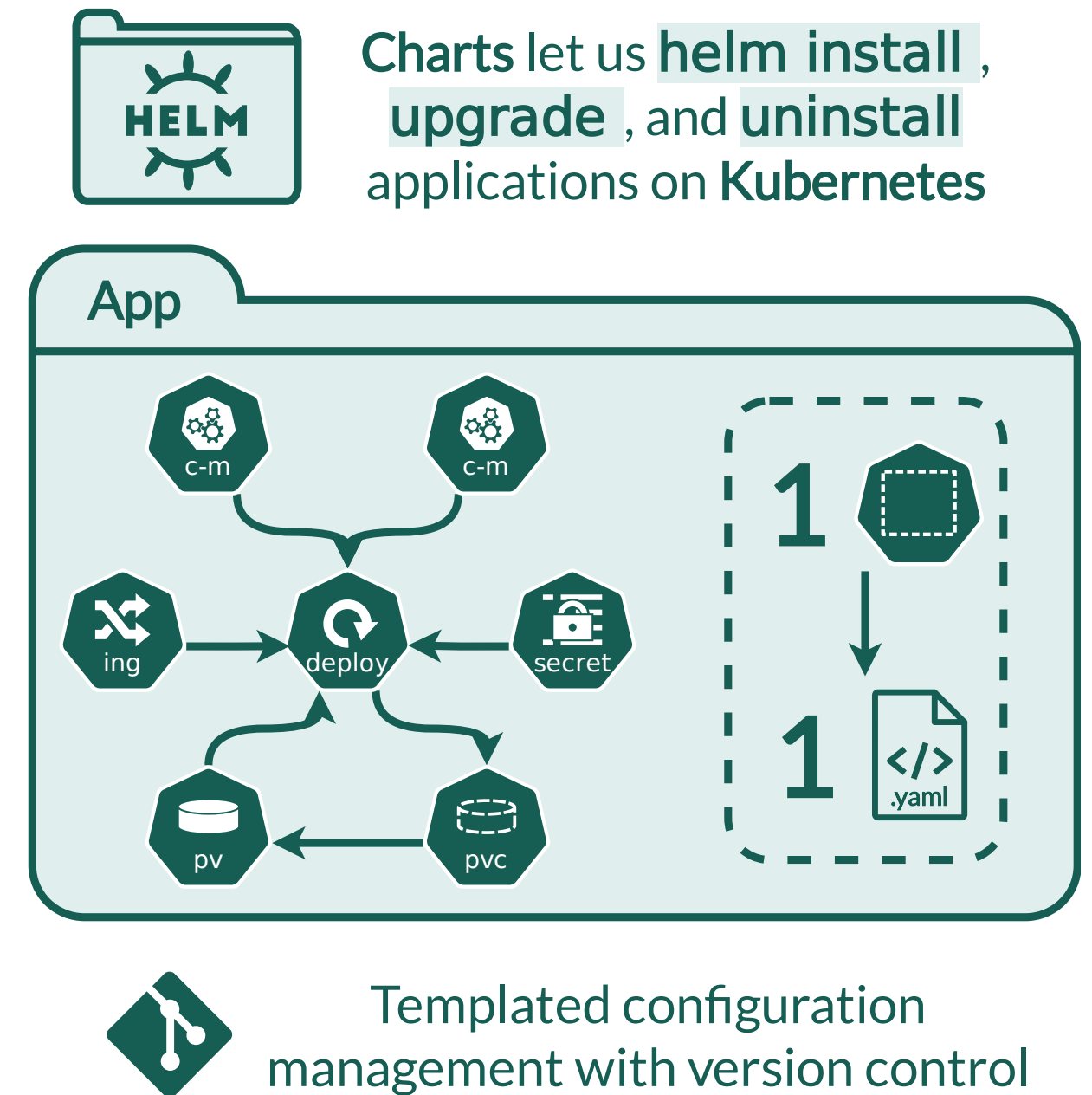
Define a Service to expose the container, or define a non-functional, named port so that humans will know where the app within the container is expected to run.

Cluster Tools v2.7.14

Cancel Edit as YAML Save

Superset and Trino Deployment

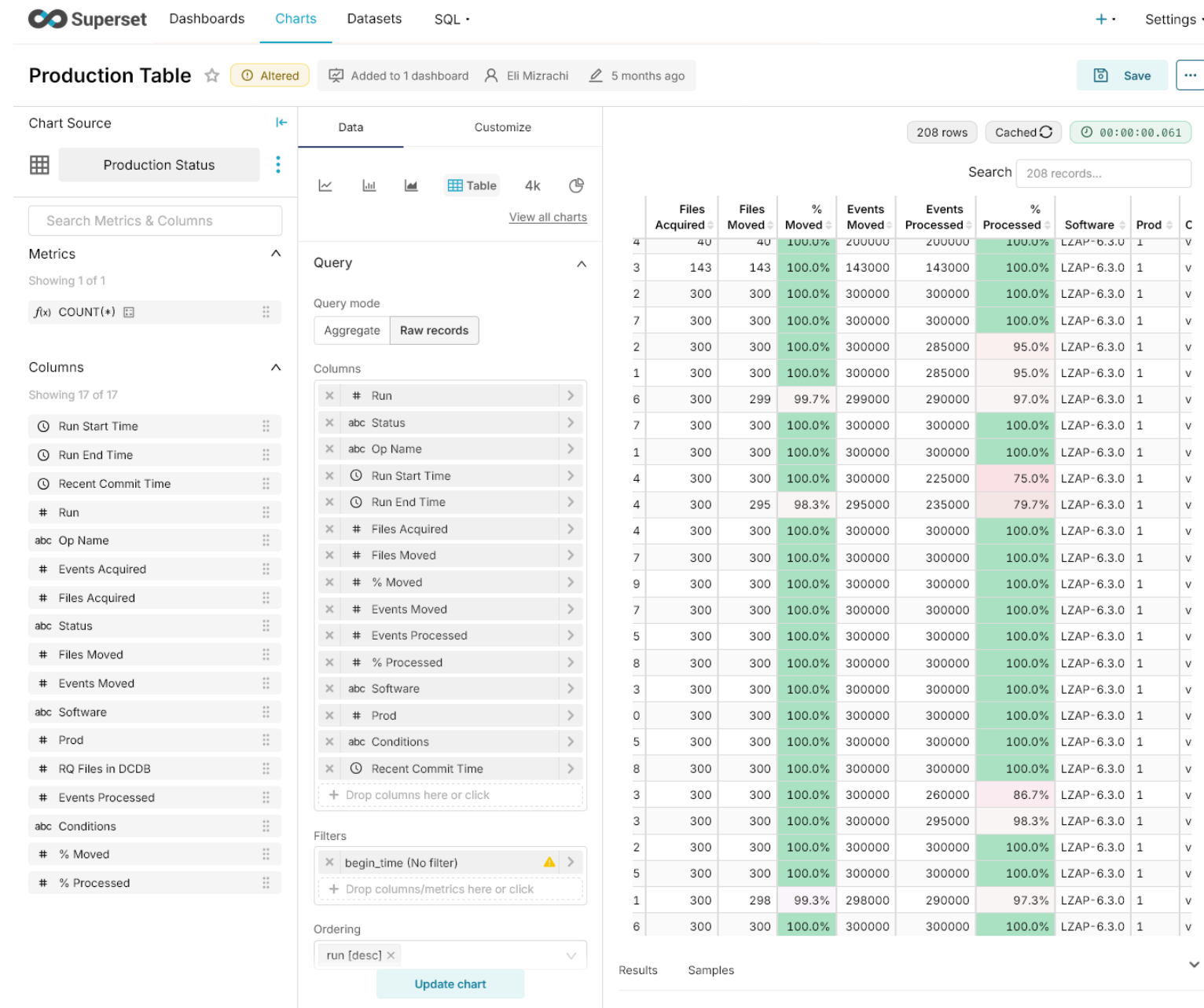
- Use `helm` to produce sets of `YAML` files using templates and single set of configuration values
- Applications packaged as “charts”:
 - Folder containing all templates and config parameters
- Charts are available for Superset and Trino
 - Final configuration by end-user (me!)
- Re-deploy charts in minutes on any `k8s` cluster (not just Spin!)



Apache Superset

- Open source Business Intelligence (BI) Application
 - Similar to Tableau, Metabase
- Single Sign On (SSO) and Role Based Authorization Control (RBAC) support
- Slack integration available for reports/alerts
- Chart configuration can be exported as **YAML** via API

Source data directly from databases or SQL queries

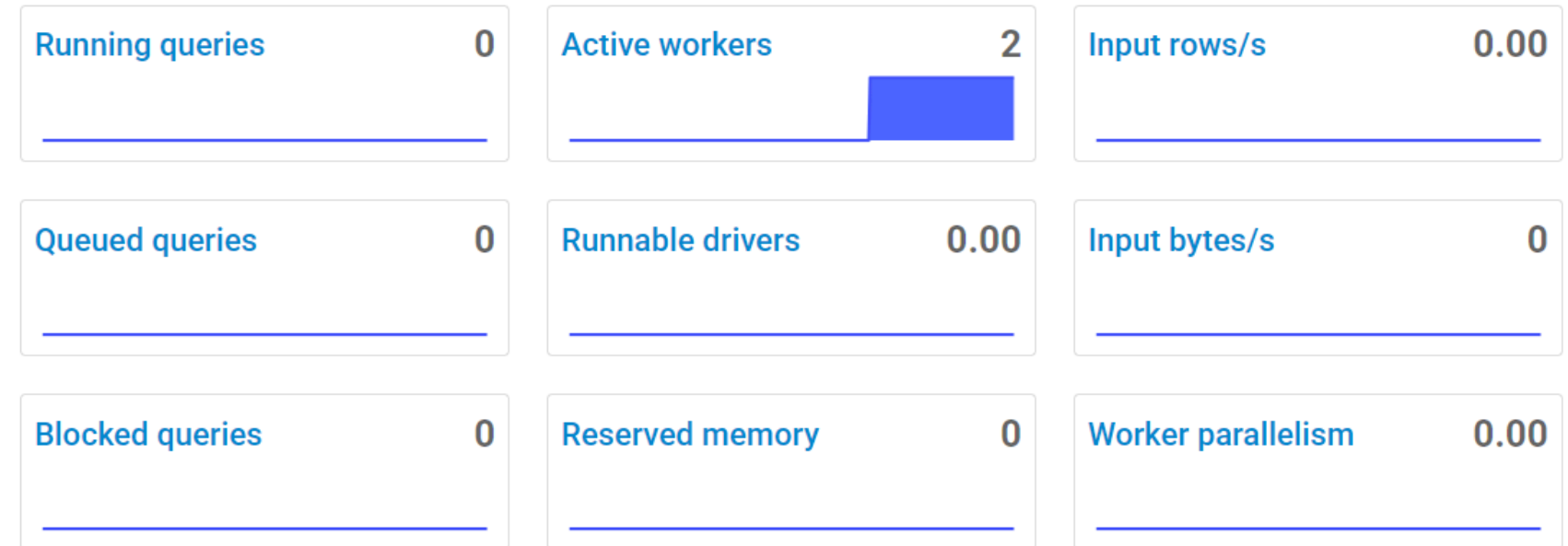


Superset drag-and-drop chart builder

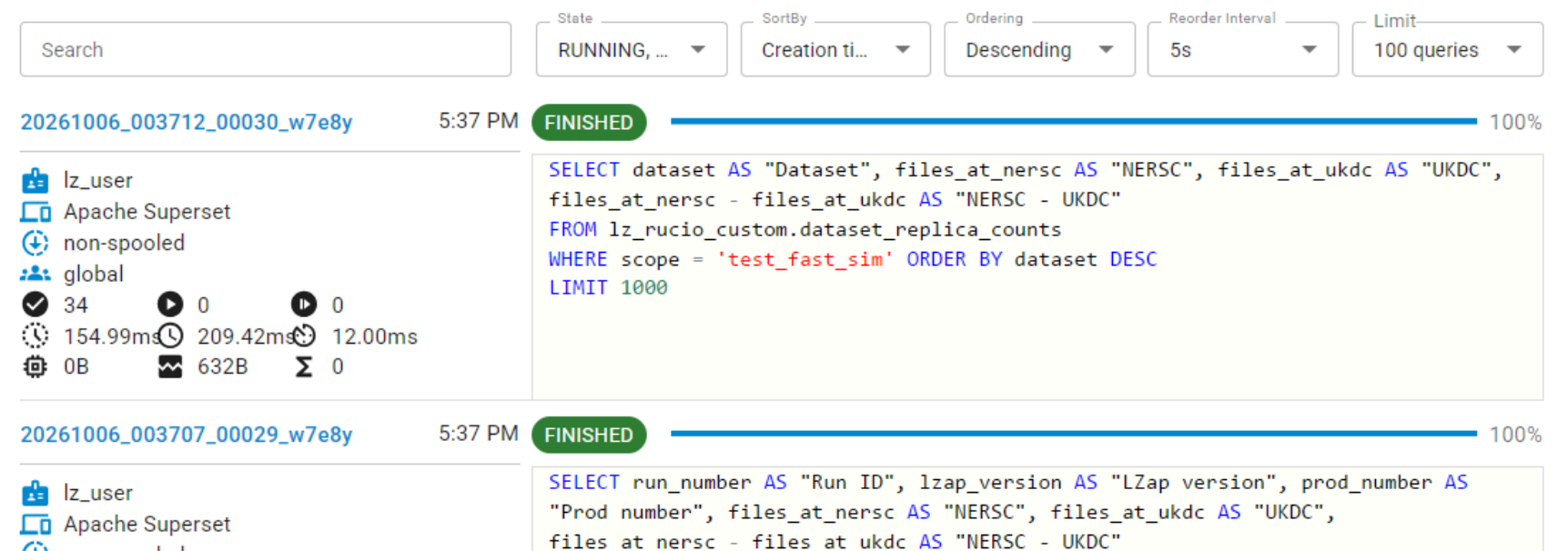
Trino

- Open source, distributed SQL query engine
- Same interface to dozens of SQL/NoSQL databases
 - Integrates with 100+ clients
- Databases behave as separate “catalogs”
 - Query, join, etc. between any of their tables
- RBAC and SSO support

Cluster overview



Query details



Trino interface shows recent queries and metrics

Superset Limitations (as of 6.1.0)

- Hacky support for error bars (can plot a confidence band)
- Many chart types *require* aggregations over columns
 - E.g. hard to pivot categorical columns which can't be aggregated
- Not suitable for heavy time-series/monitoring data; Grafana is better for this!