Science and Technology Facilities Council

Scientific Computing

Lakehouse briefing

Alexander Belozerov, Vasily Bunakov, Amali Pawula Hewage, Stuart Meacham, Tom Underwood

22 May 2026, RAL and online

Purposes of this briefing

- Inform relevant groups and individuals in STFC, Southampton and UCL
- Encourage the pre-Production (Staging) use
- See what the future engagement can be, and with who

The briefing is delivered by a few people in sections, with an opportunity to ask questions after each section

What to expect today

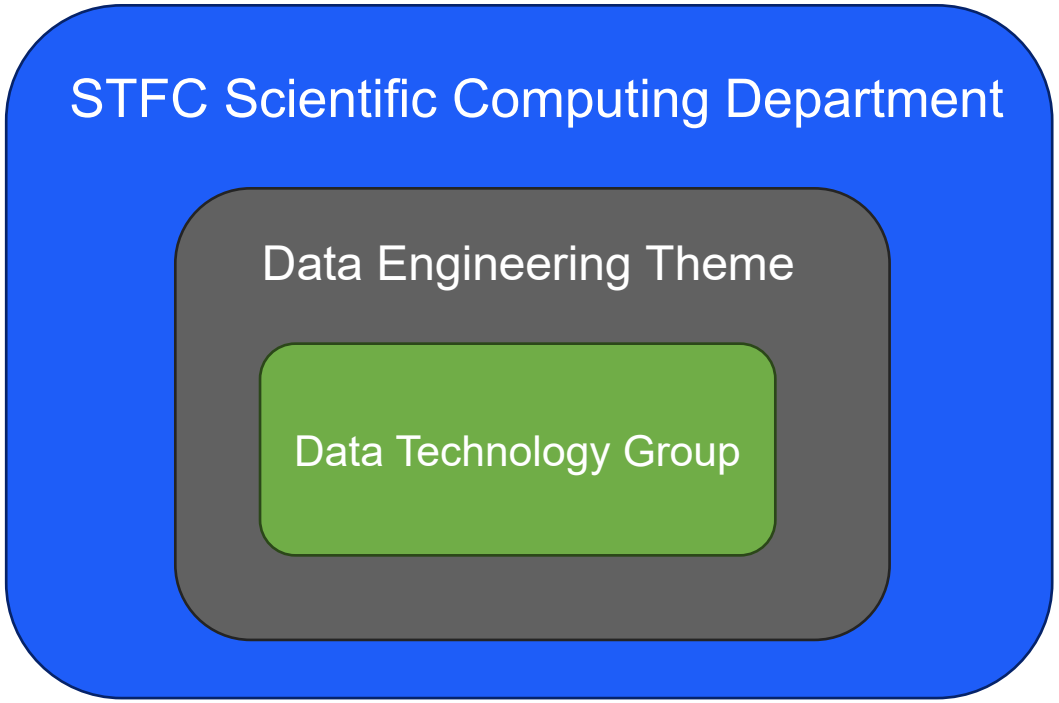
- Data Technology Group and what we do for PSDI
- A brief history of PSDI data lakehouse
- Datasets offered currently
- Core technology
- Interfaces

About DTG and PSDI

Presenter: Vasily

Science and Technology Facilities Council
Scientific Computing

About Data Technology Group



- We are research software engineers
- 5 core staff and 3 graduates
- Projects:

  PSDI (Physical Sciences Data Infrastructure) www.psdi.ac.uk

  BatCat www.batcat.info

  Materials Commons MaterialsCommons4EU

Science and Technology Facilities Council
Scientific Computing

What the DTG do for PSDI

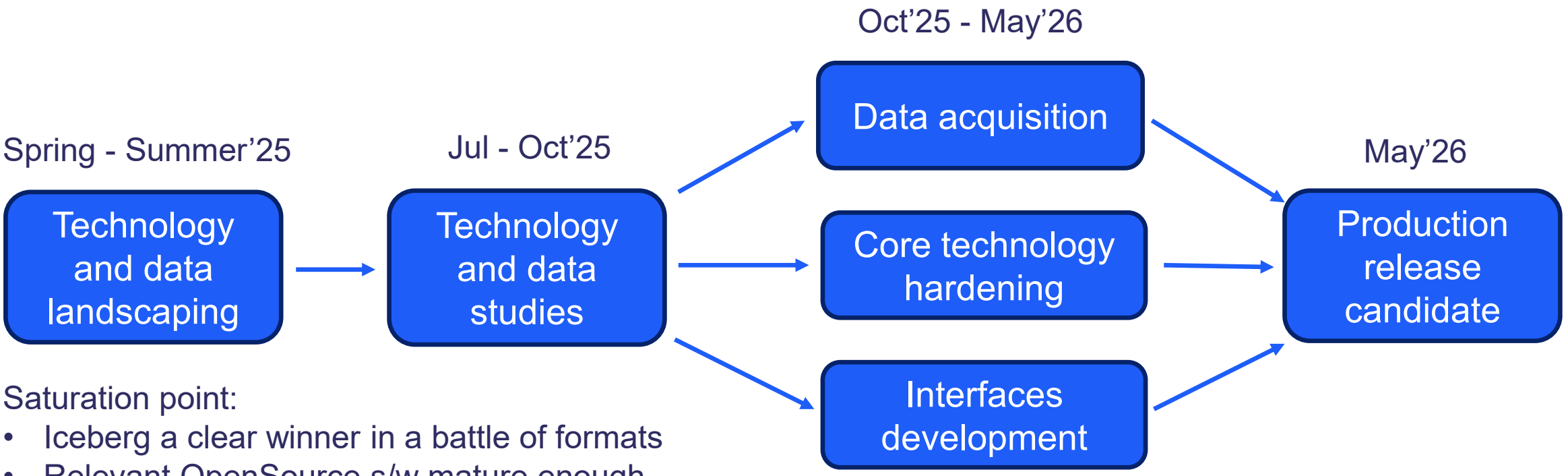
- Kubernetes clusters for Development, Staging, Production and DevOps for the entire project
- Containerisation in Docker and Apptainer, for our own and partners' software
- CI/CD for our own and partners' deployments
- Online training environment: Moodle and Jobe (and Southampton supply content)
- Index of data and publications, and interfaces to it
- Crystal structures generator (assisting University College London)
- Data sharing and synchronisation solution
- Data lakehouse for remote access to bigger data
- Mirror of Crystallography Open Database (with University of Vilnius)
- Natural language interfaces

Science and Technology Facilities Council
Scientific Computing

Software engineering
User-facing deployments

Dark blue: in production
Light blue: in production soon
Grey: proof-of-concept

How the PSDI lakehouse evolved



Saturation point:

- Iceberg a clear winner in a battle of formats
- Relevant OpenSource s/w mature enough
- New interesting data released

Production readiness

- Highly-available deployment on Kubernetes
- Inexpensive object store backend, ready to scale
- A few popular datasets prepared and tested
- Full integration with PSDI user management
- Documented examples for end-users
- We are ready to go and collect user feedback in-the-field

The lakehouse value proposition

For the research infrastructure owner

- **A novel way of data publishing:**

  Bigger datasets can be published inexpensively, with an object store as a backend

  Compared to a repository, there is a live API to dig in the datasets => appeal to a new audience of “doers”

- **User permissions and access policies are well managed**, up to the point of having a dedicated policy engine

- **Fuller implementation of FAIR principles** (see next slide)

For the research infrastructure user

- **No need to move full datasets** to integrate them in a research flow

- **A uniform API:** no need to install or learn multiple dataset-specific API connectors, or follow their evolution

- **“Time travel” is possible**, querying snapshots of evolving data

- **Separation of concerns:** a good part of data wrangling can be offloaded on the Cloud infrastructure, leaving science to scientists

FAIR principles implementation

	Data repository	Data lakehouse
Findable data	Yes	Yes
Accessible data	Yes	Yes
Interoperable data	No	Yes
Reusable data	Maybe (reusability is just best hope)	Maybe (reusability is supported by design)

Any questions so far?

Datasets offered currently

Presenter: Tom Contributor: Alexander

How we selected datasets

- Couple of bigger ones with >100M records to demonstrate that querying and filtering is performant enough
- The rest from Materials Project, as it has relevance not only to PSDI but to other projects by the DTG
- We picked what could be “flattened” and transformed in the columnar format without much loss of expressivity *
- We wanted relevance to Materials Summit in Manchester (with training in mind)

* Which will be of a lesser concern when we move on Iceberg v3 that can efficiently manage nested data structures

Datasets served currently

Name	Description	No of records	Master publication
Open Materials 2024 (OMat24)	DFT calculations on inorganic bulk materials. 400 million CPU core-hours.	> 100M	arXiv:2410.12771
Open Molecules 2025 (OMol25)	Parsed version as in Hugging Face. DFT calculations on small molecules, biomolecules, metal complexes, and electrolytes. 6 billion CPU core-hours.	> 100M	arXiv:2505.08762
Materials Project (MP) - AWS Open Data	Various physical properties derived from MP's DFT calculations for materials, e.g. absorption, density, phonons. 17 datasets.	~200k compounds	doi.org/10.1038/s41563-025-02272-0 doi.org/10.1063/1.4812323
Materials Project – Contribs	MPtrj 2022.9 & MatPes v2025.1 - DFT datasets for training universal machine-learning interatomic potentials	>2M	doi.org/10.1038/s42256-023-00716-3 arXiv:2503.04070
PubChem Compounds	Summaries with an aggregated view of what is known for a single chemical structure.	> 100M	https://pubchem.ncbi.nlm.nih.gov/docs/compounds

In progress, with Production-ready release coming later

Any questions?

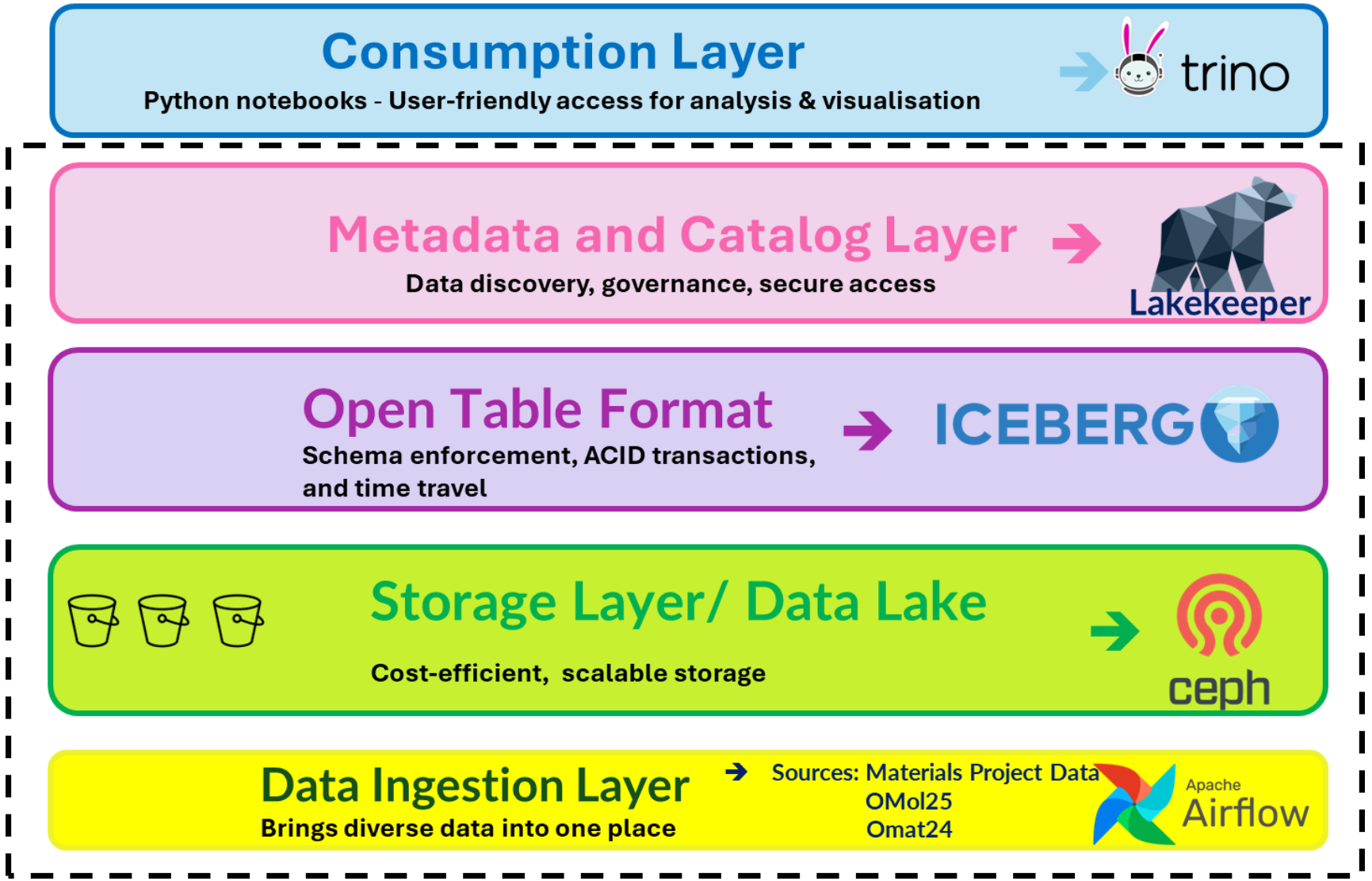
Core technology

Or why scientists need us engineers ☺

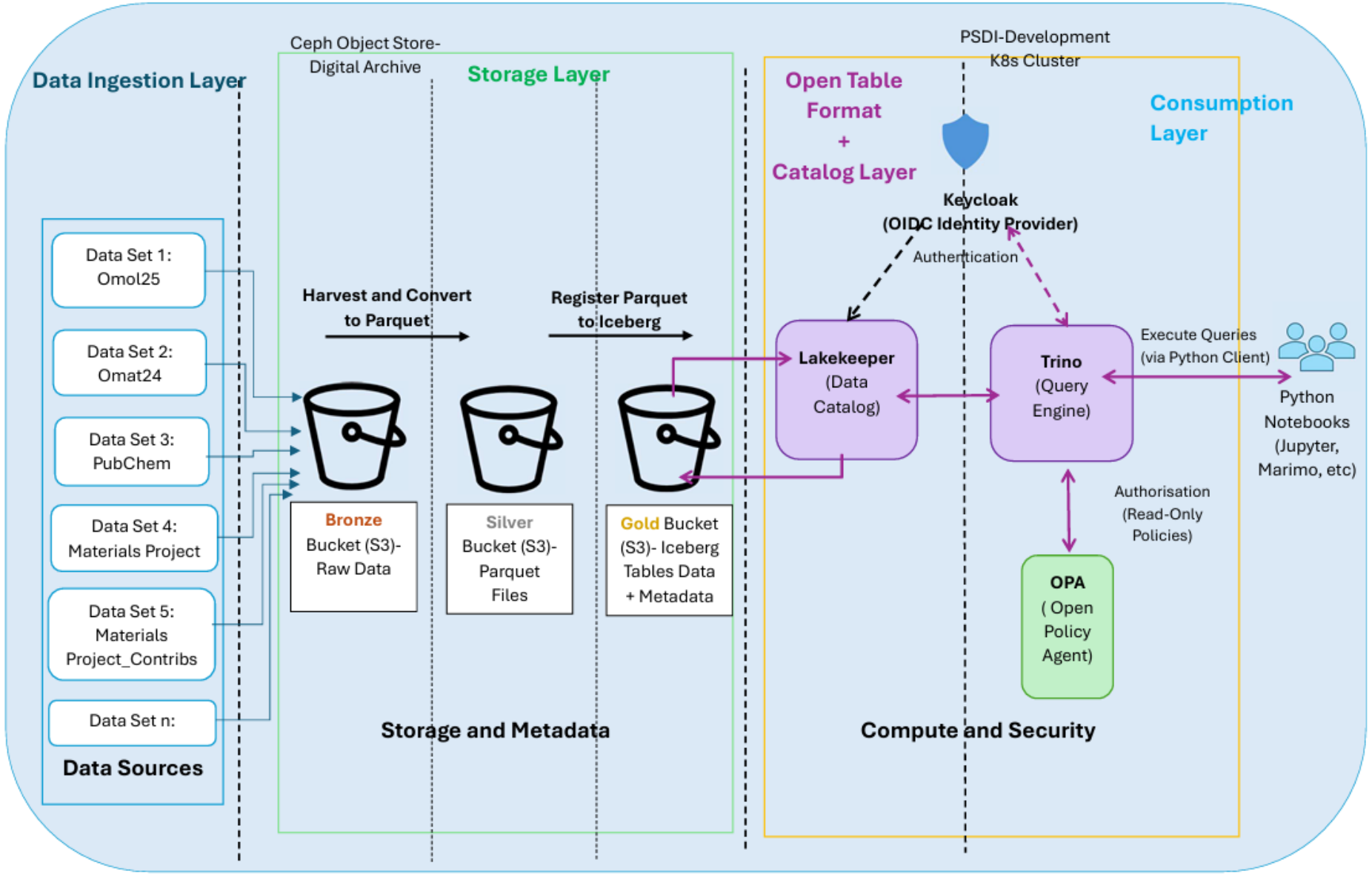
Presenter: Amali Contributor: Stuart

UKRI Science and Technology Facilities Council
Scientific Computing

5 Layers of Data Lakehouse



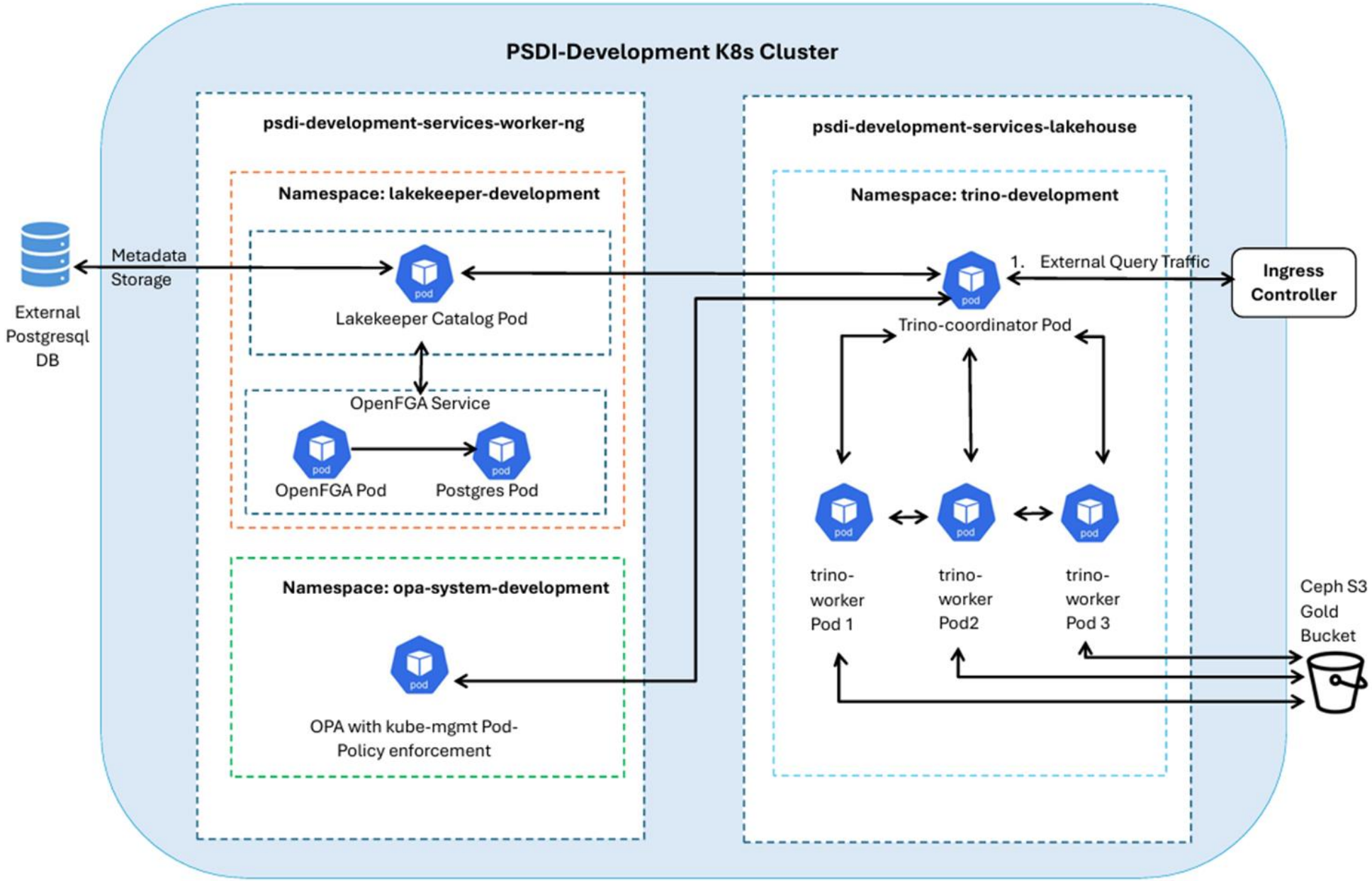
Lakehouse architecture - from raw files to SQL and Python



Science and Technology Facilities Cou

Scientific Computing

Deployment architecture - How it is deployed in K8s



Scientific Cor

Any questions?

Interfaces

Presenters: Alexander, Tom

Science and Technology Facilities Council
Scientific Computing

Available options for querying PSDI Lakehouse

SQL

"Lingua franca" of data science

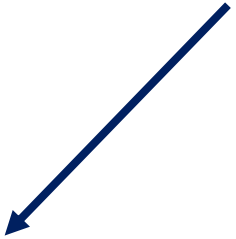
Ibis

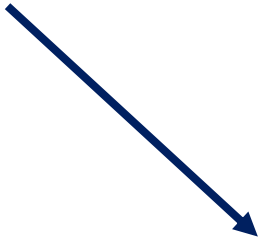
Python dataframe library
https://ibis-project.org

Science and Technology Facilities Council
Scientific Computing

Tutorials: https://github.com/PSDI-UK/lakehouse-tutorials

Available options for querying PSDI Lakehouse





This talk

SQL

"Lingua franca" of data science

Ibis

Python dataframe library
https://ibis-project.org

Science and Technology Facilities Council
Scientific Computing

Tutorials: https://github.com/PSDI-UK/lakehouse-tutorials

Available tutorials

https://github.com/PSDI-UK/lakehouse-tutorials

Marimo notebooks

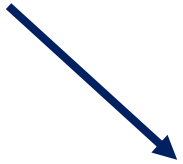
Jupyter notebooks

- better reproducibility (no hidden state),
- pure Python, so Git-friendly and can be run as scripts
- convertible to Web applications,
- now support AI co-pilot.

Available tutorials

https://github.com/PSDI-UK/lakehouse-tutorials

Marimo notebooks



This talk

Jupyter notebooks

- better reproducibility (no hidden state),
- pure Python, so Git-friendly and can be run as scripts
- convertible to Web applications,
- now support AI co-pilot.

Setting up dependencies for Ibis

```
!pip install -q "ibis-framework[trino]" trino
```

```
import ibis
from ibis.backends.trino import Backend

from trino.dbapi import connect
from trino.auth import OAuth2Authentication
```

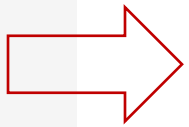
Login with PSDI account

```
TRINO_HOST = "trino-staging.psdi.ac.uk"

conn = connect(
    host=TRINO_HOST,
    port=443,
    http_scheme="https",
    auth=OAuth2Authentication(),
    catalog="psdi",
    request_timeout=300,
)

# Hand the live Trino connection to Ibis
con = Backend.from_connection(conn)

# List the available databases
con.list_databases(catalog="psdi")
```



PSDI
PHYSICAL SCIENCES DATA INFRASTRUCTURE

UK Institution Sign In

If you're affiliated with a UK research institution or university, select the option below to access your account via institutional single sign-on for secure and verified access.

UK Institution Sign In (Do not use)

Test Idp (Do not use)

Need help? Contact support@psdi.ac.uk

Sign in to your account

Email
alexander.belozerov@stfc.ac.uk

Password
••••••••

Forgot Password?

Remember me

Sign In

New user? Register

Science and Technology Facilities Council
Scientific Computing

Discover what is available

List databases

```
databases = con.list_databases()
pprint(databases)
```

```
['materials_project',
 'materials_project_contribs_matpes',
 'materials_project_contribs_mptrj',
 'omat24',
 'omol25',
 'pubchem']
```

List tables in a database

```
tables = con.list_tables(database="materials_project")
pprint(tables)
```

```
['absorption',
 'alloy_pair_members',
 'alloy_pairs',
 'alloy_systems',
 'alloys',
 'chemenv',
 'dielectric',
 'elasticity',
 'insertion_electrodes',
 'magnetism',
 'oxi_states',
 'phonon',
 'piezoelectric',
 'provenance',
 'robocrys',
 'structure_groups',
 'task_validation']
```

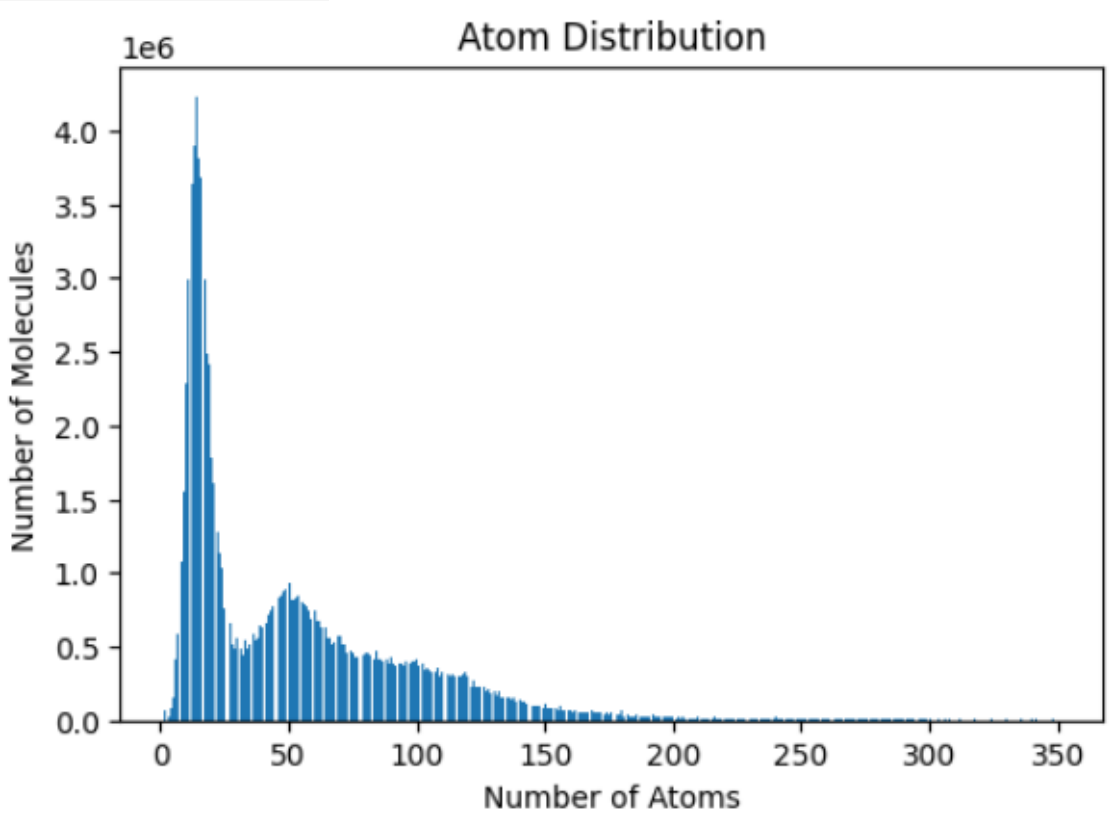
OMol25: distribution of molecules by number of atoms

```
table = con.table("omol25", database="omol25")
```

```
result = (
    table
    .group_by("num_atoms")            # group rows by the num_atoms column
    .aggregate(count=table.count())   # count rows in each group
    .order_by("num_atoms")            # sort results by num_atoms in ascending order
    .to_pandas()                      # execute query and return as a pandas DataFrame
)

num_atoms = result["num_atoms"]
counts = result["count"]

plt.figure(figsize=(6, 4))
plt.bar(num_atoms, counts)
plt.xlabel("Number of Atoms")
plt.ylabel("Number of Molecules")
plt.title("Atom Distribution")
plt.show()
```



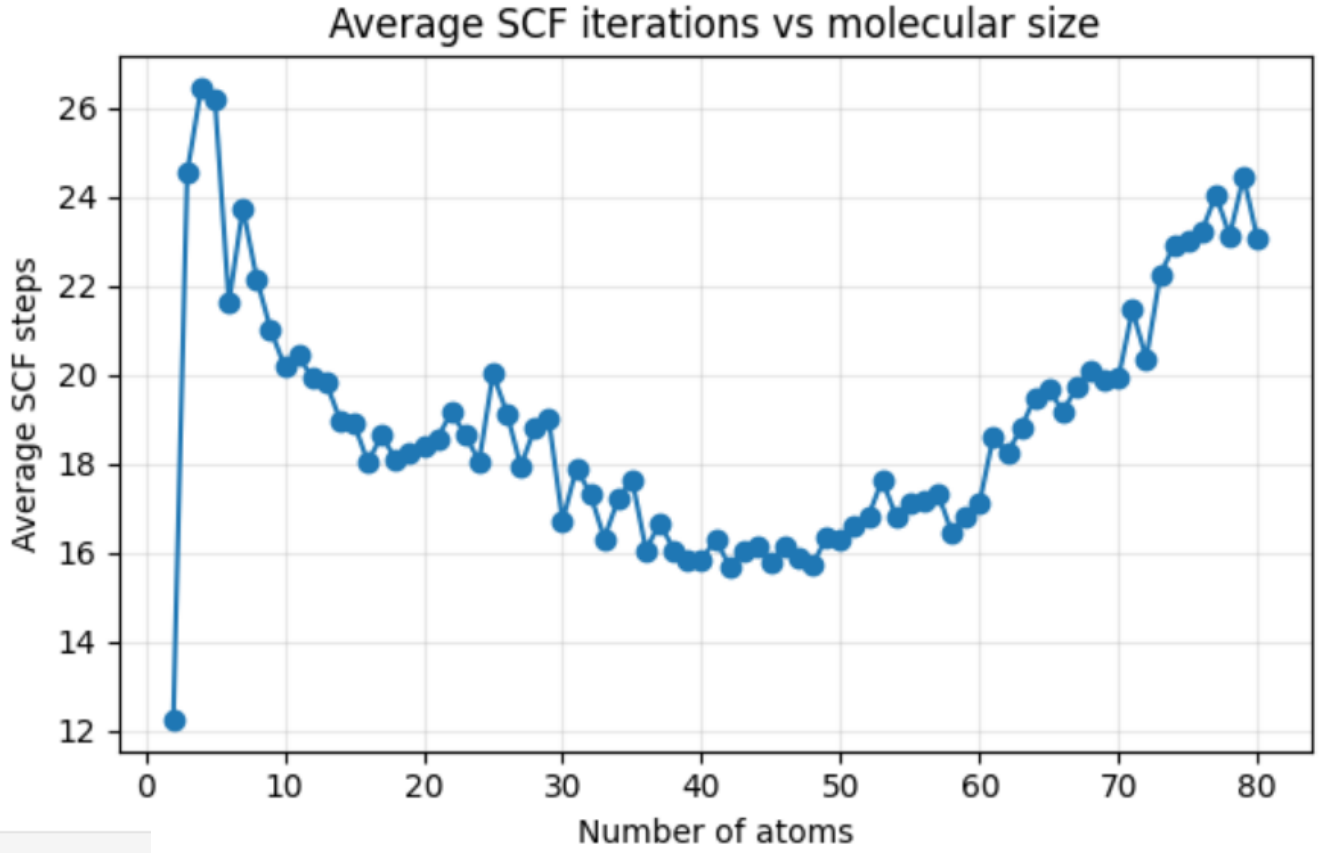
Execution time: 5 seconds

Science and Technology Facilities Council

Scientific Computing

OMol25: Average SCF iterations vs molecular size

```python
df = (
    table.filter(
        (table.charge == 0) &
        (table.num_atoms <= 80) &
        table.n_scf_steps.notnull()
    )
    .group_by("num_atoms")
    .aggregate(
        avg_scf_steps=t.n_scf_steps.mean(),
        n_samples=t.data_id.count()
    )
    .order_by("num_atoms")
    .to_pandas()
)
```




```python
plt.figure(figsize=(6, 4))
plt.plot(df["num_atoms"], df["avg_scf_steps"], marker="o")
plt.xlabel("Number of atoms")
plt.ylabel("Average SCF steps")
plt.title("Average SCF iterations vs molecular size")
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()
```

Execution time: 12 seconds
This is a quite heavy query over >100M records with grouping and ordering

Materials Project: Basics

List Materials Project tables

```
con.list_tables(database="materials_project")
```

Response:

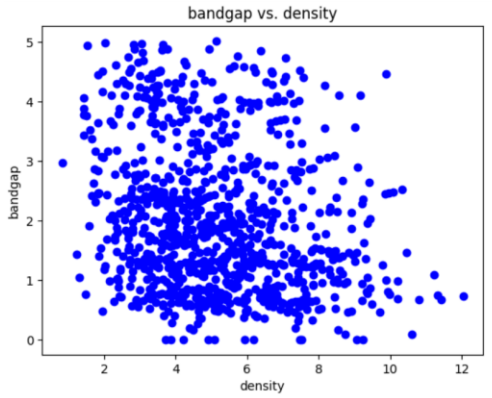
```
['absorption', 'alloy_pair_members', 'alloy_pairs',
'alloy_systems', 'alloys', 'chemenv', 'dielectric',
'elasticity', 'insertion_electrodes', 'magnetism',
'oxi_states', 'phonon', 'piezoelectric', 'provenance',
'robocrys', 'structure_groups', 'task_validation']
```

Plot band gap vs. density for materials in the **absorption** table

```
absorption_table = con.table("absorption", database="materials_project")
result = absorption_table.select("density", "bandgap")
```

Response:

	density	bandgap
0	6.156089	1.2693
1	5.192443	0.7118
2	6.113212	1.3159
3	5.666257	1.4379
4	4.886297	1.4940
..	...	...
941	1.218967	1.4301
942	1.655223	2.4209
943	1.837305	2.8276
944	1.859254	1.4049
945	1.863363	2.4619



List columns in the **absorption** table

```
absorption_table = con.table("absorption", database="materials_project")
result = absorption_table.schema()
```

Response:

```
ibis.Schema {
  nsites                  int64
  elements                string
  nelements               int64
  formula_pretty          string
  formula_anonymous       string
  chemsys                 string
  volume                  float64
  density                 float64
  density_atomic          float64
  property_name           string
  material_id             string
  deprecated              boolean

    . . .

  composition.eu          float64
  composition_reduced.eu          float64
  composition.kr          float64
  composition_reduced.kr          float64
  composition.xe          float64
  composition_reduced.xe          float64
}
```

Materials Project: Filtering & Joining

Find materials containing Si or O listed in the **absorption** table with the 5 largest band gaps. Print their formula, ID and band gap.

```
absorption = con.table("absorption", database="materials_project")

result = (
    absorption
    .filter(
        absorption.formula_pretty.re_search(r"Si")
        | absorption.formula_pretty.re_search(r"O(?![gs])")
    )
    .select(
        absorption.formula_pretty,
        absorption.material_id,
        absorption.bandgap,
    )
    .order_by(ibis.desc("bandgap"))
    .limit(5)
)
```

Response:

	formula_pretty	material_id	bandgap
0	YOF	mp-3637	5.0135
1	Li2O	mp-1960	4.9861
2	SrHClO	mp-24066	4.9690
3	NdClO	mp-23058	4.7550
4	SmOF	mp-9488	4.7331

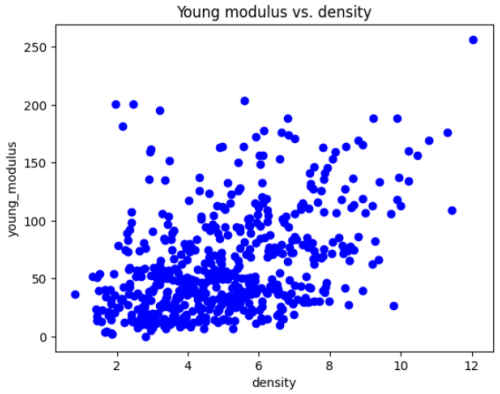
The density from **absorption** table against the bulk modulus from the **elasticity** table, for materials in both tables

```
absorption = con.table("absorption", database="materials_project")
elasticity = con.table("elasticity", database="materials_project")

result = (
    absorption
    .join(
        elasticity,
        absorption.material_id == elasticity.material_id,
        how="inner",
    )
    .filter(
        elasticity["bulk_modulus.voigt"].notnull()
    )
    .select(
        absorption.density.name("density"),
        elasticity["bulk_modulus.voigt"].name("bulk_modulus"),
    )
)
```

Response:

	density	bulk_modulus
0	4.886297	65.597
1	5.887064	52.136
2	6.239155	98.815
3	2.341314	36.397
4	8.497954	74.071
..	...	...
552	2.281112	88.916
553	1.655223	3.525
554	1.837305	3.169
555	1.859254	1.917
556	1.863363	2.149



Thank you for coming!

Try our notebooks,
build your own
and give us feedback

We can have a hackathon, too!

Appendix:
Popular questions and answers

What is PSDI data lakehouse?

- A digitally sovereign deployment of core IT components and open data in STFC Cloud
- The “third way” of research data management that complements research data repositories or the common habit of moving data closer to computation
- Somewhat like a database, but with a separation of concerns for data storage and for querying it, and a different approach to schema evolution *
- Suitable for bigger datasets, as cheaper storage can be used whilst maintaining reasonable performance
- Multiple and diverse datasets can be accessed and exploited via uniform interface

* We can only recommend (again) the first 10 minutes of Tim Berglund’s “Apache Iceberg: What It Is and Why Everyone’s Talking About It”

FAQs

- **Is lakehouse an exotic technology?**
  No, it is mainstream nowadays, although not as much in research as in business applications for advanced analytics and AI.
- **Can lakehouse serve any data?**
  It better suits structured or semi-structured numeric and textual data, e.g. XYZ, CSV, JSON or anything more specific like CIF, JCAMP-DX and alike that are de facto textual formats. As long as we can sensibly transform a data source in Parquet *, it can be served by a lakehouse.
- **Can it serve images?**
  Not directly. Possible strategies:
  a) Manage metadata in the lakehouse with links to images elsewhere.
  b) Produce numeric-and-textual representations of images and manage them, at a good scale. **
  c) Look for a different technology.
- **Can it serve Big Data?**
  Let's not forget that Big Data is not only about Volume, but also about Variety and Velocity. Lakehouse architecture can address all three Vs of Big Data, it was conceived with that in mind.
- **What are lakehouse benefits for a research user?**
  It behaves like a uniform database interface over sizeable, diverse or ever-changing data archive.

---

* See https://parquet.apache.org/

** Our current lakehouse at its launch is based on Apache Iceberg v2 specification, as the ecosystem of Open Source tools is richer for v2. Next v3 has native support for geometry and better support for any "nested" data structures. We would be much interested to see what the lakehouse can do e.g. for images segmentation and labelling in electronic microscopy, when Iceberg v3 adoption improves.

What is the lakehouse ambition for next two years?

- Maintain digital sovereignty with Open Source components for code base
- Grow data collections far beyond the initial “seed” data
- Grow user community and capitalise on user feedback
- Unleash innovation in machine and user interfaces to research data
- Invite research communities to upload their data, too, not only to read it, i.e. make the lakehouse a fully fledged data publishing platform