

Aeromancy: Towards More Reproducible AI and Machine Learning

David McClosky
QuantAQ Inc.

On reproducibility

- Goal: Make it easy to replicate or modify any existing experiment
- Reproducible experiments are critical for good science, aid debugging, and clarify data lineage
- Data pipelines and ML experiments are **not reproducible** by default
- Code depends on **full environment** (OS, compilers, libraries, etc.)
- Datasets and models are often unversioned, can change frequently
- Existing Experiment Managers (e.g., Weights and Biases, MLFlow) track many, but not all of these details

Aeromancy has opinions

Runtime Container
<> Code → git commit hash
OS → docker base image
Libraries → Debian package versions

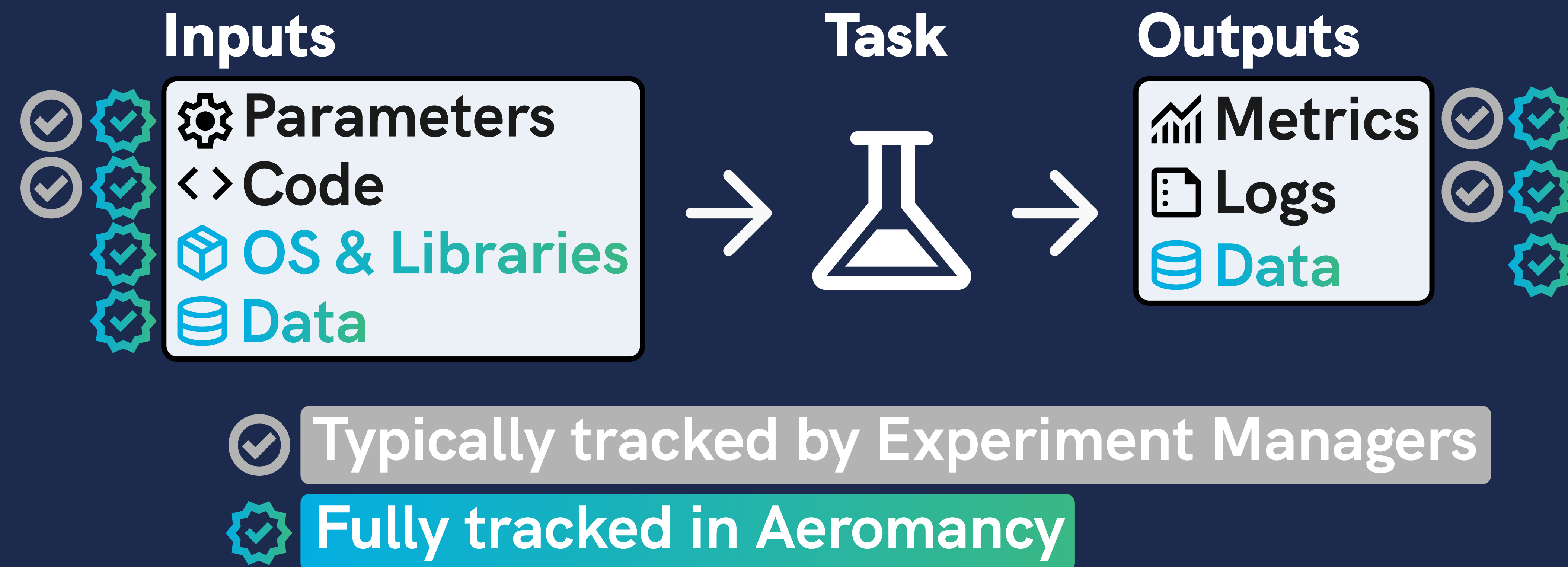
External Artifacts
Data → S3-like Object Storage

- Code **must** be checked in, clean repo
- Data **must** be immutable, versioned
- Environment **must** be a container
- Experiments can't run in **notebooks** (but read on...)

Aeromancy's stack

- Aeromancy wraps Weights & Biases
- Data stored in **S3-like blobstore** (e.g., ceph)
- Version control by git
- Virtual environments by PDM

Want reproducible data pipelines and machine learning?



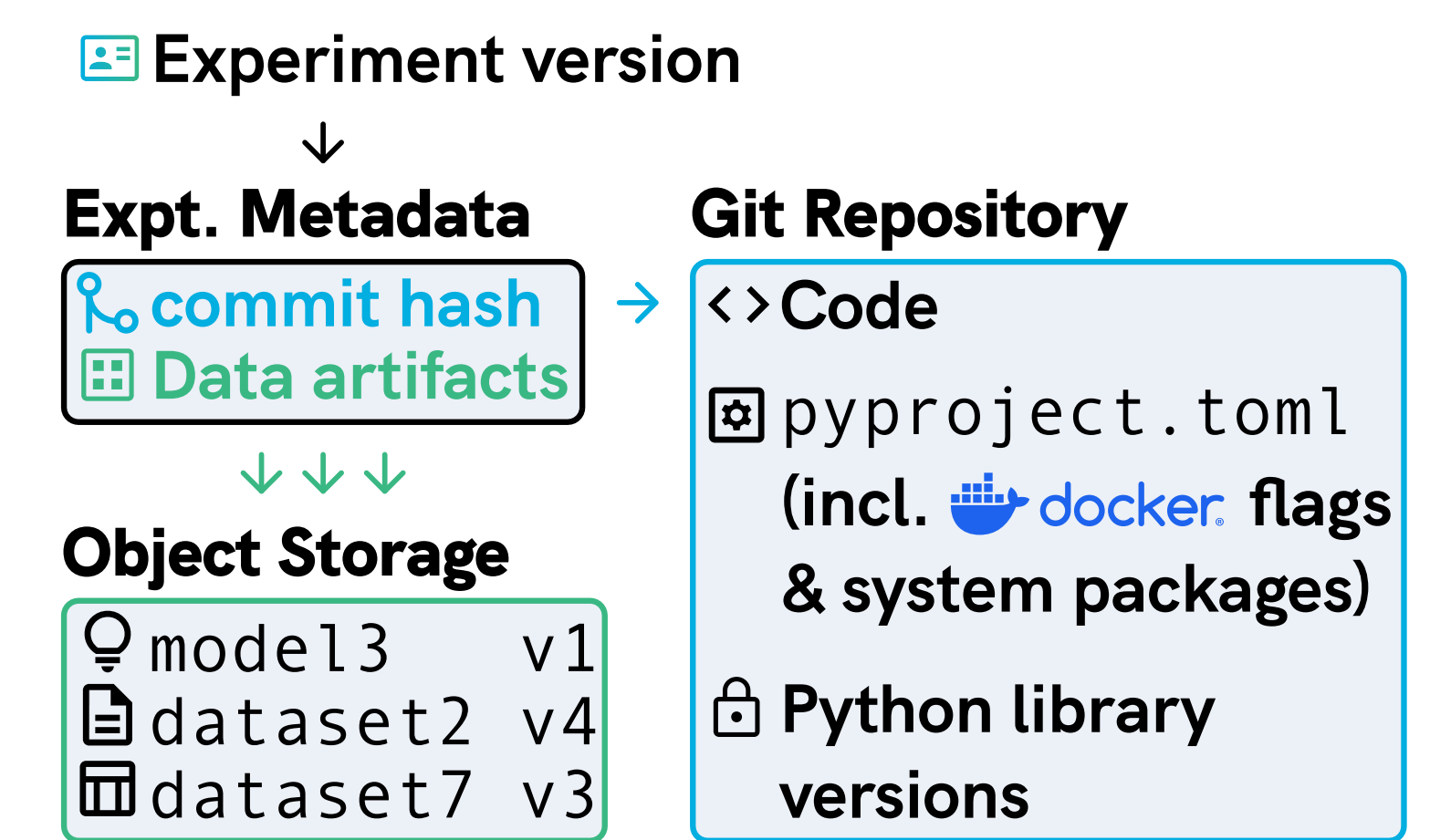
Track your whole environment, not just your code.



Take a picture to see Aeromancy on GitHub:
<https://github.com/quant-aq/aeromancy>

Versions are nested

- One version to track them all:



Docker details

- Record system package versions while building container
- Support hooks for installing system packages, custom Docker flags
- Optimized for fast rebuilds
- Special data/ folder for dataset ingestion (once per external dataset)

Reducing friction

- One command to rebuild container and launch experiments
- dev mode allows fast, offline iteration outside of containers (untracked)
- Cache data artifacts locally to save network traffic (deduplicated by hash)
- Rerun existing experiments with a single command

What about notebooks?

- Flexibility in notebooks makes it tough to fully track them
- Tracking notebooks requires backend support or disciplined use
- Still great for analysis and plots, so Aeromancy provides tools to import data/models in notebooks

Future work

- Expand integrations (e.g., MLFlow)
- Directly integrate into Experiment Managers
- Make it trivial to self-host object storage and connect with Aeromancy

