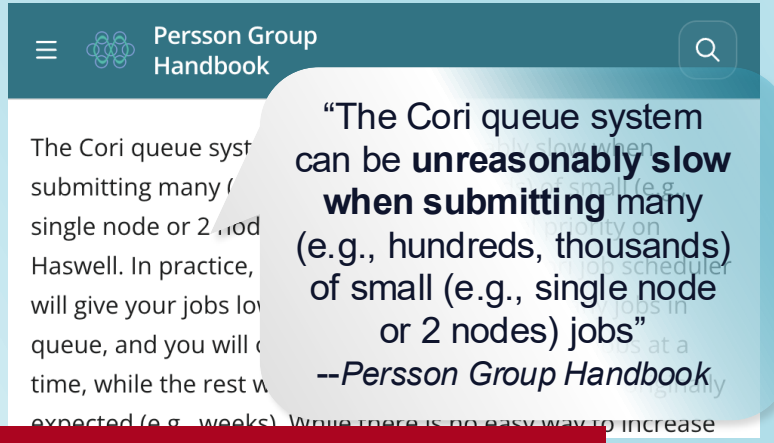
# Autonomous Science Is Inheriting a Workflow Infrastructure Problem

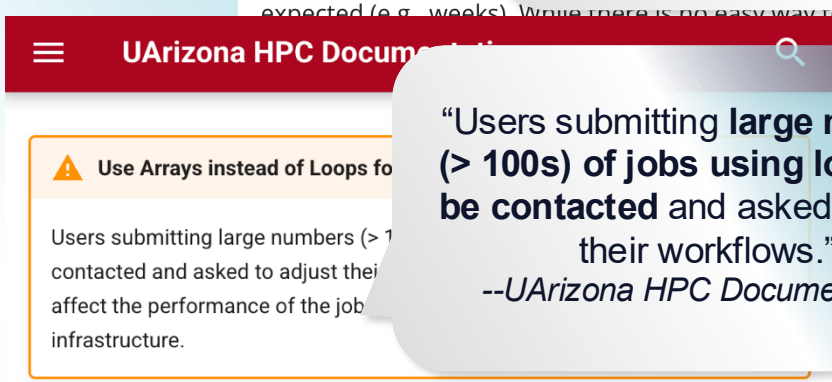
Glenn K. Lockwood, Ph.D.
Principal Technical Strategist
VAST Data

# “Please don’t submit too many jobs”

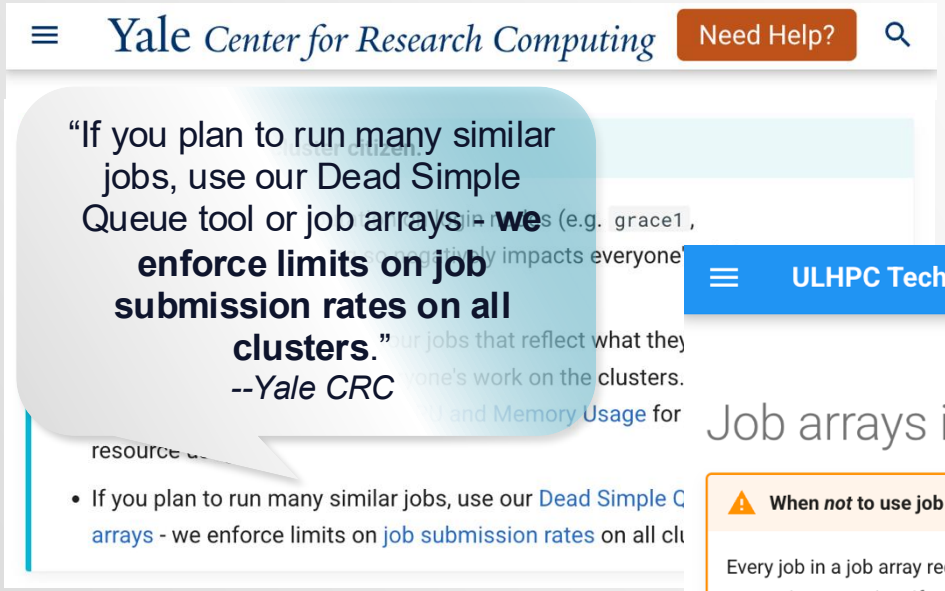
Humans are slow, and Slurm was designed for humans.



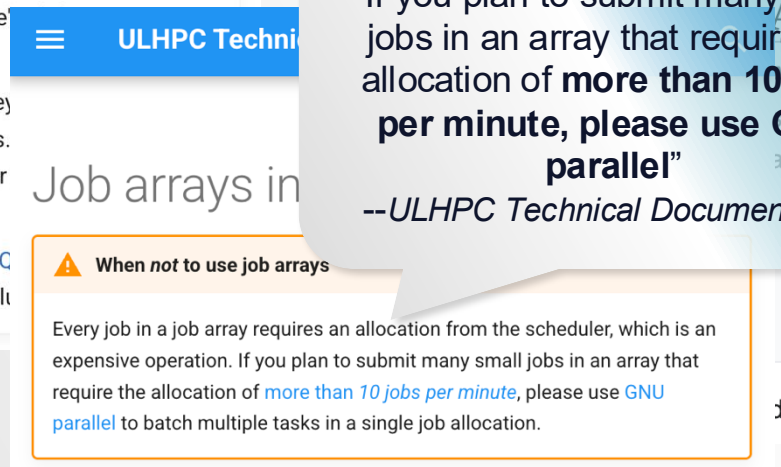

“The Cori queue system can be **unreasonably slow when submitting** many (e.g., hundreds, thousands) of small (e.g., single node or 2 nodes) jobs”
*--Persson Group Handbook*



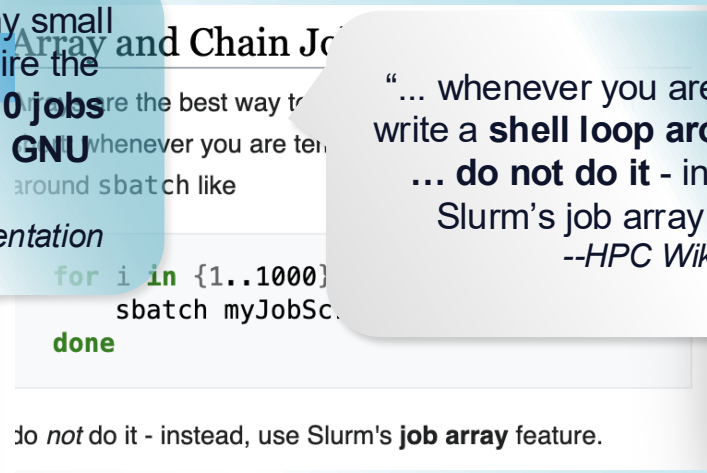

“Users submitting **large numbers (> 100s) of jobs using loops will be contacted** and asked to adjust their workflows.”
*--UArizona HPC Documentation*




“If you plan to run many similar jobs, use our Dead Simple Queue tool or job arrays - **we enforce limits on job submission rates on all clusters**.”
*--Yale CRC*



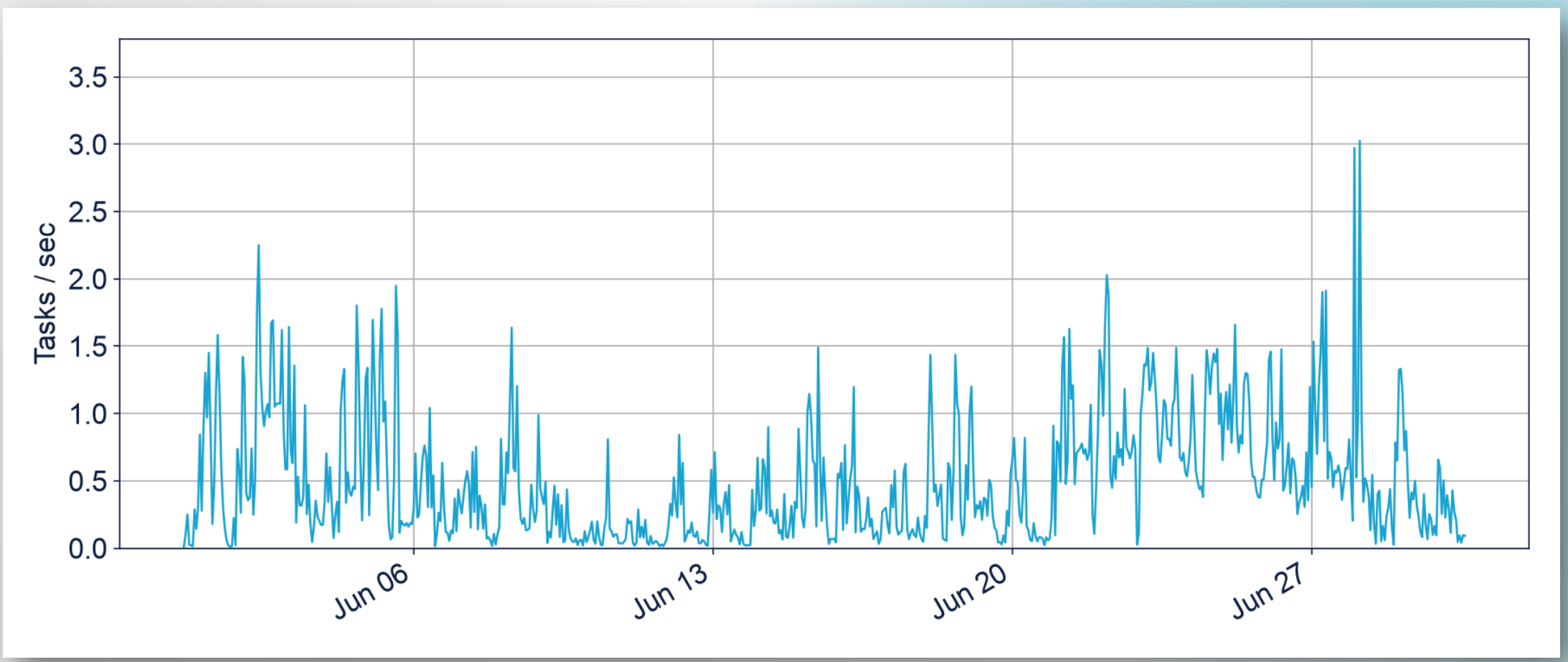

“If you plan to submit many small jobs in an array that require the allocation of **more than 10 jobs per minute, please use GNU parallel**”
*--ULHPC Technical Documentation*




“... whenever you are tempted to write a **shell loop around sbatch ... do not do it** - instead, use Slurm’s job array feature.”
*--HPC Wiki*

# Humans issue tasks in human-like ways

R-CCS Fugaku, June 2023 job submission traces (539 humans)

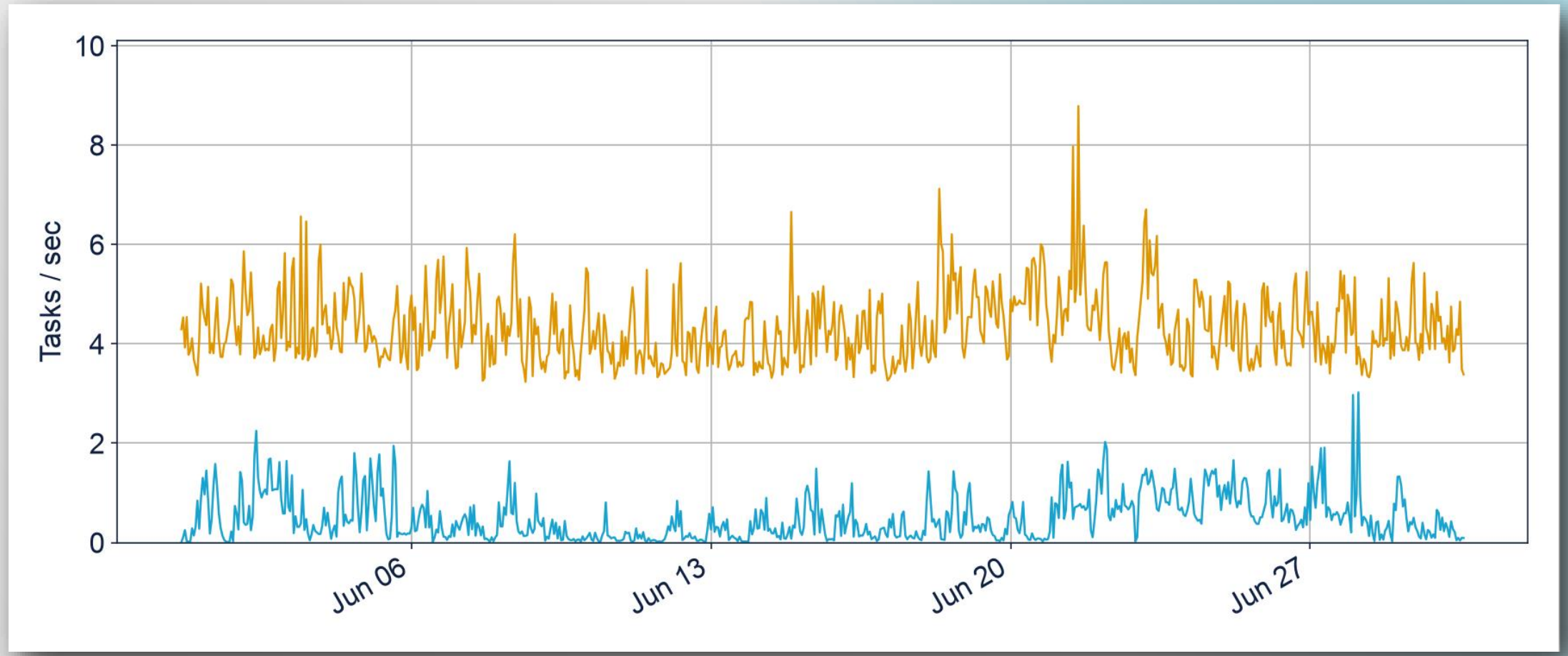





Antici et al. (2025). F-DATA: A Fugaku Workload Dataset for Job-centric Predictive Modelling in HPC Systems. *Scientific Data*, 12, 1321. https://doi.org/10.5281/zenodo.11467483

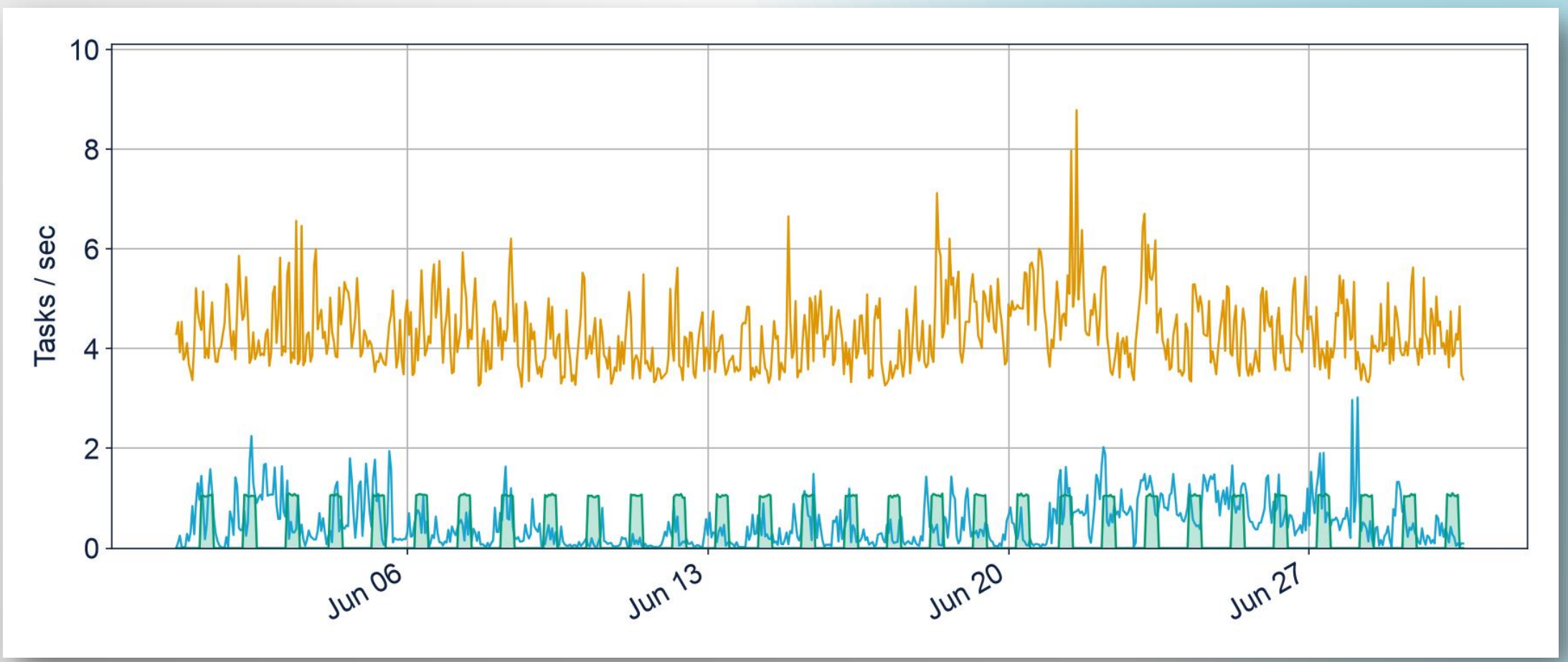
# Agents squeeze out human idle time

Modeled using Fugaku job submission traces (539 agents)



# Observational data increases baseline

Modeled from Zwicky Transient Factory (one 600 megapixel camera)

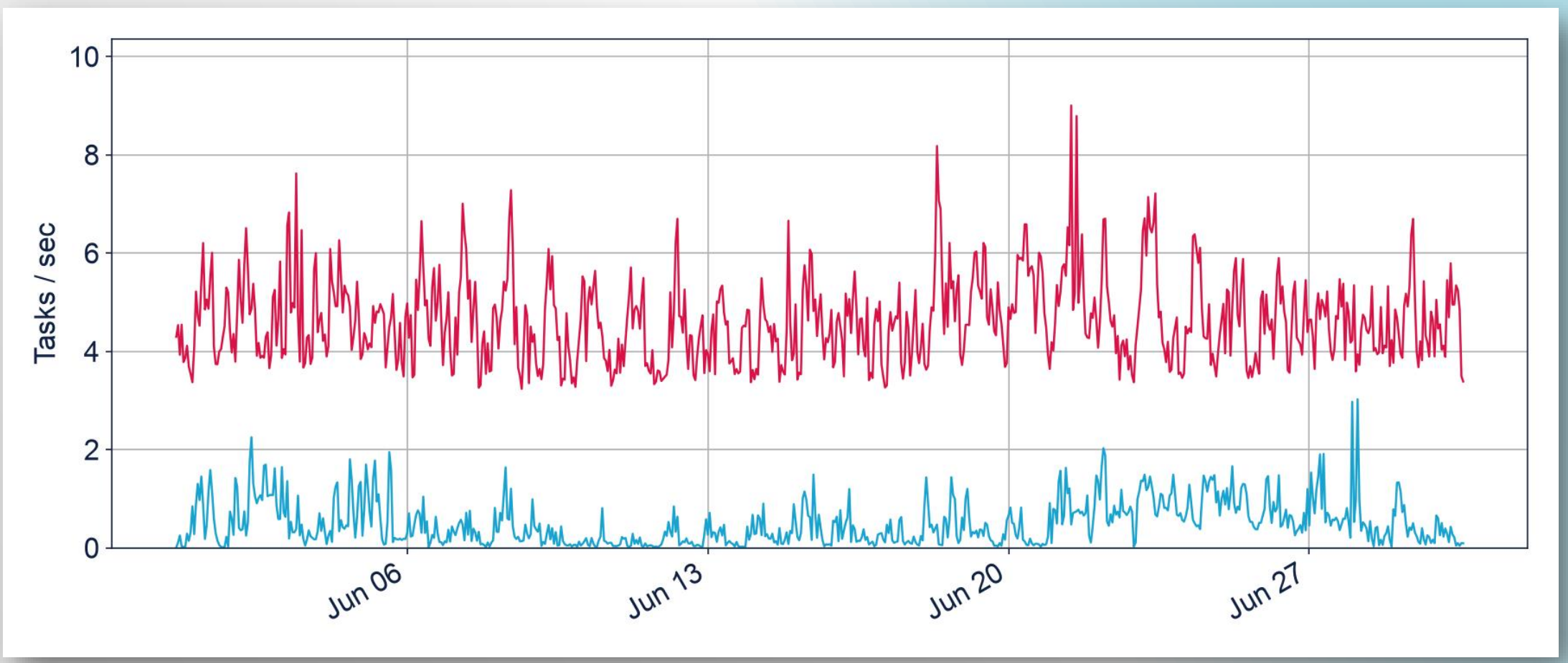





Laher et al. (2018). Processing Images from the Zwicky Transient Facility. In Robotic Telescope, Student Research and Education Proceedings, 329–36. https://doi.org/10.32374/rtsre.2017.031.

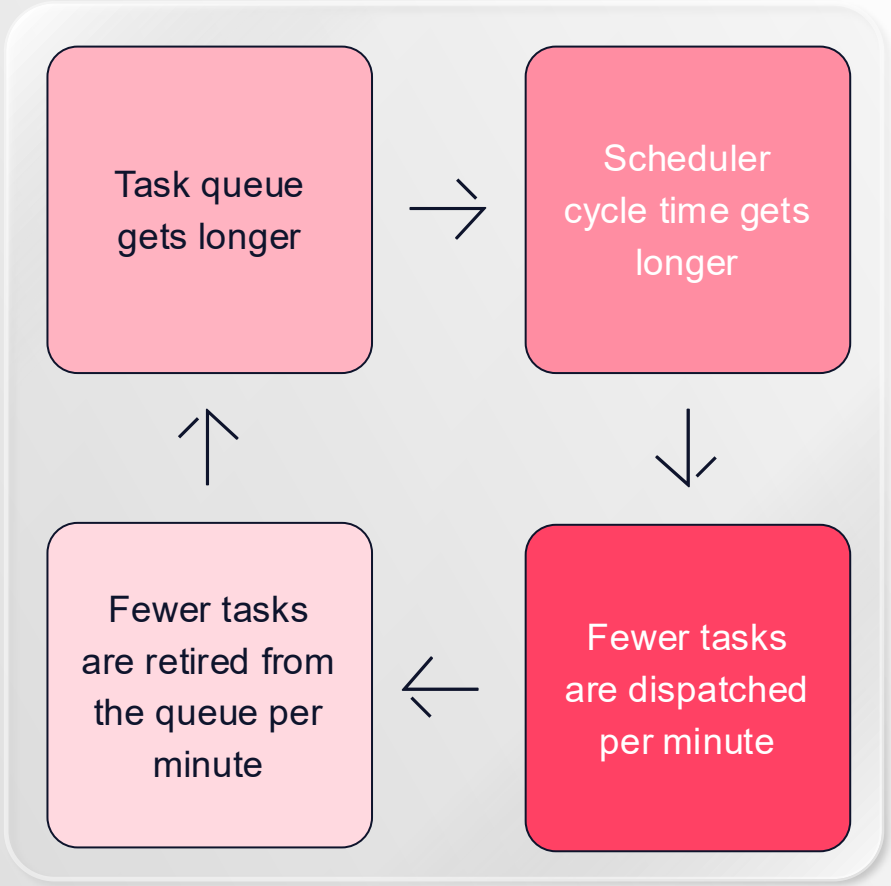
# Autonomous labs: 9x more tasks than leading HPC centers

Agents and instruments could average 16,000 job submissions per hour

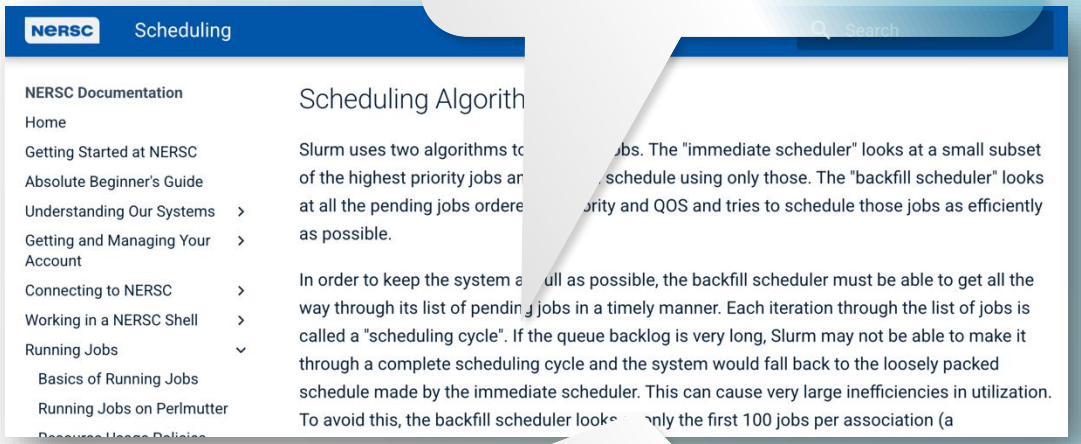


# Centralized task orchestration will become the bottleneck

Batch scheduling is at odds with real-time, high-throughput tasking of autonomous systems




"If the queue backlog is very long, **Slurm may not be able to make it** through a complete scheduling cycle …"


NERSC Scheduling

NERSC Documentation
Home
Getting Started at NERSC
Absolute Beginner's Guide
Understanding Our Systems
Getting and Managing Your Account
Connecting to NERSC
Working in a NERSC Shell
Running Jobs
Basics of Running Jobs
Running Jobs on Perlmutter

Scheduling Algorith

Slurm uses two algorithms to …bs. The "immediate scheduler" looks at a small subset of the highest priority jobs ar… schedule using only those. The "backfill scheduler" looks at all the pending jobs ordere… …rity and QOS and tries to schedule those jobs as efficiently as possible.

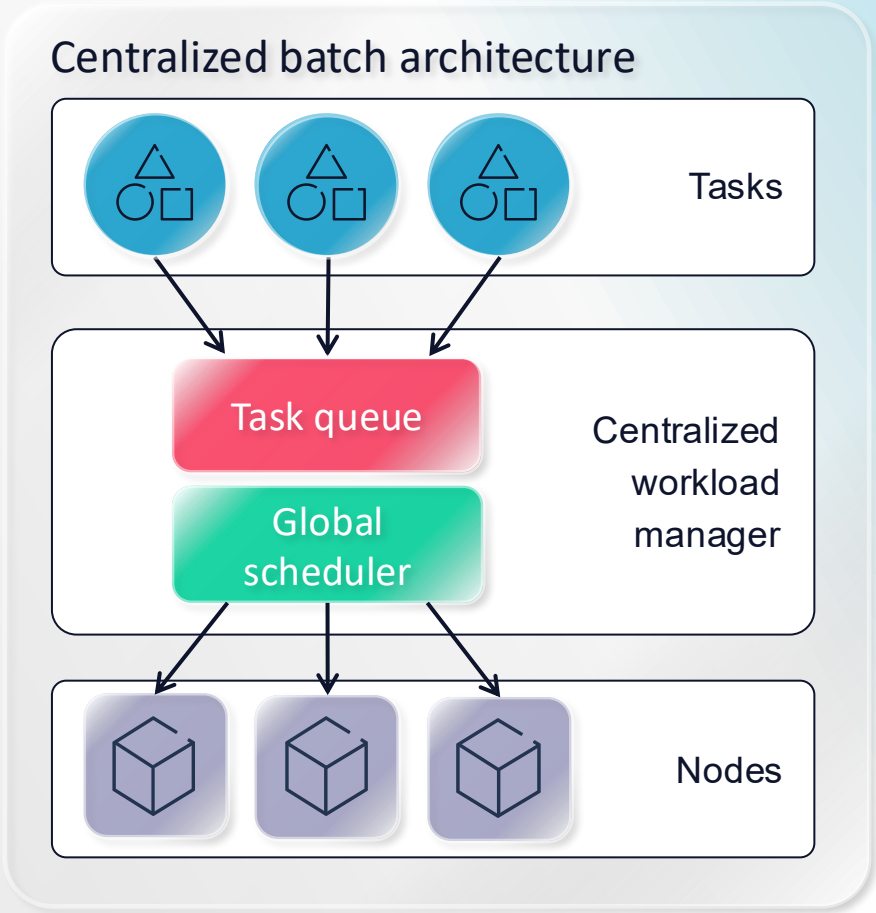
In order to keep the system a… …ull as possible, the backfill scheduler must be able to get all the way through its list of pendin… jobs in a timely manner. Each iteration through the list of jobs is called a "scheduling cycle". If the queue backlog is very long, Slurm may not be able to make it through a complete scheduling cycle and the system would fall back to the loosely packed schedule made by the immediate scheduler. This can cause very large inefficiencies in utilization. To avoid this, the backfill scheduler looks… …nly the first 100 jobs per association (a

Scheduling - NERSC Documentation. https://docs.nersc.gov/jobs/scheduling/

"This can cause **very large inefficiencies in utilization**."

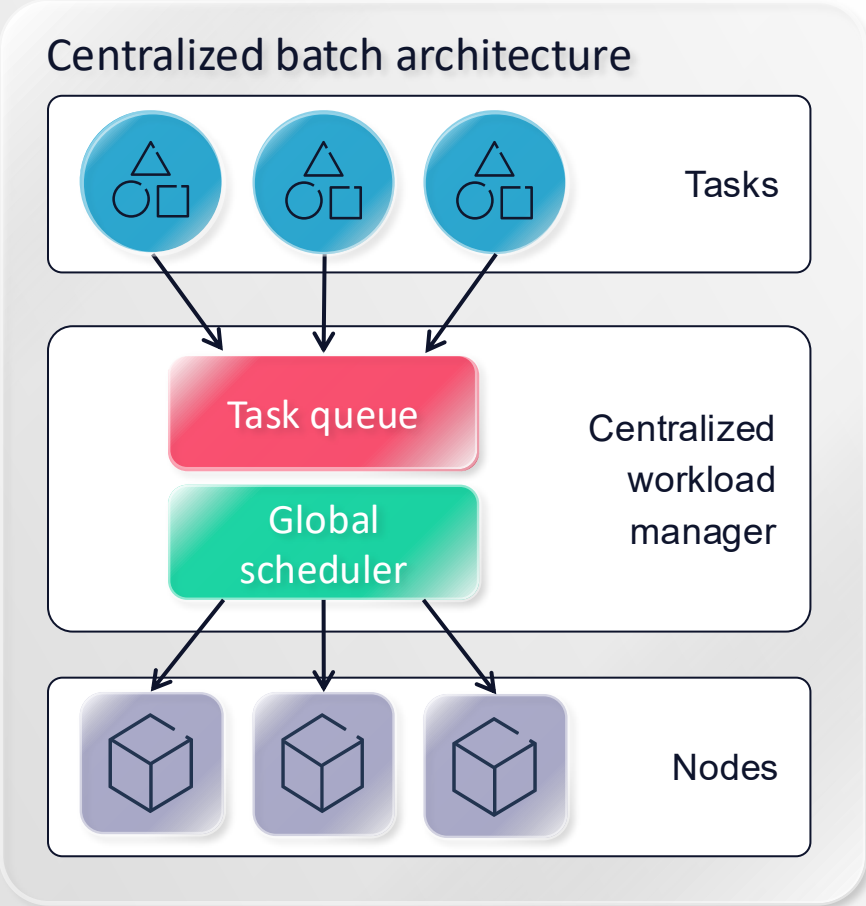
# Centralized task orchestration is fundamentally not scalable

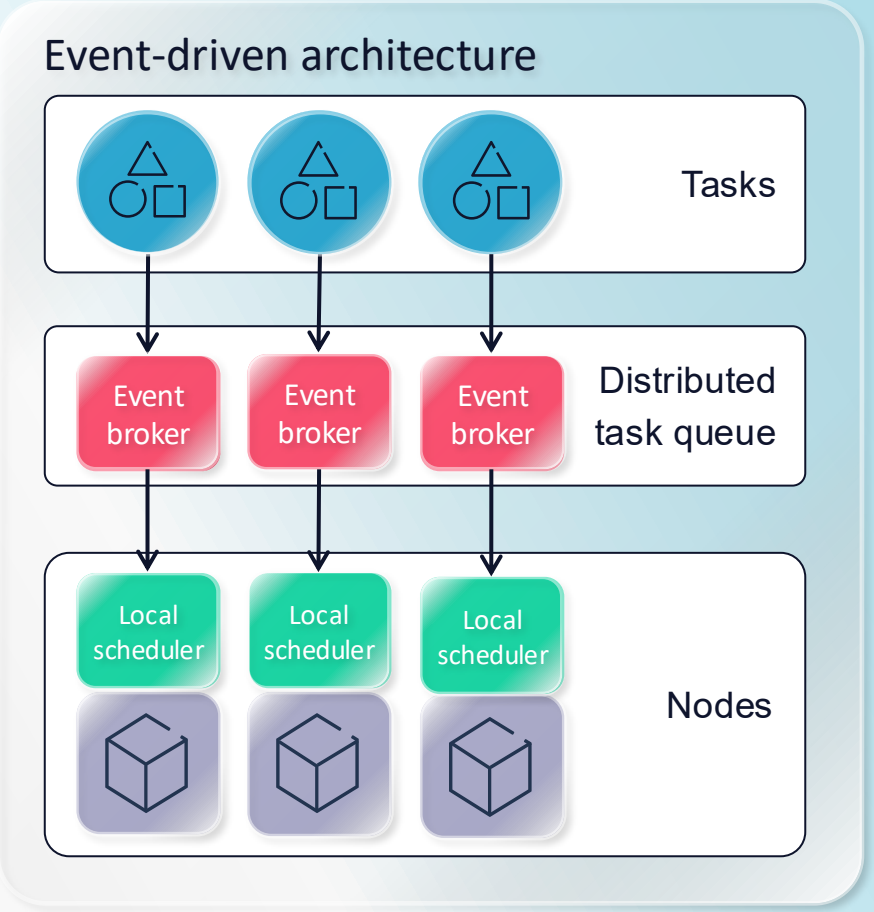
Scheduling is NP-hard and gates task ingestion and dispatch



# Event-based orchestration is fundamentally scalable

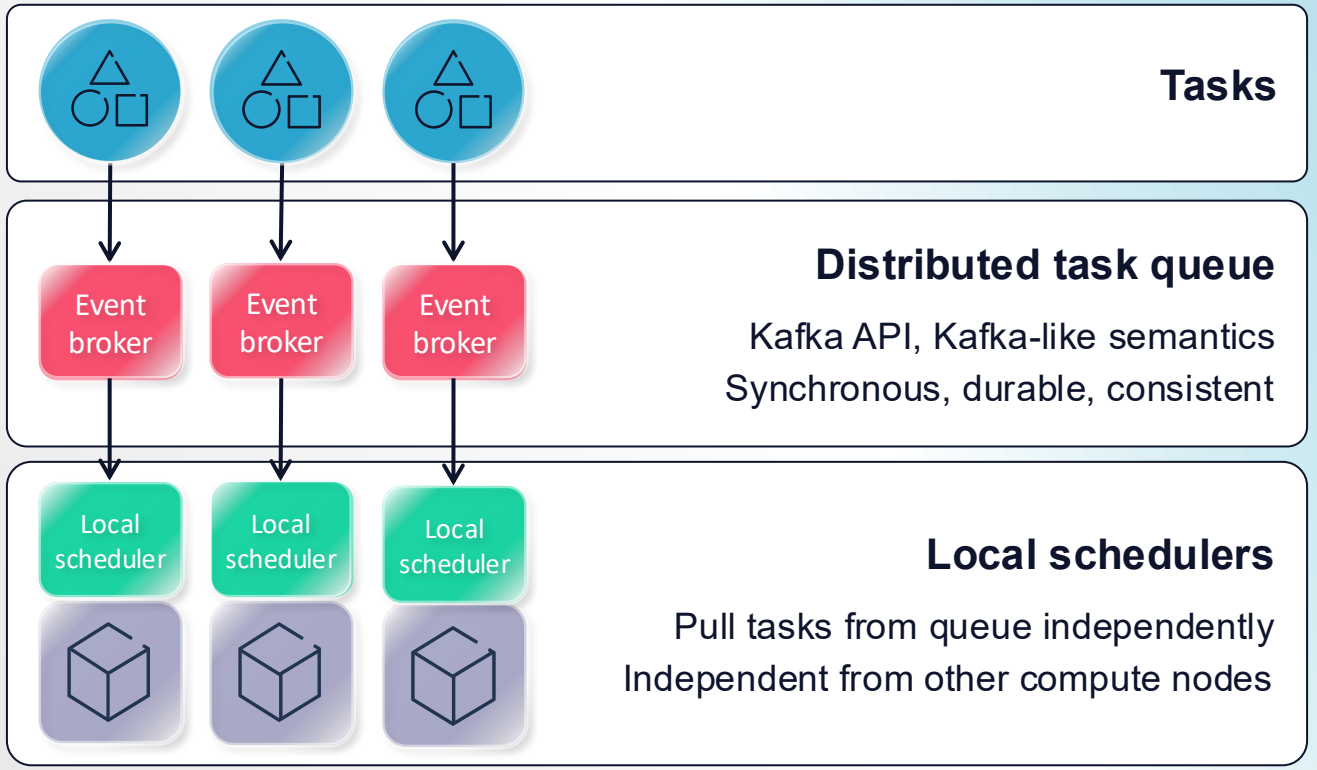
Nodes pull work from a durable queue, separating task ingest from dispatch



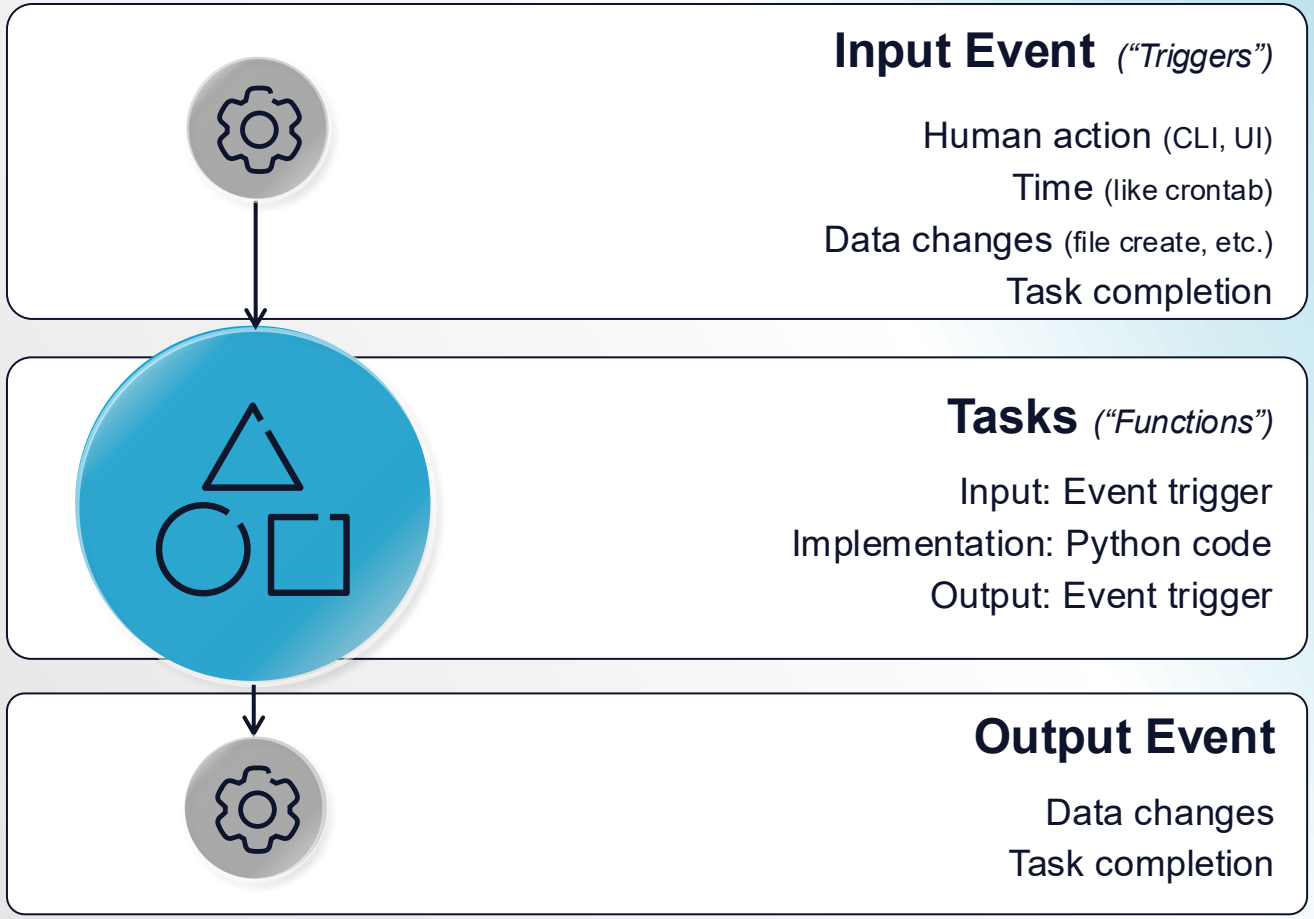
# Event-based orchestration is fundamentally scalable

Nodes pull work from a durable queue, separating task ingest from dispatch



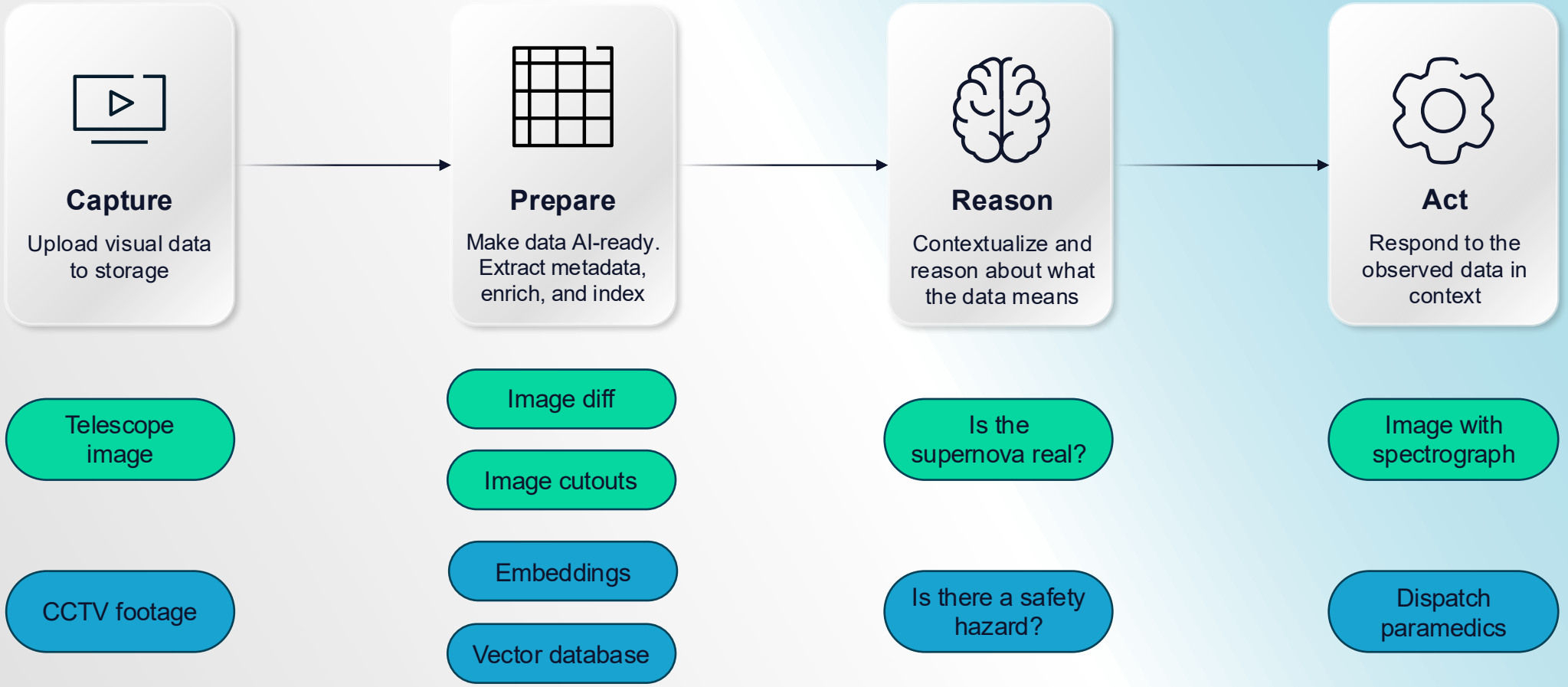
# Tasks compose into workflows through triggers

Tasks are created in response to triggers and emit triggers upon completion



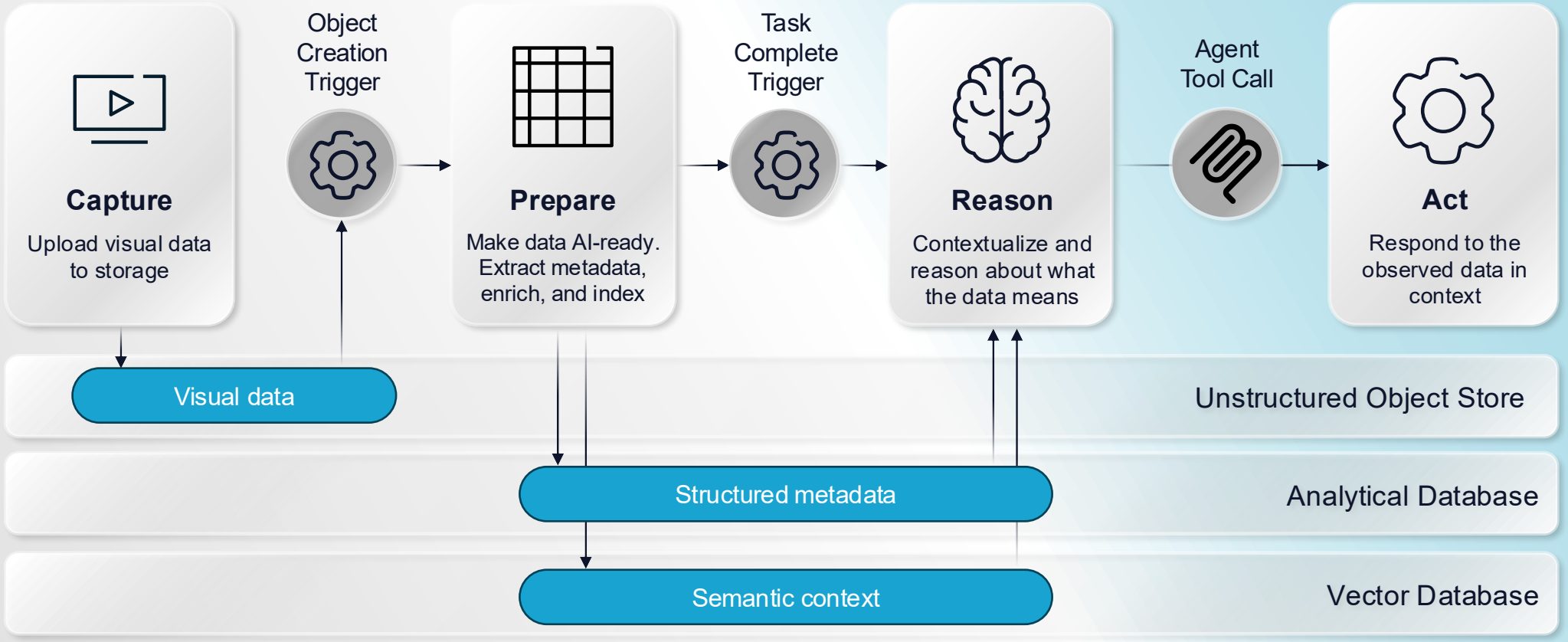
# Autonomous visual reasoning follows a common pattern

Astrophysical transient detection and autonomous public safety monitoring are computationally similar



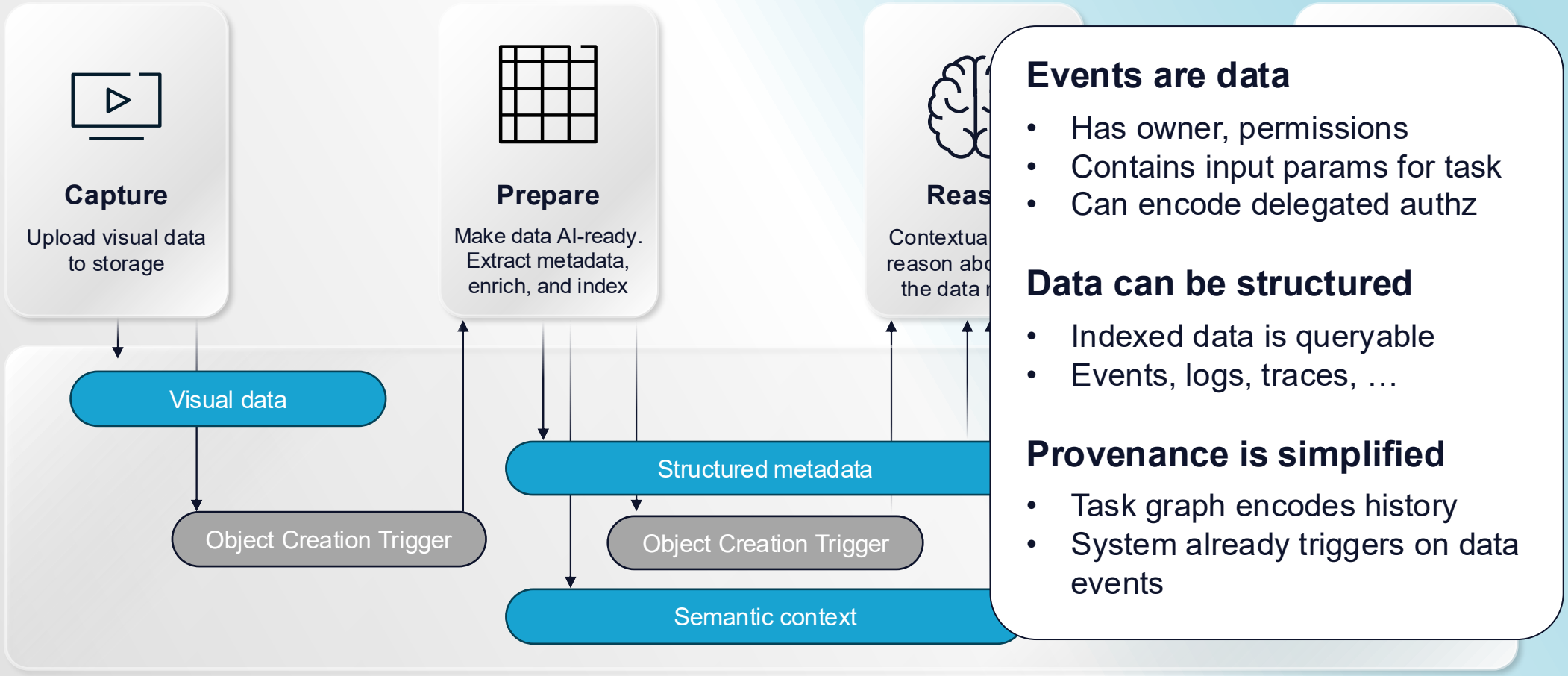
# Visual reasoning maps well to event-driven orchestration

Same pipeline and same task code scales from one sensor to tens of thousands



# Events are data too

Tasks and events can be treated as data, simplifying observability, provenance, governance



# The right tool for the right shape

Event-driven for the majority; centralized scheduling when task geometry demands it

## Event-driven architectures map to autonomous systems

- Many small tasks needing real-time responsiveness
- Observability, provenance, and governance automatically inherited by treating workflows as data

## But centralized orchestrators aren’t bad!

- Multi-node, topology-sensitive task placement
- Long-running, latency-insensitive tasks

## The ideal orchestrator combines both

- All tasks are event-driven, but
- Call out to a centralized orchestrator as-needed