



DAGs Move Robots:

Closed-Loop Orchestration for Silicon Validation Labs with Airflow



Dheeraj Turaga, Deva Madhavan & Shubham Rai



Dheeraj Turaga

- Principal Engineer @Qualcomm
- Apache Airflow Committer



Deva Madhavan

- Principal Engineer @Qualcomm
- Post-Si Methodology Lead



Shubham Raj

- Senior Engineer @Qualcomm
- Apache Airflow Committer

Where are we in relation to the community ?



Semiconductor Workflows ??

Qualcomm

The early adopter

Airflow for AI

What sort of use cases do you use Airflow for? (n=5327, multiple answers possible)



Source: <https://airflow.apache.org/blog/airflow-survey-2025/>

Presenter: Shubham Raj

Qualcomm | Oryon™ CPU

Powering the technologies you use every day



COMPUTE

PCs powered by
Snapdragon® X Series platforms

Snapdragon X2 Series



Snapdragon X Series



Multi-day
battery life



Blazing
performance



Built for
AI experiences



MOBILE

Premium smartphones
and tablets

Snapdragon 8 Series



Snapdragon 7 Series



AI-powered
experiences



Leading
connectivity



Ultimate
performance



AUTOMOTIVE

Intelligent cockpit and
advanced driver assistance

Snapdragon Cockpit Elite



Snapdragon Ride Elite



Advanced driver
assistance



Immersive
experiences



Safety &
reliability



DATA CENTER

Servers and cloud
infrastructure at scale

Qualcomm Dragonfly



Custom CPU for data centers
built on Oryon technology



High core
density &
utilization



Best-in-class
performance
per watt



Optimized for
cloud & AI
workloads



Apache
Airflow

Powers the automation behind designing every Oryon CPU

Apache Airflow orchestrates thousands of complex engineering workflows
across a global, distributed infrastructure.



DV Regressions
& Test Orchestration



Emulation
Workflows



Post-Silicon
Validation



RTL & EDA
Flows



Multi-DC
Orchestration



Scalable, Reliable
& Secure



Millions of workloads. Thousands of servers. One orchestration platform.

Apache Airflow.

Airflow Adoption 2025 - present

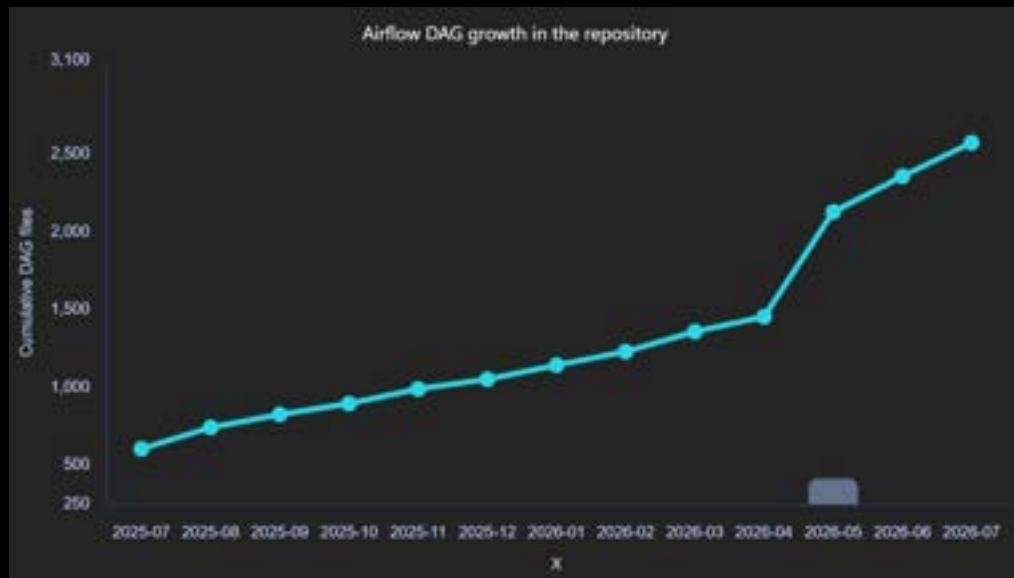
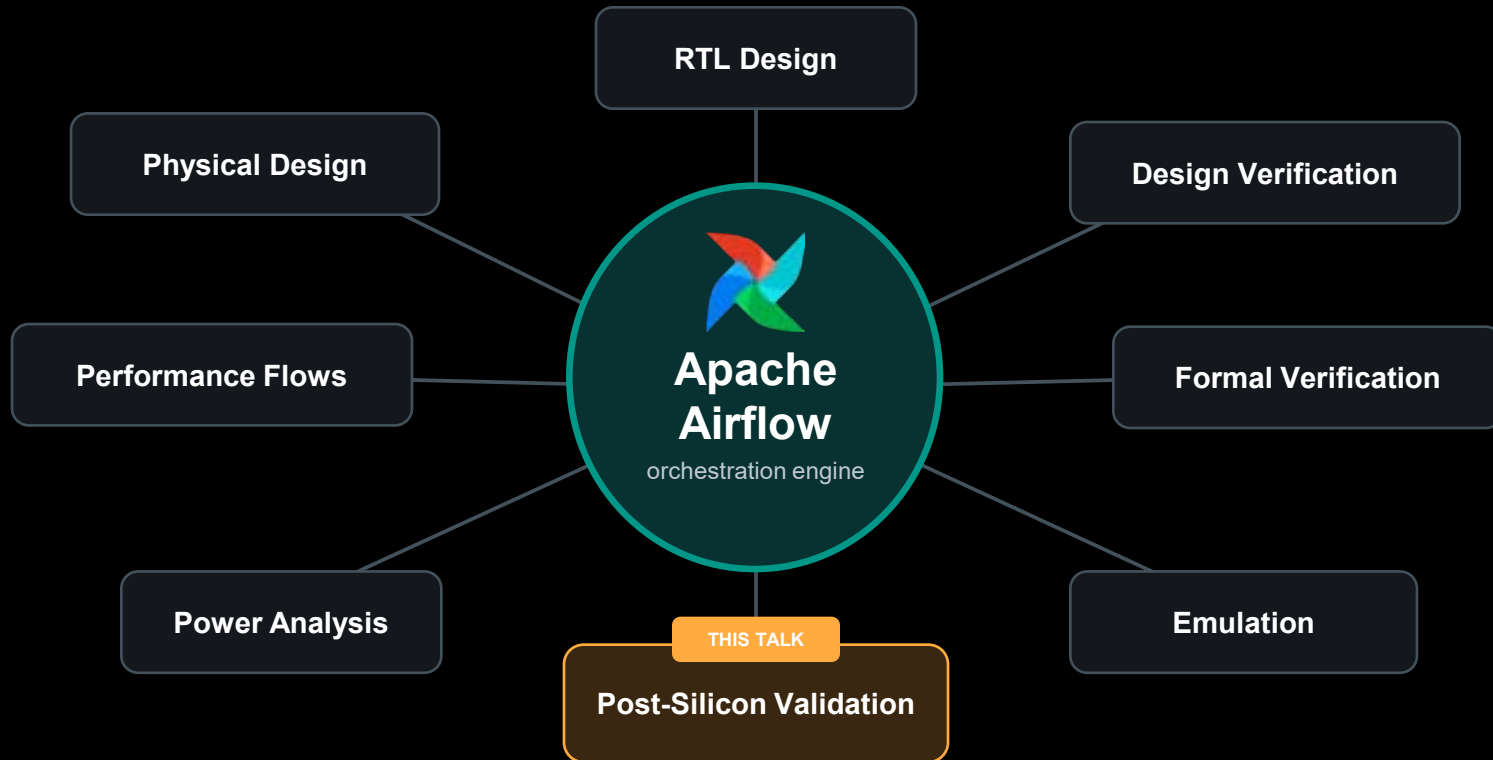


Image: Dags growth in last one year

Object	Count
Total Dags	2500+
Total Dag Runs	265K
Total Task Instances	4.5+ Million

One orchestration engine

Shared Infrastructure for Every phase of CPU design



THE PIVOT



Before silicon, DAGs move data.

- Compute jobs
- RTL + netlists
- Simulation results
- Software-defined resources

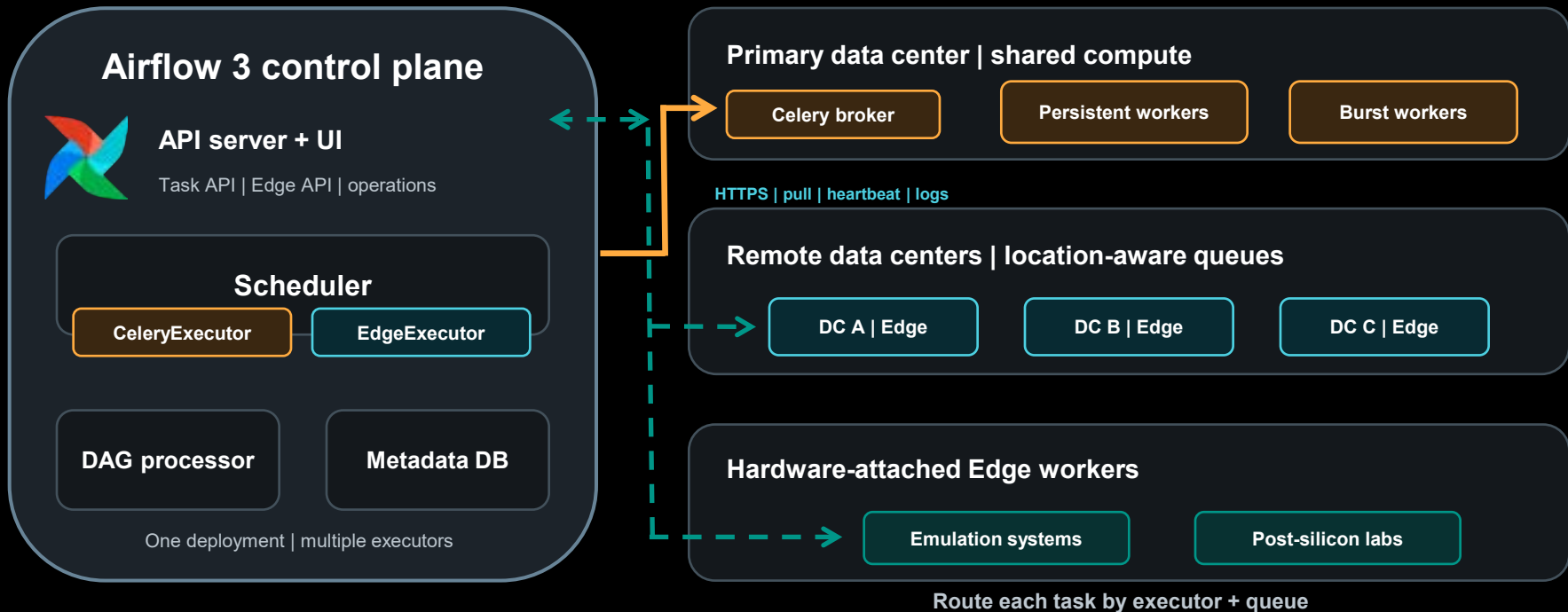
After silicon, DAGs move the physical world.

- Robotic handlers
- Physical devices
- Thermal + power systems
- Scarce, stateful equipment

What changes when an Airflow task has a physical consequence?

One Airflow 3 control plane.

Workloads everywhere. Distributed execution fabric.





Post Silicon Validation

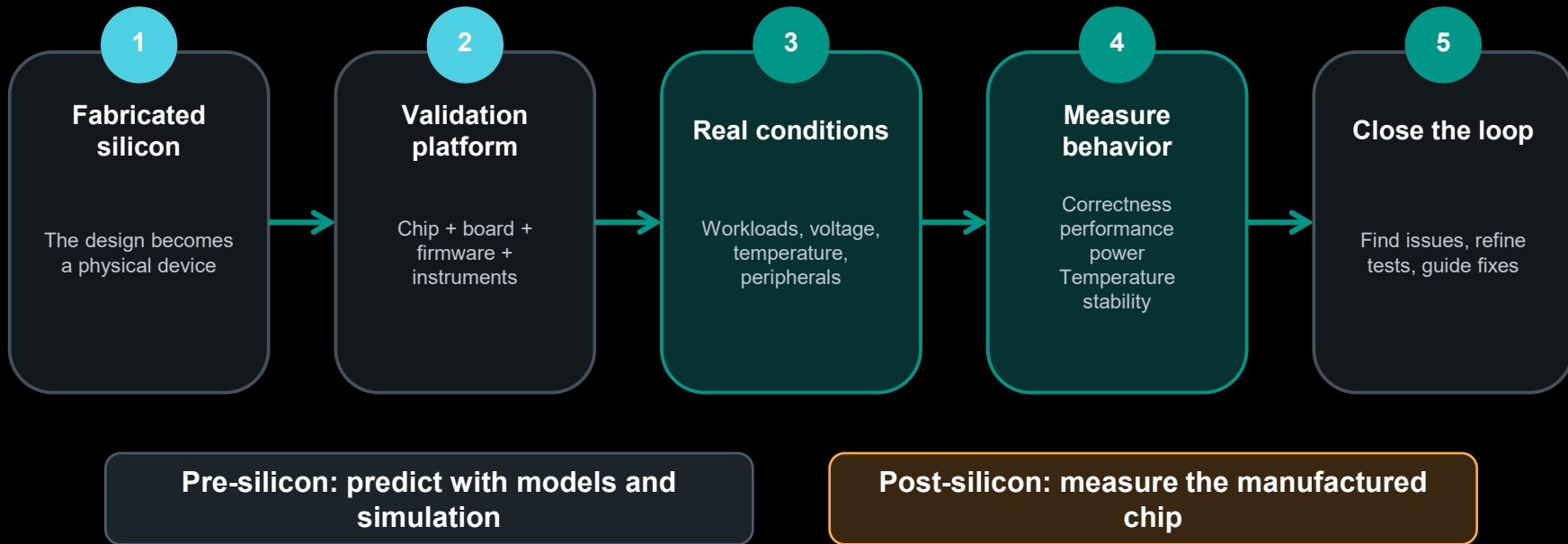
From Design intent to Real-World delivery of high-Performance
Silicon

Presenter: Deva Madhavan

Post-silicon validation



Characterizing and Optimizing Silicon for Real-World operation
Performance • Power • Temperature • Stability • Aging



The goal: build confidence that the chip behaves correctly in the real world.

POST-SILICON VALIDATION

Characterizing and Optimizing Silicon for Real-World Operation

Performance • Power • Temperature • Reliability • Longevity



MEASUREMENT & ANALYSIS UNITS

POWER MEASUREMENT



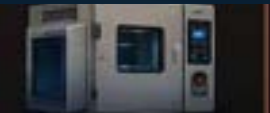
Voltage • Current • Energy

FREQUENCY / TIMING



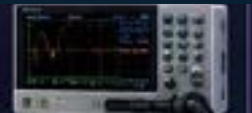
Clock • Timing • Performance

TEMPERATURE CONTROL



Thermal chamber • Heat

DEBUG & TRACE



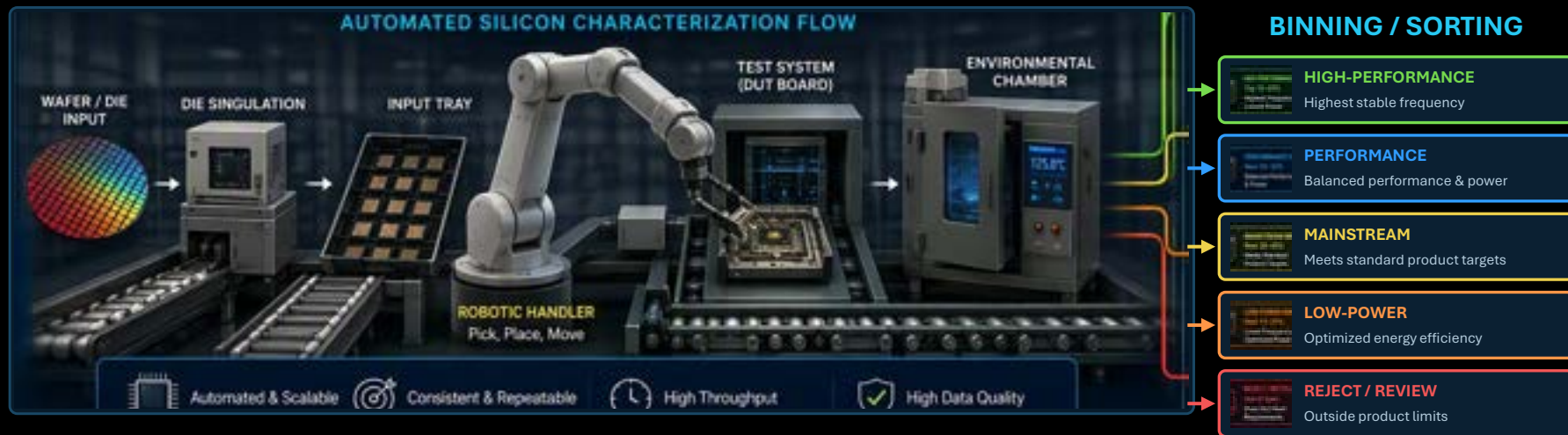
JTAG • Logs • Visibility

DATA & ANALYTICS



Capture • Analysis • Correlation

AUTOMATED SILICON CHARACTERIZATION FLOW



BINNING / SORTING

HIGH-PERFORMANCE

Highest stable frequency

PERFORMANCE

Balanced performance & power

MAINSTREAM

Meets standard product targets

LOW-POWER

Optimized energy efficiency

REJECT / REVIEW

Outside product limits

AUTOMATED & SCALABLE

CONSISTENT & REPEATABLE

HIGH THROUGHPUT

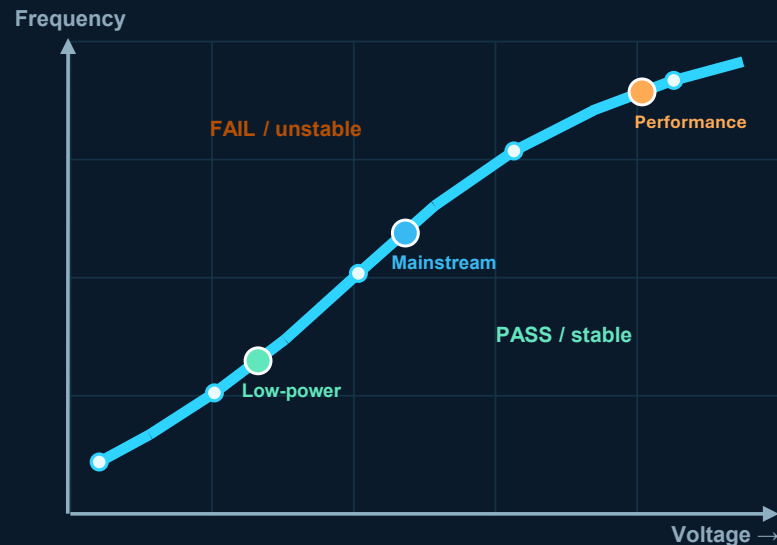
HIGH-QUALITY CHARACTERIZATION DATA

V/F Curve Defines Silicon Performance per Watt — and the Bin

Characterize each die by sweeping voltage and frequency, then classify it by the stable operating boundary.



Map the stable operating VF boundary



VF curve is the Passing stable region for silicon performance
VF Guard-band moves the VF operating point into the stable region.

Sort silicon by measured V/F capability



Efficiency bin

Lowest V_{min}
for target F

Mainstream bin

Nominal V/F
and margin

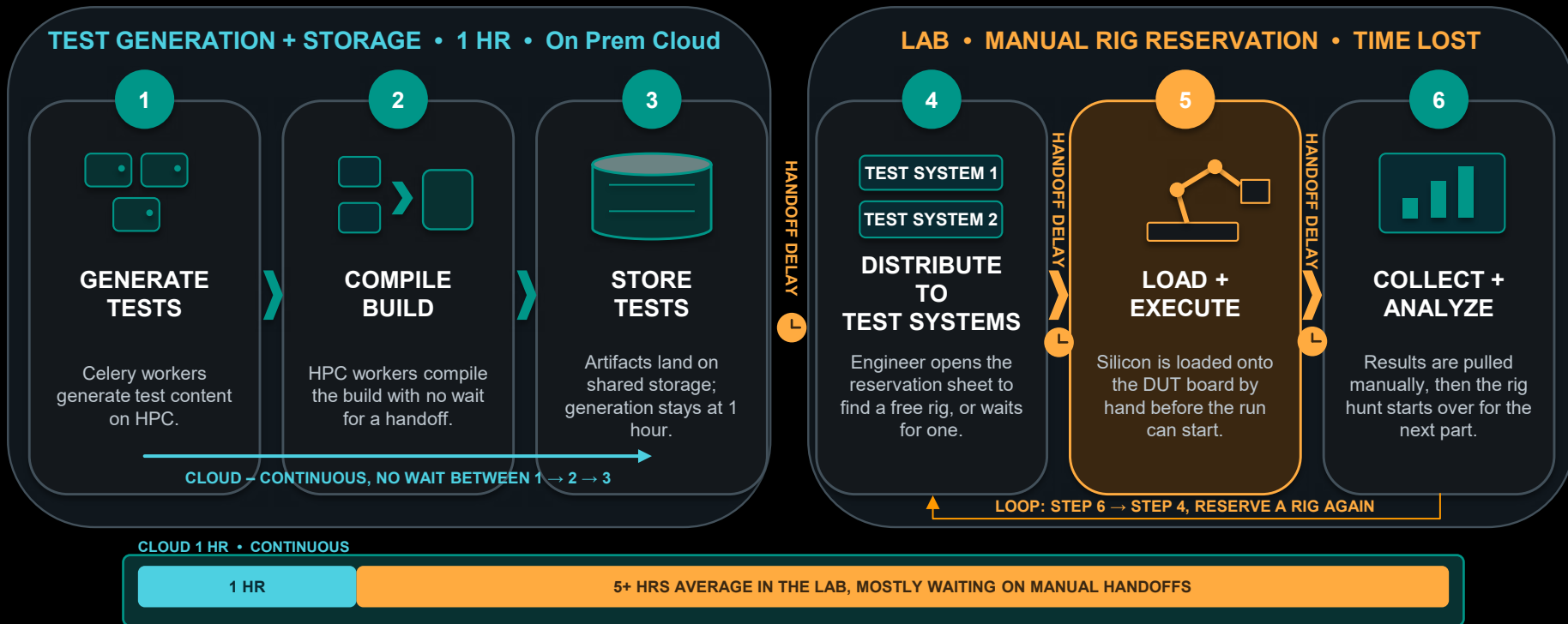
Performance bin

Highest F_{max}
at allowed V

Silicon Lab operations - Manual



Cloud steps run back to back; every lab step waits on a manual test system intervention



The Problem

- **HARDWARE ALLOCATION**

- Reservations are in spreadsheets, OneNote
- "is system 4 free?" pinged on Slack, Teams
- Sticky notes read "Used By: Name"

- **TIME MANAGEMENT**

- Conflicts - Engineers step on same DUT
- Test runs collide or get overwritten
- Results Lost, Data Loss
- Idle gaps with critical units sit unused
- Time and Geo-location differences

- **OBSERVABILITY**

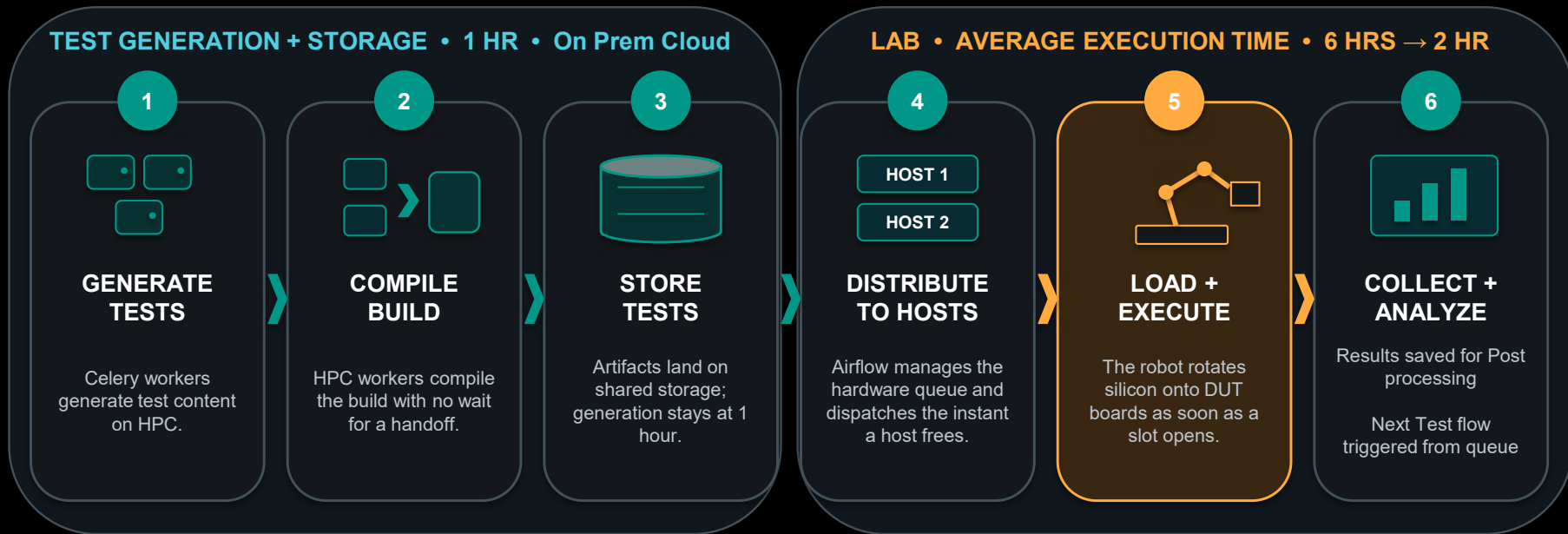
- What's running, where or for whom
- No history and track record of test coverage
- Data getting scattered and Lost
- Failures go undetected



Silicon Lab operations managed by Airflow DAGs



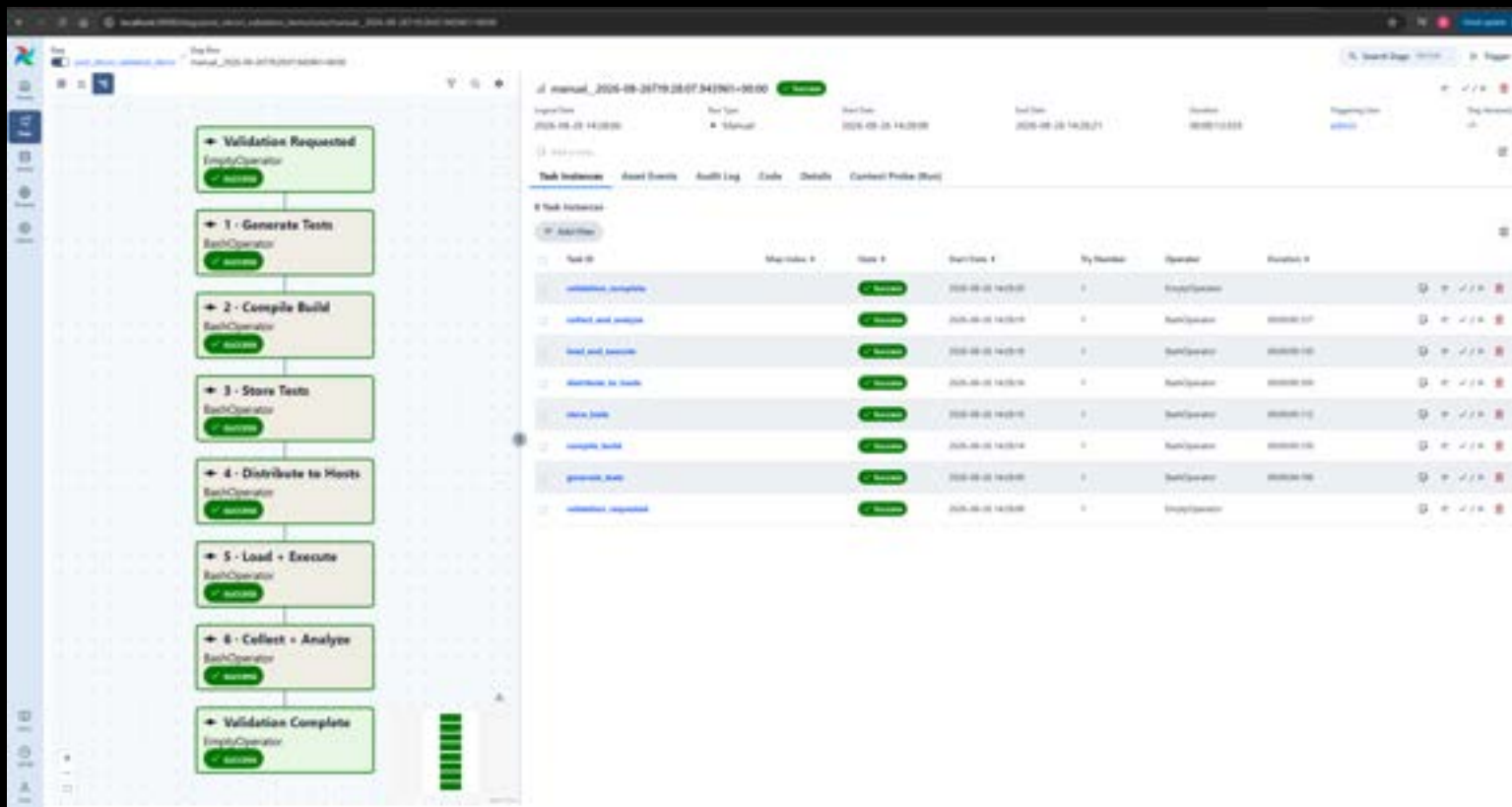
Managed hardware queues and continuous scheduling remove the waits between disconnected stages



AVERAGE 2 HRS PER CHIP, DOWN FROM 6 HRS • 3X VALIDATION THROUGHPUT • 66% TIME REDUCTION • UNIFIED DATA

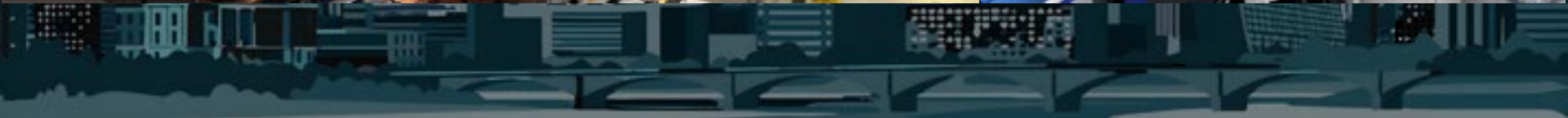
NEXT → ROBOT IN ACTION

Post-silicon validation Sample DAG





Lab Fleet of Robotic Arm Handlers





The Magic

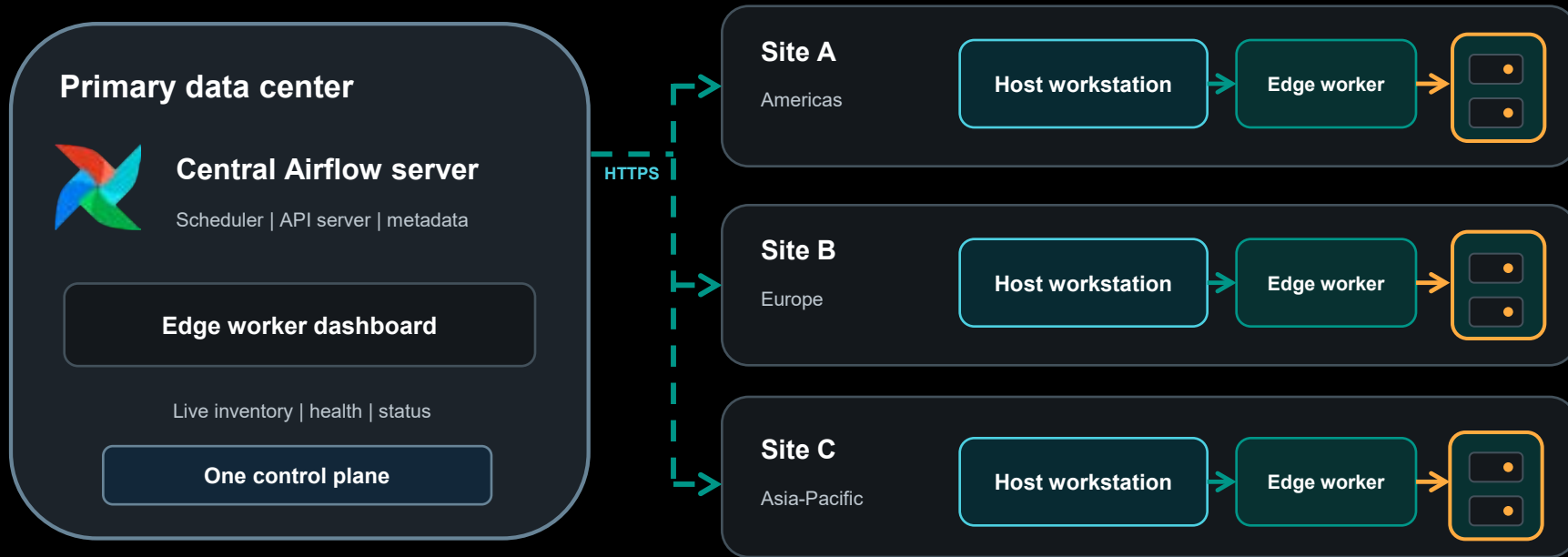
**Presenter: Dheeraj
Turaga**

Edge Executor controls

Labs around the world.



Distributed post-silicon labs



Host boots -> Edge worker starts -> registers -> reports health

One dashboard for lab fleet



localhost:20080/plugin/edge_executor/worker

Edge Worker Edge Jobs

Search workers Search queues All States

Worker Name	State	Queues	Active Jobs	System Status	Operations
HOST-42	idle	• SITE-A • HOST-42	0	Healthy	
HOST-56	starting	• SITE-B • HOST-56	0	Healthy	
HOST-71	starting	• SITE-A • HOST-71	0	Healthy	

- First online: 1 minute ago
- Last heartbeat: 1 minute ago
- Airflow version: 3.4.0
- Edge provider version: 4.1.0
- Python version: 3.10.20 (main, Apr 8 2026, 22:19:23) (GCC 12.2.0)
- Worker start time: 1 minute ago
- Concurrency: 8
- Free concurrency: 8

Operate Edge Workers from CLI or Dashboard



CLI AUTOMATION

Scriptable operations for local and remote workers

```
airflow edge --help
```

```
airflow edge [-h] COMMAND ...
Start and manage Edge Worker. Works only when using EdgeExecutor. For more information
see https://airflow.apache.org/docs/apache-airflow-providers-edge/stable/edge_executor.html.
```

```
Commands:
  add-worker-queues      Add queues to an edge worker.
  list-workers          Query the db to list all registered edge workers.
  maintenance           Set or unset maintenance mode of local edge worker.
  remove-edge-worker    Remove remote edge worker from maintenance.
  remove-edge-worker-request-maintenance  Put remote edge worker in maintenance.
  remove-edge-worker-update-maintenance-mode  Update maintenance mode of the remote edge worker.
  remove-remote-edge-worker  Remove remote edge worker entry from db.
  remove-worker-queues  Remove queues from an edge worker.
  set-worker-concurrency  Set the concurrency of a remote edge worker.
  shutdown-all-workers  Request graceful shutdown of all edge workers.
  shutdown-remote-edge-worker  Initiate the shutdown of the remote edge worker.
  status               Check for Airflow Local Edge Worker status.
  stop                 Stop a running local Airflow Edge Worker.
  worker              Start Airflow Edge Worker.
```

Lifecycle

worker • status • stop

Fleet

list • remove • shutdown

Maintenance

local + remote controls



DASHBOARD OPERATIONS

Point-and-click control for day-to-day administration



Add queue

Route new work to a worker

[add-worker-queues](#)

Available from the Edge Executor dashboard



Remove queue

Stop routing a queue

[remove-worker-queues](#)

Available from the Edge Executor dashboard



Concurrency

Adjust worker capacity

[set-worker-concurrency](#)

Available from the Edge Executor dashboard



Maintenance

Drain and service safely

[maintenance](#)

Available from the Edge Executor dashboard



Shutdown

Graceful worker stop

[shutdown-remote-worker](#)

Available from the Edge Executor dashboard

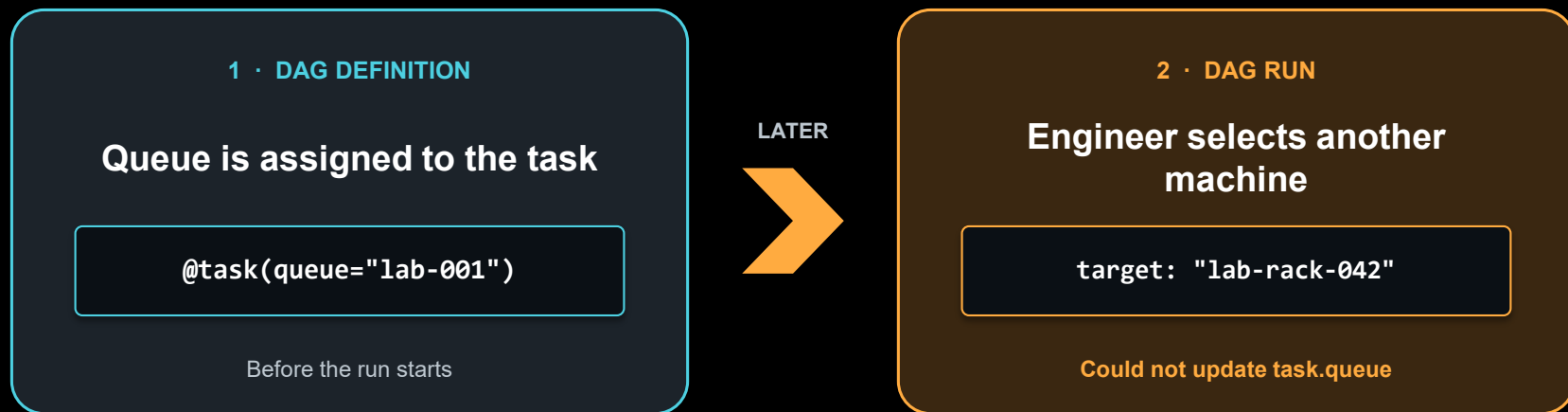
Same operational controls

→ automation via CLI + fast intervention via UI

Task queues had to be provided upfront



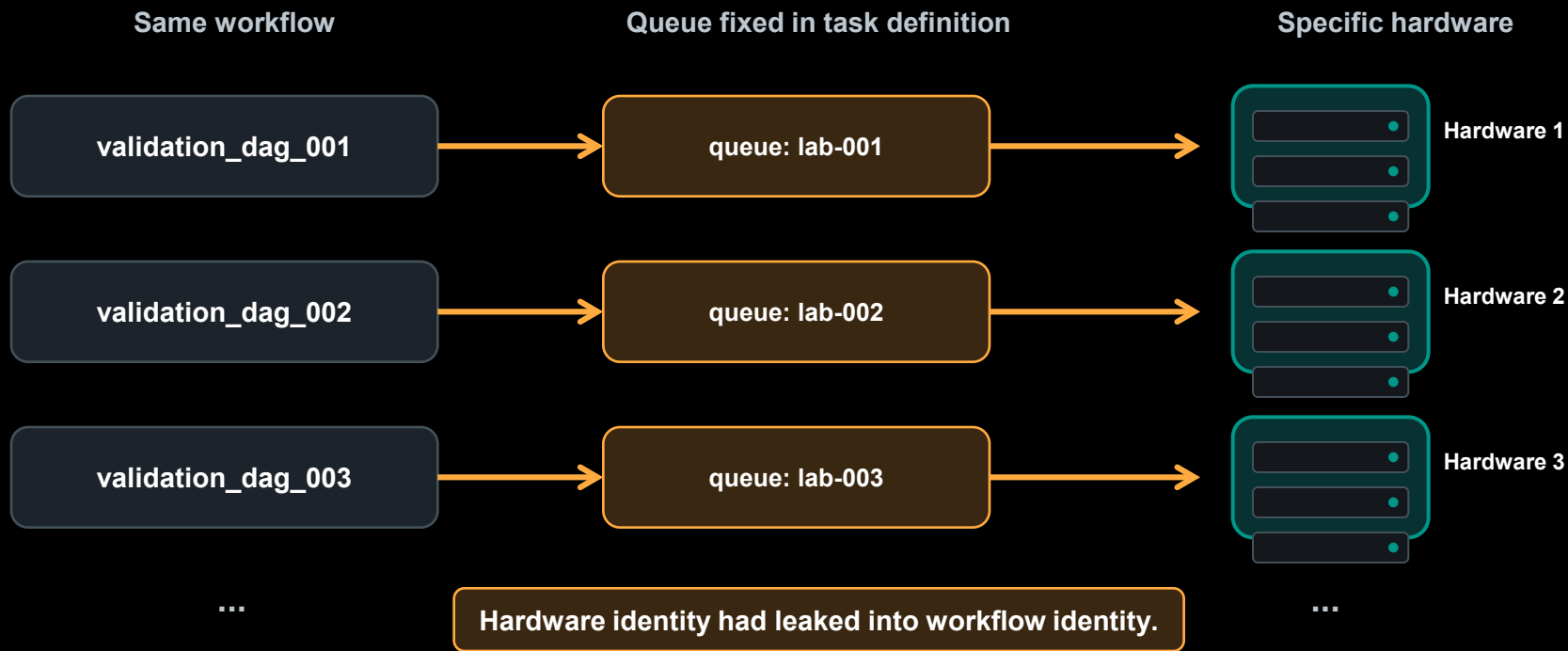
The run could choose a machine, but it could not change the task's queue.



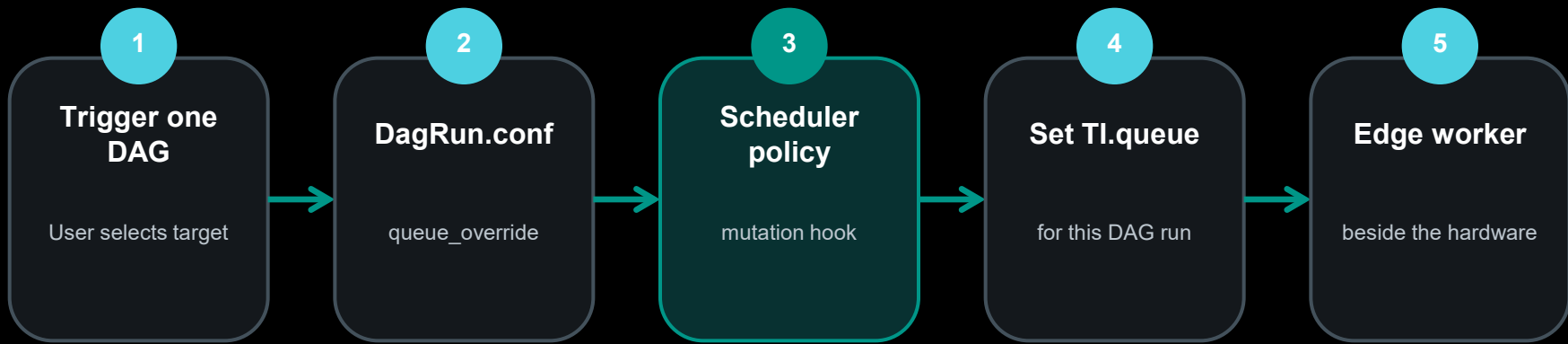
To use a different queue, the DAG had to be changed or duplicated.

At scale: one queue-specific DAG per machine →

The workaround: one DAG per machine



User chooses the Hardware not DAG



The DAG describes the methodology. The DAG run selects the machine.

Scheduler-side | before the task is queued

One hook. Run-aware routing and capacity.



Run-time context

```
{  
  "queue_override": "lab-rack-042",  
  "pool_override": "post_silicon_lab"  
}
```

QUEUE
where it runs

POOL
how many can run

Scheduler-side mutation

```
def task_instance_mutation_hook(  
    task_instance, dag_run=None  
):  
    if dag_run is None:  
        return  
  
    conf = dag_run.conf or {}  
    if queue := conf.get("queue_override"):  
        task_instance.queue = queue  
    if pool := conf.get("pool_override"):  
        task_instance.pool = pool
```

PR #68198 | Pass DagRun to task_instance_mutation_hook for run-aware task mutation

Lab needs became upstream Airflow capabilities



135

merged PRs

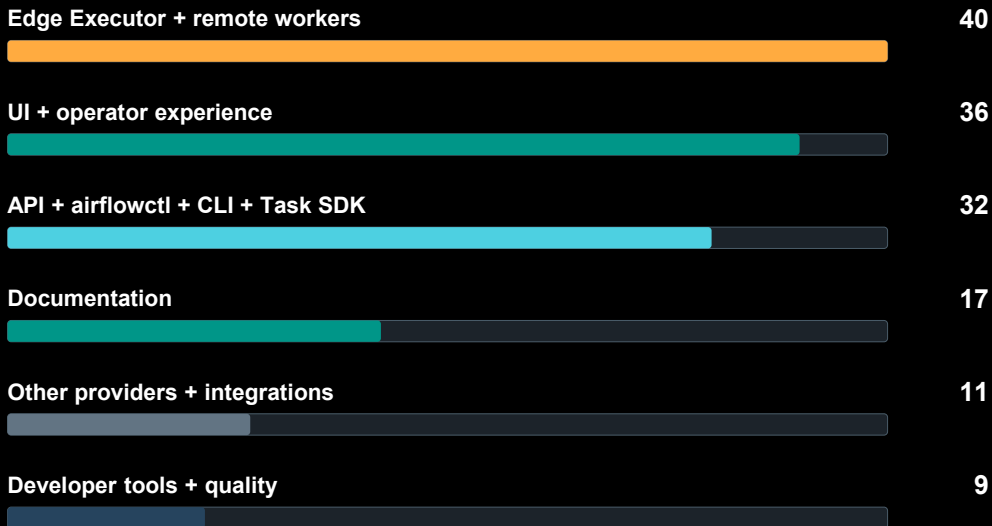
across Qualcomm contributors

**Production requirements
=
upstream capabilities**

Contributing the features, fixes, and operational improvements we rely on.

Where we contributed

GitHub area labels across the scoped 135 PRs





Thank You!!

Scan the QR Code to connect with us on LinkedIn



Shubham Raj

Apache Airflow Committer | Senior Engineer



Sriraj Dheeraj Turaga

CAD Engineer | Committer @ Apache Airflow



Devakumar Madhavan

Principal Engineer at Qualcomm

