TECHNICAL BRIEF · V1.0 · FOR VALIDATION & QUALIFICATION ENGINEERS

Hydra Command Center

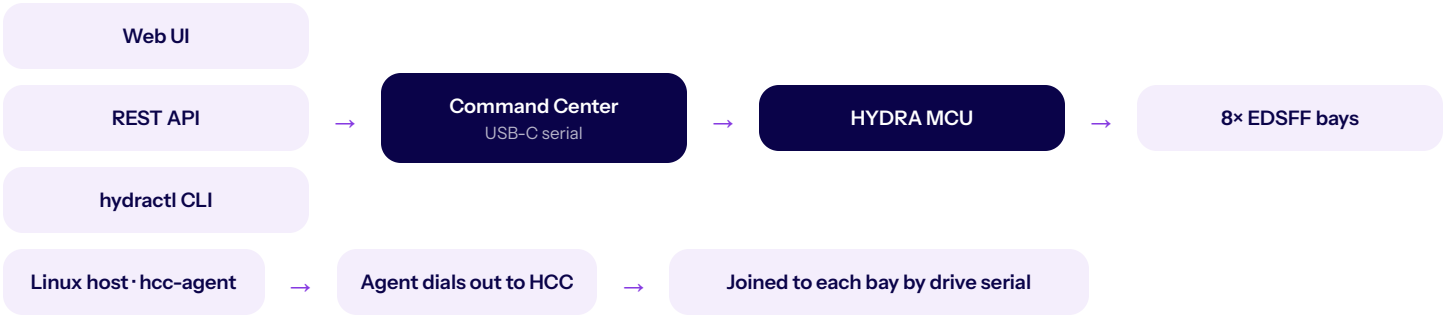
Out-of-band control, protocol access, in-band host correlation, and pass/fail qualification for the Gen6 Hydra enclosure — across eight bays, no host driver required.

A single MCU reaches every bay over sideband — SMBus/I2C, PCIe VDM, and now I3C — and exposes it over one USB-C serial link as a local REST API. Commands and events travel as two MCTP message types (**0x04** NVMe-MI, **0x08** vendor-defined), so the same wire protocol that already talks to the drives carries Command Center's own control traffic.

ARCHITECTURE

One MCU, three client surfaces, eight bays — plus an optional in-band agent

The Web UI, the REST API, and the **hydractl** CLI are three doors onto the same server process — nothing is UI-only. An optional Linux host agent dials out and joins what the host sees to each bay by drive serial, without HCC ever needing inbound access to the host.



Out-of-band, always

Every bay is readable and controllable over the MCU's sideband link whether or not a host is attached, booted, or even wired — power, presence, temperature, and identity never depend on a driver.

Fail-closed qualification

Validation gates read live hardware against a governing spec and report PASS/FAIL/INCOMPLETE — never a silent skip. A gate that can't be read is INCOMPLETE, not green.

Scriptable end to end

The same REST API behind the Web UI drives every example script and CI gate — one contract, one base URL, from a laptop bench to an unattended rack.

8
EDSFF bays, independently addressed

2
MCTP message types carry all control traffic

7
Live validation gates, fail-closed

1
USB-C serial link to the host

MONITOR & CONTROL

Per-bay device readout & bay control

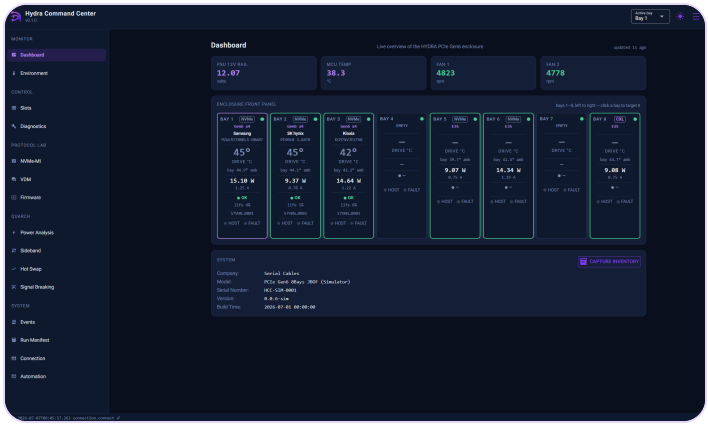
A round-robin poller refreshes every bay's identity, telemetry, and health on a fixed cadence, so the dashboard and the API read from the same current snapshot — no on-demand probe needed to answer "what's in bay 5 right now."

FIELD	SOURCE
Vendor / model / firmware / capacity	NVMe Identify over NVMe-MI, cached per bay
Drive & bay temperature	NVMe-MI health log plus MCU-side bay sensor, cross-checked
Health / life / spare	SMART / Health Information log page over NVMe-MI
SMBus cross-check	Independent SMBus read reconciled against the NVMe-MI value
CXL device health / identity	CXL CCI mailbox (Type-3 devices, Technology Preview)
SAS drive (U.3 interposer)	Interposer IO expander, probed on every slot refresh — see footnote
I3C discovery	Per-bay dynamic address assignment + identity decode over I3C (Technology Preview — see p.3); separate from the raw I3C register access described on p.5

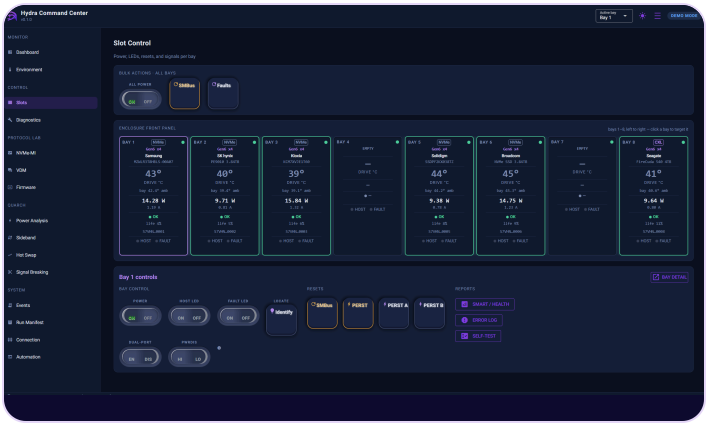
A drive mounted through a U.3 interposer does not assert bay-level presence, but HCC probes the interposer's IO expander on every slot refresh and labels the tile **SAS-interposed** automatically. The dedicated SAS page (sideband scan, confirm-gated PWRDIS) is experimental in this release. I3C discovery — dynamic address assignment and identity decode, distinct from the raw register access on p.5 — is a Technology Preview in v1.0; reading drive data over I3C, including NVMe-MI, does not yet work on current hardware.

BAY CONTROL

Power, LED, and PERST# are scoped per bay and confirm-gated in the UI — the same actions the API exposes for scripted bring-up/bring-down sequences. A slot that's mid-stress-test shows that state before you can touch its power.



Dashboard. All eight bays at a glance — identity, temperature, health, and link state refreshed on the poller's cadence.



Slot control. Per-bay power, LED, and PERST# actions, confirm-gated before anything changes.

PROTOCOL LAB

NVMe-MI, VDM, CXL CCI & I3C consoles

Four protocol consoles give direct, logged access to the sideband transports underneath the dashboard — for the cases a health tile can't show: a raw Identify dump, a vendor-defined message, a CXL mailbox command, or an I3C discovery cycle.

NVMe-MI console

Identify, Health, and vendor log pages on demand, with the raw request/response logged for every call.

VDM console

PCIe vendor-defined messages to the MCU and drives, for protocol work that falls outside the NVMe-MI spec.

CXL CCI console

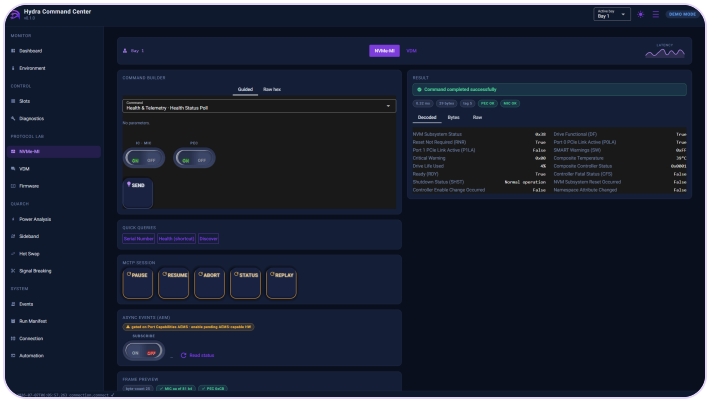
TECH PREVIEW

CXL Type-3 mailbox: identify, health, firmware info, and event log over the component command interface.

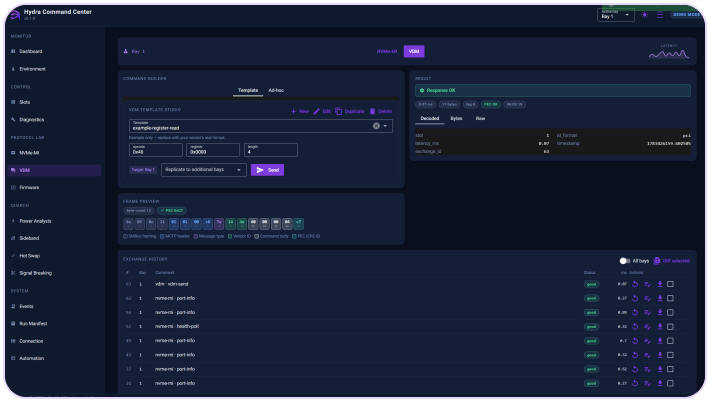
I3C console

TECH PREVIEW

Discover (dynamic address + PID/BCR/DCR decode), Run Test (repeated discovery cycles, pass/fail), and Recover (release a held bus) — per bay.



NVMe-MI console. Identify and health-log pulls with the full MI request/response pair shown.



VDM console. Vendor-defined message send/receive, logged the same way.

SAMPLE — CXL CCI FROM THE API

```
# identify a CXL Type-3 device in bay 8, then read its health
POST /api/v1/bays/8/cxl/identify
POST /api/v1/bays/8/cxl/health

{"success": true, "data": {"vendor_id": "0x8086", "fw_slot": 1,
"life_used_pct": 4, "media_status": "normal", "events_pending": 0}}
```

CXL out-of-band management and I3C discovery are both Technology Previews in this release (see the introduction) — scoped to Type-3 mailbox reads and per-bay I3C discovery/recovery respectively, not yet a general in-band data path.

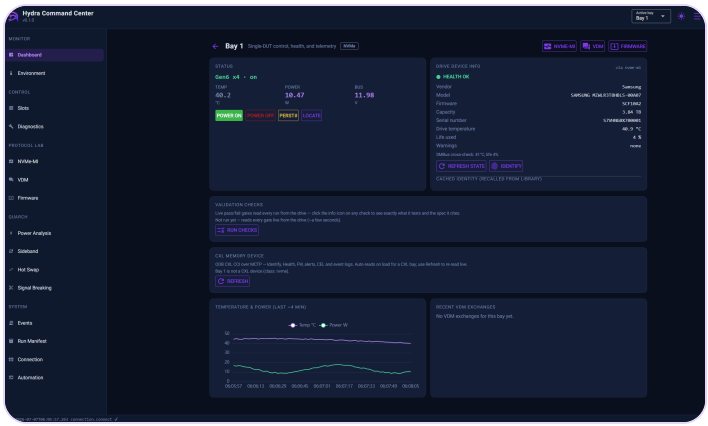
QUALIFY & PROVE

Fail-closed validation gates

Each gate reads live hardware against a governing spec and reports PASS, FAIL, or INCOMPLETE — never a silent skip. A gate set is chosen per qualification run, not hard-coded, so a bench script and a CI job can ask for exactly the gates that run matter.

GATE	READS LIVE	GOVERNING SPEC
Link	Negotiated PCIe generation & width vs. expected	PCIe Base Spec
Thermal	Drive & bay temperature vs. threshold	NVMe-MI health log
Reliability	SMART critical warning, media errors	NVMe SMART / Health log
Endurance	Percentage used, spare capacity	NVMe SMART / Health log
Firmware	Active slot matches the expected revision	NVMe Firmware Slot log
Traceability	Serial match against the device library	HCC device library
Self-test	Most recent Device Self-Test result	NVMe Device Self-Test log
CXL health	Type-3 mailbox health vs. threshold	CXL 3.x, Technology Preview
CXL events	Pending CXL event-log entries	CXL 3.x, Technology Preview
Host identity & SMART link	Host-side serial/SMART joined to the bay via hcc-agent	In-band host agent (p.6)

A gate set is selected per run — a bench smoke test and a full qualification pass don't need the same ten gates, and a gate that can't be read (no CXL device present, no host agent enrolled) reports INCOMPLETE rather than being silently dropped.



Serial-keyed device library

Identity only, never health — conflicts flag automatically when two bays report the same serial.

Device Self-Test, gated

Short and extended self-test launch and poll, confirm-gated before anything disruptive runs.

Error log for RMA triage

The NVMe error-information log is pulled and preserved per gate run, so a FAIL carries the evidence an RMA ticket needs without a second pass on the bench.

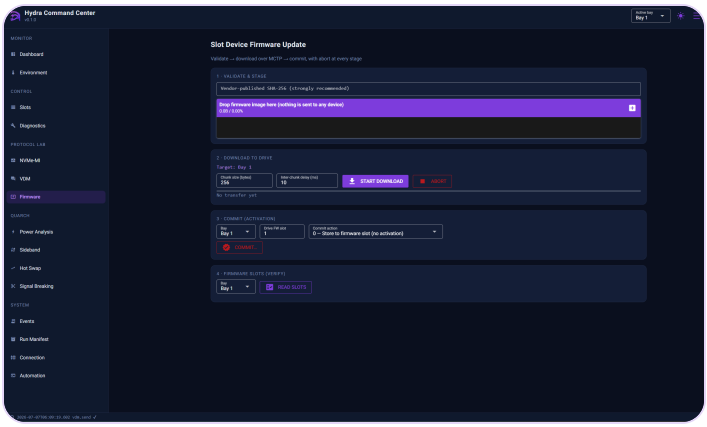
FIRMWARE, ENVIRONMENT & REPRODUCIBILITY

Safety-gated firmware, full telemetry, deterministic re-tests

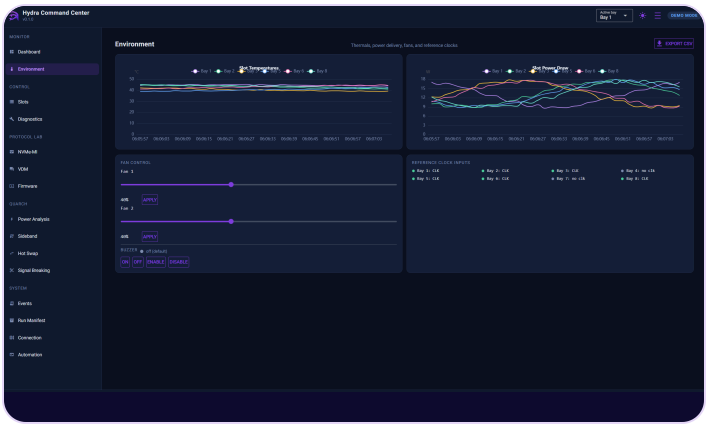
Firmware moves through the same validate → stage → commit state machine whether it's driven from the UI or a script, and every run's environment is captured so a FAIL can be reproduced, not just reported.

GATE	WHAT IT CHECKS BEFORE ADVANCING
Validate & stage	Image matches the target drive's vendor/model, slot has spare capacity for the download
Download	Firmware Image Download completes and checksums before activation is offered
Commit	Activate & reset only after an explicit confirm; slot state re-read to verify the new revision took

Live-hardware caution: a firmware commit resets the drive. Run this against an expendable drive until the image is proven on your fleet.



Firmware. Validate → stage → commit, with the active and pending slot shown throughout.



Environment. Bench ambient, enclosure fan state, and PSU rails captured alongside every run.

DIAGNOSTICS & DIRECT BUS ACCESS

Below the per-bay consoles

A built-in BIST exercises the MCU's own logic independent of any bay. Below that, raw I2C/SMBus access reaches any device on the sideband bus directly. I3C access (MCU FW v0.08+) adds a full CCC engine for MIPI I3C Basic v1.1.1 — register-level bus commands, distinct from the per-bay I3C discovery console on p.3.

REPRODUCIBILITY

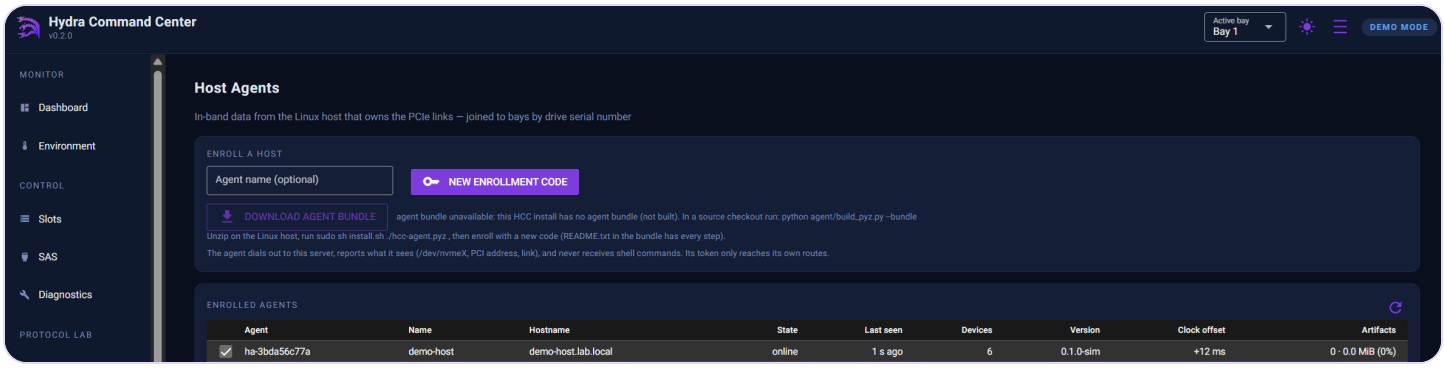
Same inputs, same verdict

Every run's firmware revision, environment snapshot, and gate set are recorded with its result, so a FAIL from last week can be re-run against the same conditions rather than argued about from memory.

NEW IN V1.0 · HOST AGENT (IN-BAND)

What the host sees, joined to the bay by serial number

An optional agent on the Linux host dials out to HCC and answers typed, allow-listed jobs — link state, AER counters, SMART, fio profiles — joined to each out-of-band bay by the drive's serial number, with no inbound access to the host required.



Host Agents. Enrolled agents, their last-seen status, and the bays each one is joined to by serial.

SECURITY MODEL

Dials out	Agent reaches HCC; HCC never opens a connection to the host
Typed jobs only	No shell, no arbitrary command execution
Scoped token	Per-agent enrollment token, revocable
Writes opt-in	allow_writes off by default in the agent config
Fails closed	A mounted/held namespace is never reset without override
TLS	Agent ↔ HCC traffic is TLS by default

INSTALL & ENROLL

Three steps

1. Install the agent package on the host. 2. Point it at the HCC base URL and enrollment token. 3. Confirm it in Host Agents — it joins to bays automatically by drive serial.

```
pip install hcc-agent
hcc-agent enroll --url https://hcc.lab:8600 \
  --token $HCC_ENROLL_TOKEN
systemctl enable --now hcc-agent
```

JOB CATALOGUE (TYPED, ALLOW-LISTED)

JOB FAMILY	WHAT IT RETURNS	WRITE?
nvme.identify family	Host-side NVMe Identify, namespace list, controller caps	Read
pci.config_dump, pci.aer_snapshot	PCIe config space and AER counters as the host's own driver sees them	Read
fio.profile	A canned fio job, streamed back at 1 Hz	Read
host.dmesg_capture, nvme.telemetry_log, cxl.list	Kernel-log window, NVMe host telemetry log, CXL device list	Read
Write recipes	Controller/link/bus reset, write-profile fio — see host_reset_recovery, p.9	OPT-IN

Watches (AER, dmesg) run on the agent and report deltas over the same channel — no polling loop needed on the HCC side.

NEW · WHAT TO EXPECT ON WINDOWS

Windows ready is now a bench-validated claim, not just an installer claim

At large block sizes Windows matched Linux on the same Gen6 x4 drive — **19.3 vs 19.5 GB/s at 128k**. At small block sizes and high queue depth Windows reached roughly half of Linux's IOPS (**~300k vs 600–665k at QD32/128**); Windows held level or ahead at low queue depth. *(Source:* `nvme2x-perf-profile` *.)*

NEW IN V1.0 · WORKLOADS

Saved, repeatable tests with a verdict

A workload is a saved, repeatable sequence of phases — preset or custom — that arms, runs, and exits with a verdict: passed, review, failed, or incomplete. The same engine backs the UI, the API, and the workload_run example script.

Errors are counted, never auto-failed

A review verdict means "look at the counters" — the run isn't silently marked PASS, but it isn't auto-failed either.

STOP TEST is never gated

A run can be stopped at any phase, from the UI or Ctrl-C on the script, and the server honors it immediately.

Destructive steps need an ARM

Anything that writes to the drive or power-cycles a bay requires an explicit arm before it will start.

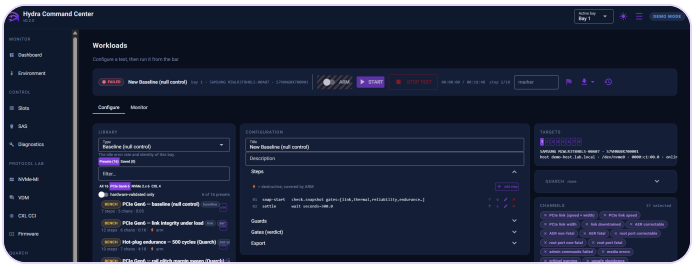
SIXTEEN SHIPPED PRESETS

PRESET	PURPOSE	BENCH
pcie-gen6-baseline	Link, power, and thermal baseline before anything else runs	PROVEN
pcie-gen6-link-integrity	Link retrain / error-counter sweep under load	PROVEN
pcie-gen6-rail-margin-sweep	Quarch-driven rail margining across the PCIe power envelope	PROVEN
pcie-gen6-downtrain-hunt	Repeated hot-plug cycles hunting for a downtrained link	BENCH
nvme2x-admin-conformance	NVMe 2.x admin-command conformance sweep	PROVEN
nvme2x-perf-profile	fio block-size / queue-depth performance matrix	PROVEN
nvme2x-reset-recovery	Controller / link reset with recovery timing	PROVEN
nvme2x-selftest-under-load	Device Self-Test launched concurrently with I/O	PROVEN
nvme2x-write-endurance	Sustained write pass with SMART before/after delta	PROVEN
cxl-t3-health-watch	CXL Type-3 mailbox health polled across the run	PROVEN
cxl-t3-event-sweep	CXL event-log drain and classification	BENCH
cxl-t3-inband-inventory	CXL device inventory cross-checked host-side via hcc-agent	BENCH
cxl-t3-power-thermal	CXL device power/thermal watch under sustained I/O	BENCH
hotplug-endurance	Bay power-cycle endurance loop	BENCH
thermal-soak	Fan-commanded thermal stress with an abort ceiling	BENCH
quarch-power-margin	Quarch rail margining / glitch injection	BENCH

Verdicts are passed, review, failed, or incomplete and export as a diffable JSON manifest, an aligned CSV, NDJSON, and a self-contained HTML report. Every channel is folded into 1s and 10s aggregate buckets on disk for 24h. **BENCH:** 9 of 16 shipped presets are now bench-proven end to end on hardware — the four pcie-gen6 presets but downtrain-hunt, all five nvme2x presets, and cxl-t3-health-watch. The rest run normally; BENCH is information, not a lock.

From live telemetry to a report you can file

Two new capabilities turn a workload run from a live view into evidence: a statistics panel that summarizes every channel per phase, and a one-file qualification report built straight from a finished run.

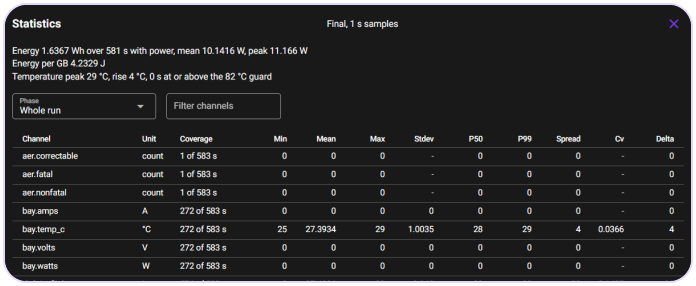


Every preset above runs through this same engine, shown live in the Workloads view at left — and now feeds two new ways to turn a run into evidence.

NEW · STATISTICS PANEL

Every channel, summarized per phase

Coverage, min, mean, max, standard deviation, P50, P99, IOPS-per-watt, and joules-per-GB — computed per phase for every monitored channel, not just the headline throughput number. Available live via `hydractl workload stats <run>` or `GET /workloads/runs/{id}/statistics`.



Channel	Unit	Coverage	Min	Mean	Max	Stdev	P50	P99	Spread	Cv	Delta
aer.correctable	count	1 of 583 s	0	0	0	-	0	0	0	-	0
aer.fatal	count	1 of 583 s	0	0	0	-	0	0	0	-	0
aer.nonfatal	count	1 of 583 s	0	0	0	-	0	0	0	-	0
bay.amps	A	272 of 583 s	0	0	0	0	0	0	0	-	0
bay.temp_c	°C	272 of 583 s	25	27.3934	29	1.0035	28	29	4	0.0366	4
bay.volts	V	272 of 583 s	0	0	0	0	0	0	0	-	0
bay.watts	W	272 of 583 s	0	0	0	0	0	0	0	-	0

NEW · QUALIFICATION REPORT (HTML)

A fileable, signable report — not just a dashboard

Generated from a finished run with `hydractl workload qualify <run> --document QR-0042 --operator "A. Tester" -o report.html`, the report is one self-contained, printable HTML file with five sections: **1.** summary and findings, **2.** device and setup, **3.** results by phase — power vs. performance (IOPS, MB/s, mean/peak watts, IOPS/W, J/GB, P99 latency), **4.** health before/after, and **5.** evidence and sign-off, with document number, customer, tester, and reviewer fields. It must be generated within roughly 25 hours of the run, while per-sample telemetry is still retained.

Why it's the stronger proof point

A statistics panel is a view; the qualification report is a deliverable — a qualification engineer can file it, send it to a customer, or print it and sign it, with the Power-vs-Performance evidence already laid out by phase.

Reading sustained throughput

A 120s sequential 1MB step on a Gen6 x4 drive cycled on both Linux and Windows: ~22.9 GB/s at 24W, then 5–6 GB/s at 13–14W, averaging ~12 GB/s overall — one sensor passed 70°C while the bay read 31°C.

FAULT INJECTION · OPTIONAL QUARCH HARDWARE

Hot-Plug Lab & Power Analysis

With an optional Quarch module attached, HCC drives hot-plug and power-fault sequences through the same confirm-gated, scriptable contract as everything else — and feeds the results into the same workload engine.

Hot-Plug Lab — one signal matrix

Surprise removal, staged insertion, and repeated cycling across every signal the breaker paddle controls, from one matrix.

Hardware glitch engine

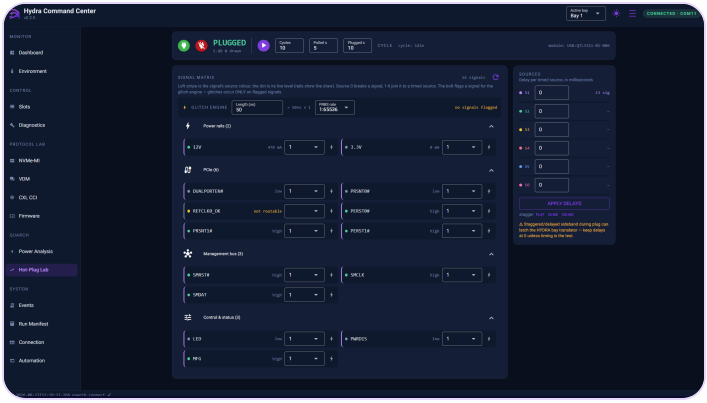
The Quarch module injects real electrical faults — this isn't a simulated power event.

Power Analysis

High-rate rail capture synchronized to the fault sequence, so a glitch and its electrical signature land in the same trace.

Confirm-gated, scriptable, workload-ready

Every disruptive step needs an explicit confirm, and the same sequences run from quarch_power_margin or a saved workload preset.



Hot-Plug Lab. Surprise removal, staged insertion, and cycling sequences on the breaker paddle.

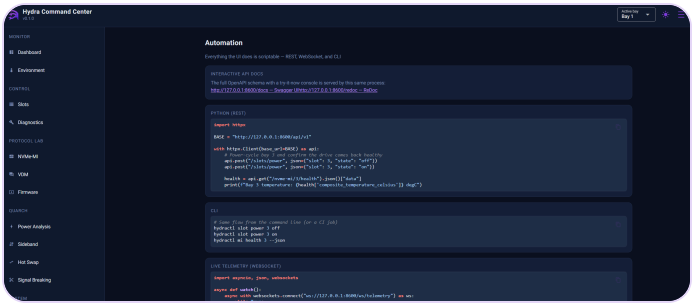


Power Analysis. High-rate rail capture synchronized to the fault-injection sequence.

AUTOMATE & INTEGRATE

One envelope, 23 example scripts, one maturity table

Every response from the API answers the same envelope, and every subsystem's maturity is stated plainly — so a validation team knows exactly what to trust today and what's still a preview.



Automation. Live links to Swagger UI, ReDoc, and the raw OpenAPI schema served by your own instance.

Every example script shares the same exit-code contract: **0** passed, **1** a real failure, **2** a connection or setup error — nothing was validated.

422
Request failed validation — the envelope's error field explains what

502
MCU didn't answer in time — a hardware-side timeout, not a client error

409
Conflicting operation already in progress on that bay

no-store
Live telemetry endpoints are never cached by an intermediary

SUBSYSTEM MATURITY IN THIS RELEASE

SUBSYSTEM	STATUS
Enclosure control	FULLY SUPPORTED
NVMe-MI health / temp / spare / SMART	FULLY SUPPORTED
Telemetry persistence / event log / run manifest	FULLY SUPPORTED
Host Agent (in-band) & agreement gates	FULLY SUPPORTED
Workloads engine — presets carry a BENCH marker	SUPPORTED · PRESETS: BENCH
Quarch fault injection	SUPPORTED · LINUX: UDEV
CXL CCI	TECHNOLOGY PREVIEW
I3C discovery	TECHNOLOGY PREVIEW
Slot-device (drive) firmware update	USE EXPENDABLE DRIVE
SAS page (U.3 interposer)	EXPERIMENTAL
VDM Template Studio	DEMONSTRATION

Quarch-fixture scripts need the vendor USB driver (Windows, proven) or a one-time udev rule (Linux, documented); CXL CCI and I3C discovery need MCU firmware that supports their respective command sets.

SERIAL CABLES · GEN6 HYDRA ENCLOSURE

Put Command Center on your Gen6 bench

One control plane for enclosure, protocol, and in-band correlation — Serial Cables, LLC · Englewood, CO · HYDRA 8-Bay Gen6 JBOF | serialcables.com