Flux: A Safety-First AC/DC PDU Control Plane for Research Racks - An Evidence-Aware Architecture, Evaluation Plan, and Integration Roadmap

Aaron Singh

Abstract **- Flux models rack AC input, DC distribution, branch limits, outlet sequencing, redundant feeds, alarms, and remote PDU telemetry. A queen controller coordinates policy-limited child nodes through mocked protocols and an auditable command log. The initial project never switches live mains equipment and makes no compliance or certification claim. This paper defines the system model, evidence contract, risks, and staged validation plan. No physical-system or performance result is claimed without a committed reproducibility artifact.**

Index Terms - power distribution unit, AC power, DC distribution, rack telemetry, control plane

I. INTRODUCTION

Semiconductor research infrastructure combines servers, cameras, stages, instruments, and supporting equipment with distinct startup, capacity, and fault behavior. Flux evaluates how telemetry and policy can coordinate those loads while keeping all first-phase commands inside a simulation.

Research contribution

The contribution is not a performance claim. It is a reproducible reference architecture, a falsifiable evaluation protocol, and an evidence schema that records inputs, decisions, outputs, failure modes, and claim boundaries.

II. TECHNICAL CONTEXT

A rack power controller must distinguish desired state from permissible state. Flux evaluates authorization, branch capacity, startup dependencies, redundant-feed state, and telemetry freshness before a modeled command can reach an outlet. Every acceptance and rejection is auditable.

III. RESEARCH QUESTIONS

1 Can policy checks prevent commands that exceed configured capacity?

2 How should loads be sequenced after loss of a feed?

3 Which telemetry best predicts approaching branch or thermal limits?

IV. SYSTEM MODEL AND ARCHITECTURE

A queen controller receives desired state and current telemetry. Policy checks validate authorization, branch capacity, sequencing dependencies, and interlock state before commands reach child PDU models. Each child records outlet state, voltage, current, power, alarms, and command provenance.

For each branch b: sum of active load current must remain below the configured derated limit, while dependency and interlock predicates remain true.

The model is a design and test abstraction. It establishes invariants and expected-value checks; it does not convert simulated variables into measured hardware behavior.

V. DETAILED OPERATION AND RATIONALE

1 Define single-line topology, feeds, branches, outlets, loads, and legal transitions.

2 Replay startup, shutdown, transfer, overcurrent, undervoltage, and communications-loss scenarios.

3 Mock MQTT and Modbus interfaces without live equipment.

4 Reconcile device telemetry against computed capacity and energy totals.

VI. DATA AND EVIDENCE CONTRACT

Experiments use synthetic voltage, current, power, temperature, outlet, breaker, and feed-loss telemetry together with generated server and instrument load profiles. Any later device capture must remove secrets and identify calibration and sampling limitations.

Each run must preserve a machine-readable configuration, random seed where applicable, source revision, environment description, raw result, summary method, and known limitations. Derived plots must point back to the exact records used to produce them.

VII. IMPLEMENTATION BLUEPRINT

Topology registry. Defines feeds, branches, outlets, loads, and dependencies. Its acceptance artifact is versioned single-line model.

Policy engine. Checks authorization, capacity, sequence, and interlocks. Its acceptance artifact is decision record with reason codes.

Child PDU model. Simulates outlet state and electrical telemetry. Its acceptance artifact is state and measurement stream.

Queen coordinator. Reconciles desired state across redundant feeds. Its acceptance artifact is ordered command and recovery log.

VIII. EXPERIMENTAL METHODOLOGY

Evaluation proceeds in four gates. Gate 1 checks deterministic behavior and invariants. Gate 2 applies controlled noise, tolerance, and injected-failure sweeps. Gate 3 repeats measurements and reports distributions rather than a favorable single run. Gate 4 introduces an external data source or bounded lab adapter only after the first three gates are reviewable.

Primary metrics are reported with sample counts, aggregation rules, uncertainty, and failure cases.

IX. REPRODUCIBILITY AND ACCEPTANCE CRITERIA

A result is accepted only when another clean environment can reproduce the reference behavior from a named revision. Every comparison uses identical budgets and held-out conditions. Failed runs remain part of the evidence record. Unknown hardware, calibration, and production conditions are labeled unknown rather than assigned synthetic values.

X. FAULT INJECTION AND NEGATIVE TESTING

Negative testing is part of the main evaluation rather than an appendix. Each risk is converted into an explicit scenario with an expected detection, rejection, abstention, or safe-state outcome. A silent success under missing or contradictory state is treated as a failed test.

Stale telemetry. The failure consequence is unsafe command uses old capacity. The required control is freshness gates and fail-closed policy.

Inrush overlap. The failure consequence is startup exceeds branch capacity. The required control is sequencing and load profiles.

Feed ambiguity. The failure consequence is redundancy is overstated. The required control is explicit source and transfer state.

Live-control misuse. The failure consequence is simulation treated as certified control. The required control is mocks only; read-only adapter first.

XI. ANALYSIS AND INTERPRETATION

Analysis reports distributions, failure counts, and representative traces rather than a single favorable score. Baselines share the same inputs, budgets, stopping rules, and held-out conditions. When multiple configurations are explored, the selection procedure is recorded so the test set is not used as a tuning signal.

The primary metric families are Capacity headroom, Energy-accounting error, Alarm detection time, Rejected unsafe commands, Recovery sequence correctness. Each is reported with its unit, aggregation rule, sample count, spread, and failed or undefined scenarios. An improvement is meaningful only when reference correctness is preserved and the result repeats across declared evaluation seeds or groups.

XII. INTEGRATION INTERFACES

Integration occurs through versioned records rather than hidden shared state. Inputs declare units, coordinate frames or topology identifiers, timestamps, validity windows, and provenance. Outputs include a status, result payload, limitation flags, and a stable link to the run configuration.

Adapters are introduced in read-only or simulated form first. A live adapter cannot inherit the trust assigned to the reference model; it requires separate authentication, calibration, timeout, retry, and safe-state evidence.

XIII. EXTERNAL VALIDITY AND RELEASE CRITERIA

The initial conclusions are bounded by the declared simulator, public data, protocol mocks, and sampled conditions. They do not automatically generalize to other equipment, optical assemblies, wafers, stages, power systems, operating ranges, or production environments. External validity is evaluated as a new experiment, not assumed from architectural similarity.

A portfolio release requires the paper source, generated PDF, configuration schema, reproducible entry point, tests, raw or referenced evidence, and a concise limitations record. If one of these artifacts is missing, the project may remain a research design but cannot be presented as an evidence-backed implementation.

XIV. CLAIM BOUNDARIES, SAFETY, AND RISKS

Simulation and protocol mocks only. Flux does not switch live AC mains, replace a licensed electrical design, or claim electrical-code compliance, product certification, arc-flash safety, or production reliability.

The primary research risk is a category error: treating simulation, public benchmark data, or protocol mocks as proof of physical-system performance. The mitigation is provenance at every boundary, fail-closed behavior for missing state, and an explicit review before any live interface is enabled.

XV. GOALS, MILESTONES, AND INTEGRATION

1 Implement power topology and accounting models.

2 Add outlet sequence and authorization policies.

3 Inject feed, breaker, sensor, and communications faults.

4 Integrate FabSystems server and instrument load profiles.

5 Validate a read-only lab PDU adapter before considering control.

XVI. CONCLUSION

Flux now has a research plan comparable in rigor and presentation to the established portfolio papers. Its architecture is inspectable, its evaluation is falsifiable, and its limitations are explicit. Credibility will come from completing the staged evidence artifacts, not from promoting unverified numbers.

REFERENCES

[1] Open Compute Project, 'Rack and Power Project.'

[2] Aaron Singh, 'Flux research specification,' local portfolio repository, 2026.

TABLE I
SYSTEM COMPONENTS AND EVIDENCE RESPONSIBILITIES

Component	Responsibility	Required evidence
Topology registry	Defines feeds, branches, outlets, loads, and dependencies	Versioned single-line model
Policy engine	Checks authorization, capacity, sequence, and interlocks	Decision record with reason codes
Child PDU model	Simulates outlet state and electrical telemetry	State and measurement stream
Queen coordinator	Reconciles desired state across redundant feeds	Ordered command and recovery log

TABLE II
PLANNED EVALUATION MATRIX

Metric family	Acceptance record
Capacity headroom	Repeated result with configuration, seed, revision, and limitations
Energy-accounting error	Repeated result with configuration, seed, revision, and limitations
Alarm detection time	Repeated result with configuration, seed, revision, and limitations
Rejected unsafe commands	Repeated result with configuration, seed, revision, and limitations
Recovery sequence correctness	Repeated result with configuration, seed, revision, and limitations

TABLE III
RISK REGISTER AND CONTROLS

Risk	Failure consequence	Control
Stale telemetry	Unsafe command uses old capacity	Freshness gates and fail-closed policy
Inrush overlap	Startup exceeds branch capacity	Sequencing and load profiles
Feed ambiguity	Redundancy is overstated	Explicit source and transfer state
Live-control misuse	Simulation treated as certified control	Mocks only; read-only adapter first

TABLE IV
PROJECT GOALS AND REQUIRED PROOF

ID	Goal	Completion evidence
G1	Implement power topology and accounting models.	Reviewable source, test, trace, or measurement artifact
G2	Add outlet sequence and authorization policies.	Reviewable source, test, trace, or measurement artifact
G3	Inject feed, breaker, sensor, and communications faults.	Reviewable source, test, trace, or measurement artifact
G4	Integrate FabSystems server and instrument load profiles.	Reviewable source, test, trace, or measurement artifact
G5	Validate a read-only lab PDU adapter before considering control.	Reviewable source, test, trace, or measurement artifact