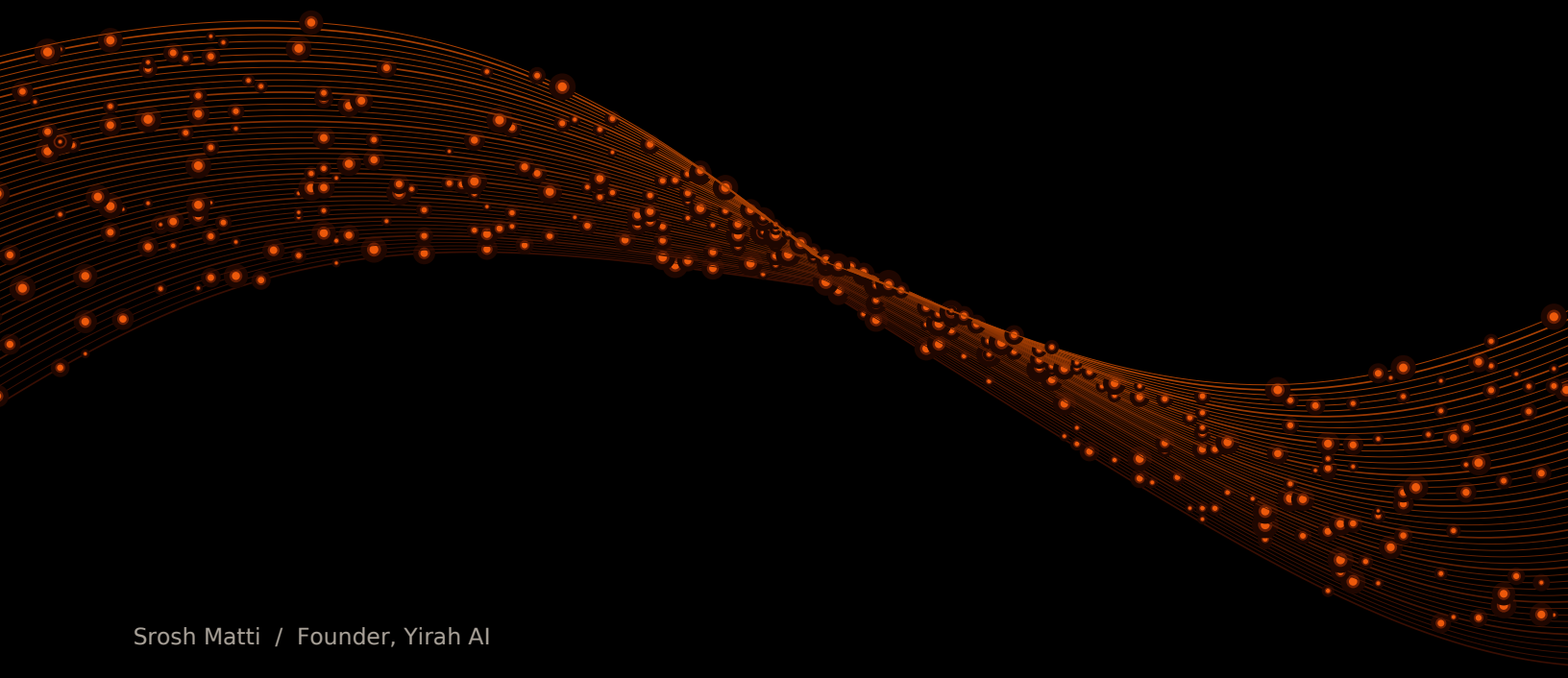
YIRAH AI

AREG / TECHNICAL WHITEPAPER

Adaptive resource intelligence.

A value-aware decision layer
for AI infrastructure.



Srosh Matti / Founder, Yirah AI

Version 0.8 · Revision 03 · 8 September 2026

ARCHITECTURE / MATHEMATICAL MODEL / EVIDENCE

00 / READ THIS FIRST

Purpose, scope and current status

Abstract. Yirah AI is developing AREG, a software decision layer that evaluates how queued AI work should use constrained accelerator capacity. Its objective is to improve customer-defined useful work while preserving performance, placement and facility guardrails. The current implementation supplies observation and evidence foundations plus an offline, non-actuating public-trace evaluator. It does not operate a customer scheduler.

This revision documents build **0.8.3.dev0**, including GPU-model compatibility, a declared capacity-stress profile and a frozen holdout. Development improved modeled weighted value by **6.55%**; the compatibility holdout changed it by **-0.0049%** and worsened P95 wait by **13.64%**. The release gate remains failed. These are replay results, not measured customer savings. [1-4]

Document control	This revision
Technology / category	AREG / Adaptive Resource Intelligence
Software baseline	0.8.3.dev0 · source revision cf10591a880f
Document baseline	Revises the 7 September 2026 v0.8 paper
Audience	AI infrastructure engineers, technical evaluators and design partners
Review method	Source and stored evidence reviewed; full traces were not rerun for this paper

Reading map

System boundary and architecture 03
Observation, telemetry and evidence 04
Mathematical model and implemented policy 05
Decision procedure and engineering status 06
Evaluation design and frozen gates 07
Development and holdout results 08
Interpretation and validity limits 09
Deployment and operational boundary 10
Economic measurement 11
Pilot protocol and engineering exit criteria 12
Reproducibility and provenance 13
References 14

01 / SYSTEM ARCHITECTURE

A decision layer beside the scheduler

The engineering question is whether explicit workload value can improve decisions when compatible GPU capacity is scarce. A busy GPU does not by itself establish a valuable outcome. Conversely, idle capacity is not automatically recoverable: it may be incompatible with queued work, fragmented, reserved, or constrained by service requirements.

AREG brings workload intent, resource state and policy into an inspectable decision record. Customer policy supplies value or priority; the software must not infer commercial value from utilization alone. The intended loop is observation, modeling, recommendation and comparison with measured outcomes. [1,5]

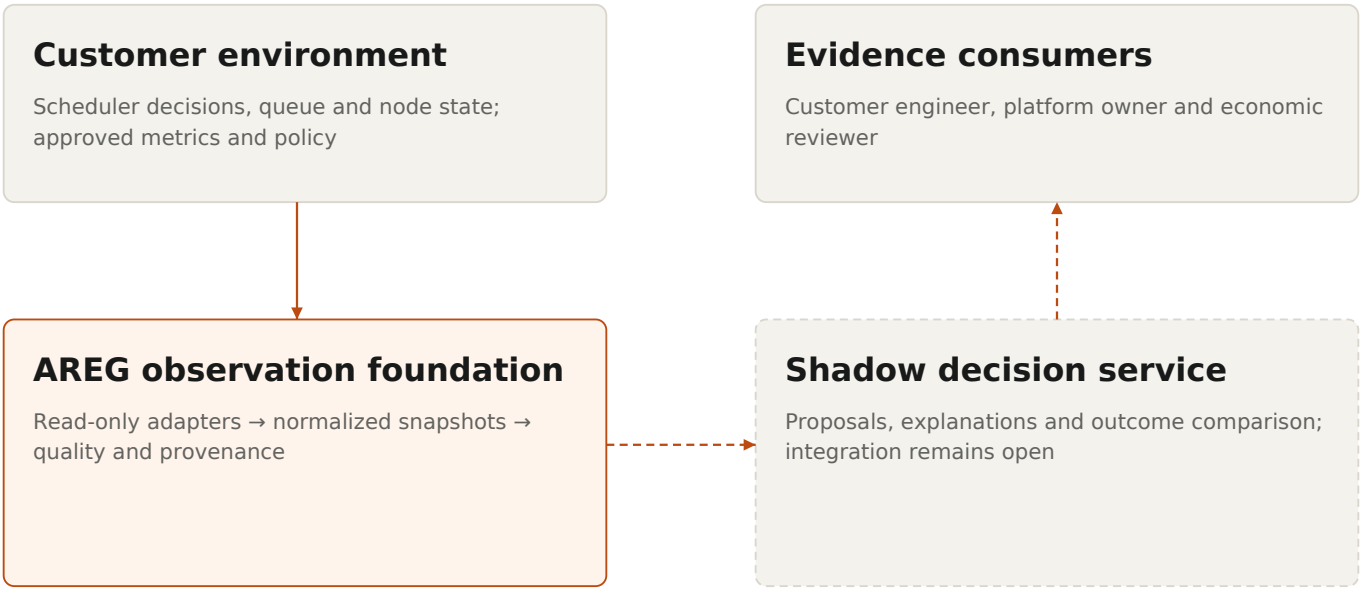


Figure 1. Target system context. Solid blocks describe the observation foundation; the dashed block is an integration target. Information flows out for review. AREG has no production command path.

Implemented boundary versus target architecture

The public evaluator reads trace files and writes reports. It compares three replay policies inside a modeled cluster; it is not connected to Kubernetes. The live observation code reads approved infrastructure signals and records evidence. These paths should not be presented as an already integrated production optimizer.

Facility power, thermal limits, memory, topology and fairness belong in the full decision boundary. The current compatibility replay enforces GPU-model pool capacity only. A richer resource model does not establish that every represented constraint is enforced in every execution path.

Architecture figures are SysML-inspired communication views, not normative SysML models or a claim of certification. The exact implementation and declared experimental scope take precedence over diagram shorthand.

02 / DATA AND EVIDENCE

Observe without inventing missing data

The observation foundation normalizes source-specific records behind versioned contracts. Snapshots carry source identity, observation and ingestion times, quality flags, job state and resource information. Salted pseudonyms replace raw infrastructure identifiers in exported evidence. The Kubernetes adapter is scoped to pod and node reads; workload payloads are outside its data boundary. [5]

Signal source	Normalized metric	Unit
DCGM_FI_DEV_GPU_UTIL	gpu.utilization	Ratio
DCGM_FI_DEV_POWER_USAGE	gpu.power	kW
DCGM_FI_DEV_FB_USED	gpu.framebuffer_used	GiB
DCGM_FI_DEV_GPU_TEMP	gpu.temperature	Celsius

NVIDIA DCGM Exporter exposes GPU metrics to Prometheus. AREG uses a fixed query set and read-only instant queries, recording collection time and query identity with normalized samples. This observes device signals; it does not turn GPU temperature into facility thermal headroom or telemetry into a billing ledger. [5,7,8]

Evidence should answer four questions

Question	Minimum record
What was observed?	Snapshot identity, timestamps, accepted samples and source quality
What policy was evaluated?	Policy version, parameter fingerprint and evaluation profile
Why that proposal?	Candidate ranking, exclusions and applicable constraints; target decision contract
Can the record be checked?	Canonical digest, chain continuity and applied=false

Failure and integrity semantics

Missing metrics, stale timestamps, non-finite values, invalid responses and untrusted clocks become explicit quality flags. A missing required constraint input must prevent a claim that the affected recommendation is safe. The in-process telemetry ledger validates hash-chain continuity; durable persistence and complete operator reporting remain engineering work. [5]

A hash chain detects changed records relative to a trusted anchor. It does not independently authenticate the original sensor, prove the truth of a measurement, or prevent an attacker with full storage control from replacing an entire unanchored chain. Customer identity mapping, key handling and retention therefore require a deployment design.

03 / MATHEMATICAL FORMULATION

The model actually evaluated

The v0.8.3 evaluator uses a preliminary density-and-aging policy. It is distinct from the documented v0.7 reservation/backfill kernel. The equations below describe the reviewed replay implementation; a power-aware production objective remains a separate design target. [2]

Symbol	Meaning
j, m, t	Job, compatible GPU-model pool, replay time
g_j, τ_j, a_j	GPU demand, observed duration in seconds, submission time
w_j, W, A	Modeled class weight, aging window, maximum aging multiplier
β, γ, K	Duration exponent, GPU-demand exponent, candidate bound

$$A_j(t) = \min[A,\ 1 + \max(0,\ t - a_j)/W] \quad (1)$$

$$q_j(t) = w_j A_j(t) \,/\, [\max(1,\ \tau_j)^{\beta}\ g_j^{\gamma}] \quad (2)$$

The first K queued candidates are filtered for a feasible placement before ranking. The duration floor is one second. Score is a ranking quantity; it is not dollars per second.

$$C'_m = \max[sC_m,\ \max_{j:\ m \in M(j)}\ g_j] \quad (3)$$

C_m is observed capacity, M(j) the compatible models, and s the stress scale. Maximum demand is zero if no job is compatible. The floor prevents scaling alone from excluding the largest compatible job.

$$m^* = \arg\min_{m \in F(j,t)}\ [C'_m - U_m - g_j] \quad (4)$$

F(j,t) contains compatible pools with $g_j \le C'_m - U_m$. Best fit debits one pool exactly once; U_m is used capacity. An empty feasible set leaves the job waiting.

Frozen stress configuration. K = 64; β = γ = 0.5; W = 14,400 s; A = 8; HP:Spot weights = 5:1; s = 0.01. These are the selected scenario parameters, not universal deployment defaults. [3,4]

For a future facility-aware formulation, power would be constrained in kW and temperatures or thermal models in their own units. Power and temperature must not simply be added in a score denominator. This revision retires the older paper’s mixed “power + thermal” expression as an implementation claim.

04 / ALGORITHM AND IMPLEMENTATION

From queue state to a replay decision

The evaluator advances through arrivals and completions. Durations are read retrospectively from the trace. At each event, available capacity is updated before the next admission attempt. The same normalized input, capacity model and time horizon are used for every policy. [2]

01 **Advance time**
Release completed jobs; append newly arrived jobs in deterministic order.

02 **Find a feasible candidate**
Inspect queue head for FIFO or the bounded window for the other policies.

03 **Choose and place**
Select by duration or AREG score; choose one compatible best-fit model pool.

04 **Record the outcome**
Schedule a simulated completion, debit capacity once and record local selection time.

Capability status at the reviewed revision

Component	Verified scope	Remaining boundary
Observation	Pod/node adapter; Prometheus/DCGM normalization	Real non-production environment integration
Evidence	Versioned contracts; canonical hashes; in-process chain	Durable store, complete decision API and report
Public replay	FIFO, shortest-feasible and AREG preliminary	No live cluster connector or actuation
Compatibility	One compatible aggregate GPU-model pool per job	No node fragmentation, CPU, memory or network placement
Policy freeze	Policy and evaluation fingerprints in stored reports	New evidence needed for deployment generalization

Capped aging encourages older work but does not guarantee starvation freedom. The current replay should not be credited with the older kernel’s oldest-job reservation or exact subset search. Those historical capabilities remain documented evidence whose original source snapshot has not been recovered into the current repository. [1,2]

Earlier test counts describe earlier builds. This paper reviews current source and stored reports; it does not claim a new test run, live-cluster trial, or independently reproduced full-trace benchmark.

05 / EVALUATION DESIGN

Freeze the experiment before judging it

The public evidence program separates development from frozen evaluation. The 2026 Alibaba trace supplies an earlier development baseline; a separate 2023 population produced a non-discriminating aggregate-capacity holdout. The newer v0.8.3 stress test uses the 2023 gpuspec20 and gpuspec25 variants to examine compatibility sensitivity. [1,9]

Those two variants share their underlying workload and primarily differ in GPU-constraint assignments. They are not independent workload populations. Capacity is deliberately modeled at 1% of each observed pool, subject to the maximum-compatible-job floor. Neither the scale nor the replay state describes Alibaba’s actual available production fleet.

Comparator and search discipline

FIFO preserves strict queue order. Shortest-feasible chooses the shortest job in the same bounded candidate window used by AREG. Results are reported against the strongest value-blind comparator by completed weighted value. Production schedulers can have richer objectives and constraints; these policies are engineering comparators.

Development searched candidate limits {32, 64, 128}, six duration exponents, five GPU exponents, three aging windows and three caps: a declared 810-configuration grid. The selected configuration was frozen before the gpuspec25 evaluation. Reusing that holdout for further tuning would consume its holdout status.

Gate	Frozen requirement	What current traces establish
Weighted value	At least +15% vs strongest comparator	Evaluable under modeled class weights
Queue wait	P95 regression no greater than 10%	Evaluable as the censored replay metric
Decision time	P95 no greater than 50 ms	Local selection routine only
GPU capacity	Zero modeled capacity violations	Pool-level capacity check
Facility constraints	Zero power/thermal violations	Not evaluable; telemetry unavailable
Non-actuation	Every decision remains unapplied	No cluster mutation path in replay

All required gates must pass together in the same evaluation. “Not evaluable” is not a pass. Policy and evaluation fingerprints make configuration changes detectable; they do not establish scientific independence or remove modeling bias. [1-4]

06 / EXPERIMENTAL RESULTS

The development gain did not generalize

The v0.8.3 artifacts contain 6,203 replayed jobs from 8,152 source pod rows, with 1,949 CPU-only or invalid/unscheduled rows excluded by the adapter. Both variants use 1,213 source nodes. The values below come directly from the stored development and holdout reports. [3,4]

Metric	Development: gpuspec20	Holdout: gpuspec25
Weighted value, AREG / baseline	20,432 / 19,176	20,420 / 20,421
Weighted-value change	+6.549854%	-0.004897%
Completed jobs, AREG / baseline	6,168 / 5,796	6,164 / 6,165
P95 wait, AREG / baseline	31,450 / 69,205 s	10,878 / 9,572 s
P95 wait change	54.555307% lower	13.643962% worse
AREG local decision P95	0.019840 ms	0.015694 ms
Capacity violations / applied actions	0 / 0	0 / 0
Overall frozen gate	Fail	Fail

MODELED WEIGHTED-VALUE CHANGE

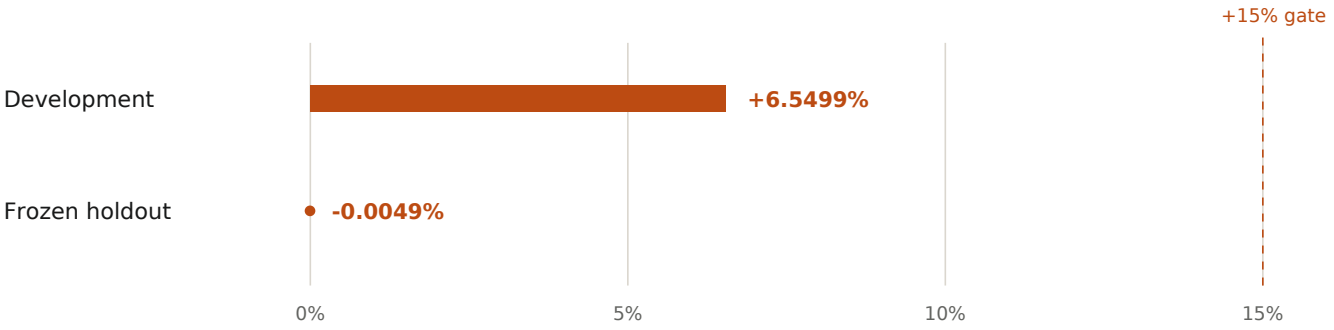


Figure 2. Same linear scale for both evaluations. The holdout marker is essentially at zero; its exact negative value is labeled. Baseline: shortest-feasible.

Interpretation. The holdout completed one fewer Spot-class job with the same number of HP-class completions. Both the weighted-value and queue-wait gates failed. Zero pool-capacity violations do not establish facility power or thermal safety, which remain unmeasured.

Earlier evidence. The 2026 development run reported +0.009164% modeled value; the separate 2023 aggregate holdout tied at 0%. The historical v0.7 paper reported a best +12.1% replay, also below its +15% gate. These differently scoped experiments must not be pooled into a single product-performance number. [1]

07 / VALIDITY AND INTERPRETATION

What the evidence does and does not say

Value is modeled, not commercially measured

HP and Spot receive weights of five and one. In the 2023 adapter, LS is mapped to HP and Burstable/BE to Spot. These are scenario assumptions, not a discovery of customer revenue. An increase in weighted completions establishes a result only under those weights and that evaluation horizon.

Retrospective durations give the policies extra information

Observed schedule-to-deletion or execution duration is available to the replay before selection. A live service would need estimates available at decision time. Shortest-feasible and AREG therefore benefit from retrospective knowledge. Duration uncertainty, cancellations, retries and runtime variation must be evaluated before production comparisons.

The reported P95 is a censored lower-bound statistic

$$\tilde{w}_j = \max[0, \min(s_j, T) - a_j] \quad (5)$$

s_j is replay start time; for jobs not started by horizon T, substitute T. The code computes the empirical 95th percentile of these observed-or-censored waits.

This is not a censoring-adjusted estimate of eventual P95. Waits can continue beyond T. Report pending jobs and horizon sensitivity. If comparator wait is zero, explicitly report that a relative change is undefined.

Compatibility perturbation is narrower than external validation

The development advantage is sensitive to compatibility assignments in a shared workload. This neither rejects all possible AREG policies nor validates the current policy in production. Aggregate pools omit node fragmentation, memory, fabric topology and interference.

Timing and utilization need careful interpretation

Decision P95 measures the local selection routine. Network transport, telemetry collection, evidence persistence and service latency are excluded. Replayed GPU utilization is a modeled occupancy statistic, not device utilization measured by DCGM. The stress model’s poor FIFO behavior is not evidence about a production FIFO implementation.

> **PRESENT CONCLUSION**
>
> The current evidence supports continued engineering of a reproducible evaluation system. It does not establish general scheduling superiority, customer savings or a reduction in required data-center footprint.

08 / SECURITY AND DEPLOYMENT

Keep the production boundary explicit

The accepted architecture places a future shadow deployment inside the customer’s security boundary. It observes approved state with credentials incapable of applying scheduling decisions. The existing scheduler retains control of binding, eviction, admission and workload execution. The invariant **applied=false** is part of the decision contract. [5]

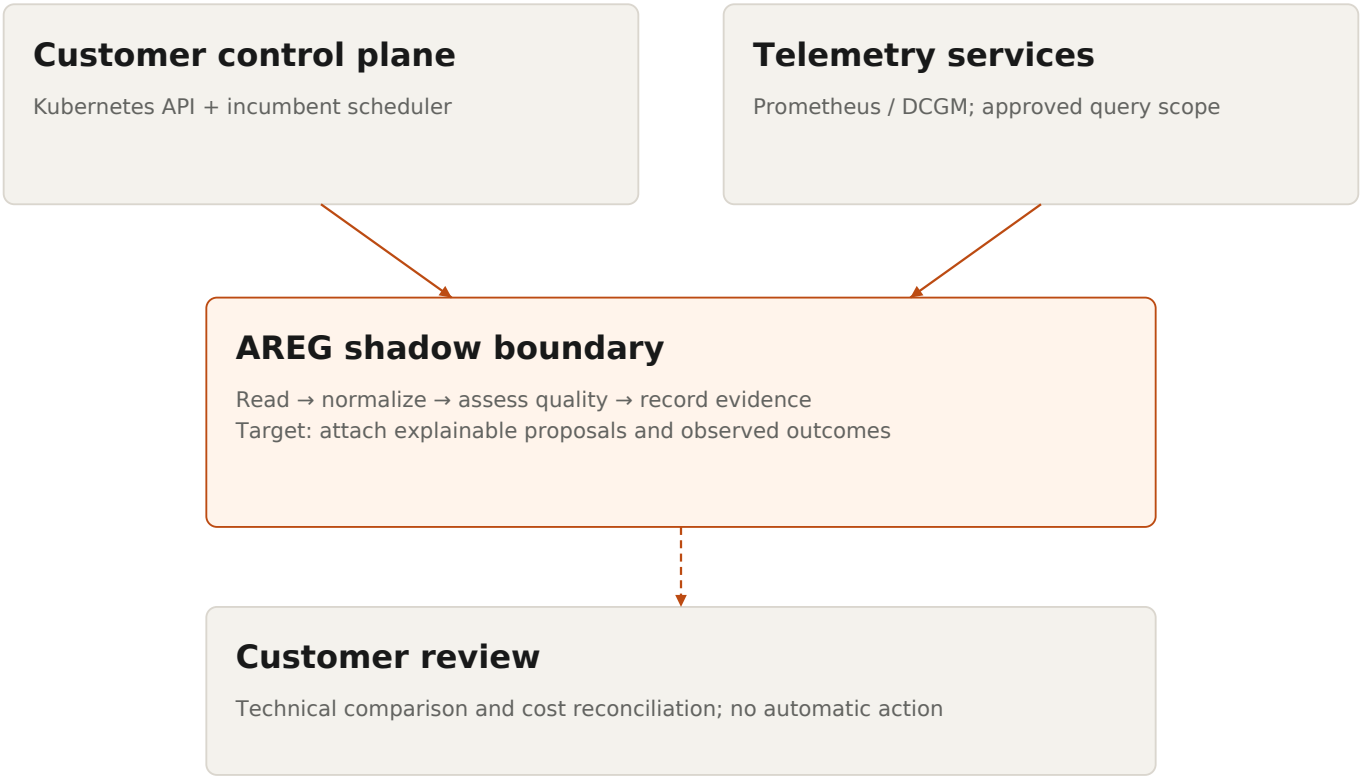


Figure 3. Target customer-hosted deployment. The broken arrow denotes reporting integration still to complete. No arrow returns commands to production.

Access. Scope Kubernetes permissions to the declared read operations and resources. Review rendered RBAC before connecting an environment; reject unexpected mutation verbs or resources. RBAC is a deployment control and must be verified in the actual cluster. [5,6]

Transport. The Prometheus client is designed for GET-only instant queries, TLS outside explicit loopback tests, exact-origin allow-listing, redirect rejection, bounded requests and responses, disabled ambient proxies and runtime token injection. Credentials must not enter source control or evidence records. [5,7]

Data handling. Exclude prompts, model contents, datasets, logs, commands, environment variables and secrets. Define evidence retention, tenant separation, pseudonym-key handling and export permissions with the customer. Salted identifiers reduce exposure but do not make rich operational traces automatically anonymous.

09 / MEASUREMENT FRAMEWORK

Translate capacity into economic evidence

The commercial hypothesis is that better decisions can increase useful outcomes on already-paid-for infrastructure or defer additional capacity. Scheduling more jobs is not automatically equivalent to using fewer GPU-hours, reducing a cloud invoice, or avoiding a server purchase. Each claim requires its own measurement boundary.

$$V_{\pi}(T) = \Sigma_j \, w_j \cdot 1[c_{j,\pi} \leq T] \tag{6}$$

π denotes a policy; $c_{j,\pi}$ is job completion time. Count weighted completions by T. Customer pilots require customer-approved w_j; public replay weights remain modeled.

$$U_V = 100 \times (V_{AREG} - V_B) / V_B \tag{7}$$

B is the strongest declared comparator by weighted value, and V_B must be positive. This is the replay uplift definition, not a savings percentage.

$$H_{\pi} = \Sigma_j \, g_j \tau_{j,\pi}/3600; \; \Delta H = H_B - H_{AREG} \tag{8}$$

GPU-hours avoided require a matched workload/outcome and service constraints. Reordering fixed-duration jobs alone does not reduce their intrinsic GPU-hours. Capacity headroom and energy are separate measures.

$$\Delta C = \Sigma_m \, r_m \Delta H_m - C_{incremental} \tag{9}$$

This illustrative model applies to avoidable billed GPU-hours at customer-authoritative rates r_m. Include software, integration and operational costs. Committed or owned capacity needs different accounting.

Claim	Evidence required
Useful work per dollar	Approved outcome weights; reconciled cost allocation; same observation period
Realized invoice savings	Billing records plus an attributable reduction in avoidable charges
Capacity expense deferred	Demand forecast, protected SLOs, headroom duration and a procurement decision
Energy reduction	Time-aligned power integration at a declared device or facility boundary

Report measured, modeled, inferred and unavailable values separately. Keep realized spend reductions, modeled capacity deferral and additional output distinct to prevent double-counting the same benefit. Current public evidence has no customer-authoritative financial inputs.

10 / VALIDATION PROTOCOL

A pilot that can return a clear answer

The next milestone is a bounded, read-only evaluation on one workload family with a customer-owned definition of value. A proposed 30-60 day observation period is a planning range, not a scheduled engagement or an assurance that the current build is ready to connect.

Stage	Deliverable	Exit condition
Prepare	Scope, permissions, value policy, comparator and frozen thresholds	Customer approves data boundary; required signals are available
Observe	Baseline workload and telemetry records	Clock, completeness and identity checks pass
Evaluate	Versioned shadow proposals and counterfactual report	No actuation; comparable windows; uncertainty disclosed
Review	Technical result plus reconciled economic assessment	Each frozen gate has one recorded pass/fail/not-evaluable result

Engineering work before a complete customer pilot

Integrate the service. Connect approved observations to schema-valid, independently checked proposals. Capture incumbent actions and outcomes, not only what AREG would have recommended. Validate the complete flow on a real non-production environment.

Preserve evidence. Add durable storage, export controls, integrity anchoring and reporting. Record each proposal’s snapshot, policy, constraints and uncertainty; reproduce sampled decisions.

Exercise failure behavior. Cover stale and partial telemetry, clock skew, connector loss, evidence-store outage, token rotation and load. Missing safety inputs must yield an explicit blocked or unevaluable outcome.

Improve external validity. Test duration predictions, node placement and customer-defined value on a new held-out workload. Slurm integration and production actuation remain separate work.

Requirements traceability

Requirement	Artifact to inspect	Acceptance evidence
Non-actuation	RBAC + decision contract	No mutation privilege; every applied field is false
Safety	Quality flags + constraint report	Required facility signals present; zero violations
Performance	Frozen benchmark + timing records	All uplift, wait and latency conditions pass
Reproducibility	Hashes + policy + evaluation record	An independent reviewer reproduces sampled results

11 / PROVENANCE AND REPRODUCTION

Make the result reproducible

This paper is anchored to software revision **cf10591a880f175cc01c9b2c7f97b5f1b6c37df6** on the v0.8 development branch, inspected on 8 September 2026. Source and saved artifacts were reviewed; the complete raw trace was not downloaded or rerun for this revision. [1-5]

Evidence manifest

Record	Repository path
Policy implementation	src/yirah/public_evidence/replay.py
Method and limitations	docs/PUBLIC_EVIDENCE.md
Development artifact	evidence/public/alibaba-gpu-2023-gpuspec20-compatible-development-v3.json
Holdout artifact	evidence/public/alibaba-gpu-2023-gpuspec25-compatible-holdout-v3.json
Version provenance	docs/VERSION_HISTORY.md

Frozen identifiers

Policy fingerprint

```
sha256:26d2235e9225a43574a6116ecfa4954c
d652637133deff6f4eb9351080a3775f
```

Evaluation fingerprint

```
sha256:d9a699b87e9e97c465791fd2a4396b16
dbb8d34e24d55c15fc81c4c40044d527
```

Policy identity covers formula and tunable parameters. Evaluation identity additionally covers the compatibility profile and capacity stress. The report also retains input SHA-256 digests, source-row exclusions, modeled assumptions and a per-gate status.

Reproduction sequence

1. Obtain the original source files from the publisher under their applicable terms; retain input digests. The repository does not vendor raw datasets.
2. Use the exact software revision and declared dataset adapter. Verify row counts, exclusions and the evaluation horizon.
3. Apply the frozen policy and evaluation configuration without tuning on the holdout. Verify both fingerprints.
4. Run all comparators under the same model. Export complete policy metrics, unavailable signals and the overall gate decision.

Source-history boundary. Earlier compression milestones and v0.7 remain historical records. Missing original source has not been reconstructed or attributed to the current replay. [10]

12 / SOURCE RECORD

References

Repository references below are pinned to the reviewed revision. External documentation was accessed on 8 September 2026. A source describes its own scope; documentation for a platform does not certify Yirah's integration with that platform.

[1] Yirah AI. Public evidence program and version provenance.
Defines experiments, data assumptions, release gates and historical results.
Open source ↗

[2] Yirah AI. Public replay implementation.
Authoritative score, candidate selection, placement, capacity scaling and metric definitions.
Open source ↗

[3] Yirah AI. v0.8.3 compatibility development evidence.
Stored gpuspec20 metrics, configuration and fingerprints.
Open source ↗

[4] Yirah AI. v0.8.3 frozen compatibility holdout.
Stored gpuspec25 metrics, input digests and failed gate conclusion.
Open source ↗

[5] Yirah AI. Architecture decision records and adapter boundaries.
Read-only boundary, versioned contracts, Kubernetes and Prometheus/DCGM design.
Open source ↗

[6] Kubernetes Authors. Using RBAC Authorization.
Roles, resources and verbs used to scope cluster permissions.
Open source ↗

[7] Prometheus Authors. HTTP API.
Instant-query API and response structure.
Open source ↗

[8] NVIDIA. DCGM Exporter.
Prometheus-compatible GPU telemetry export.
Open source ↗

[9] Alibaba Cluster Trace Program. 2023 GPU trace.
Source dataset family; GPU-constraint variants and trace documentation.
Open source ↗

[10] Yirah AI. Version history and source provenance.
Separates document-only historical milestones from surviving source.
Open source ↗

Revision note

This revision replaces the earlier build summary, updates the mathematics to the implemented public policy, includes the v0.8.3 development and holdout evidence, clarifies censoring and economic attribution, and redesigns the document for technical review. It does not change the software or the frozen release criteria.