A Budget-Based Frugal Observability Governor for Decentralized Edge Agents

Qiao Zhang
Orange Innovation, Belfort, France
qiao.zhang@orange.com

ATLAS Workshop @ ICWE 2026

9 June 2026

Outline

Message

Servers can analyze telemetry, but edge devices still pay to send it.

Proposal

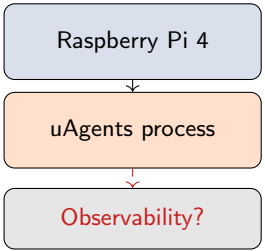
A frugal governor emits only anomalies and budgeted state changes.

Edge Agents Create an Observability Gap

- Small edge nodes run agents near physical sensors.
- We use **uAgents**: lightweight, decentralized, and P2P.
- Runtime footprint is small: **62 MB RAM** on Python 3.12.
- But uAgents has no built-in observability mechanism.

Core Question

How do we monitor edge agents without turning monitoring into the largest workload?



Why the Obvious Answers Fail

OpenTelemetry

- Standard observability SDK
- Adds trace/span metadata around each event
- Can produce larger records than raw JSON logs

Probabilistic Sampling

- Very small logs
- Can miss rare anomalies because drops are random
- No guarantee that important anomalies are kept

Needed

Reduce bandwidth and memory pressure while preserving safety-critical information.

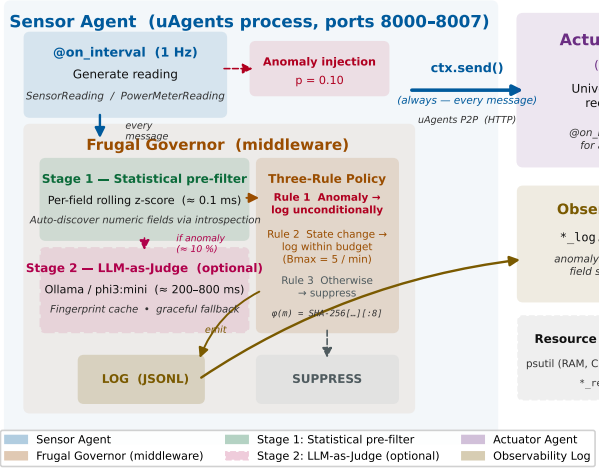
Contributions at a Glance

- **Frugal Governor** middleware for uAgents with a three-rule budget policy
- **Fixed-τ and Threshold detectors** for schema-specific anomaly priority
- **Generic extension** using Pydantic introspection and optional local LLM-as-Judge
- **LangGraph validation** for graph-based LLM orchestration
- **8-method evaluation** over controlled trials

Short Pitch

The Governor is not "less logging"; it is **priority-aware observability under a budget**.

Where the Governor Fits



- Middleware inserted between uAgents handler and network send call.
- No uAgents source-code modification.
- Three components: anomaly detector, state fingerprinter, budget counter.

Three-Rule Decision Policy

Rules are evaluated in strict priority:

Rule 1 — Anomaly Priority

If anomalous $\Rightarrow$ **emit unconditionally**.

Rule 2 — State-Change Budget

If fingerprint changed and $B < B_{\max} \Rightarrow$ emit.

Rule 3 — Budget Suppression

Otherwise $\Rightarrow$ suppress: zero bytes.

Default Budget

- $B_{\max} = 5$ emissions per minute
- Counter resets every 60 seconds
- SHA-256 fingerprint over message content
- Anomalies never consume budget

Safety Property

Budget exhaustion cannot hide an anomaly.

Detector Variants

Fixed-τ

- Temperature anomaly if $T > 27.0\,^{\circ}\mathrm{C}$
- Deterministic and cheap
- Requires domain threshold

Threshold

- Rolling z-score over $W = 20$
- Flags $|T - \bar{T}|/\sigma_T > 2.0$
- Removes manual threshold, still schema-specific

Progression

Fixed-τ
manual, schema-specific

$\Downarrow$

Threshold
automatic, schema-specific

$\Downarrow$

Generic
automatic, schema-agnostic

How the Rolling Z-Score Works

Step 1 — Sliding Window ($W = 20$)

For each new sample x_t, update local baseline from recent history:

$$\bar{x}_t, \sigma_t \leftarrow \text{last } W \text{ samples}$$

Step 2 — Z-Score

$$z_t = \frac{x_t - \bar{x}_t}{\sigma_t + \varepsilon}, \quad \varepsilon = 10^{-6}$$

Guards against division by zero on constant signals.

Step 3 — Decision ($\theta = 2.0$)

Anomaly $\iff |z_t| > \theta$. Flags readings $> 2\sigma$ from the recent local mean.

Why Sliding Window?

- **Adaptive baseline**: a gradual temperature drift does not trigger an alert; a sudden spike does.
- **No domain knowledge**: θ is dimensionless — works for temperature, voltage, frequency, etc.
- **Cheap per update**: updates compact summary statistics, not all past readings.
- Applied **per field**: each numeric field in the Pydantic model has its own z-score tracker.

Generic Extension: Schema-Agnostic Governor

Stage 1 — Statistical Pre-filter (≈0.1 ms)

- Introspects **any** Pydantic model via `.dict()`.
- Keeps one rolling z-score tracker per numeric field.
- Fires if any field exceeds $|z_t| > 2.0$.

Stage 2 — LLM-as-Judge (200–800 ms, optional)

- Invoked **only** when Stage 1 fires.
- Local `phi3:mini`; prompt includes values and z-scores.
- Returns `anomaly_detected`, `severity`, `reason`.
- Fingerprint cache: ≈4–5 LLM calls per trial.

Schema-Agnosticism in Practice

Same governor code handles both schemas without modification:

- `SensorReading`: {temperature, humidity}
- `PowerMeterReading`: {voltage, current, power_factor, frequency}

Graceful Degradation

If Ollama is unavailable, Stage 2 is skipped — Stage 1 result is used directly. No crash, no data loss.

The 8 Evaluated Methods

Method	Category	Description
Verbose	Baseline	Logs every message unconditionally; anomaly label still uses $T > 27^{\circ}$C.
Probabilistic	Sampling	Logs each message with 10% probability. Cheap but misses 90% of anomalies.
Frugal Fixed-τ	Frugal	Governor with hardcoded threshold ($T > 27^{\circ}$C). Budget suppresses normals.
Frugal Threshold	Frugal	Governor with rolling z-score ($W = 20$, $\theta = 2.0$). No manual threshold.
OTel SDK	Industry	OpenTelemetry exporter. Logs every event with trace metadata; no filtering.
Generic Stat-only	Generic	Schema-agnostic governor; Stage 1 (z-score) only, LLM disabled.
Generic LLM-judge	Generic	Schema-agnostic governor; Stage 1 + Stage 2 (local `phi3:mini`).
LangGraph	LLM-orchestrated	LangGraph `StateGraph` + ChatOllama pipeline per message.

Design Goal

Cut log volume while keeping threshold-labeled anomaly recall close to full-logging baselines.

Key Trade-off

LLM methods add auditability but cost RAM and latency.

Evaluation Setup

Scenario A

SensorReading agent, 120-second trials, five random seeds. Normal readings centered at 22°C; anomalies centered at 28.5°C; injection probability 10%.

Scenario B

Adds Generic and LangGraph variants, plus a PowerMeterReading schema to test schema-agnosticism.

Methods

Verbose, Probabilistic, Frugal Fixed-τ, Frugal Threshold, OTel SDK, Generic Stat-only, Generic LLM-judge, LangGraph.

Threat to Validity

Experiments run on an Intel i9 laptop as a Raspberry Pi 4 proxy. Physical ARM validation is future work.

Scenario A: Key Numbers

Method	Log size	Reduction	Recall
Verbose	29,815 B	0%	91.9%
Probabilistic 10%	1,444 B	95.2%	7.2%
Fixed-τ	3,458 B	88.4%	91.9%
Threshold	4,356 B	85.4%	75.2%
OTel SDK	44,334 B	-48.7%	93.7%

Best Safety/Size Trade-off

Fixed-τ matches Verbose recall while cutting logs by **88.4%**.

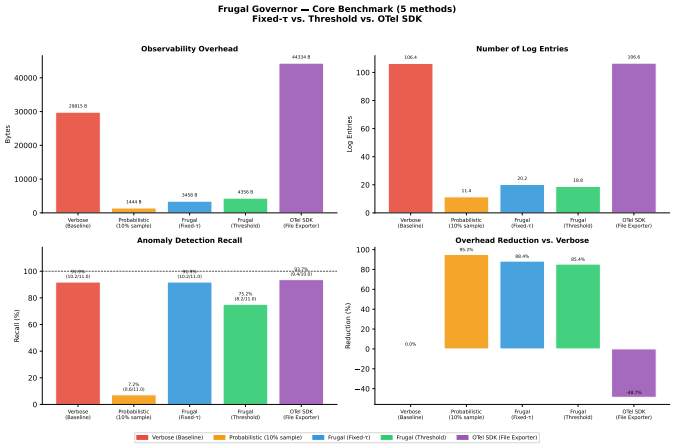
Unsafe Shortcut

Probabilistic sampling cuts logs by **95.2%**, but misses most anomalies.

Cloud Baseline

OTel keeps high recall but produces **49% more data** than Verbose.

Frugal Governor Core Benchmark (5 methods)



Read the Trade-off

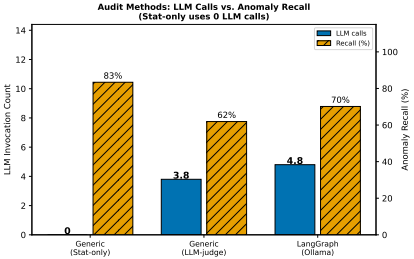
- Probabilistic is smallest, but recall collapses.
- OTel keeps recall, but emits more data than Verbose.
- Fixed-τ and Threshold cut bandwidth while preserving anomaly priority.

Scenario B: Generic and LangGraph Trade-offs

Method	Size	Red.	Recall
Generic Stat	5,628 B	81.1%	83.4%
Generic LLM	4,574 B	84.7%	61.9%
LangGraph	6,903 B	76.8%	70.2%

Interpretation

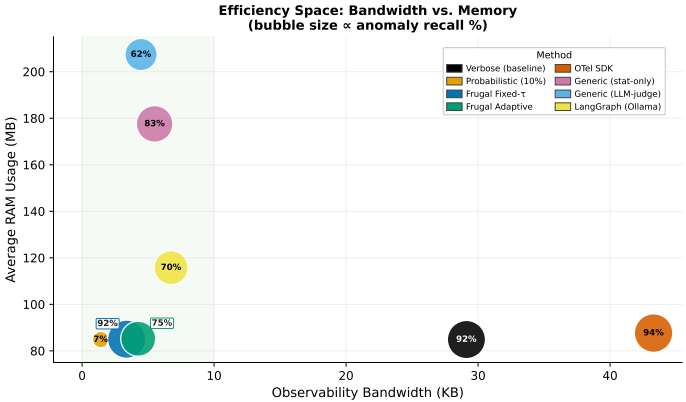
LLM variants add human-readable reasons and severity. They are for auditability, not raw recall.



Efficiency

Only ≈4–5 LLM calls per 120-second trial because the statistical filter and cache absorb most traffic.

Efficiency Space



- Lower x-axis: less bandwidth.
- Lower y-axis: less RAM.
- Larger bubble: higher anomaly recall.

Best Edge Balance

Frugal Threshold and Fixed-τ stay near the low-RAM, low-bandwidth region.

Cloud Baseline

OTel keeps high recall but moves far right because it logs more data than Verbose.

Conclusion

Takeaway

Best trade-off: **rule-first frugal governance**, not heavier telemetry or LLM judging.

Key Result

Fixed-τ: **88.4%** reduction with **91.9%** recall.
Threshold: automatic alternative, **85.4%** reduction.
OTel keeps recall but emits **48.7% more** than Verbose; LLM adds auditability, not raw recall.

Design Principle

Budget normal traffic; never budget anomalies.

Limitations and Next Steps

Limits Seen in Tests

- Synthetic readings, not physical sensors
- x86 laptop proxy for Raspberry Pi 4
- Fixed 120 s trials favor faster methods
- LLM calls add latency/RAM and reduce throughput
- Fixed seeds, but unequal sample counts
- Fixed z-score/window can miss drift bursts
- CO_2 metric inconclusive in 120 s runs
- Cross-field-only anomalies not tested yet

Next Steps

- Physical Pi 4 + real sensor trials
- Fixed-length replay trace for fair recall
- Longer runs for energy/CO_2 evidence
- Score before baseline update; tune window
- Add cross-field anomaly scenarios
- Async/smaller local LLM path

Thank You

Questions?

- qiao.zhang@orange.com
- Orange Innovation, Belfort, France