Network-as-a-Product MVP – SLA Metrics & Analytics Infra

John Mull 12/08/2025

My first thoughts are we should consider developing this as an installable python package with getter/setter methods for the metrics data points. This will be essential to providing consistent metrics as a network in Bring-Your-Own-Container pipelines. The package should be versioned and published to PyPi for reuse in ai-runner and pytrickle (both use frame processor patterns). CI tests and even other development tools like scope could benefit from a standard python metrics module.

Add a comment…

Status: Draft → Review → Approved

Authors: Evan | Qiang | Doug | Rick | Cloud SPE | Network Advisory Board

Timeline: 2 Weeks (Complete by End of November)

Press '/' for commands…

1. Overview & Purpose

This RFC defines the **Design Spec for Milestone 1.0 (NaaP MVP)** — to deliver a publicly observable SLA reporting system for the Livepeer Network-as-a-Product (NaaP).

It establishes a unified set of **GPU and Network metrics**, **technical architecture**, and **API interfaces** that enable real-time monitoring of Orchestrator performance and network demand across geographies and workflows.

This RFC establishes the data foundation for self-adaptive scaling and SLA-based orchestration in Milestone 2 by standardizing the telemetry schema and analytic models introduced here.

Success Criteria

Deliverable #1:

A dashboard that displays the core metrics (See Section 4) from the whole AI Network.

User story: As a community member, I can use this dashboard to learn for any specific O, and workflow, how is performing historically, and in realtime, defined by the well-defined set of metrics in Section 4.

Deliverable #2:

A MVP Gateway that specializes in executing the “Test Loads”, to monitor the network performance metrics.

User story: As a community member, I have confidence in the overall network performance, because a dedicated Gateway continuously monitors the network performance and reliability, with a transparent and public contributed high quality testing datasets.

2. User Stories

User Role	As a …	I want to …	So that I can …
Orchestrator	Enroll and monitor my GPU capacity on the Livepeer Network	Know real-time SLA compliance and competitive positioning	optimize service competitiveness through SLA visibility
Gateway Provider	Operate a public gateway utility and load test tool	Validate GPU reliability and feed metrics into network analytics	select the best resources for the workload requests
Inference Provider (e.g. Daydream)	Deploy AI workflows and view service SLA data	Ensure their inferences meet industry-leading latency and cost targets	have full confidence of underlying infra, and its SLAs
Community Observer / Researcher	Access public dashboard and APIs	Monitor network health and transparency of decentralized GPU performance	have a comprehensive view of what NaaP is and how NaaP is doing
Core Engineer	Validate metrics pipeline integrity to ensure the network meets published SLAs	Know the QoS of the network before committing engineering efforts	reduce the overall infra risk

3. Technical Architecture (High Level)

3.1 Overall Architecture

[GPU Node / Orchestrator Agent] - AI Runner

│ GPU Metrics Collector (FLOPS, Latency, CUE, Reliability)

│ AI Runner reports metrics directly to “streamr” infra

▼

[Trickle Protocol Transport Layer]

│ Event Streaming (heartbeat, SLA events)

▼

[Daydream Gateway / Load Tester Gateway]

│ Synthetic workloads → Kafka → ClickHouse Cloud

│ Gateway reports pathway metrics such as (e2e latency, swap rate) directly to “streamr” infra

▼

[Analytics Layer / Metabase Dashboard / Public API]

│ Expose metrics per GPU, per O, per Region, per Workflow

Qiang Han 12/03/2025

@Speedy Bird Technologies review this, to ensure this is a and have any comments you

Show 5 replies

Josh Allmann 12/10/2025

One clarification: the “pull” ap pulling data from _orchestrato is where the data is coming fr

B Brad Perrin 12/06/2025 (edited)

> Cloud SPE sink “streamr’ telemetry events, to a data lake warehouse, for dashbording

▼

~~[Explorer UI] → Public Dashboard (iframe / embedded Metabase)~~

Instead, a standalone Cloud SPE dashboard will be step 1, before make it embeddable in Explorer. This is to reduce the interdependencies, before the MVP is implemented

Daydream Metabase dashboard, is enriched, with new set of metrics defined in Section 4

NOTE: for all telemetry events that define a metrics/measurements, the following id must be present to ensure the completeness of the event:

- Stream ID, used for a unique identifier to define what this metric is measuring about
- O address, used for identifying the target GPU
- model id, used for identifying the model this metric is running on

> example: ai3.brzeczyszczykiewicz.work:8935 #streamdiffusion-sdxl-faceid

- Gateway id/ip, used for identifying the source gateway that originates the stream

---

3.2 Load Test Gateway

The **Load Test Gateway** is a **community-hosted validation gateway** that continuously performs **synthetic workload tests** on the Livepeer Network.

Its goal is to independently verify that the **network-wide SLAs remain in good standing**, using transparent, reproducible, and open data sources.

Purpose

- Serve as a **neutral SLA verifier** that periodically tests Orchestrator performance and network health.
- Provide **benchmark datasets** to simulate real workloads across different models and regions.
- Contribute **reference SLA results** to the public analytics layer (Kafka → ClickHouse → Explorer).

Design Overview

[Test Load Gateway]
- Community hosted nodes (open participation)
- Runs synthetic workloads based on shared datasets
- Periodic jobs scheduled to cover diverse workflows

▼

[Trickle Protocol Transport Layer]
- Streams telemetry and result payloads (FPS, latency, reliability)

▼

[Analytics Layer / SLA Verifier]
- Validates and aggregates results
- Flags outliers and SLA violations

▼

[Explorer Dashboard / Public API]
- Displays network SLA compliance and trends

I think the GPU metrics shoul
from the Orchestrator unless
sure the IPs of the runners wi

Show 1 reply

S Speedy Bird Technologies 12
Yes, runners will push their m
the Orch. This also enables p
operate and as pointed out, n

Thom 12/11/2025
Does someone need to mana
for and maintain the streamr i
it truly decentralized?

Qiang Han 12/03/2025 (edited)
this is implemented in parelle
ensure the “daydream” currer
infra being consistent during t

Show 1 reply

Qiang Han 12/08/2025
not sure I understand your qu
full, let me try:
…

Qiang Han 12/03/2025
@Speedy Bird Technologies
review this, to ensure this is a
and have any comments you

S Speedy Bird Technologies 12
This is correct

Josh Allmann 12/05/2025
How are you thinking about m
latency? Asking because we
looking at that internally too s

Show 4 replies

Josh Allmann 12/10/2025
I think this PR is exactly what
since it measures G <> O late
directly without adding in the

👍 1

Qiang Han 12/03/2025
@Speedy Bird Technologies
share what you have in mind,
complete telemetry metric ev

Show 1 reply

S Speedy Bird Technologies 12/10 (edite
yes, we must have IDs in the
order to correlate information
nodes and jobs. Stream Id is

Attribute	Description
Operator	Community members / contributors (such as Cloud SPE)
Data Source	Open test datasets curated by Daydream & community
Workload Types	Synthetic, workload-specific, and random prompt sets
Update Frequency	Configurable (default: every 10 minutes per region)
Output	Verified SLA metrics pushed to public dashboard
Visibility	Fully transparent and queryable via /gpu/metrics and /network/demand APIs

NOTE:

It is important to have an evolving test dataset that can capture the evolvement of

- deployed workloads, and models
- should avoid one dataset fit—4-all
- prompt diversity, and model specific benchmark prompt set need to be used to capture the performance characteristics
    - For example, core set of stable diff training prompts needs to be sampled for sd related testing, not just any random prompt.

4. Metrics Table

4.1 GPU Performance & Reliability Metrics (TBC, and reviewed by Evan)

Rafal Leszko 12/04/2025
How will this dataset be creat
O can server different model,
output data may not be comp

S Speedy Bird Technologies 12
We would test each model tha
has published as available (w
lookup will drive which param

B Brad Perrin 12/06/2025
I think an important part of thi
be adding a way that syntheti
workloads can be marked an

Show 1 reply

S Speedy Bird Technologies 12
We need to determine first if t
is available to take work. If n
would skip over it until the ne

Qiang Han 12/03/2025
@Speedy Bird Technologies
like to get more of your inputs
what is the cloud spe plan for

Show 1 reply

S Speedy Bird Technologies 12
As mentioned above, we will
repository that defines the tes
parameters per model. These

Josh Allmann 20h
I would say this is not very int
from an operational perspecti
it would be more affected by t
implementation rather than or
performance. Maybe it's helpf
engineers that are optimizing

B Brad Perrin 12/06/2025
This feels like its better from t
Gateway. Unless just talking a
hardware failures?

Qiang Han 12/08/2025
agree. the table here only lists
metrics not yet specify where
be collected.

Josh Allmann 20h

Category	Metric	Measure	Dimension(s)	Target / Range	Aggre	
Performance	Output FPS	Frames per second	O wallet, GPU ID, workflow id, region, time	≥ target fps (model specific)	5s ?	
Performance	Prompt-to-First-Frame Latency	ms	O wallet, GPU ID, workflow id, region, time	≤ XX ms		GPU/Orchestrator
Performance	E2E Stream Latency	ms	O wallet, GPU ID, workflow id, region, time	≤ target		Gateway
Performance	Jitter Coefficient	σ(fps)/μ(fps)	O wallet, GPU ID, workflow id, region, time	≤ 0.1		GPU/Orchestrator
Performance	Startup Time	s	O wallet, GPU ID, workflow id, region, time	≤ threshold		GPU/Orchestrator
Reliability	Failure Rate	%	O wallet, GPU ID, workflow id, region, time	< 1%		GPU/Orchestrator
Reliability	Swap Rate	%	O wallet, GPU ID, workflow id, region, time	< 5%		Gateway
Economics	CUDA Utilization Efficiency (CUE)	Achieved FLOPS / Peak FLOPS × 100	O wallet, GPU ID, workflow id, region, time	≥ 80%		GPU/Orchestrator
Network	Up/Down Bandwidth	Mbps	O wallet, GPU ID, vram, cuda ver, workflow id, region, time	n/a		GPU/Orchestrator

Same thought as the prompt-frame latency: doesn't seem operationally interesting. Help engineers that are trying to op model performance, but they better ways of doing this. This

Implementation Analysis for Section 4.1 Metric

NaaP Metrics Catalog

Metric Name	Category	Measure	Implementation Compl…	Status
✅ Output FPS	Performance	Frames per second	Low	Exists Today
✅ Startup Time	Performance	s	Low	Exists Today
⚠️ Failure Rate	Reliability	%	Medium	Partial
⚠️ Jitter Coefficientss	Performance	σ(fps)/μ(fps)	Medium	Partial
⚠️ Swap Rate	Reliability	%	Medium	Partial
❌ Up/Down Bandwidth	Network	Mbps	Medium	Does Not Ex
❌ E2E Stream Latency	Performance	ms	High	Does Not Ex
❌ CUDA Utilization Efficiency (CUE)	Economics	Achieved FLOPS / Peak FLOP	High	Does Not Ex
❌ Prompt-to-First-Frame Latency	Performance	ms	High	Does Not Ex

4.2 Network Demand Metrics

Metric	Measure	Dimension(s)	Source	Purpose
Total Streams	Count	Gateway, Orchestrator, Workflow, Region, Time	Gateway	Demand profile
Total Inference Minutes	Minutes	Gateway, Orchestrator, Workflow, Region, Time	Gateway	Utilization
Inference Minutes by GPU Type	Minutes	Gateway, GPU Type, Workflow, Region, Time	Gateway	Supply matching
Capacity Rate	% total inference mins/all GPU inference cap mins	Gateway, Orchestrator, GPU Type, Workflow, Region, Time	Gateway	Supply/Demand Gap m
Missing Capacity	count	Gateway, Orchestrator, Workflow, Region, Time	Gateway	Supply/Demand Missin
Staking LPT	LPT	Orchestrator, Workflow, Region, Time	Gateway	Supply
Fee Payment	ETH	Orchestrator, Workflow, Region, Time	Gateway	Demand

? network demand metrics are computed hourly, and can be aggregated at daily, weekly

5. Core Analytics Models

(what initial dashboard should look like to inform what key insights)

Model	Description	Output	Usage

some ideas for your thought

Model Name	Description	Input Data Sources	Output
sla_compliance_score	Weighted aggregation of latency, reliability, and jitter	gpu_metrics	Score (0-100)
cue_efficiency_index	GPU compute utilization adjusted for cost	gpu_metrics + fee_data	CUE %
demand_supply_heatmap	Maps active workloads to available GPUs	network_demand + gpu_inventory	Region×Workflow matrix

Qiang Han 12/03/2025
@Speedy Bird Technologies
extended scope that can be c
milestone 1. however, when y

S Speedy Bird Technologies 12
For the MVP, Cloud SPE inter
publish an API endpoint that c
read by the dashboard. This

6 API Specification

Gateway implements a set of APIs that allow SLAs related metrics to be queried

6.1 Base URL

```
https://api.livepeer.network/v1/sla
```

J John Mull 12/11/2025 (edited)

The `network/demand` endpo
useful for scaling supply.

Could we also aggregate adv
network capabilities to an end
`/network/capabilities`?…

6.2 Authentication

- Public Read API (no auth) for aggregate views
- Rate limiting
- JWT auth for Orchestrator and Gateway private metrics

6.3 Endpoints

(TBC, reviewed by Cloud SPE)

Endpoint	Method	Description	Params	Response
/gpu/metrics	GET	Retrieve per-GPU realtime metrics	o_wallet, gpu_id, region, workflow, time_range	JSON with performance/reliability fields
/network/demand	GET	Aggregate network demand data	gateway, region, workflow, interval	Stream/inference mins stats
/sla/compliance	GET	Return SLA compliance score for Orchestrator	orchestrator_id, period	Score (0-100)
/datasets	GET	List public load test datasets	workflow, type (good, random, bad)	Dataset metadata

```
/gpu/metrics sample output { "o_wallet":
"0x8fC1234a5B9dE0C7d43C8aA92b1D1234567890AB", "orchestrator_agent_version":
```