

Turning GPU-Hours into Tokens, and Tokens into a Service

Agent Router and NVIDIA Dynamo



The Anh Nguyen

CNCF Ambassador · Viettel Networks



Quang Minh Phung

Software Engineer · Viettel Networks



CLOUD & AI INFRASTRUCTURE · 24 OCT 2026



AWS Hanoi Office

Level 7, Grand Terra, 36 Cat Linh, Hanoi

MEET THE SPEAKERS



Quang Minh Phung

Software Engineer @ Viettel Networks

Control plane: create/destroy endpoints, billing,
multi-tenant routing, rate limiting

LinkedIn [/in/minhpg27](#) GitHub [pqminh27](#) Facebook [pqminh27](#)



The Anh Nguyen

Software Engineer, building & shaping Viettel AI/GPU Platform

CNCF Ambassador · Kubestronaut · NVIDIA-Certified Professional, AI Infrastructure · AI Operations
CNCF Glossary VN Lead · Kubernetes SIG Docs VI Approver

LinkedIn [/in/ntheanh201](#) GitHub [ntheanh201](#) Facebook [ntheanh201](#) Blog [ntheanh201.id.vn](#)

BEFORE WE START

Our Dynamo setup is an NVIDIA case study

The **Dynamo part** of this talk is written up on the Viettel Cloud Tech Blog and listed in **NVIDIA Dynamo External publications**. Today we add more: **the control plane for multi-cluster serving**.

NIXL 4

Deployment guide 8

Case study 8

Announcements 13

Showing 8 of 8

Per page 12

 Viettel Cloud

Oct 4, 2026

Turning GPU-Hours Into Tokens: How Viettel Cloud Serves Multiple LLM Families With NVIDIA Dynamo

Case study · Agentic · Routing · Autoscaling · Disaggregated serving

1 of 8 case studies

-  Viettel Cloud
-  Prime Intellect
-  Pinterest
-  Potoroom
-  Cognition
-  H Company
-  Amazon Ads
-  DigitalOcean / Workato

BLOG

NVIDIA

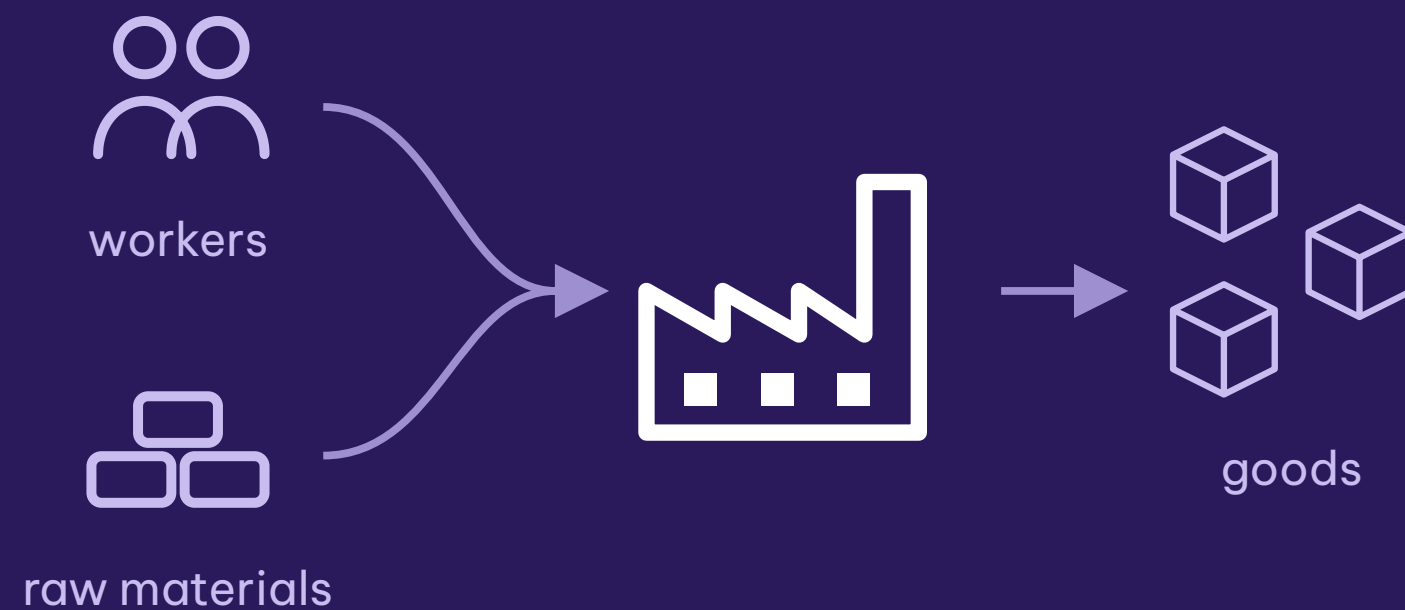
viettelcloud.vn/en/tech-blog/4/142

docs.nvidia.com/dynamo/external-publications

Capacity is now counted in **gigawatts**

INDUSTRIAL AGE

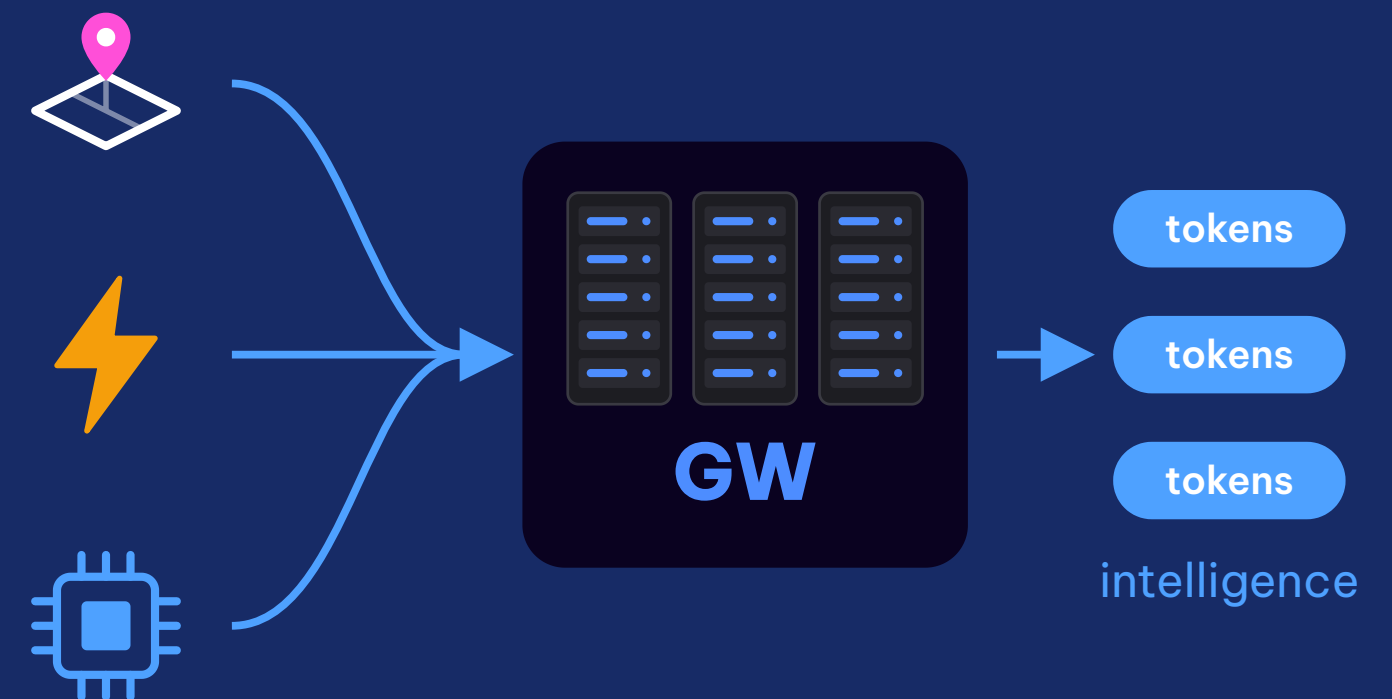
The factory



labor + materials → goods

INTELLIGENCE AGE

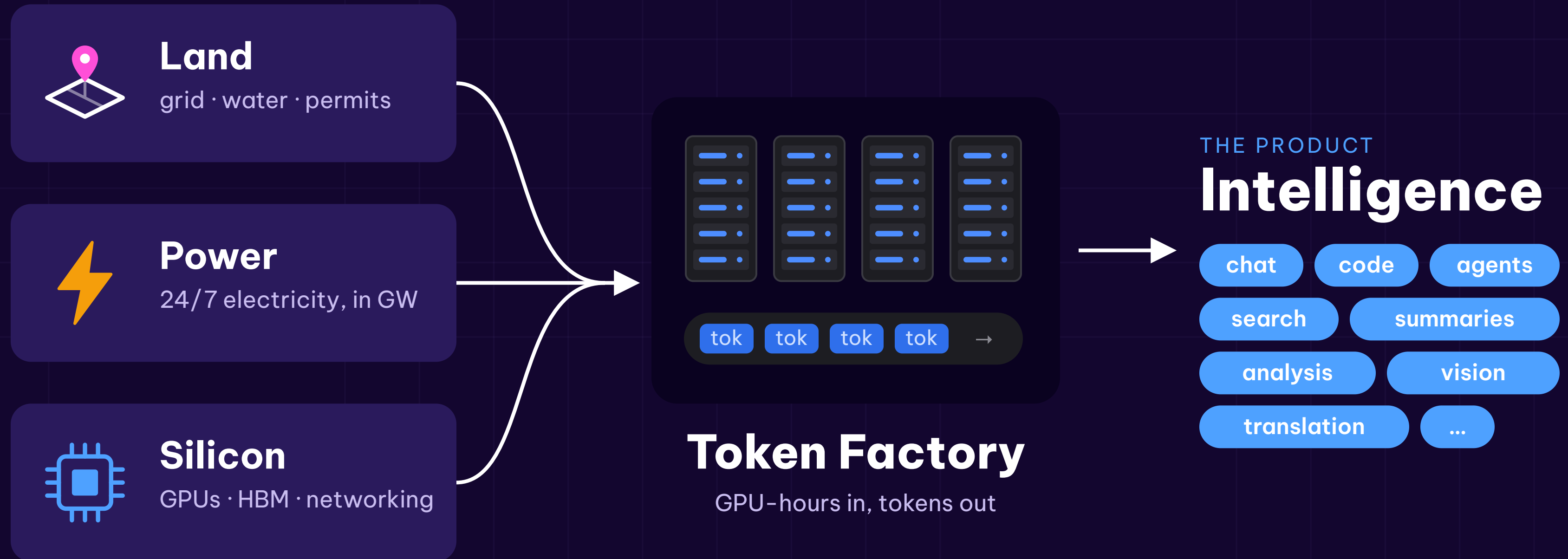
The token factory



gigawatts → tokens → intelligence

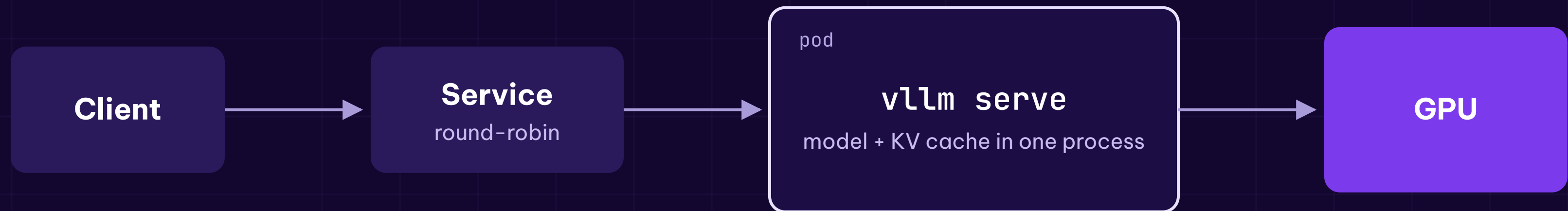
THE NEW PRODUCTION LINE

Tokens are the unit of **intelligence**



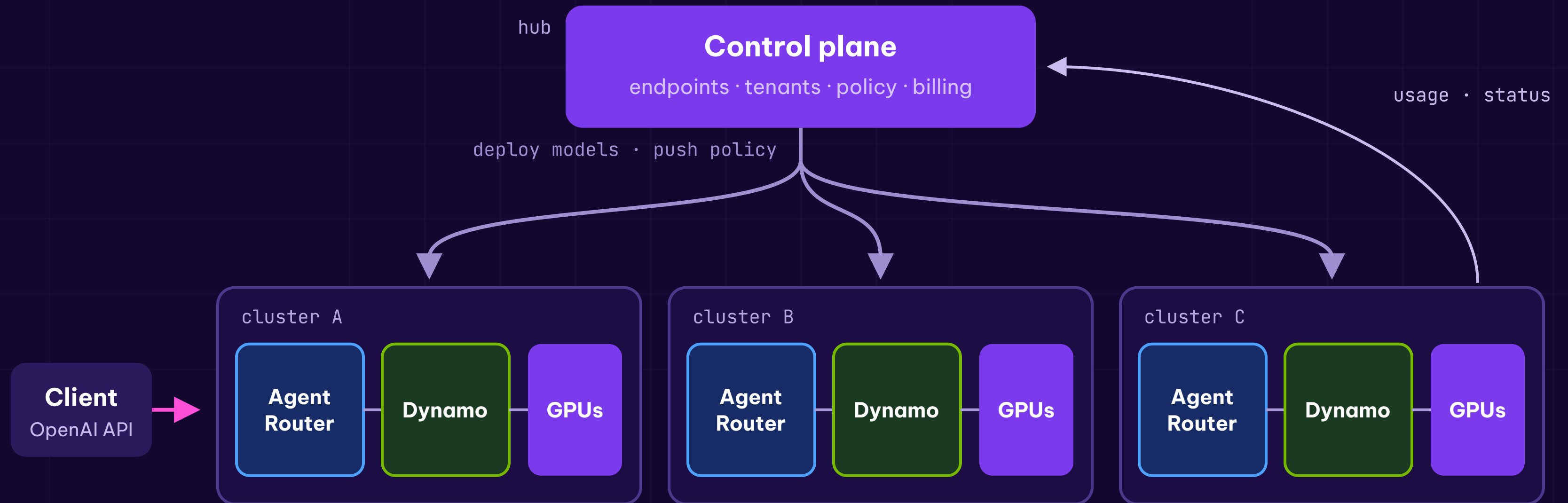
WHERE WE START

We all start here



It works. Until you have many users, many models, and 100k-token prompts.

One hub, many GPU clusters



Hub

decides where each model runs, who can call it, what it costs

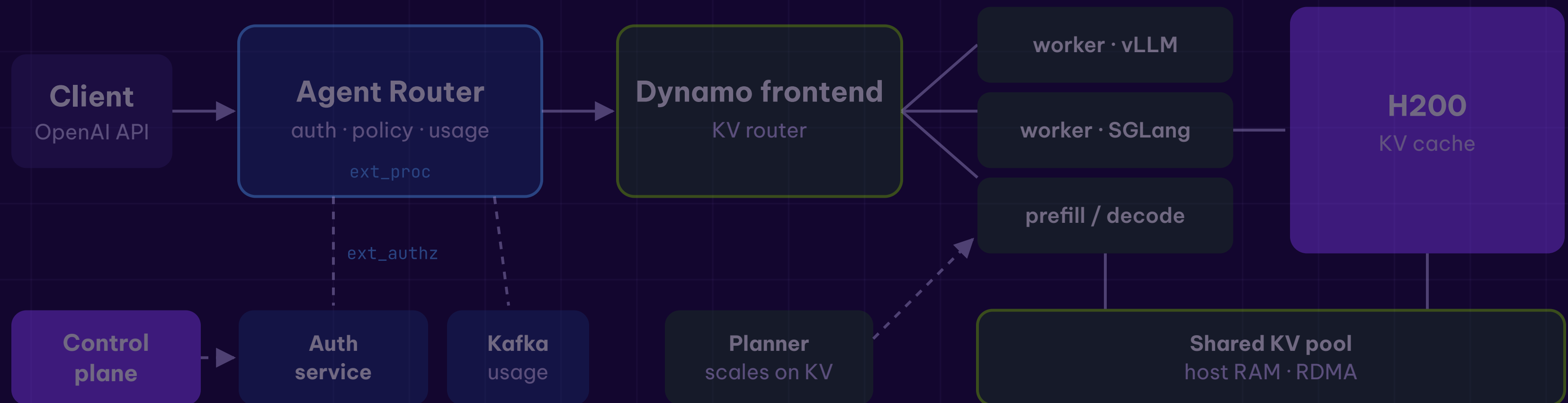
Spoke

serves on its own: auth, routing, KV cache, GPUs

Requests

go straight to a cluster, never through the hub

Add one piece at a time, along the request path



● Part 1: Agent Router

a front door per cluster · identity and limits · regional policy · usage and billing

● Part 2: NVIDIA Dynamo

KV-aware routing · shared KV cache · split prefill / decode · the network underneath · autoscale · picking an engine

PART 1

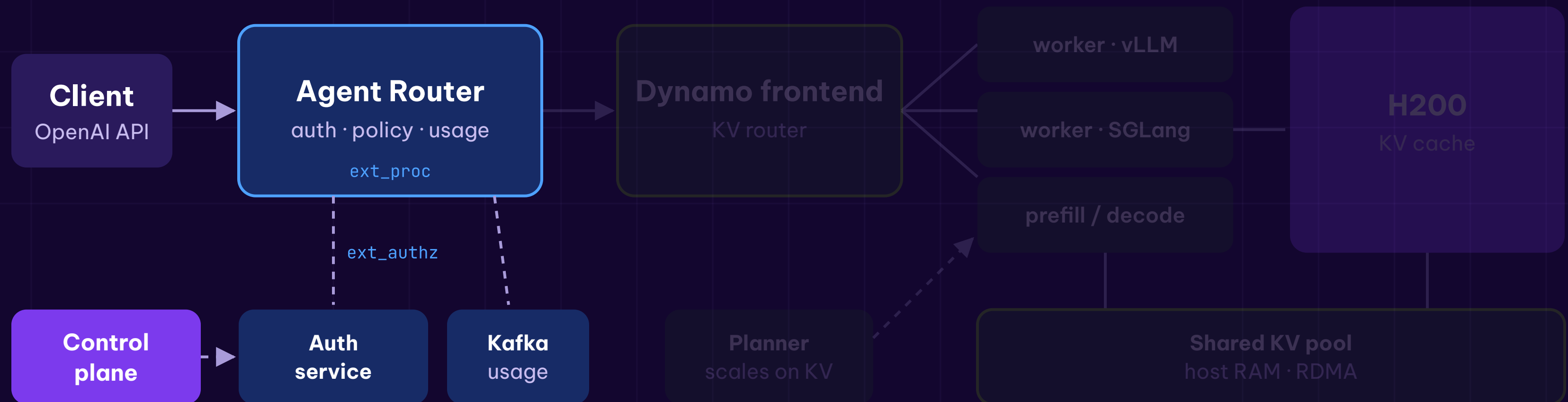
AGENT ROUTER

Who gets in, and how much they can use

OPEN SOURCE · AN AGENTIC AI FOUNDATION PROJECT · FORMERLY ENVOY AI GATEWAY

A front door for each cluster

Authentication before inference. Usage tracking for billing. Each cluster serves its own endpoints behind an Agent Router.

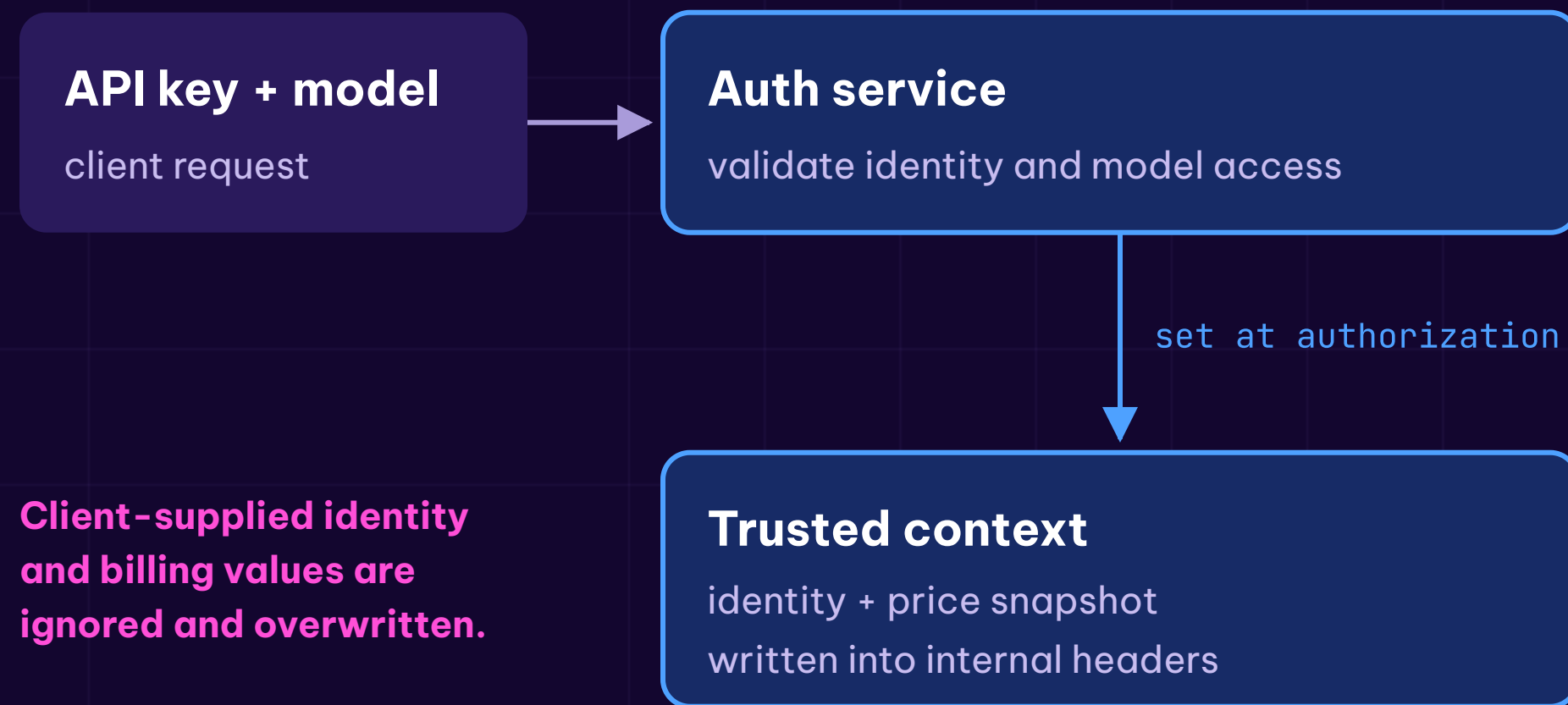


Formerly Envoy AI Gateway. Usage events to Kafka carry trusted identity and the price snapshot.

Identity and admission

Public API key flow. The **auth service** decides who is calling and what they may use, before the model runs.

AUTHENTICATION AND TRUSTED CONTEXT



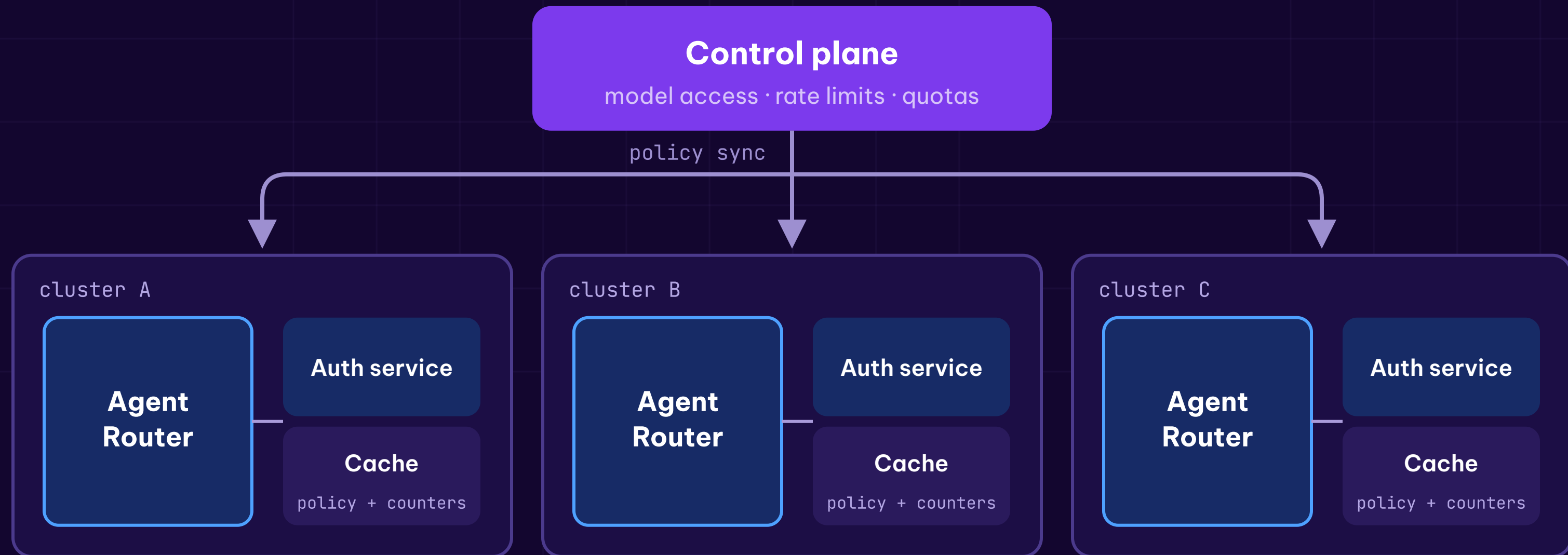
Configured checks: request rate · concurrency · token limits / quota · credit

All applicable checks must pass **before inference**.

CONFIGURED LIMITS



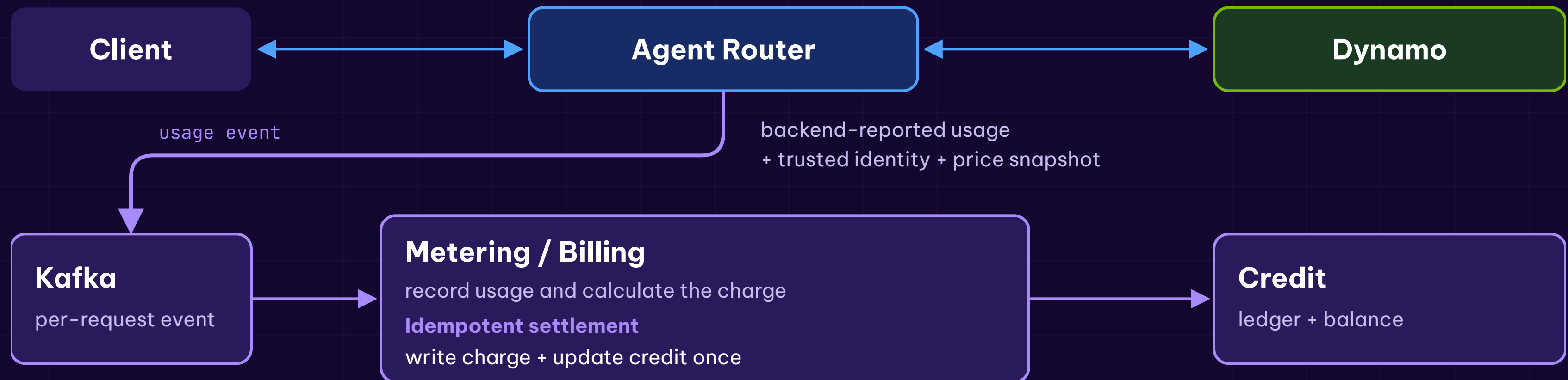
Regional policy enforcement



Normal path: regional checks, **no call to the control plane.**

Actual usage and billing

Usage-pipeline failures **do not block** inference responses.



01 No side doors

Authorize protected routes.

02 Fail closed on auth

Reject if auth cannot be verified.

03 Trust verified context

Ignore and overwrite client values.

Credit balance updates asynchronously · charges use actual usage and the request's price snapshot

PART 2

NVIDIA DYNAMO

More tokens from every GPU-hour

Three problems. None was about the model.

GPU-hours can't be stored. One that doesn't turn into tokens is simply lost.

1 Headroom left on the table

The agentic-coding endpoint got less throughput per GPU than the hardware allowed. First-token latency got worse under load.

→ routing • shared KV • split P/D

2 Scaling was manual, slow

Traffic came in bursts. A human had to notice, and a new replica took over six minutes to serve.

→ Planner scales on KV cache

3 Engine choice was a bet

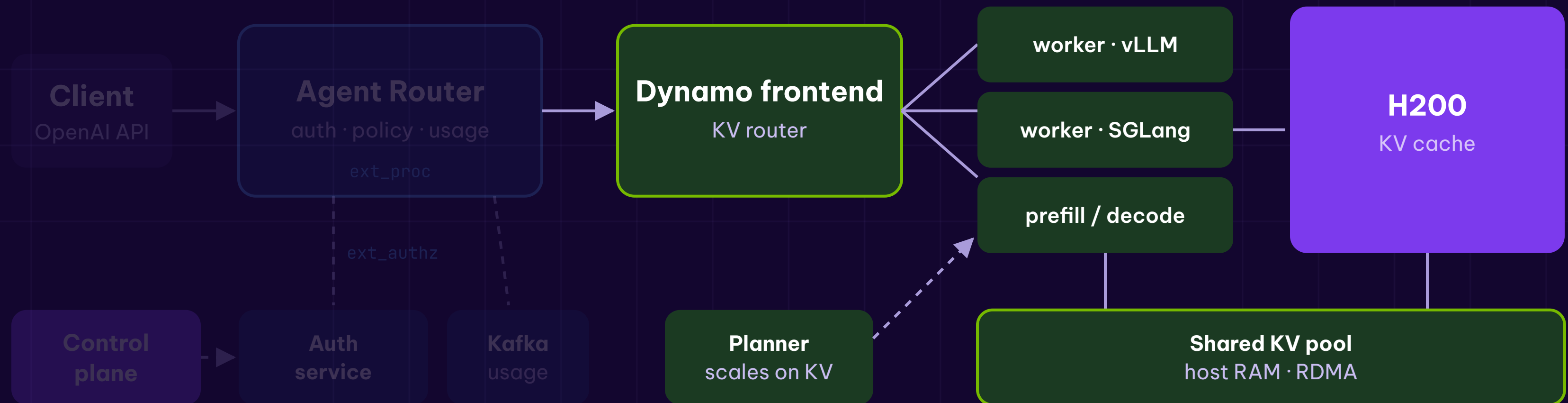
Engines ship faster kernels every week. A choice made at deploy time could be far off, with no cheap way to check.

→ swap the engine, measure again

Same root cause: the fix belongs **above the inference engine.**

The engine runs the model. Dynamo runs the rest.

Dynamo is **not an inference engine**. It sits around vLLM, SGLang or TensorRT-LLM and decides where each request goes, how many replicas run, and where the KV cache lives.



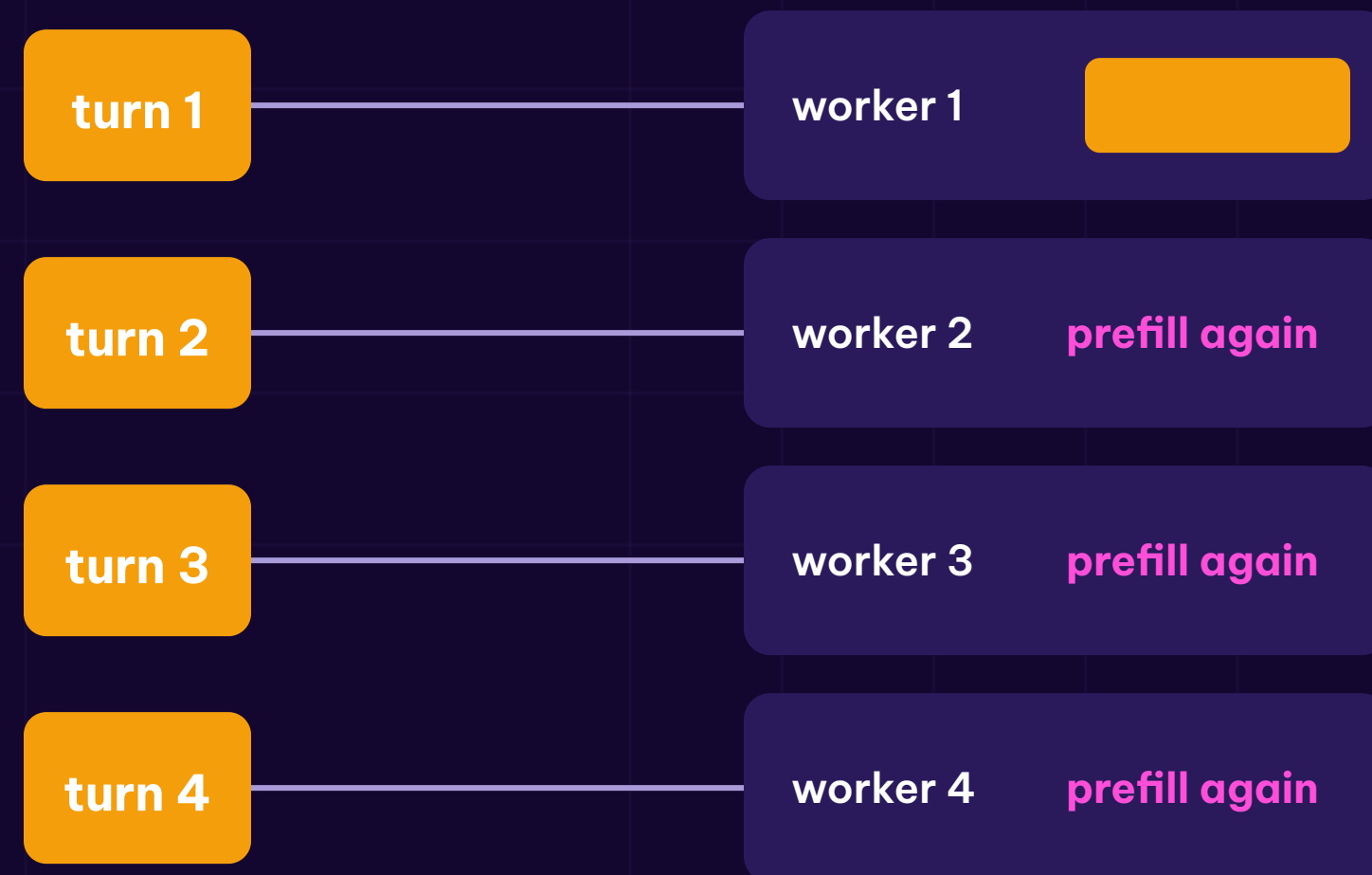
Open source. One `DynamoGraphDeployment`. Each feature is **a component you turn on**.

Dynamo, Grove and KAI Scheduler are part of NVIDIA DSX OS · more at the end

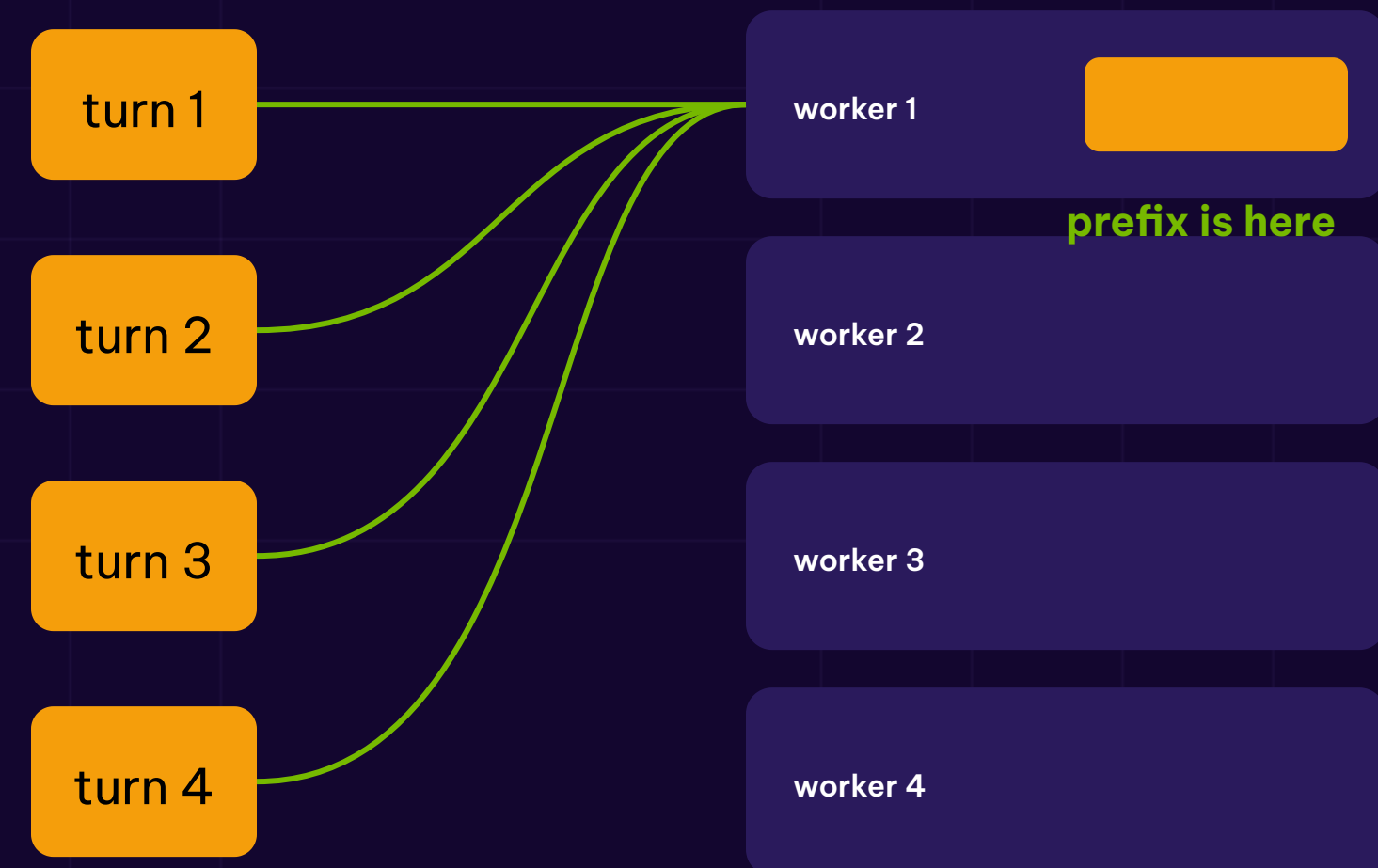
Round-robin wastes the KV cache

KV cache is the work the GPU does to read a prompt. Send the next turn to the worker that already holds it, and that work is skipped.

Round-robin



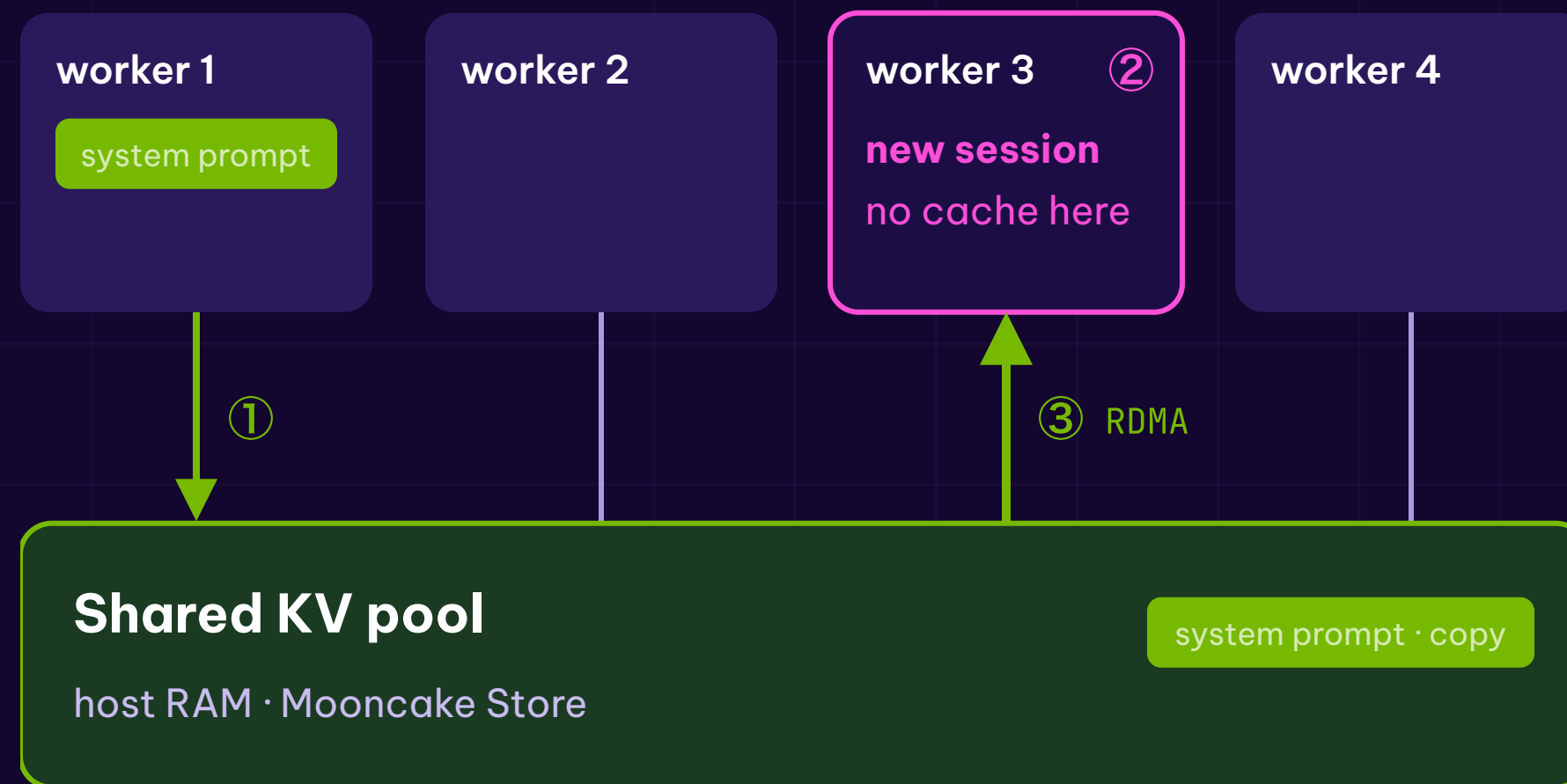
KV-aware routing



Prompts are ~83k tokens, **95%** repeated from earlier turns. Round-robin: **3 of 4** turns go to a worker that never saw the session.

What if the prefix is on another worker?

Workers share KV cache through **a pool in host RAM**. Any worker can fetch a prefix over RDMA instead of computing it again.

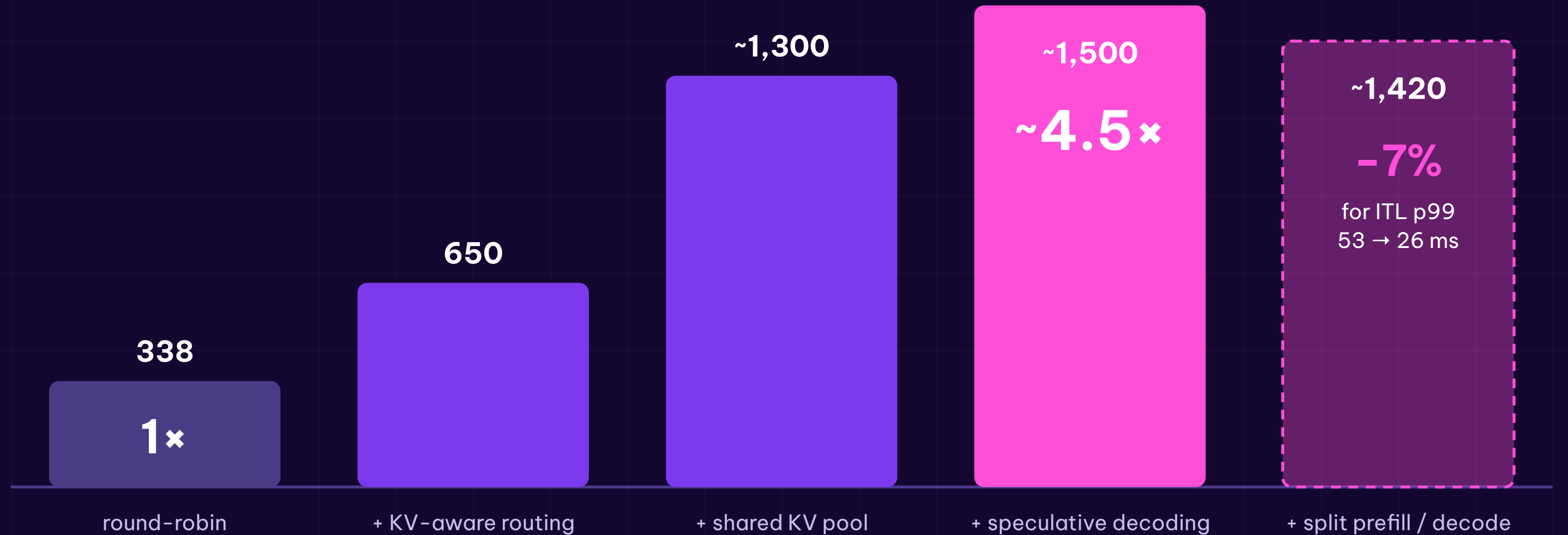


- ① Worker 1 computes the system prompt once and writes its KV to the pool
- ② A new session lands on worker 3: no cache there, routing can't help
- ③ Worker 3 reads the prefix from the pool over RDMA, no recompute

Pool = host RAM from every node,
one store for the cluster

Cache hit: **17%** (round-robin) → **29%** (routing) → **82%** (shared pool)

~4.5× throughput on the same 2 H200 nodes



output tok/s • MiniMax-M3-NVFP4 • 64 session • real traces replayed with AIPerf

Long prompts freeze the stream

Prefill reads the whole prompt and is compute-heavy. **Decode** writes tokens one by one. On the same GPU, one long prefill pauses everyone's stream.

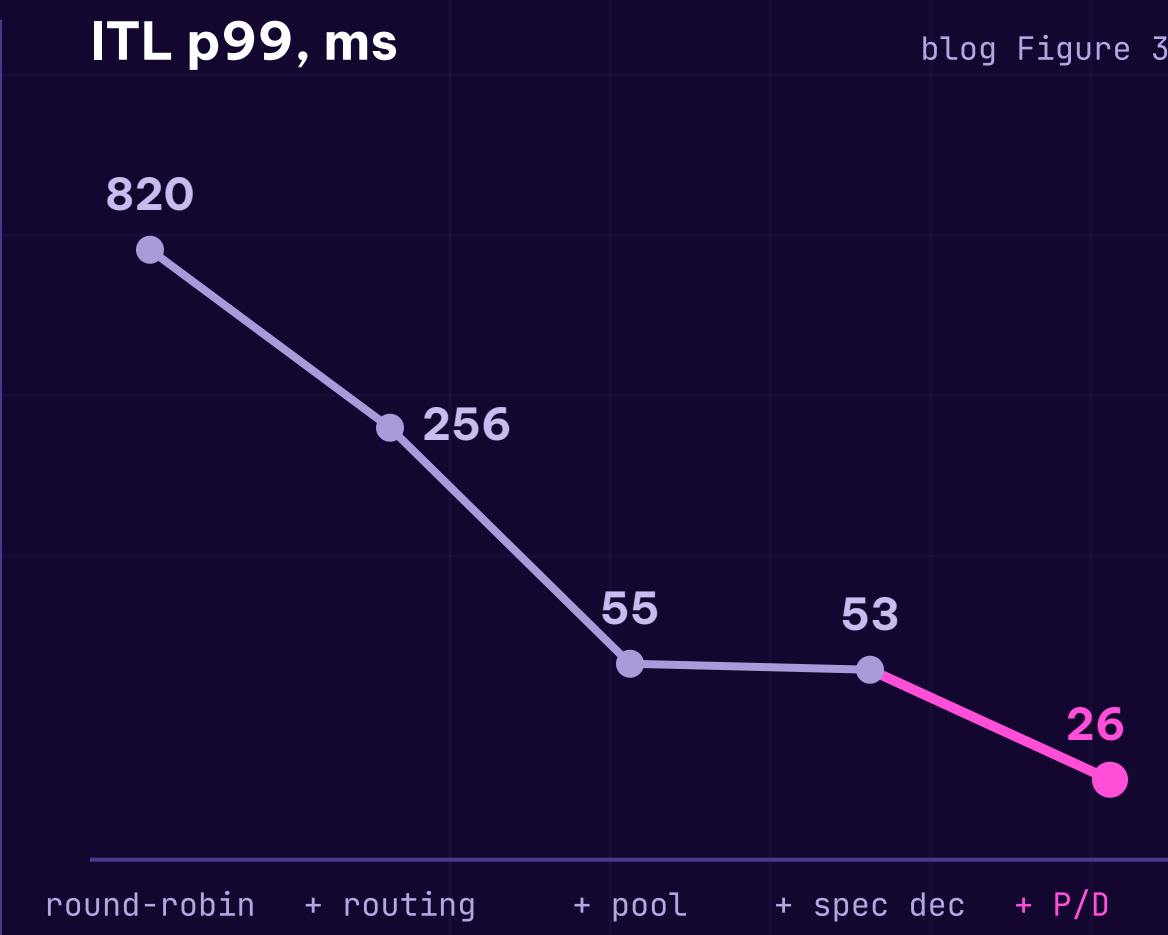
One engine



Split prefill / decode

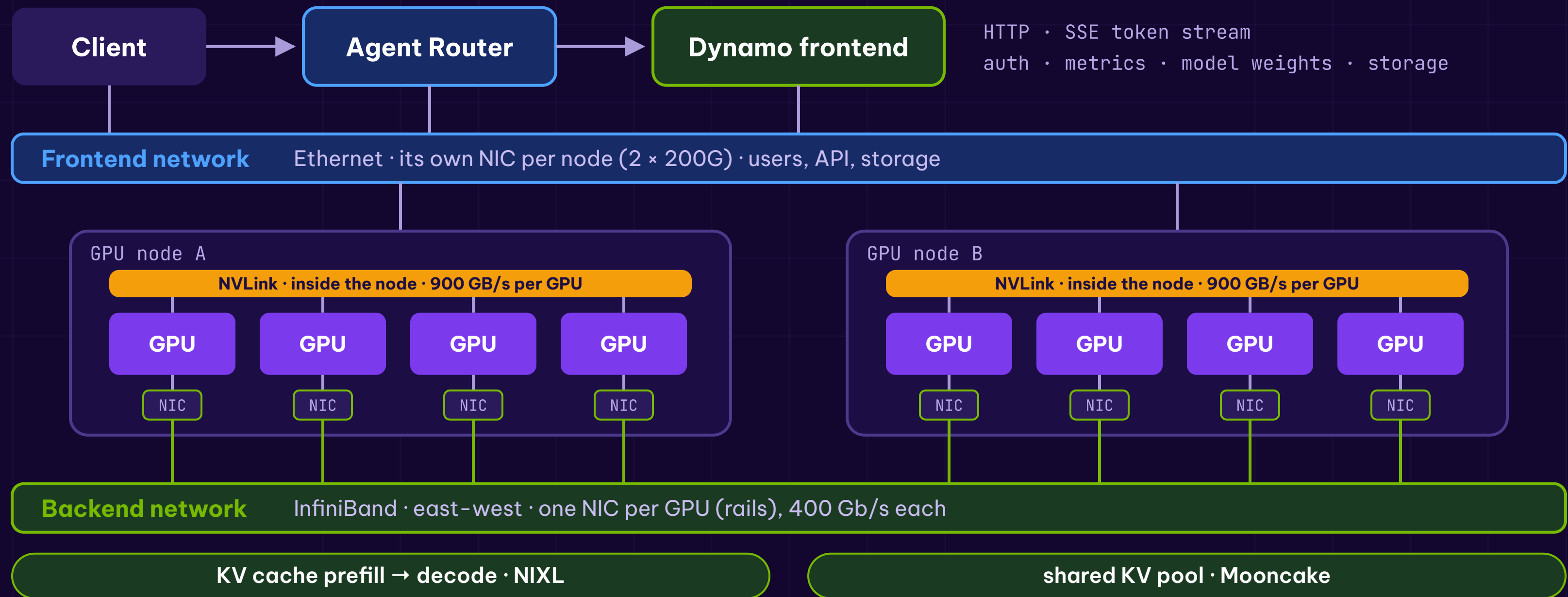


Split P/D: ITL p99 **53** → **26** ms for ~7% throughput. All layers: 820 → 26 ms.



Two networks under one request

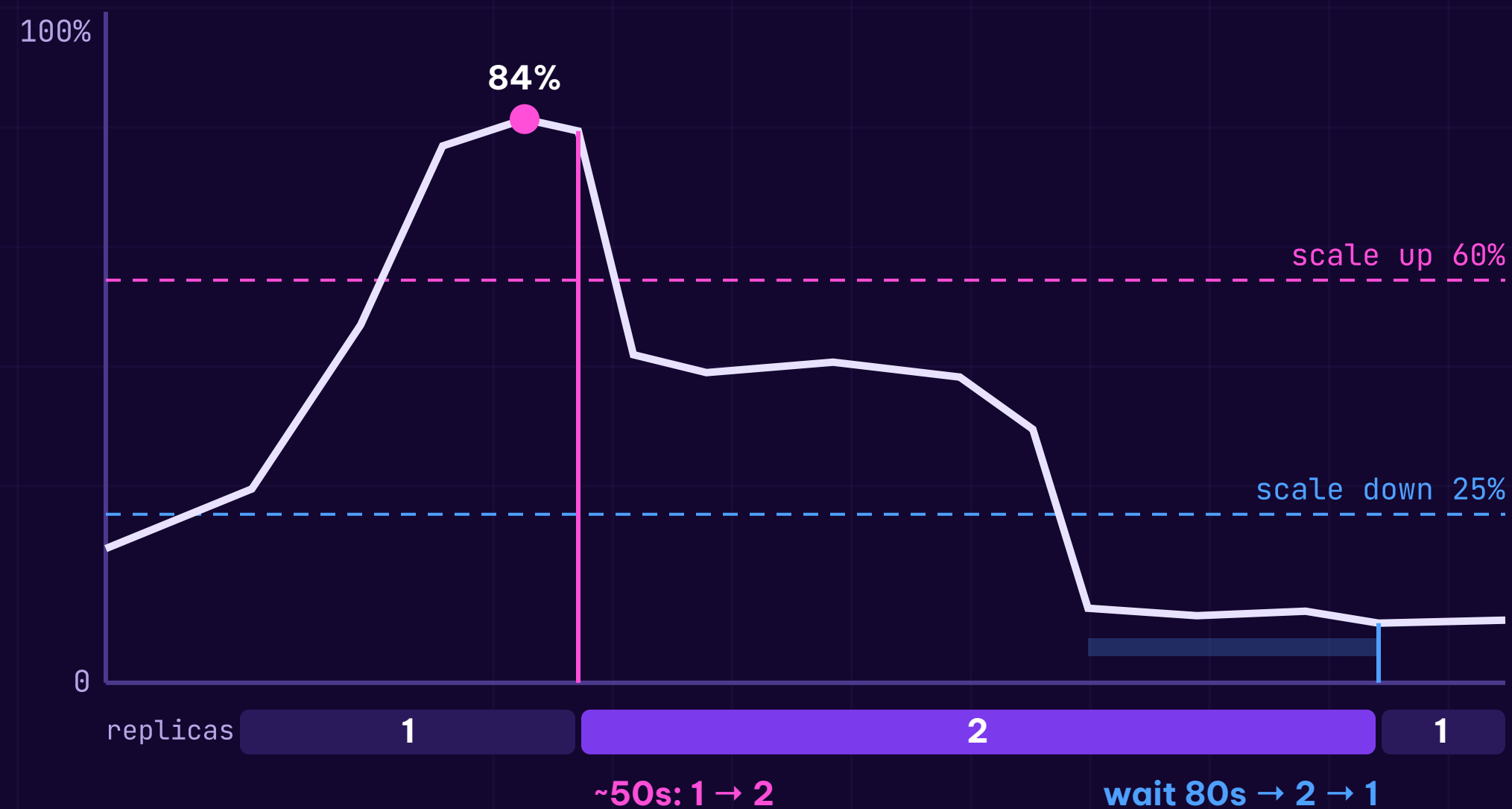
A GPU cluster has more than one network: **Ethernet** for users, APIs and storage, **InfiniBand** for GPU-to-GPU traffic.



Requests and tokens ride the **frontend**. The KV cache and GPU-to-GPU traffic ride the **backend**.

Scale on KV cache, not GPU utilization

The **Planner** adds and removes replicas. It watches how full the KV cache is: a direct sign of how much work each worker is holding.



KV-cache utilization over time (sketch; marked numbers are real)

Any worker over 60%

add one replica

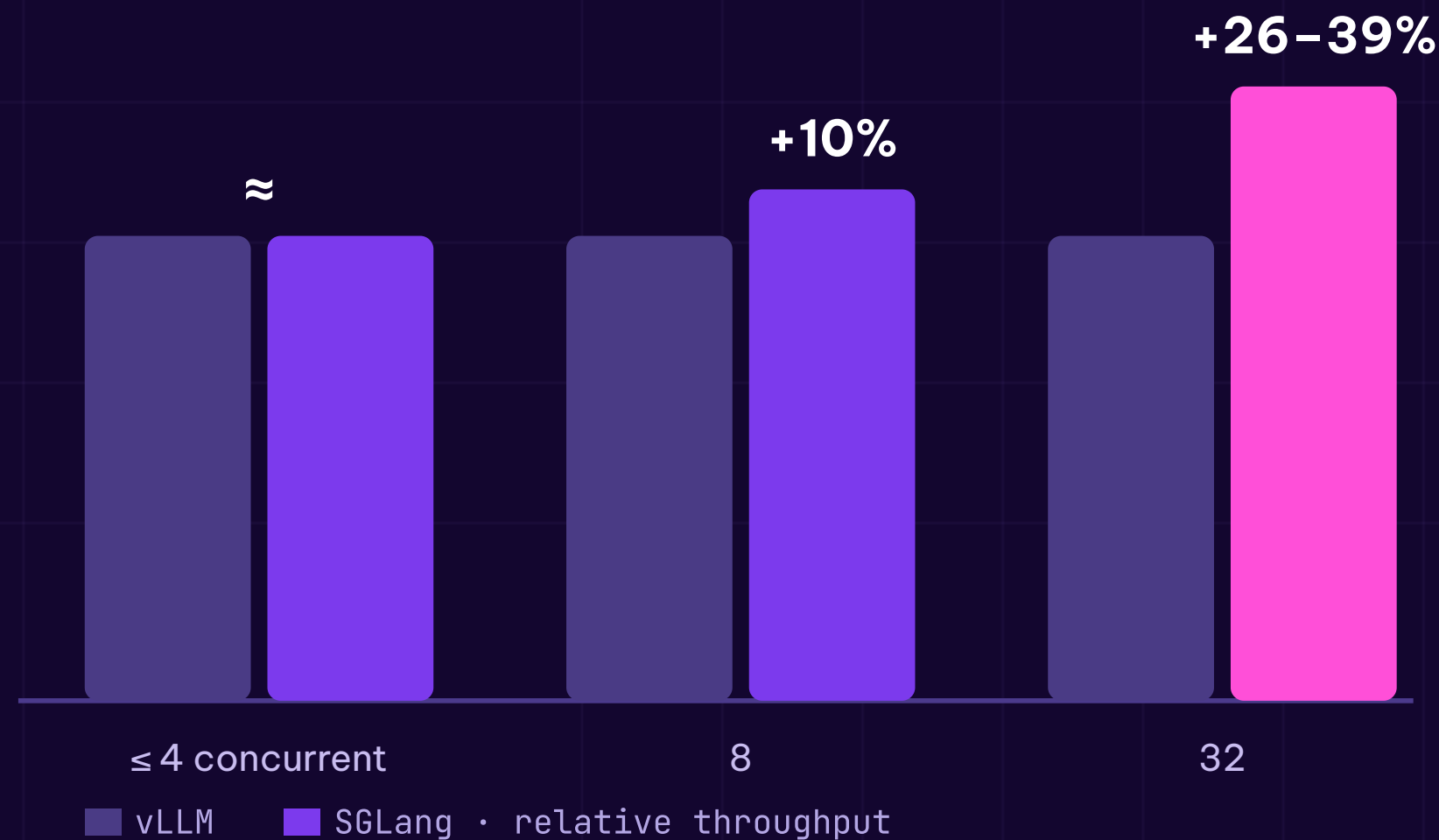
Every worker under 25%

remove one, only when all are quiet

TTFT p95: 44.5 s → 1.4–3.4 s with 2 replicas. No flapping, 0 errors.

vLLM or SGLang? Measure, don't guess.

Dynamo runs the same graph on **either engine**. Swapping is one component in the YAML, so each endpoint can use whichever wins its benchmark.

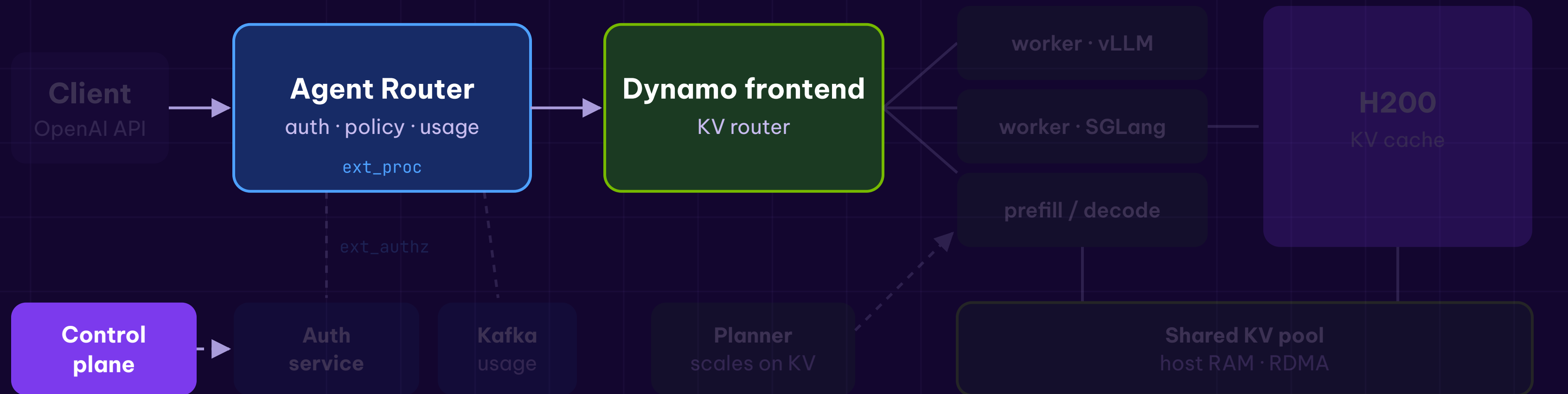


```
backendFramework: sglang
services:
  Frontend: ... # unchanged
  SGLangWorker:
    image: .../sglang-runtime
    command: [... dynamo.sglang]
```

Switching engines = editing **one component**. The winner may change next release, so being able to measure again is what matters.

PUTTING IT TOGETHER

Each layer makes one decision



Control plane

which cluster · which endpoints

Agent Router

who gets in · how much · which model

Dynamo frontend

which worker · how many replicas

Each decision lives in one place, so we change it **without touching the other layers.**

WHAT'S NEXT

Dynamo is one piece of NVIDIA DSX OS

DSX OS is NVIDIA's software layer for AI factories: a catalog of mostly open source components, each installed on its own. We adopt it **one piece at a time**, where we have measured a problem.

PLATFORM SOFTWARE

Scheduling and runtime

KAI Scheduler PR

Workload-Map PR

Workload management

Run:ai PR

Service orchestration

NVCF

Optimized inference

Dynamo

Grove

INFRASTRUCTURE SOFTWARE

Cluster runtime

GPU Operator

Network Operator

Topograph

Cluster orchestration

AICR PR

 Viettel Cloud: a contributing org*

Resiliency and remediation

NVSentinel

NVCRE PR

Monitoring and observability

Fleet Intelligence

Security and confidential computing

Attestation Suite

Provisioning and multi-tenant ops

DPF · gateway on DPU

NICo

Infrastructure management

Switch Config Mgr

RMS

Facilities operations

out of scope

Next: **Karmada + KAI + Workload-Map** to place models across clusters.

in use

trying it

later

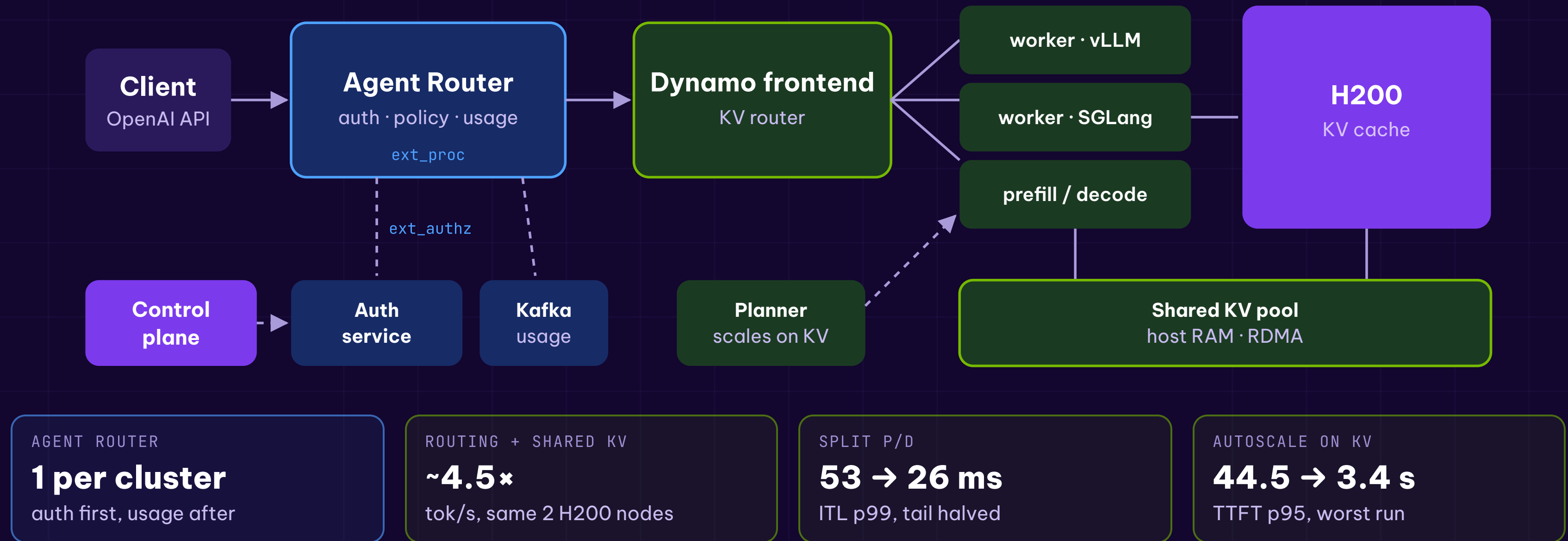
not a fit yet

PR

 merged upstream * in the AICR v1.0 contributors-by-organization list; orgs come from contributors' public GitHub profiles

RECAP

From vLLM serve to a platform



DYNAMO WRITE-UP
AGENT ROUTER

viettelcloud.vn/en/tech-blog/4/142
github.com/theagentrouter/agent-router

LISTED BY NVIDIA
DYNAMO

docs.nvidia.com/dynamo/external-publications
github.com/ai-dynamo/dynamo

Q&A

THANK YOU

Quang Minh Phung

LinkedIn [linkedin.com/in/minhpg27](https://www.linkedin.com/in/minhpg27)

GitHub github.com/pqminh27

Facebook [facebook.com/pqminh27](https://www.facebook.com/pqminh27)

The Anh Nguyen

LinkedIn [linkedin.com/in/ntheanh201](https://www.linkedin.com/in/ntheanh201)

GitHub github.com/ntheanh201

Facebook [facebook.com/ntheanh201](https://www.facebook.com/ntheanh201)

Blog ntheanh201.id.vn

