

Telco AI Grid: Serving, Governing, & Monetizing AI

A Linux Foundation Networking publication

Authors

Ranny Haiby, Ron Haberman, Shujaur Mufti,
Fatih E. Nar, Ravi Sharma, Murat Parlakisik

Table of Contents

1. What an AI Grid actually requires	5
2. Serve: A distributed inference fabric shaped by network physics	6
2.1 Turning AI chaos into Model-as-a-Service (MaaS).....	8
2.2 The sovereignty dividend	8
3. Govern: Determinism that survives regulatory scrutiny	9
3.1 Why this matters more with agents.....	10
3.2 A proposed maturity target: L6*.....	11
4. Monetize: Tokens are the new data buckets	12
4.1 The charging flow, end to end.....	14
4.2 Distribution and the ecosystem: CAPIF and CAMARA.....	16
5. The open source foundation	17
5.1 Essedum: the integration layer.....	17
5.2 OpenAN: telecom-grade agent primitives.....	17
5.3 Horizontal building blocks.....	17
6. The integrated business case.....	18
7. References	20

Introduction

The AI industry is spending billions to build what telecom operators already have: a physically distributed footprint with power, fiber, and skilled operations behind it, stretching from national data centers to the radio edge. Operators reach this footprint through a mix of ownership, leases, and infrastructure partnerships. Not every site is ready for compute; power, cooling, backhaul, space, and access rights decide which locations qualify. But as AI inference moves out of centralized clouds toward users, devices, and machines, this footprint becomes the natural home for a new class of infrastructure. The industry has started calling it an “AI Grid”: a control plane that logically unifies geographically distributed sites into a single, coherent AI fabric, with resource, intent, decision, and policy engines deciding what runs where and how.

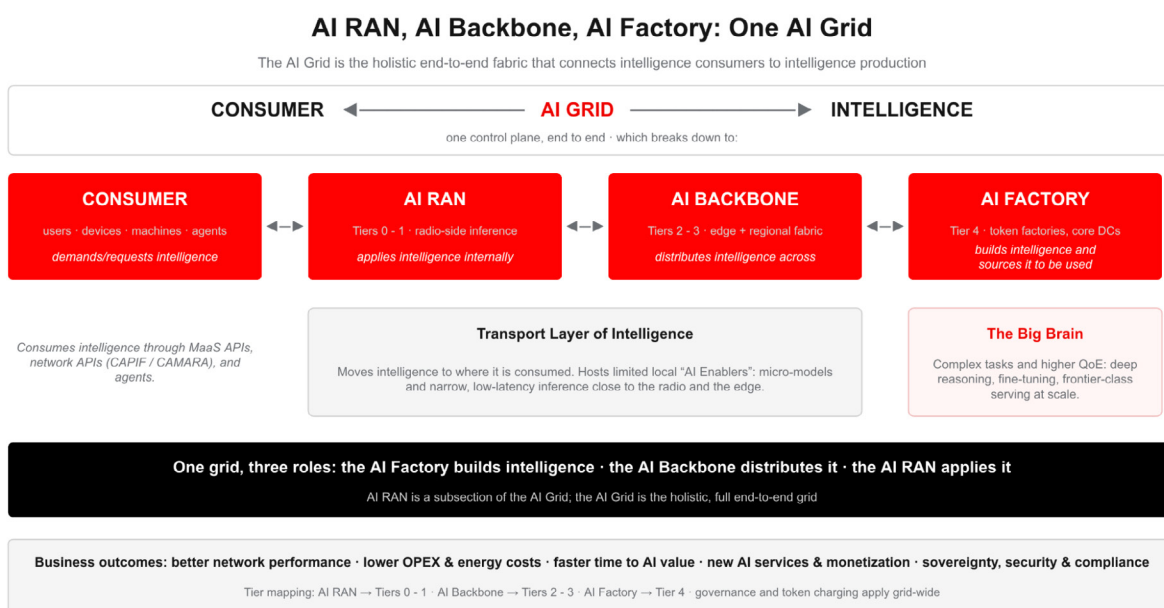


Figure 1. One AI Grid: The AI Factory builds, the AI Backbone distributes, the AI RAN applies intelligence.

The taxonomy that follows is the authors’ own, not yet industry-standard. At the coarsest level, the AI Grid is a single fabric standing between intelligence consumers and intelligence providers. Zoom in one level and it decomposes into three subsystems with three distinct roles:

- **The AI Factory** builds intelligence, where training, fine-tuning, and frontier-class serving happens;
- **The AI Backbone** distributes the intelligence, the edge and regional fabric that moves models and inference to where they are consumed (if possible and if it makes sense), hosting limited local AI enablers along the way; and

- **The AI RAN** applies intelligence within the radio fabric, where true autonomy meets the physical network. It is important to highlight that two paradigms coexist here and this article keeps them distinct:
 - a. AI for RAN; where inference optimizes the radio network itself and which is the operational reality today, and
 - b. AI and RAN & AI on RAN; in the AI-RAN Alliance sense, where RAN-site infrastructure is shared between radio functions and consumer-facing inference as capacity and demand allow.

The AI RAN is a subsection of the grid, not a peer (nor an overlay) to it, and where parts of the industry use “AI grid” narrowly for the distribution layer alone, this article deliberately names that layer the AI Backbone so that the grid can describe the whole.

Zoom in once more and the three roles resolve into the five placement tiers of Figure 2, where workload, network, and business KPIs decide what runs where. Serve, govern, and monetize, the three legs of this article, operate across every level of that hierarchy.

Most of the attention so far has gone to the GPU cluster buildouts. Raw compute at the edge is only one part of the problem, though. For an operator, an AI Grid is only investable if it answers three questions at the same time:

- **Serve.** Can it run the right model at the right place against the right KPI envelope (latency, throughput, accuracy, cost), as one fabric rather than a scatter of GPU islands?
- **Govern.** Can it make AI decisions auditable to the standard regulators expect from critical infrastructure, including when third-party workloads run on the network?
- **Monetize.** Can it charge for AI consumption at the token level through the charging infrastructure the operator already runs, without standing up a parallel billing stack?

This article walks through all three legs with a reference architecture whose AI platform, serving, routing, and gateway layers can be assembled from open-source components (Kubernetes, cloud-native AI serving with vLLM and Llm-d, and the Kuadrant AI gateway), then integrated with the operator’s existing 5G core and charging systems. It then shows how they connect to the 5G core functions and 3GPP interfaces operators already operate ^[3].

TL;DR: The principal components of the AI Grid exist today. Most of the remaining work is integration and productization, with targeted R&D still required in cross-tier routing, governance validation, and charging mediation.

1. What an AI Grid actually requires

Before the architecture, the requirements. Translated into engineering terms, inference at the edge of a telco network demands four capabilities that ordinary Kubernetes-with-GPUs does not provide:

1. **Prompt awareness.** The platform must understand the intent, context, and complexity of an incoming request so it can route to the right model tier. A field technician's yes/no classification should never wake up a 405B model.
2. **Hard multi-tenancy.** Hardware partitioning where supported (such as Multi-Instance GPU, MIG), allocated through dynamic resource allocation ^[15] and combined with software, network, storage, and identity isolation for independent tenants sharing grid capacity.
3. **AI-as-a-Service.** Models exposed as metered, self-service APIs with subscriptions, quotas, and tiers, in a Model-as-a-Service (MaaS) operating model rather than a ticket queue to a data science team.
4. **Token monetization.** The ability to convert computational output into revenue by charging for generated tokens, with real rating, billing, and settlement behind it.

The first two are platform engineering. The last two are where telcos hold an underappreciated structural advantage, and where this article spends most of its time.

2. Serve: A distributed inference fabric shaped by network physics

An AI Grid is a hierarchy of business/workload placement tiers. Network physics constrains what can run where, but placement is decided against layered KPI sets: workload KPIs (time-to-first-token, throughput, accuracy, context depth), network KPIs (bandwidth, jitter, availability, energy), and business KPIs (cost per token, data sovereignty, SLA class). Each tier optimizes a different balance of the three.

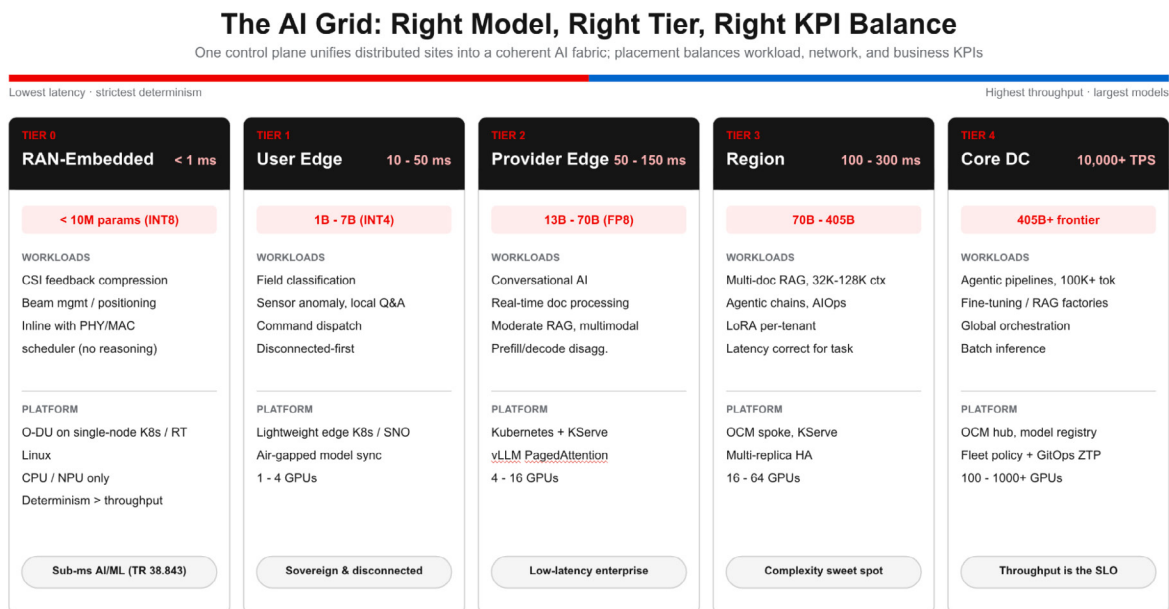


Figure 2. The AI Grid's five placement tiers, radio to core.

The tiers read from the radio inward; latency figures are service-side envelopes under warm-model, controlled-load conditions, and end-to-end results vary with network RTT, prompt length, queuing, cache state, and hardware. In typical telco footprint terms, Tier 0 lives at the cell site, Tier 1 at the user edge, Tier 2 at aggregation sites, Tier 3 in regional data centers, and Tier 4 in central data centers:

- **Tier 0, RAN-embedded.** This is the AI-for-RAN case: the subset of RAN AI functions that execute on the real-time radio path under sub-millisecond or slot-level constraints: CSI feedback compression, beam management, positioning (the 3GPP TR 38.843 family of use cases ^[1]). RAN AI functions with slower control loops belong in Tier 1 or a RIC-adjacent placement. Models here are tiny (under 10M parameters, quantized), run on CPU or NPU inside the slot budget, and are deliberately excluded from generative serving. Determinism is more important than throughput at this tier.

- **Tier 1, User edge.** Where the AI-and-RAN paradigm begins: consumer-facing inference on infrastructure at or near the radio site. The fastest LLM tier: 1B-7B models, INT4-quantized, on single-node or lightweight edge Kubernetes footprints that keep working when disconnected. Narrow-domain work such as field classification, sensor anomaly triage, local Q&A, and command dispatch. Expect ten to fifty milliseconds to the first token on one to four GPUs, and the tier can be configured for site-local processing so sensitive data does not leave the site.
- **Tier 2, Provider edge.** The 13B-70B band for connected devices: conversational AI, real-time document processing, moderate RAG and multimodal workloads at 50-150 ms. This is where prefill/decode disaggregation can start paying for itself on large, concurrent, or long-context workloads: tuned well (KV-transfer overhead and prefill/decode pool ratios both matter), it keeps time-to-first-token consistent under batch load.
- **Tier 3, Region.** The complexity sweet spot: 70B-405B models with 32K-128K contexts for multi-document RAG, agentic chains, and AIOps, with per-tenant LoRA adapters served through a shared fleet. Higher latency here is usually the right trade for task depth, though tier choice should rest on measured accuracy, latency, and cost for the target workload rather than parameter count alone.
- **Tier 4, Core.** Throughput is the SLO: frontier-class serving (405B+), 100K-token contexts, fine-tuning factories, global orchestration agents, and batch inference at fleet-wide throughputs measured in millions of tokens per second, with the fleet's model registry, promotion pipelines, and policy authority anchored here.

The unifying principle is right model, right tier, right KPI balance, with latency as one axis among throughput, accuracy, energy, sovereignty, and cost. The unifying substrate today is fleet management: Kubernetes provides the common lifecycle and fleet substrate across the tiers, from single-node far edge to multi-cluster regional fleets (Tier 0's real-time inference runs in dedicated RAN processes and accelerators, outside the Kubernetes request path), with Open Cluster Management ^[19] coordinating fleet-wide policy, placement, and GitOps zero-touch lifecycle, while model registries and object-storage pipelines distribute model artifacts, trained in the core and delivered to the edge. vLLM ^[14] and Llm-d ^[13] already provide load-aware, prefix-cache-aware, and SLO-oriented routing across model-server endpoints within configured inference pools and clusters. The unresolved AI Grid problem is coordinated routing across independently governed sites, regions, network domains, and failure boundaries, so that a request can cascade from a saturated edge cluster to the next-best one on live inference state rather than DNS-level weighting alone. The [Llm-d community](#) is building toward it, but hasn't shipped yet. Because a common Kubernetes-based operational model can manage traditional IT, 5G network functions, and AI workloads, across separate clusters, node pools, security zones, and failure domains where needed, the operator gets one operational paradigm instead of three.

Tier fitment is a measurement problem before it is an architecture problem. The authors' open-source **TelcoAIBench** suite ^[21] exists for exactly this step: it measures accuracy per telco workload class (3GPP prose, RAN logs, network math), latency and token economics, and deployment footprint (VRAM, precision, serving stack) for any OpenAI-compatible endpoint, and maps each benchmarked model to the closest-to-the-edge tier it measurably fits. Placement decisions argued from parameter count alone do not survive contact with those numbers.

2.1. Turning AI chaos into Model-as-a-Service (MaaS)

Serving infrastructure becomes a business only when an operating model sits on top of it. The pattern we see working is MaaS: a central AI platform team curates a model catalog (each model with declared use cases, cost, and risk profile), serves common models from a shared GPU pool, and exposes them through an API gateway with self-service onboarding.

Developers, internal teams first and enterprise customers next, consume models through the gateway, with quotas, rate limits, and fair sharing enforced by policy instead of by meetings. This is deliberately a dual-constituency model: the operator's own teams (AIOps, network automation, customer care) and external tenants share the same grid, with tenancy isolation keeping them apart and priority policy deciding who gets capacity first. Network-critical internal workloads take precedence; external tenants buy defined service classes below that. The same quota and metering machinery is also the foundation the Monetize section builds on: what starts as internal fair-sharing becomes external billing. The same move that curbed shadow IT curbs shadow AI: make the right path the easy path, and the shared-resource model keeps unit economics sane.

2.2. The sovereignty dividend

There is a strategic overlay to this fabric. Operators already run trusted national-scale infrastructure under strict data-residency obligations, with regulator relationships measured in decades. Combining that trust with localized GPU compute and ultra-low-latency connectivity positions telcos to offer **sovereign AI**: domain models and enterprise pipelines where data stays inside the jurisdiction, or even inside the premises. The technique split matters here: RAG keeps sensitive knowledge in a controlled corpus outside the model, while governed fine-tuning approaches such as InstructLab ^[17] are reserved for information explicitly approved to become part of model behavior. National and sovereign AI services may command a premium where residency, regulatory assurance, and local operational control carry measurable value.

3. Govern: Determinism that survives regulatory scrutiny

Telecommunications networks are critical infrastructure. When an AI system recommends deactivating a cell sector to save energy, or an agent reconfigures a slice, “the model said so” is not an answer a regulator, an enterprise customer, or an incident review board will accept. If an automated decision contributes to an outage or a degraded emergency service, investigators will ask which model acted on what data, within which validated bounds, and whether humans retained an effective ability to override. The grid must be able to answer.

Deterministic governance therefore belongs in the grid architecture itself, with the same standing as serving and routing. To be precise about the claim: the guardrails are deterministic; the language model is not. The architecture provides deterministic policy enforcement and reproducible evidence boundaries around a probabilistic synthesis component. The pattern, developed in depth in recent research on AI-RAN deterministic patterns and practices ^[18], constrains AI to act as a bounded synthesis mechanism inside a traceable decision framework:

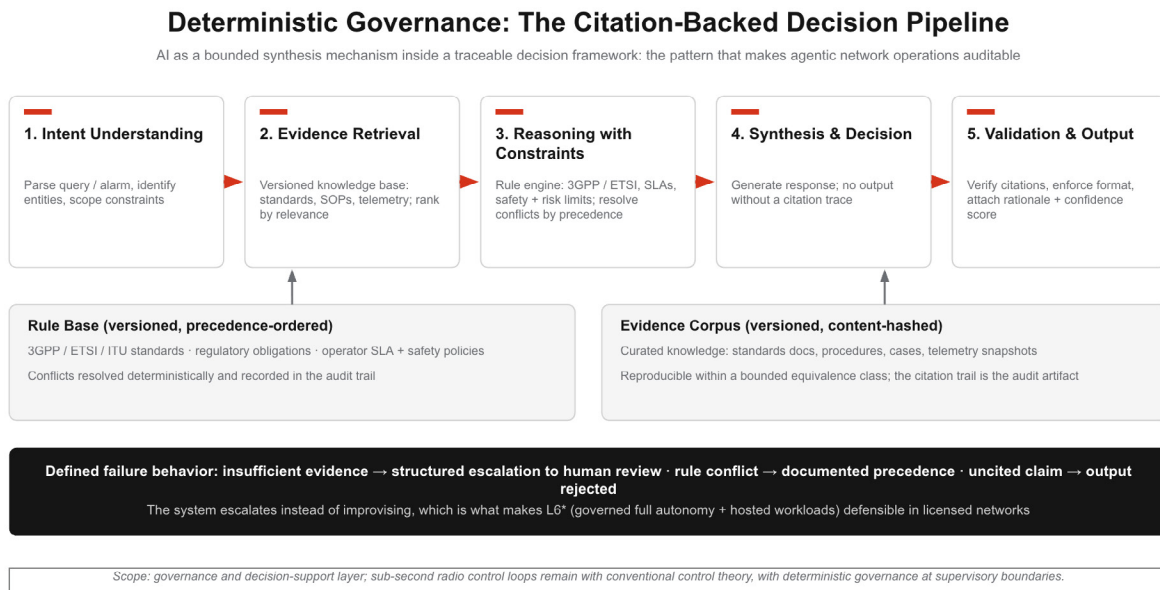


Figure 3. The deterministic governance pipeline: every decision carries a citation trail.

Three components interact.

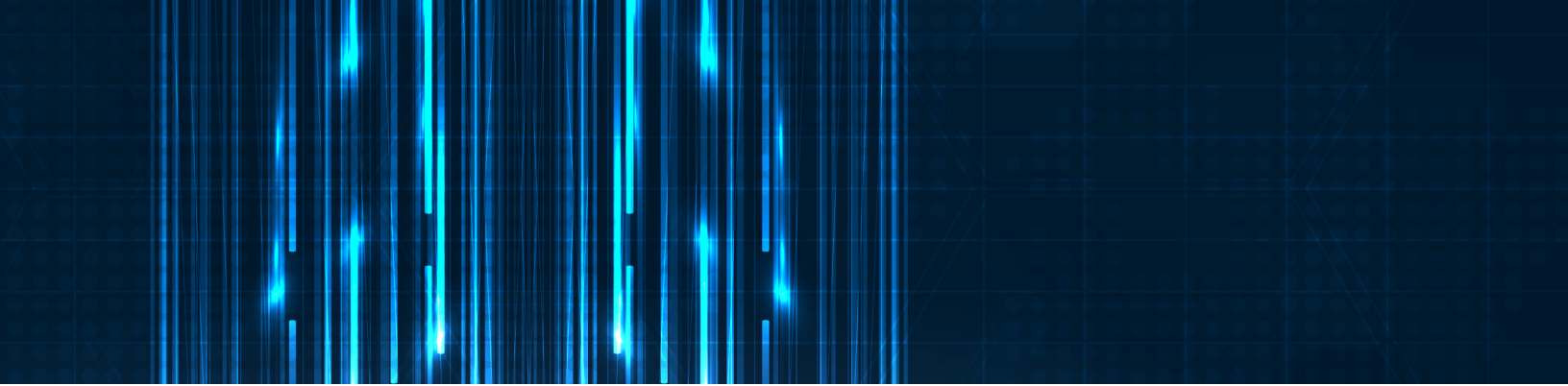
- An **explicit rule base** derived from telecom standards (3GPP, ETSI, ITU), operator SLA and safety policies, and regulatory obligations. It is versioned and carries documented precedence, so rule conflicts resolve deterministically.
- A **retrieval layer** that grounds every AI-assisted output in a curated, versioned, content-hashed evidence corpus.
- And a **synthesis layer** whose outputs must cite both the governing rules and the retrieved evidence. No output without a citation trace. That trace provides auditability; correctness still depends on the validation gate and the escalation paths below.

Just as important is defined failure behavior. When retrieval returns insufficient evidence, the system returns a structured escalation to human review instead of synthesizing an answer. When rules conflict, documented precedence applies and the conflict is recorded. When an output cites material that was never retrieved, it is rejected before delivery. The system escalates instead of improvising.

Concretely: An energy-optimization model recommends sleeping a cell sector during a projected low-traffic window. The pipeline checks emergency-services continuity obligations, minimum coverage thresholds, adjacent-cell handover headroom, and approved change windows against live evidence. If everything clears, the deactivation instruction ships with its citation trail: the traffic forecast, the capacity evidence, the rules satisfied, the projected savings. If an emergency-services session is active on the sector, the system does not act. It returns a structured escalation citing the exact constraint that blocked it, the evidence that triggered it, and the rule-base version applied. Either way, a post-hoc investigator can reconstruct the whole decision.

3.1. Why this matters more with agents

Agentic AI is entering network operations: LLM-powered agents that select tools, plan multi-step actions, and reason iteratively over live telemetry. The architectural shift is the relocation of decision-making from a fixed model output to a runtime composition of model, tools, and plan, and the trust boundary moves with it. The same deterministic constraints (rule-sanctioned action spaces, citation- and policy-backed justification for every consequential decision and action, human override) are what make agentic operations defensible in licensed networks, mitigating recognized failure modes like hallucinated actions and prompt injection. Citation-backed reasoning must also be paired with least-privilege tool access, bounded action spaces, transactional execution with rollback, and risk-based approval gates: a well-cited action can still be operationally unsafe.



3.2. A proposed maturity target: L6*

TM Forum's autonomous-networks scale runs from L0 through L5 ^[11]. As the authors' proposed extension beyond L5, and not a TM Forum designation, we define a target state we call **L6***. It pairs **L6**, governed full autonomy, meaning intent-driven self-optimization strictly bound by deterministic, citation-backed governance for every customer-facing or regulated decision, with the **Hosted-Workload Profile** (also the authors' term): multi-tenant workload isolation at the infrastructure layer, contractual and verifiable SLAs at the interface layer, and tenant-aware auditable governance at the decision layer. The two are orthogonal, and that is the point. A network can be highly autonomous and still be unfit to host third-party AI workloads. Full autonomy and external hosting cannot coexist responsibly unless deterministic governance binds them. L6* is the AI-native network, and it is the maturity level at which "Networks for AI" becomes a product operators can actually sell.

Operators do not need to start at L6*. Building traceable model governance, structured decision logging, and risk-tiered approval processes at L2-L3 today creates exactly the operational maturity that will align with governance expectations as they harden across jurisdictions.

4. Monetize: Tokens are the new data buckets

Here is the part we believe deserves far more attention than it gets. Every business that sells AI inference needs metering, quotas, rating, billing, and settlement. AI providers are building these systems from scratch for inference. Telcos already operate mature charging, policy, account, and settlement infrastructure that can be extended to this workload, hardened over decades and integrated with every subscriber and enterprise relationship they have.

Two problems sit at the AI + 5G intersection, and both keep AI revenue off the operator books:

- **Policy translation.** How do API-level governance policies (authentication, rate limiting, routing) translate into 5G network policies (QoS, session management, slice admission) so AI inference traffic gets the right network treatment?
- **Token monetization.** How do telcos charge for AI inference at the token level using existing 5G charging infrastructure (CHF/CCS) instead of building parallel billing systems?

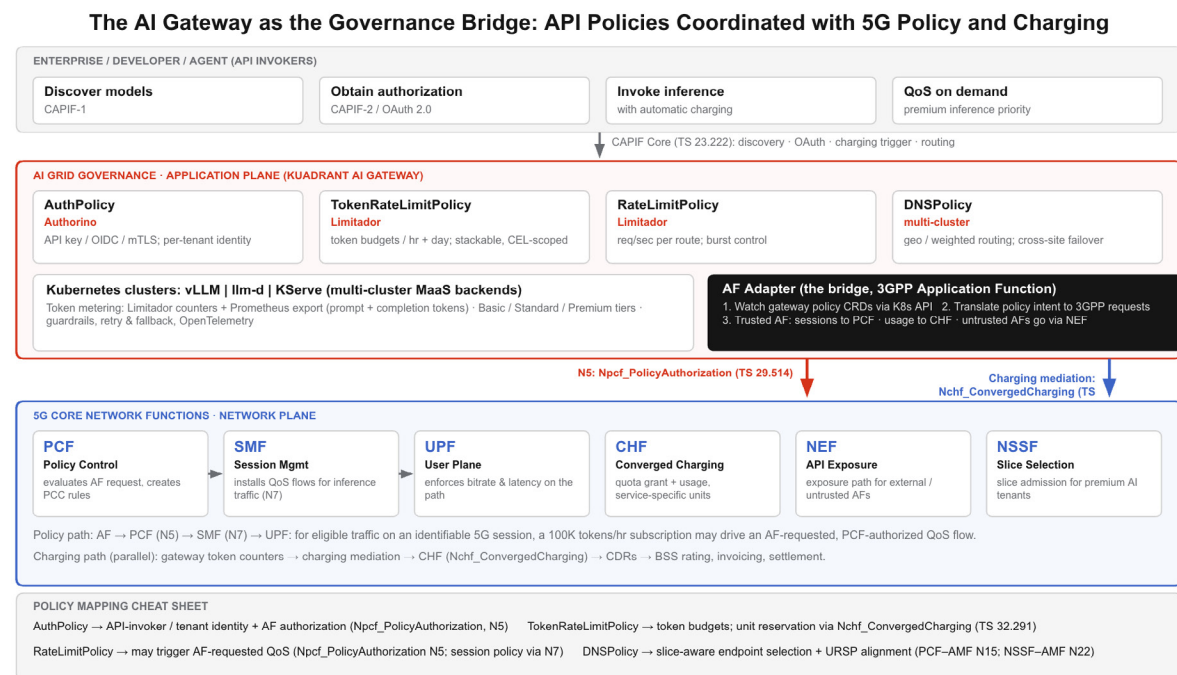


Figure 4. The AI gateway bridges application-plane policy to 5G policy and charging.

The key insight for the second problem: **5G converged charging is unit-agnostic by design** [8]. The CHF, consumed through the Nchf services defined in 3GPP TS 32.290 [9] and TS 32.291 [10], rates time, volume, events, and service-specific units. AI token usage could be represented as operator-defined service-specific units (call them AI service units, with text tokens as the first unit type) under rating groups per model and tier, extensible to input, cached-input, output, and multimodal units. That framing also absorbs the fact that a token is not a standardized unit across models. Existing CHF and BSS capabilities provide the foundation for rating, quota control, charging records, and settlement, with AI-specific product configuration and mediation added through the gateway adapter. Prepaid and postpaid AI bundles can use a commercial model familiar from data bundles: Basic, Standard, and Premium MaaS tiers alongside data plans. The remaining build item is the bridge between the gateway and the charging system.

In this reference architecture, a Kubernetes-native AI gateway built on the open-source Kuadrant project [12] sits in front of the model-serving fleet: authentication via Authorino, token-aware rate limiting via Limitador, multi-cluster routing and failover via DNSPolicy, all expressed as Kubernetes-native policy custom resources. Those constructs align with 5G core concepts at a high level, as analogies to coordinate rather than one-to-one equivalences:

Gateway policy (Kuadrant)	Function	Potential 5G coordination	3GPP interface
AuthPolicy	API key / OIDC / mTLS	API-invoker / tenant identity + AF authorization; subscriber and PDU-session context where applicable	Npcf_PolicyAuthorization (TS 29.514 [6], N5)
TokenRateLimitPolicy	Per-tenant token budget (tokens/hr)	Spending-limit-style budget policy; unit reservation and reporting via converged charging	Nchf_ConvergedCharging (TS 32.291); cf. Nchf_SpendingLimitControl [7] (N28)
RateLimitPolicy	Requests/sec per route	May trigger or constrain AF-requested QoS for identified inference flows	Npcf_PolicyAuthorization (N5); session policy via Npcf_SMPolicyControl (TS 29.512 [5], N7)
DNSPolicy	Geo/weighted application-backend selection	Slice-aware endpoint selection + URSP alignment	PCF-AMF (N15); NSSF-AMF (N22)

TokenRateLimitPolicy is the closest conceptual match: both represent usage-budget state at the policy layer. Spending Limit Control exposes policy-counter status to the PCF; enforcement is applied by the relevant gateway or network-policy function. Actual token-unit reservation, usage reporting, and reconciliation would run over Nchf_ConvergedCharging, which already supports granted and used service-specific units. The AF adapter makes the bridge concrete: it watches gateway policy CRDs through the Kubernetes API, translates policy intent into 3GPP request formats, posts Npcf_PolicyAuthorization sessions to the PCF, and reports token usage to the CHF. Because an external gateway is not a trusted 5G core function by default, this adapter is where operator control lives: mapping API invokers to subscriber accounts, assigning rating groups, guarding against replay and duplicates, and reconciling model usage against charged usage. The adapter can interact directly with the PCF when deployed as a trusted operator AF; an external or untrusted AF would normally reach these capabilities through the NEF. The PCF evaluates, creates PCC rules ^[4], the SMF installs QoS flows, the UPF enforces them. The result: for inference traffic carried over an identifiable 5G session, an enterprise that buys a 100K-tokens-per-hour MaaS subscription can also receive an AF-requested QoS flow, subject to PCF authorization and operator policy. The commercial treatment and the network treatment then derive from a common source of policy intent, which addresses the policy-translation problem once identity, session, and flow correlation are in place.

4.1. The charging flow, end to end

In the proposed architecture, which is not a plug-and-play standard integration, an operator-controlled charging adapter at the AI gateway performs a **charging trigger function (CTF)**-like role for inference consumption, mediating token usage into Nchf_ConvergedCharging:

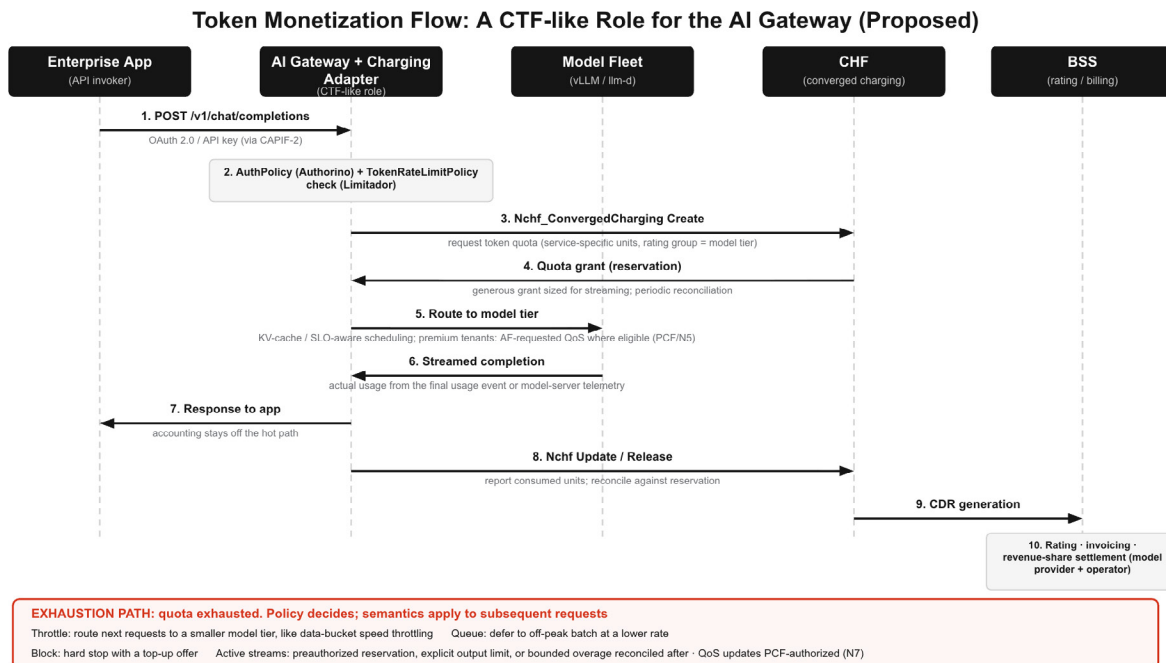


Figure 5. Token charging flow: quota, metering, and CDRs through the CHF.

Walking the sequence: the enterprise app calls the OpenAI-compatible endpoint with credentials obtained through CAPIF onboarding (1). The gateway authenticates and checks the token budget (2), then requests a token quota from the CHF as service-specific units under the tenant's rating group (3). The CHF grants a reservation (4), sized generously because token consumption is bursty and mid-stream, with periodic reconciliation preferred over tight synchronous grants. The gateway routes to the right model tier (5); actual prompt and completion usage comes from the final OpenAI-compatible usage event or from trusted model-server telemetry (6-7); the gateway then reports consumed units to the CHF and reconciles them against the reservation (8). CDR generation (9) and rating, invoicing, and revenue-share settlement (10) ride the operator's existing BSS.

The exhaustion path needs explicit semantics, and they apply to subsequent requests rather than mid-generation model swaps. When a bucket runs out, the operator chooses per tier: **throttle** (route the next requests to a smaller model tier, the AI analog of data-bucket speed throttling), **queue** (defer to an off-peak batch window at a lower rate), or **block** (hard stop with a top-up offer, like a data add-on). An active stream is handled by the preauthorized reservation, an explicit output limit, or bounded overage reconciled afterward. For eligible traffic on an identifiable 5G session, the AF adapter may request updated QoS treatment in parallel, subject to PCF authorization and operator policy.

Two engineering items remain open. Quota-grant latency must accommodate streaming inference, hence reservations plus reconciliation. And rating prompt and completion tokens differently, as many AI providers price them, would require an extension beyond the current total-token accounting path at the gateway.

4.2. Distribution and the ecosystem: CAPIF and CAMARA

Metering without distribution does not make a business. Publishing MaaS model endpoints through **CAPIF** (the 3GPP Common API Framework, TS 23.222 ^[2]) places AI inference within the same service-discovery, security and authorization, invocation-logging, charging-support, and optional dynamic-routing framework that operators are already deploying for network APIs. Aligning with **CAMARA** and GSMA Open Gateway makes the commercial motion familiar. CAMARA defines concrete API products ^[16], such as Quality on Demand and Carrier Billing; the patterns below are the authors' proposed commercial models, designed to be compatible with CAPIF, CAMARA, and operator BSS:

Pattern (CAMARA-aligned)	AI Grid equivalent	How it works
Transactional (per API call)	Per-inference billing	Each completion is a charging event
Usage-based (data volume)	Per-token billing	ServiceSpecificUnits in the CHF
Tiered bundles	MaaS subscriptions	Basic / Standard / Premium
QoS on demand (QoD)	Inference priority	Eligible premium traffic may receive AF-requested QoS flows
Revenue sharing	Provider + operator split	Per-model CDRs enable settlement

That last row matters more than it looks. Per-model usage records give revenue sharing a practical accounting basis; contract, dispute, reconciliation, tax, and fraud processes still have to operate around them. Settlement between parties who do not fully trust each other is a discipline telcos have practiced for decades in roaming and interconnect.

Governance hierarchy completes the operating model. Using Gateway API policy attachment semantics (defaults and overrides), the operator sets platform-wide ceilings on the Gateway ("never more than 200K tokens/hour, regardless of what a team requests") while enterprise tenants self-service their own limits on their routes within those guardrails. Operator sovereignty and tenant autonomy come from the same policy machinery.

5. The open source foundation

Everything above assumes an implementation substrate, and for regulated infrastructure that substrate has to be inspectable. The reference architecture already leans on open source for the Serve and Monetize legs: Kubernetes, vLLM, Llm-d, and Kuadrant. The Govern leg raises the bar further, because auditability claims are only as strong as the ability to read the code that enforces them. Recent LF AI & Data research on sovereign AI found that 81% of surveyed organizations view open source as the primary enabler of sovereign AI ^[20], which is the same logic as the sovereignty dividend argued earlier. Beyond the components already named, two Linux Foundation Networking projects and two horizontal efforts are directly relevant to gaps this article leaves open.

5.1. Essedum: the integration layer

Essedum, an LF Networking project, addresses the integration plumbing that stalls telecom AI between proof of concept and production ^[22]. It is a microservices framework with multi-cloud execution that integrates Langflow as its agent designer, AI Gateway for centralized model access and routing, and Langfuse for observability, with Model Context Protocol support built in; its roadmap adds the Salus Responsible AI framework for decision auditing. For the grid, Essedum is a candidate workflow layer coordinating the model, agent, and application tiers in tasks such as 5G slice feasibility checks, which places it squarely in the AI-for-networks lane the Serve section describes.

5.2. OpenAN: telecom-grade agent primitives

OpenAN, an LF Networking candidate project contributed by China Mobile and Huawei, targets TM Forum Autonomous Networks Level 4 with components built for network operations: a Registry Center, an Orchestration Center, and an SDK implementing the Agent-to-Agent Protocol for Telecom, A2A-T ^[23]. A2A-T extends general agent protocols with structured task schemas, network-event subscription, agent-to-agent negotiation, and an authorization model that requires explicit approval before any network-changing operation. That last primitive is the same instinct as the deterministic governance pattern in the Govern section: bounded action spaces and approval gates rather than free-running agents. OpenAN's Level 4 target and the L6* target proposed above sit on one trajectory; L6* adds the hosted-workload isolation and citation-backed decision discipline on top.

5.3 Horizontal building blocks

Two efforts outside LF Networking matter to the serving and governing layers. The Agentic AI Foundation stewards the Model Context Protocol (MCP), originally open sourced by Anthropic, as the standard interface between models and external tools; Essedum integrates it natively, which gives the grid a common contract between inference layers and network control layers. Under LF AI & Data, OPEA standardizes modular retrieval-augmented generation pipelines, a direct fit for the sovereign on-premises corpus pattern described in the sovereignty dividend, and KServe offers an alternative model-serving control plane for operators standardized on it, complementing the vLLM and Llm-d path this article uses as its reference.

6. The integrated business case

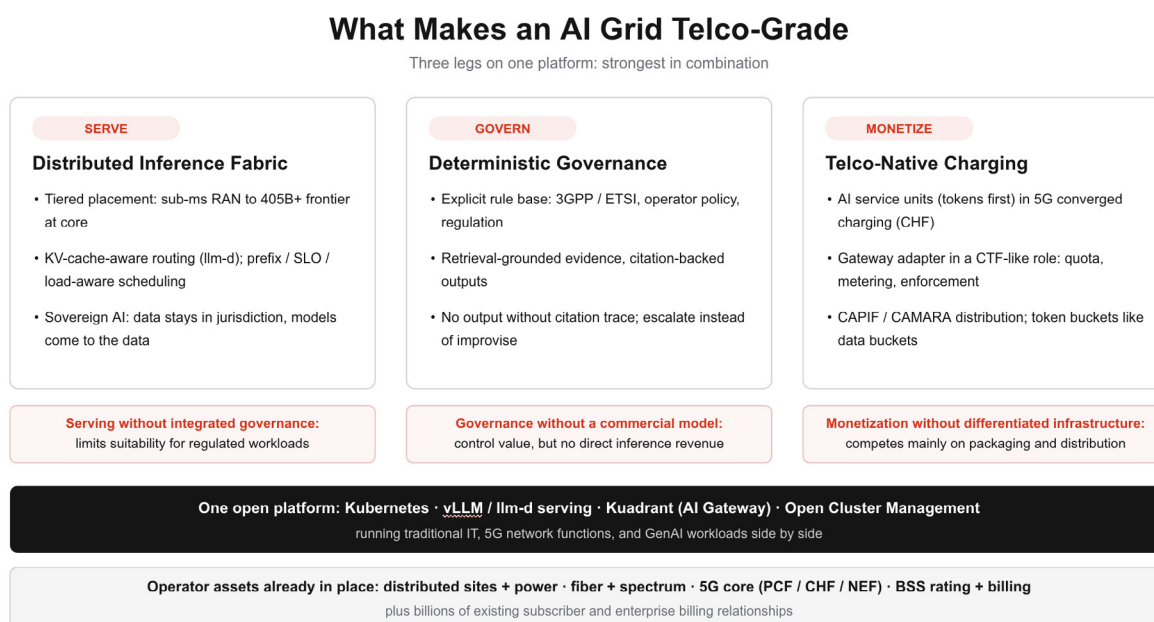


Figure 6. Serve, Govern, and Monetize on one platform.

Each leg can create value on its own, but the combination is what gives an operator a differentiated proposition. A GPU footprint without governance cannot host regulated workloads. Governance without monetization is a well-audited cost center. Monetization without a differentiated fabric leaves the operator competing on aggregation and packaging rather than infrastructure advantage. Together they form a differentiated position in regulated, latency-sensitive deployments: an AI platform with defined performance envelopes at every tier, real-time RAN execution and throughput-oriented core serving included, auditability to the standard demanded of critical infrastructure, and a charging and distribution ecosystem with billions of live billing relationships behind it.

It is worth being honest about what remains open, because the open items shape the roadmap:

- **Token units vary across models.** Per-model rating groups handle this commercially today; if token-denominated charging takes off, it becomes a conversation for 3GPP SA5.
- **Enterprise demand for edge inference is still forming.** Which is precisely the argument for building on infrastructure and charging systems you already own, with open standards (Gateway API, CAPIF, CAMARA) instead of parallel proprietary stacks. The new capital at risk is the integration work; the underlying assets are already on the books.

- **The governance tooling does not yet exist as a packaged product.** Versioned rule engines, content-hashed evidence stores, and synthesis layers that validate citations exist as components. Assembling them into telco-grade platform tooling is exactly the productization work the ecosystem, ourselves included, is doing now.
- **Energy claims need discipline.** AI that optimizes the network while quietly consuming more power than it saves is a net loss. Measurement boundaries must include the AI infrastructure itself.
- **Cross-tier, cross-site inference-aware routing doesn't exist as a product yet.** DNS-level, fleet-management, and in-pool inference-aware routing mechanisms are available at their respective scopes. Deciding, in real time and based on live inference state rather than static weighting, whether a request should cascade to the next tier or the next site remains open: an integrated implementation spanning independently governed sites and network domains has not shipped yet, and the llm-d community is building toward it.

The telco industry spent a decade learning that it could not out-cloud the cloud. The AI Grid is a different game. It plays to what is native to operators (distribution, determinism, governance, and charging) and turns assets already on the books into what the AI market lacks most: low-latency capacity, auditable trust, and a route to revenue. The operators who connect those assets will find they have been building this market's foundation all along.

7. References

- [1] 3GPP TR 38.843 (Rel-18): Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR air interface.
- [2] 3GPP TS 23.222 (Rel-18): Functional architecture and information flows to support Common API Framework for 3GPP Northbound APIs (CAPIF); Stage 2.
- [3] 3GPP TS 23.501 (Rel-18): System architecture for the 5G System (5GS); Stage 2.
- [4] 3GPP TS 23.503 (Rel-18): Policy and charging control framework for the 5G System (5GS); Stage 2.
- [5] 3GPP TS 29.512 (Rel-18): 5G System; Session Management Policy Control Service (Npcf_SMPolicyControl); Stage 3.
- [6] 3GPP TS 29.514 (Rel-18): 5G System; Policy Authorization Service (Npcf_PolicyAuthorization); Stage 3.
- [7] 3GPP TS 29.594 (Rel-18): 5G System; Spending Limit Control Service (Nchf_SpendingLimitControl); Stage 3.
- [8] 3GPP TS 32.240 (Rel-18): Charging management; Charging architecture and principles.
- [9] 3GPP TS 32.290 (Rel-18): Charging management; 5G system; Services, operations and procedures of charging using Service Based Interface (SBI).
- [10] 3GPP TS 32.291 (Rel-18): Charging management; 5G system; Charging service (Nchf_ConvergedCharging); Stage 3.
- [11] TM Forum IG1252: Autonomous Networks Levels Evaluation Methodology (defines levels L0-L5).
- [12] Kuadrant Operator documentation: TokenRateLimitPolicy and Token-Based Rate Limiting (for AI Workloads), release 1.3.x documentation, docs.kuadrant.io
- [13] llm-d project: Architecture documentation, llm-d.ai/docs/architecture
- [14] vLLM documentation: Disaggregated Prefilling (experimental), docs.vllm.ai
- [15] Kubernetes documentation: Dynamic Resource Allocation, kubernetes.io/docs
- [16] CAMARA project / GSMA Open Gateway: Quality on Demand and Carrier Billing API definitions, camaraproject.org
- [17] InstructLab documentation: Knowledge Overview, docs.instructlab.ai
- [18] Authors' prior research on deterministic AI-RAN patterns and practices, EnterpriseAI series on Medium, medium.com/enterpriseai.
- [19] Open Cluster Management documentation: multicluster registration, placement, policy, and application lifecycle, open-cluster-management.io.
- [20] LF AI & Data Foundation: Global sovereign AI research report (2025), finding 81% of organizations view open source as the primary enabler of sovereign AI, lfaidata.foundation.
- [21] TelcoAIBench: open, self-contained benchmark suite and portal for telco AI model evaluation and AI Grid tier fitment, open-experiments.github.io/telcoaibench.
- [22] LF Networking: The Future of Telecom AI: Unlocking Value and Escaping the Plumbing Trap with Essedum, lfnetworking.org, 2026.
- [23] LF Networking: LF Networking Brings A2A-T to Live Network through OpenAN for Agent-Driven Autonomous Networks, lfnetworking.org, June 2026.