Cubie FAQ

High-Level Customer FAQ for Data Center Owners and Stakeholders

Pre-GPU trust gate | Capacity recovery | Cybersecurity | Compliance evidence

Plain-English positioning

Cubie checks whether an AI workload should be allowed to spend GPU capacity before it reaches the GPU path. It does not try to answer the prompt. It underwrites whether the request deserves accelerator spend.

Customer-safe one-liner

Cubie helps data centers reclaim paid-for inference capacity by stopping invalid, stale, spoofed, out-of-policy, or capacity-destructive workloads before they consume VRAM, KV cache, queue time, power, or tenant SLA headroom.

1. Executive FAQ

For CEOs, GMs, board members, and commercial owners.

1. What is Cubie?

Cubie is a pre-GPU trust gate for AI workloads. It evaluates whether a workload should be admitted before the request consumes GPU memory, queue time, power, KV cache, or tenant SLA headroom.

The easiest analogy is credit underwriting. Before a bank funds a loan, it checks identity, risk, eligibility, and repayment capacity. Before a GPU “funds” an inference request, Cubie checks whether the workload is valid, authorized, fresh, policy-compliant, and safe to admit.

2. What problem does Cubie solve for a data center?

Many AI data centers measure GPU utilization, but utilization alone does not prove monetization. A fleet can look busy while bad workload flow consumes capacity without producing completed tenant output.

Cubie focuses on token efficiency and inference goodput: how much paid-for GPU cycle time turns into completed, billable workload.

- It reduces work that should not have reached the GPU path.
- It helps protect tenant throughput from malformed, stale, replayed, unauthorized, or capacity-destructive requests.
- It creates evidence that a workload was checked before execution.

3. Why does this matter now?

AI data centers are increasingly constrained by power, cooling, rack density, GPU availability, debt service, and tenant economics. When adding power or hardware is slow, the practical growth lever is getting more completed workload from the same fleet.

Cubie is positioned as a brownfield optimization layer: improve token efficiency without requiring a GPU fleet replacement or a tenant-stack rewrite.

4. What is the main business outcome?

The target outcome is reclaimed paid-for inference capacity: more completed tenant workload per GPU-hour or megawatt from infrastructure the operator already owns and funds.

- For the CEO: more sellable capacity from the same asset base.
- For the CFO: less stranded capacity against the same power, depreciation, and debt burden.
- For the CTO: an earlier control point before runtime and model execution.
- For the CISO: stronger pre-execution enforcement and audit evidence.

5. Who is the best-fit customer?

Cubie is most relevant when the operator owns the GPU economics and feels the pain of wasted capacity.

- Private AI clusters and enterprise GPU fleets.
- Sovereign AI and regulated data center operators.
- Neocloud or inference providers with power, rack, or capital constraints.
- Token-marketplace operators that need more completed output per megawatt.
- Operators serving regulated tenants that need workload admission evidence.

6. When is Cubie less compelling?

Cubie is less urgent where the operator does not bear the cost of inefficient workload behavior, where capacity is abundant, or where all value is already captured through customer-paid usage regardless of waste.

Even then, security, compliance, and SLA risk may still create a reason to evaluate it.

2. Core Diligence FAQ

Direct answers to the skeptical questions data center buyers will ask.

7. Is Cubie just NeMo Guardrails at the same level?

No. Guardrails control model behavior. Cubie controls whether the workload should reach the GPU path in the first place.

NeMo-style guardrails, provider guardrails, and application filters are still useful. Cubie is not trying to replace them. It adds a different control point: admission before accelerator spend.

Customer-safe wording

"Guardrails govern model behavior. Cubie governs compute admission."

8. Is this just a "we are more clever than the other vendors" proposal?

It should not be presented that way. The credible claim is not that Cubie is smarter than NVIDIA, Google, or the LLM. The credible claim is that Cubie solves a narrower problem earlier and more cheaply.

Cubie does not need to solve the prompt. It needs to decide whether the request is valid enough to spend GPU capacity.

9. Why would a pre-GPU gate make a better decision than the processor/model?

It is not making a better answer decision. It is making a different decision. The processor/model decides what answer to generate. Cubie decides whether the workload deserves access to the expensive execution path.

That is why the underwriting analogy matters. A lender does not run the borrower's business; it checks whether capital should be deployed. Cubie does the same for inference capacity.

10. Does Cubie inspect prompts?

The preferred deployment posture is no semantic prompt inspection and no prompt understanding. Cubie should operate on workload evidence: identity, authorization, freshness, replay state, hashes, topology, policy, budget, hardware proof, and telemetry.

If a specific deployment requires raw prompt or payload inspection, that should be disclosed and reviewed because it changes the privacy, security, and regulatory posture.

Customer-safe wording

"Cubie does not need to read the tenant's prompt to decide whether the workload is admissible."

11. A prompt-at-a-time filter lacks full context. How does Cubie address that?

A prompt-at-a-time filter is weak. Cubie should not be positioned that way.

Cubie's stronger story is workload-state aggregation. It can evaluate not only a single request, but also repeated behavior, cascades, replay patterns, trust-boundary movement, and workload drift across time. That is how it becomes an infrastructure control rather than a simple filter.

12. What exact decision does Cubie make?

At the core, Cubie makes an admit or deny decision. At the platform level, that can be wrapped into operational actions such as admit, deny, hold, escalate, or quarantine.

The important point is that the decision happens before GPU capacity is spent.

- Admit: the workload can enter the GPU path.
- Deny: the workload fails required checks.
- Hold: the workload is not rejected, but should wait due to timing, policy, or capacity.
- Escalate: the workload requires human or higher-policy approval.

- Quarantine: the workload appears suspicious or attack-like.

13. What baselines should Cubie beat?

Cubie should be compared against the customer's current stack, not an artificial "send everything to the biggest model" baseline.

Relevant baselines include no pre-GPU gate, API gateway controls, rate limits, JWT-only checks, guardrails, prompt classifiers, provider-native routing, model downshifting, runtime schedulers, KV-cache optimizers, and internal platform rules.

Customer-safe wording

"We do not claim to beat every layer. We claim to add value at an earlier admission boundary."

14. Why would NVIDIA, Google, OpenAI, Anthropic, or an internal platform team not just absorb this?

If Cubie were only prompt routing or generic filtering, absorption risk would be high. The defensible wedge is operator-owned admission control across tenants, models, hardware, policies, and regulated workloads.

NVIDIA can optimize NVIDIA runtime. Google can optimize Gemini. Cubie gives the data center operator a neutral trust boundary before accelerator spend.

15. How is this different from Gemini downshifting or model routing?

Model routing decides which model or reasoning level should handle a request after the request is accepted into the model path. Cubie decides whether the request should enter that path at all.

Those capabilities are complementary. Cubie sits earlier.

16. Is a 10-15% savings claim credible?

It is credible only after the denominator is defined and validated. "15% savings" is weak unless it says whether the savings refer to GPU cost, tokens, latency, total inference COGS, cost per accepted answer, or completed workload per megawatt.

The stronger metric is net cost per completed tenant workload at equal quality, safety, and latency.

Customer-safe wording

"We quantify reclaimable inference capacity from your telemetry before asking you to believe a percentage."

17. Are incremental savings meaningful if the AI market is growing exponentially?

They are meaningful when the data center is constrained by power, cooling, rack space, debt, or GPU supply. In that environment, efficiency is not a side project. It is capacity expansion.

Every recovered GPU-hour can become capacity that does not require new grid power, new racks, new debt, or a hardware refresh.

18. Is Cubie a product or a feature?

Prompt classification is a feature. Pre-GPU admission, capacity recovery, security enforcement, audit evidence, and tenant-facing controls can be a product.

The customer should judge Cubie by whether it creates an operator-owned control boundary with measurable business and security value.

3. CTO and AI Platform FAQ

Integration, architecture, latency, and workload behavior.

19. Where does Cubie sit architecturally?

Cubie is best positioned at the pre-GPU boundary: top-of-rack, gateway, secure enclave, policy gateway, MCP gateway, or other operator-controlled admission point before accelerator dispatch.

The exact insertion point depends on the data center architecture. The principle stays the same: check before GPU spend.

20. Does Cubie require replacing GPUs?

No. The best commercial framing is brownfield: improve the fleet already deployed. Cubie should not be sold as a rip-and-replace hardware project.

The value is especially strong when the operator cannot quickly add power, racks, or another GPU cluster.

21. Does Cubie replace NVIDIA runtime optimization, vLLM, KServe, or other serving-stack improvements?

No. Runtime optimization improves work after it is admitted. Cubie protects admission before the runtime spends accelerator capacity.

A mature deployment can use both: Cubie for admission, runtime tools for serving efficiency.

22. How does Cubie help with context bombs, retry storms, KV-cache problems, or agent loops?

Cubie does not need to semantically understand every bad prompt. It helps by forcing workloads through deterministic admission before they consume VRAM, KV cache, queue slots, or SLA headroom.

The practical target is blast-radius reduction: stop invalid or capacity-destructive work before it becomes GPU work.

23. What latency should we expect?

Use conservative language. Cubie is designed for extremely low-latency admission, but full-path latency must be validated in the customer environment.

A pilot should report p50, p95, p99, and tail behavior under production-like load, including logging, telemetry, policy checks, and failover behavior.

24. What happens if Cubie is uncertain or a dependency fails?

The safest production posture depends on the workload class. Regulated or high-risk workloads may fail closed. Less sensitive workloads may degrade, hold, or route to manual review.

The customer should decide policy by workload class, tenant tier, SLA, and compliance requirements.

25. Does Cubie need to store customer data?

The preferred posture is minimal-data operation: workload metadata, hashes, authorization state, policy state, telemetry references, and audit outcomes. Raw prompt or payload storage should not be assumed and should not be required for the core value proposition.

For regulated customers, the deployment should explicitly document what is logged, what is redacted, what is retained, and who can access it.

4. CFO and Commercial FAQ

How to evaluate the business case without overclaiming.

26. What is the financial model?

The cleanest model is capacity recovery: measure how much paid-for inference capacity is not becoming completed tenant workload, then validate how much Cubie can reclaim.

A simple formula is: annual paid-for inference capacity multiplied by measured goodput leakage equals the capacity opportunity. The final commercial model should be tied to operator-verified recovery, not vendor-estimated savings.

27. What should we measure?

The top business metric is completed tenant workload per GPU-hour or megawatt at equal quality, safety, and latency.

Secondary metrics include queue time, retry rate, invalid request rate, KV-cache pressure, denial rate, false-deny rate, false-allow rate, SLA impact, p95/p99 latency, and audit evidence generated.

28. How should a pilot be structured?

A credible pilot should avoid “trust us” economics. Start with telemetry, then shadow decisions, then controlled enforcement.

- Phase 1: Telemetry baseline - current workload leakage and capacity pressure.
- Phase 2: Shadow mode - Cubie scores without enforcement.
- Phase 3: Low-risk enforcement - replayed, stale, unauthorized, malformed, or out-of-budget requests.
- Phase 4: Verified recovery - measure completed workload and tenant impact.
- Phase 5: Commercial rollout - tie fees to verified recovered capacity where appropriate.

29. What is the expected ROI?

The honest answer is that ROI is deployment-specific. It depends on workload mix, tenant behavior, invalid traffic, agent loops, retry patterns, GPU utilization, power constraints, and current admission controls.

Do not lead with a universal ROI number. Lead with an operator-specific measurement process.

5. Cybersecurity FAQ

For CISOs, SOC teams, incident responders, and AI security owners.

30. Does Cubie solve cybersecurity?

No. Cubie should not be presented as a full cybersecurity platform.

Cubie reduces specific AI workload and agent-action risks by enforcing admission before execution. It complements IAM, WAF, API gateways, SIEM, EDR, CNAPP, DLP, and zero-trust controls.

Customer-safe wording

“Cubie does not replace the security stack. It adds a pre-execution trust gate for AI workloads and agent actions.”

31. What cybersecurity risks can Cubie help reduce?

Cubie is relevant where the risk is tied to whether a workload or agent action should be allowed to execute.

- Unauthorized or out-of-scope agent tool calls.
- Replayed, stale, or reused tokens.
- Spoofed surface identity without internal proof.
- Cross-realm or boundary-crossing actions without required evidence.
- High-rate invalid admission attempts that would otherwise consume backend resources.
- Malformed or capacity-destructive workload patterns before GPU execution.
- Incomplete auditability around why a workload was allowed or denied.

32. Does Cubie solve prompt injection?

No. Cubie does not semantically solve all prompt injection. A prompt injection can still influence a model if downstream permissions and application logic allow the harmful action.

Cubie helps when the injected behavior attempts to cross a policy, identity, consent, tool, budget, or trust boundary. In that case, Cubie can deny, hold, escalate, or produce evidence before execution.

33. How does Cubie fit into zero trust?

Cubie aligns with zero-trust principles at the workload level: verify the action before execution, avoid implicit trust, enforce least privilege, fail safely, and produce an audit trail.

The difference is that Cubie applies this logic to AI workload admission and agent actions, not just user login or network access.

34. Does Cubie replace IAM?

No. IAM identifies users, services, workloads, and permissions. Cubie consumes identity and policy evidence as part of a broader admit/deny decision.

Think of IAM as one input. Cubie is the pre-execution decision layer that asks whether the action should proceed right now under current context.

35. Does Cubie replace WAF, API gateways, or rate limits?

No. Those controls are useful, but they are usually not enough for AI workloads. They may know the request shape, source, or rate, but not whether the workload is valid, fresh, authorized, budgeted, and safe to spend GPU capacity.

Cubie should sit alongside those controls, not pretend they are obsolete.

36. What should Cubie send to the SOC or SIEM?

The useful security output is not raw prompt text. It is structured admission evidence.

- Admit/deny/hold/escalate outcome.
- Reason or witness category.

- Tenant or workload scope.
- Time, freshness, and replay state.
- Policy boundary crossed or attempted.
- Hash/reference of the workload artifact, where appropriate.
- Severity, confidence, and recommended response workflow.
- Linkage to incident, compliance, or tenant evidence records.

37. What should security teams validate in a pilot?

Security teams should test false allows, false denies, policy bypass attempts, replay attempts, stale-token behavior, identity spoofing, high-rate invalid traffic, logging fidelity, and incident workflow integration.

They should also confirm the data-minimization posture: what is logged, what is never logged, what is hashed, and how long records are retained.

38. How does Cubie relate to confidential computing?

Cubie complements confidential computing. Confidential computing protects execution boundaries. Cubie helps decide whether a workload should enter that boundary in the first place.

It should not be described as a replacement for enclave security, attestation, key management, or confidential runtime controls.

6. Legal, Compliance, and Regulated Tenant FAQ

EU AI Act readiness, audit evidence, and honest boundaries.

39. Is Cubie itself an AI system?

The runtime should be positioned as deterministic infrastructure control, not as a generative AI model or semantic prompt evaluator. Final classification depends on the exact implementation, deployment mode, and legal review.

If an onboarding or design-time tool uses an LLM to help configure policy, that component should be evaluated separately from the runtime admission gate.

40. Does Cubie make us EU AI Act compliant?

No. Cubie supports EU AI Act readiness; it does not replace legal compliance, system classification, conformity assessment, governance procedures, or deployer obligations.

Cubie can help produce control evidence: admission records, denial reasons, policy enforcement history, human-oversight hooks, incident evidence, and technical documentation support.

Customer-safe wording

“Cubie supports compliance evidence. It does not make a customer compliant by default.”

41. What value does Cubie provide to regulated tenants?

Regulated tenants do not only need throughput. They need evidence that workloads were governed before execution.

Cubie can support tenant-facing services such as workload admission records, denial explanations, policy history, audit exports, and incident traceability.

42. What should legal and compliance review before production?

They should review data processing scope, prompt/payload handling, audit retention, cross-border data movement, Article 50-style transparency workflows where applicable, incident reporting workflows, tenant disclosures, and whether the deployment changes the operator’s role or obligations.

7. Tenant, Marketplace, and Partner FAQ

How the value shows up beyond the infrastructure team.

43. What does a tenant actually experience?

A tenant should experience more predictable throughput, fewer bad-workload cascades, clearer rejection reasons, and stronger audit evidence for regulated workloads.

Cubie should be transparent enough to explain why a workload was denied without exposing sensitive payload content.

44. How can Cubie become an incremental service line?

The same admission layer that protects GPU capacity can be packaged as a tenant-facing trust and compliance feature.

Examples include regulated workload admission logs, audit-ready evidence exports, tenant-level policy enforcement reporting, and optional premium governance tiers.

45. Why would token-marketplace or inference-routing operators care?

Operators that monetize token output care about completed workload, not just activity. If invalid or low-yield traffic consumes capacity, it crowds out revenue-generating work.

Cubie gives these operators a way to protect capacity before routing or GPU execution.

46. What should channel partners or systems integrators emphasize?

Emphasize brownfield deployment, customer telemetry validation, security-stack complementarity, and tenant-facing evidence services. Avoid overclaiming savings, compliance, or universal latency.

8. Proof-Backed Points to Use Carefully

High-level proof logic without repo filenames or theorem citations.

47. What are the strongest proof-backed “mic drop” moments?

Use these at a high level. Keep detailed proof packages for technical diligence or NDA review.

- Conservative gate: a workload only passes when all required trust conditions pass.
- Explainable failure: a denied workload has a named reason or witness category.
- Boundary enforcement: actions crossing a trust boundary must satisfy the gate first.
- No partial success: an incomplete trust state should not be treated as a clean pass.
- Deny-before-spend: invalid work is stopped before it consumes accelerator capacity.
- Surface identity is not enough: the request must carry internal evidence, not just look valid.
- Aggregation matters: repeated or cascading bad workload behavior can be detected across groups and time windows.
- Auditability matters: the system creates evidence for why work was admitted or denied.

48. What should we not overclaim?

Do not turn proof-backed engineering into broad commercial claims that have not been validated on the customer’s fleet.

- Do not claim guaranteed savings.
- Do not claim Cubie replaces NVIDIA, Google, NeMo, vLLM, KServe, or the security stack.
- Do not claim full-path production latency without deployment-specific measurements.
- Do not claim universal no-prompt logging unless the deployment is configured and verified that way.
- Do not claim EU AI Act compliance by default.
- Do not claim Cubie solves all prompt injection or cybersecurity risk.

9. Pilot Success Criteria

What a serious customer should require before production.

Area	What to validate	Why it matters
Goodput	Completed workload per GPU-hour and per megawatt.	This is the business case.
Latency	p50, p95, p99, and tail behavior under realistic load.	A trust gate cannot create a new bottleneck.
Accuracy	False deny, false allow, false hold, and false escalation.	Bad decisions damage tenants or security.
Security	Replay, stale-token, spoofing, boundary-crossing, and high-rate invalid traffic tests.	Proves cybersecurity relevance without overclaiming.
Integration	Gateway, ToR, MCP, policy, runtime, SIEM, and tenant reporting fit.	Determines production feasibility.
Economics	Operator-verified reclaimed capacity, not vendor-modeled savings.	Keeps the commercial model honest.
Compliance evidence	Admission records, denial reasons, oversight hooks, audit export.	Supports regulated tenants and internal governance.

10. Language Guide

Use these phrases externally; avoid the weaker or riskier alternatives.

Use	Avoid
Pre-GPU trust gate	Smarter preprocessor
Compute underwriting	Prompt classifier
Reclaim paid-for inference capacity	Guaranteed savings
Improve token efficiency / inference goodput	Improve utilization only
Operator-owned admission boundary	Competes directly with every runtime optimizer
Supports EU AI Act readiness	EU AI Act compliant by default
Reduces specific AI workload-admission risks	Solves cybersecurity
Telemetry-verified recovery	Universal percentage claim

Best final answer

Cubie is not a smarter prompt preprocessor. It is a pre-GPU trust gate that underwrites whether an AI workload should be allowed to spend GPU capacity before the GPU funds the request. The business case is measured in reclaimed paid-for inference capacity, higher completed tenant workload per GPU-hour or megawatt, reduced bad-workload blast radius, and stronger audit evidence for regulated AI deployments.