MetaFuze FuZeCORE + FuZeLLM

A practical platform for private inference, controlled model flexibility, and RRLM-based orchestration.

Revision: 20260103-v7

Customer-facing summary - proprietary routing internals withheld (available under NDA).

Overview

MetaFuze.ai delivers private AI through two composable layers:

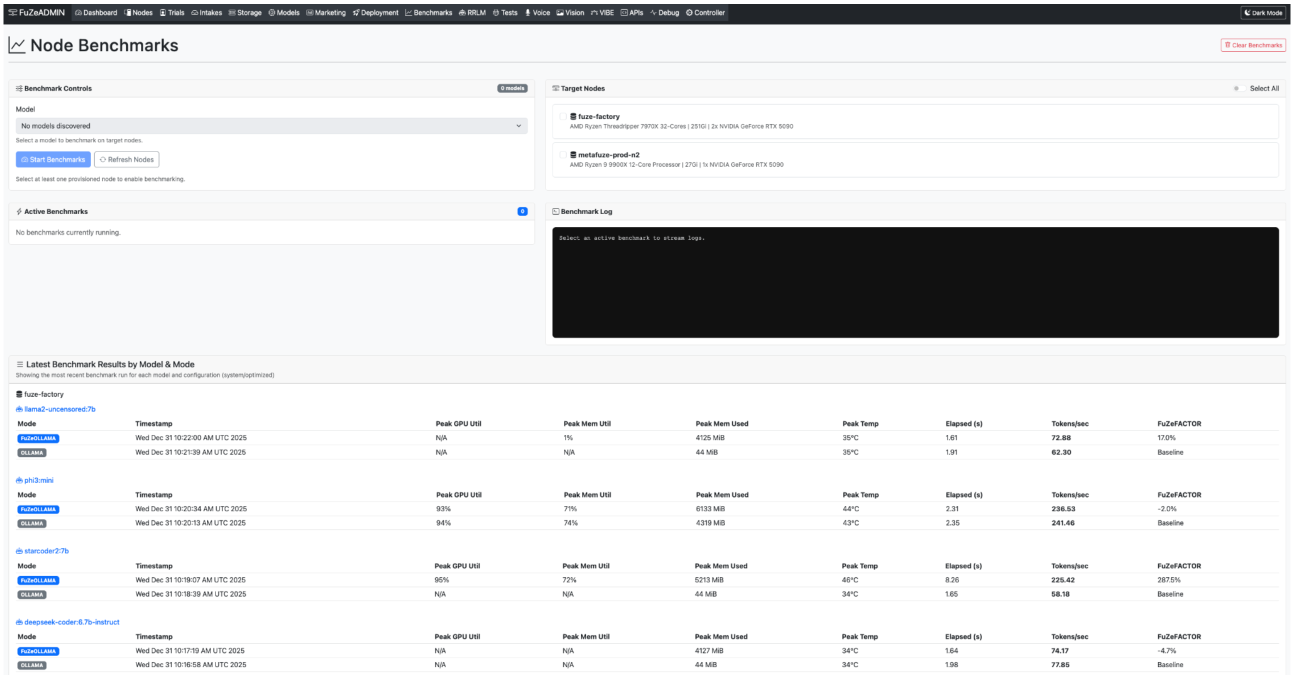
- **FuZeCORE**: a cost-effective hosted LLM endpoint, tuned and benchmarked across runtime stacks and hardware classes to reduce GPU overspend.
- **FuZeLLM**: an optional add-on that introduces RRLM-based routing, persona packs, and policy-aware orchestration across multiple model backends.
- **FuZeADMIN**: the control plane that governs model catalog, rollout/rollback, and fleet operations across FuZeBOX and FuZeCLOUD.

A key design principle is a **dumb UI**: front-end clients send user input and display results, while routing, policy, safety boundaries, and auditing live in backend services.

Many organizations adopt private LLMs by selecting a large GPU instance and a large model 'just in case'. This approach is expensive, and it often underutilizes hardware. FuZeCORE replaces guesswork with benchmark evidence.

What gets benchmarked

FuZeLABS maintains a repeatable harness to compare runtime stacks and tuning profiles (for example, baseline Ollama vs FuZe-optimized profiles, and other supported inference engines) under consistent settings. The goal is not one global winner; it is the best-fit combination for a customer's hardware and target workload.



Node Benchmarks

Benchmark Controls

Model: No models discovered

Select a model to benchmark on target nodes.

Select at least one provisioned node to enable benchmarking.

Target Nodes

fuze-factory
AMD Ryzen Threadripper 7970X 32-Cores | 251Gi | 2x NVIDIA GeForce RTX 5090

metafuze-prod-n2
AMD Ryzen 9 9900X 12-Core Processor | 27Gi | 1x NVIDIA GeForce RTX 5090

Active Benchmarks

No benchmarks currently running.

Benchmark Log

```
Select an active benchmark to stream logs.
```

Latest Benchmark Results by Model & Mode

Showing the most recent benchmark run for each model and configuration (system/optimized)

fuze-factory

llama2-uncensored:7b

Mode	Timestamp	Peak GPU Util	Peak Mem Util	Peak Mem Used	Peak Temp	Elapsed (s)	Tokens/sec	FuZeFACTOR
FuZeOLLAMA	Wed Dec 31 10:22:00 AM UTC 2025	N/A	1%	4125 MiB	35°C	1.61	**72.88**	17.0%
OLLAMA	Wed Dec 31 10:21:39 AM UTC 2025	N/A	N/A	44 MiB	35°C	1.91	**62.30**	Baseline

phi3:mini

Mode	Timestamp	Peak GPU Util	Peak Mem Util	Peak Mem Used	Peak Temp	Elapsed (s)	Tokens/sec	FuZeFACTOR
FuZeOLLAMA	Wed Dec 31 10:20:34 AM UTC 2025	93%	71%	6133 MiB	44°C	2.31	**236.53**	-2.0%
OLLAMA	Wed Dec 31 10:20:13 AM UTC 2025	94%	74%	4319 MiB	43°C	2.35	**241.46**	Baseline

starcoder2:7b

Mode	Timestamp	Peak GPU Util	Peak Mem Util	Peak Mem Used	Peak Temp	Elapsed (s)	Tokens/sec	FuZeFACTOR
FuZeOLLAMA	Wed Dec 31 10:19:07 AM UTC 2025	95%	72%	5213 MiB	46°C	8.26	**225.42**	287.5%
OLLAMA	Wed Dec 31 10:18:39 AM UTC 2025	N/A	N/A	44 MiB	34°C	1.65	**58.18**	Baseline

deepseek-coder:6.7b-instruct

Mode	Timestamp	Peak GPU Util	Peak Mem Util	Peak Mem Used	Peak Temp	Elapsed (s)	Tokens/sec	FuZeFACTOR
FuZeOLLAMA	Wed Dec 31 10:17:19 AM UTC 2025	N/A	N/A	4127 MiB	34°C	1.64	**74.17**	-4.7%
OLLAMA	Wed Dec 31 10:16:58 AM UTC 2025	N/A	N/A	44 MiB	34°C	1.98	**77.85**	Baseline

Figure 1: sample benchmark view (illustrative). Benchmarks are generated by a consistent harness to support apples-to-apples comparisons. Exact workloads and settings are provided with customer quotes.

Outcome

Customers start with the smallest model class and GPU profile that meets throughput/latency targets, and expand only when evidence shows it is needed. This reduces ongoing GPU spend and avoids paying for parameters that are not required.

FuZeLABS work focuses on measurable outcomes: higher tokens-per-second, predictable latency under load, and stable behavior across upgrades. Implementation details vary by engine and hardware. Public documentation describes benefits without disclosing proprietary techniques.

Optimization areas

- Runtime profile tuning (threading, batching, memory settings) aligned to GPU/CPU and workload.
- CUDA and kernel-level tuning where supported by the underlying stack, validated by repeatable benchmarks.
- Guardrailed multi-model hosting so customers can run a curated set of models concurrently when capacity allows.

Optional KV/cache persistence

For deployments that want faster follow-up responses on similar queries, MetaFuze can enable an optional cache profile. When enabled, cache scope is explicitly bound to a workspace and policy domain (not global), with clear retention controls. This can improve responsiveness for repeated workflows while preserving privacy boundaries.

FuZeLLM extends FuZeCORE by introducing routing across multiple model backends (for example, general assistant, coding-focused models, and domain specialists). The routing system keeps track of the active identity context (company/persona/person/domain) so quality signals are interpreted within the correct domain.

Q-network learning loop (high-level)

FuZeLLM can incorporate a reinforcement-style scoring layer that learns which routing choices lead to better outcomes. Public-facing examples of reward signals include: follow-up coherence (did the next user message indicate success), sentiment and frustration signals, task completion cues, and domain-specific acceptance checks. Proprietary model selection and weighting logic is available under NDA.

Persona packs

Persona packs provide pre-defined behavioral guardrails and tool-use patterns (e.g., engineering, architecture, operations, medical, executive). Higher tiers can enable a broader set of personas, while keeping the UI unchanged.

MetaFuze FuZeCORE + FuZeLLM

A practical platform for private inference, controlled model flexibility, and RRLM-based orchestration.

Buyers are rightly skeptical of 'no lock-in' claims. MetaFuze focuses on **no model lock-in** with operational guardrails. FuZeADMIN is the control plane that turns flexibility into a governed process.

Revision: 20260103-v7

Customer-facing summary - proprietary routing internals withheld (available under NDA).

- Curated catalog by tier to avoid unmanaged 'model zoo' behavior.
- Versioned artifacts with checksums; pin versions per workspace or environment.
- Pre-prod validation using the same benchmark harness and a customer acceptance prompt set.
- Safe promotion with rollback to a known-good version when behavior or performance changes.
- Fleet management: add FuZeBOX workers and FuZeADMIN distributes models across the fleet; hybrid expansion to FuZeCLOUD is supported.

TrulyPrivate boundary

TrulyPrivate keeps inference and data inside the customer's controlled boundary. Logs, policies, and model artifacts remain within the deployment scope defined by the customer.

Tiers are pre-configured around an approximate GPU memory budget to reduce decision fatigue. Model fit and concurrency are configuration dependent (architecture, context length, precision/quantization, and memory headroom).

Tier	Includes	Typical model class	Examples (illustrative)
Small	FuZeCORE only	Quantized up to ~30B class (Q4/Q8)	codegemma:7b, mistral:7b-instruct, phi3:mini, qwen2.5-coder:14b
Medium	FuZeCORE + FuZeLLM	Up to ~70B class (FP16/Q8), config dependent	70B class Llama variants (Q8), gpt-oss-20b, larger coding and ge
Large	FuZeCORE + FuZeLLM	Up to ~120B class (FP16), config dependent	120B class general models, optional larger MoE deployments via

* Configuration dependent. Model fit varies by architecture, context length, precision/quantization, and available memory headroom. Very large models require multi-node sharding and are quoted based on hardware/network requirements.

Appendix B: glossary

RRLM: routing layer that selects among model backends based on intent, policy, and learned outcome signals.
Persona pack: pre-defined behavioral guardrails and tool-use patterns for a specific role or workflow.
TrulyPrivate: a deployment boundary that keeps inference, data, and artifacts inside customer-controlled infrastructure.