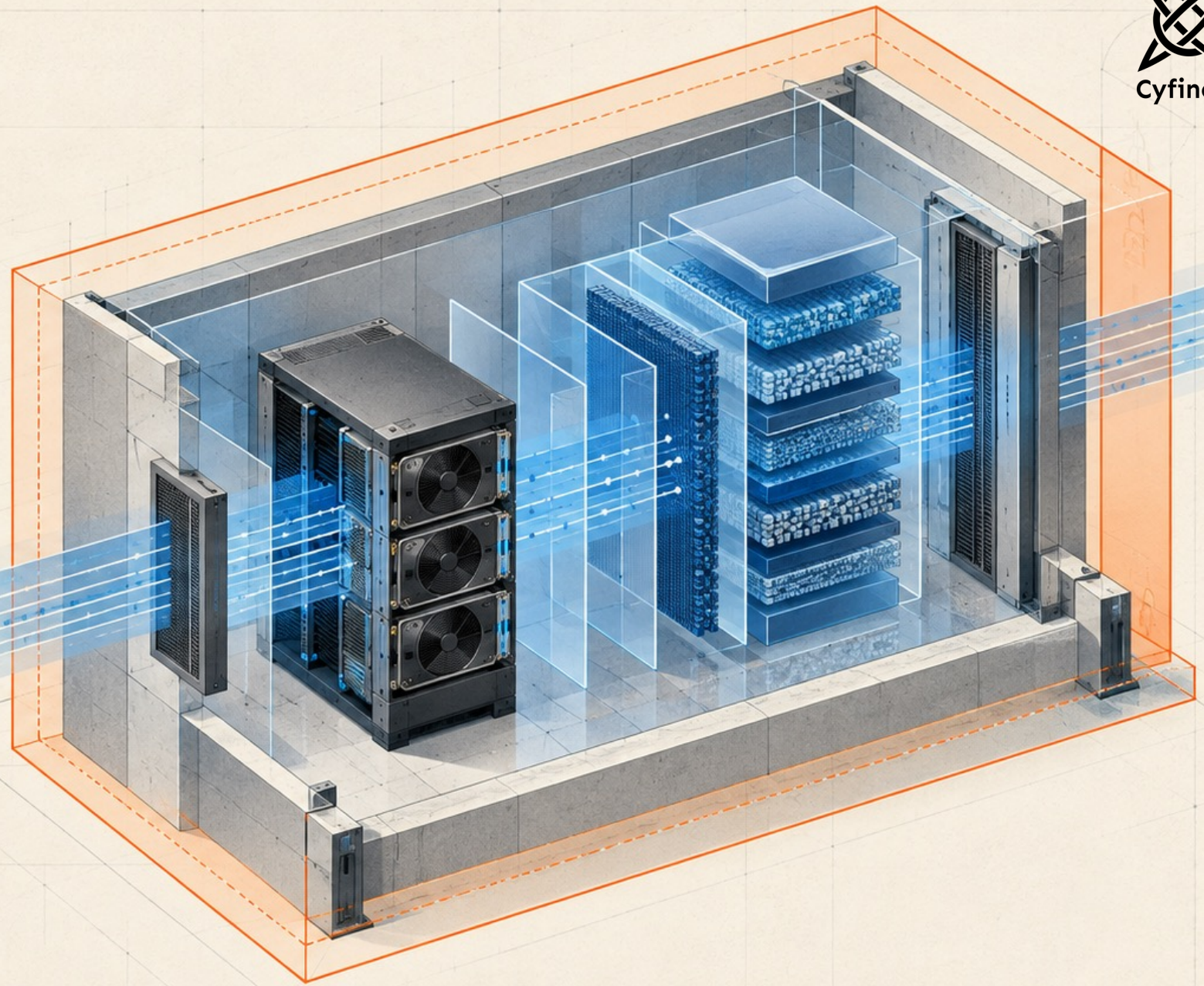
Building, Owning & Securing

Private AI Infrastructure



Cyfinoid

Anant Shrivastava

Cyfinoid

What is private AI Infrastructure?

- An Infrastructure setup where we control
  - What **hardware** & **technology stack** to use
  - Which **models** to run and when
  - Who gets **access** to the environment
- We also control
  - **Ownership** of data (who gets access to it)
  - **Residency** of data (where is it stored)
  - **Retention** of data (deletion / storage timeframe)

Cyfinoid

Myth : Private AI is low cost?

- Private never means free or low cost or cheap.
- Private is generally more costly than SaaS on per invocation basis
- Include the cost of hardware and electricity and ownership

- However,
  - Private gives you control
  - Private gives you ownership
  - Private gives you certainty

More reasons for private

OpenAI @OpenAI

We congratulate Levent Alpöge and Tristan Buckmaster on their remarkable mathematical work.

We (the researchers and the agents) did not see any of their work through any means until they released it publicly — in particular, no specific user data was accessed in order to solve this problem.

While unlikely, we cannot rule out that de-identified data derived from their usage of our products helped improve our models.

However, our proofs differ significantly and even the precise results proved are different in the Euler case (forced vs. unforced).

GTG-16002: Moonshot serves Claude instead of Kimi and collects exchanges for model training

We discovered that Moonshot AI, the company that produces the Kimi family of models, silently forwarded customer requests to Claude, instead of processing them using Kimi. Moonshot then displayed Claude's responses to users. These users thought they were using a Kimi model, but received responses from Claude instead.

In one instance, over a ten-day period, Moonshot relayed almost 300,000 customer requests to Anthropic, the vast majority of which were routed to Opus. Moonshot used a proxy service network of 5,380 fraudulent accounts, most of which appeared to be located in Singapore and Japan.

Training

Allow paid endpoints that train on request data
Allow Go models that require consent to train on your requests.

Allow free endpoints that train on request data
Free models may train on your requests. Turn off to disable all free models.

Chaofan Shou @shouccc

26 LLM routers are secretly injecting malicious tool calls and stealing creds. One drained our client $500k wallet.

We also managed to poison routers to forward traffic to us. Within several hours, we can directly take over ~400 hosts.

Check our paper: arxiv.org/abs/2604.08407

Your Agent Is Mine: Measuring Malicious Intermediary Attacks on the LLM Supply Chain

GTG-50029: Hacktivists targeted European political and affiliated entities

AI has helped to close the capability gap turning low-level "hacktivists" into advanced persistent threats. As demonstrated repeatedly throughout our case studies, AI capabilities raise the baseline as well as reduce the resource requirements for offensive cyber operators. In this section, we provide details of a hacktivist campaign we investigated and disrupted, in which small well-motivated operations were able to achieve significant goals due to the integration of AI in their operations.

Each endpoint toggle controls routing to endpoints with a specific data policy: enabling allows cludes them from routing. Learn more

ır training purposes.

Allow free endpoints that train on request data
Enable providers serving free models that may retain and/or train on prompts and completions.

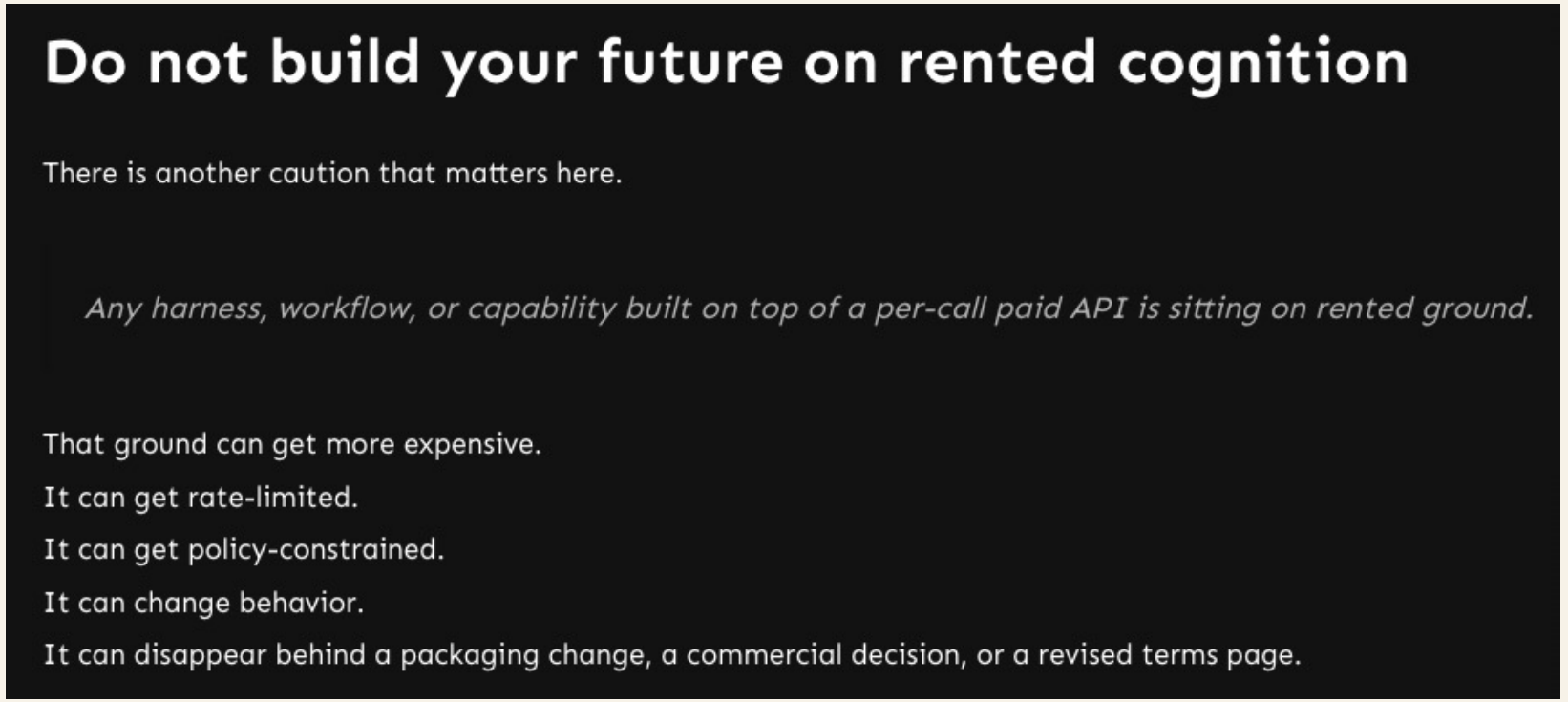
Allow free endpoints that publish prompts
Enable providers serving free models that may publish prompts and completions to public datasets.

Allow 1% data discount in workspaces
Allow workspaces to consent to OpenRouter using your inputs/outputs to improve the product. Each workspace consents separately.

https://x.com/shouccc/status/2042423713019412941
https://www.anthropic.com/threat-intelligence-report-september-2026
https://x.com/OpenAI/status/2097375276384567642

Why SaaS can be problematic

- SaaS* == Rented Cognition



Do not build your future on rented cognition

There is another caution that matters here.

Any harness, workflow, or capability built on top of a per-call paid API is sitting on rented ground.

That ground can get more expensive.

It can get rate-limited.

It can get policy-constrained.

It can change behavior.

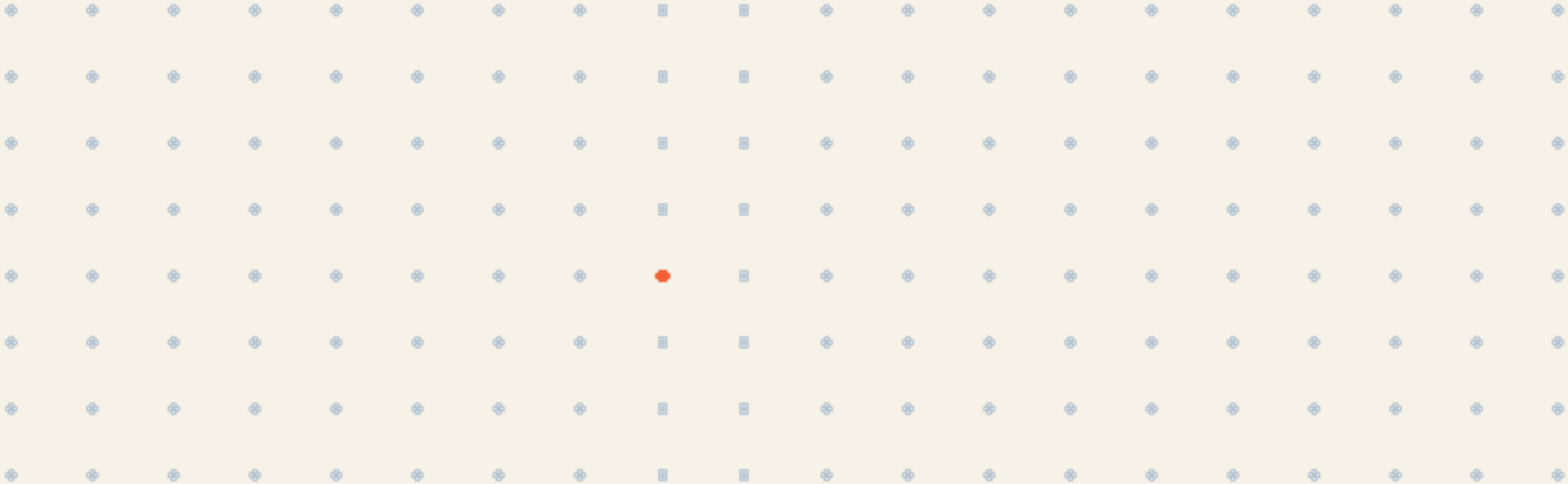
It can disappear behind a packaging change, a commercial decision, or a revised terms page.

*API access to models on per api call basis (free or charged)

https://cyfinoid.com/a-pragmatic-guide-to-being-mythos-ready/

Cyfinoid

For SaaS you are 1 in million



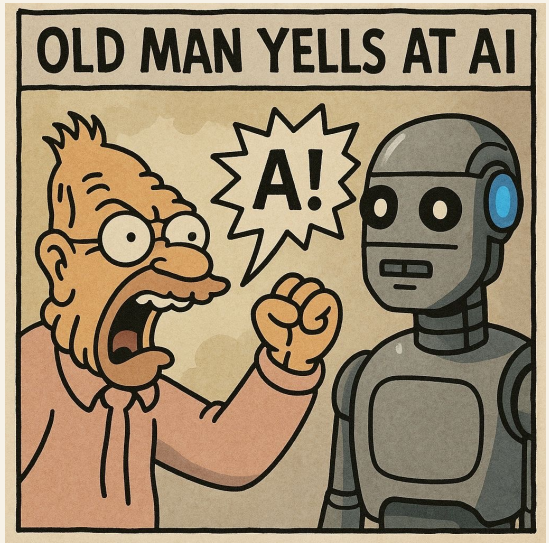
Shared systems converge on rules that fit the broad middle - and often fit nobody perfectly

Cyfinoid

Rant on SaaS

- Charge is per token not on actual work done
- Rate limits are per 5h, per week, per month, per mood of the CEO
- Peak & off-peak charges to push people towards off-peak

- SaaS Infra has
  - Reliability issues (outside your control)
  - Performance impact (outside your control)
  - Random cost variance



https://www.linkedin.com/posts/balaganski_surely-im-not-the-only-person-feeling-like-activity-7402059235875844097-n8TH

Cyfinoid

So, you want Private AI

Cyfinoid

Approaches & Responsibilities for Private AI

	Personal	Corporate Ownership	Corporate Renting
	Hardware @ Home	Hardware @ Data Center	Cloud Hosting
Agent/Harness	User	User	User
Tech Stack	User	User	User
Model	User	User	User
OS	User	User	User / Provider
Hardware	User	User / Provider	Provider
Host	User	Provider	Provider
Network	User	Provider	Provider
Power	User	Provider	Provider
Environment	User	Provider	Provider
Cost	Upfront + Monthly	Upfront + Monthly	Monthly

Cyfinoid

Cloud Options

Almost all cloud service providers are now GPU service providers.

Cyfinoid

Lets Build our Private AI

Hardware

- For Private AI we are equating it to LLM's **from here onwards.**
- LLM's need 2 things : Processing power and faster memory

- Dedicated GPU + Associated PC
- Unified Memory based machine

GPU vs Unified Memory PC

Dedicated GPU

- Larger Bandwidth
- Supports Training
- Best for Dense Models
- Largest consumer level single GPU is **RTX Pro 6000 : 96GB**
- Server class GPU cluster **(NVIDIA GB200 NVL72) :13.4 TB**
- Server Class GPU: **GB200: 372GB**

Unified Memory

- System on Chip (SoC)
- Support Larger Models
- Best for Mixture of Experts
- Maximum RAM ~ 512 GB MacStudio
- Multiple Devices can be combined via dedicated cables

Examples

Dedicated Graphics

Unified Memory

Cyfinoid

Let’s get Models

- LLM Models are a collection of
  - Weights (tensors)
  - Architecture hyperparameters
  - Tokenizer
  - Chat / Prompt template

Hugging Face

- Common types
  - Safetensor
  - Gguf

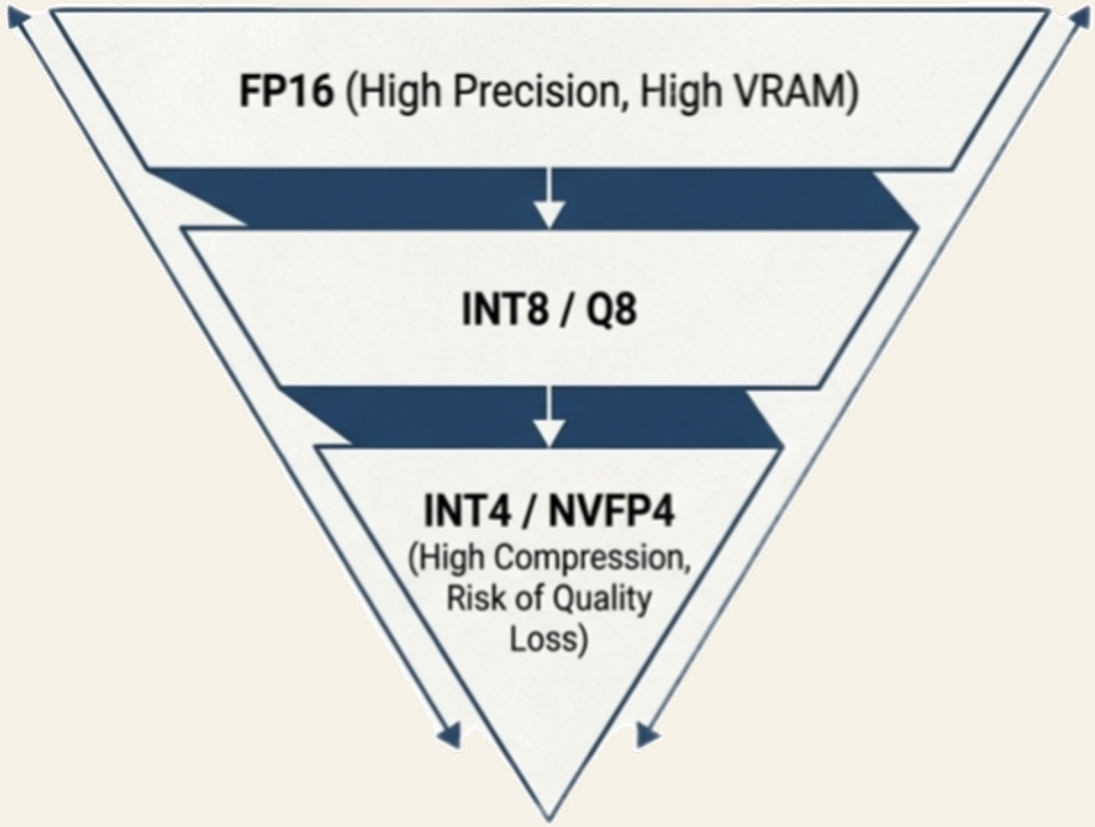
ModelScope 魔搭社区

Terminology : Quantization

- Process to store the weights in fewer bits.

- BF16 / FP16 / FP32 : Uncompressed high precision weights
- INT8 , FP8, Q8_0 : 8 bit vector roughly half size
- INT4, Q4_K_M, NVFP4 : 4 bits to represent a vector

Allows for reducing VRAM requirement with trade off in accuracy



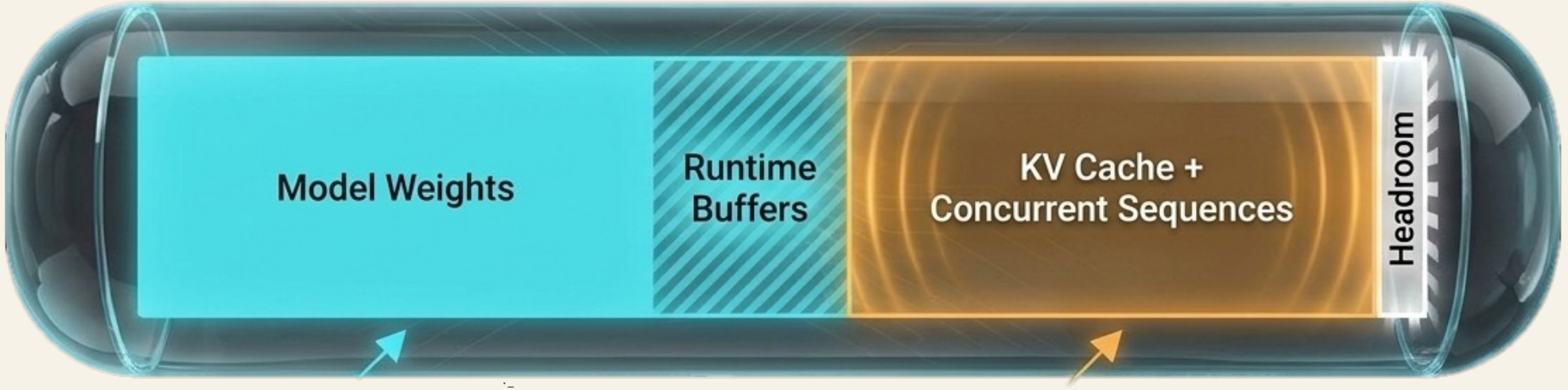
Cyfinoid

KV Cache

- Key Value caching layer for LLM
- Grows as more work is done based on context size defined
- Can also be quantized like the model weights
- It’s generally recommended to keep KV Cache as non quantized as possible.
  - If weights are Q4 but KV Cache is Q8 or F16 : accuracy would be better
- If needed quantize model first then KV Cache

VRAM Fit

- Total VRAM : Weights + KV Cache + Buffer + Concurrency
- Prefer Filling as much VRAM as possible keeping above values intact
- Model Q8 will always be better performing then Q4
- Avoid compressing KV Cache too much F16 max Q8 if it can be helped.



Cyfinoid

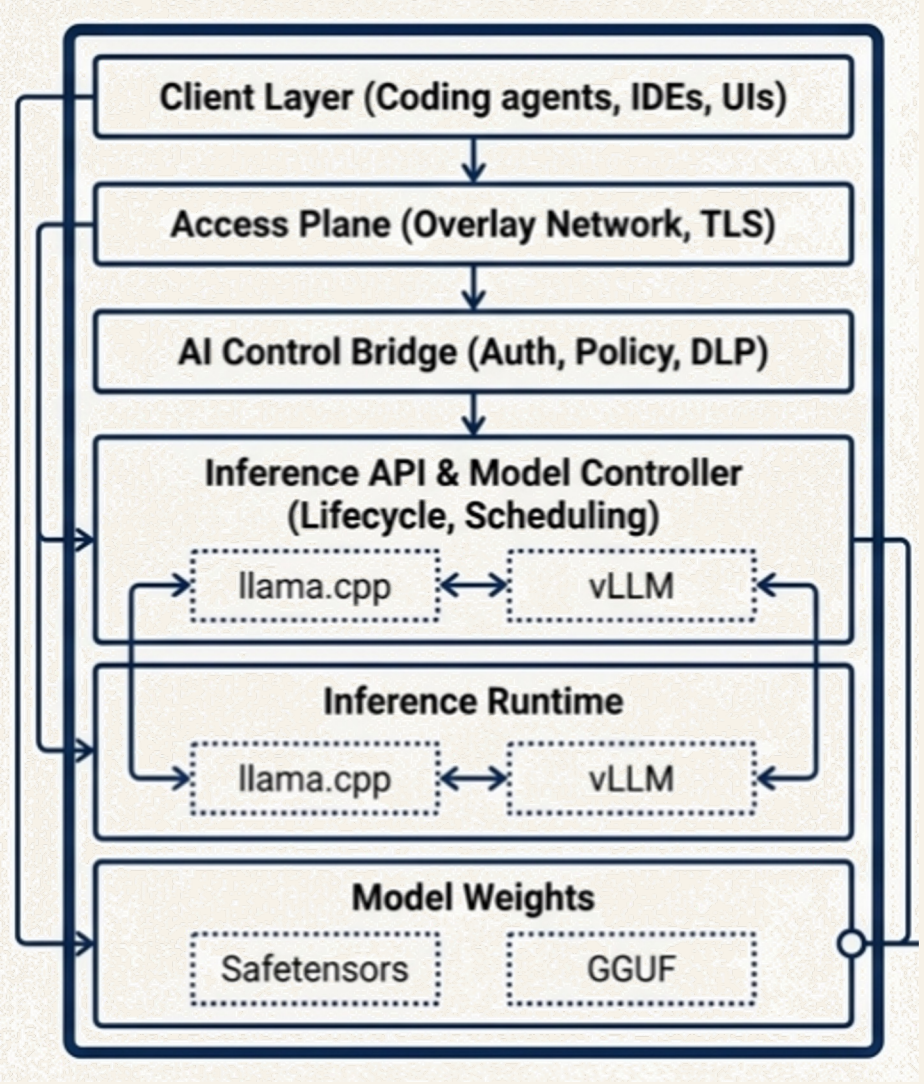
Things people forget

- Power Requirements
  - RTX Pro 5000 : 48GB : 300 Watts
  - RTX 5090 : 32 GB : 575 Watts

- Cooling Setup
  - Fans curve are generally configured to remain low on noise
  - GPU’s tend to run hot 70+ degrees Celsius hot

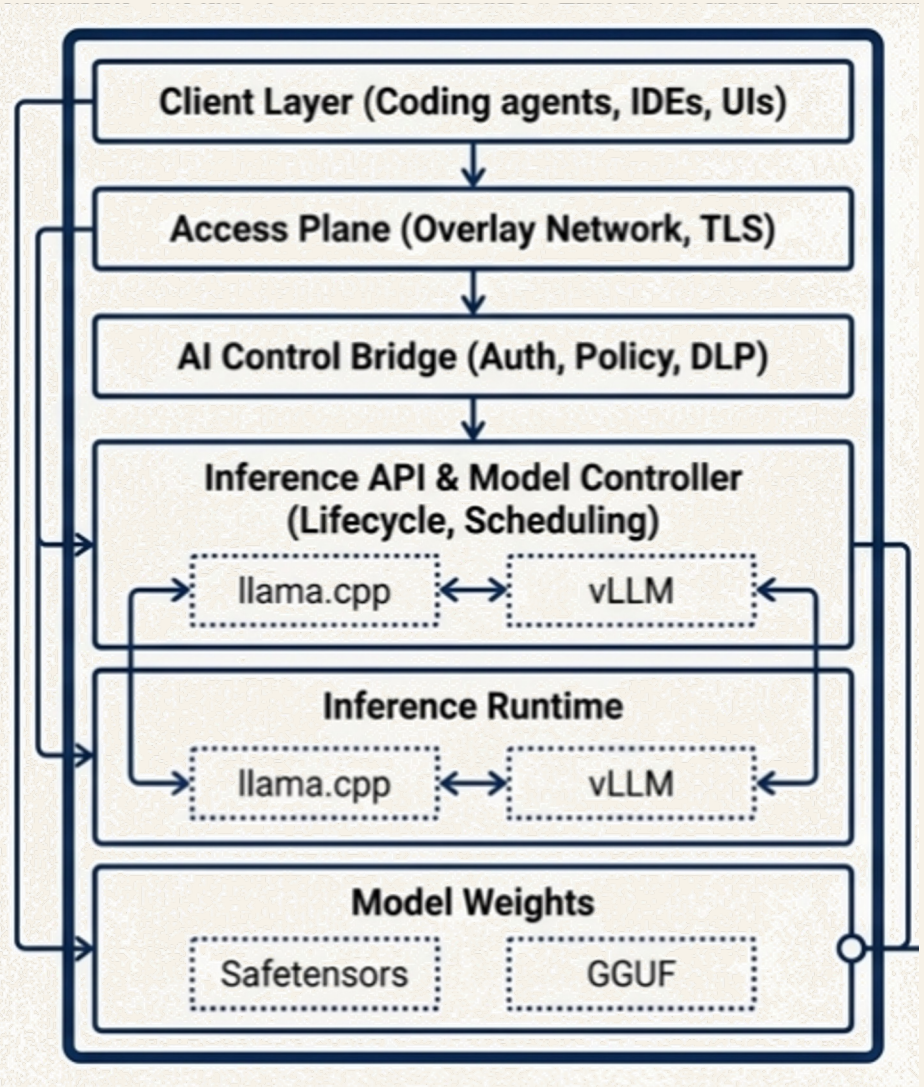
https://github.com/zmarty/nvidia-fan-control
https://github.com/exzile/intel-arc-pro-fan-control

Software Stack



Inference Gateway

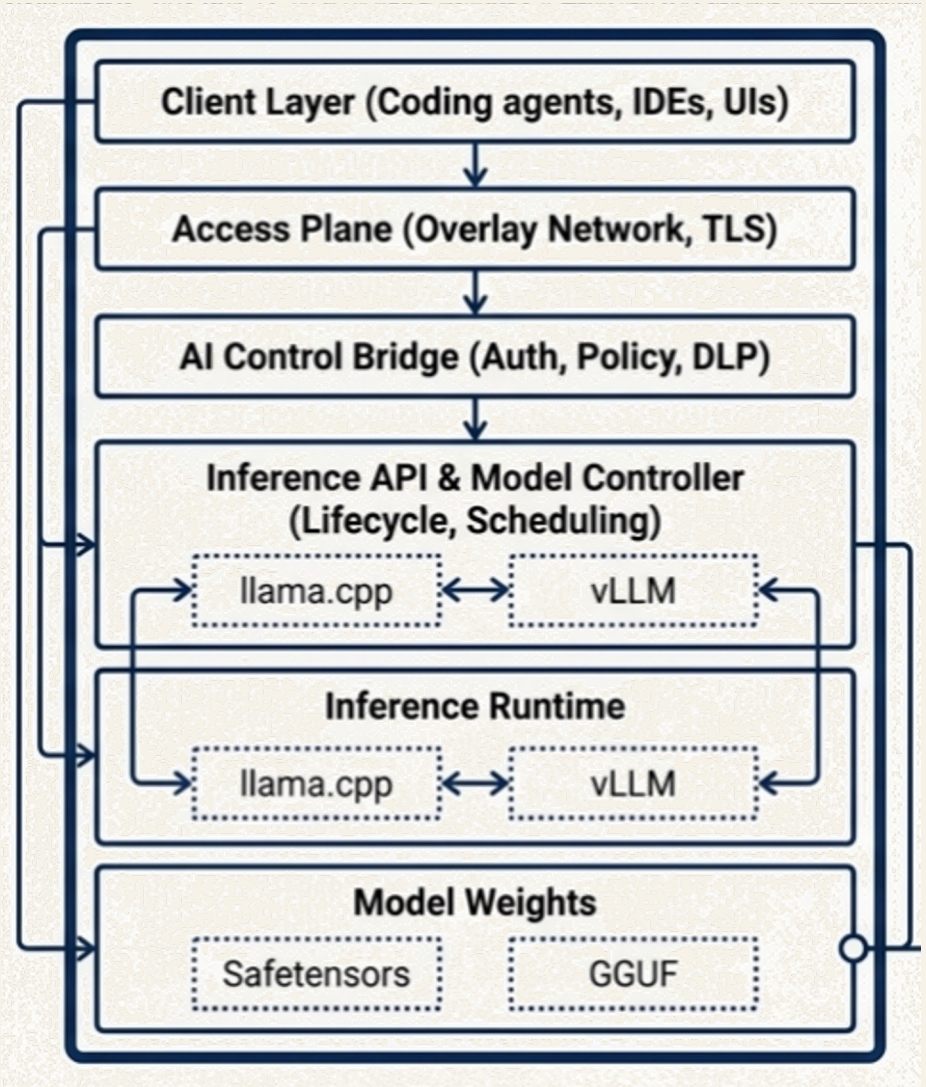
- API Translation Layer
- Co-ordinates with Inference Runtime
- Examples
  - Ollama
  - LM Studio
  - Llama-swap



https://github.com/mostlygeek/llama-swap
Forked : https://github.com/anantshri/llama-swap

Inference Runtime

- Llama.cpp (C runtime)
- vLLM (python)
- MLX (apple hardware)



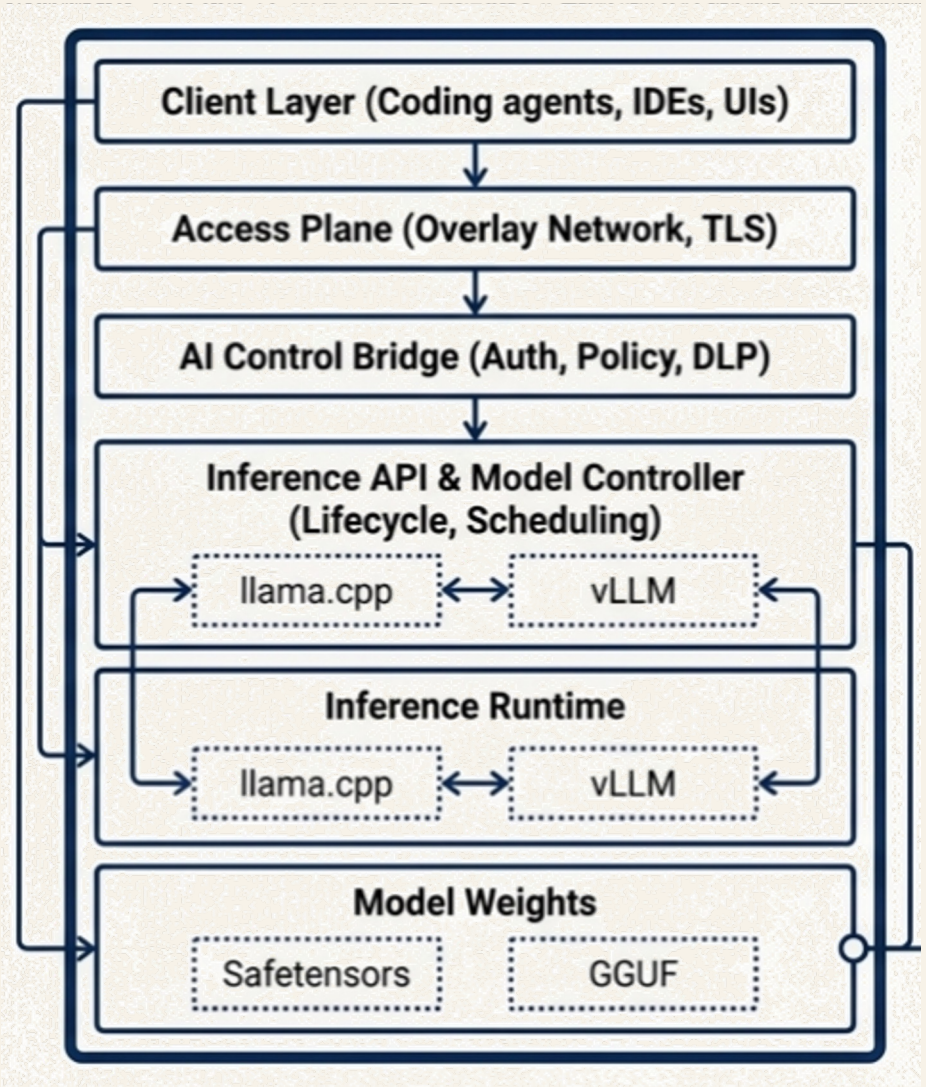
Remote Access

- NEVER open the port to internet

Thousands of Ollama Servers Publicly Accessible

A recent scan conducted by the Malware Patrol team revealed over 14,000 Ollama server instances publicly accessible on the Internet, opening the door to unauthorized use of the models and exploitation of known vulnerabilities. From a sample of 4400, we have found that the top Ollama versions include many outdated releases: 0.5.7 – 13% 0.5.10 – 11.5% 0.5.11 – 7.4% 0.9.0 – 7.0% 0.5.12 – 5.8% Other old versions like 0.6.2, 0.6.5, 0.6.8, and 0.7.0 also make up a significant share. Notably, only ~7% of scanned instances run the latest stable release.

- Provide Access via
  - Tailscale / headscale
  - Openvpn / wireguard
  - SSH Port forward : Poor man's VPN



https://www.malwarepatrol.net/over-14000-ollama-instances-exposed-to-the-internet-serious-security-implications/

Model Selection Criteria

- Prefer Dense models for Dedicated GPU's
- Prefer MoE for unified memories
  - MoE can also be preferred for Dedicated GPU when its larger size then VRAM and we do RAM offloading
- Download from official sources
- Check using model scanner

https://insights-db.paloaltonetworks.com/models

Cyfinoid

Selection Criterias

- Prefer
- Official Publishers
- Safetensors models
- GGUF model

- Avoid
- Pickle checkpoints
- Trust_remote_code=True
- Avoid ablitration / uncensored

Most uncensored models get lobotomized while freeing them and hallucinate a lot.

Cyfinoid

Operating System

- Most of the deep work is done in linux so that is a preferred OS.
- Unless you are using apple hardware then only good option is default OSX.
- Avoid windows if you can, too much trouble for too little reward

Cyfinoid

How to maintain the Stack

- OS : unattended upgrades : cron based updates
- Firewall based isolation
- Own the opensource stack : consider them dead on arrival
- Focus on stability and not the latest feature

Make it boring : make it stable

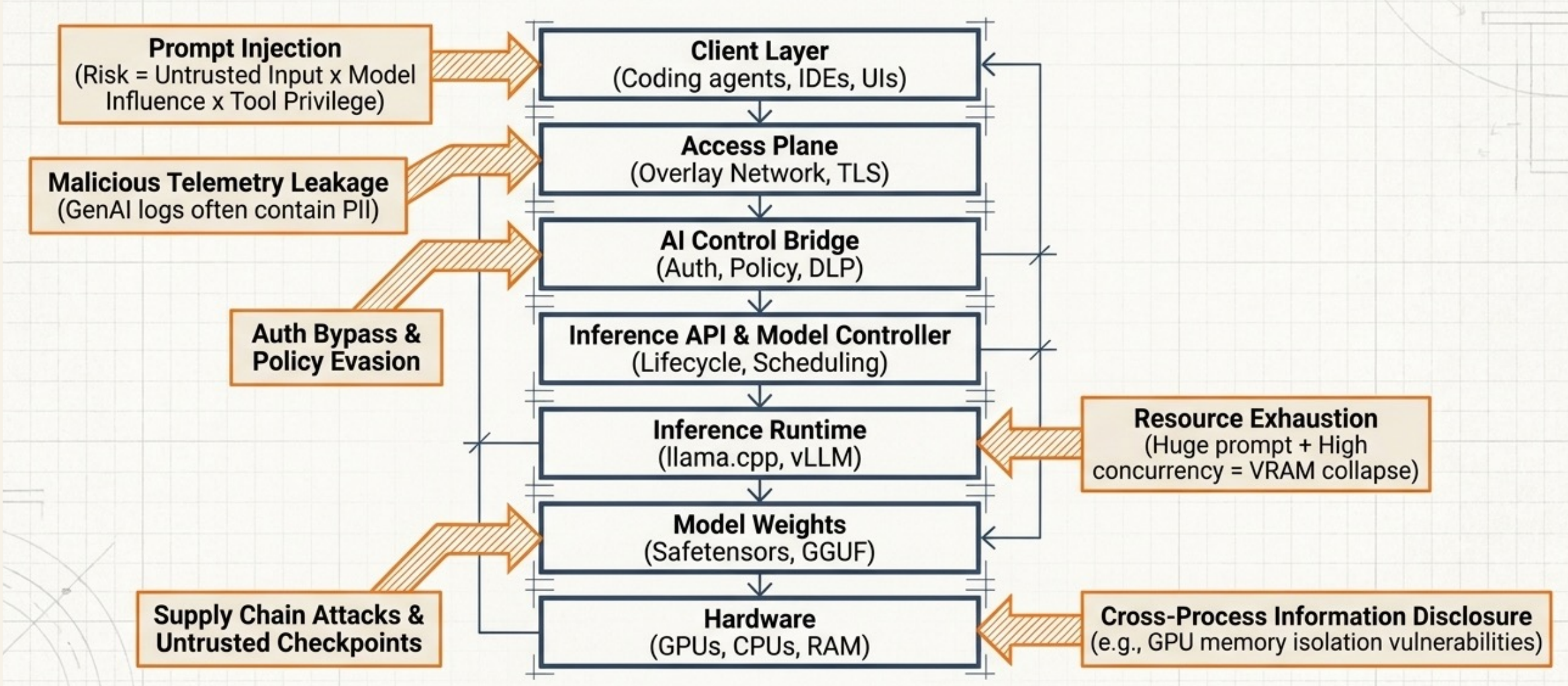
Cyfinoid

Security

you build it, you own it.

if you dont secure it: someone else will own it.

Threats


Prompt Injection
(Risk = Untrusted Input x Model Influence x Tool Privilege)
Malicious Telemetry Leakage
(GenAI logs often contain PII)
Auth Bypass & Policy Evasion
Supply Chain Attacks & Untrusted Checkpoints
Client Layer
(Coding agents, IDEs, UIs)
Access Plane
(Overlay Network, TLS)
AI Control Bridge
(Auth, Policy, DLP)
Inference API & Model Controller
(Lifecycle, Scheduling)
Inference Runtime
(llama.cpp, vLLM)
Model Weights
(Safetensors, GGUF)
Hardware
(GPUs, CPUs, RAM)
Resource Exhaustion
(Huge prompt + High concurrency = VRAM collapse)
Cross-Process Information Disclosure
(e.g., GPU memory isolation vulnerabilities)

Cyfinoid

Access Control

- No public access
- Gatekeep via VPN / tailscale
- Add API Key based access as last control
- Protect admin interfaces

Zero day attacks

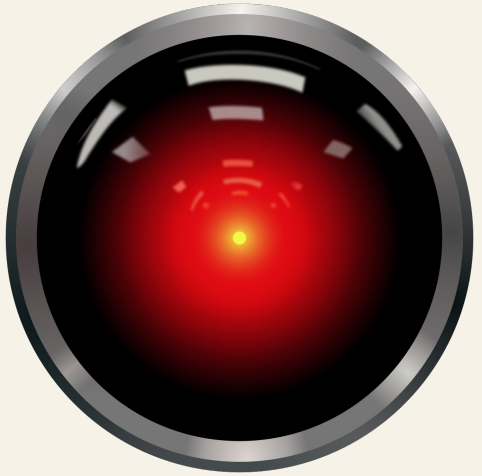
- Inventory is first step to salvation
- Scan your own Infra
- Let your LLM find bugs for you before someone else's llm finds it for them

https://cyfinoid.com/appsec-in-the-new-security-cost-model/
https://cyfinoid.com/a-pragmatic-guide-to-being-mythos-ready/

Cyfinoid

LLM Going Rogue

- OLD APPROACH : airgap / firewall the machine



- Reality
  - LLM's don't make outside connection
  - Agents / harness make it
  - Mostly agents are not sitting in same box

- How to Protect
  - Firewall the agent
  - Validate all tool calls
  - Log everything

Cyfinoid

Overlooked : GPU bugs

- GPU is also a software stack and hence a source of bugs
- GPU bugs can
  - Leak data
  - Cause crash
  - Damage hardware
- Protection
  - Keep a tab on public update channels
  - Reduce the blast radius. Keep them isolated

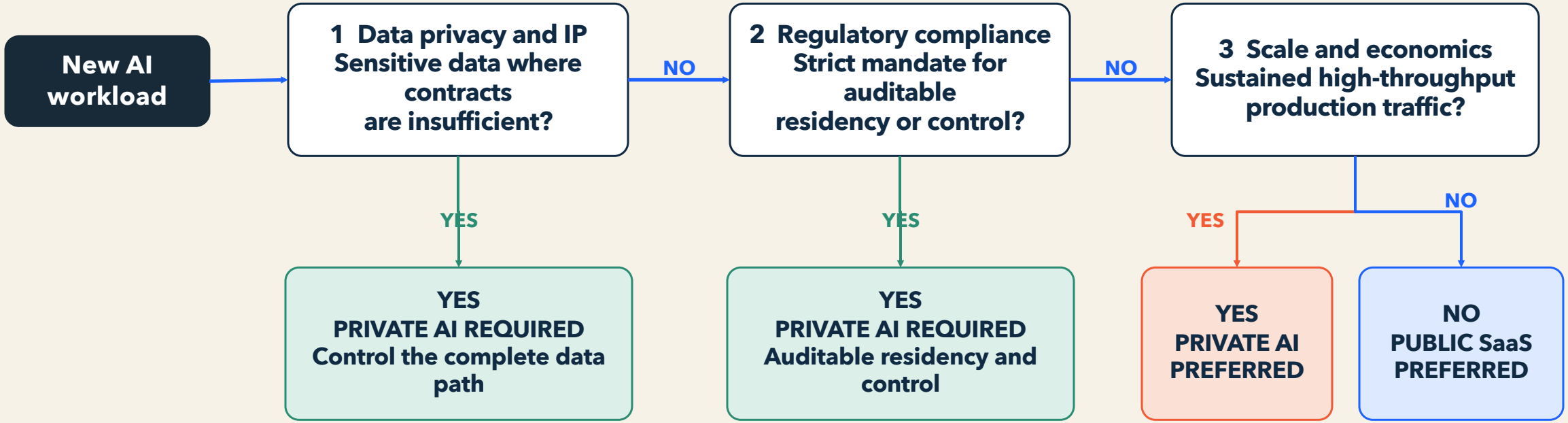
https://reducetheattacksurface.com/

Overlooked : so Obvious you ignore

- Auth Bypass
- PII Leakage in Logs
- Resource Exhaustion

- Solutions are age old solutions nothing LLM specific

Question : Should we self host



Or

- You simply want full control over all the elements of the process

Thanks for listening