⚙️ vLLM vs Ollama

Choosing the Right LLM Inference for Real Systems

🧠 AI is powerful

🏗️ Systems make it usable

🚀 Inference is an engineering decision

The Real Question

❌ **Wrong Question**

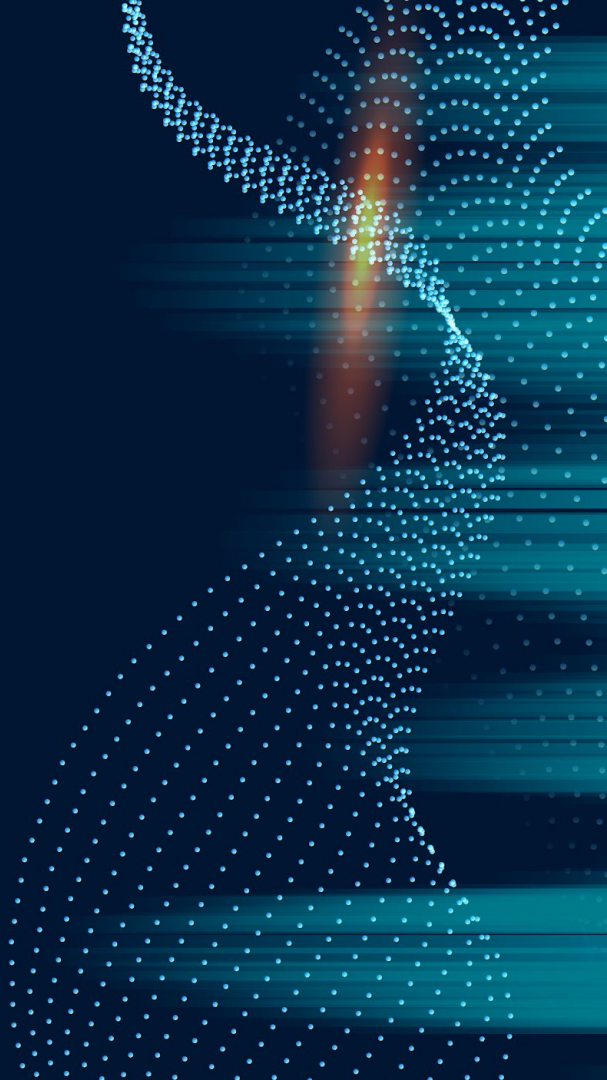
“Which LLM tool is better?”

✅ **Right Question**

“Which inference pattern fits my system?”

- Local vs Production
- Single-user vs Multi-tenant
- Low latency vs High throughput
- Cost vs Performance

👉 **Inference is infrastructure**



Ollama 🧠

Developer-First Inference

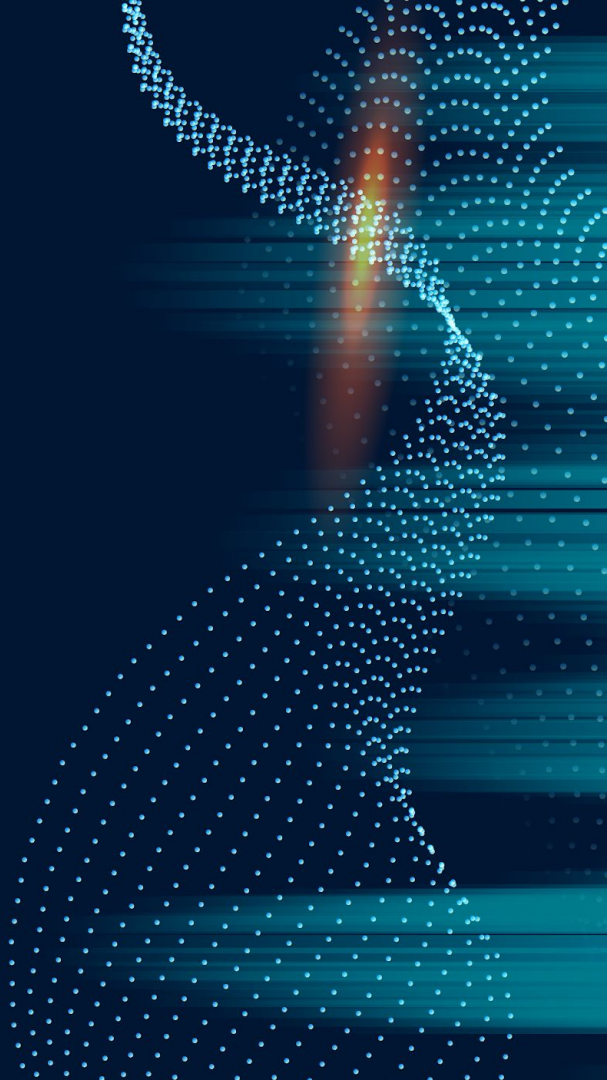
✅ Best for:

- Local development & demos 🧪
- Fast iteration & experimentation ⚡
- Single-user or small teams 👤

🛠️ Characteristics:

- Simple CLI & API
- Minimal setup
- Runs locally (CPU/GPU)
- Low operational overhead

💡 Think: **Developer productivity**



vLLM ⚙️

Production-Grade Inference

✅ Best for:

- High-throughput systems 🚀
- Multi-tenant workloads 🏢
- GPU-optimized inference 🖥️
- Platform & API layers 🌐

🛠️ Characteristics:

- Efficient batching
- PagedAttention (GPU memory optimization)
- High concurrency
- Designed for scale

💡 Think: **Platform reliability & performance**

Key Engineering Differences

🧩 Systems View

Area	Ollama	vLLM
Target	Dev / Local	Production
Concurrency	Low	High
GPU Efficiency	Basic	Advanced
Ops Overhead	Minimal	Higher
Scaling	Manual	Built-in patterns

👉 Same models. **Different systems.**

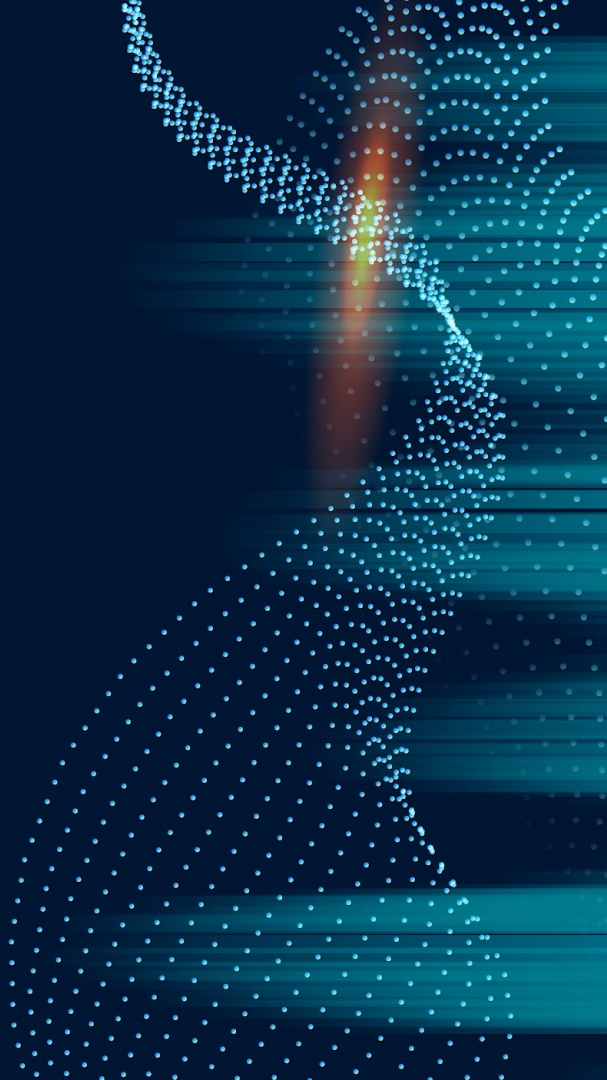
How Systems Engineers Should Think

🏗️ **Inference Is a Systems Decision**

Ask these questions:

- 🔄 How many concurrent users?
- ⏱️ Latency vs throughput requirements?
- 💰 GPU cost constraints?
- 🔐 Multi-tenant isolation?
- 📈 Future scaling needs?

🧠 Models change.
🏗️ Architecture stays.



When to Migrate: Ollama ➜ vLLM

🔁 The Inflection Point

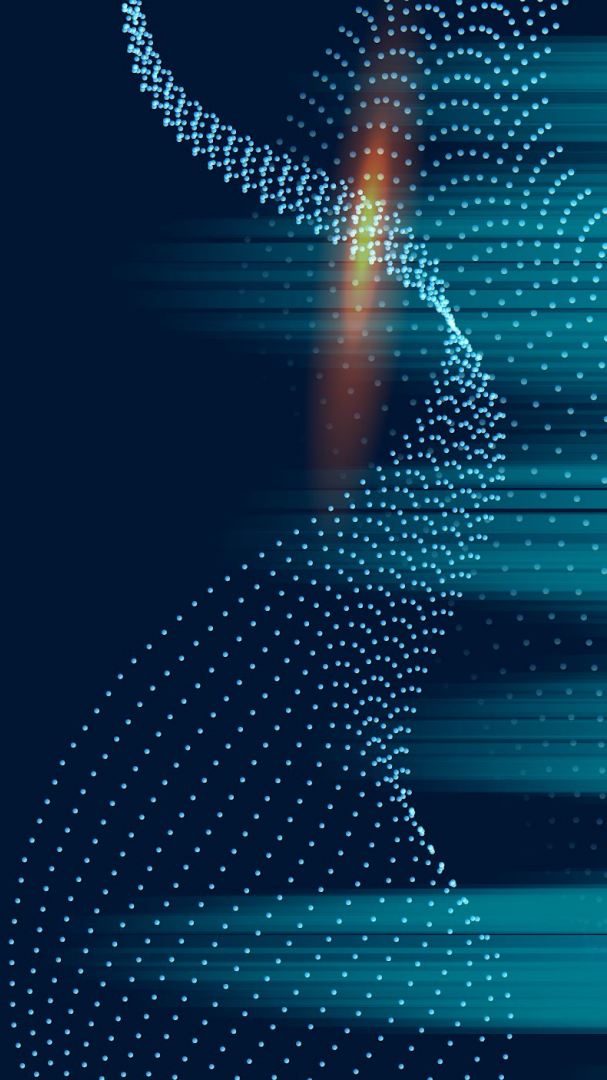
You should start thinking about **vLLM** when:

🚨 Traffic increases

- Multiple users hitting the model concurrently
- API-backed applications, not just local usage

🚨 Latency becomes inconsistent

- Queueing delays under load
- GPU memory pressure



When to Migrate: Ollama ➜ vLLM

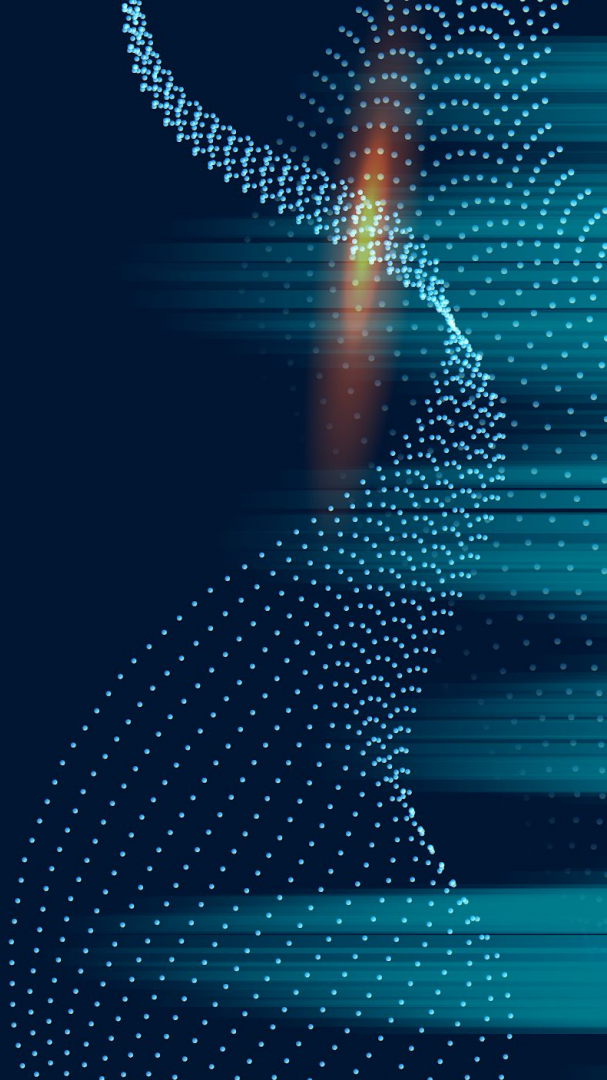
🚨 **Costs start to matter**

- Inefficient GPU utilization
- Need for batching and memory optimization

🚨 **Platform responsibility appears**

- SLAs, SLOs, uptime guarantees
- Multi-tenant isolation

👉 **Ollama = build fast**
👉 **vLLM = scale responsibly**



Docker Deployment Patterns 🐳

Container-First Inference

Ollama with Docker

✅ Best for:

- Local dev environments
- Demos & labs
- Internal tools

🛠️ Pattern:

- Docker / Docker Compose
- Single container per model
- Simple volume mounts

vLLM with Docker

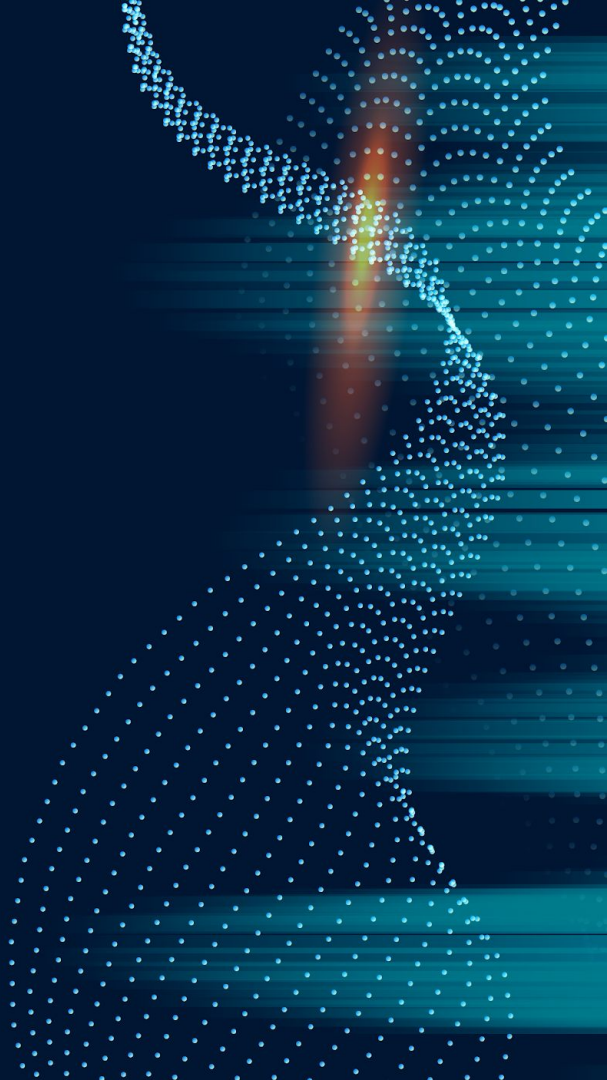
✅ Best for:

- API services
- GPU-backed containers
- CI/CD-driven deployments

🛠️ Pattern:

- GPU-enabled containers
- Explicit resource limits
- Reverse proxy (NGINX / API Gateway)

👉 Docker is the **starting point for both**



Kubernetes Patterns ☸️

- **Production AI Systems**
- **vLLM on Kubernetes**

🏗️ Designed for:

- Horizontal scaling
- Multi-tenant workloads
- Enterprise environments

🔧 Common components:

- GPU node pools
- HPA / custom autoscaling
- Resource quotas
- Ingress / API Gateway
- Observability (metrics + logs)

Production AI Systems

vLLM on Kubernetes

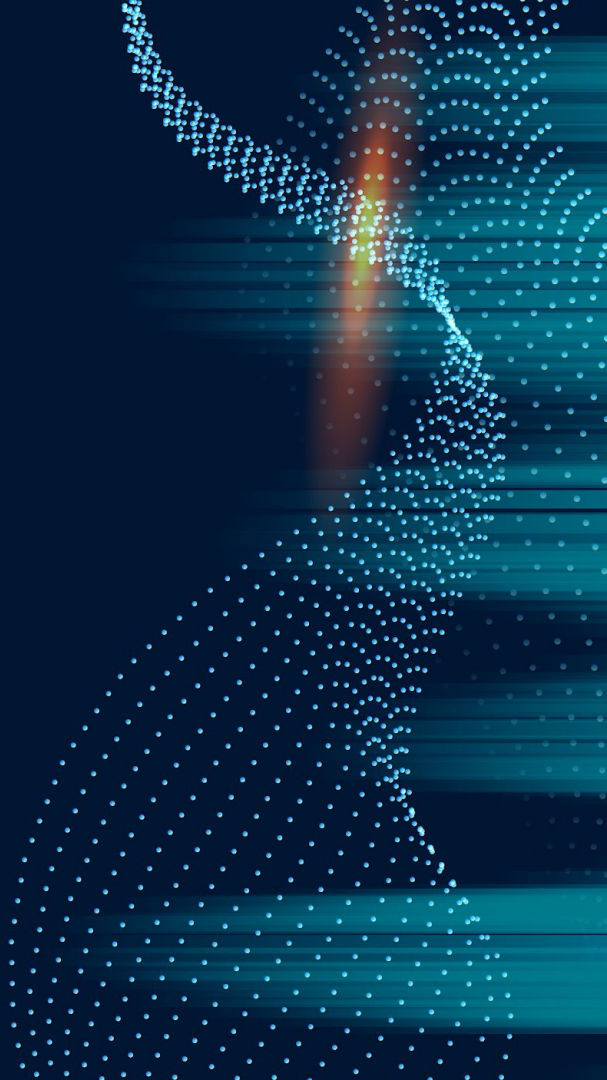
📈 Benefits:

- Predictable scaling
- Cost control
- Fault isolation

👉 Kubernetes is where **AI becomes a platform**

🎯 **Final Note for the Deck**

> “AI maturity is not about models — it’s about systems.”

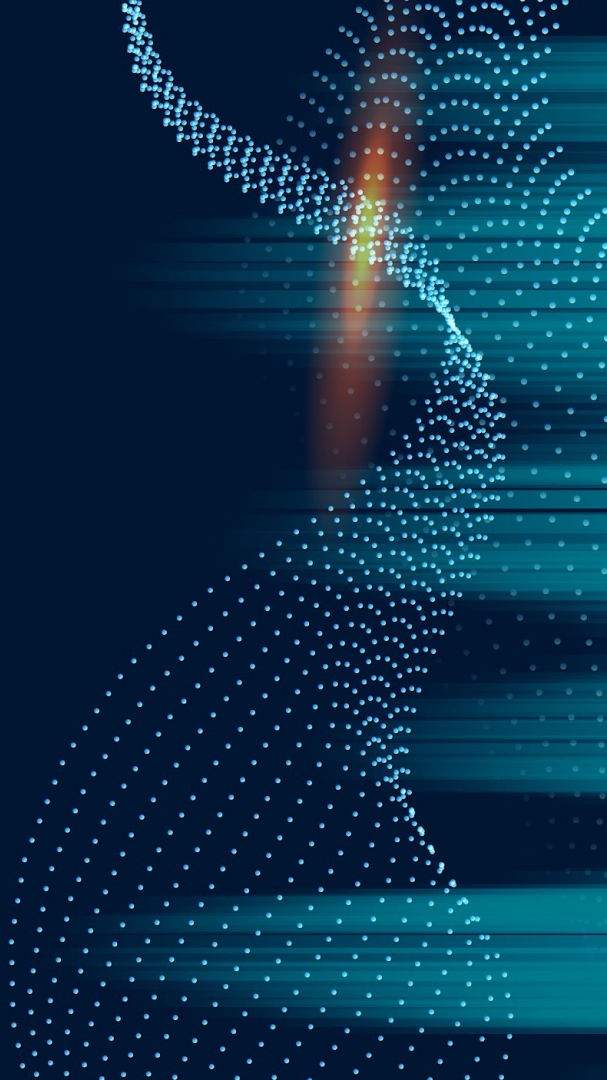


🎯 Build Smart. Scale Intentionally.

✔️ Use **Ollama** to learn, prototype, and move fast
✔️ Use **vLLM** when reliability, scale, and efficiency matter
✔️ Match inference to workload maturity

🚀 AI systems succeed **only when infrastructure is done right**

💬 What inference patterns are you seeing in real systems?



Questions? Reach out to us ٩(◕‿◕。)۶

📌 Website: https://remoder.com

📘 LinkedIn (Company): https://www.linkedin.com/company/remoder

👨‍💻 LinkedIn (Sanjars): https://www.linkedin.com/in/sanjars/

📹 YouTube: https://www.youtube.com/@remoder-inc

📧Contact: hello@remoder.com OR Directly Message Me:
https://www.linkedin.com/in/sanjars/

🚀 Want the Full Walkthrough or Hands On Labs?

This entire project — step-by-step, production-ready, with diagrams, videos and code — is covered inside our **AI Systems Engineering - mentored path.**

More information: https://remoder.com/%F0%9F%A7%A0-ai-systems-engineer