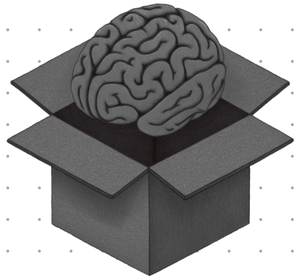
FOSS UNITED

TEAM VANTABLACK.

Observing LLM's with FOSS

Solving the **"Black Box"** Problem



OpenTelemetry

JAEGER

opik

OpenLLMetry

./sarvatarshan --about-me

sarvatarshansankar20 — -zsh — 135x35

```
Last login: Sat Dec 27 10:11:12 on ttys010
sarvatarshan@foss-meetup:~$ whoami
> Name : Sarvatarshan Sankar
> Role : 3rd Year CSBS Undergrad @ KPRIET
> Badge : Google Student Ambassador | IEEE XTREME 19.0 Ambassador @ KPRIET
sarvatarshan@foss-meetup:~$ checl-passion --tags
> Found 2 results:
 1.Machine Learning
 2.Generative AI
 3.Agentic AI
sarvatarshan@foss-meetup:~$ cat project_status.log
[CRITICAL] LLM Cost exceeded: ₹560!
sarvatarshan@foss-meetup:~$ meetup --status
[ALERT] First-time Speaker Mode Active! Excitement Level: 110%
```

“The ₹560 Lesson”

A True Story of APIs, Hackathons, and Unexpected Bills

Prabakaran

Sharvesh

Mentor: Mr. Santhosh N C

(aka) #KongunaatuKubestronaut

API not working	
Billing enabled, APIs work	
₹560 bill arrives	

Trust nobody, not even your API costs

“Me checking my GCP bill”

"Enna koduma... API costs idhu 😭"

❌ No idea how much each question cost
❌ Couldn't see response times
❌ No clue about token usage
❌ When it failed, I just... didn't know

"End-to-End Execution Context"

✅ Real-time cost tracking
✅ Latency monitoring
✅ Token usage visible
✅ Error detection

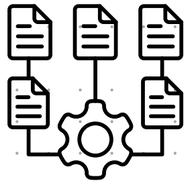
LLMs Today - The Scale

Mind-Boggling Numbers

~$200M

Cost to train ONE frontier model

Training a frontier model today (100–500B parameters) can cost in compute alone using 30K GPUs



20 MW

Power consumption (= small town!)

Power consumption to run AI labs with GPUs

$200B+

Global LLM market by 2030

OpenAI, Anthropic, Google, Meta, and xAI dominate 60–70% of global LLM value creation

And YOU need to monitor this beast?

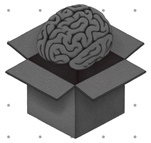
📰 Real Headlines (Not Kidding):

"Startup burns $50K in one month on GPT-4 testing that never stopped”

"Medical chatbot gives wrong advice. No way to track what it said"

"API bill 10x higher than expected. Nobody was watching"

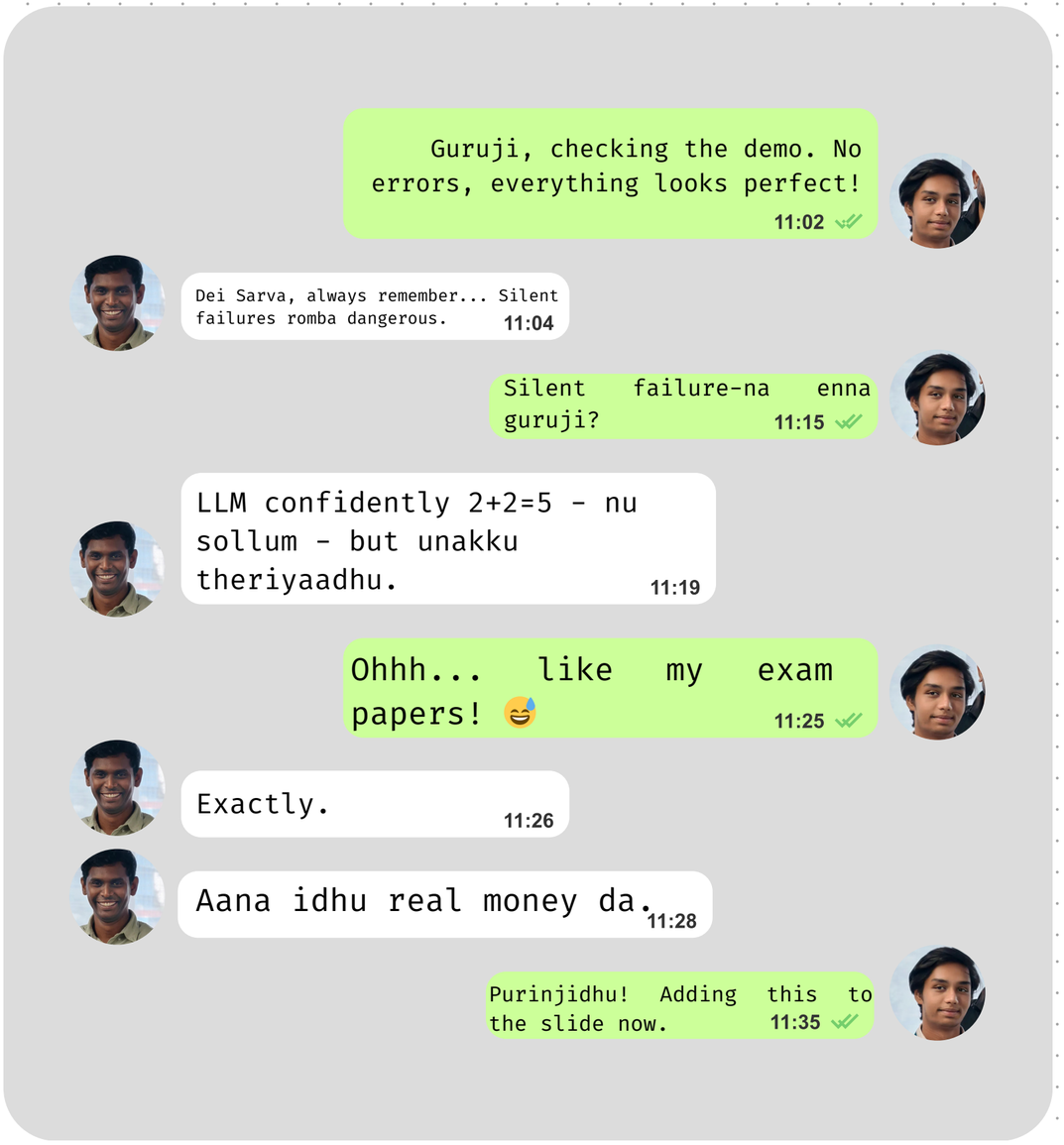
Naa oru thadavai solliten... **Monitor your LLMs**...



LLM = Black Box & My Mentor's Warning

What happens inside an LLM call?

How many tokens? → Theriyaadhu
How much cost? → Theriyaadhu
How long it took? → Theriyaadhu
Did it hallucinate? → Theriyaadhu
Quality good or bad? → Theriyaadhu

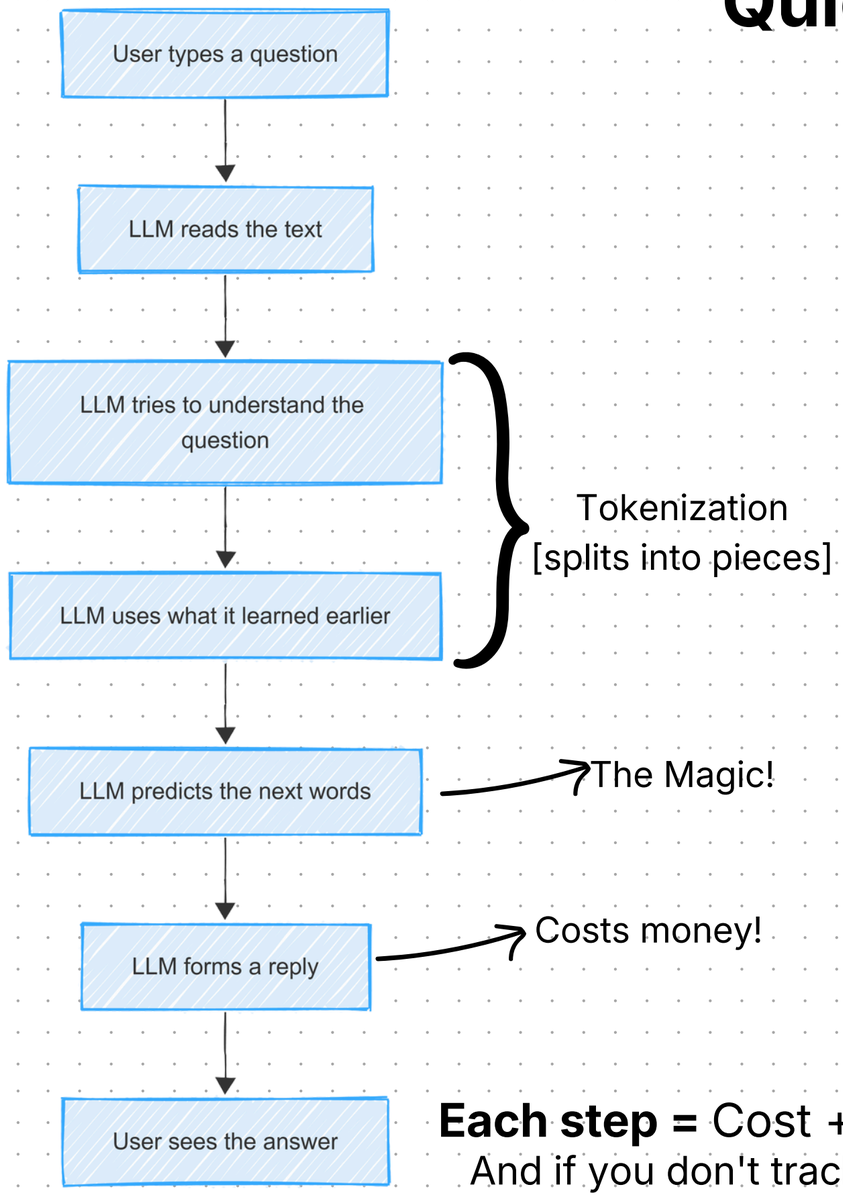


The Core Problem

The Problem in One Line:

"You can't fix what you can't see."
“You can't optimize what you can't measure.”
“You can't trust what you can't trace."

Quick LLM 101 & Tokens



Each step = Cost + Time + **Tokens**

And if you don't track it... RIP wallet 💸

Tokens = Money

"Hello" = 1 token = *₹0.001*

"Internationalization" = 8 tokens = *₹0.008*

Your question = 50 tokens = *₹0.05*

LLM response = 500 tokens = *₹0.50*

One chat = *₹2-₹20*

1000 users/day = *₹60,000/month!*

OBSERVABILITY

Style-ah solve panalam! 😎

Like Rajini entering in slow-mo...
Observability enters your codebase

What it does:

Traces & Spans (*Enna nadandhuchu - step by step*)

Metrics (*Evlo fast? Evlo cost?*)

Logs & Events (*Enga error vandhuchu?*)

Evaluations / Evals (*Badhil correct-a? Quality ok-va?*)

Annotations & Prompts (*Endha prompt nalla work aachu?*)

Lineage & Versioning (*Pazhaya version enga? Yaaru maathuna?*)

The FOSS Hero Squad

Jaeger
Never misses a trace.
Pinpoint accuracy.

OpenLLMetry
The God Mode

Opik
The Powerhouse

OpenTelemetry
The Industry Standard. The Leader everyone follows.

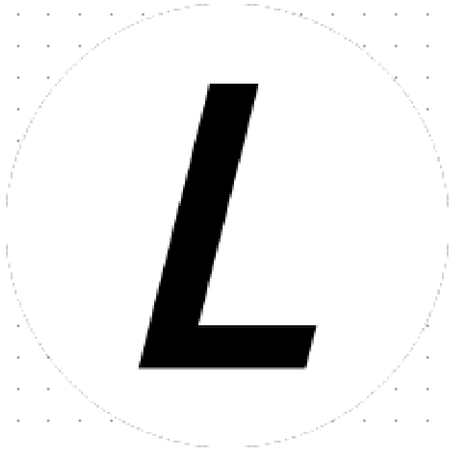
Langtrace
Futuristic & Fast. Traces everything in style

All FREE. All OPEN SOURCE.

Tool #1 - Langtrace

The Speedster

Focuses mainly on traces - LLM Calls, Vector DB queries, Framework workflows



Setup Time: 2 minutes

Lines of Code: 2

Difficulty: Beginner

What it does:

- Auto-captures all LLM calls
- Shows costs & tokens automatically
- Beautiful dashboard
- Zero manual work

Follows parent-child relationships
Crucial for RAG systems, since it also traces Embedding generation, Vector search, Document retrieval

Best for: "I need visibility NOW"

Tool #2 - OpenTelemetry

Industry Standard

A common standard to collect system data

Setup Time: 30 minutes

Lines of Code: More

Difficulty: Intermediate

Best for: *"I need full control"*

What it does:

- Vendor-neutral (no lock-in)
- Works with ANY backend
- Complete customization
- Enterprise-grade

Integrated Ecosystem:

Tool #3 - OpenLLMetry

The Hybrid

Non-intrusive instrumentation

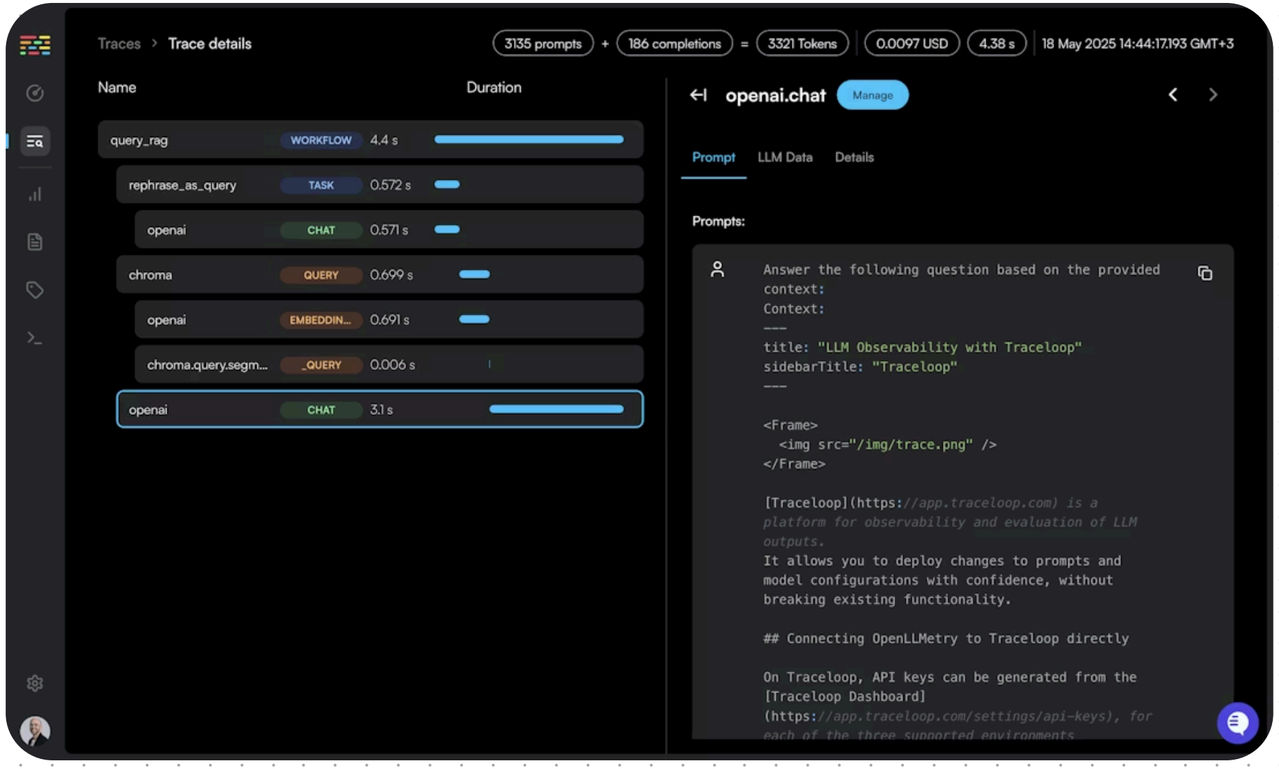
OpenLLMetry

Thin extension layer, that works on top of OpenTelemetry
But with LLM-specific auto-instrumentation

What it does:

- Adds extra spans
- Adds standard LLM fields
- Follows OpenTelemetry rules

Best for: *"I want automatic + flexible"*



Tool #4 - Opik

The Evaluator

Not just monitoring - EVALUATION

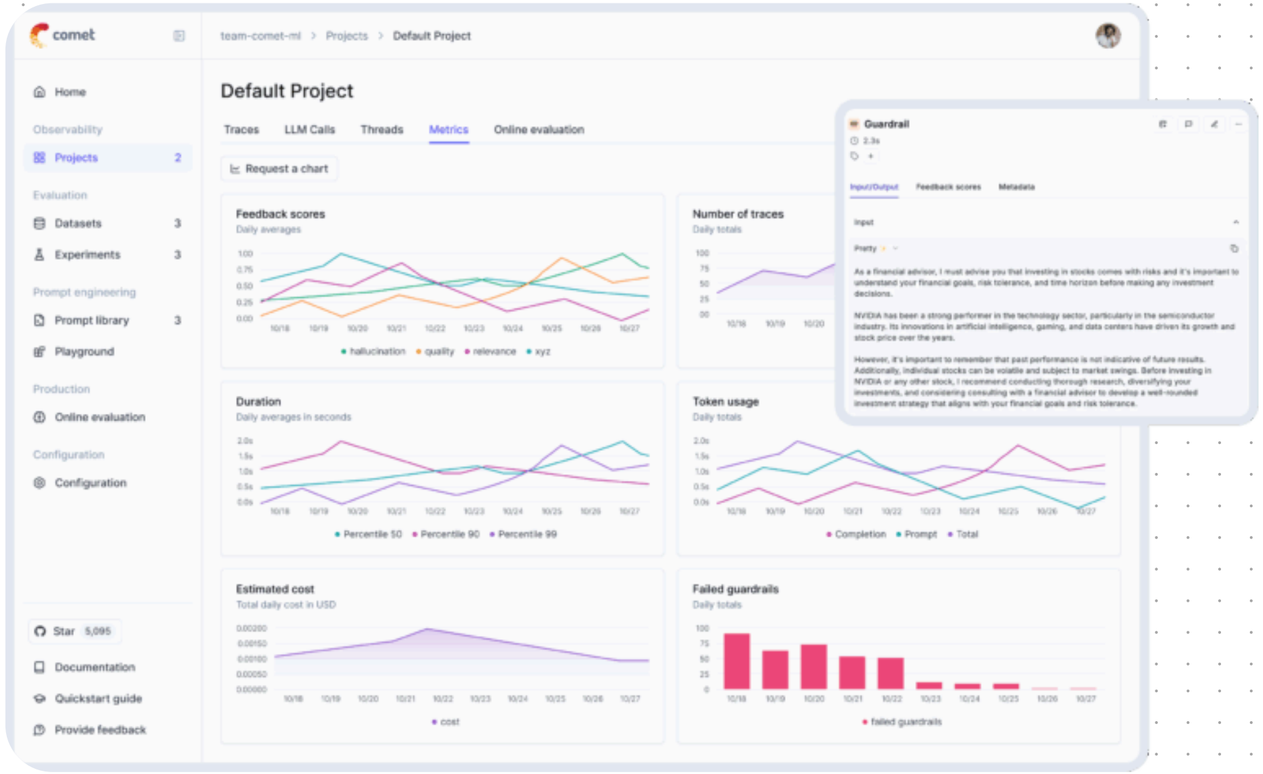
Used to observe, evaluate & improve applications

“Opik is like a coach - it watches your AI, tells how to improve”

What it does:

- Helps with : Trace logging, evaluations, automation to improve prompts
- Agentic flow support

Best for: *"I need to test quality"*



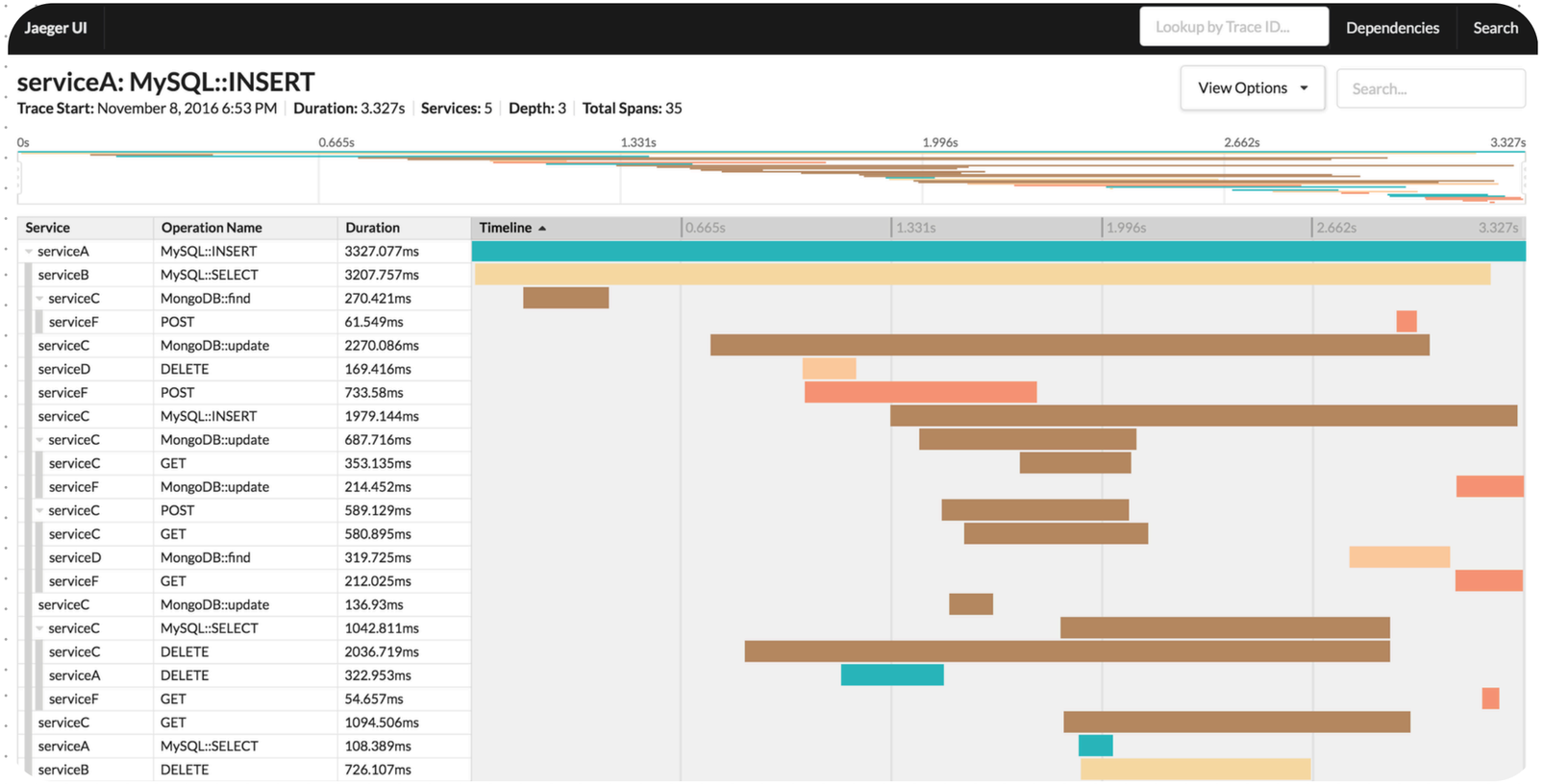
Tool #5 - Jaeger

The Visualizer

Makes traces BEAUTIFUL

- Timeline view of requests
- See exactly where time is spent
- Debug like a pro

Best for: *"Show me what's happening"*



COMPARISON

Tool	Focus	LLM Tracing (Prompts/Responses)	Token & Cost Tracking	Evaluation & QA	Production Monitoring	OTEL Compatible	UI Included
Langtrace	OTEL-native LLM tracing	✅	✅	✅	✅	✅ Native	✅
OTEL + Prometheus + Jaeger + Grafana	Infra + app observability	⚠️ Manual instrumentation	⚠️ Manual	❌	✅(Very Mature)	✅ Native	✅
OpenLLMetry	LLM telemetry via OpenTelemetry	✅ (as OTEL spans)	✅	❌	✅ (via Grafana/ Jaeger etc.)	✅ Native	❌ (uses external OTEL UI)
Opik by Comet	LLM observability + evaluation + safety	✅	✅	✅ (LLM-as-judge, testing)	✅	❌	✅

The Magic Moment

Padam padam padam... Demo time! 📽️

Theory mudinjichu.
Ippo... **LIVE DEMO TIME!**
Readyyyy???

What you'll see:
Problem *(no observability)*
Quick Fix *(Langtrace-la 2 lines)*
Foundation *(OpenTelemetry-oda power)*
Mass Comparison *(ella tools-um ore time-la!)*

Mind = Blown

What We Just Saw:

› *ONE chatbot*
› *FOUR frameworks watching*
› *Same questions*
› *Different perspectives*

All tracked. All visible. All FREE.

What Observability Gives You:

💰 ***Cost Control*** → *Catch expensive queries early*
⚡ ***Performance*** → *Optimize slow calls*
🛡️ ***Reliability*** → *Detect failures fast*
📈 ***Quality*** → *Track accuracy over time*
🎯 ***Compliance*** → *Audit trail for regulations*

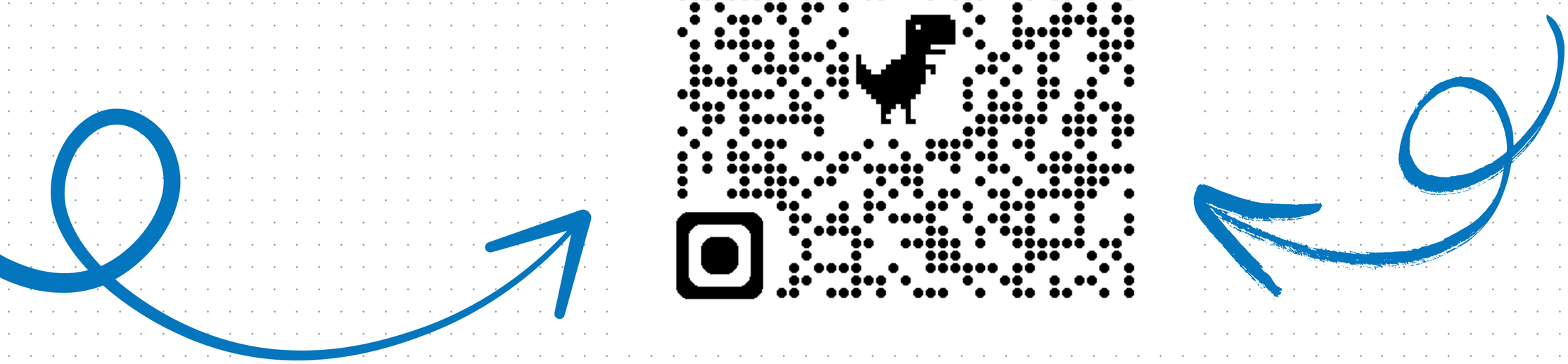
3 Things I Learned the Hard Way:

1. **Start Simple**
Begin with Langtrace, scale later

2. **Measure Everything**
If it costs money, track it

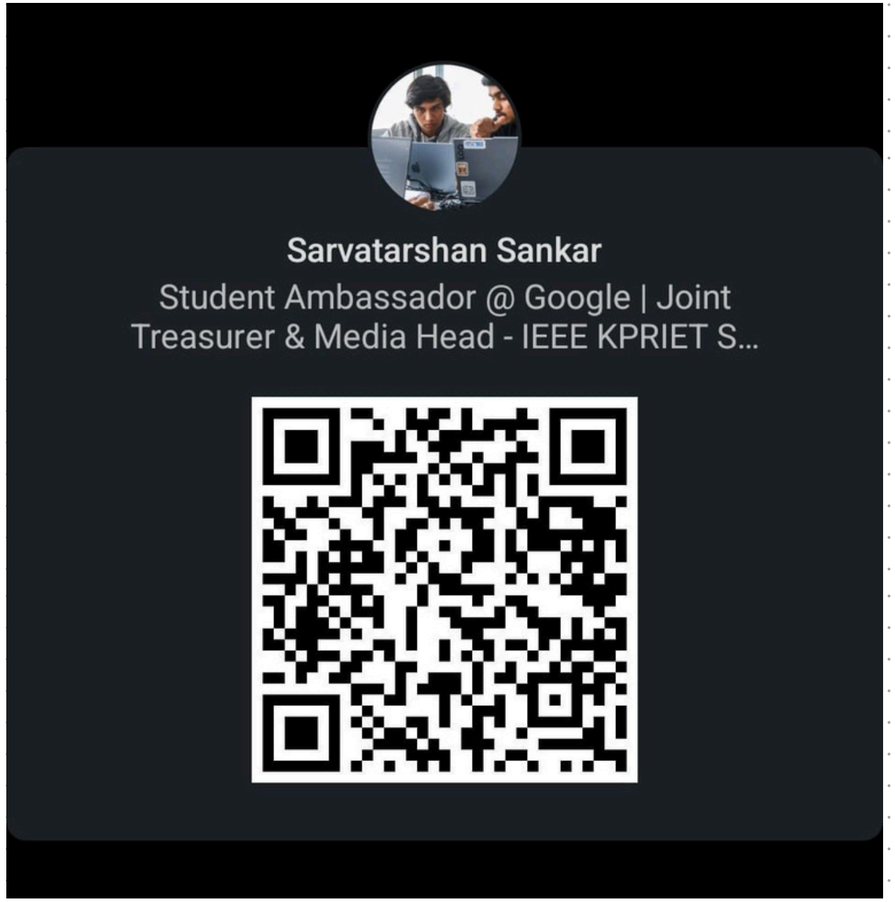
3. **Open Source Rocks**
You don't need expensive tools

Want to Try?



Free. Open Source. Unga use-ku ready.

Thank you!



Connect with me on LinkedIn

Production Readiness Checklist:

✅ Cost Alerts (set budget limits)
✅ Error Tracking (catch failures)
✅ Latency Monitoring (speed matters)
✅ Quality Metrics (accuracy checks)
✅ Audit Logs (compliance ready)

All possible with these FOSS tools!

📚 **Learn More:**

Langtrace: *docs.langtrace.ai*
OpenTelemetry: *opentelemetry.io*
Jaeger: *jaegertracing.io*
Opik: *comet.com/opik*

My Demo Repo:
github.com/sarva-20/LLM-Observability-FOSS

FOSS United:
fossunited.org