

PREMIUM



PREMIUM PARTNER



TECHNOLOGY PARTNER



DIAMOND



PLATINUM



GOLD



SILVER



BRONZE



FUTURE MAKER SILVER



FUTURE MAKER BRONZE



MEDIA PARTNER



BUILD ROBUST LLM-POWERED APPS

Waldek Mastykarz · Garry Trinder



WHO HERE HAS BUILT SOMETHING WITH AN LLM?

Getting it to do something cool is the easy part. It's everything around it that's hard.



HOW YOU END UP WITH THE MOST EXPENSIVE MODEL IN EVERY CALL

- Your coding agent picks the biggest model it knows
- Someone on social says "just use the best model"
- A tutorial confirms it
- **Which model, when to call it, what to do when it fails** — nobody talks about those



Our deployments were blocked — not because of a bug, but because we'd used up our LLM allowance for the day



THE HIERARCHY OF SOLUTIONS

Large LLM (GPT-5.5, Opus)	← Complex reasoning
Small/Fast LLM (GPT-4o-mini, Phi)	← Classification, extraction
No LLM at all (code, regex)	← Pattern matching, lookups, templates
Mock / stub (any string)	← You don't care about the output



DEV PROXY

An open-source API simulator that sits between your app and the API

- Intercepts HTTP traffic — your app doesn't know it's there
- No code changes needed
- Works with any tech stack



DEMO: THE BURN

10 support tickets · 4-step pipeline · every step hits GPT-5.4



50,000 TOKENS – TEN TICKETS, ONE RUN

	Tokens/day	Team of 5, one month
You, 20 runs	1,000,000	100,000,000
With a coding agent, 50 runs	2,500,000	250,000,000

That's before a single user touches the app



50,000 → 0

Same app. Same code. One config change.



Your machine

- API always responds
- Under the rate limit
- Clean answers
- Fast responses

Reality

- API goes down at 2 PM
- You hit the rate limit
- Model hallucinates
- 8-second responses



DEMO: THE LIE

Three failures, escalating



API SAYS NO — APP FALLS OVER

429 Too Many Requests. Raw error. No handling.



200 OK — THE CONTENT IS HALLUCINATED

The app renders it as truth



SAME GARBAGE. ONE VERSION SHIPS IT. THE OTHER CATCHES IT.

The only difference: we tested for it during development



THIS ISN'T NEW

Retry. Fallback. Validation. Caching. Circuit breakers.

Integration engineering. We've known how to do this for decades.

We just forgot because LLMs feel like magic.



REMEMBER THE CI/CD STORY?

Now we use Dev Proxy from the start — local models, failure simulation, token tracking before we push

Our quota stays intact. Our pipeline stays green.



FIVE THINGS TO DO MONDAY MORNING

1. **Match the tool to the task** — mock → no model → small model → large model
2. **Mock when you don't care** — half your dev time isn't about LLM output
3. **Develop against local models** — zero tokens, zero code changes
4. **Simulate failures on your machine** — before reality breaks it for you
5. **Track every token from day one** — stop guessing, start measuring



**BUILD INTENTIONALLY.
TEST FOR THE UGLY.
MEASURE FROM DAY ONE.**

