Kabir Moghe '26

AI Engineer Europe
Spring 2026

This spring, I spent a few days at the AI Engineer conference held in London, where startups, engineers, and larger tech- and AI-focused companies meet to learn about the state of the industry, present their own developments, and network. The conference offers aspiring engineers like myself the opportunity to engage with a growing crowd of builders that are eager to learn about maximizing their output as developers. In parallel, the conference's events present updates on where AI itself (including models, applications, and related paradigms) is headed.

During my time there, I attended several workshops and talks that were highly informative, both with regards to guiding personal growth as an AI engineer but also in just providing up-to-date information on AI trends. Out of those, two specific events come to mind as being particularly captivating.

The first was a "crash course" on strong evaluation patterns for LLM-based applications. As someone who's had to (and will have to in my future role(s) as an engineer) develop robust methods to ensure good performance, avoid application regressions, and maintain solid user experiences, the presenter spoke to many of my experiences, both good and bad, with the topic. As a notable example, those seeking to evaluate their AI-based applications (e.g., an AI chat assistant, potentially voice-based) must deal with the fact that outputs are free-flowing, non-deterministic text, meaning that evaluating output quality, policy adherence, or other usually quantitative measures now become highly subjective. As such, a common technique is to use another LLM as a judge ("LLM-as-a-judge"), though the speaker cautioned that this approach is often highly misused. Many succumb to the "god-evaluator" pattern by giving these judges highly complex rubrics to generate — something I can attest to through my first attempts a year prior. Instead, he suggested deploying multiple judges, each with smaller, well-scoped responsibilities and encouraged the use of binary ratings in place of Likert scales (i.e., to avoid "rate the responses <use of citations> on a scale from 1-5"). All in all, the talk was niche but highly informative for things I'd been building.

The second talk was more generalist and focused on the rapid rise of OpenClaw, the AI agent that proposed to essentially manage one's life through a convenient medium like text or WhatsApp. Peter Steinberger (the agent's creator) delivered the talk himself, focusing on highly critiqued aspects of the application. Out of those, security emerged as the agents biggest "weakpoint." Steinberger explained how maintaining OpenClaw's software, dealing with attacks, and properly sandboxing the agent proved highly challenging. Looking back, his talk was surprisingly candid and informative on what challenges looked like at a lower level.

Overall, the experience was amazing. I learned a great deal, not to mention the many great folks I met in line, at dinner, and at events. I'm very happy and grateful I had the chance to attend!