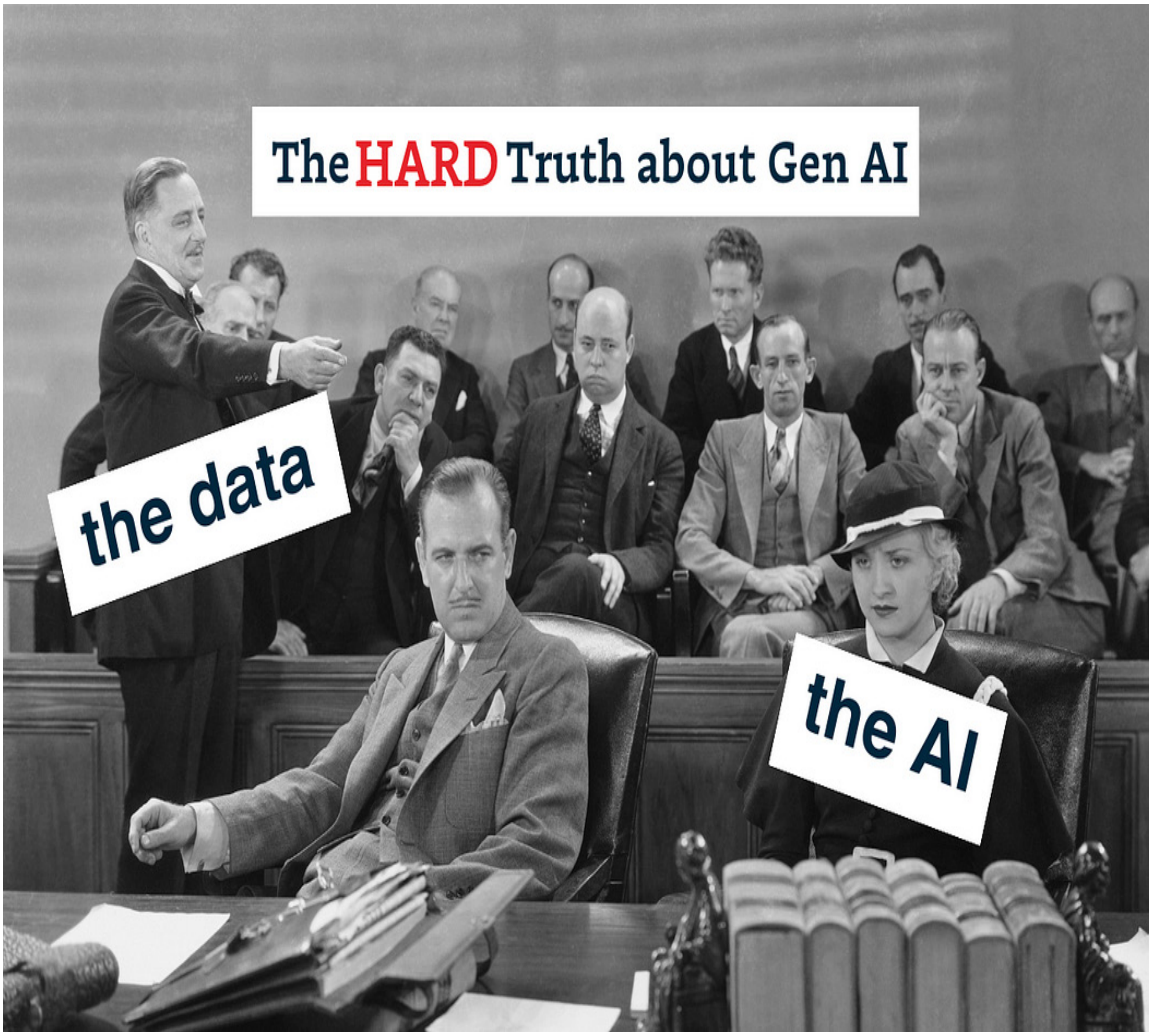
5 Hard Truths About Generative AI for Technology Leaders



GenAI that drives real business value takes real work. But it's worth it.

GenAI is everywhere you look, and organizations across industries are putting pressure on their teams to join the race – 77% of business leaders fear they're already missing out on the benefits of GenAI.

Data teams are scrambling to answer the call. But building a generative AI model that actually drives business value is hard.

And in the long run, a quick integration with the OpenAI API won't cut it. It's GenAI, but where's the moat? Why should users pick you over ChatGPT?

That quick check of the box feels like a step forward, but if you aren't already thinking about how to connect LLMs with your proprietary data and business context to actually drive differentiated value, you're behind.

That's not hyperbole. I've talked with half a dozen data leaders just this week on this topic alone. It wasn't lost on any of them that this is a race. At the finish line there are going to be winners and losers. The Blockbusters and the Netflixes.

If you feel like the starter gun has gone off, but your team is still at the starting line stretching and chatting about "bubbles" and "hype," I've rounded up 5 hard truths to help shake off the complacency.

Hard truth #1: Your generative AI features are not well adopted and you're slow to monetize.

"Barr, if GenAI is so important, why are the current features we've implemented so poorly adopted?"

Well, there are a few reasons. One, your AI initiative wasn't built as a response to an influx of well-defined user problems. For most data teams, that's because you're racing and it's early and you want to gain some experience.

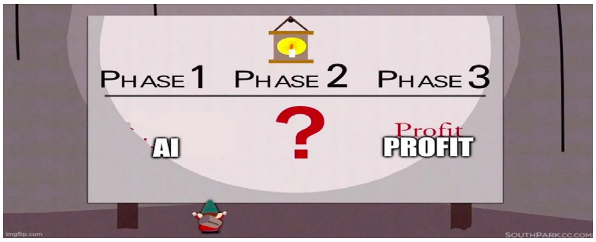
However, it won't be long before your users have a problem that's best solved by GenAI, and when that happens – you will have much better adoption compared to your tiger team brainstorming ways to tie GenAI to a use case.

And because it's early, the generative AI features that have been integrated are just "ChatGPT but over here."

Let me give you an example. Think about a productivity application you might use everyday to share organizational knowledge. An app like this might offer a feature to execute commands like "Summarize this," "Make longer" or "Change tone" on blocks of unstructured text. One command equals one AI credit.

Yes, that's helpful, but it's not differentiated.

Maybe the team decides to buy some AI credits, or maybe they just simply click over on the other tab and ask ChatGPT. I don't want to completely overlook or discount the benefit of not exposing proprietary data to ChatGPT, but it's also a smaller solution and vision than what's being painted on earnings calls across the country.



That pesky middle step from concept to value. Image courtesy of Joe Reis on Substack.

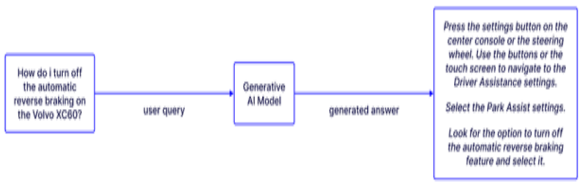
So consider: What's your GenAI differentiator and value add? Let me give you a hint: high-quality proprietary data.

That's why a RAG model (or sometimes, a fine tuned model) is so important for Gen AI initiatives. It gives the LLM access to that enterprise proprietary data. I'll explain why below.

Hard truth #2: You're scared to do more with Gen AI.

It's true: generative AI is intimidating.

Sure, you could integrate your AI model more deeply into your organization's processes, but that feels risky. Let's face it: ChatGPT hallucinates and it can't be predicted. There's a knowledge cutoff that leaves users susceptible to out-of-date output. There are legal repercussions to data mishandlings and providing consumers misinformation, even if accidental.



Hallucinated Generative Chat

Sounds real enough, right? Llama 2 sure thinks so. Image courtesy of Pinecone.

Your data mishaps have consequences. And that's why it's essential to know exactly what you're feeding GenAI and that the data is accurate.

In an anonymous survey we sent to data leaders asking how far away their team is from enabling a GenAI use case, one response was, "I don't think our infrastructure is the thing holding us back. We're treading quite cautiously here – with the landscape moving so fast, and the risk of reputational damage from a 'rogue' chatbot, we're holding fire and waiting for the hype to die down a bit!"

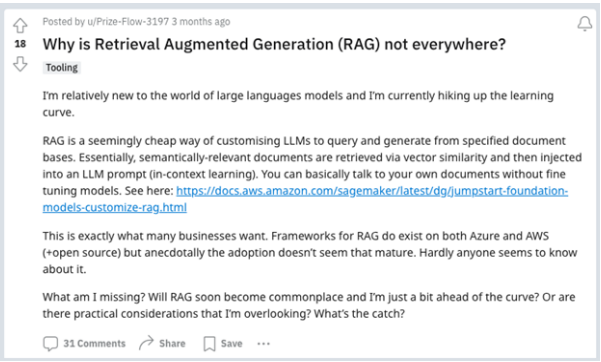
This is a widely shared sentiment across many data leaders I speak to. If the data team has suddenly surfaced customer-facing, secure data, then they're on the hook. Data governance is a massive consideration and it's a high bar to clear.

These are real risks that need solutions, but you won't solve them by sitting on the sideline. There is also a real risk of watching your business being fundamentally disrupted by the team that figured it out first.

Grounding LLMs in your proprietary data with fine tuning and RAG is a big piece to this puzzle, but it's not easy…

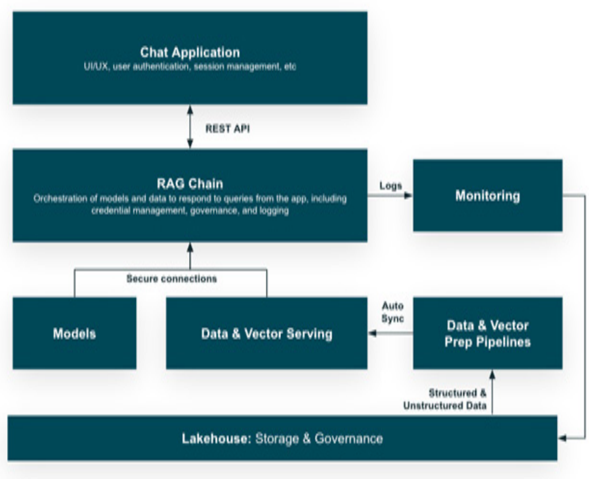
Hard truth #3: RAG is hard.

I believe that RAG (retrieval augmented generation) and fine tuning are the centerpieces of the future of enterprise generative AI. But although RAG is the simpler approach in most cases, developing RAG apps can still be complex.



Can't we all just start RAGing? What's the big deal? Image courtesy of Reddit.

RAG might seem like the obvious solution for customizing your LLM. But RAG development comes with a learning curve, even for your most talented data engineers. They need to know prompt engineering, vector databases and embedding vectors, data modeling, data orchestration, data pipelines… all for RAG. And, because it's new (introduced by Meta AI in 2020), many companies just don't yet have enough experience with it to establish best practices.



RAG application architecture. Image courtesy of Databricks.

Here's an oversimplification of RAG application architecture:

1. RAG architecture combines information retrieval with a text generator model, so it has access to your database while trying to answer a question from the user.

2. The database has to be a trusted source that includes proprietary data, and it allows the model to incorporate up-to-date and reliable information into its responses and reasoning.

3. In the background, a data pipeline ingests various structured and unstructured sources into the database to keep it accurate and up-to-date.

4. The RAG chain takes the user query (text) and retrieves relevant data from the database, then passes that data and the query to the LLM in order to generate a highly accurate and personalized response.

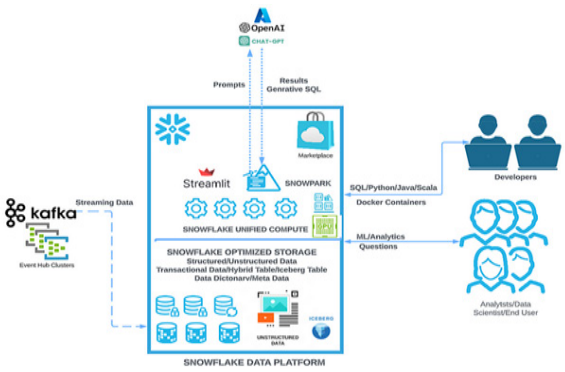
There are a lot of complexities in this architecture, but it does have important benefits:

1. It grounds your LLM in accurate proprietary data, thus making it much more valuable.

2. It brings your models to your data rather than bringing your data to your models, which is a relatively simple, cost-effective approach.

We can see this becoming a reality in the modern data stack. The biggest players are working at a breakneck speed to make RAG easier by serving LLMs within their environments, where enterprise data is stored.

Snowflake Cortex now enables organizations to quickly analyze data and build AI apps directly in Snowflake. Databricks' new Foundation Model APIs provide instant access to LLMs directly within Databricks. Microsoft released Microsoft Azure OpenAI Service and Amazon recently launched the Amazon Redshift Query Editor.



I believe all of these features have a good chance of driving high adoption. But, they also heighten the focus on data quality in these data stores. If the data feeding your RAG pipeline is anomalous, outdated, or otherwise untrustworthy, what's the future of your generative AI initiative?

Hard truth #4: Your data isn't ready yet anyway.

Take a good, hard look at your data infrastructure. Chances are if you had a perfect RAG pipeline, fine tuned model, and clear use case ready to go tomorrow (and wouldn't that be nice?), you still wouldn't have clean, well-modeled datasets to plug it all into.

Let's say you want your chatbot to interface with a customer. To do anything useful, it needs to know about that organization's relationship with the customer. If you're an enterprise organization today, that relationship is likely defined across 150 data sources and 5 siloed databases… 3 of which are still on-prem.

If that describes your organization, it's possible you are a year (or two!) away from your data infrastructure being GenAI ready.

Which means if you want the option to do something with GenAI someday soon, you need to be creating useful, highly reliable, consolidated, well-documented datasets in a modern data platform… yesterday. Or the coach is going to call you into the game and your pants are going to be down.

Your data engineering team is the backbone for ensuring data health. And, a modern data stack enables the data engineering team to continuously monitor data quality into the future.

Hard truth #5: You've sidelined critical Gen AI players without knowing it.

Generative AI is a team sport, especially when it comes to development. Many data teams make the mistake of excluding key players from their GenAI tiger teams, and that's costing them in the long run.

Who should be on an AI tiger team? Leadership, or a primary business stakeholder, to spearhead the initiative and remind the group of the business value. Software engineers to develop the code, the user facing application and the API calls. Data scientists to consider new use cases, fine tune your models, and push the team in new directions. Who's missing here?

Data engineers.

Data engineers are critical to GenAI initiatives. They're going to be able to understand the proprietary business data that provides the competitive advantage over a ChatGPT, and they're going to build the pipelines that make that data available to the LLM via RAG.

If your data engineers aren't in the room, your tiger team is not at full strength. The most pioneering companies in GenAI are telling me they are already embedding data engineers in all development squads.

Winning the GenAI race

If any of these hard truths apply to you, don't worry. Generative AI is in such nascent stages that there's still time to start back over, and this time, embrace the challenge.

Take a step back to understand the customer needs an AI model can solve, bring data engineers into earlier development stages to secure a competitive edge from the start, and take the time to build a RAG pipeline that can supply a steady stream of high-quality, reliable data.

And, invest in a modern data stack to make data quality a priority. Because generative AI without high-quality data is just a whole lotta' fluff.