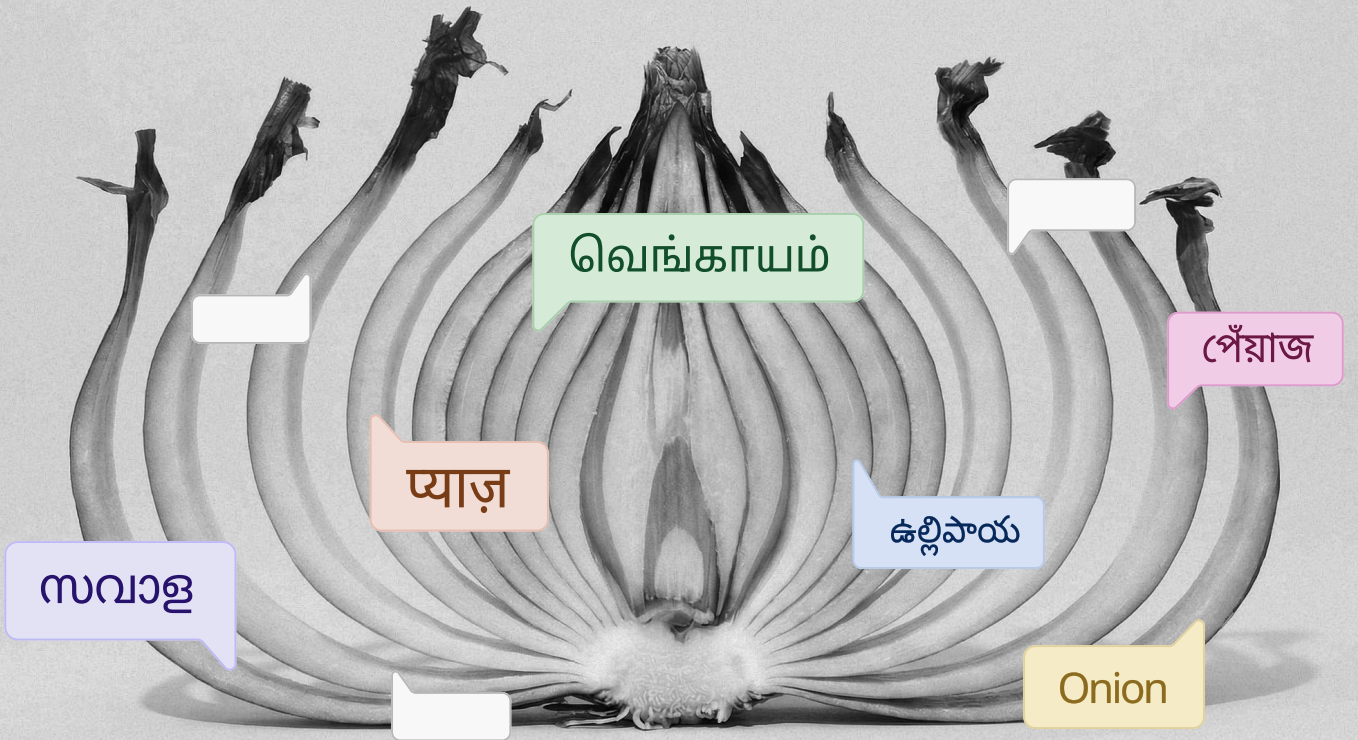


The Onion Problem

Why AI search in Indian commerce needs a different architecture

Sreeraman MG





Introduction

India has close to 900 million internet users, and that majority of them engage with content in Indic languages.

Most conversations about AI in Indian commerce start with language. India has 22 official languages, thousands of dialects, and a billion-plus consumers who think, speak, and shop in ways that don't map to how global platforms were built.

The usual prescription: build multilingual models, translate everything, and the problem is solved.

But it isn't that simple. Language is the part everyone talks about. The harder problem underneath is that product discovery in India requires a kind of commercial intelligence that foundation models handle partially at best, and that nobody has fully built yet.

I've been thinking about this for a while. Working closely with brands and building AI-led search experiences, one thing became obvious: what it takes to make AI-powered search work for Indian consumers is quite different from the common narrative, and there are still meaningful gaps.



The Onion Problem

An onion is *vengayam* in Tamil, *pyaaz* in Hindi, *eerulli* in Kannada, *savola* in Malayalam.

A customer in Chennai searching "vengayam" on a platform where the product catalogue is tagged only in English. They get zero results. The product exists, the inventory is available but the search system simply doesn't recognize that these are all the same thing.

At first this might sound like a translation problem. But that's only part of what's going on.

"Thali" in a jewellery catalogue means mangalsutra. "Thali" in a food delivery app means a meal plate. The word is the same, but the meaning depends entirely on context. The product category has to resolve the ambiguity, not a dictionary.

"Chudi" means bangles in one region and a dress type in another. A search for "kurta" in Lucknow brings up a very different set of products than the same search in Kochi. Even something like "Party wear" varies depending on the customer's city, age, and price bracket.

Now multiply this across millions of SKUs. For most Indian consumers, especially those not searching in English, this is what everyday product discovery actually looks like.

Where foundation models help and where they don't

I want to be precise here because most conversations about AI in Indian commerce either overstate what LLMs can do or dismiss them entirely.

Foundation models are genuinely good at language understanding. With a well-structured RAG setup over a product knowledge base, you can get quite far. An LLM can understand that "daal dhokli kaise banate hain" is a recipe query on a grocery app and extract ingredients from it. It can parse "something nice for my mom's birthday under 2000" into structured filters. For intent understanding and query parsing, LLMs are already good enough.

Where they fall short is regional commercial vocabulary and product-to-occasion mappings. When a customer types "something for Onam," the model needs to understand that Onam is a harvest festival in Kerala, that people generally look for kasavu sarees, gold jewellery, and mundu sets, that price expectations differ from Diwali shopping, and that this query is usually relevant around August–September.

Current models have some of this cultural knowledge, but it's patchy, inconsistent, and not grounded in actual purchase behaviour. They know what Onam is, but not necessarily what sells during Onam in Thrissur versus Kochi.

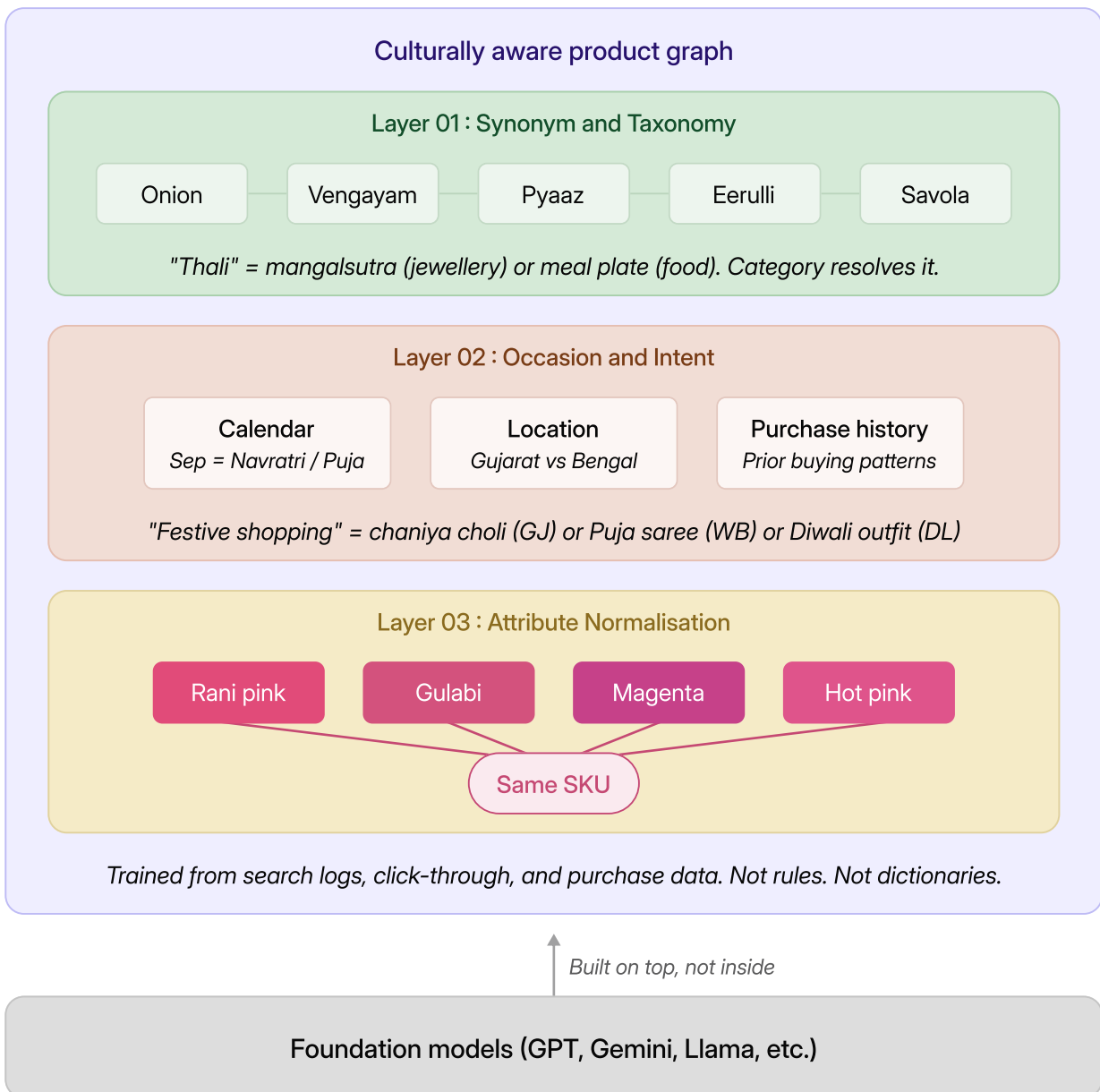
Similarly, a fine-tuned embedding model with a good synonym table can handle many of the cross-language product matches. But it won't catch that "chaniya choli" in Surat and "lehenga" in Delhi are the same category, or that "gulabi" and "rani pink" should map to the same colour, unless it's been trained on Indian commercial data specifically.



The gap is in the last mile of commercial context. That's where I think a purpose-built intelligence layer is needed.

What I think needs to be built: a culturally aware product graph

I've been calling this a "culturally aware product graph." It's a proposed architecture, not a shipped product. But based on the problems we see in everyday commerce, I believe it needs three layers sitting between the foundation model and the product catalogue.





Layer 01

Synonym and Taxonomy Resolution

This is the most structured layer. *Vengayam* = *onion* = *pyaaz* = *eerulli*. But it can't be a flat lookup table. It has to be contextual. "Thali" in jewellery maps to mangalsutra. "Thali" in food maps to a meal plate. The product category determines which synonym applies.

At scale, this can't be built by manually writing mappings. That approach won't scale. It needs to learn from actual user behaviour. For example, if most users who search for "vengayam" end up clicking on and buying products tagged "onion," that becomes a strong signal. The synonym is validated by what people do, not just by what a dictionary says. Over time, the graph grows from real behaviour.

Layer 02

Occasion and Intent Mapping

This is harder. "Festive shopping" in September can resolve to very different product sets depending on where the user is. In Gujarat, it might mean chaniya choli for Navratri. In Bengal, Durga Puja sarees. In Delhi, Diwali office outfits. The query is the same, but the expected results are not.

This can't be a static mapping. It needs to combine three signals: calendar awareness (which festival is approaching), location signals (where the user is or where they're shipping to), and historical purchase patterns for that region (what people in this area actually bought during the same period last year).

It also needs to handle more everyday ambiguity. A query like "Budget phone for my father" requires understanding that "budget" has a different threshold for different user segments, that "for my father" implies a preference for larger screens and simpler interfaces. None of this is explicitly stated, but it still affects what should be shown.

Layer 03

Attribute Normalisation

This is where the real mess lives. Indian product catalogues are among the most inconsistent in the world, with sellers describing the same product in very different ways depending on their region, language, and level of digital sophistication.

For example, one brand lists fabric as "georgette," another spells it "jorjet," and a third uses "chiffon" for something that looks identical in product photos. Colour names are even worse. "Rani pink," "magenta," "hot pink," and "gulabi" might all describe the same shade.

Because of this, rule-based normalization breaks down. The variations are effectively infinite. I believe the approach that will work is multimodal embedding-based clustering, jointly trained on product images and their textual descriptions. The intuition is simple: two products described differently but photographed similarly end up close in embedding space. So, when the model sees that products tagged "gulabi" and products tagged "rani pink" have near-identical image embeddings and similar purchase patterns, it can learn the equivalence without anyone writing an explicit rule.

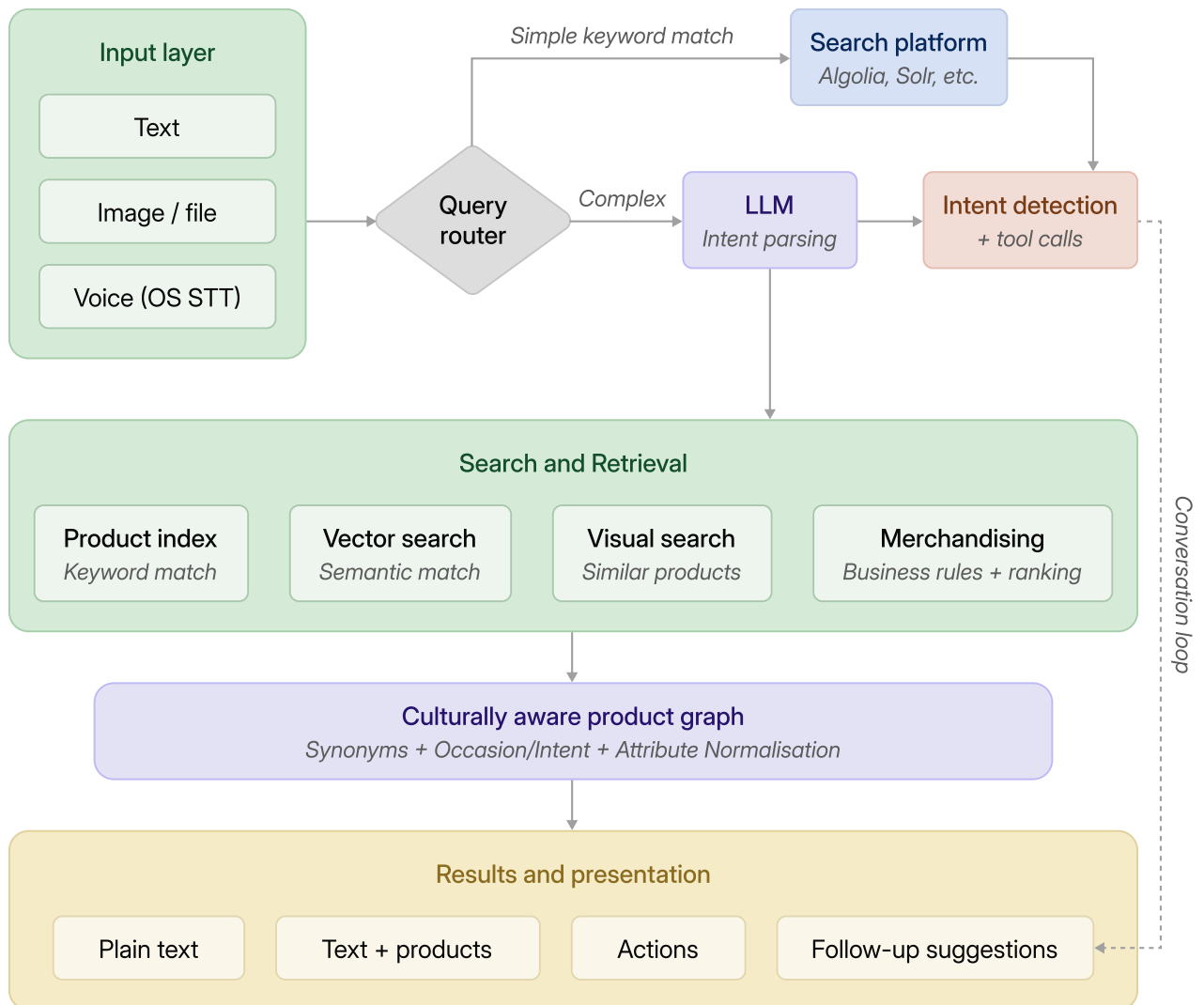


An honest caveat: this approach has real limitations. Multilingual embedding quality varies significantly across Indian languages. Semantic search on transliterated queries (Hindi written in English script, which is extremely common) is notoriously unreliable. A customer typing "kurta" in English script versus कुरता in Devanagari can get meaningfully different results, and closing this gap requires language-specific embedding tuning that few teams have done well.

Cold start is another problem: when a new seller uploads products with novel regional terminology, the system has no behavioural data to learn from yet.

How I think AI search architecture should work

AI search architecture (proposed)





When designing AI Search experiences, I found that the search bar itself needs to be rethought. The traditional ecommerce search box assumes the user will type a keyword, the system will match it against a product index, and results will appear. I believe that assumption breaks for Indian commerce in at least four ways.

01 / The input can't be text-only.

I've seen store staff photograph items instead of typing names. Customers share shopping lists as images. A recipe screenshot can become a grocery query. The search interface needs to accept images, voice, and attachments alongside text. This is something we designed into our AI Search Bar with support for text, image attachments, voice input, and conversational agent interaction.

02 / Not every query belongs in a search index.

"Daal dhokli kaise banate hain?" typed into a grocery app shouldn't directly hit the product database. It should first go to an LLM that understands the recipe, extracts the ingredients, and then queries the product index for each one. This requires a query router: simple keyword searches go directly to the search platform, while more complex, intent-based queries are routed through an LLM for understanding first. In our architecture, this is a query router that classifies intent and decides the processing path before any search happens.

03 / Response format should vary by intent.

Some queries need a product grid. Others need a text answer with product cards embedded. In some cases, the user may want to take an action, like adding items to a cart or checking delivery. The system has to determine not just what the user wants, but also how to present it.

04 / The conversation should continue.

After showing results, the system should suggest follow-up queries, turning a search into a guided exploration rather than a dead-end results page. A user who searched "running shoes" might benefit from seeing "running shoes under ₹3000" or "running shoes for flat feet" as follow-up suggestions. We designed this as a suggested pills system that creates a feedback loop between the user and the AI.

The speech-to-text design decision

One design choice worth sharing: for voice input in the search bar, we deliberately chose OS-level speech-to-text rather than routing voice through an LLM.

First, it's what users already know. Most smartphone users in India are already used to voice typing. Asking them to speak to a new AI system adds unnecessary friction. Using the OS keyboard's built-in speech recognition keeps the experience simple and familiar.



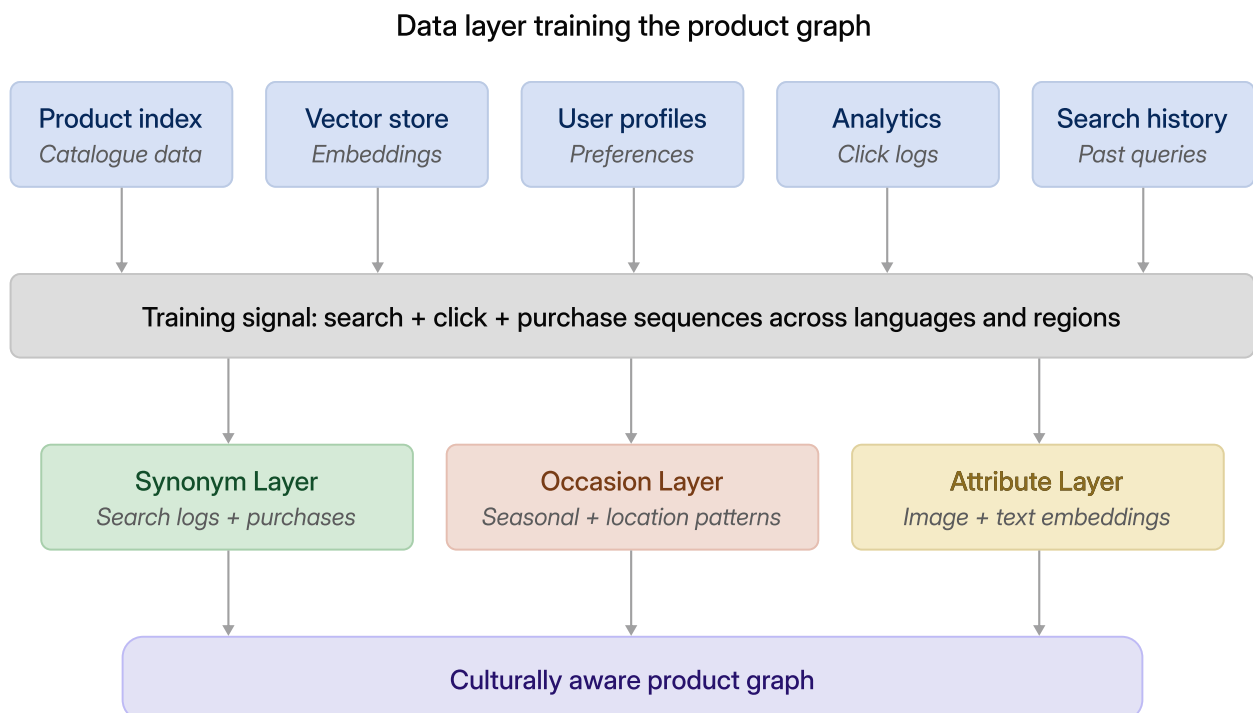
Second, it avoids unnecessary token consumption. Running voice through an LLM when all you need is transcription is wasteful. The LLM should only activate when the transcribed text needs to be understood, not for the transcription itself. When you're building AI products for Indian price points, this kind of cost discipline ends up shaping a lot of architectural decisions.

For the conversational agent mode, where the user explicitly wants to talk to an AI shopping advisor, live transcription can feed into the LLM. But for basic voice search, the OS can handle it. Different interaction modes, with very different cost profiles.

The data underneath all of this

None of this intelligence layer works without the right data foundation. You need a product index (catalogue data), a vector store (embeddings for semantic matching), user profiles (preferences from history), analytics data (click logs, conversion data), and search history (past queries and outcomes).

If built properly, the culturally aware product graph would sit on top of all five. The synonym layer comes from search logs matched against actual purchase behaviour. The occasion layer is shaped by seasonal patterns, tied to location. The attribute normalisation layer relies on product image embeddings aligned with textual descriptions.





Who has the advantage here

I want to be careful with this claim. Large platforms operating in India, both global and domestic, already have massive transaction datasets. They have the raw material.

But the advantage, in my view, comes down to incentives. Global platforms optimize their search infrastructure for their largest markets. The ROI on building something like a culturally aware product graph for the US or Europe is low, because the underlying problem is far less pronounced. Product catalogues are more standardised, language is more uniform, and occasion-based commerce is less regionalized.

For India, this layer is essential. That misalignment of incentives, more than data access, is what creates the real opportunity.

What needs to exist

I don't think any single company will build a complete culturally aware product graph for Indian commerce. The data is too distributed, and the regional variation runs too deep. But the overall architecture is starting to become clearer.

Foundation models handle language understanding. A commerce-specific intelligence layer handles synonym resolution, occasion mapping, and attribute normalisation. A multimodal search interface handles text, image, voice, and conversation. A query router decides which queries need an LLM and which don't. And underneath all of this, a data layer continuously trains the system using real search, click, and purchase behaviour.

Some parts of this are already being built across the industry, and some are still unsolved. For anyone working on AI in Indian commerce, this feels like one of the more interesting problem spaces right now. The companies that get this right will have an advantage that compounds over time, because the intelligence layer improves with every transaction, and that data is geography-locked.

The intelligence isn't in the model. It's in the data that trains the layer on top of the model. And that data only comes from operating at scale in Indian markets, across languages, and serving real consumer behaviour.



***Sreeraman MG** is Founder of Fynd, an AI-native unified commerce platform. He has spent over a decade building commerce technology for brands and retailers, first in India and now across global markets.*

Fynd