The End of the Unstructured Data Era

October 2, 2024

Retrieval Augmented Generation. RAG.

I'm going to attempt to push past the marketing speak and focus on a few key areas that I think are really special about what we're doing in the space of information retrieval for AI applications. Due to the fact that there are several **fundamental** innovations that change the definition of how enterprises will define terms such as "real time" and "enterprise-ready," this post is a relatively long one. Buckle up.

Table of Contents:

In August of 2020, a team at Facebook first presented a paper on **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**. This paper introduced an idea of how to augment the information retrieval of an AI model with additional data that a model could query and retrieve from external data sources. Four years later, the market has now developed the tools and ecosystem required to establish RAG as the 'killer app' that can usher in the value of AI to enterprises looking to embrace faster knowledge-sharing and decision-making.

So, what is the challenge that enterprise RAG-based systems are attempting to solve?

As organizations start to bring AI agents into their environment, it's clear that there are things that an agent can and does know, and there are things trained AI models can't yet accomplish. Some challenges include:

- AI models are trained and fine-tuned periodically, so they may not be equipped with the latest and greatest information needed to power a business in real time.
- Generative AI models rely on the accuracy of generative predictions - and they can hallucinate when they rely solely on patterns within their training dataset.
- Models are also bound by the information they've been trained on, and optimized for the amount of memory needed in the machines you can deploy AI agents on. Therefore, AI teams will often look to minimize the size of the model and augment AI applications with external data sources (versus doing much much larger training and fine-tuning work and then fitting inference models into expensive, memory-rich machines).
- Finally, AI models are not typically designed to enforce enterprise access control or permissions. Therefore, it can be impractical to train them on large enterprise datasets that have a wide variety of access permissions, since users cannot be given unfettered access to the information these models gain. Conversely, it's inefficient to train a model per enterprise user on just the data they are permitted to see. Even if you could, permissions change more frequently than models do.

Fortunately, AI retrieval provides the answer to the above problems and paves the way for general enterprise adoption of agentic AI applications. A generalized workflow looks as follows:

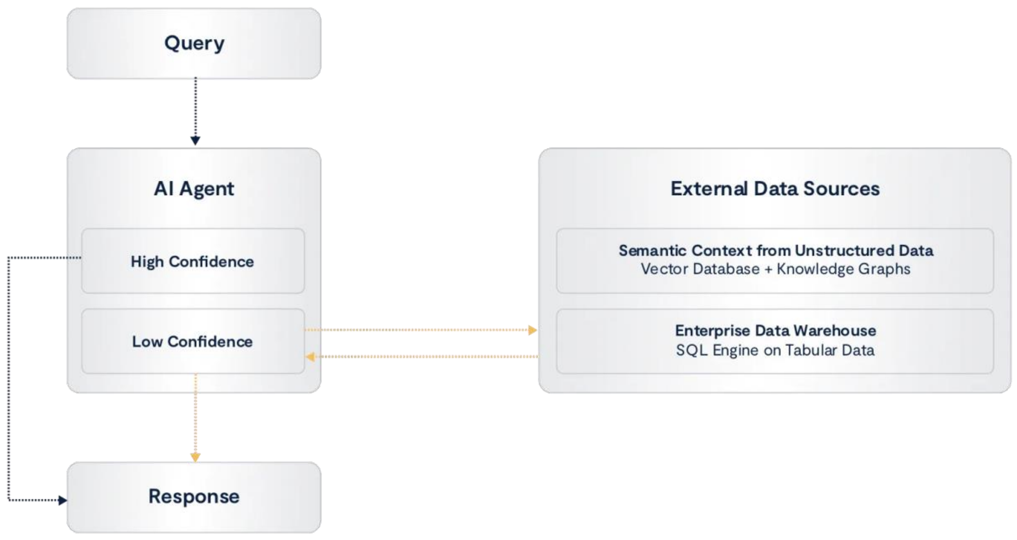
1. An AI inference model, typically an LLM, will receive a request/prompt from some user

2. The model will attempt to answer the question itself, if possible, and then resort to going to external data sources when the response's confidence score is not high enough

3. At this point, the model can retrieve external data typically from one of two sources

a. If the answers live inside an organization's unstructured data, this will have been indexed by some AI embedding model and the context from enterprise file and object storage systems will be then stored and indexed in a vector database

b. Alternatively, structured data that exists within enterprise data warehouses can be easily queried by an AI model using SQL


Query
AI Agent
High Confidence
Low Confidence
External Data Sources
Semantic Context from Unstructured Data
Vector Database + Knowledge Graphs
Enterprise Data Warehouse
SQL Engine on Tabular Data
Response

Retrieval solves for many of the key drawbacks of AI models within the enterprise.

AI Model Challenge	Retrieval Solution
Models are trained and fine-tuned periodically – their data may not be current.	Retrieval systems can source data from real-time or near-real-time sources.
Models can hallucinate.	Retrieval allows AI models to not have to know everything, and to rather just serve as interpreters of queries that can source facts.
Big and smart models can be expensive to deploy.	Retrieval allows for AI models to stay small while augmenting them with massive data volumes that are semantically indexed.
AI models typically can't enforce granular permissions.	Retrieval aligns the query output with individual user permissions, enabling data isolation across sensitive enterprise datasets.

OK… so, we've solved enterprise AI and everything's right in the world… right? No. While there are several different approaches to retrieval in the market, none has passed our sniff test so far.

Problem #1: Real Time Is Not Really Real Time

Many vendors in the real-time RAG space talk about their ability to answer queries quickly, and don't address that the data they've indexed is the product of some periodic batch process that isn't current with what's happening in real time in an organization.

While datasets for retrieval can be updated more frequently than a model is fine-tuned, this process is not yet able to feed information to AI models in real time.

- Vector embeddings are created from file data only periodically, often on a batch schedule.
- File system ACLs are not atomic with vector databases, also synced only periodically.
- Enterprise data warehouses (EDWs) struggle to ingest data in real time, so real-time data starts in an event bus and is difficult to query against until it is ETL'd into an EDW, which is also done as a periodic batch process.

And all of this source data can change… data doesn't only get expanded, data can be removed from the enterprise knowledgebase. This makes the role of retrieval systems even more complicated when they can and do expose context that is suddenly out of date or inaccurate.

Consider, finally, that a human is only one type of user of an AI model. AI agents are now being combined to execute complex AI workflows and enhance each other. If a business is beginning to run at the speed of AI, it's unthinkable that AI engines should wait minutes or days for up-to-date business data.

Why is building scalable + real-time systems hard? It's the architecture (I'll come back to this).

Problem #2: Vector Database Scalability Trades Scale for Performance

As we poll our average Fortune 500 customers, putting aside the large hyperscalers, we find customers that are sitting on 100s of petabytes of unique enterprise data. In order to appropriately index this data using AI and data vectorization tools, we'll need systems that can store and index trillions of vector embeddings and be able to search on them in real time (in order to preserve a quality user experience). This is not the design point that the early vectorDB efforts were built around - as they rushed to get

out solutions for chatbots that front-ended small datasets such as PDF document stores.

Sure, people today may say… 'I only have to curate a small number of documents and feed them into a vector DB, since the embedding models require hand-holding'. At VAST, we're not worrying about the state of AI-based data indexing as it is today. We're watching the exponential improvements in embedding models and seeing a time in the not-too-distant future where these tools will be of a caliber that they can be used to index the entirety of an enterprise knowledge base and even help to curate enterprise data. At that point, hyper-scale vector stores and search tools are table stakes.

The trillion-way vector problem is already a reality for large AI builders like OpenAI and Perplexity that have gone and indexed the internet. This is the scale of product we want to build… for everyone.

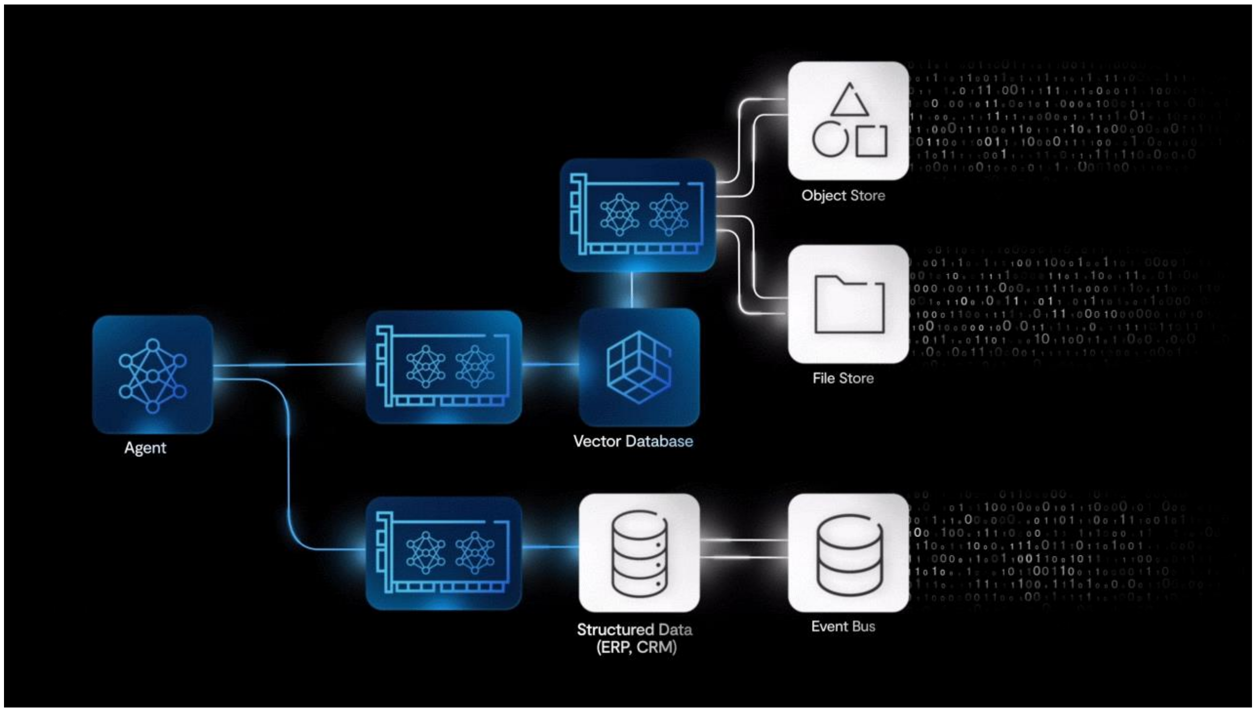
Finally, you won't just need to read from vector search engines in constant time — you'll also need to be able to create, store and index embeddings in real time. At this point, we find ourselves back in the place we were with problem #1… architecture is a core limiter for being able to scale datasets and transacting into them with high performance when technologies lean on the classic shared-nothing systems designs.

Problem #3: It's All Just Kinda Messy

RAG pipelines involve many steps:

- Set up a system alongside your primary file and object storage systems that can support AI-based indexing, since legacy file systems don't handle AI parallel processing well. If you want to use a cloud-based data platform, just copy all of your enterprise file systems into some cloud storage bucket.
- Then build some GPU cluster to periodically scan and create these embeddings, with all of the workflow orchestration and gluecode to make this automatic.
- Then build a vector database that can store and index these embeddings, as well as help build knowledge graphs. Stamp these out into silos of information insight as you hit scaling walls.
- Remember, to save money, you should also tier your embeddings off to some slow object storage with the intent of never having to query against that data.
- Then build an enterprise data warehouse to support your Text-to-SQL queries.

- But to do this, you'll also need to build an event bus that captures real-time event streams since the data warehouse is not transactional, hoping you don't need to correlate the two at the same time.
- Then try your best to establish unified policy enforcement across a bunch of independent systems and hope that permissions don't fall too far out of sync that anyone notices.
- And finally, spend the next three years trying to tune this infrastructure so that you can get as close to real time as possible so that your business isn't continually getting served old or stale data.



AI Retrieval Problems: The Big Takeaway

With respect to these pipelines… I believe the world has been doing the wrong thing.

The enterprise just wants more intelligence from file and object storage. I think of vectors and knowledge graphs as just higher forms of file system metadata… why wouldn't we want this to run natively from within the file system if it was possible? This would solve the problems of data and metadata atomicity and make it easy for AI agents to make data conversational, talking to storage almost as a new protocol. In order for this to happen, a file system would need support for real-time transactional and analytical metadata analytics, a job engine, event triggers, etc. Well, that's a data platform.

And why wouldn't we want real-time transactional data warehouses? Of course we do. It's just not been possible… until now.

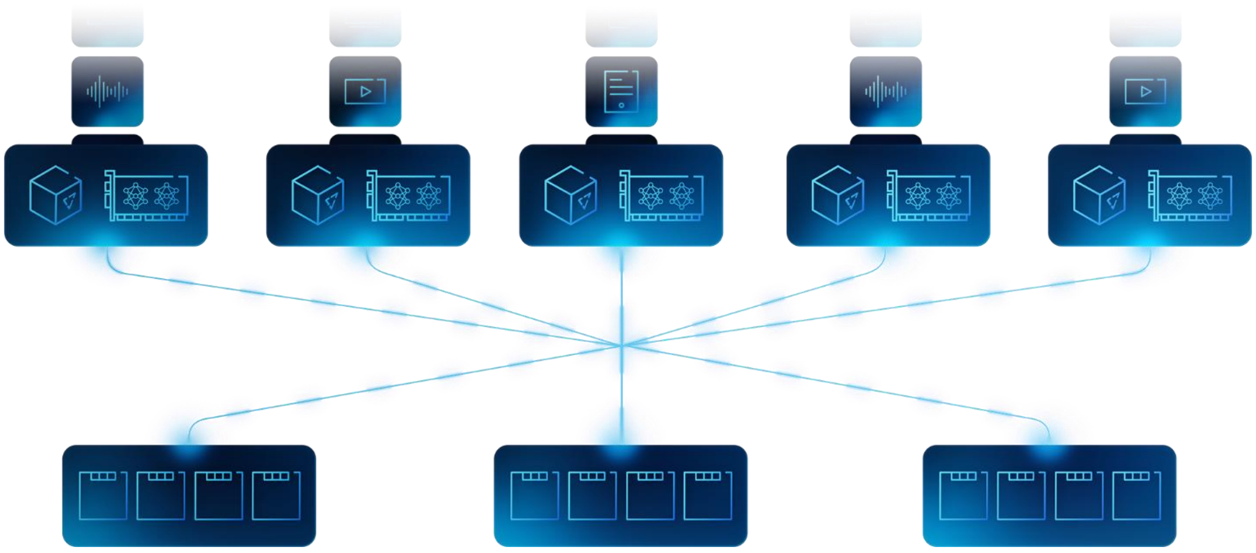
Introducing the InsightEngine

This is why, today, we're delighted to take the covers of our collaboration with an AI leader and introduce the VAST Data InsightEngine with NVIDIA.

VAST InsightEngine with NVIDIA is an all-encompassing system to support AI Retrieval that allows organizations to ingest all of their structured and unstructured data streams in real time; to index them using AI embedding tools in real time; and to serve answers to agentic applications with real-time search tools that work at any scale, and with the enterprise security that comes naturally from building intelligence into an enterprise data foundation. It's fast, scalable, secure and simple.

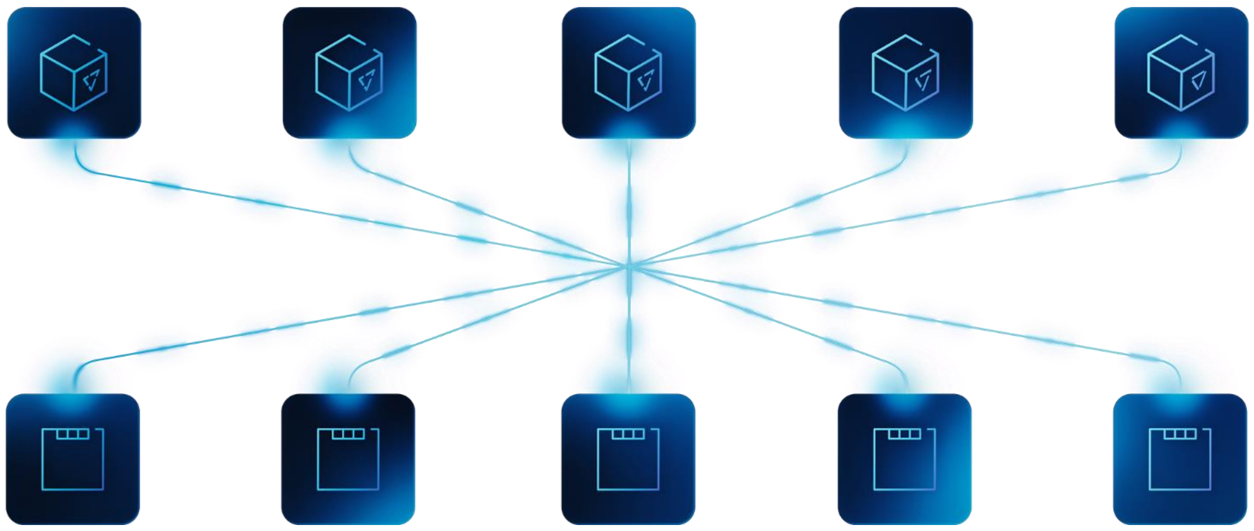
VAST InsightEngine with NVIDIA is designed to run **NVIDIA NIM** microservices, part of the **NVIDIA AI Enterprise** software platform, natively from within the VAST Data Platform, leveraging VAST containers that support NVIDIA accelerated computing for AI acceleration. Inside the system, real-time workflows are built into the **VAST DataEngine** that leverage native application triggers that automatically create embeddings on data and update vector indexes as new unstructured data comes into the system. To start, we'll support NVIDIA NeMo Retriever NIM microservices that create embeddings from text and video, and others will be added in the future.

Now, it's one thing to create a separate storage space for all of your AI Retrieval data… this, to us, is crazy. This is why, from the earliest days, we've been working on building an enterprise scale-out NAS and object storage system (among other things) that is Universal in nature. We don't want customers to take their enterprise data to some GPUs — we want them to bring GPUs to their enterprise data that is on AI-ready infrastructure... so that intelligent applications can benefit from getting the broadest access to all of an organization's unstructured data.



To enable semantic search of this unstructured data that lives in the [VAST DataStore](), we're proud to announce that the [VAST DataBase]() has been extended to support a new data type: vector embeddings. Thanks to the scalability of the [VAST DASE architecture](), we can now build table stores that support trillions of vectors, making it possible to index an entire enterprise. But scale is only one challenge — we also needed to make these data structures easy to update and query in real time.

As many of you know, VAST has been working with [Similarity algorithms]() for almost 10 years now (to optimize data reduction and make flash as affordable as HDD storage), and it turns out that these same algorithms are now very popular to help with searching large vector spaces. The VAST DASE architecture pioneers new ideas around transactional parallelism and provides all of the system's containers the ability to share a common, real-time view of a namespace organized across low-cost, low-latency flash. The result is a system that makes it possible to transact millions of events per second and to update and search indices in milliseconds.



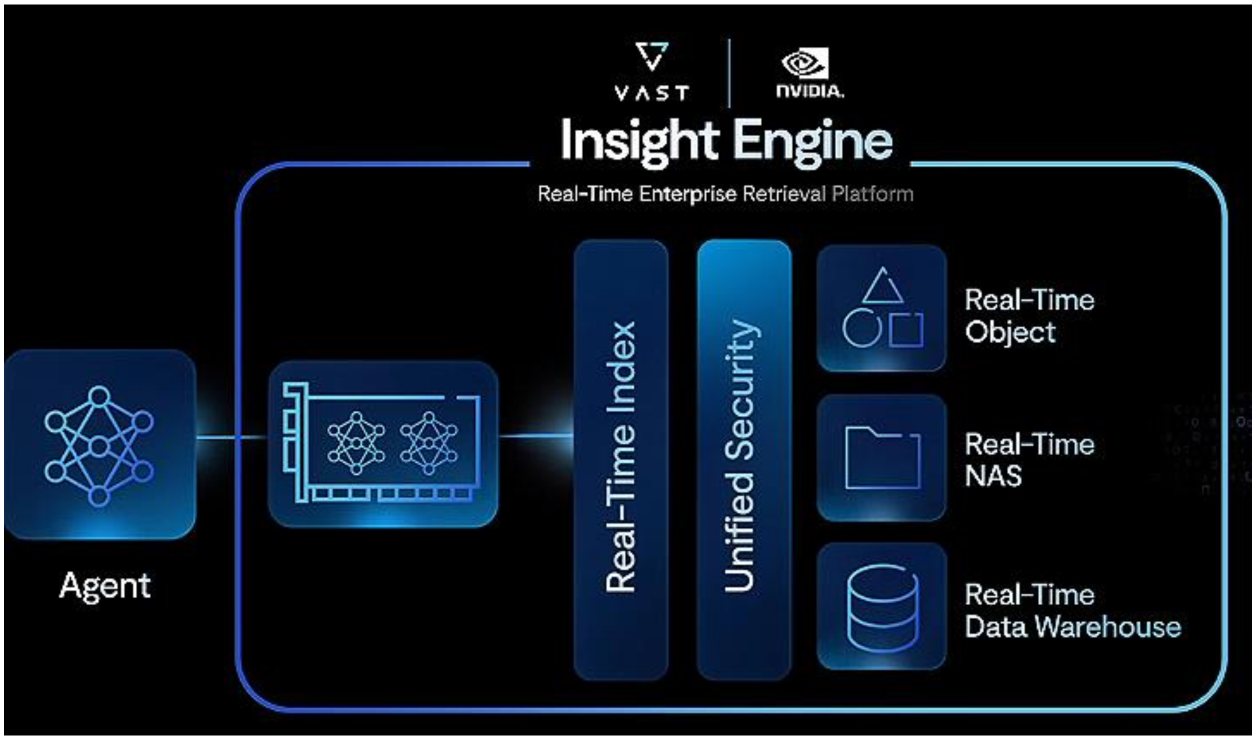
Security-wise, the platform will pull permissions and policy data up into the VAST DataBase, such that every vector search happens against permissions data that is atomically synced with the data source. No more do you need to sync independent permissions systems. Permissions are enforced and updated in real time thanks to the transactional nature of the VAST DataBase and unified data management.

Finally, for SQL-to-Text operations, we're also proud to announce that the VAST DataBase now also supports a Kafka-compatible event broker. This is a seemingly small footnote on an otherwise very long blog... but the effects of this are going to be profound across the data science community as we take the world to a place where you no longer need an external event bus to capture enterprise event streams. Now, you can stream directly into VAST DataBase tables, where event data is then instantly part of a (predominantly) columnar query.

The End of the Unstructured Data Era

Looking into the future, AI will become increasingly embedded into core business applications. File systems and object storage, alone, force IT teams to build cumbersome retrieval pipelines and wrestle with the complexity of stale data, stale permissions and a lot of integration and gluecode headaches. Most importantly, these pipelines are not working as fast as AI engines enable us to. The idea of a standalone file system is fading as new business priorities need more from data infrastructure.

At VAST, we want to store, index and serve the world's data to tomorrow's engines of discovery. To get there, our simple mission is to eliminate infrastructure constraints, bottlenecks and tradeoffs. This mission forces us to think further up the stack, and to challenge the conceptions of how systems should be built to realize the greatest gains. VAST InsightEngine with NVIDIA is the culmination of what we've been working toward for almost a decade and underscores the need for unstructured data to be treated as a first-class citizen in the age of Enterprise AI. But unstructured data, alone, is not enough.



Files, objects, tables, triggers… and now AI workflows that run natively in our Engine. We look forward to sharing more as we get closer to product launch in early ’25.

Thanks for reading.

Jeff