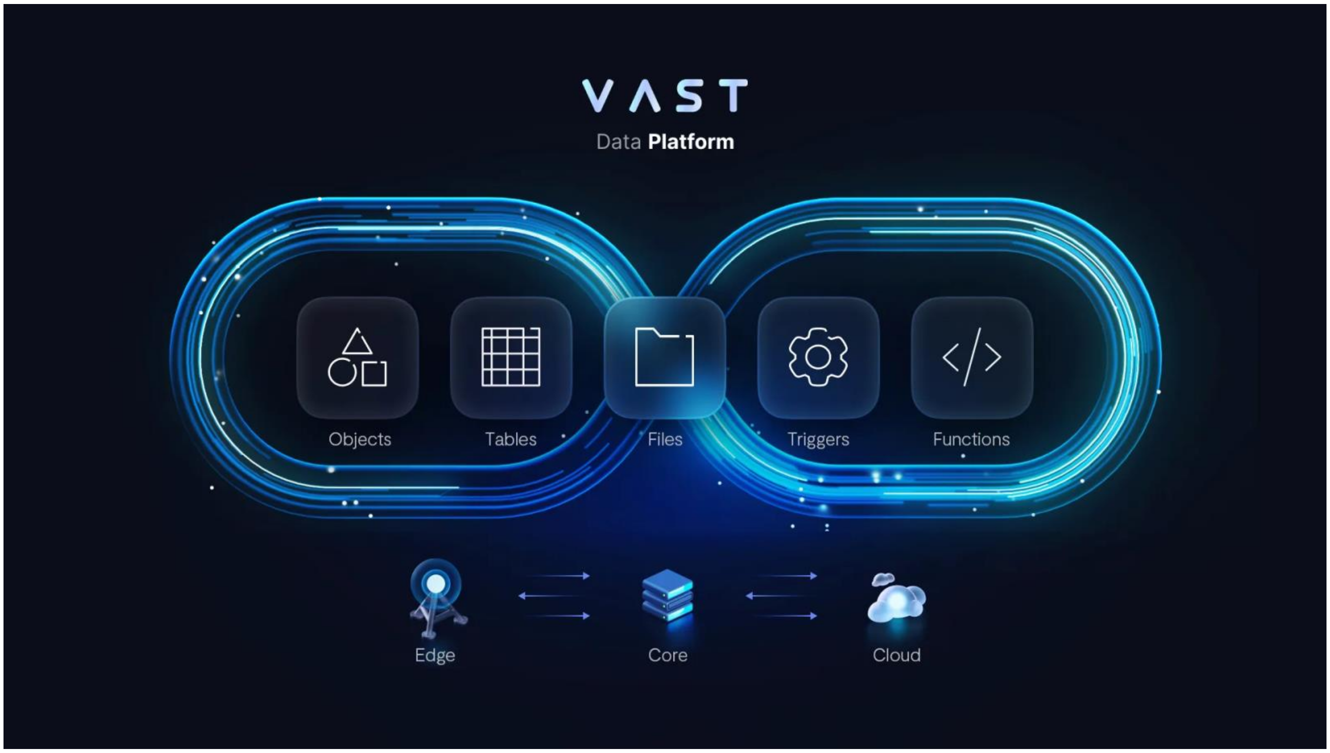
August 1, 2023

The Grand Unification Theory of AI Infrastructure

Blog Post by Jeff Denworth, VAST Data



Blog titles are tough to get right. Some other options I considered for today's momentous announcement... "The Master Plan, Part Un & Deux", "Breaking Tradeoffs: Part II". Having said that, given how many conversations I've had over the past few days about collapsing the IT stack for AI, my mind keeps wandering back to universal constructs and the power of stack unification. Get ready for words, this blog is a doozy.

<clears throat, adopts Ted-talk speaking style>

All the things.

This is the aim of a data platform... to be all the things for data-driven applications (compute services, data services and storage services). Today, we introduced the world to a revolutionary system concept that makes AI computing radically easier, more scalable and more efficient than ever before: **The VAST Data Platform**. This is an idea conceived in late 2015 and purpose-built for the age of AI; it's a concept we then built into a company that was founded the same month and year that OpenAI was founded.

We are VAST Data, and this is the story we've wanted to tell you since December of 2015.

The idea is relatively simple.

Let's build a global computing and storage platform that organizes data from the natural world, a machine (in software) that enriches and applies structure to unstructured data to give AI applications the tools needed to derive greater insights from data.

To truly build such a system, a system engineered for continuous global inference and learning, we needed to throw convention out the window and start from a blank sheet of paper. In doing so, the VAST Data Platform has been conceived as a single monolithic data-driven system that seeks to break so many of the tradeoffs that have led to decades of compromise and point solutions.

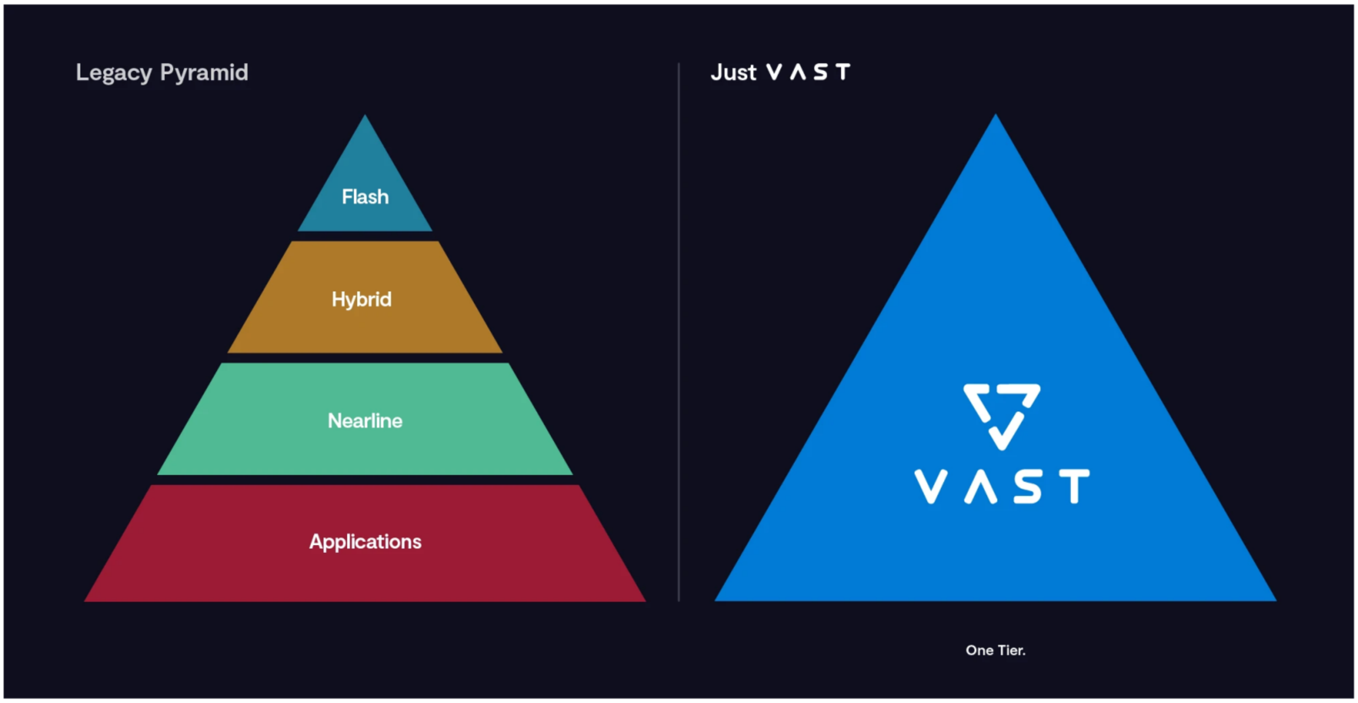
In this blog, I'll attempt to articulate the infrastructure tradeoffs that we've been working to eliminate now that we are able to unveil our entire Data Platform concept. Keeping with the theme of universal constructs, let's talk about gravitational collapses.

Collapsing Storage Tiers

When we started, we realized that deep learning is essentially applied statistics… an approach to computing where vast datasets can fuel learning engines by giving the best representation of the natural world. Flash is an essential component for deep learning, for this reason there is no HDD-based solution that Nvidia allows, for example, for their SuperPOD environments.

When we started VAST, the cost of legacy all-flash infrastructure was 10x the cost of HDD-based alternatives. No one was thinking of flash for the capacity tier. On the other hand, if it would be possible to build archives from flash, there would be no need for any faster storage tiers and it would be possible to just compute right from the archive. At the time, there were many new storage startups working on making all-flash even faster, VAST was the only company making all-flash practically affordable.

How do we do it? By combining a new approach to data reduction with an architecture that embraces the idea of using low-cost, laptop-grade flash we've built a new approach to data storage infrastructure that is now becoming the file and object foundation of the modern data center… **the VAST DataStore**. Formerly referred to as Universal Storage, the efficiency characteristics of this distributed file and object storage system are married with the scale and resilience of the VAST DASE architecture to make it possible to build a single tier of affordable exabyte-scale flash within the data center. When your archive is built from flash, then infrastructure no longer fights against the random read challenges introduced by deep learning and deep data analysis workloads.



Collapsing Database Tiers

The VAST Data Platform offering has taken the market by storm. After selling 10 exabytes of software to power a wide variety of enterprise and research applications, it's become one of the fastest growing products in IT infrastructure history. In 2019, we were both proud and disappointed that IDC declared that the product was "the storage architecture of the future"... proud because of the prestigious distinction, but disappointed because we knew it could be so much more and we didn't want the market to make too many preconceptions about who we were to become.

Deep Learning pipelines require a variety of data management tools. Unstructured data is the foundation for storing data that comes from the natural world, and the power of AI is to understand the context of files that have no schema in order to create structure from unstructured data. While file systems are critically important (it's the only required data system in a SuperPOD, for example), databases are equally important. Unstructured is the raw content, and structure becomes the corresponding that comes from data labeling, data prep and AI inference.

Now - to build a system that can constantly learn, we realized that database services were as critical as unstructured data services and that we needed to challenge fundamental design assumptions of database management systems (DBMSs) in order to create a system that could ingest multi-variant data streams in real-time while also providing the foundation for instantaneous structured queries that could correlate data all the way to the archive. Given the scale we operate at, the concept of a transactional and analytical system needed to be reimagined from the ground up.

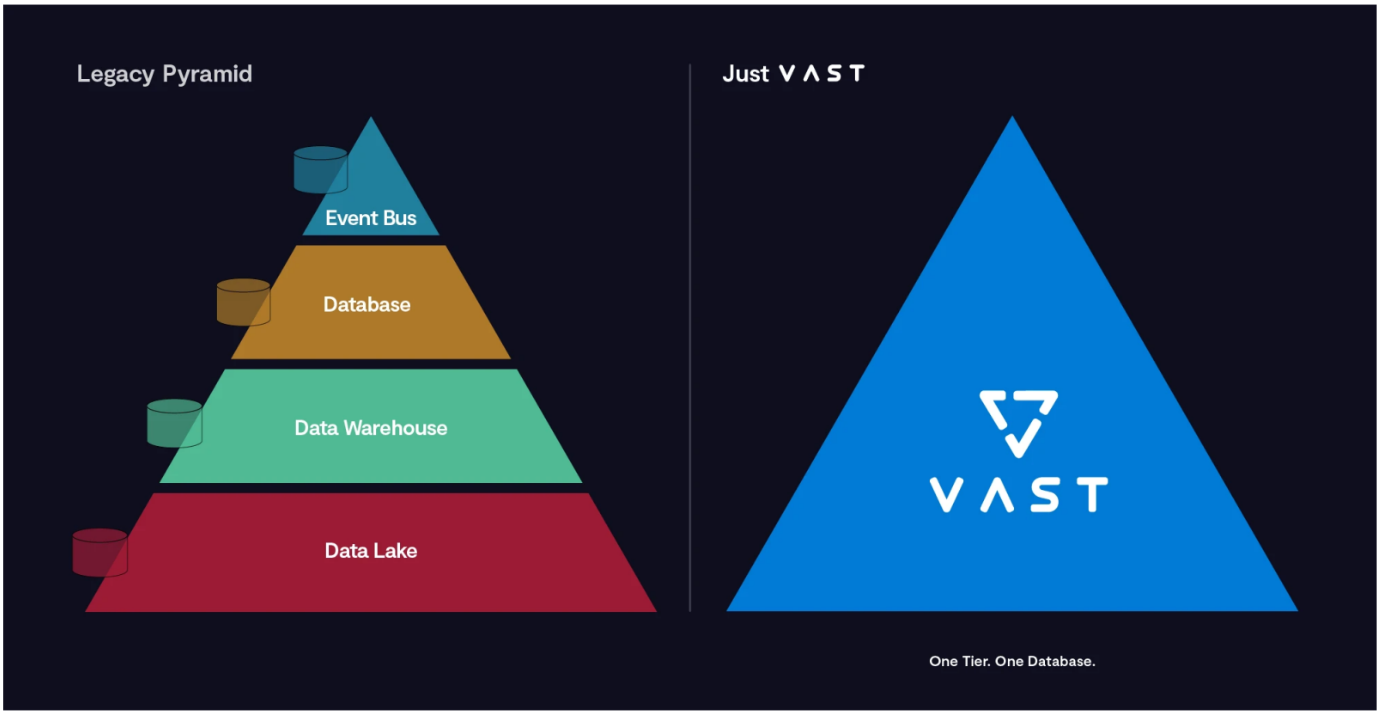
How do we do it? First, we now offer a tabular format within the system. Tables are simply a new data format presented by the VAST Data Element Store (our storage namespace). This table store can be accessed by a version of SQL we've developed (VAST SQL) and also works with other modern query engines such as Apache Spark and Trino. Second, we've built a new database architecture that writes and updates small records into a large persistent memory buffer (storage class memory) in record form, and then migrates records into fine-grained columnar database objects that are radically smaller than a classic Parquet or Orc row group. Granularity equals query precision and speed. Since we've built a 32KB columnar object that is 4,000x smaller than a conventional table partition, our approach results in the ability to service filtered queries up to 100x faster than legacy data lake approaches. Oh, ACID updates are also easy since our architecture is log-structured (writes in free space) and since it's easy to update 32KB columnar objects vs re-writing large and versioned Parquet objects. No more database vacuuming, no more worries about updates.

Let's jump into a time machine and go back to the early '70s, the time before data warehouses. If the database systems of the 1960's and 1970's were able to transact **and** scale their data **and** could efficiently handle queries, would technologies such as data warehouses and data lakes ever be needed? Likely not. This is the place we are aiming to get back to by calling our approach to storing and analyzing tabular data **the VAST**

DataBase. The VAST DataBase is the world's first transactional system that marries ACID transactional semantics that scale across an nearly-infinite number of tables while enabling the system to query datasets that extend from real-time streams all the way to exabyte scale archives.

One tier.

One database.



Collapsing Files & Tables

For decades, databases have always been independent from the file and object storage systems they've been deployed to catalog. Yet, content does not exist in a vacuum, and context has always been managed from some other system of record because no product was ever built to bring together structured and unstructured data into one unified system that could transact, read, analyze and store data all the way to exabyte proportions. Until now.

One of the major unifications that happens with the VAST Data Platform is the synthesis of content and context. At a high level, we think of most database services to be extensions of our data catalog concept.

- A social networking site, for example, can store their photos as objects in the VAST DataStore, serve data globally with the VAST DataSpace and keep the enriched metadata in VAST DataBase. In the past, they'd need to build an object storage system, combine this with a CDN and then deploy a series of NoSQL databases to keep track of objects. Now: simple.

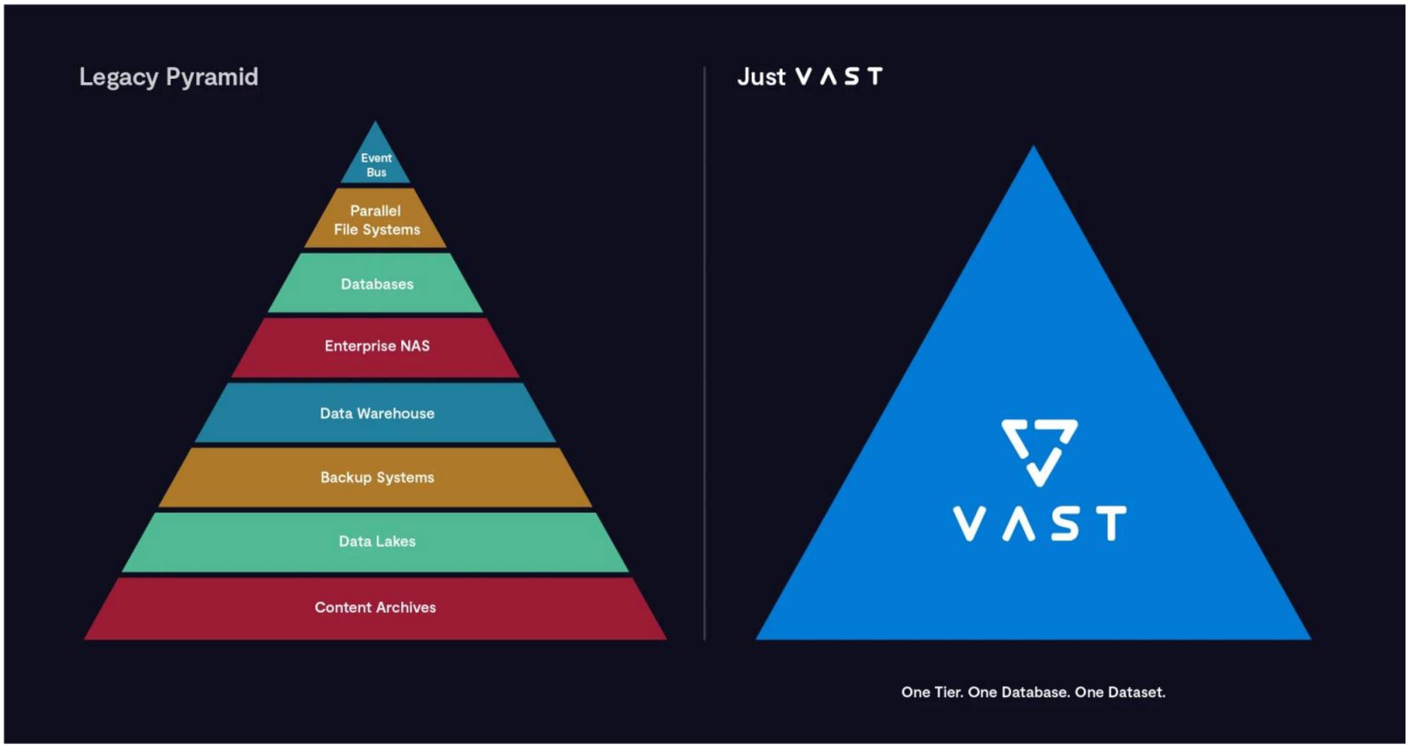
- A deep learning pipeline requires a series of databases to capture data labels and to build training example indices that are used for managing training pipelines. These systems sit alongside object-based storage archives and parallel scratch file systems for AI scratchpads. Now the DataStore delivers both high-capacity and high-performance file access (collapsing tiers) and the DataBase provides the semantic layer for training data that eliminates the need for SQL engines and HDD-based data lakes. The Platform's Cataloging services ensure the database is never out of sync with the content store. Now: simple.

There have been several famous attempts to evolve databases into unstructured databases (my term), and none have succeeded. However, by planning for a unified data management architecture that scales and blurs the lines between transactional analytics, we believe that smart architecture can make the impossible possible.

One tier.

One database.

One dataset.



Collapsing Namespaces

Next, as we thought about deep learning, we thought about dataset size and gravity.

Modern deep learning applications learn from data that comes from the natural world. Computer vision (sight), natural language processing (sound), large language models (free text) all combine to create datasets that are so large that customers cannot easily move them across different data centers. How large? VAST works with organizations who have single clusters that reach into the 100PBs scale, and they are now working with us to build toward exabyte-scale. These massive payloads hold the keys to the future by providing the insights that systems can access to learn and infer from data.

It's impossible to easily move petabytes or exabytes of data when you want to ensure global access across a multitude of data centers. Global data replication systems create copies in every data center, which is not necessarily practical when dealing with this level of scale. Generally a single replica copy is sufficient to ensure application availability, and any additional copies are superfluous. On the other hand, cloud-based file and database management tools are more suited to providing global data access by applying caching techniques to make read operations fast at the edge of the network.

Reads are easy to handle since the different edges of the network don't need to worry about consistency or transaction management. Today's CDN services can have 1,000s of data centers that accelerate reads at the edge - that problem is well solved.

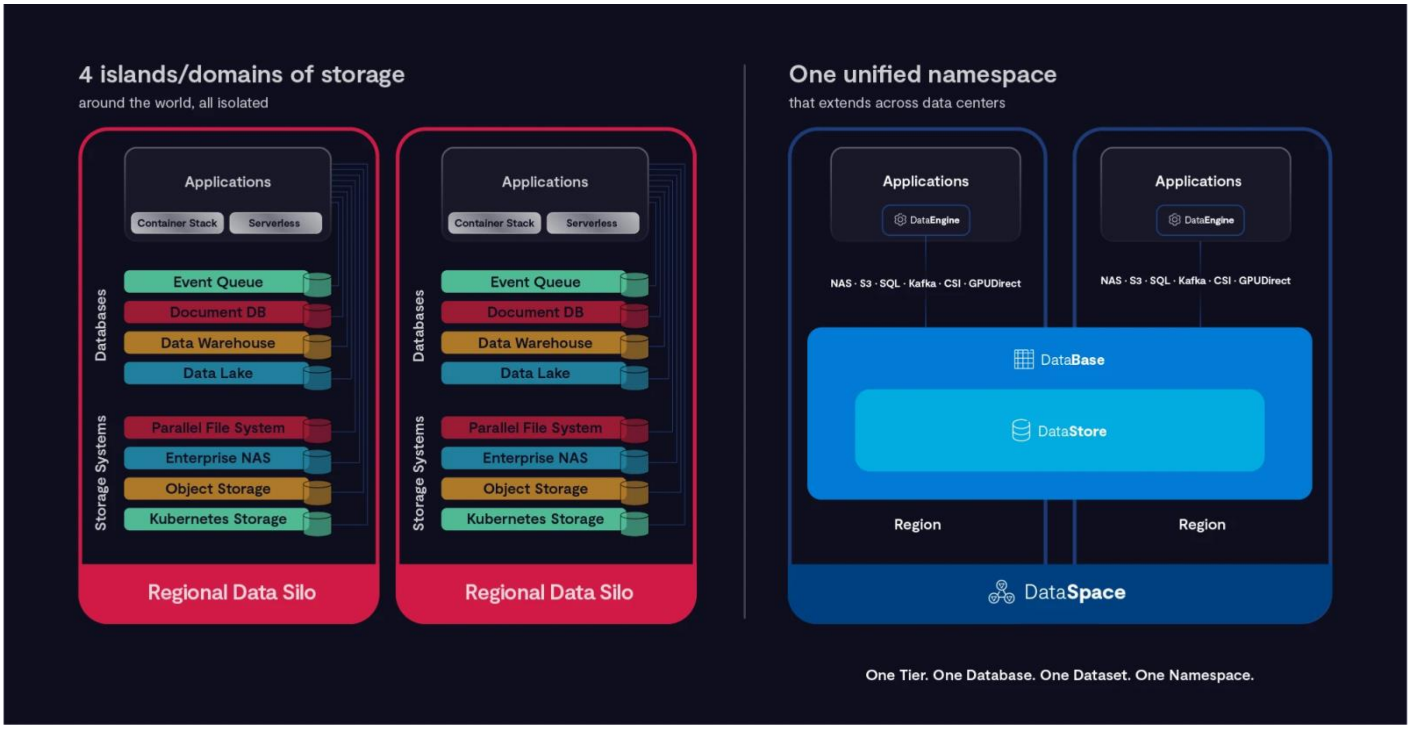
The challenge we thought about was the challenge of global transactional consistency enforcement. If you build a federation of systems that all provide the ability to access a global corpus of data, any site that transacts into this data needs to be coordinated across other sites that also want a coherent view of that data. Historically, transactions that happen across a global data namespace need to be synchronized across any number of locations and need to be managed and enforced by some central data owner that is responsible for arbitrating transactions across a global network. While this approach of central transaction management ensures strict consistency, this comes at the cost of transactional performance - since every site then needs to write at the speed of the wide area network as they check their data across the WAN into a remote data owner. While this might work for document management, our conclusion was that a global collection of AI computers could not afford to write at WAN speeds.

In an effort to break the tradeoff between strict consistency and high-performance read-write access at the edge of the network, we've pioneered a new approach to global data access that we call the **VAST DataSpace**. How do we do it?

- First, VAST clusters can peer to any other VAST cluster to create a data sharing relationship across global networks. These systems will subscribe to each other's data with a selective level of granularity, taking the peering down to the path-level (directory-level, bucket-level, table-level). The system provides the ability to create either a syncing relationship between sites or to let any site take a global, writeable snapshot clone of a path in case

- Each site then caches a consistent view of a path's top-level metadata - such that any site can consistently search through the namespace by starting from a strictly-consistent point. Additional buffering happens for read operations using local caching, global prefetch and tools to pre-warm cache for applications that will require low time-to-first-byte.

- Finally, we've implemented a new approach to consistency management that eliminates the challenges of global consistency management by rethinking where lock enforcement happens. Inspired by some developments in the web3 space, we've invented a new decentralized lock manager that allows transaction state management to flow across a network of clusters and be enforced where the state is changing. This can only be done by bringing lock management all the way to the file, object & table level… and with this decentralized and fine-grained approach to lock management, all of the sites can write as they please while, in the random case of a write collision, we can still enforce strict levels of global consistency.



One Tier. One Database. One Dataset. One Namespace.

When we bring it all together, users just get fast access to data from any point on the network.

One tier.

One database.

One dataset.

One namespace.

Collapsing Clouds

Clouding isn't easy. Every customer we talk with about "doing cloud right" spends much time talking about replatforming applications and going cloud native. Cloud strategies either fail to launch or fail to be agile when customers run up against two laws of gravitation:

- Data has gravity, which is compounded by egress toll gates imposed by cloud vendors
- APIs create gravity, by requiring applications to support to platform-specific APIs

When we started building the VAST DataSpace, one of the aims of this effort was to also allow customers to extend namespaces across different public cloud platforms. How do we do it? Today we're announcing support for AWS, Azure and Google - I'm sure we'll add others as customers express interest in other cloud environments.

One of the leading cloud vendors once asked me why we're building support for their platform when they already have different point solutions that (in total) compete to varying degrees with our offering. The answer was easy: choice and fit.

By adding support for leading public clouds, the DataSpace creates a platform abstraction where applications no longer fight against API or data gravity. You can just run your applications using a common set of data APIs on the platform you choose, whenever you choose. No refactoring. No lock-in.

Now - as for "fit". What we're also finding is that many of the toolsets offered in leading cloud platforms (unstructured data systems, structured data platforms) were almost all built in an era that pre-dates deep learning, and they often lean on hard drive-based constructs as well as other architectural decisions that result in all of the tradeoffs VAST is trying to break. So, with today's announcement we're also happy to be able to talk with customers about running AI applications in cloud against data infrastructure that is fit for purpose.

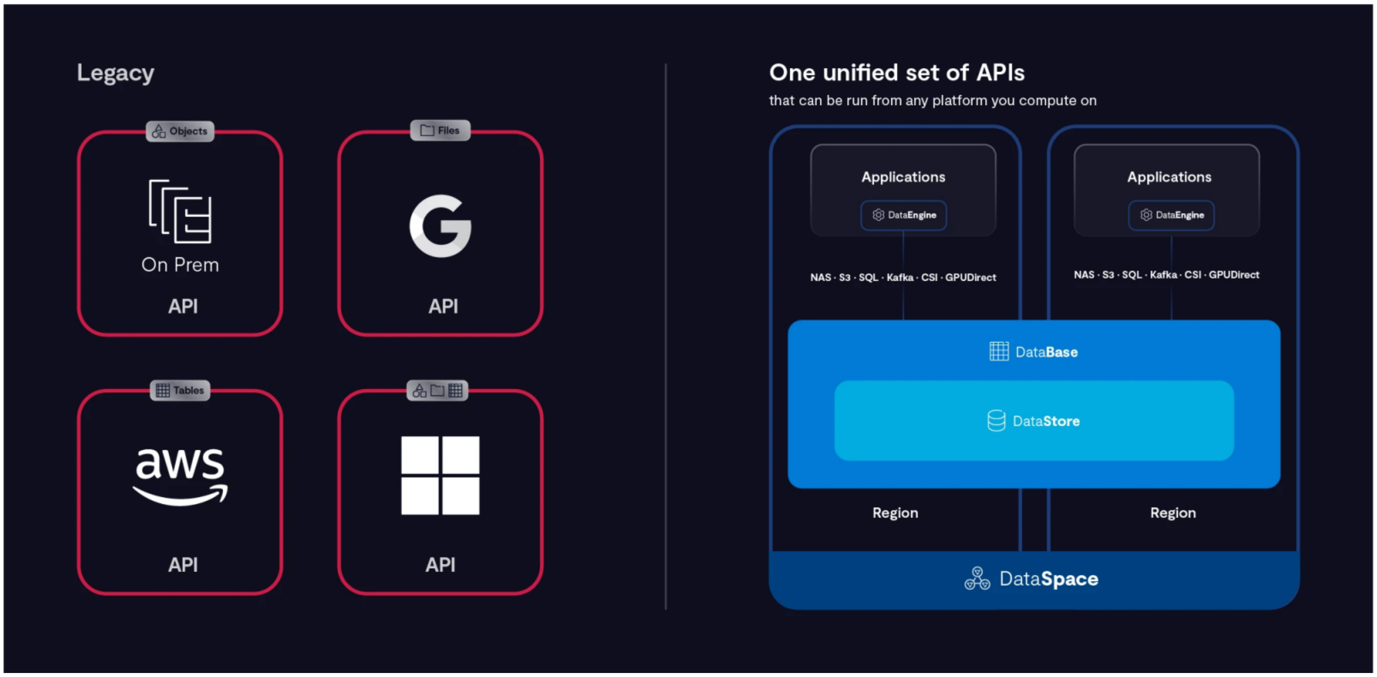
One tier.

One database.

One dataset.

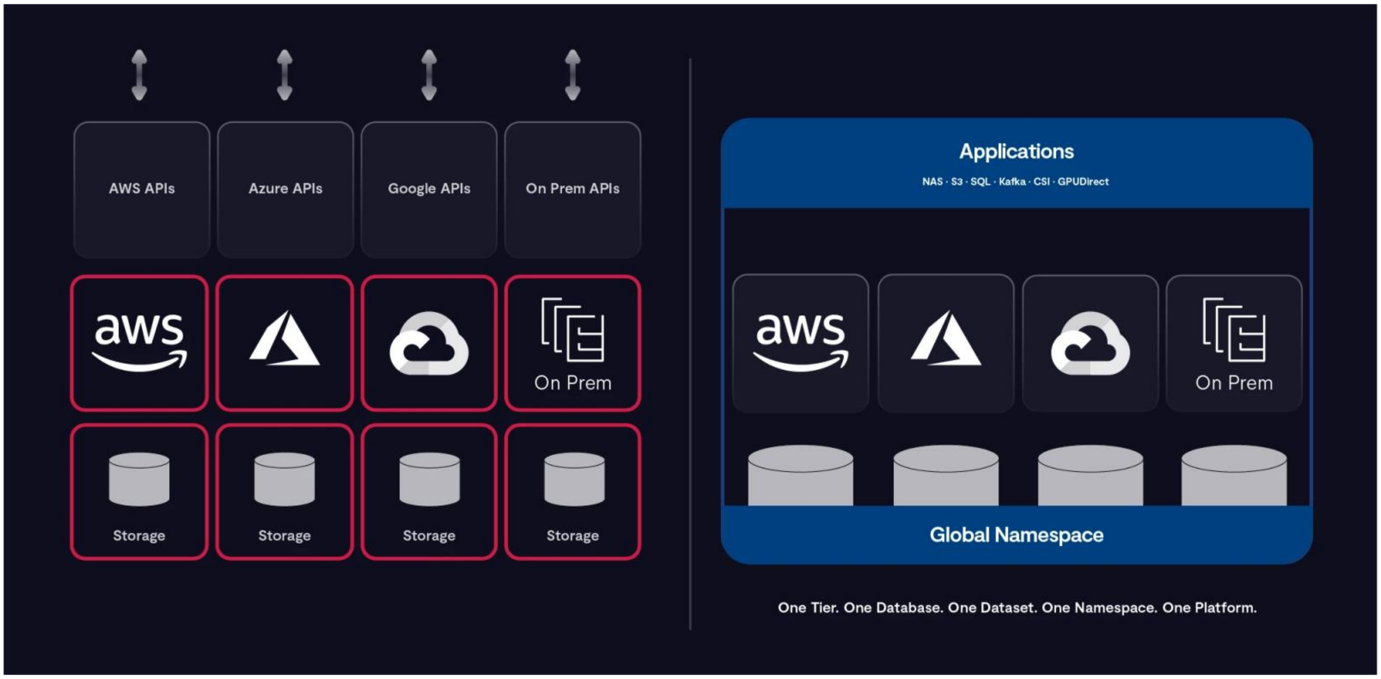
One namespace.

One platform.



One unified set of APIs

that can be run from any platform you compute on



One Tier. One Database. One Dataset. One Namespace. One Platform.

Collapsing Application Definitions

OK, this last one can take a lot of twists and turns but I’m now in too-verbose territory so I’m going to keep it simple.

Since the dawn of computational data analytics (I'm going back to the late 1970s here), enterprise software architecture has largely fallen into one of two camps:

- Event-Centric Systems are largely responsible for capturing and processing data streams as they happen. Event architectures are tightly related to transaction processing and messaging systems and typically limit the ability to analyze data in a system of record. Since data lakes and data warehouses have never possessed the ability to handle scalable transactions, event architectures typically only support correlative analysis of real-time data against a limited data set that can fit in an event queue. Today, Apache Kafka is the standard for open and scalable event processing.

- Data-Centric Systems, on the other hand, are designed to capture and harvest insights from large datasets… datasets so large that they would overwhelm a conventional database or event bus architecture. Datasets that need to be queried using tools optimized for deep data analysis (typically columnar data orientation) using data-parallel analytics methods across a wide collection of computers. Popular data-centric programming languages include SQL and R.

The implicit tradeoff in all of this is that applications can't run deep correlative analysis against events that are happening in real-time, nor can deep analytics infrastructure process events in real time. The result is a dichotomy between systems that requires organizations to integrate disparate systems that all step down to their lowest common denominator while also forfeiting the opportunity to marry real-time and big data.

Today, we're announcing the final element of the VAST Data Platform story, **the VAST Data Engine**. The engine is the last piece of the 'thinking machine' puzzle (The Quest to Build Thinking Machines (vastdata.com) that refines raw unstructured data into insights that can be discovered by structured query language via the VAST DataBase. The VAST Engine is a computational framework that runs in containers across the VAST DataSpace. More than just a container stack, the Engine introduces support for triggering functions across a dynamically scalable set of CPU, GPU and DPU resources from edge to cloud.

The DataEngine will ship in 2024, and we'll be unveiling a lot more details as the offering gets closer to general availability, but today I'd like to preview a capability that I just call VAST Streams (actual name: TBD). The Engine not only creates a functional programming environment to refine data through deep learning training and inference functions, but it also extends the type of data ingest that the system can support to also provide a Kafka connector that will allow data to be topics to be captured as real-time tables.

How do we do it? The DASE architecture is built with a persistent write buffer that allows a batch of messages to be written directly into a VAST table in record form. As mentioned above, a background process then converts these records into a flash-optimized columnar object format that makes it easy to run accelerated queries at any scale. Triggers then call functions as data is streamed into a topic, and event streams just become another source of data that can create a continuous loop of data enrichment and processing in the VAST Data Platform.

With this, we hope to break down decades-old application definitions to build systems that can interact with the natural world and create real time associations, realizations and discoveries by comparing new experiences with past learnings.

One tier.

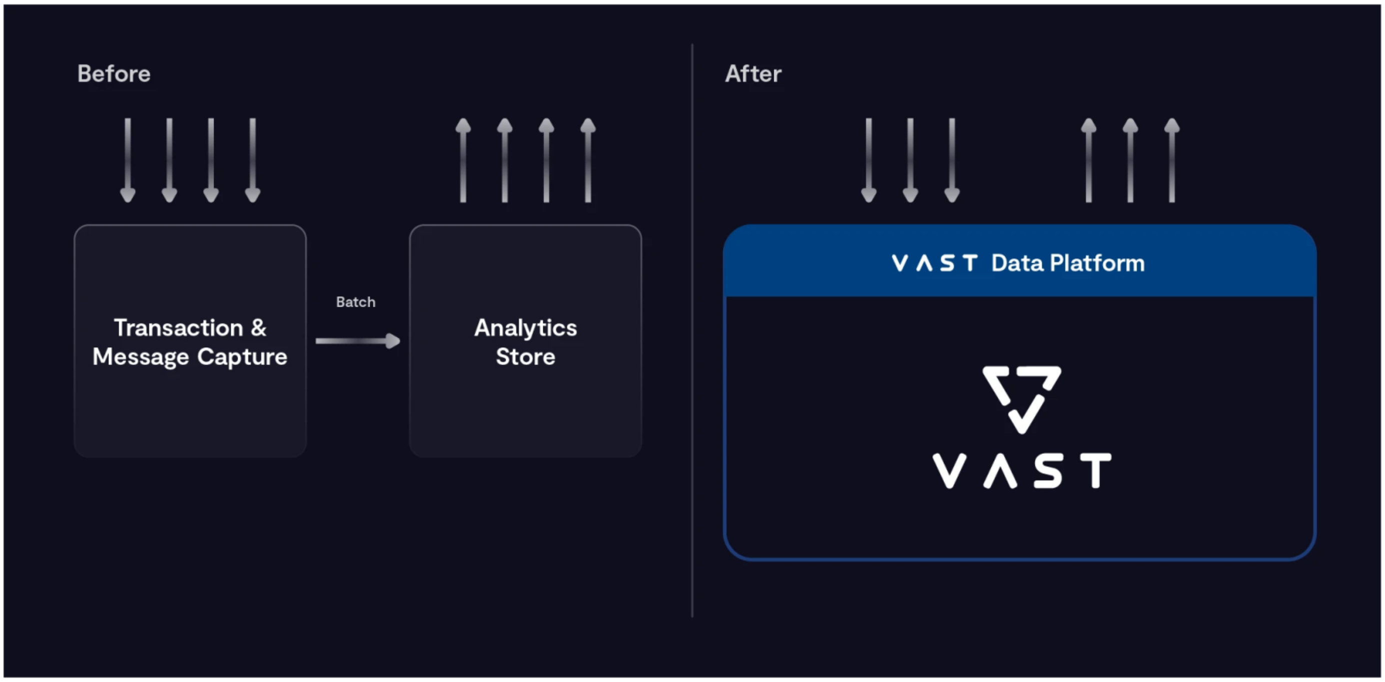
One database.

One dataset.

One namespace.

One platform.

One perpetual learning machine.



We are VAST and this is our vision for the future of computing.

This is the VAST Data Platform.

Thank you.

- Jeff