AWS re:Invent

DECEMBER 1 – 5, 2025 | LAS VEGAS, NV

ANT201

What’s new in search, observability & vector databases with OpenSearch

Carl Meadows

Director of Product

AWS

Mukul Karnik

Director of OpenSearch

AWS

Corey Nolet

Principal Architect

Nvidia

Popular use cases

High-performance, customizable **Search** with seamless hybrid vector and lexical search for highly relevant results

Backend for **AI/ML powered applications** with a powerful vector database, integration with AI frameworks like LangChain and native ML composable workflows

Observability and **Security Analytics** allowing you to detect, identify, and resolve operational and security issues across your applications and infrastructure

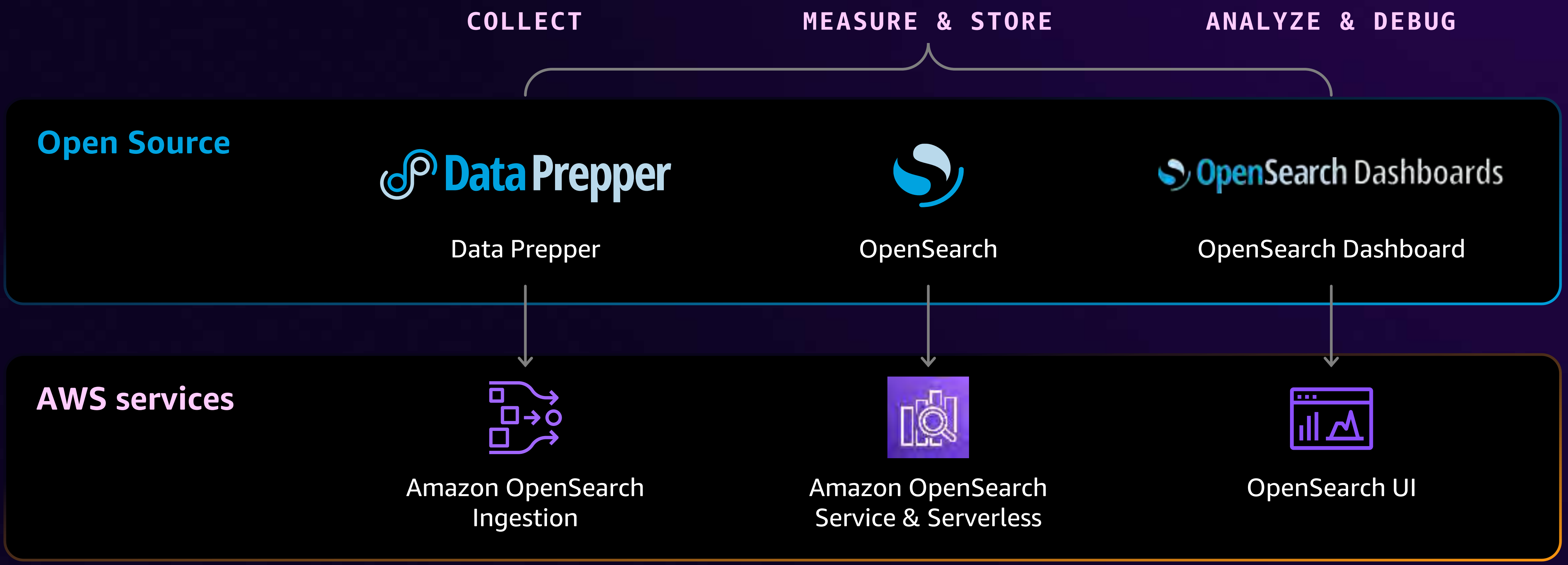
Search

AI/ML-powered applications

Observability

Security analytics

OpenSearch: Leading innovation across AWS and Open Source



COLLECT

MEASURE & STORE

ANALYZE & DEBUG

Open Source

Data Prepper

OpenSearch

OpenSearch Dashboard

AWS services

Amazon OpenSearch Ingestion

Amazon OpenSearch Service & Serverless

OpenSearch UI

OpenSearch

OpenSearch is a **community-driven, open-source** search, analytics, and vector database platform with integrated tools for observability, security, visualization, and AI-powered applications

Roadmap:

OpenSearch
SOFTWARE FOUNDATION
THE LINUX FOUNDATION
aws
IBM
SAP
Uber
aiven
ByteDance
DTEX
OpenSource Connections
Aryn
Canonical
eliatra
portal26
ATLASSIAN
DigitalOcean
NetApp Instaclustr
seacom

OpenSearch by the numbers

1.3B+
Project downloads

1M+
Monthly page views
For opensearch.org

100+
Solution providers

3K+
Active contributors

400+
Active organizations

In the top
20
Of LF projects by contributor activity

130+
GitHub repositories

4K+
Slack workspace members

7K+
User forum members

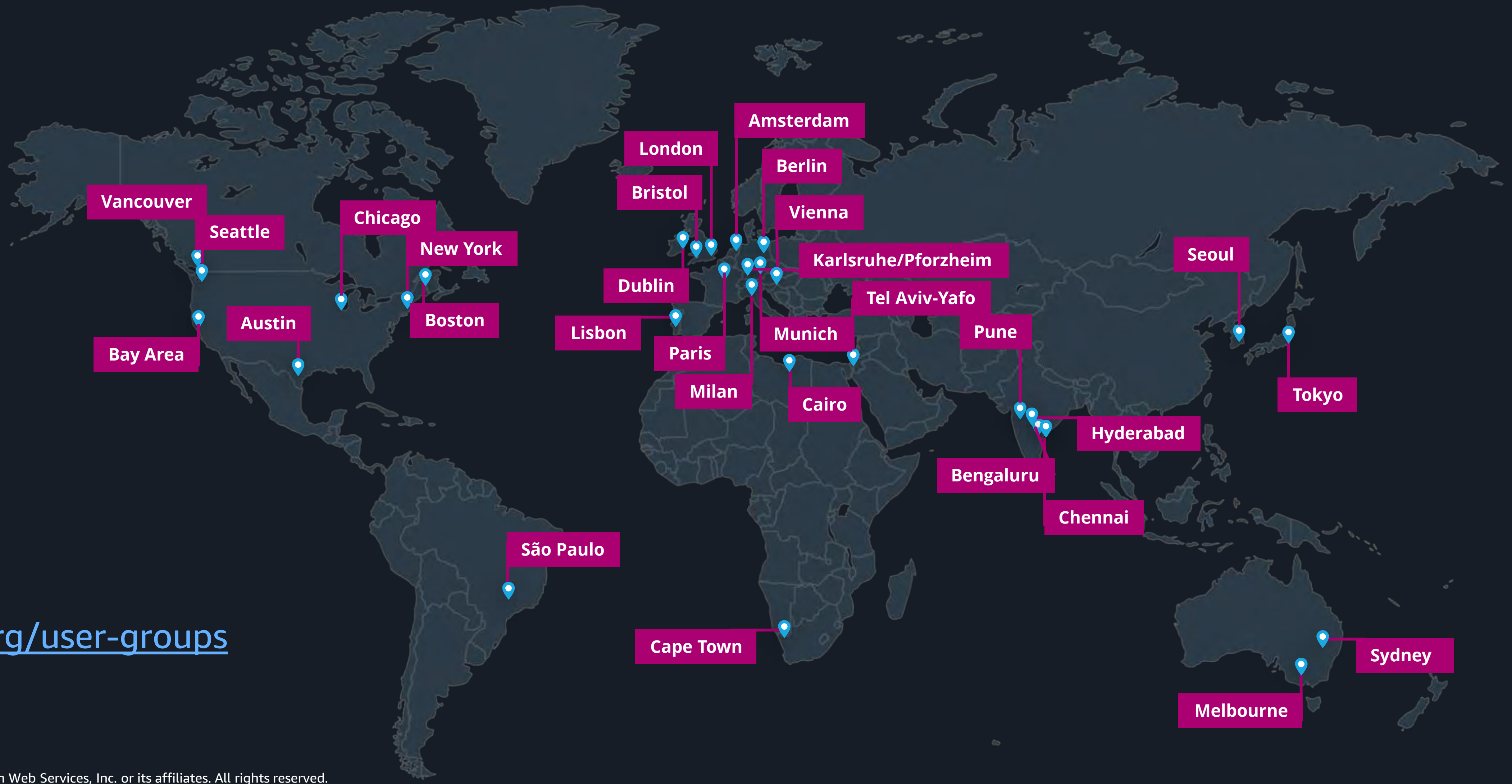
OpenSearch Global Community Engagement

✓

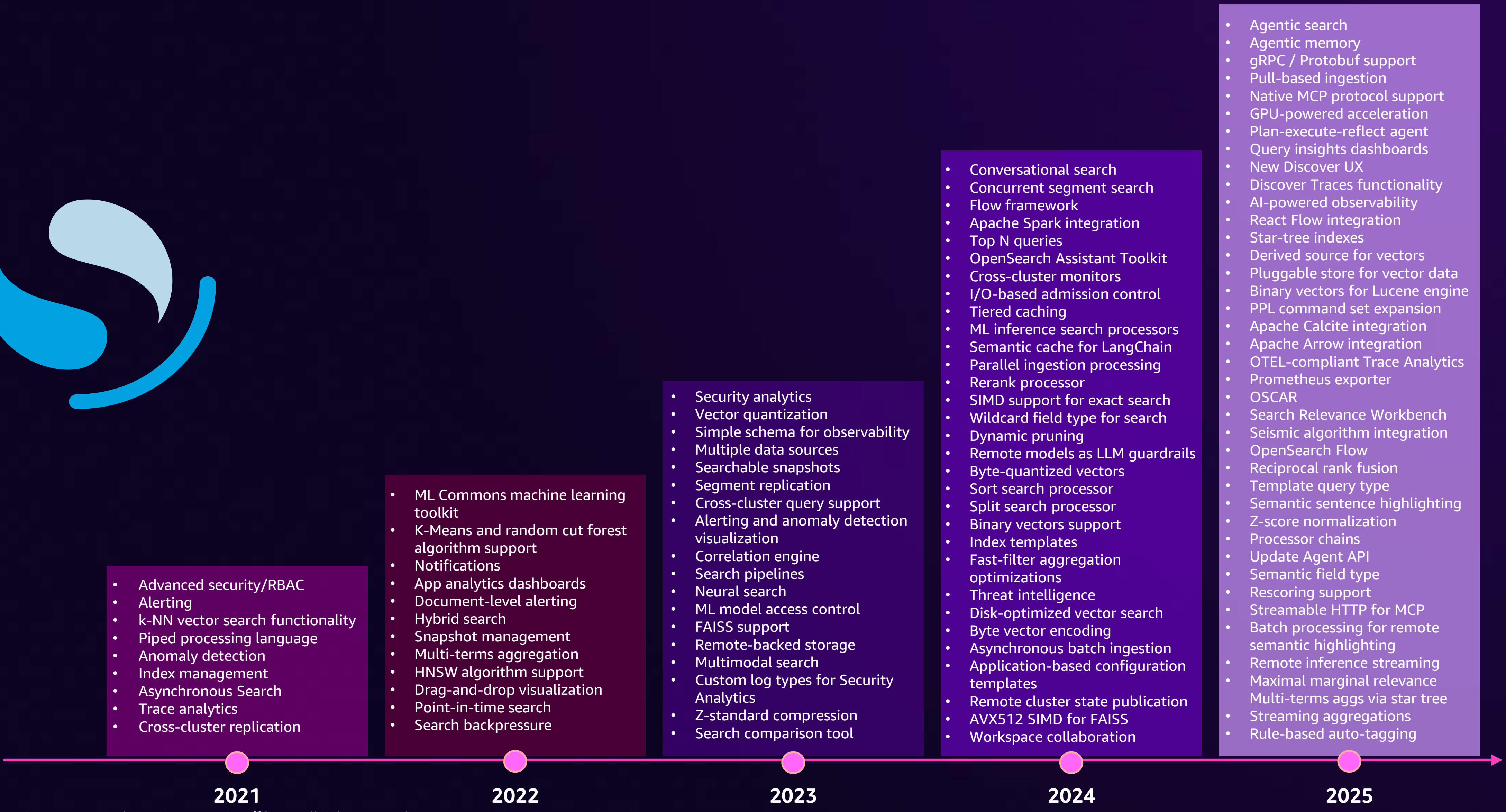
30 user groups

18 countries

opensearch.org/user-groups



OpenSearch Project Innovations



2021

- Advanced security/RBAC
- Alerting
- k-NN vector search functionality
- Piped processing language
- Anomaly detection
- Index management
- Asynchronous Search
- Trace analytics
- Cross-cluster replication

2022

- ML Commons machine learning toolkit
- K-Means and random cut forest algorithm support
- Notifications
- App analytics dashboards
- Document-level alerting
- Hybrid search
- Snapshot management
- Multi-terms aggregation
- HNSW algorithm support
- Drag-and-drop visualization
- Point-in-time search
- Search backpressure

2023

- Security analytics
- Vector quantization
- Simple schema for observability
- Multiple data sources
- Searchable snapshots
- Segment replication
- Cross-cluster query support
- Alerting and anomaly detection visualization
- Correlation engine
- Search pipelines
- Neural search
- ML model access control
- FAISS support
- Remote-backed storage
- Multimodal search
- Custom log types for Security Analytics
- Z-standard compression
- Search comparison tool

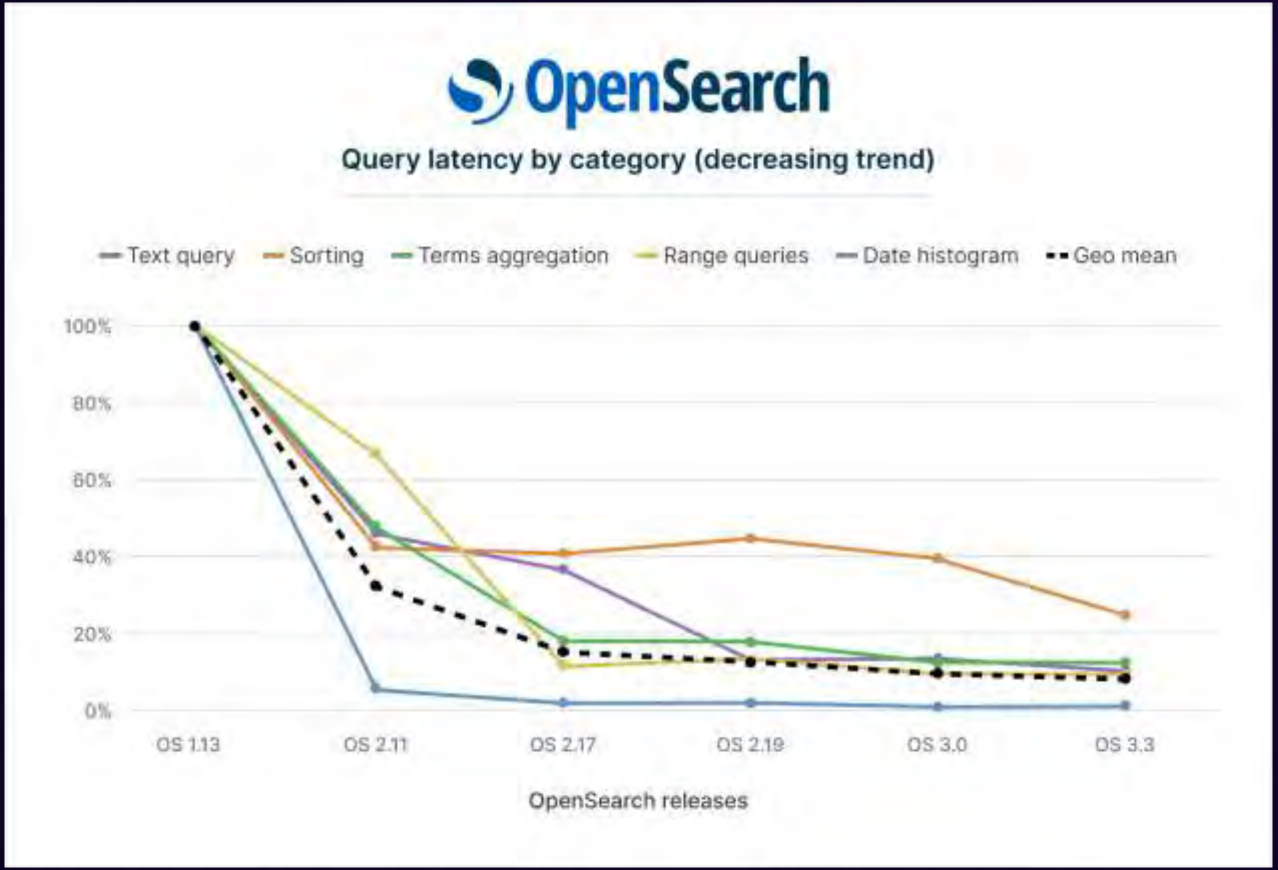
2024

- Conversational search
- Concurrent segment search
- Flow framework
- Apache Spark integration
- Top N queries
- OpenSearch Assistant Toolkit
- Cross-cluster monitors
- I/O-based admission control
- Tiered caching
- ML inference search processors
- Semantic cache for LangChain
- Parallel ingestion processing
- Rerank processor
- SIMD support for exact search
- Wildcard field type for search
- Dynamic pruning
- Remote models as LLM guardrails
- Byte-quantized vectors
- Sort search processor
- Split search processor
- Binary vectors support
- Index templates
- Fast-filter aggregation optimizations
- Threat intelligence
- Disk-optimized vector search
- Byte vector encoding
- Asynchronous batch ingestion
- Application-based configuration templates
- Remote cluster state publication
- AVX512 SIMD for FAISS
- Workspace collaboration

2025

- Agentic search
- Agentic memory
- gRPC / Protobuf support
- Pull-based ingestion
- Native MCP protocol support
- GPU-powered acceleration
- Plan-execute-reflect agent
- Query insights dashboards
- New Discover UX
- Discover Traces functionality
- AI-powered observability
- React Flow integration
- Star-tree indexes
- Derived source for vectors
- Pluggable store for vector data
- Binary vectors for Lucene engine
- PPL command set expansion
- Apache Calcite integration
- Apache Arrow integration
- OTEL-compliant Trace Analytics
- Prometheus exporter
- OSCAR
- Search Relevance Workbench
- Seismic algorithm integration
- OpenSearch Flow
- Reciprocal rank fusion
- Template query type
- Semantic sentence highlighting
- Z-score normalization
- Processor chains
- Update Agent API
- Semantic field type
- Rescoring support
- Streamable HTTP for MCP
- Batch processing for remote semantic highlighting
- Remote inference streaming
- Maximal marginal relevance Multi-terms aggs via star tree
- Streaming aggregations
- Rule-based auto-tagging

OpenSearch Performance Improvements



11x faster than OpenSearch 1.3

2.5x faster vector search

Amazon OpenSearch Service

Simplify AI-powered search, observability, and vector database operations with a secure, cost-effective managed service

Operational Simplicity

Fully managed OpenSearch in the AWS Cloud. Serverless and managed clusters. API-driven deployments, upgrades, and patches

Performance

10x performance gains from OpenSearch 1.3 to current version. Single-digit ms latencies for lexical and vector queries

Cost efficiency

Storage tiering for voluminous log data, specialized instances, tune cluster sizes or autoscaled to match request traffic

Integrations

Quickly and easily connect all of your data for faster, better insights

100,000+

Monthly active customers processing with **more than 10 trillion requests** per month

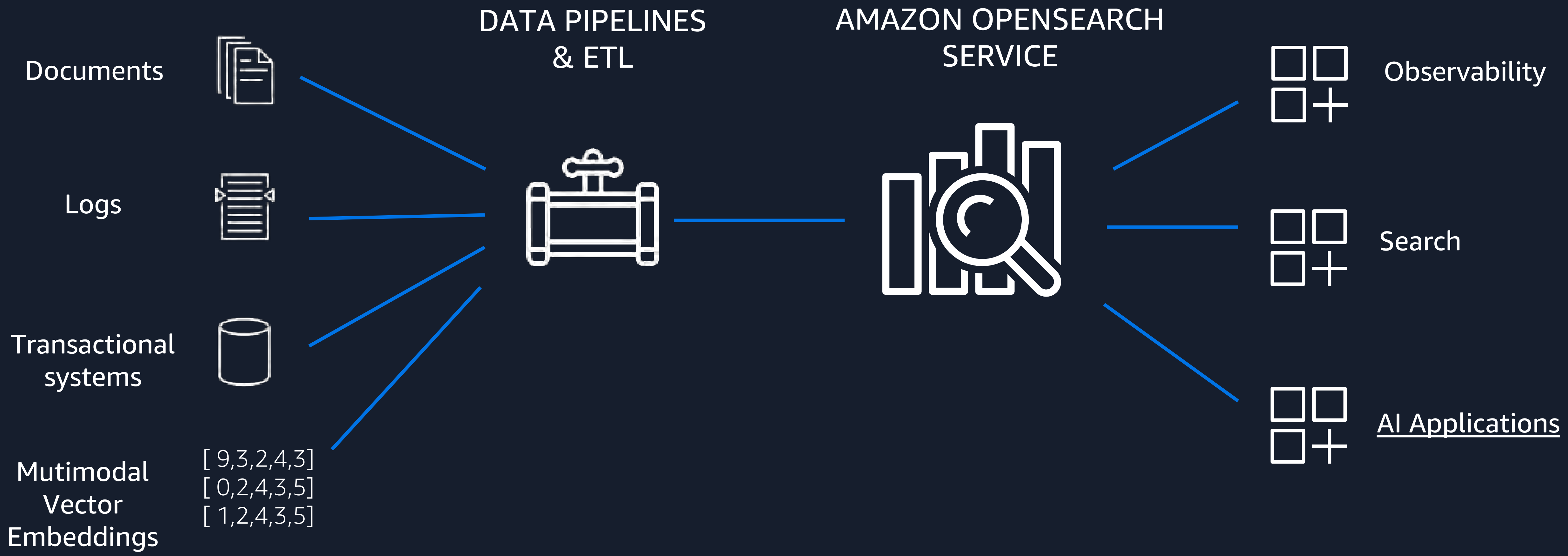
Service-managed patching and upgrades with 24x7 monitoring and self-healing, no down-time upgrades

Fine-grained access control, customer-managed encryption keys, audit logging, IDC/SAML/Cognito/IAM integration

One-click, multi-AZ deployments, up to 99.99% SLA

Specialized instance types using S3 as a backing store and hourly snapshots

Amazon OpenSearch Service landscape



Documents

Logs

Transactional systems

Mutimodal Vector Embeddings

[9,3,2,4,3]
[0,2,4,3,5]
[1,2,4,3,5]

DATA PIPELINES & ETL

AMAZON OPENSEARCH SERVICE

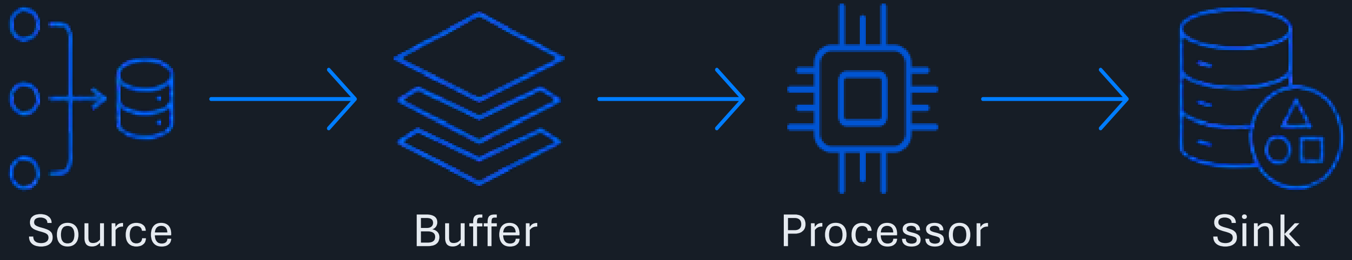
Observability

Search

AI Applications

Amazon OpenSearch Ingestion

POWERED BY Data Prepper



- Native support for wide variety of sources, processors and sinks
- Filtering, enriching, transforming, normalizing, and aggregating data for downstream analytics and visualization
- Inbuilt support for trace analytics, conditional routing, dead letter queues and a lot more features centered around observability

Performance and scale enhancements

Increased OpenSearch Compute Units (OCU) memory

- **15 GB of memory per OCU** (up from 8 GB) at no additional cost, enabling more complex, memory-intensive tasks

Enhanced Autoscaling

- Enhanced autoscaling with new signals such as SQS queue size, buffer lag and HTTP connections for more responsive scaling

OpenSearch Ingestion Connector ecosystem

SOURCES

OBSERVABILITY
- HTTP,
- Otel logs, metrics, and traces

OBJECT STORES
- S3
- S3-SQS

EVENT STREAMS
- Kinesis Data Streams
- Amazon Managed Streaming for Kafka
- Confluent Kafka

SAAS
- **Atlassian Jira**
- **Atlassian Confluence**

DATABASES
- DynamoDB
- DocumentDB
- **RDS/Aurora MySQL**
- **RDS/Aurora PostgreSQL**

MIGRATION
- Elasticsearch
- OpenSearch

SECURITY
- Amazon Security Lake

PROCESSORS

FILTER
- Select entries
- Delete entries
- Anomaly detection

ENRICH
- **AWS Lambda**
- **Batch AI Inference**
- GeoIP

OPEN TELEMETRY
- Otel Metrics
- Otel Traces
- Service Maps

MUTATE
- Add entries
- Decompress
- Flatten
- Rename keys
- Split event
- Obfuscate

ROUTE
- Conditionally route

PARSE
- Grok
- User agent
- parse JSON
- parse XML

SINKS

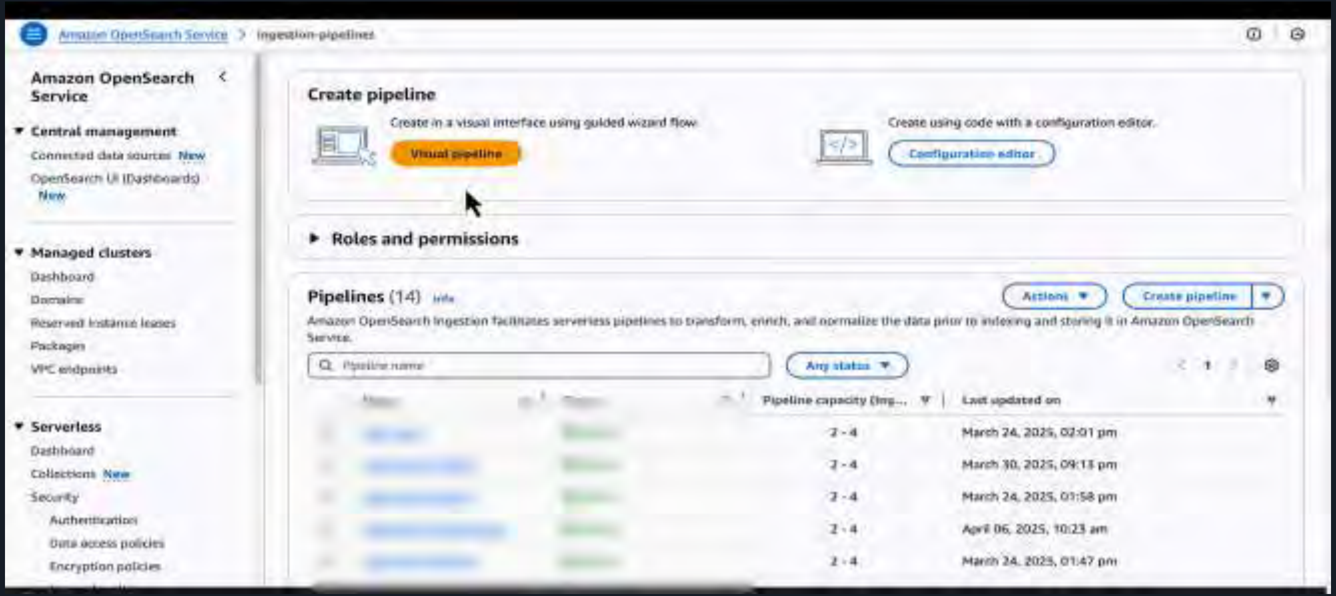
SEARCH
- OpenSearch managed clusters
- Serverless collections

OBJECT STORES
- S3

Improved user experience

Visual Pipeline Builder

- Guided visual workflow via a graphical UI
- Automatic IAM role and permission creation
- Real-time, enhanced validations



Create pipeline

Create in a visual interface using guided wizard flow.

Create using code with a configuration editor.

Pipelines (14) Info

Amazon OpenSearch Ingestion facilitates serverless pipelines to transform, enrich, and normalize the data prior to indexing and storing it in Amazon OpenSearch Service.

[illegible]	[illegible]	Pipeline capacity (Ing...	Last updated on
[illegible]	[illegible]	2 - 4	March 24, 2025, 02:01 pm
[illegible]	[illegible]	2 - 4	March 30, 2025, 09:13 pm
[illegible]	[illegible]	2 - 4	March 24, 2025, 01:58 pm
[illegible]	[illegible]	2 - 4	April 06, 2025, 10:23 am
[illegible]	[illegible]	2 - 4	March 24, 2025, 01:47 pm

Log Analytics & Observability

Log data is increasing rapidly

Downtime costs money (and stress)

$42,000 per hour

Average hourly cost downtime per year

- Interruptions to IT operations
- Opportunity loss, cost of not doing business

87 hours per year

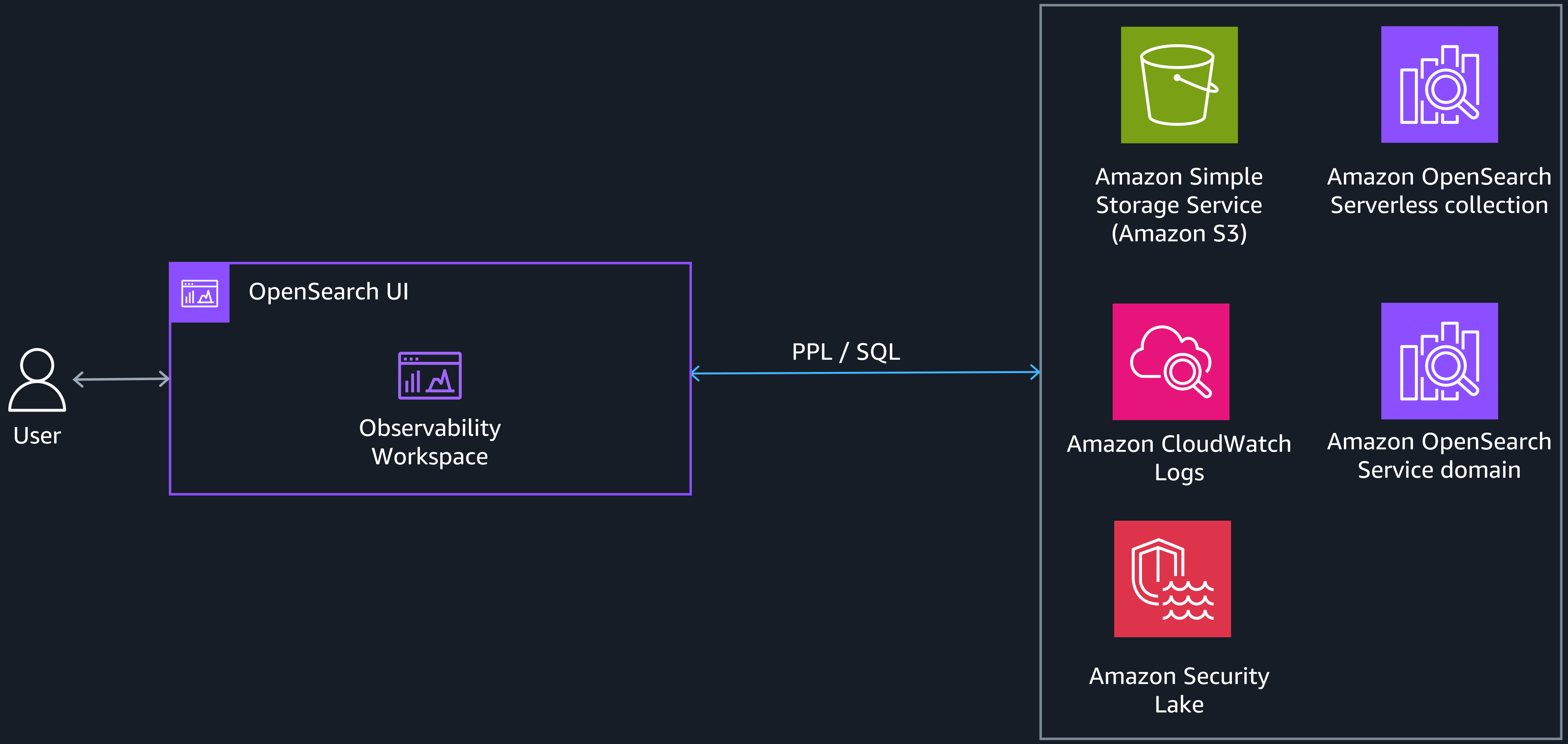
Of downtime per year*

- Penalties of not meeting customer SLAs
- Brand alienation

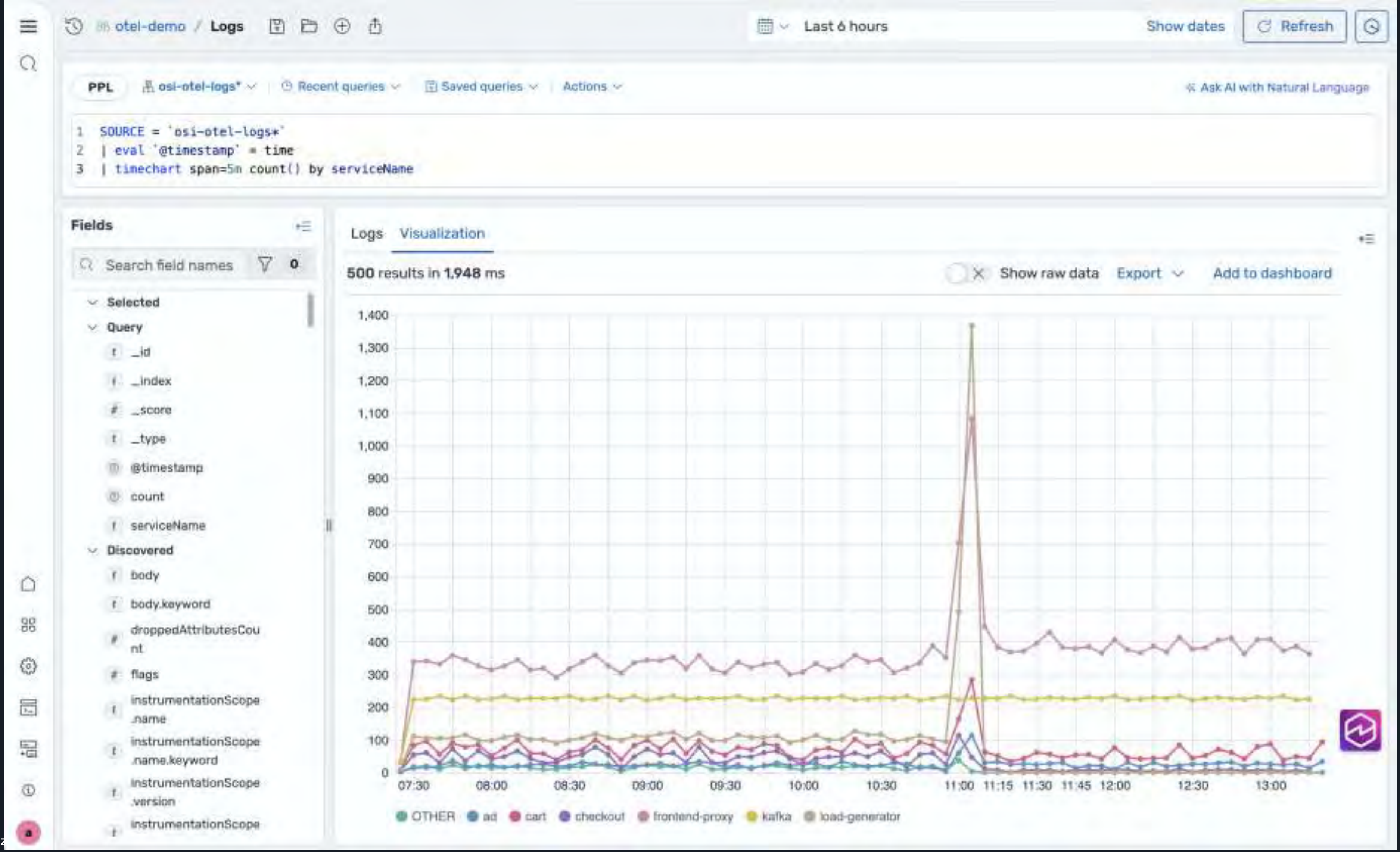
$3.6m+
yearly cost

*Gartner

Single Pane of Glass for Log Analytics


User
OpenSearch UI
Observability Workspace
PPL / SQL
Amazon Simple Storage Service (Amazon S3)
Amazon OpenSearch Serverless collection
Amazon CloudWatch Logs
Amazon OpenSearch Service domain
Amazon Security Lake

OpenSearch Dashboard - Discovery



```
SOURCE = `osi-otel-logs*`
| eval `@timestamp` = time
| timechart span=5m count() by serviceName
```

500 results in 1,948 ms

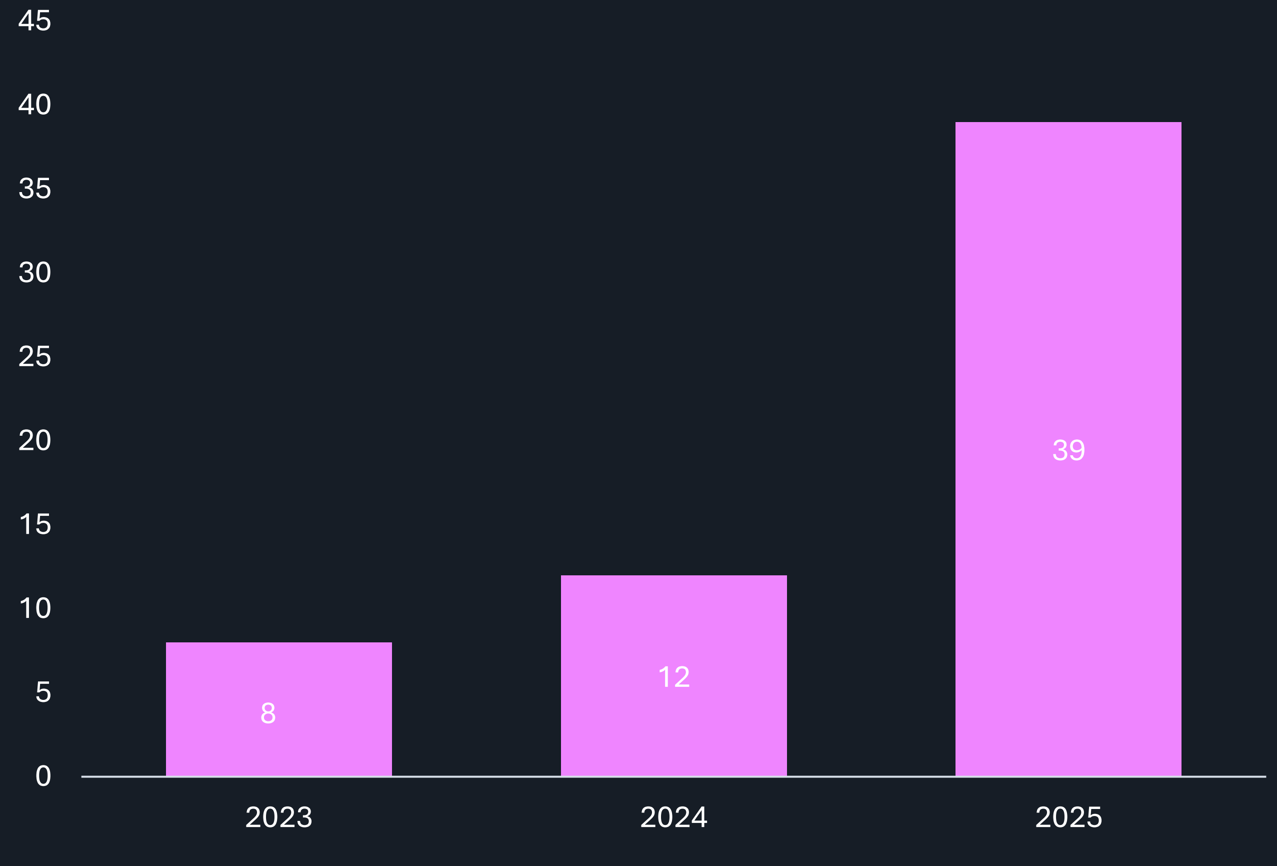
Piped Processing Language (PPL)

```
search command | command 1 | command 2
```

- search
- where
- fields
- dedup
- stats
- sort
- eval
- …
- top
- rare

Many supported commands

Expanded Analytical Toolkit



New Commands

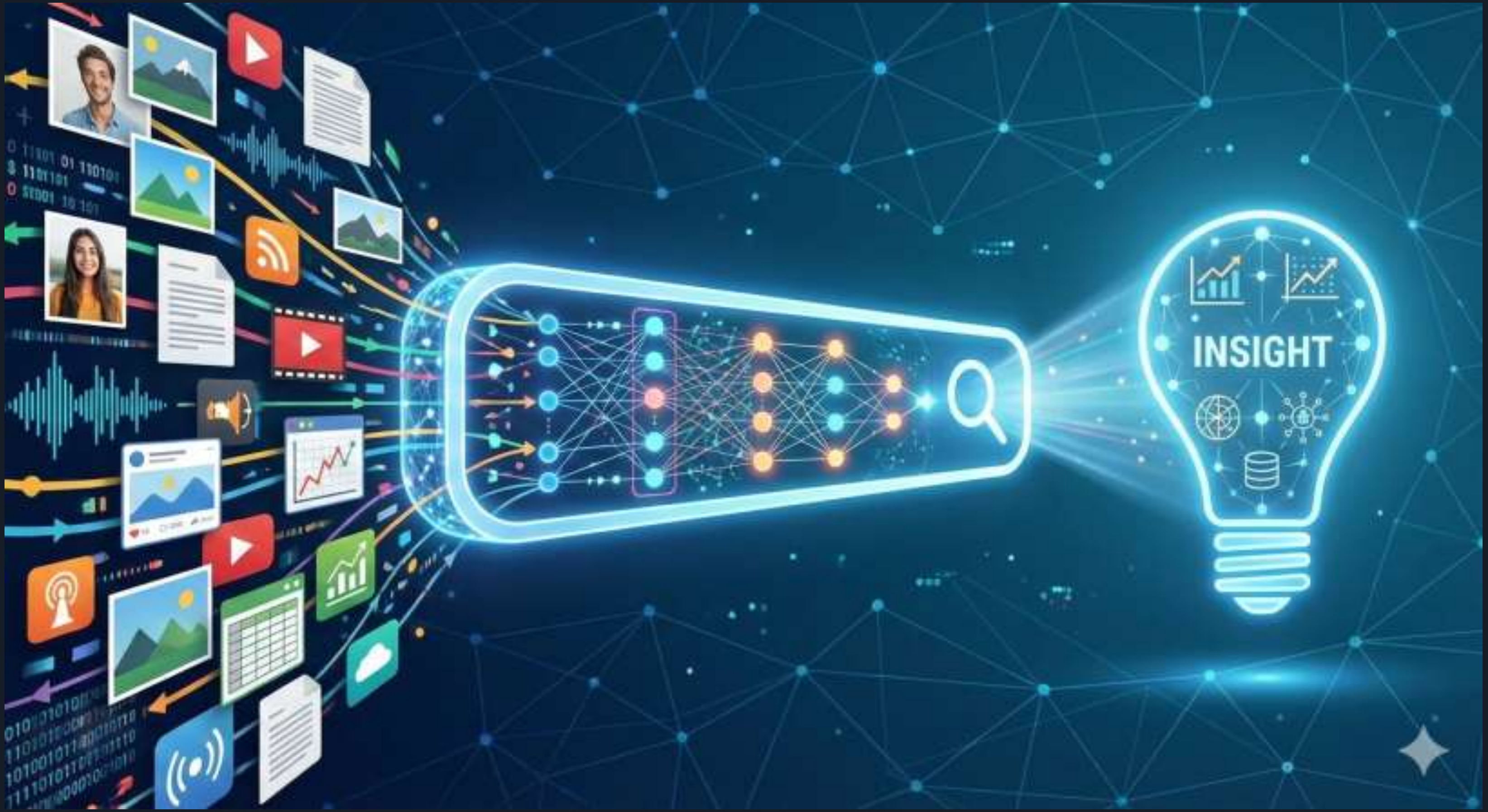
- 35+ commands added
- Filter, extract, and parse unstructured text
- Join datasets together
- Temporal and distribution analysis
- Data transformations at runtime

New Log Discovery Experience Demo

Search and AI

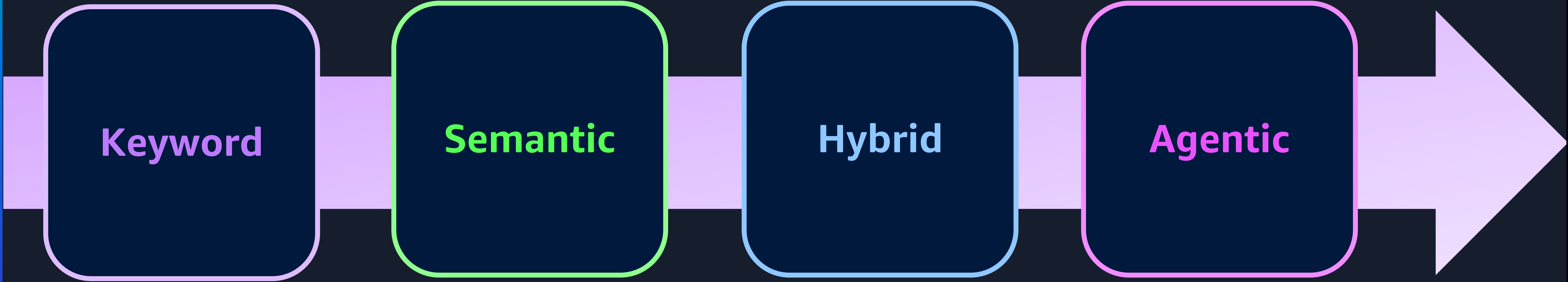
Information retrieval

The process of finding and accessing relevant information



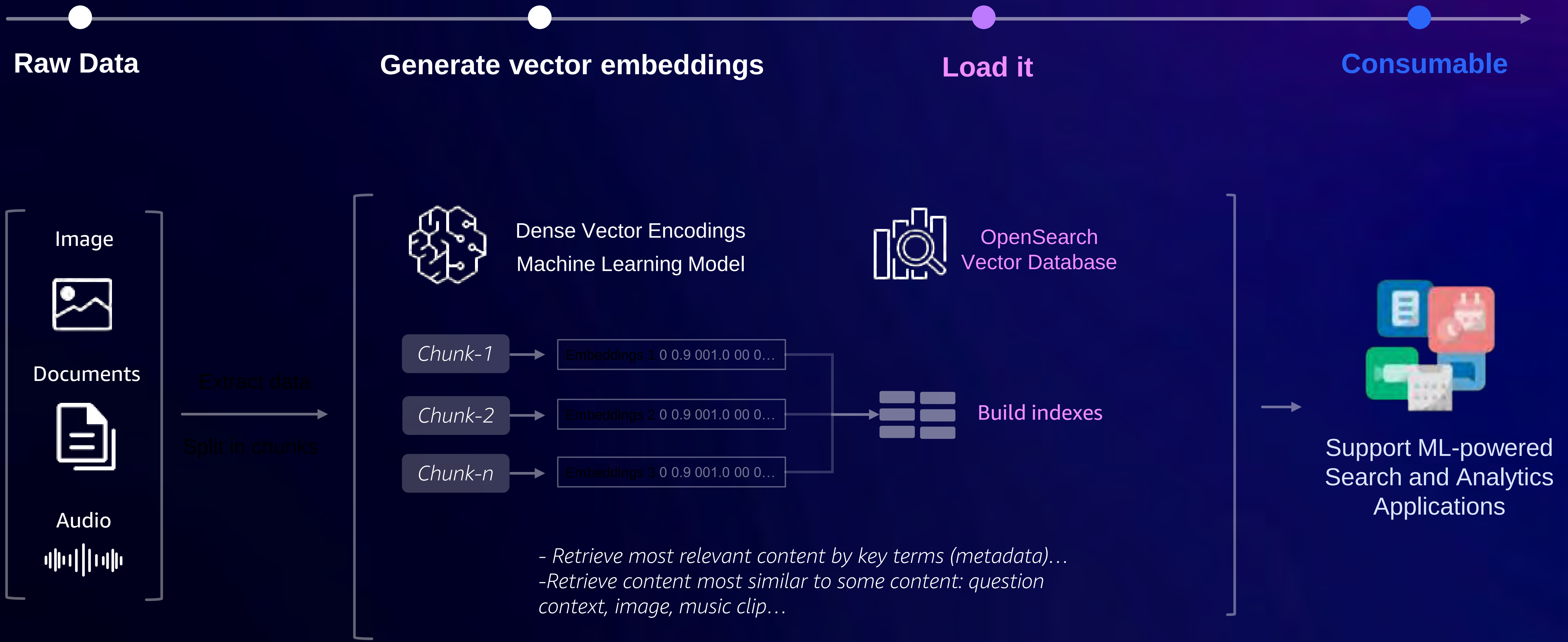
The Evolution of Search

FROM KEYWORDS TO UNDERSTANDING



Search is all about *retrieving relevant information* in response to a user goal

Search Workflow



Raw Data

Generate vector embeddings

Load it

Consumable

Image

Documents

Audio

Extract data

Split in chunks

Dense Vector Encodings
Machine Learning Model

OpenSearch
Vector Database

Chunk-1

Embeddings 1 0 0.9 001.0 00 0…

Chunk-2

Embeddings 2 0 0.9 001.0 00 0…

Chunk-n

Embeddings 3 0 0.9 001.0 00 0…

Build indexes

- Retrieve most relevant content by key terms (metadata)…
-Retrieve content most similar to some content: question context, image, music clip…

Support ML-powered Search and Analytics Applications

NEW

Automatic Semantic Enrichment

BOOSTING SEARCH RELEVANCE

serverless

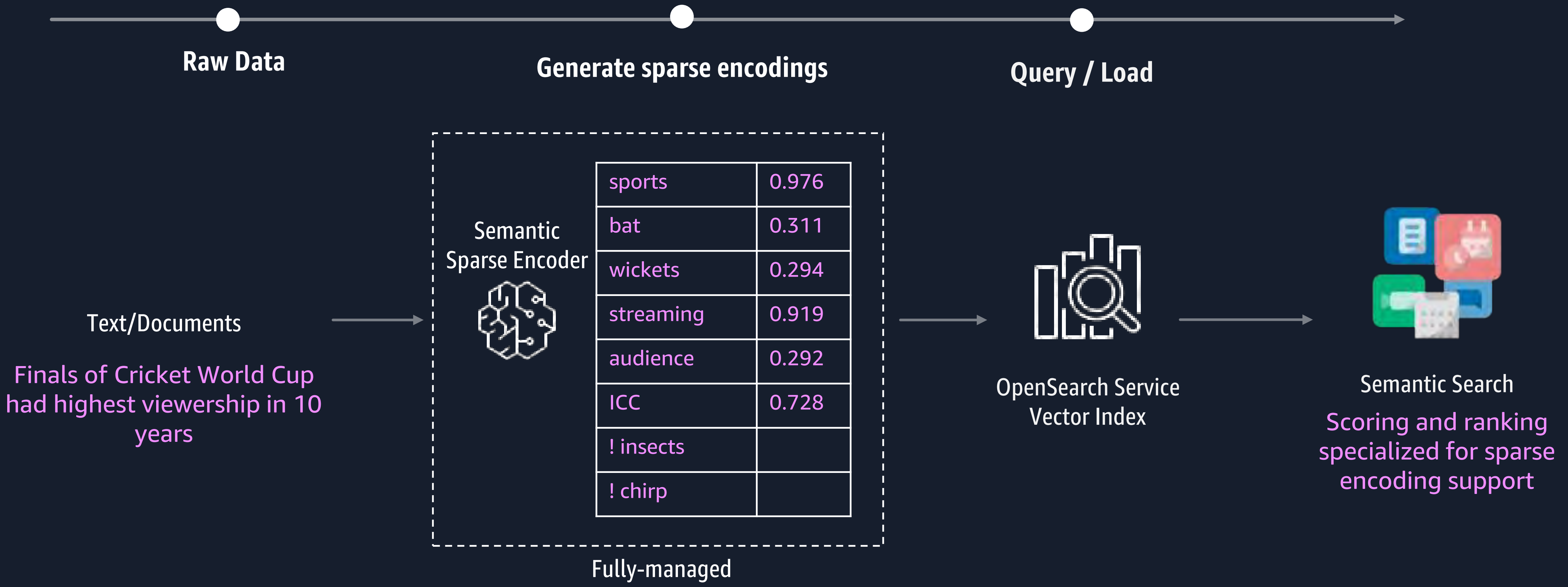
Enhanced Search Relevance
Sparse model improves search relevance without latency impact. Bridges lexical and semantic search capabilities.

Simplified Implementation
Out-of-the-box semantic indexes. No ML model hosting or management required.

Pay-as-You-Use Pricing
Low usage charges during ingestion only. Eliminates continuous costs and infrastructure overhead.

Multi-language Support
Supports 15 languages including English, Chinese, Arabic, Spanish, and more.

Automatic Semantic Enrichment



Raw Data

Generate sparse encodings

Query / Load

Text/Documents

Finals of Cricket World Cup had highest viewership in 10 years

Semantic Sparse Encoder

sports	0.976
bat	0.311
wickets	0.294
streaming	0.919
audience	0.292
ICC	0.728
! insects	
! chirp	

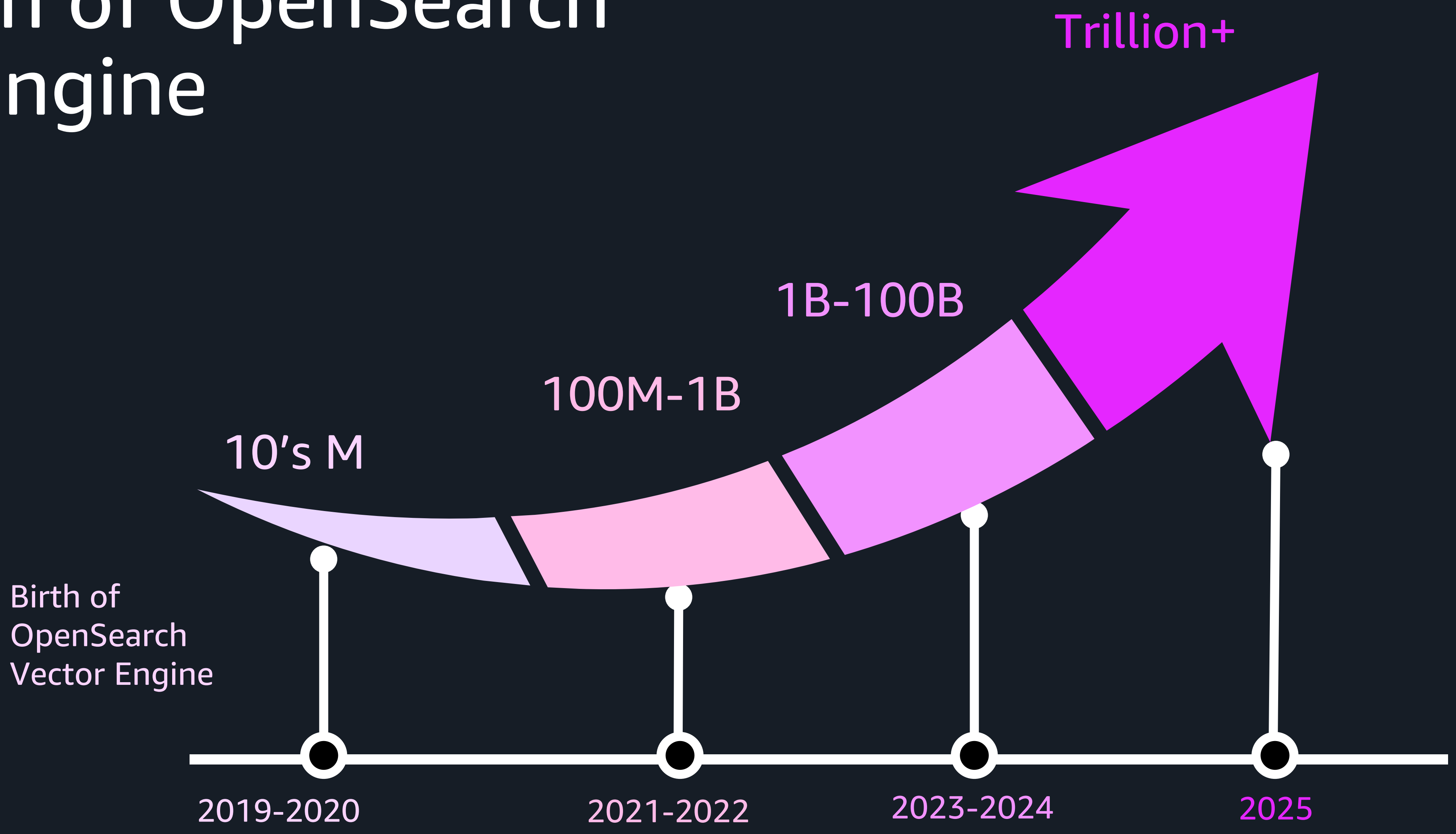
Fully-managed

OpenSearch Service Vector Index

Semantic Search

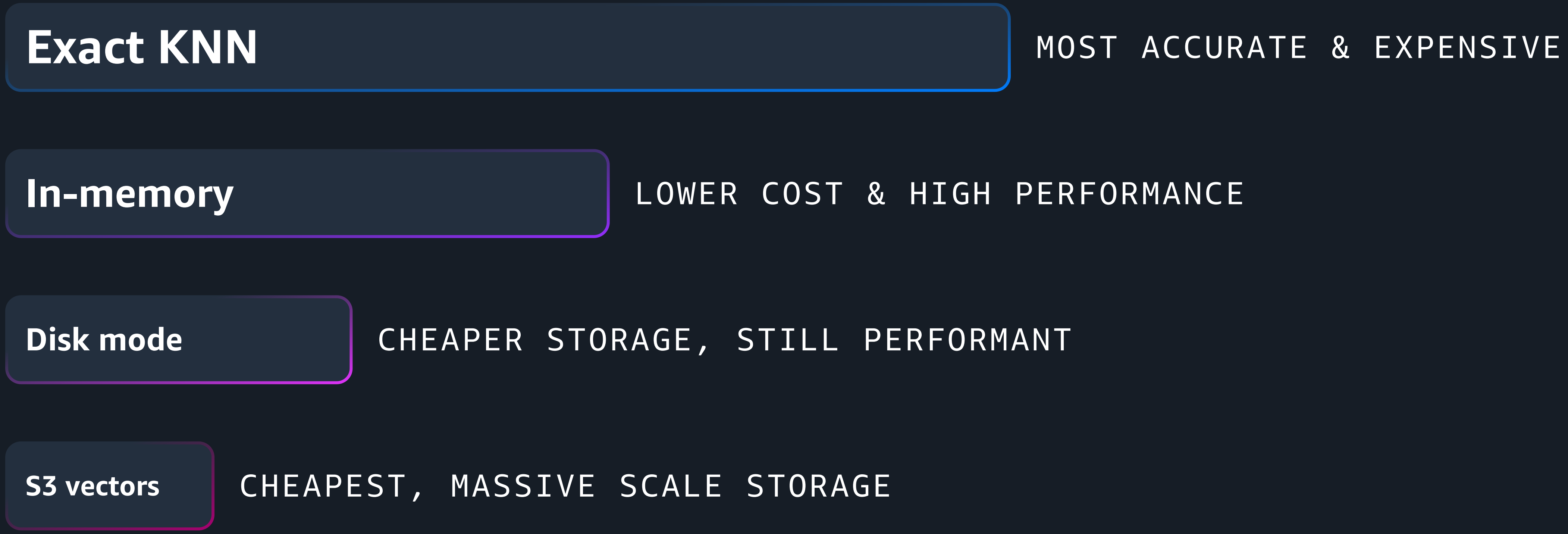
Scoring and ranking specialized for sparse encoding support

Evolution of OpenSearch Vector Engine



Tiered vector storage

COST REDUCTIONS AT MASSIVE SCALE



Exact KNN MOST ACCURATE & EXPENSIVE

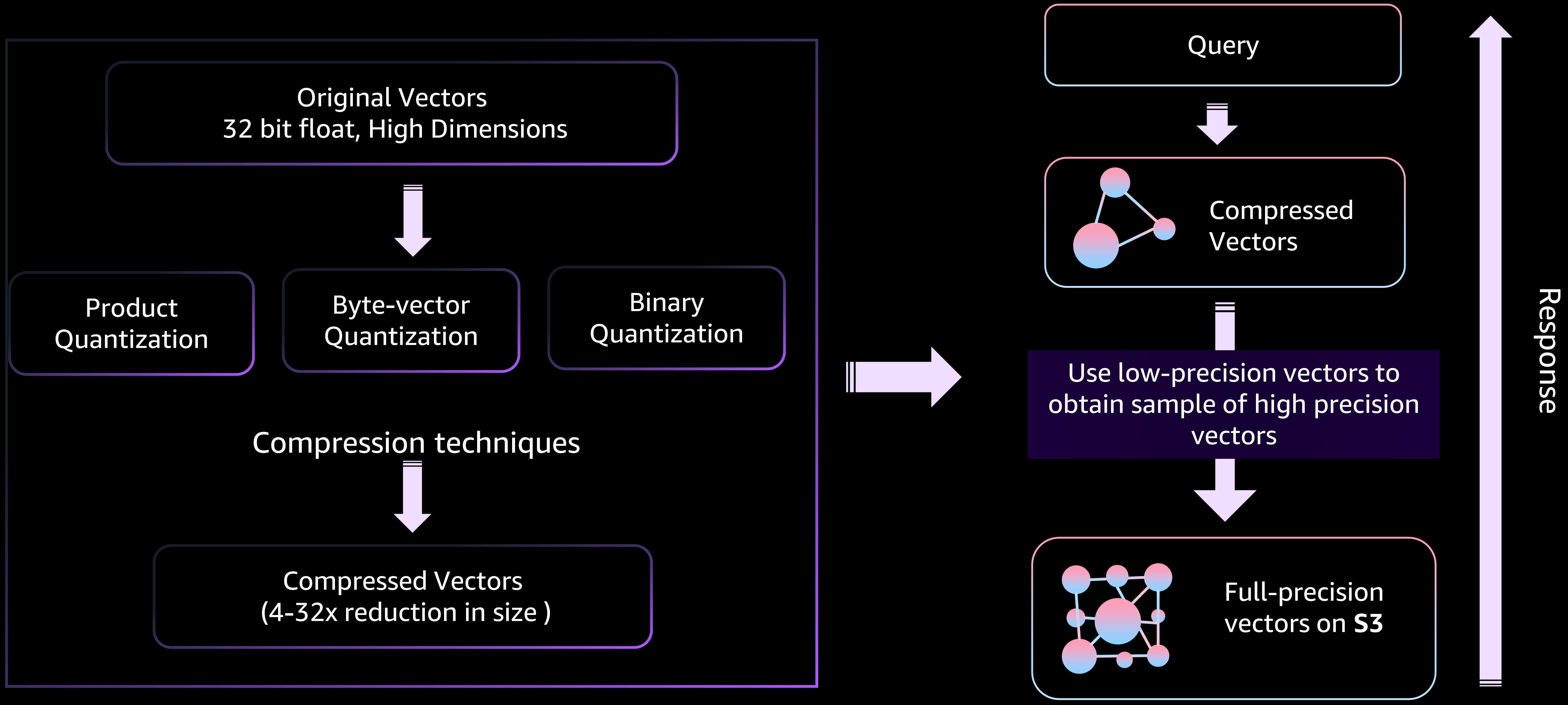
In-memory LOWER COST & HIGH PERFORMANCE

Disk mode CHEAPER STORAGE, STILL PERFORMANT

S3 vectors CHEAPEST, MASSIVE SCALE STORAGE

Disk-Optimized Vector Search

UP-TO-32X MEMORY REDUCTION LEADING TO COST SAVINGS



Original Vectors
32 bit float, High Dimensions

Product Quantization

Byte-vector Quantization

Binary Quantization

Compression techniques

Compressed Vectors
(4-32x reduction in size)

Query

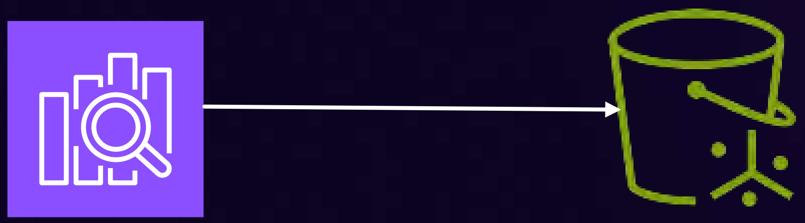
Compressed Vectors

Use low-precision vectors to obtain sample of high precision vectors

Full-precision vectors on **S3**

Response

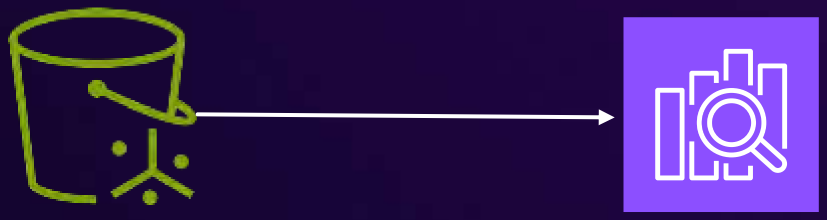
Performance and cost-optimized tiering with Amazon S3 integration



Amazon OpenSearch Service managed cluster

S3 Vectors

To reduce cost, set engine type to “s3vector” in field mapping to store the vectors in service managed S3 vector bucket



S3 Vector Index

Amazon OpenSearch Serverless

To improve latency, single click point in time export to OpenSearch Serverless collection

NVIDIA journey with Amazon OpenSearch Service

Corey J. Nolet

Principal Architect

Vector search & database, data mining & ML

NVIDIA, Inc.

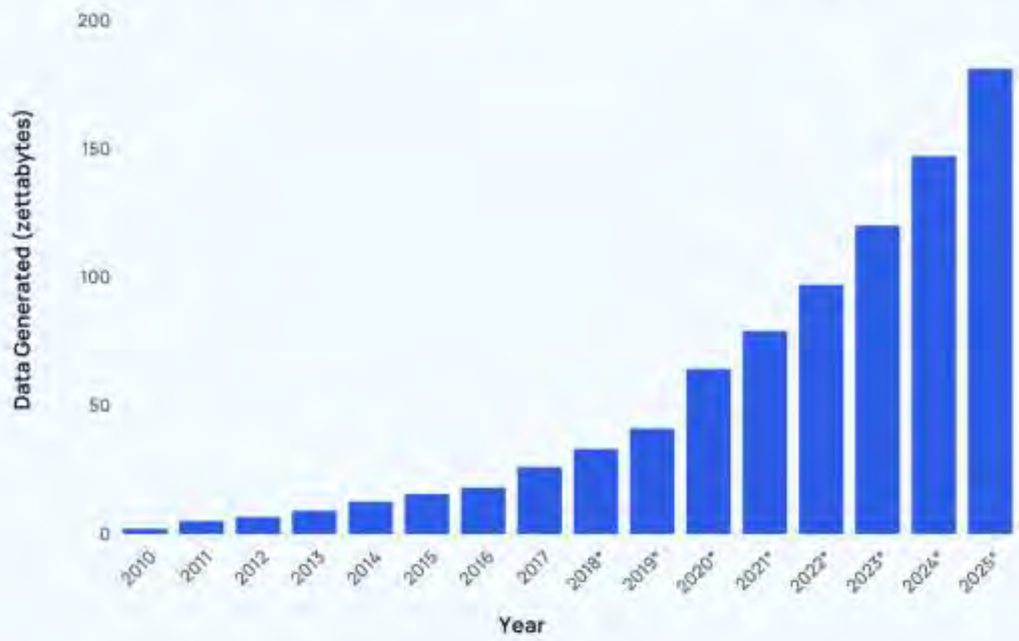
Vectors are the language of AI

- Semantic search and AI are exploding as organizations want to ***embed everything***
- ***Exponential growth*** over the past 7 years
- Accounts for ***most new data*** being stored

type

- Organizations ***aren't able to use*** most of their unstructured data
- Indexing is ***time consuming***
- The largest organizations today require ***trillion-vector scale***

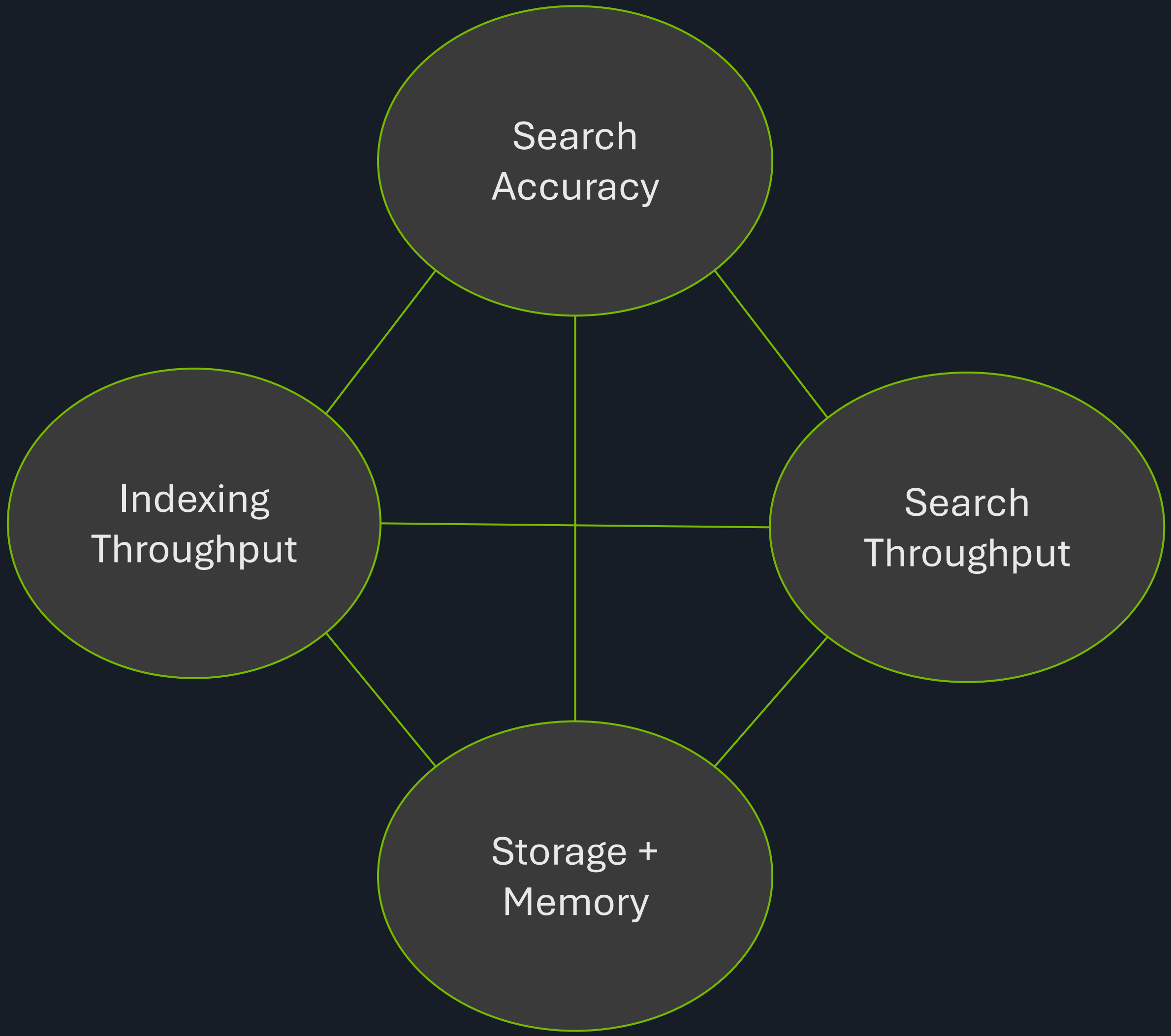
Global Data Generated Annually



Video is responsible for over half (53.72%) of all global data traffic.

It’s all about trade-offs

- Vector search indexes are closer to ***machine learning*** than traditional databases
- ***Highly accurate search*** typically means ***lower throughput*** and ***higher memory/storage***.
- ***Higher throughput*** typically means ***lower accuracy*** and/or ***memory/storage***.
- Exact trade-offs are often workload specific, requiring a level of ***tuning***.



Challenges

Index build

Accurate indexes can be slow to build

Interoperability

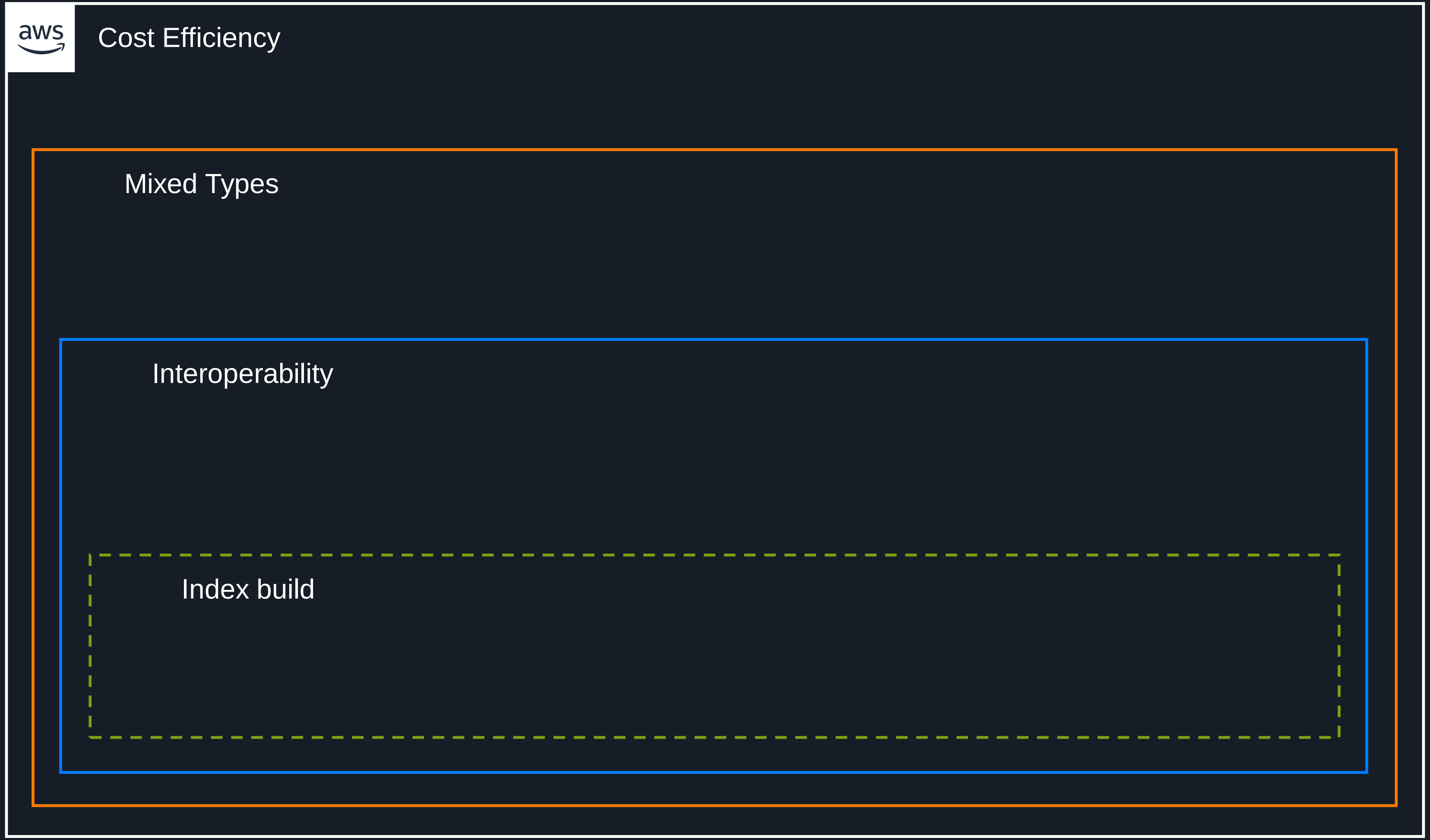
CPU is often good enough for online vector retrieval

Mixed Types

Real-world workflows require both structured and unstructured data

Cost Efficiency

Idle GPUs are not cost effective



Cost Efficiency

Mixed Types

Interoperability

Index build

NVIDIA cuVS

Best Performance
20X faster index build time, 11X lower latency

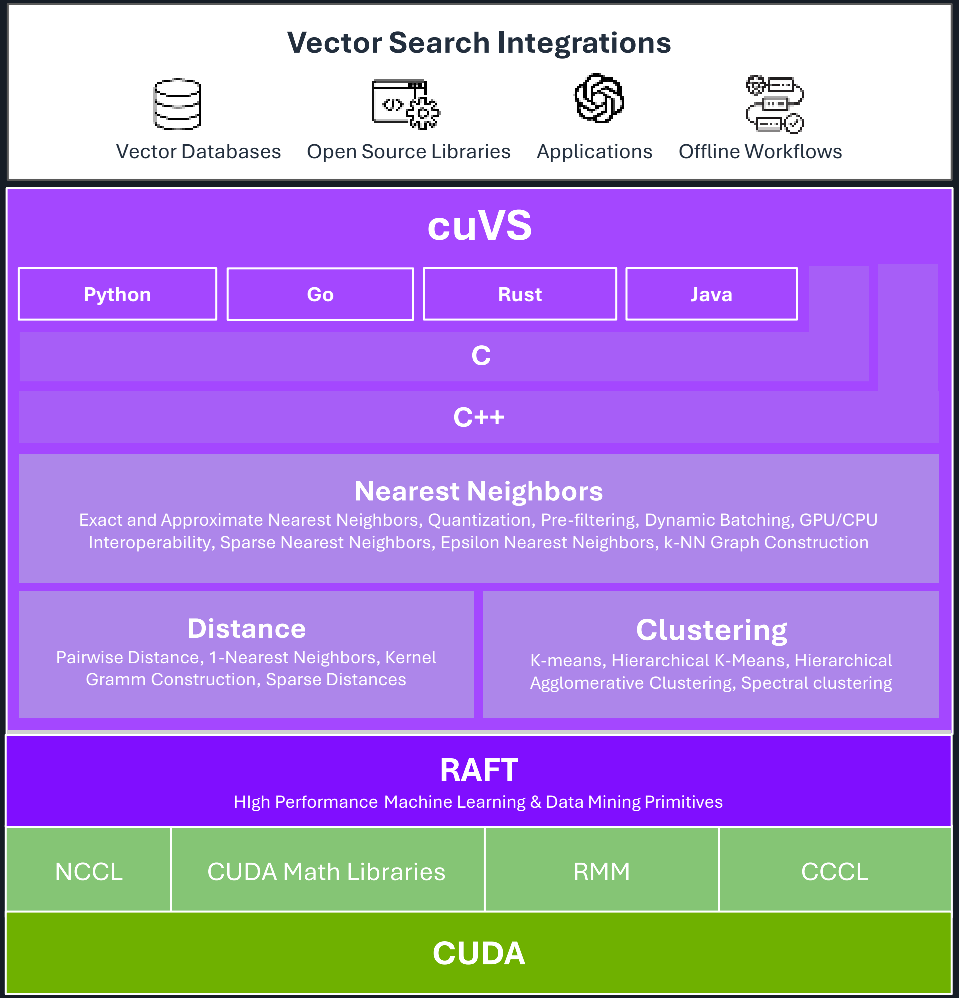
Advanced Algorithms
Performance-tuned approximate nearest neighbor search

Flexible Integration
Supports multiple languages including C, C++, Python, and Rust, for easy integration into vectorized data applications

Interoperable
interoperable between CPU and GPU enabling index building on a GPU and searching on a CPU

Scalable
Enables massive-scale vector search and clustering workloads with GPU acceleration

Fully open source Apache 2.0 License



Vector Search Integrations

Vector Databases

Open Source Libraries

Applications

Offline Workflows

cuVS

Python

Go

Rust

Java

C

C++

Nearest Neighbors
Exact and Approximate Nearest Neighbors, Quantization, Pre-filtering, Dynamic Batching, GPU/CPU Interoperability, Sparse Nearest Neighbors, Epsilon Nearest Neighbors, k-NN Graph Construction

Distance
Pairwise Distance, 1-Nearest Neighbors, Kernel Gramm Construction, Sparse Distances

Clustering
K-means, Hierarchical K-Means, Hierarchical Agglomerative Clustering, Spectral clustering

RAFT
HIgh Performance Machine Learning & Data Mining Primitives

NCCL

CUDA Math Libraries

RMM

CCCL

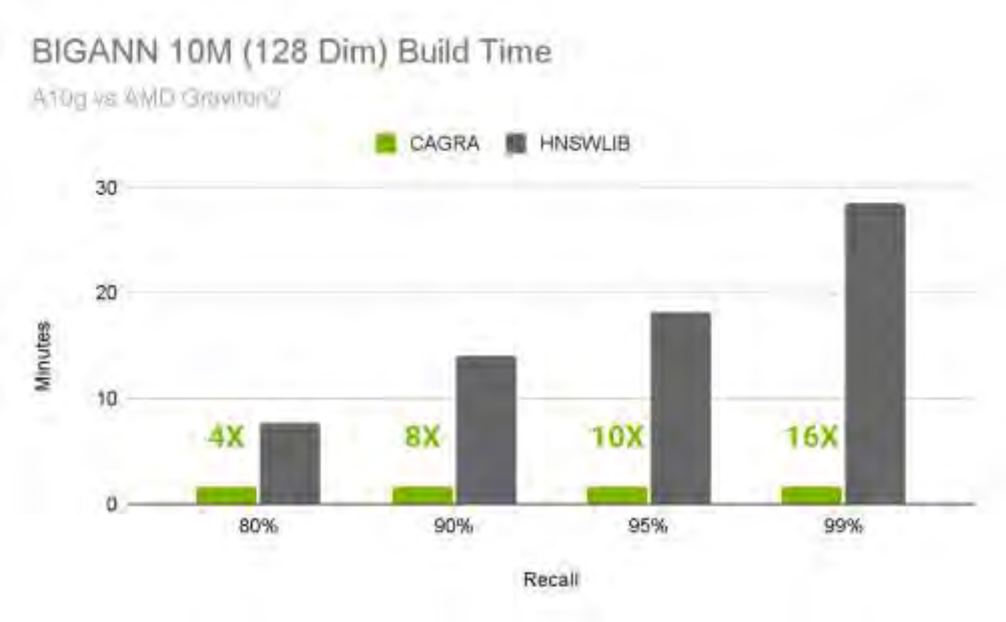
CUDA

https://developer.nvidia.com/cuvs

CAGRA

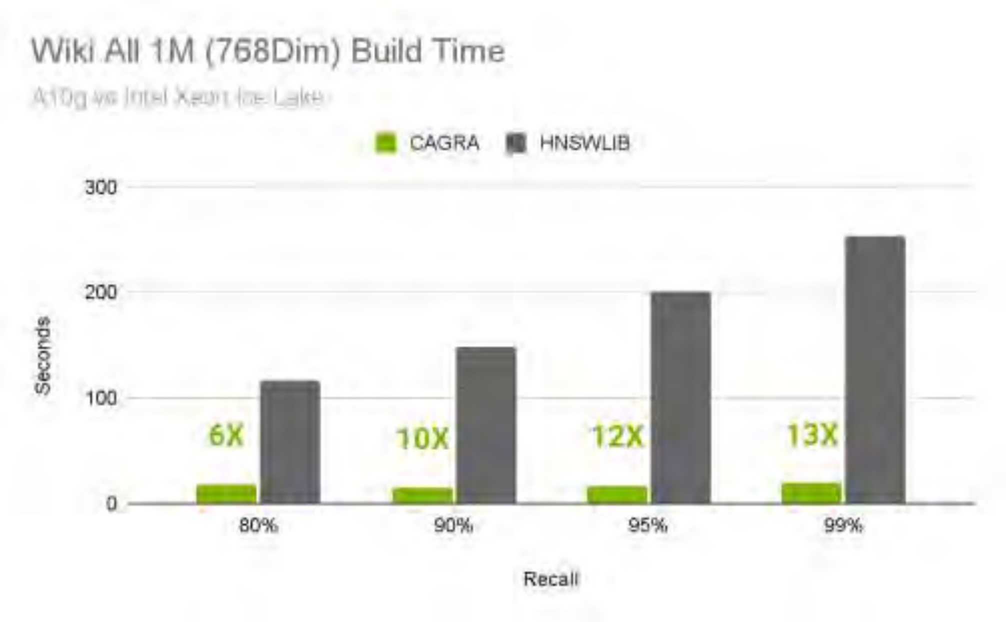
GPU-Accelerated State-of-the-Art Graph-Based ANN

- Open-source, graph-based, *GPU-native algorithm*
- Parallelizes graph construction, *lowering build times* significantly
- Parallelizes *individual search queries* resulting in *high throughput,* especially for *large batches*
- *Lowers latency* for *online queries*



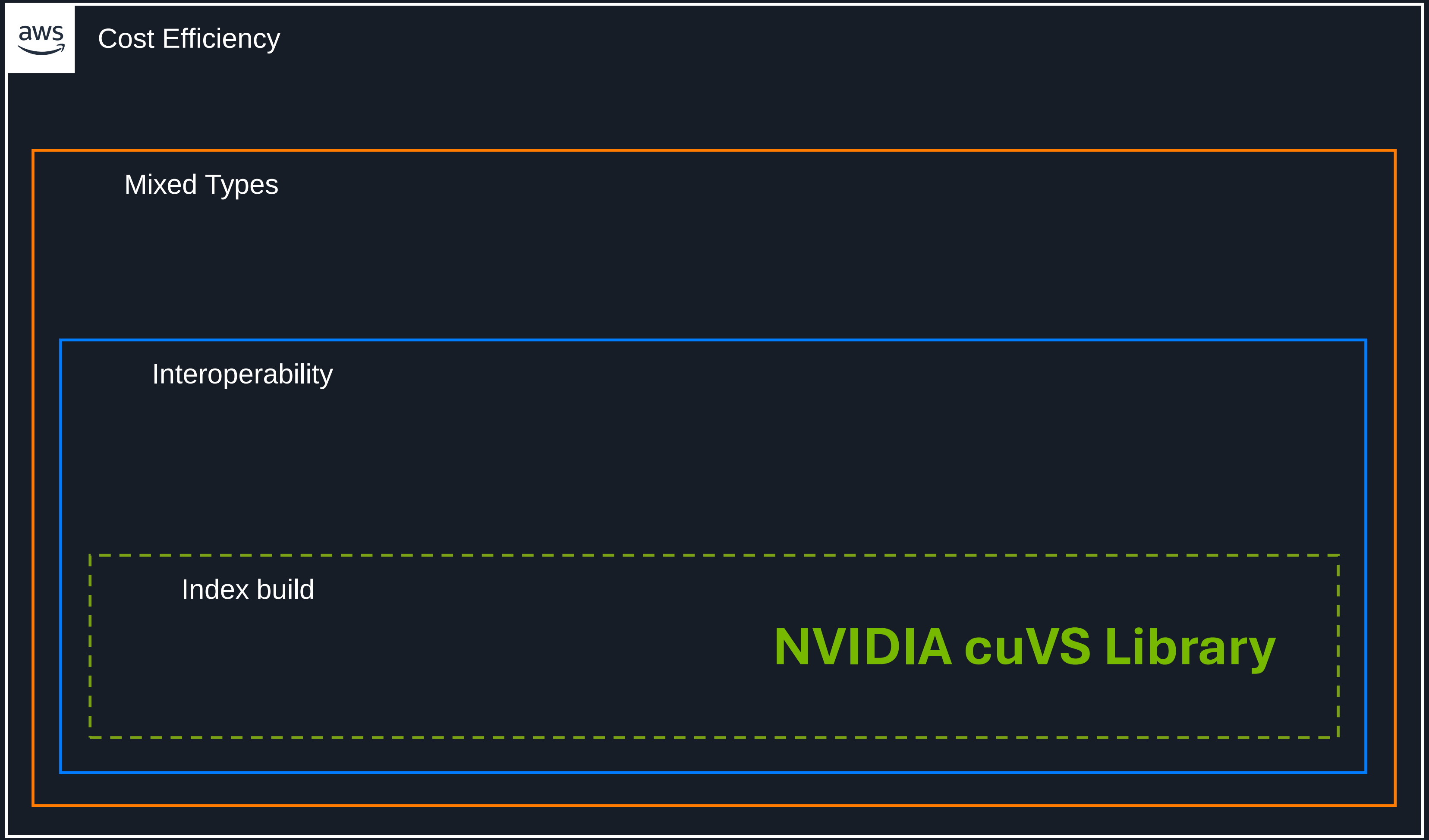
BIGANN 10M (128 Dim) Build Time

A10g vs AMD Graviton2



Wiki All 1M (768Dim) Build Time

A10g vs Intel Xeon Ice Lake



Cost Efficiency

Mixed Types

Interoperability

Index build

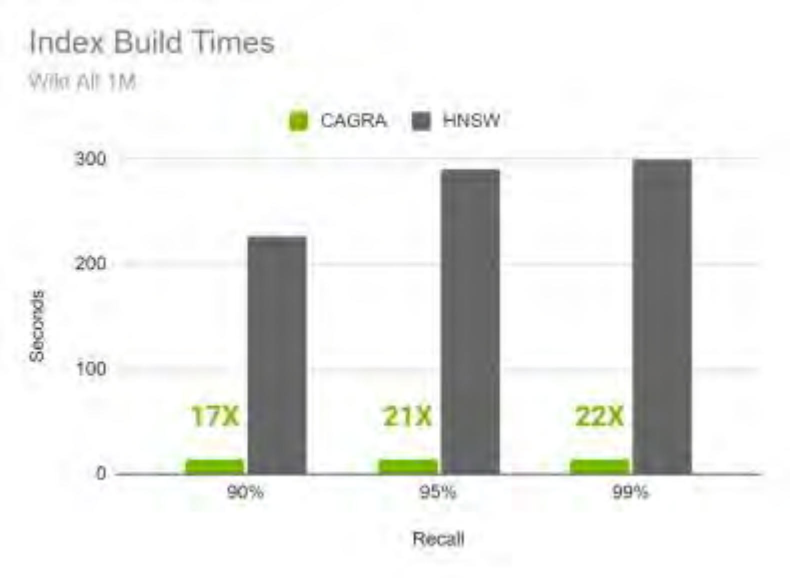
NVIDIA cuVS Library

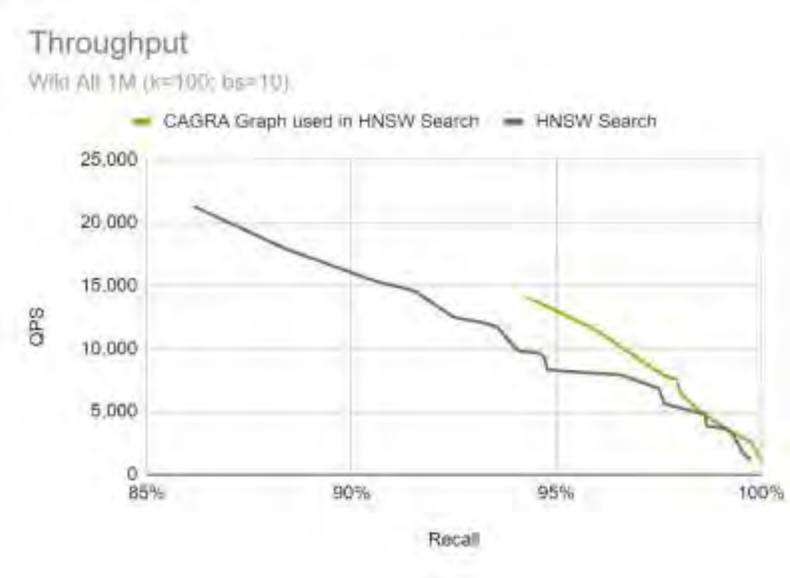
Converting **CAGRA** to **HNSW**

Building index on GPU and searching on CPU

Interoperability between CPU and GPU

- Building HNSW indexes is slow – can take ***hours or days***
- CAGRA indexes can be ***built 20x*** faster than HNSW
- HNSW can search a graph ***built with CAGRA***
- Flat CAGRA graph on CPU can even ***outperform HNSW on CPU*** search at larger dimensions

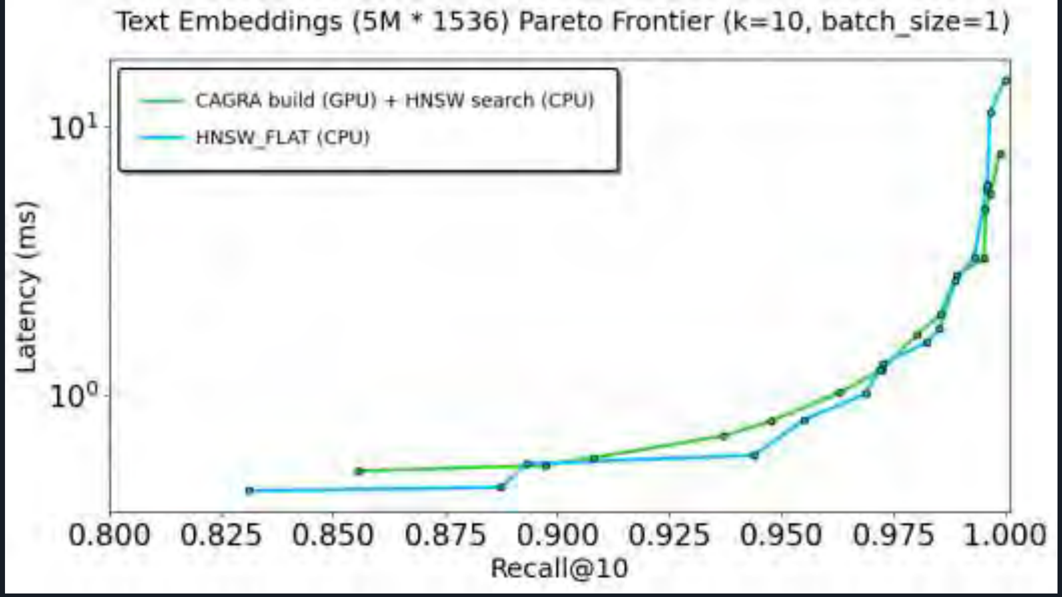


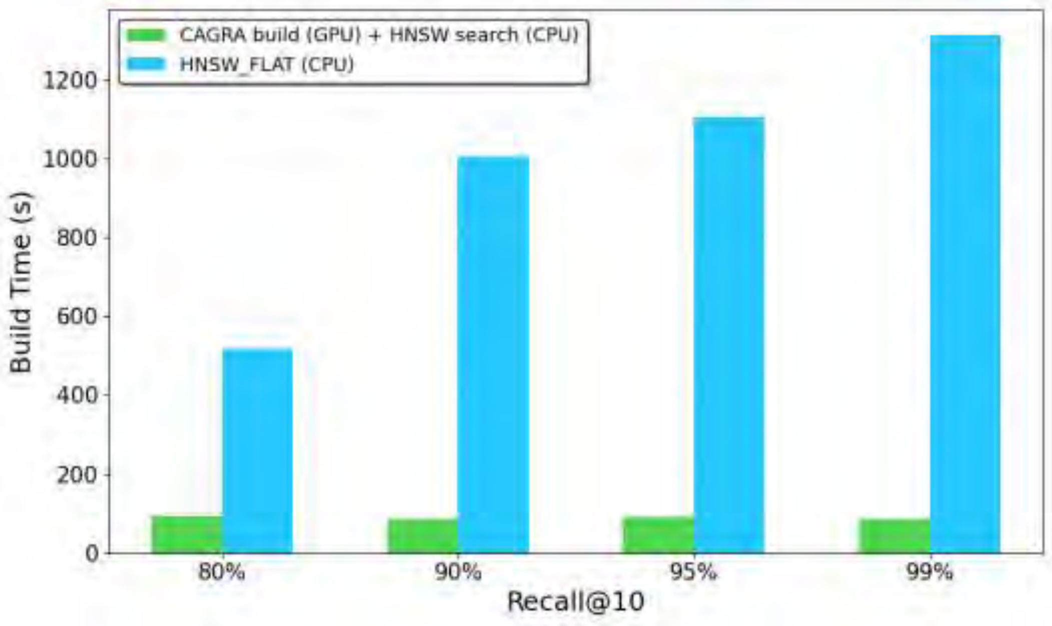


Faiss Library w/ cuVS Backend

Collaboration between AWS, Meta, & NVIDIA

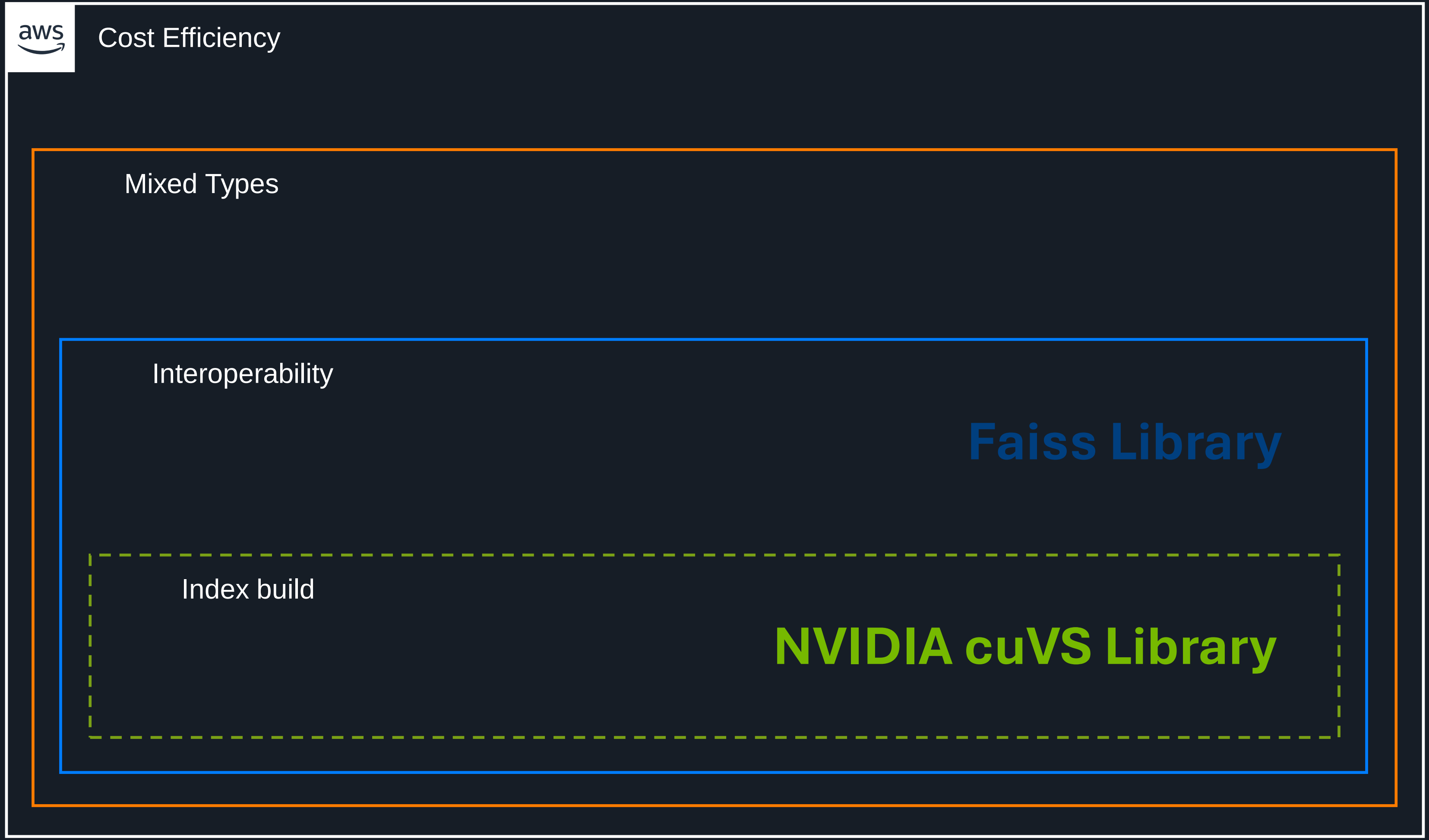
- Use *cuVS in Faiss* with minimal code change.
- Improved build and search performance *on the GPU*
- Build *indexes on GPU*, search on CPU





OpenAI Dataset (5M x 1536)

H100 GPU and an Intel Xeon Platinum 8480CL CPU.



Cost Efficiency

Mixed Types

Interoperability

Faiss Library

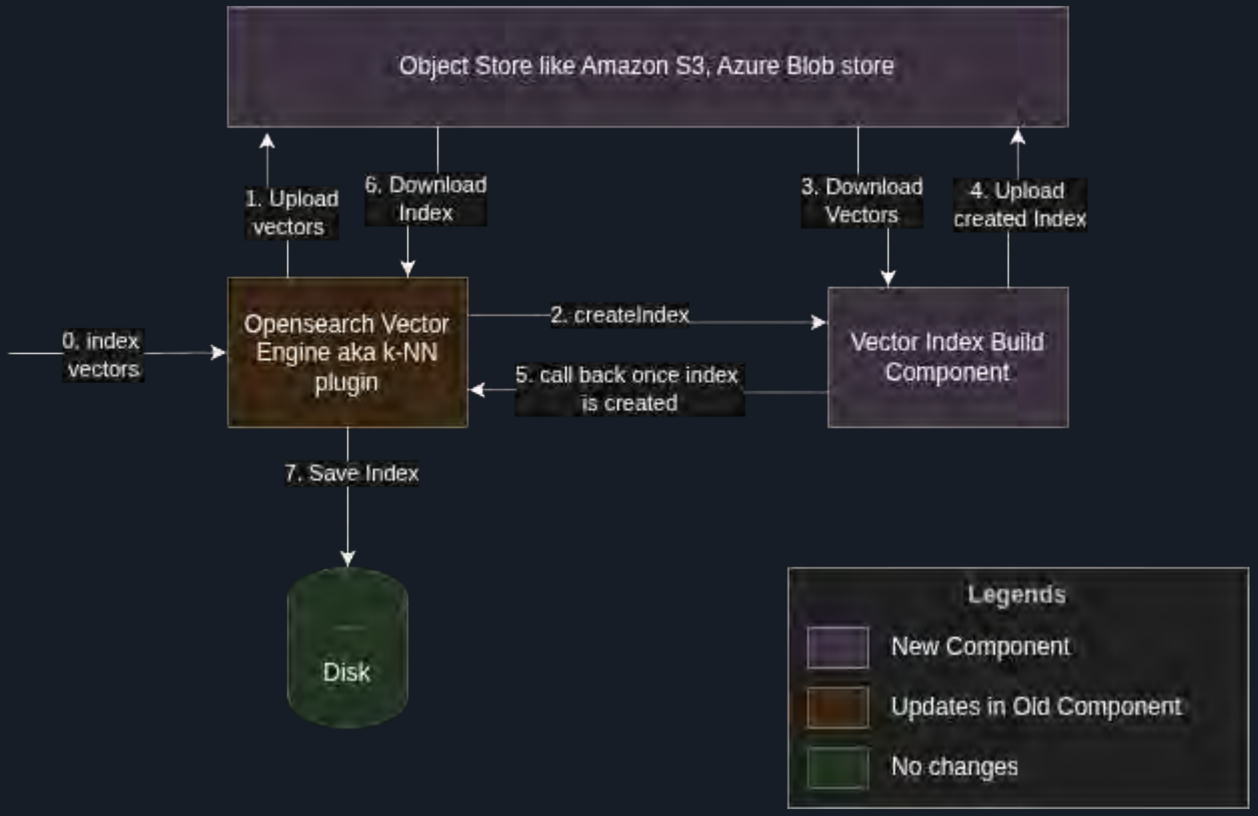
Index build

NVIDIA cuVS Library

GPU-Acceleration in Amazon OpenSearch Service

NVIDIA and AWS bring cuVS to OpenSearch Service

- ***Externalize index building*** to a separate process
- Trivial ***scale-out***
- Use GPU ***only when it makes sense***



Object Store like Amazon S3, Azure Blob store

1. Upload vectors

6. Download Index

3. Download Vectors

4. Upload created Index

0. index vectors

Opensearch Vector Engine aka k-NN plugin

2. createIndex

5. call back once index is created

Vector Index Build Component

7. Save Index

Disk

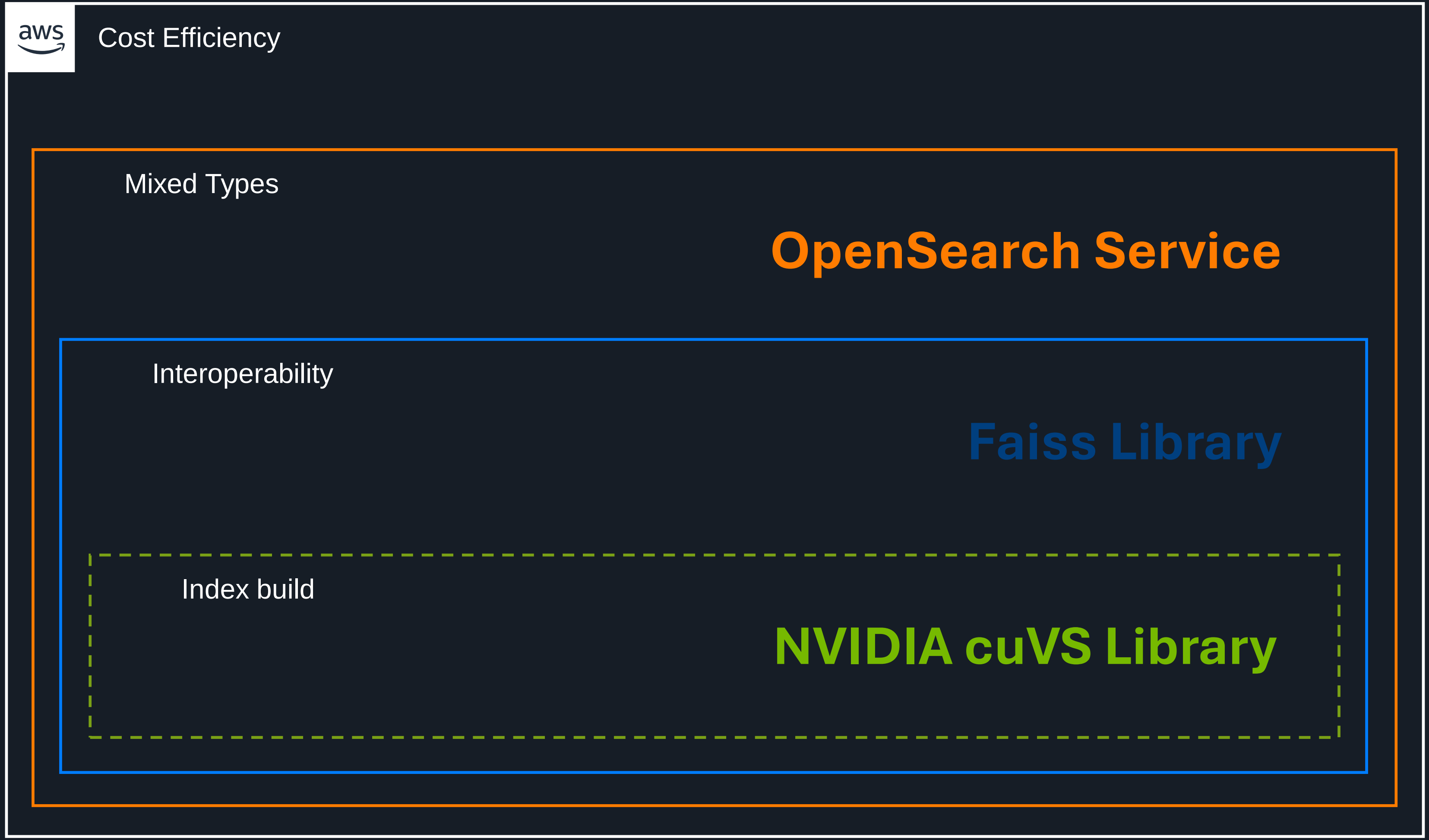
Legends

🟪 New Component

🟫 Updates in Old Component

🟩 No changes

https://opensearch.org/blog/gpu-accelerated-vector-search-opensearch-new-frontier/



Cost Efficiency

Mixed Types

OpenSearch Service

Interoperability

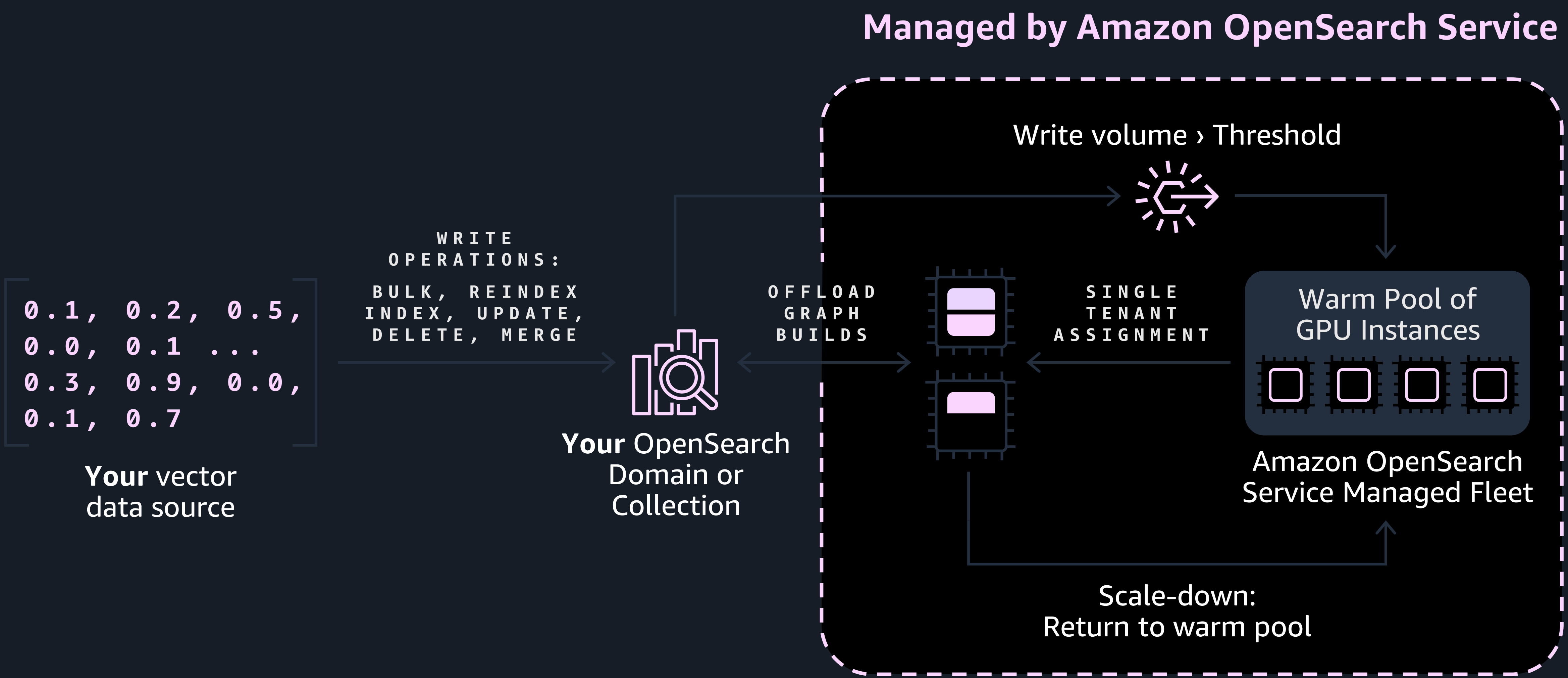
Faiss Library

Index build

NVIDIA cuVS Library

Serverless GPU-Accelerated Vector Indexing

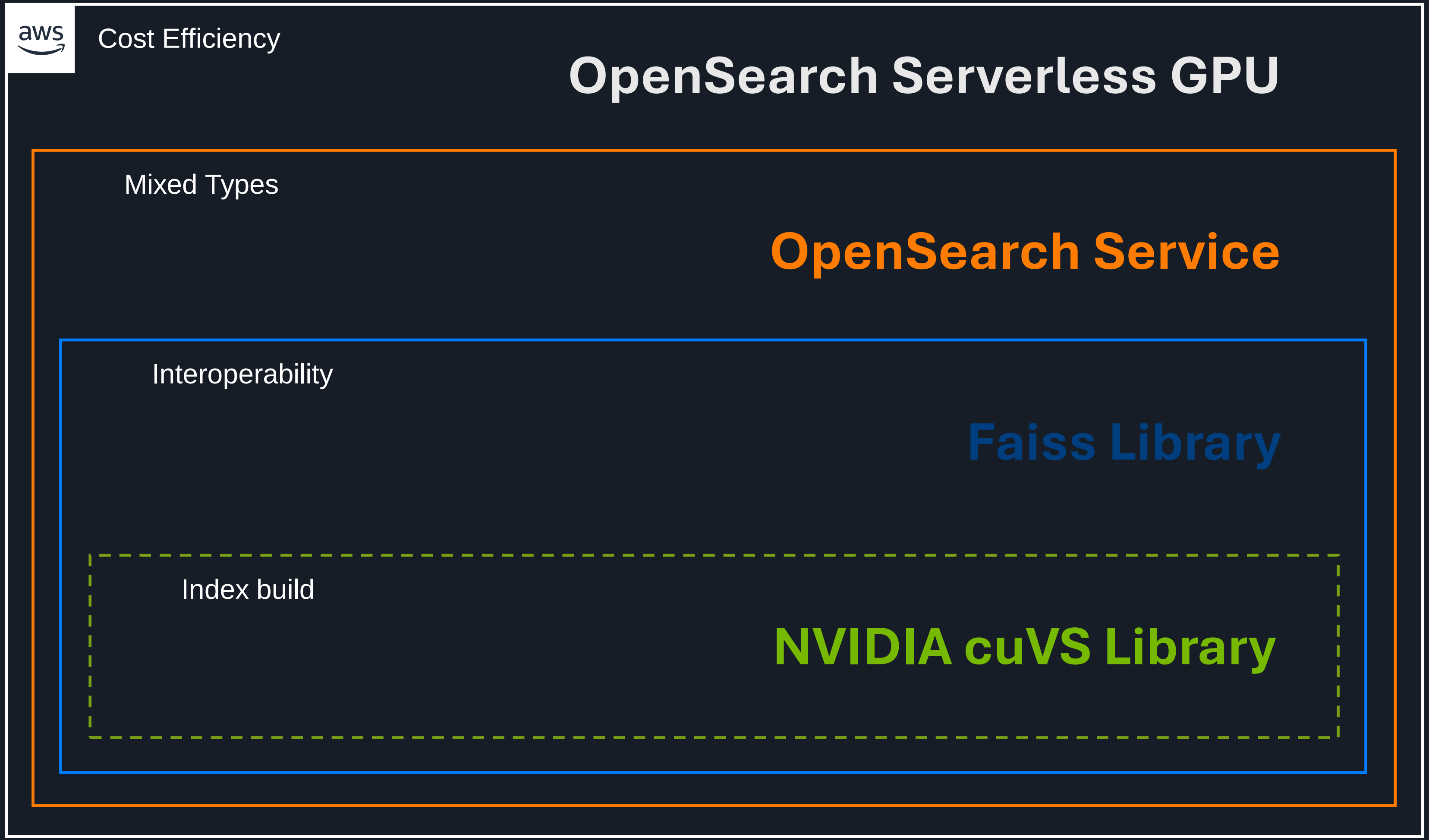
Bringing cost efficiency and performance to OpenSearch Service



Improved Performance and Cost Efficiency w/ Serverless GPU

Faster and lower cost vector ingest on the GPU

Dataset	CPU-Only		With GPU (OCU @ $0.24/hr.)		Improvement	
	Index & Merge	Domain Cost during Index Build	Index & Merge	Total Costs During Index Build	Cost	Speed
Cohere Embed V2: 1Mx768	1.4 hr	$1.00	9.9 min	$0.13	**12.0x**	**8.5x**
Cohere Embed V2: 10Mx768	8.5 hr	$37.82	36.8 min	$3.10	**12.2x**	**13.9x**
Cohere Embed V3: 113Mx1024	28.7 hr	$712.47	4.5 hr	$121.70	**5.9x**	**6.4x**
SIFT 1Bx128	31.9 hr	$1118.09	2.8 hr	$109.86	**10.2x**	**11.4x**



Cost Efficiency

OpenSearch Serverless GPU

Mixed Types

OpenSearch Service

Interoperability

Faiss Library

Index build

NVIDIA cuVS Library

Summary

Index build

cuVS can build more accurate indexes faster on the GPU

Interoperability

Faiss library can build indexes on the GPU while enabling search on the CPU

Mixed Types

OpenSearch Service enables structured, lexical, and unstructured search

Cost Efficiency

OpenSearch Serverless GPU can build indexes up to 10x faster at 75% lower cost.

Search Innovations

Rich configurations, expert-driven process

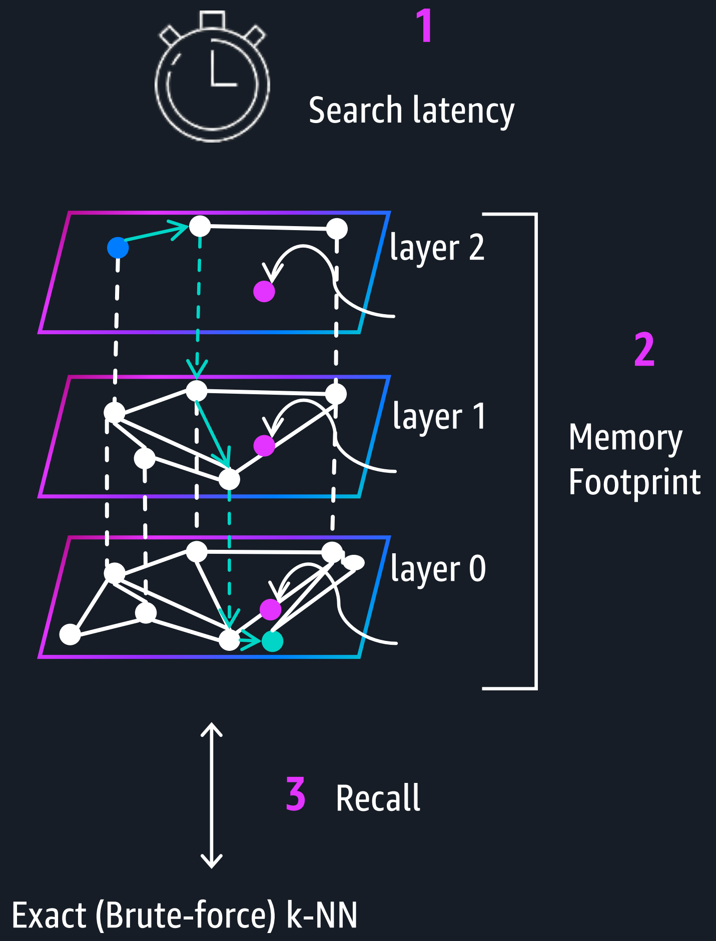
Select Index Parameters

Algorithms:
HNSW, ef_construction, m…

Quantization:
Scalar, Binary,
Product

Engine Settings:
Disk-optimized,
In-memory,
Infrequent Queries

Build and Evaluate Index



1 Search latency

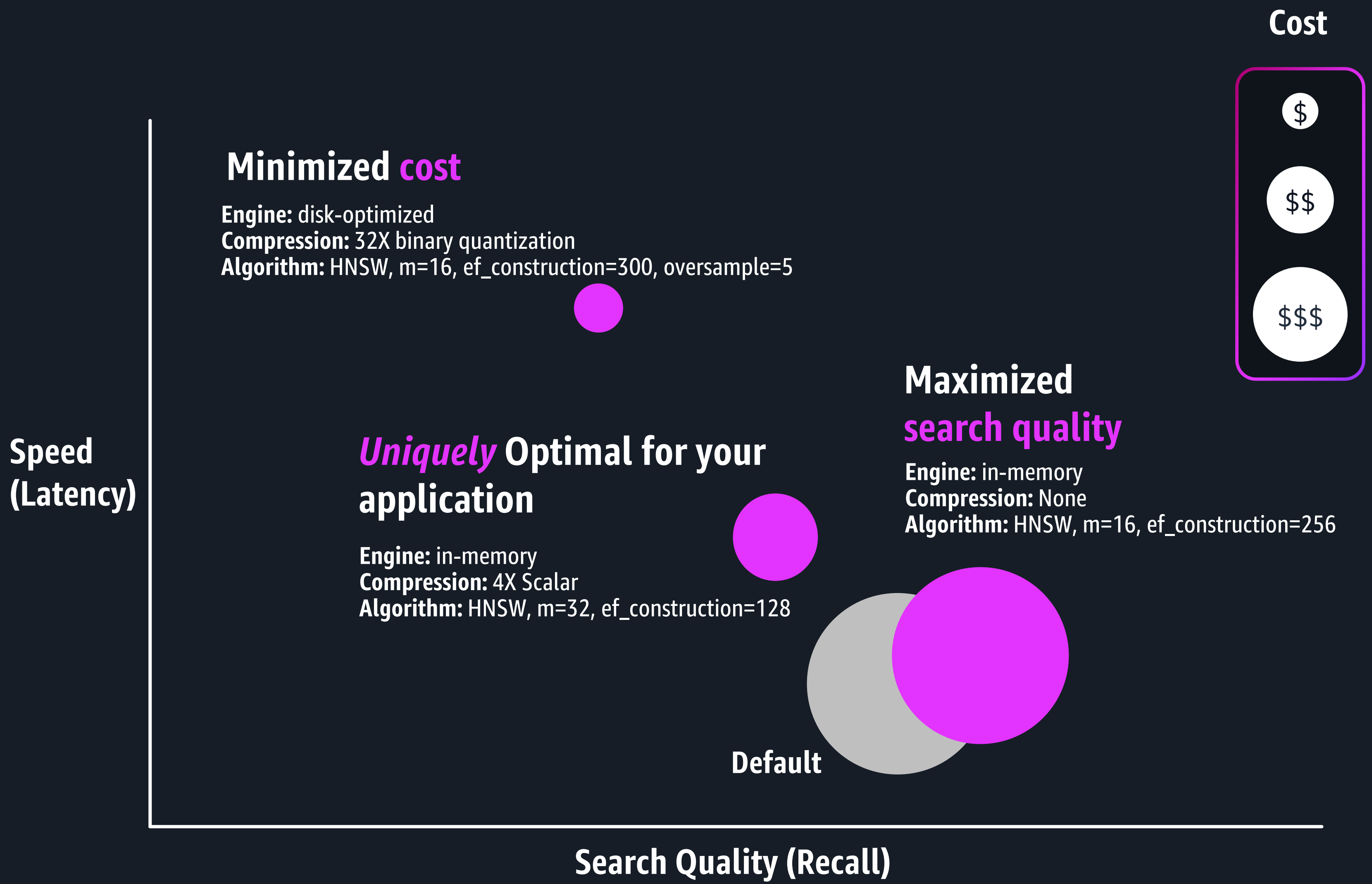
2 Memory Footprint

3 Recall

Exact (Brute-force) k-NN

Adjust Parameters and Repeat

Finding the Optimal trade-offs



Minimized **cost**

Engine: disk-optimized
Compression: 32X binary quantization
Algorithm: HNSW, m=16, ef_construction=300, oversample=5

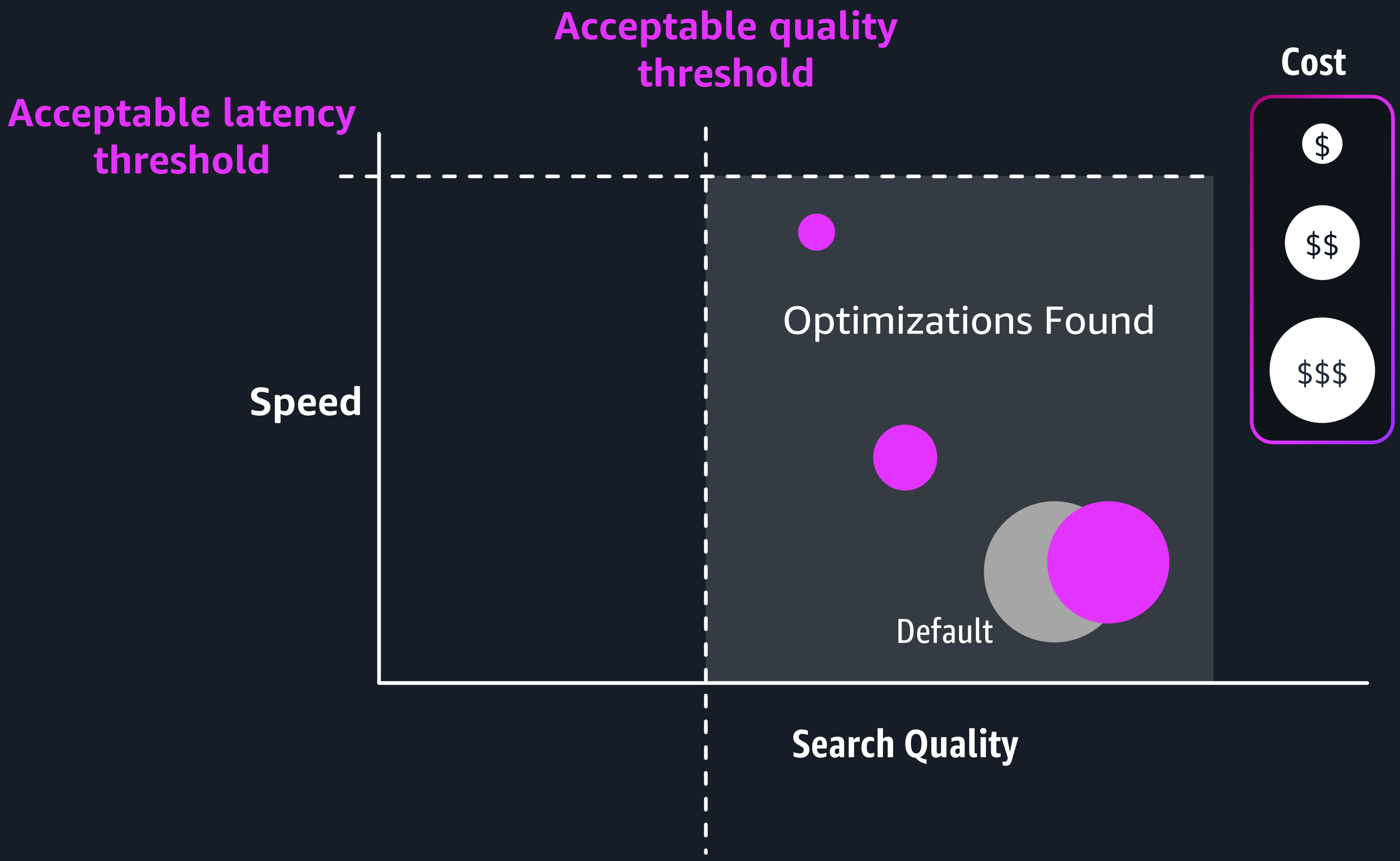
Uniquely **Optimal for your application**

Engine: in-memory
Compression: 4X Scalar
Algorithm: HNSW, m=32, ef_construction=128

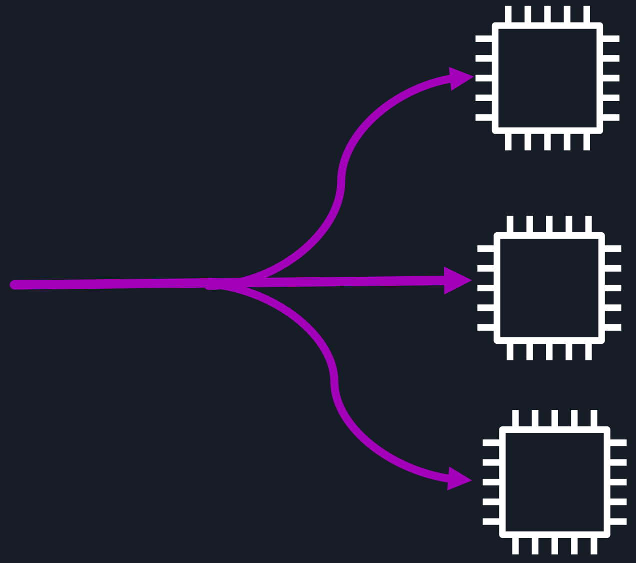
Maximized search quality

Engine: in-memory
Compression: None
Algorithm: HNSW, m=16, ef_construction=256

Let’s simplify! Auto-optimize Vector Indexing



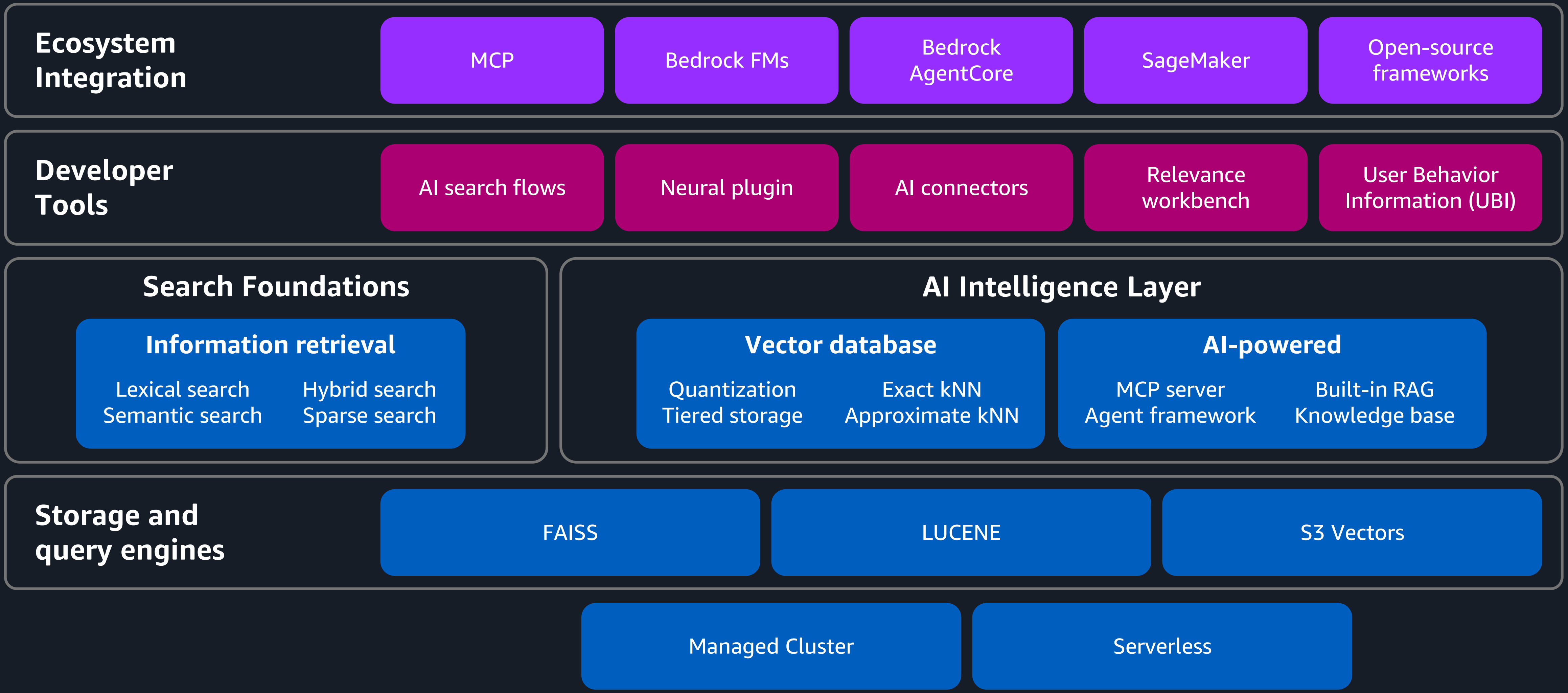
Parallelize Index Builds and Evaluations



Serverless Auto-optimize jobs with a predictable flat rate

OpenSearch Service stack

BEYOND TRADITIONAL SEARCH



Ecosystem Integration

MCP

Bedrock FMs

Bedrock AgentCore

SageMaker

Open-source frameworks

Developer Tools

AI search flows

Neural plugin

AI connectors

Relevance workbench

User Behavior Information (UBI)

Search Foundations

Information retrieval

Lexical search
Semantic search
Hybrid search
Sparse search

AI Intelligence Layer

Vector database

Quantization
Tiered storage
Exact kNN
Approximate kNN

AI-powered

MCP server
Agent framework
Built-in RAG
Knowledge base

Storage and query engines

FAISS

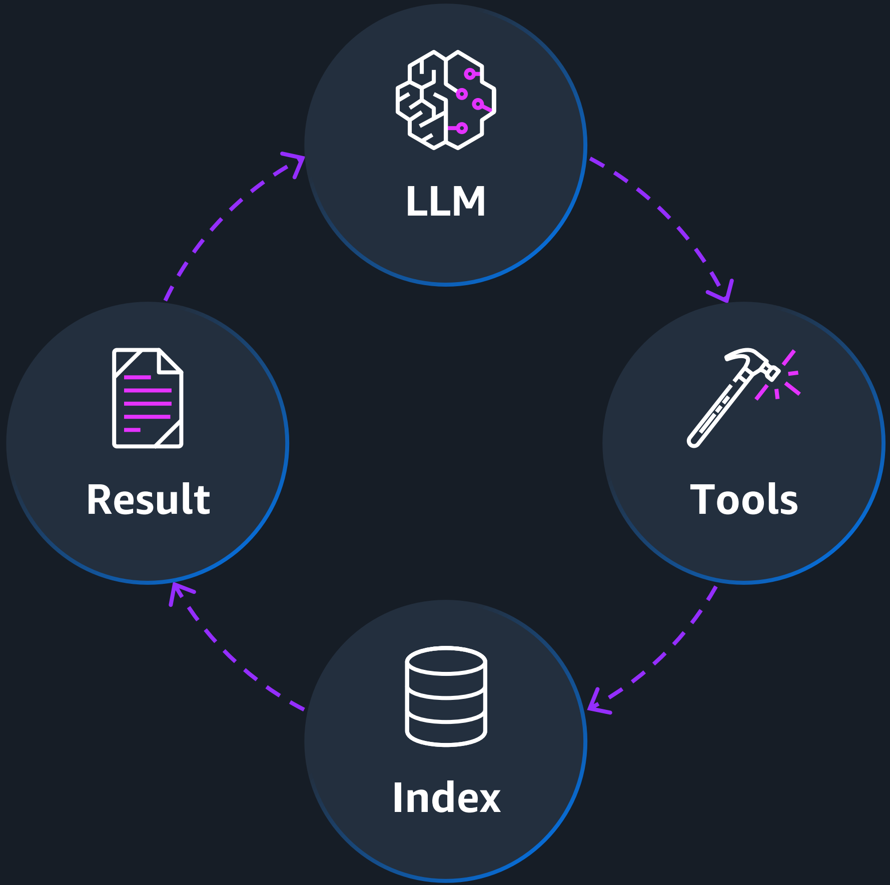
LUCENE

S3 Vectors

Managed Cluster

Serverless

Search for Agent driven workloads



Agents need context

Even the best LLMs are effective only when then have relevant, timely, and complete context

Agent workloads are dynamic

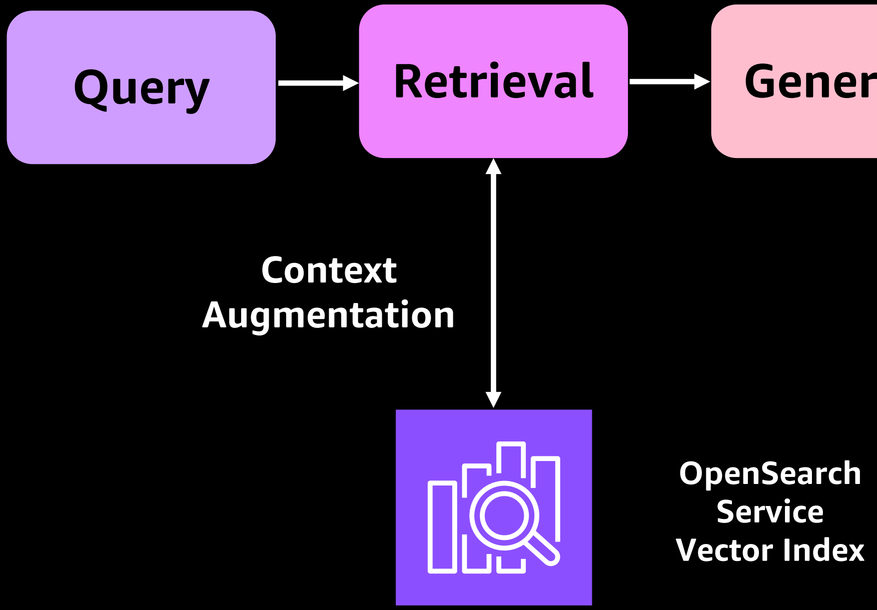
Highly concurrent and diverse queries, continuous indexing, and iterative

Retrieval is the foundational layer

Gives agents access to context aware vector data at scale

Agents need relevant and accurate context for the task at hand

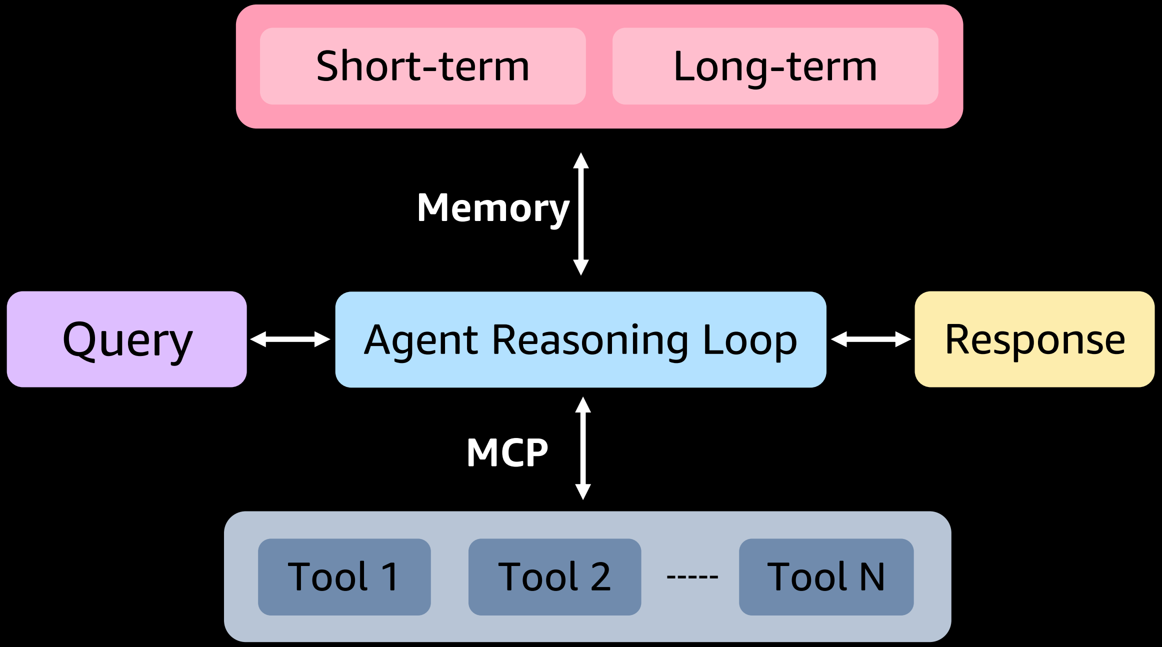
Context enhancement with RAG and Agentic Search



RAG retrieves information, can’t reason

RAG is static but can’t adapt

RAG is single-turn, can’t iterate



Agentic search is multi-turn and iterative

Reasoning driven retrieval

Access multiple tools and data sources

OpenSearch Service building foundations for Agentic Search

MCP Server

Agentic Memory

Specialized Agents

OpenSearch Service MCP integrations

Seamlessly connect with AI systems with external data and tools

- Built-in server, Standalone server, and MCP connector in OpenSearch Service and open source
- Growing list of tools (ListIndexTool, SearchIndexTool, QueryPlanningTool…)
- Support multiple OpenSearch Service clusters
- Secure Authentication with configurable MCP credential

crewai

STRANDS AGENTS

LangChain

LlamaIndex

OpenSearch Agentic Memory

OpenSearch v3.3

Intelligent memory management for context
auto memory update, knowledge extraction, data drift

Ease of use with a REST API for memory operations
Add/delete, update, Search, Get…

Data life cycle management
temporal data, tiered storage

Security and access control
user access, index permissions, configure namespace access

OpenSearch v3.3

OpenSearch Specialized Agents

Flow agent

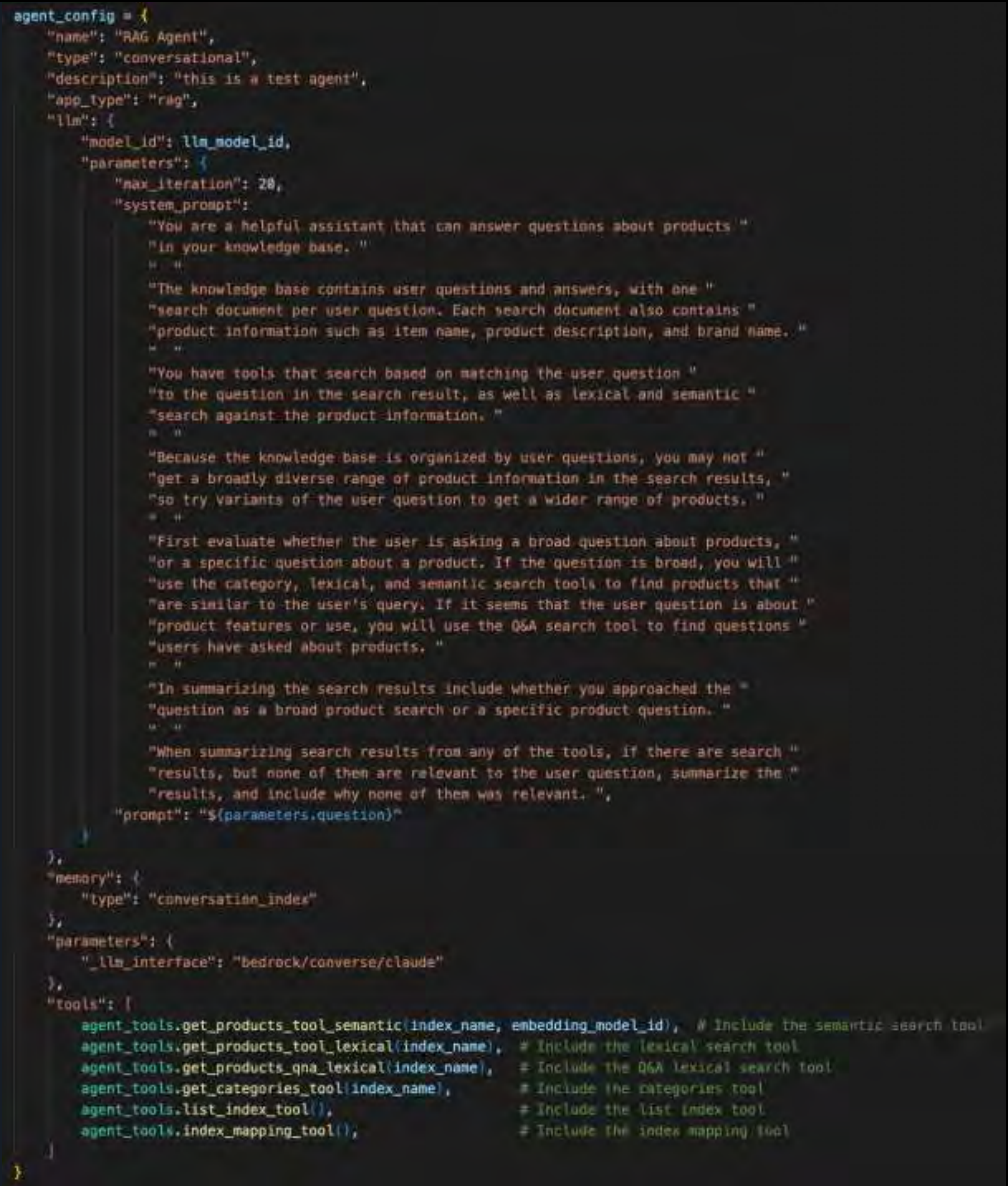
Runs tools in a specified order.
Use for: RAG

Conversational agent

LLM sequences tool execution. Use for reasoning and conversation

Plan-execute-reflect agent

LLM reasons based on tools. Use for long, exploratory processes



```
agent_config = {
    "name": "RAG Agent",
    "type": "conversational",
    "description": "this is a test agent",
    "app_type": "rag",
    "llm": {
        "model_id": llm_model_id,
        "parameters": {
            "max_iteration": 20,
            "system_prompt":
                "You are a helpful assistant that can answer questions about products "
                "in your knowledge base. "
                "  "
                "The knowledge base contains user questions and answers, with one "
                "search document per user question. Each search document also contains "
                "product information such as item name, product description, and brand name. "
                "  "
                "You have tools that search based on matching the user question "
                "to the question in the search result, as well as lexical and semantic "
                "search against the product information. "
                "  "
                "Because the knowledge base is organized by user questions, you may not "
                "get a broadly diverse range of product information in the search results, "
                "so try variants of the user question to get a wider range of products. "
                "  "
                "First evaluate whether the user is asking a broad question about products, "
                "or a specific question about a product. If the question is broad, you will "
                "use the category, lexical, and semantic search tools to find products that "
                "are similar to the user's query. If it seems that the user question is about "
                "product features or use, you will use the Q&A search tool to find questions "
                "users have asked about products. "
                "  "
                "In summarizing the search results include whether you approached the "
                "question as a broad product search or a specific product question. "
                "  "
                "When summarizing search results from any of the tools, if there are search "
                "results, but none of them are relevant to the user question, summarize the "
                "results, and include why none of them was relevant. ",
            "prompt": "${parameters.question}"
        }
    },
    "memory": {
        "type": "conversation_index"
    },
    "parameters": {
        "_llm_interface": "bedrock/converse/claude"
    },
    "tools": [
        agent_tools.get_products_tool_semantic(index_name, embedding_model_id),  # Include the semantic search tool
        agent_tools.get_products_tool_lexical(index_name),  # Include the lexical search tool
        agent_tools.get_products_qna_lexical(index_name),   # Include the Q&A lexical search tool
        agent_tools.get_categories_tool(index_name),        # Include the categories tool
        agent_tools.list_index_tool(),                      # Include the list index tool
        agent_tools.index_mapping_tool(),                   # Include the index mapping tool
    ]
}
```

OpenSearch Service Infrastructure Enhancements

Amazon OpenSearch Service Infrastructure

Scalability

1000 nodes and 25 PB in a single Cluster

Availability

99.99% availability for production workloads

Durability

11 9s of Durability with OpenSearch Optimized Instances

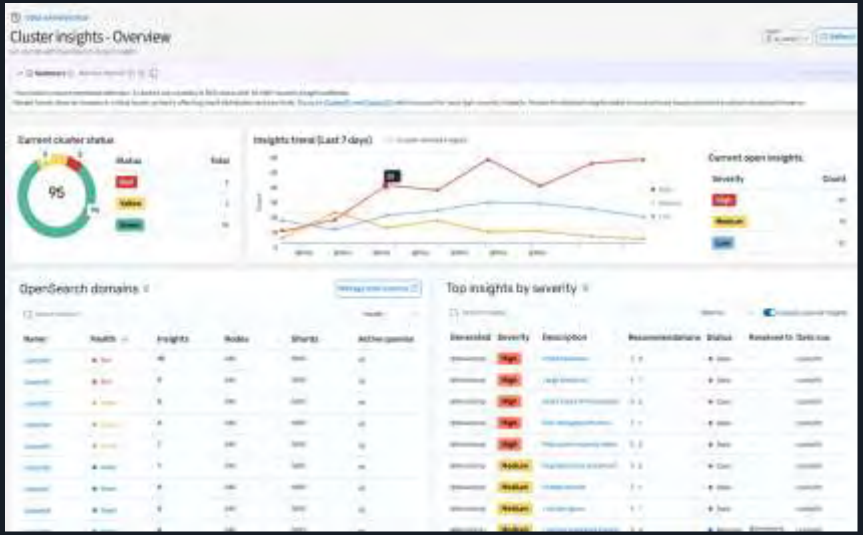
Performance

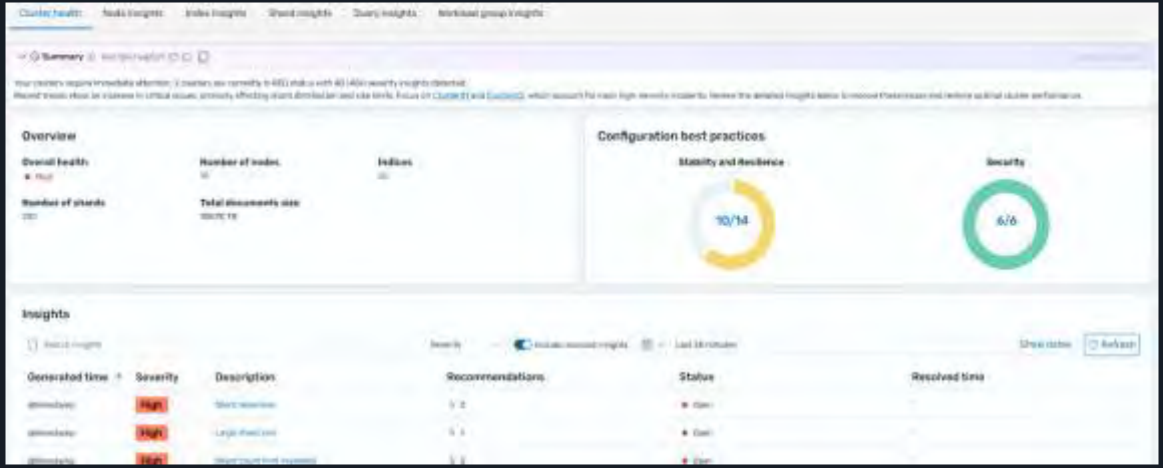
OpenSearch 3.3 11x more performant than OpenSearch 1.0

Improved Ease of Operations

CLUSTER INSIGHTS

All-in-one dashboard transforms complex cluster management into streamlined operations with automated performance analysis and troubleshooting.





** Available on OpenSearch Service managed clusters 2.17+*

Continuous Innovation

OpenSearch Optimized Instances OR2 and OM2

- OR2: 26% indexing throughput improvement on OR1 and 70% improvement over R7g
- OM2: 15% indexing throughput improvement on OR1 and 66% improvement over M7g

Derived source (2.19+) reduces storage by 40% and with ~20% faster indexing and merges

Support for **custom script plug-ins**

Amazon OpenSearch Serverless

Easy to administer
No sizing, scaling, and tuning of clusters, and no shard and index lifecycle management

Fast
Automatically scale resources to maintain consistently fast data ingestion rates and query response times

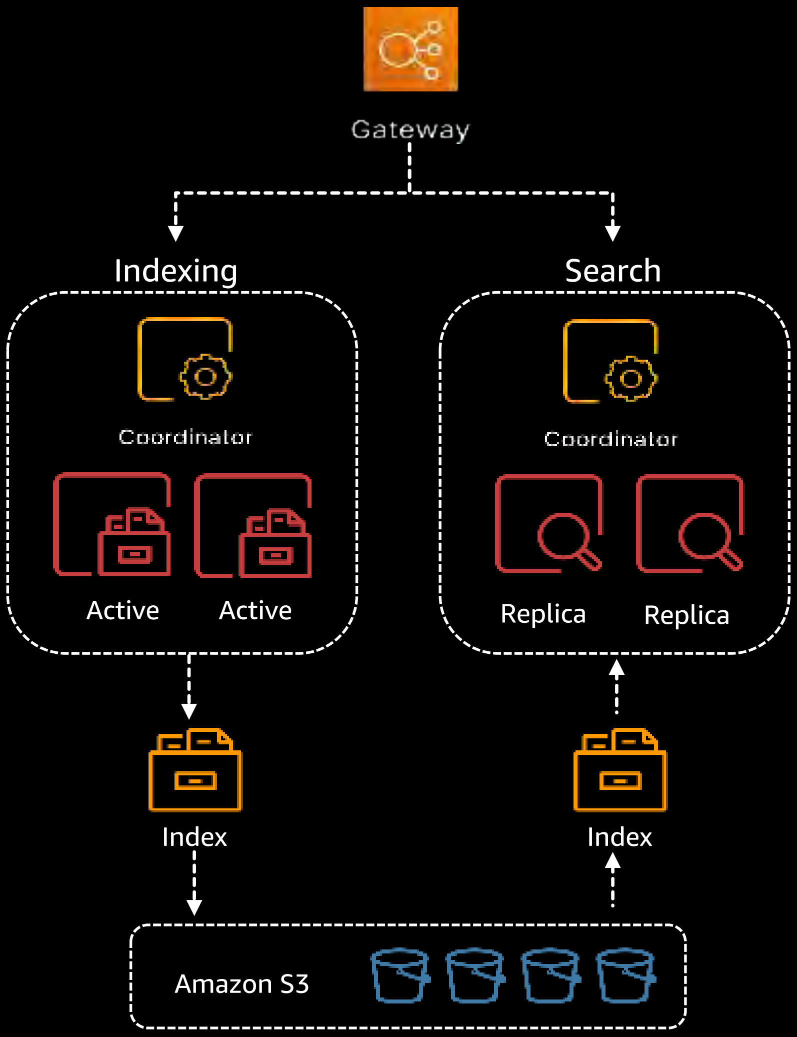
Ecosystem
Get started in seconds using the same OpenSearch clients, pipelines, and APIs

Cost-effective
Pay only for the resources consumed

OpenSearch Serverless Innovation

New capabilities

- 100TB time-series collections
- Region expansion from 15 to 22 regions
- Data plane audit logging
- Snapshot restore



OpenSearch Serverless Modular Architecture

AWS TRAINING AND CERTIFICATION

Learn more about innovations announced at re:Invent 2025

Gain knowledge of AWS services and features the moment they’re announced.

Access our expertly curated learning plan. Ready when you are.

Scan to dive deeper

Thank You

Please complete the session survey in the mobile app