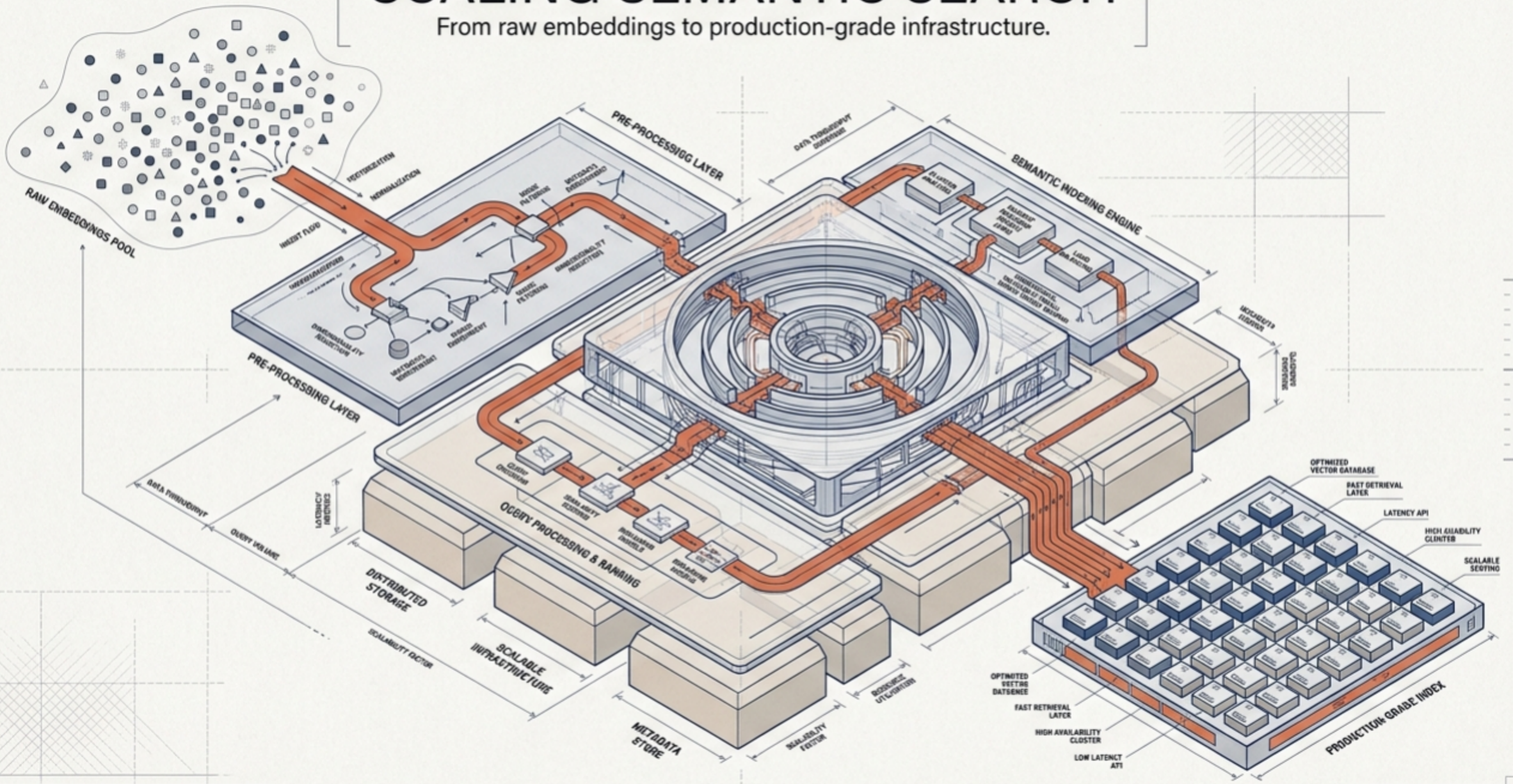
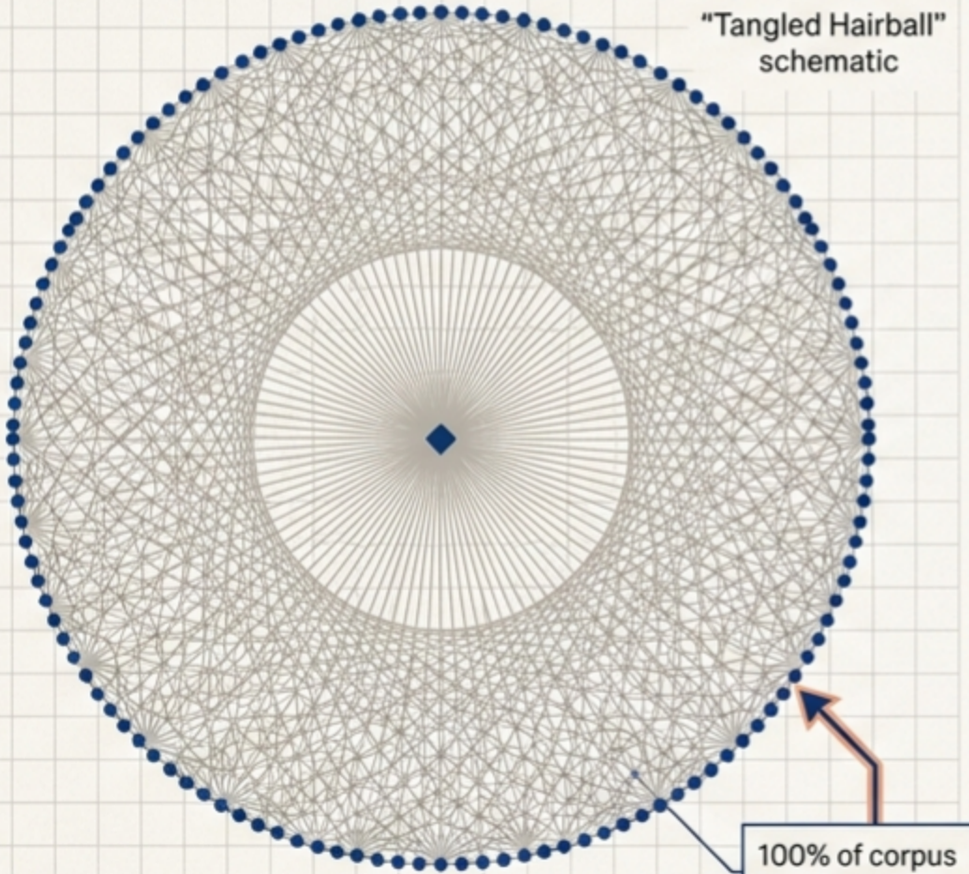
# SCALING SEMANTIC SEARCH

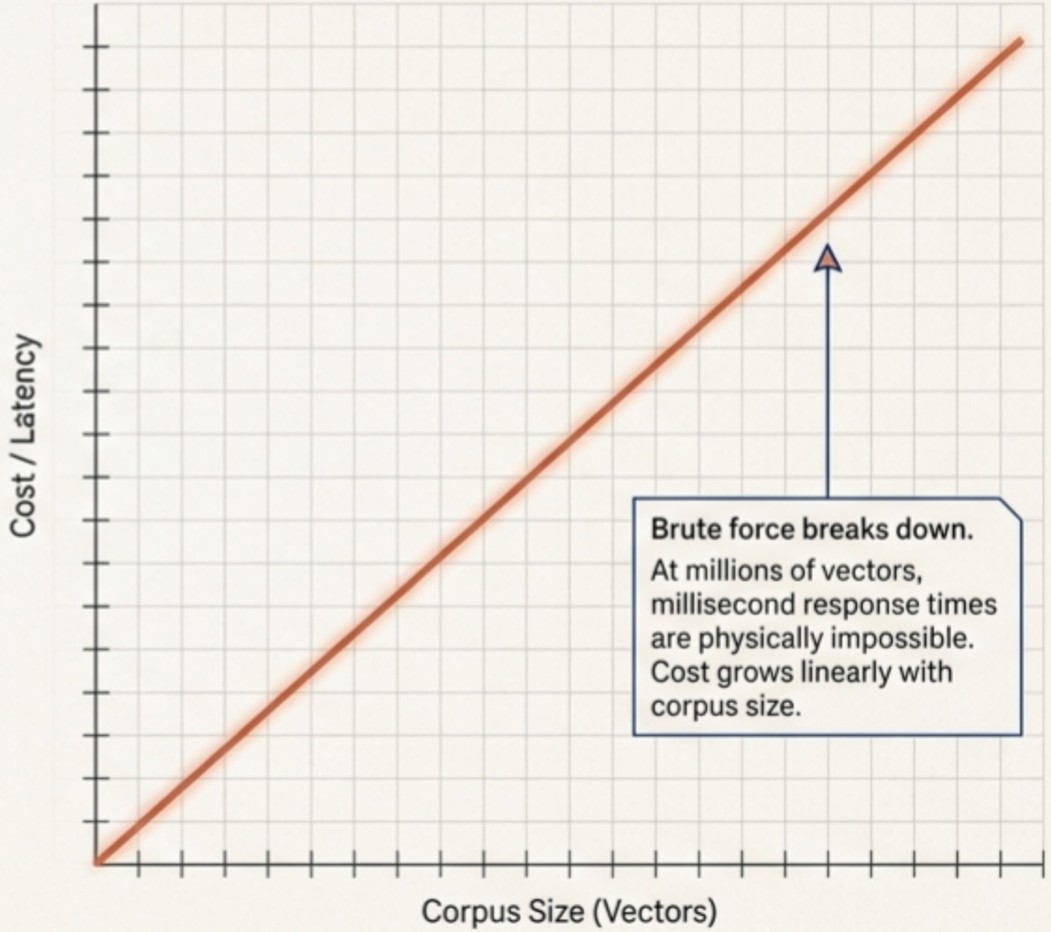
From raw embeddings to production-grade infrastructure.



# The O(N) Reality Check

Exact nearest-neighbor search becomes a physical bottleneck at scale



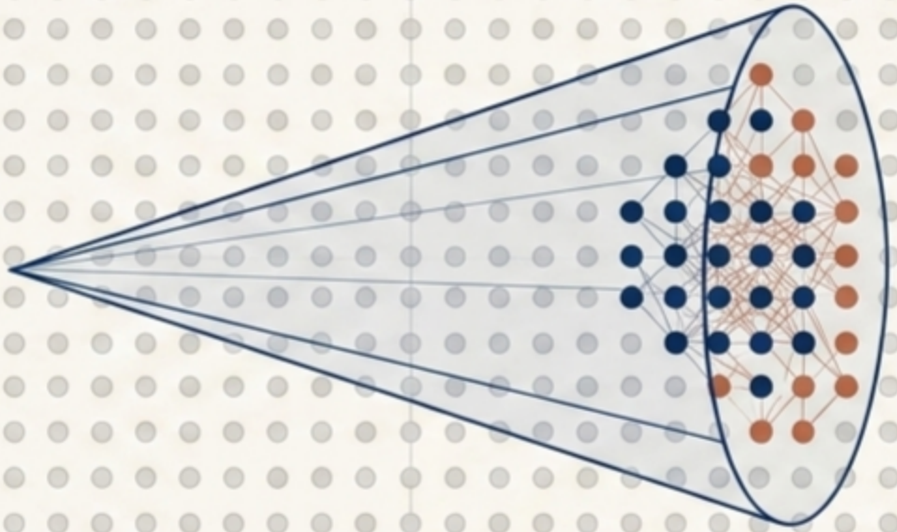



# The Strategic Compromise: ANN

Approximate Nearest Neighbor (ANN) indexing trades exactness for exponential speed.

### Pre-organization

Vectors are mapped and structured in advance.



### Targeted Search

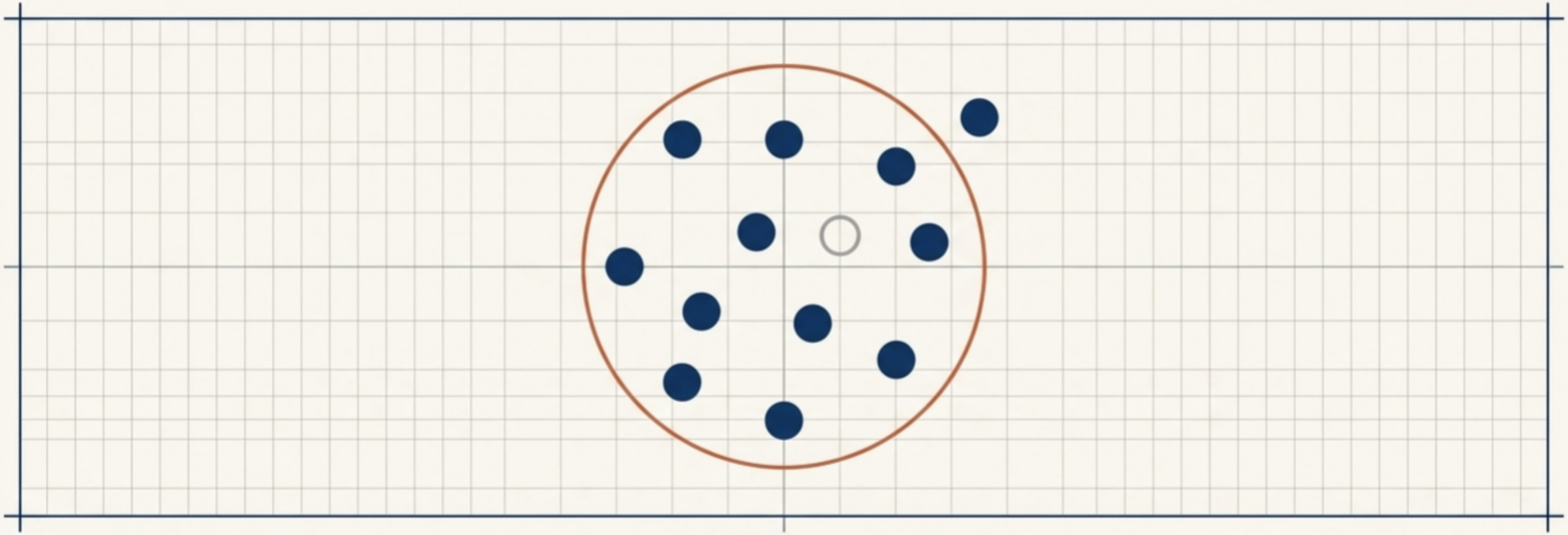
The query only evaluates a highly promising subset.

### The Trade-off

Surrender mathematical exactness for scalable speed-ups.

# Quantifying Accuracy: Recall@k

Recall@k quantifies the exactness lost in the algorithmic compromise.



$$\text{Recall@k} = \frac{\text{True Top-k found by ANN}}{\text{Total True Top-k}}$$

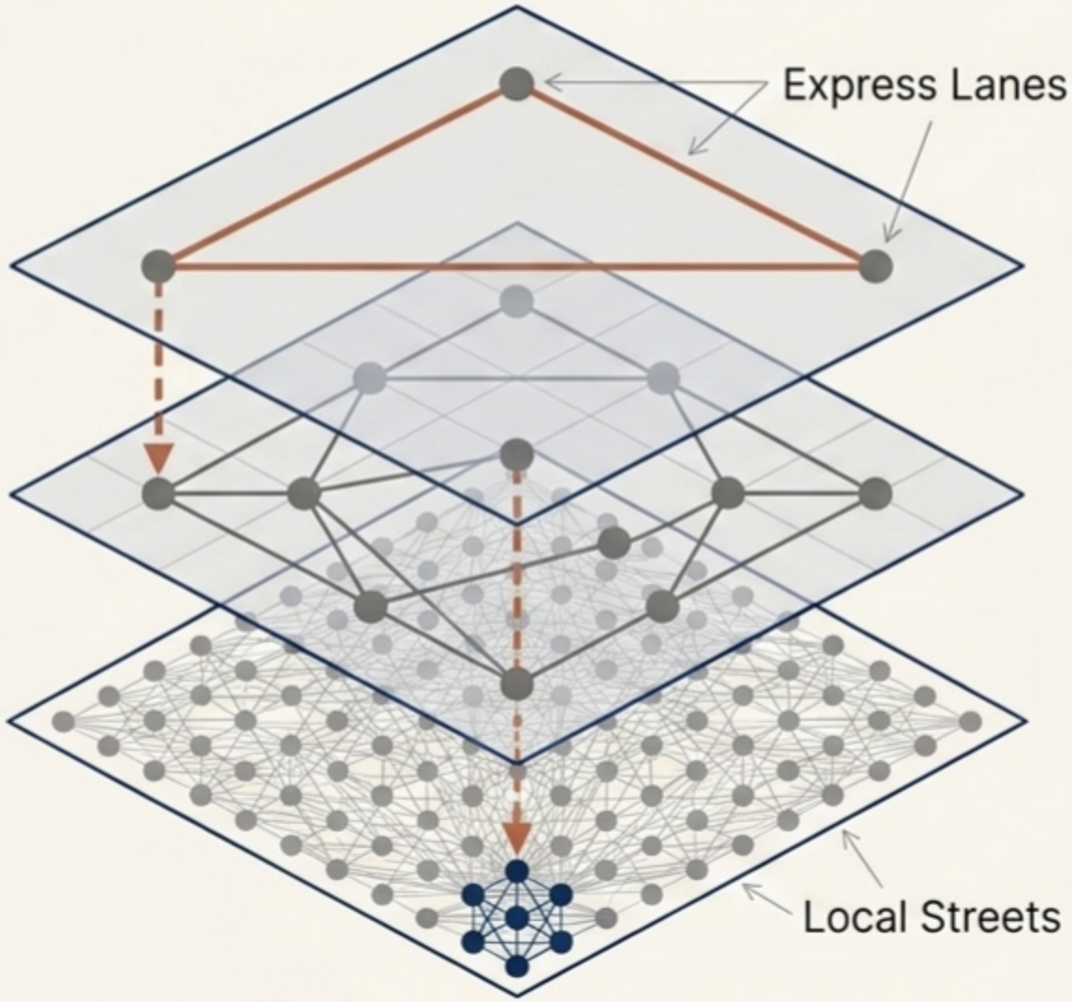
A 0.95 recall score means the index successfully retrieved 95% of the exact mathematical answers, but at a fraction of the computational cost.

# HNSW: The Express Lane Graph

HNSW routes queries through hierarchical express lanes to find local neighborhoods.

### Mechanism

Search hops greedily down layers.




### Constraint

Highly memory-hungry.

### Performance

Exceptionally fast, very high recall.

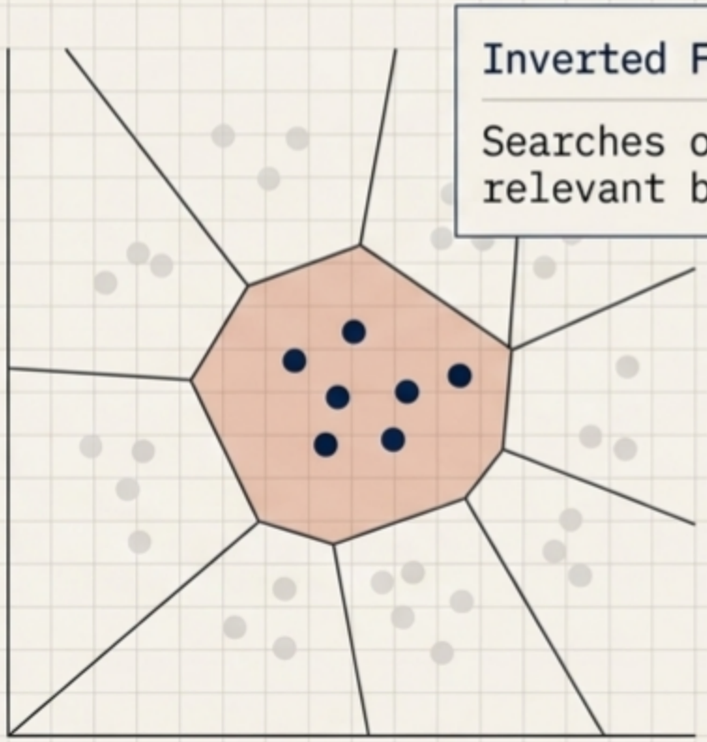
### Status

The industry default for sub-billion scale.

# IVF & PQ: Clustering and Compression

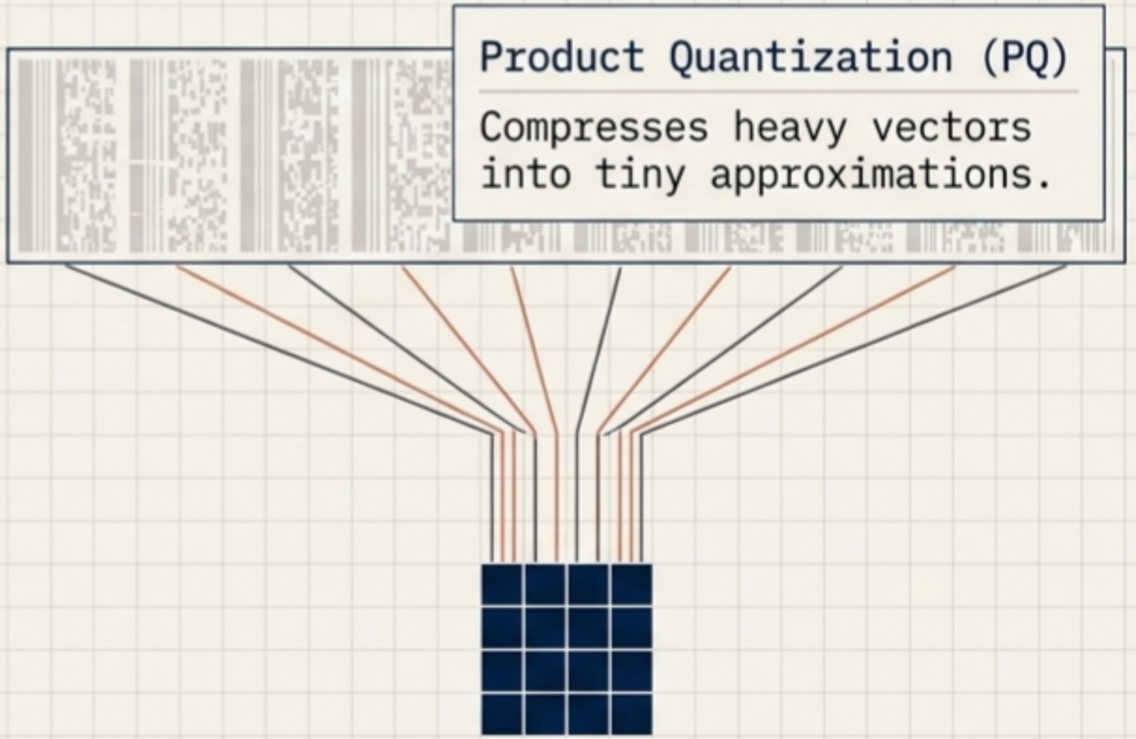
IVF and PQ combine territorial clustering with aggressive vector compression.

Often combined (IVF+PQ). IVF restricts the search space, while PQ compresses vectors so millions can fit within limited RAM constraints.



**Inverted File Index (IVF)**

Searches only the most relevant buckets.



**Product Quantization (PQ)**

Compresses heavy vectors into tiny approximations.

# Selecting Your Index

Selecting an index balances raw speed, memory footprint, and recall accuracy.

| Metric / Trait | HNSW | IVF + PQ |
|---|---|---|
| Mechanism | Graph routing | Bucket search + Compression |
| Speed |  Highest |  Fast |
| Memory Footprint |  High Memory |  Low Memory |
| Recall Accuracy |  Highest |  Good |
| Best For | Default for sub-billion scale | Extreme scale where RAM is a bottleneck |

# Algorithms Are Not Infrastructure

Raw algorithms require surrounding infrastructure to serve production applications.

**Raw Index (e.g., FAISS)**

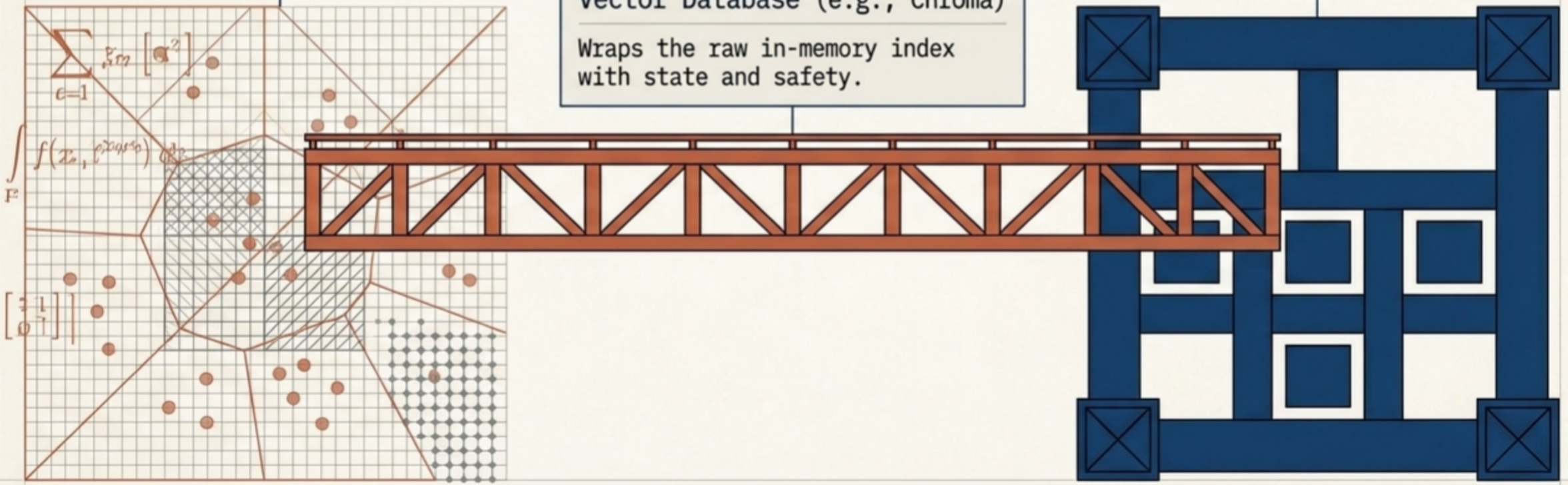
Brilliant, fast, in-memory vector math. Fundamentally volatile and stateless.

**Vector Database (e.g., Chroma)**

Wraps the raw in-memory index with state and safety.

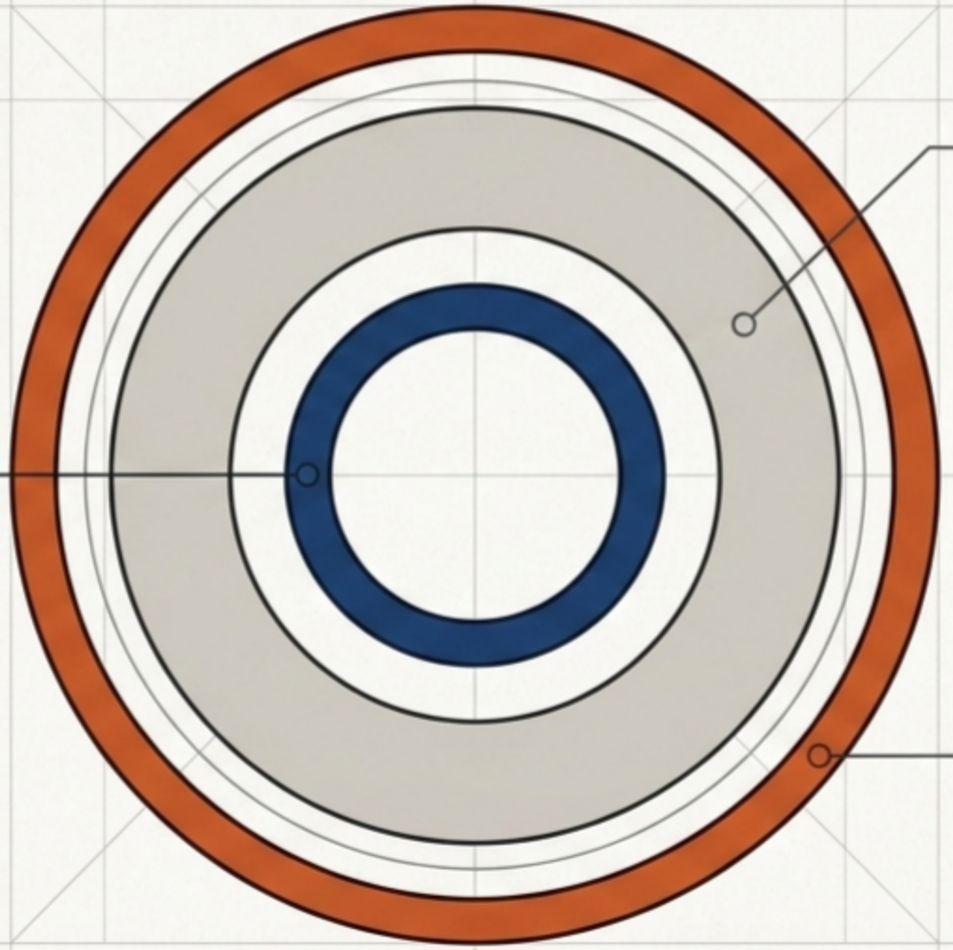
**Production App**

Requires scale, safety, context, and multi-tenancy.



# The Vector Database Value Add

Vector databases wrap raw math in persistence, filtering, and state management.



**Inner Core: Raw ANN Index**

Executes the fast vector math.

**Middle Layer: Metadata & Filtering**

Applies pre/post-filtering based on contextual tags.

**Outer Shell: Persistence & CRUD**

Handles disk storage, CRUD operations, and serves API endpoints.

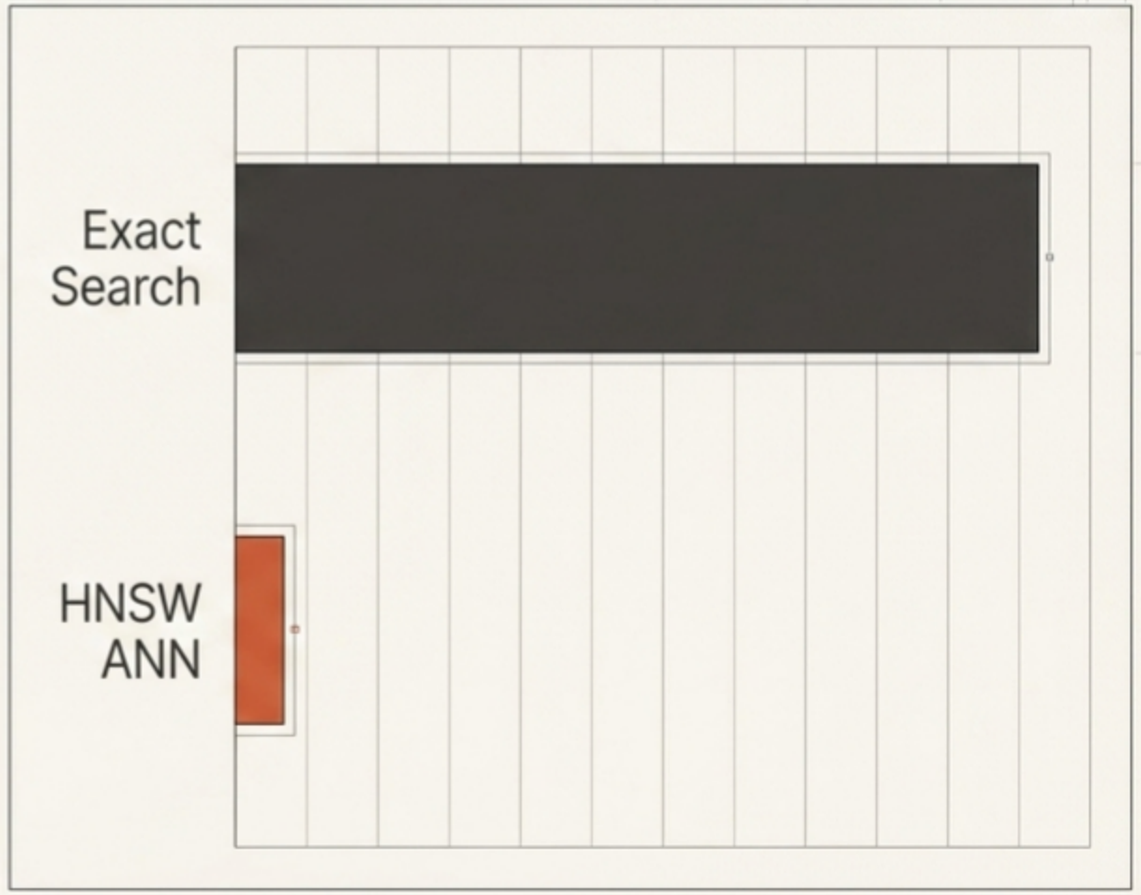
# The Trade-off in Practice

The recall-speed trade-off yields millisecond latency with near-perfect accuracy.

**Dataset Setup:**
~100k embedding-like vectors.

**~4x Faster execution time vs. exact search.**

**Recall@10 = ~0.999 (virtually zero loss in practical accuracy).**



# Key Terms Reference

The vocabulary of scalable semantic search.

## ANN

Approximate Nearest Neighbor search.

## Recall@k

Metric for algorithmic accuracy.

## HNSW

Layered graph-based index.

## IVF

Territorial cluster/bucket index.

## PQ

Aggressive vector compression.

## FAISS

Meta's raw in-memory index library.

## Chroma

Full-featured vector database.

## Metadata Filtering

Searching by tags alongside vectors.

# Check Your Understanding

Test your understanding of search scalability.

| | |
|---|---|
| **Q:** Why does brute-force search stop being viable as the corpus grows? | **A:** Its per-query cost grows linearly; touching every vector is too slow for production. |
| **Q:** What does ANN approximate, and how do we measure the cost? | **A:** It approximates the true nearest neighbors; the exactness lost is measured by Recall@k. |
| **Q:** What does a vector database add over a raw ANN index? | **A:** Persistence, metadata filtering, CRUD operations, and application serving. |