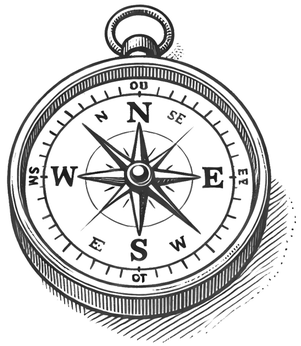
Sketches in this talk are generated by DALL·E



Compass: Encrypted Semantic Search with High Accuracy

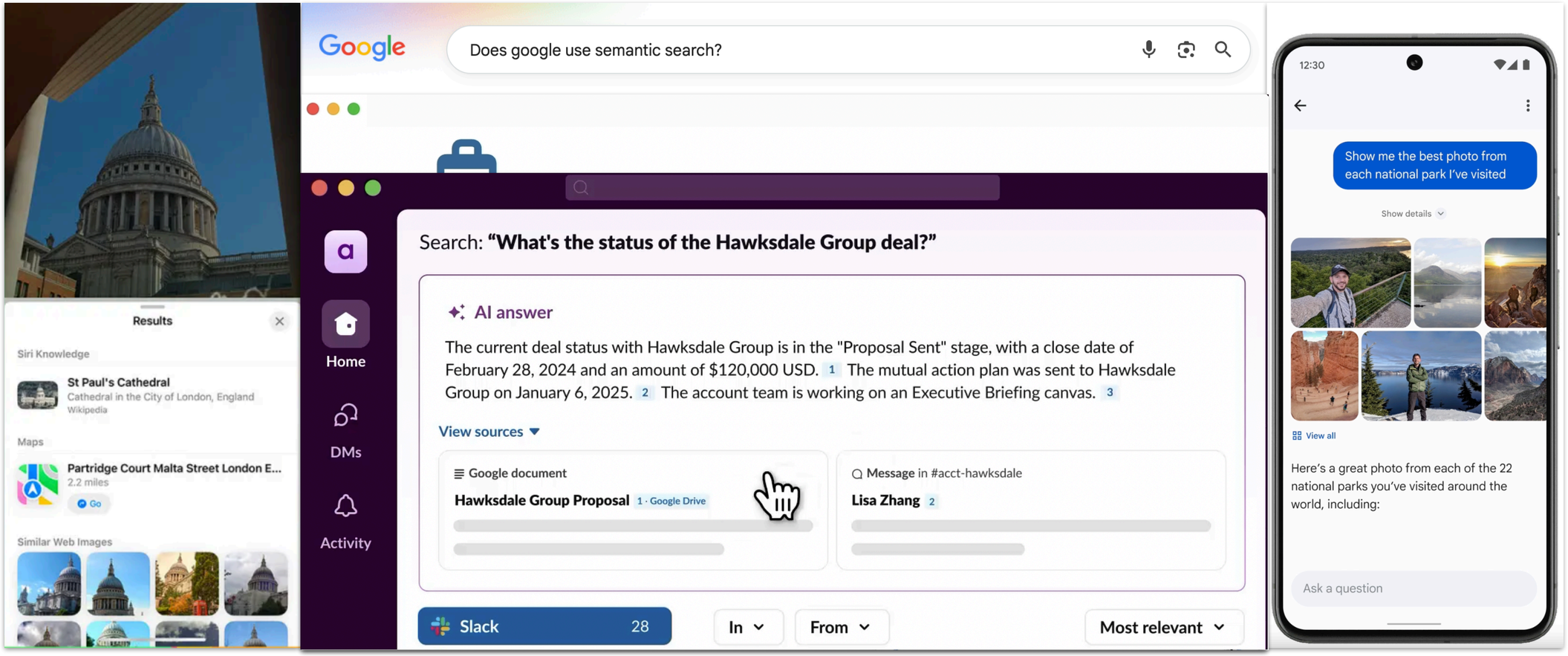
OSDI'25

Jinhao Zhu

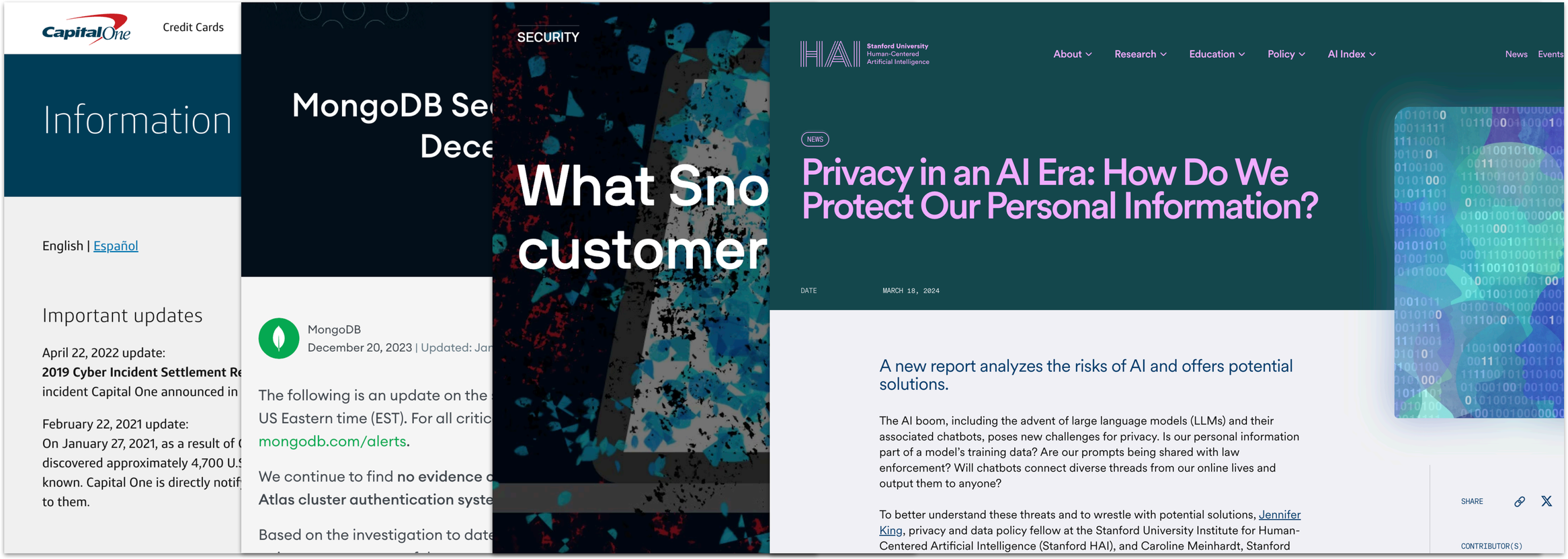
Joint work with Liana Patel, Matei Zaharia, and Raluca Ada Popa

UC Berkeley Stanford

Semantic Search Is Everywhere



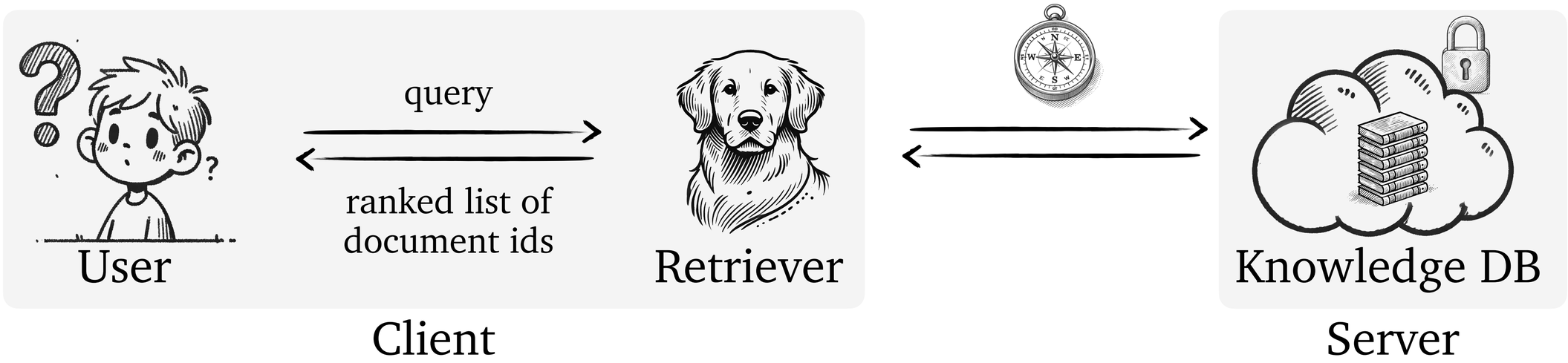
Data Breaches Still Happen



CapitalOne 2019 MongoDB 2023 Snowflake 2024 Privacy Concerns on GenAI Services

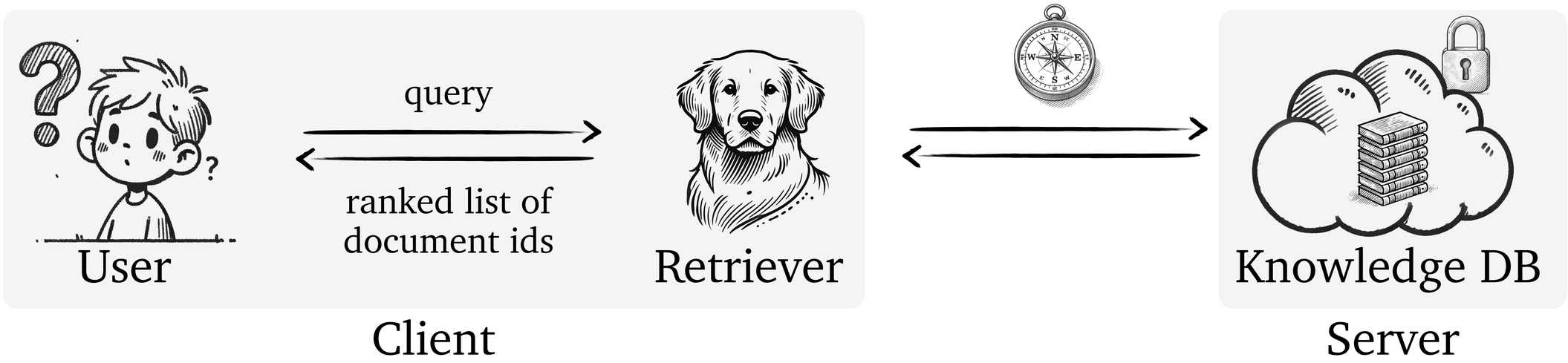
We need an encrypted semantic search scheme!

Compass



Private semantic search on an encrypted DB of embeddings.

Goal

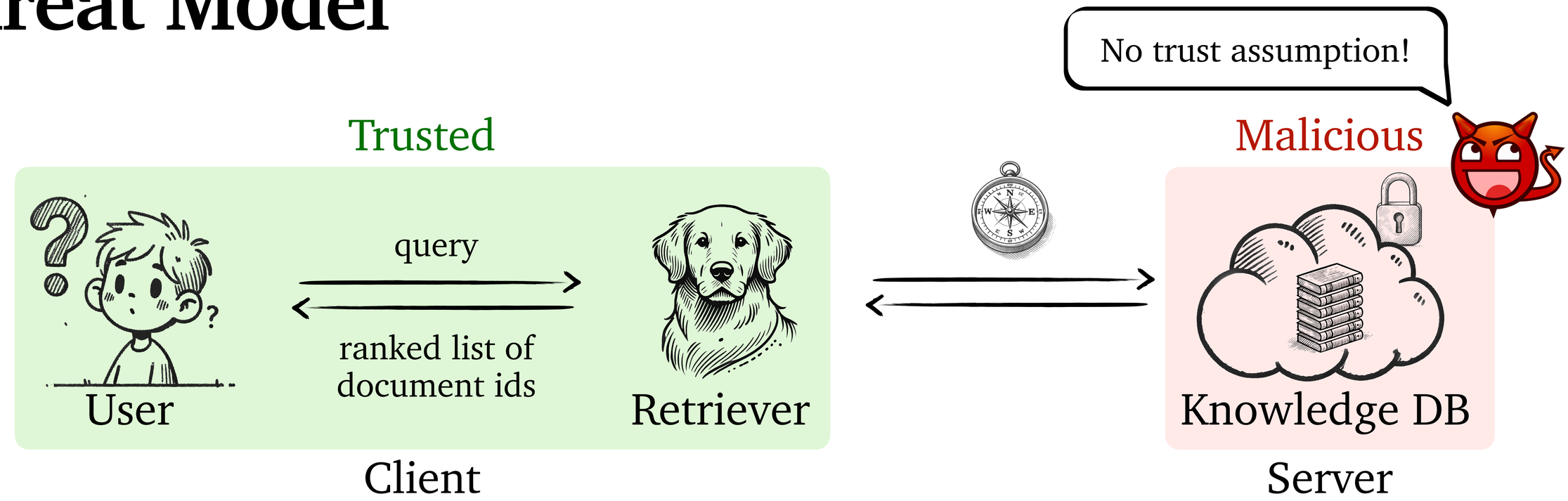


#1 Accuracy: On par with the state-of-the-art plaintext semantic search.

#2 Efficiency: Sub-linear to the size of the database.

#3 Security

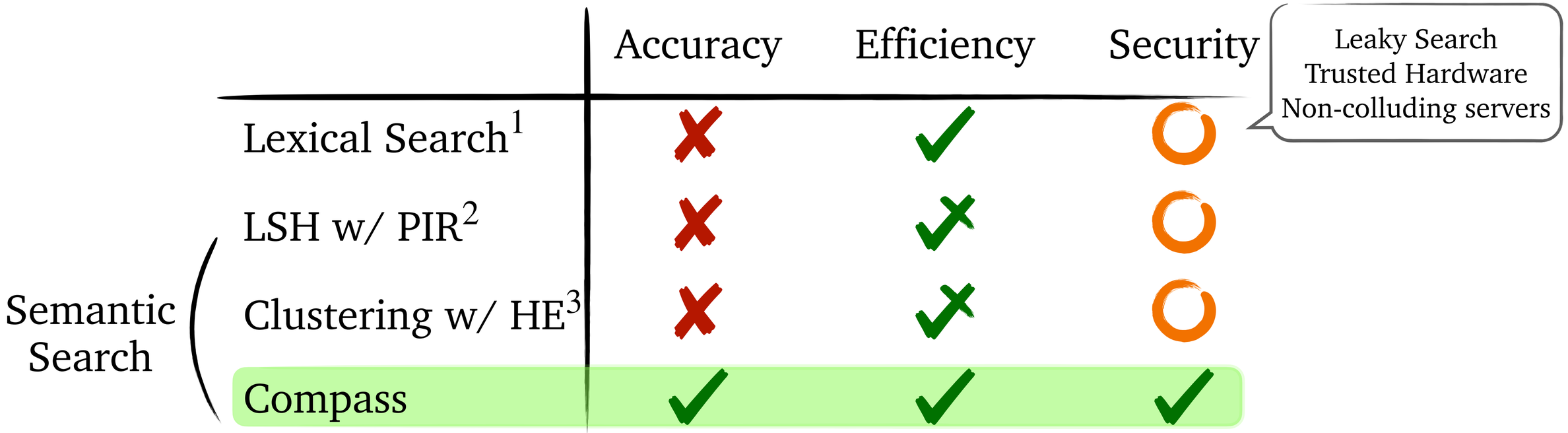
Threat Model



#3 Security:

- Server cannot learn DB, query and results.
  - No access patterns leakage [IKK12, CGP+16,…].
- Server cannot tamper with results undetectably
- **Non-goal:** Timing & DDOS attacks

Prior works



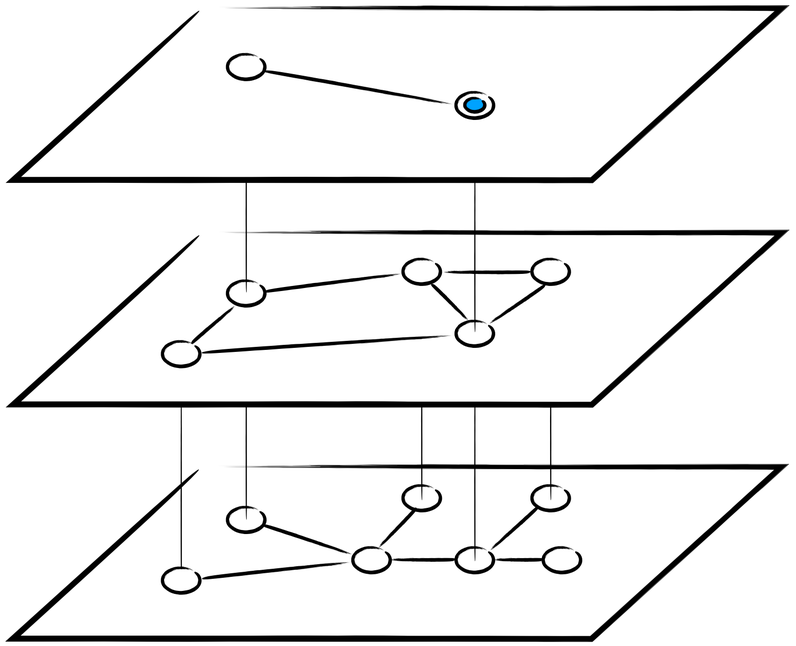
1: CGKO06, KPR12, SPS13, NPG14,CJJ+13, CJJ+14, MPC+18, AKM19,DFL+20, SWLL21, DRPI22 …

2: SLD22 3: HDCZ23

How to achieve the SOTA plaintext search quality?

~~Linear Scan~~

Hierarchical **N**avigable **S**mall **W**orld [MY16]

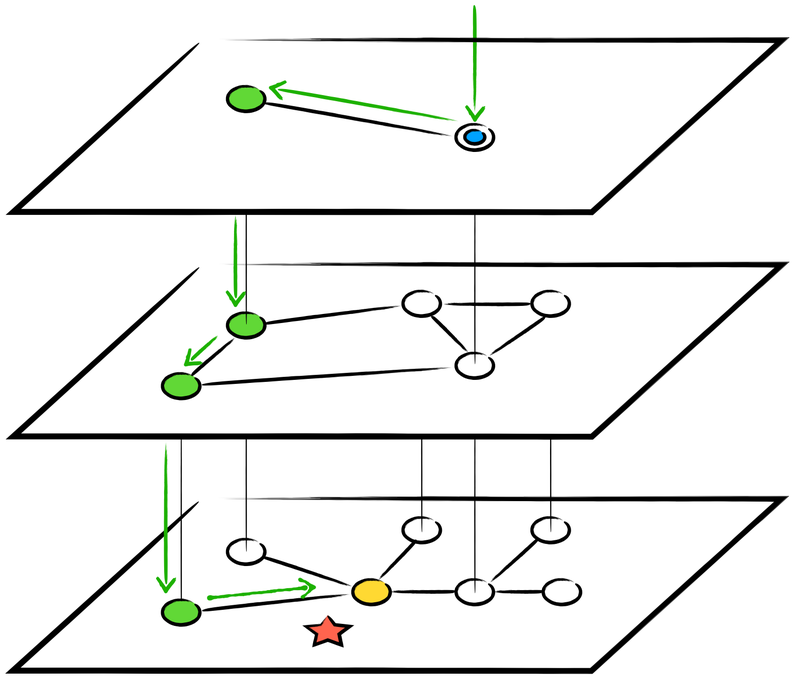


A state-of-the-art search index, widely used by industry: drant Pinecone

Hierarchical **N**avigable **S**mall **W**orld [MY16]

Search Workflow

🤔: How to support such a complex graph walk in a secure way?



- Entry Point
- Query
- Candidate Node
- Nearest Node

① Start from the top-layer entry point
② Greedily search for the local optimum
③ Shift to the next layer
④ Repeat 2-3 until the bottom layer

Compute-storage Separation

Client
(Compute)

Network

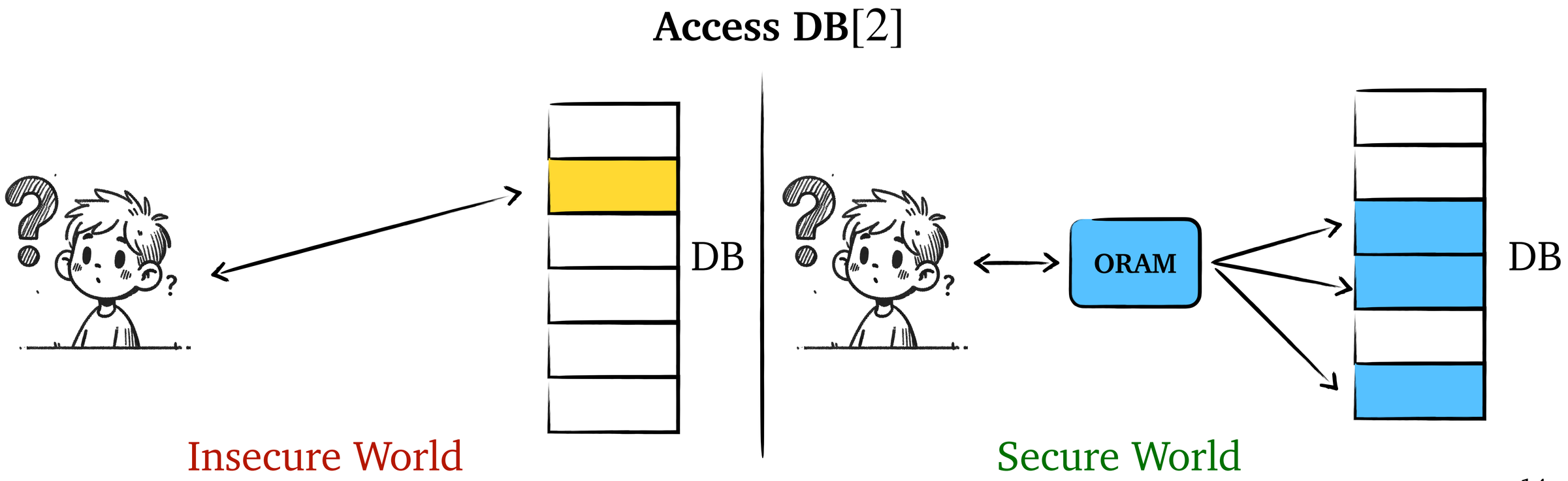
Server
(Storage)

🤔: Private Information Retrieval[CGKS95]?

🫠: $O(N)$[HSS08]

Oblivious RAM [GO96, PathORAM, ..]

A cryptographic technique that hides access patterns to storage, ensuring that an observer cannot infer which data is being accessed.



Compass core ideas

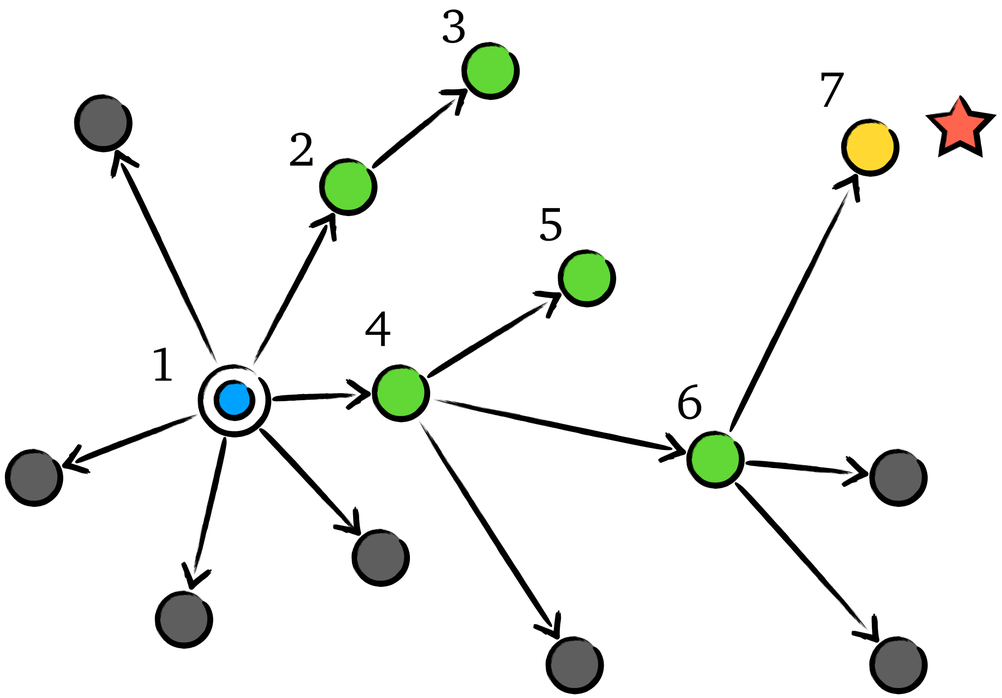
HNSW	ORAM
✓ logarithmic in search complexity[1]	✓ logarithmic in communication
✓ Bounded node degree	✓ distance-metric agnostic
	✓ easy to obtain malicious security

Challenge: HNSW is designed for local search.

Directly applying HNSW over ORAM is inefficient.

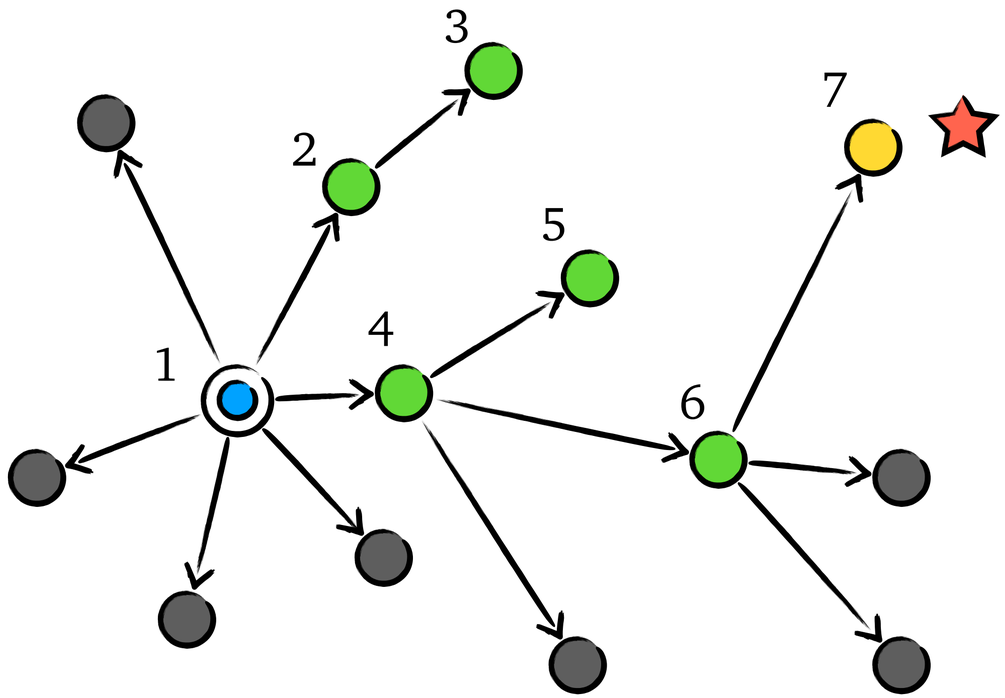
1. Assuming the HNSW graph approximates a Delaunay graph

Example



- Entry Point
- Query
- Candidate Node
- Nearest Node
- Visited Node

Cost



- Entry Point
- Query
- Candidate Node
- Nearest Node
- Visited Node

Round trips

Bandwidth

It takes **7 search steps** and **13 visited nodes** to reach the nearest node.

Cost

A real example

On SIFT1M dataset:

- it takes **36** search steps and **1000+** visited node to achieve a good accuracy

High network bandwidth

High number of network round trips

Compass core ideas

HNSW

- ✓ logarithmic in search complexity
- ✓ Bounded node degree

ORAM

- ✓ logarithmic in complexity
- ✓ distance-metric agnostic
- ✓ easy to obtain malicious security

A co-design between search and crypto

Over-the-network Graph traversal algorithm

Graph-traversal tailored ORAM

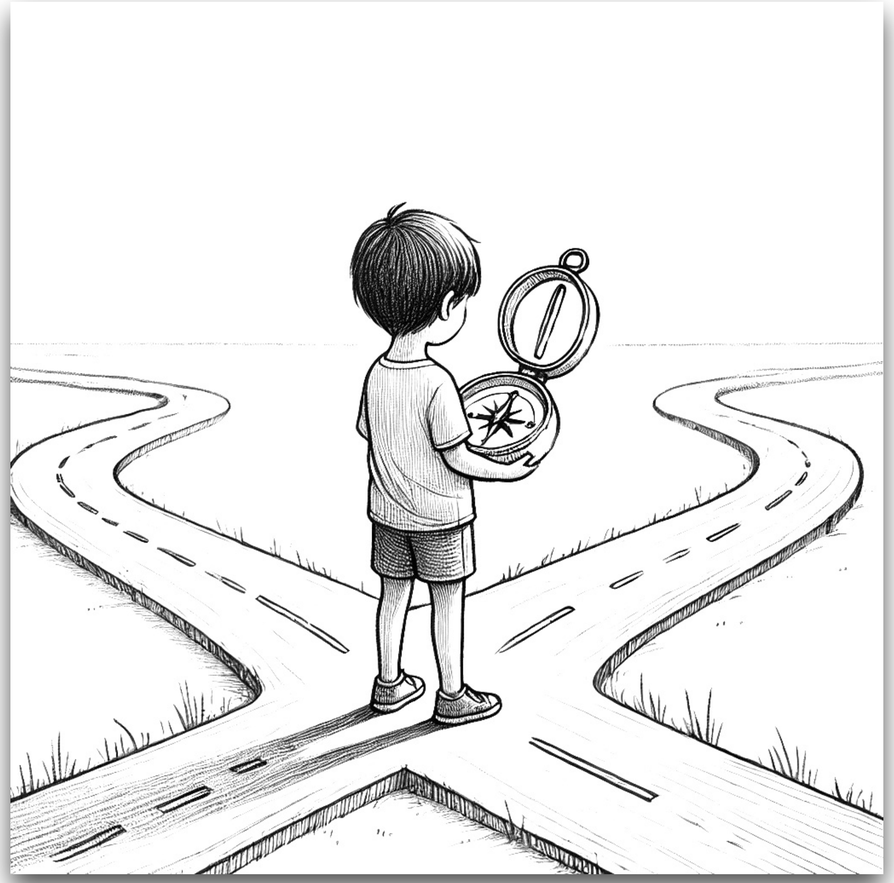
Over-the-network Graph traversal algorithm

#1: Directional Neighbor Filtering → Reduce network bandwidth

#2: Speculative Neighbor Prefetch → Reduce network round trips

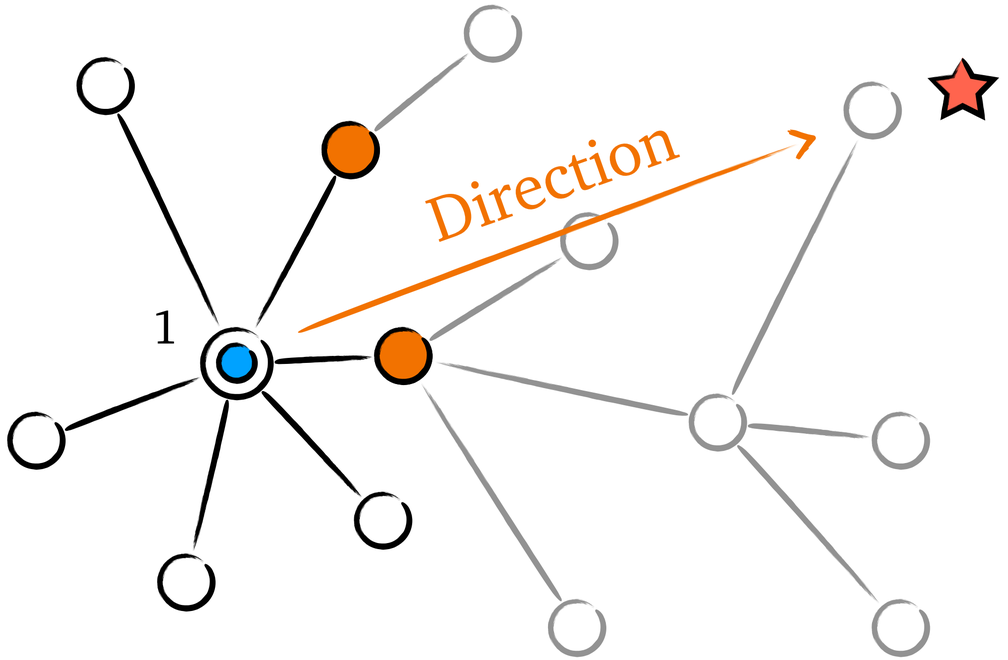
Directional Neighbor Filtering

Consider only a subset of neighbors that are in the direction of the query.



Directional Neighbor Filtering

Consider only a subset of neighbors that are in the direction of the query.

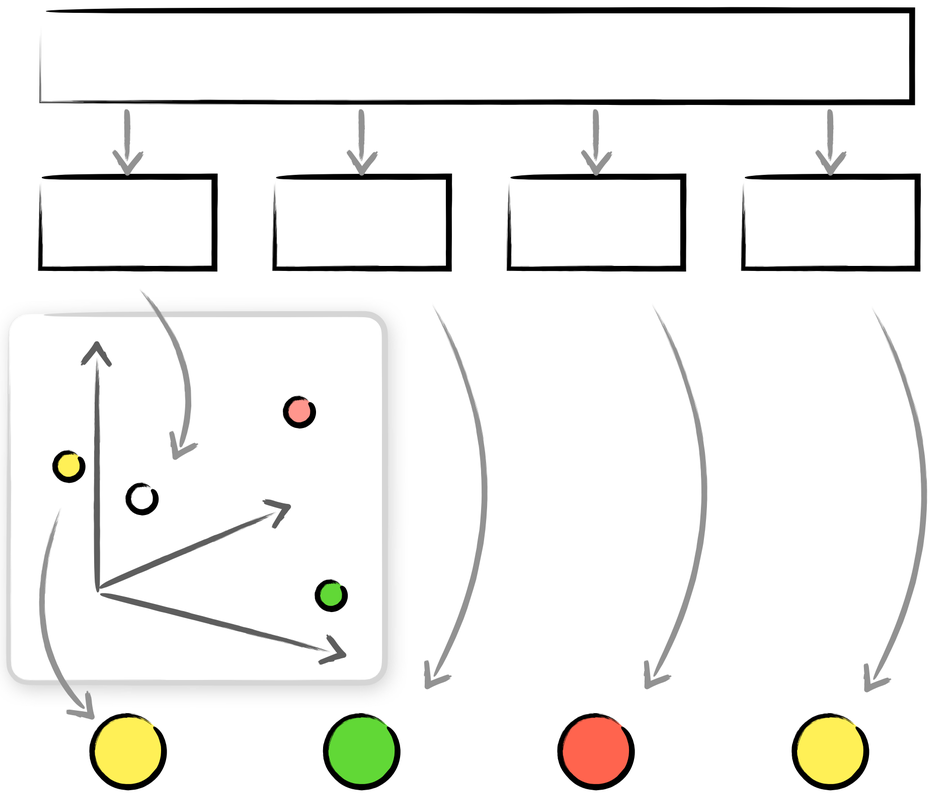


How to determine the direction of neighbors **without** their coordinates?

Quantized DB as Local Hints

Benefits:

- High Compression Rate:
  - 1-3% of the original DB size.
- Isn’t quantization lossy?
  - Accurate enough for directional filter.



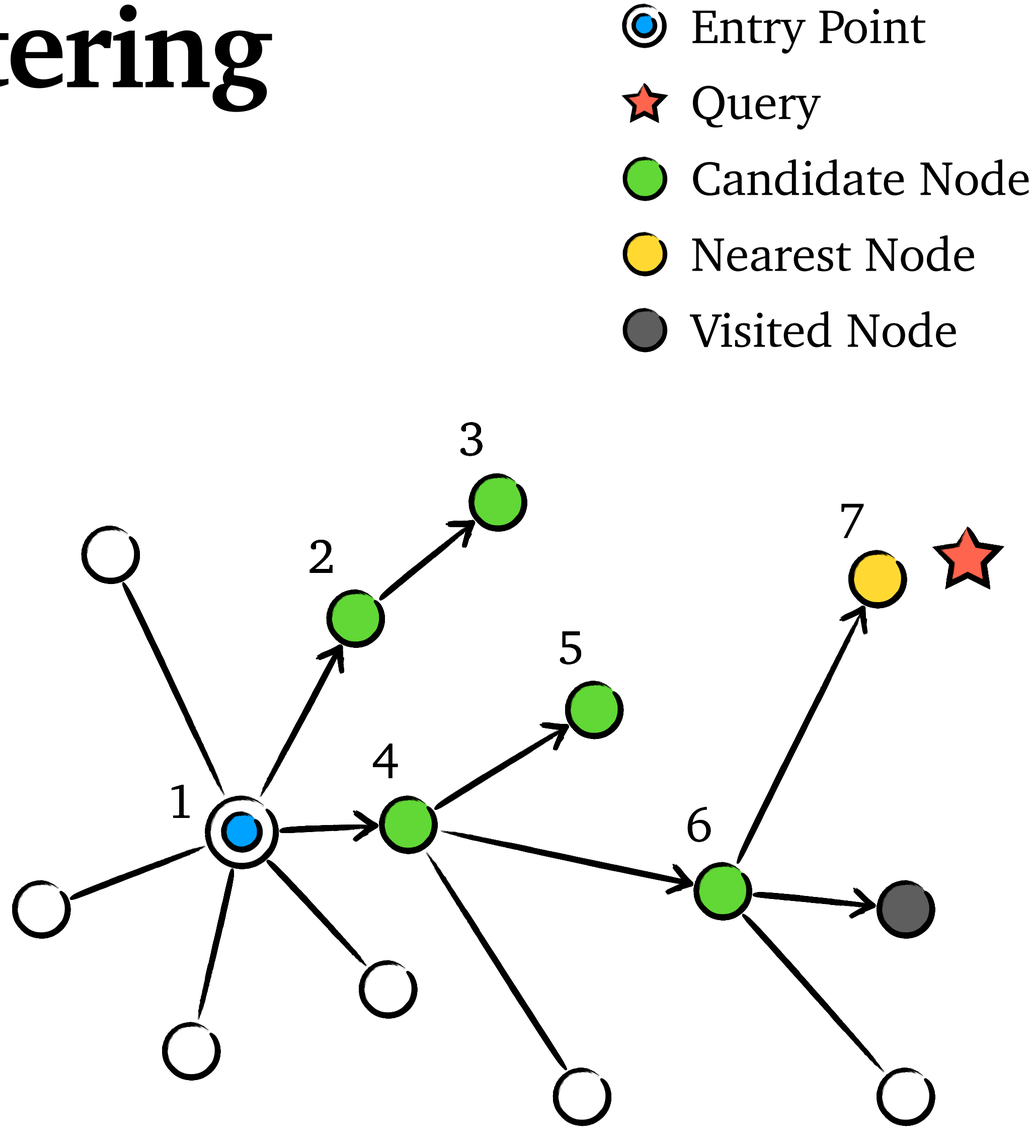
Product Quantization

Directional Neighbor Filtering

Example

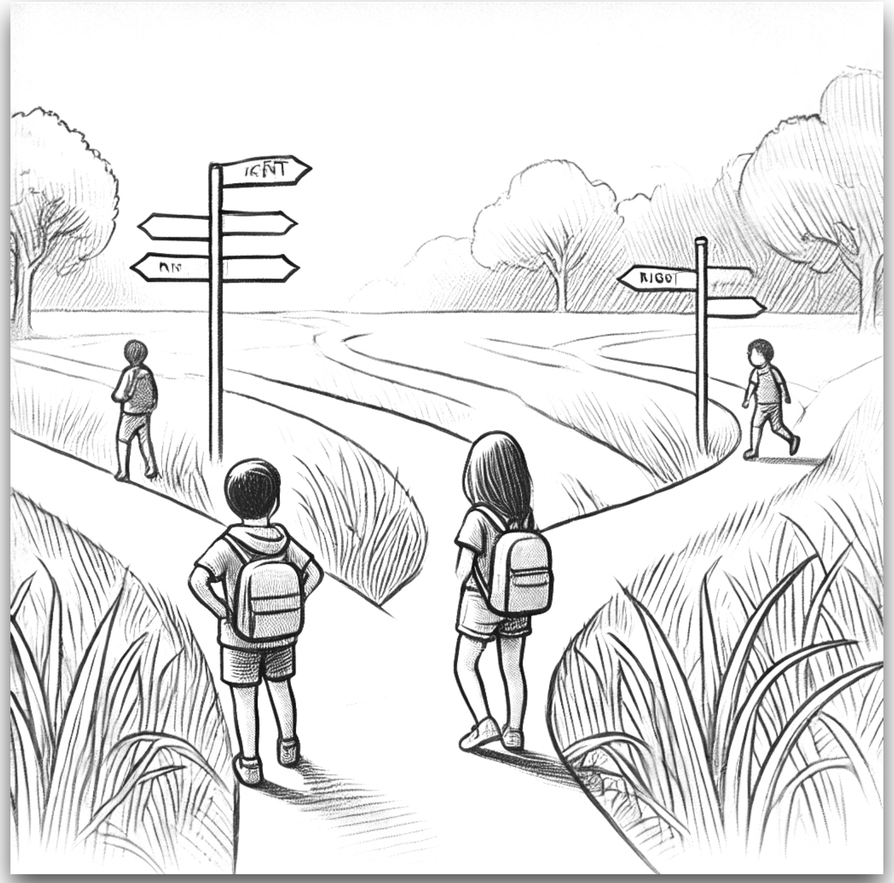
- Pick two neighbors per candidate
- Cost with this technique
  - **7** search steps and 7 visited nodes
- w/o this technique:
  - **7** search steps and **13** visited nodes

Empirical savings: 10×



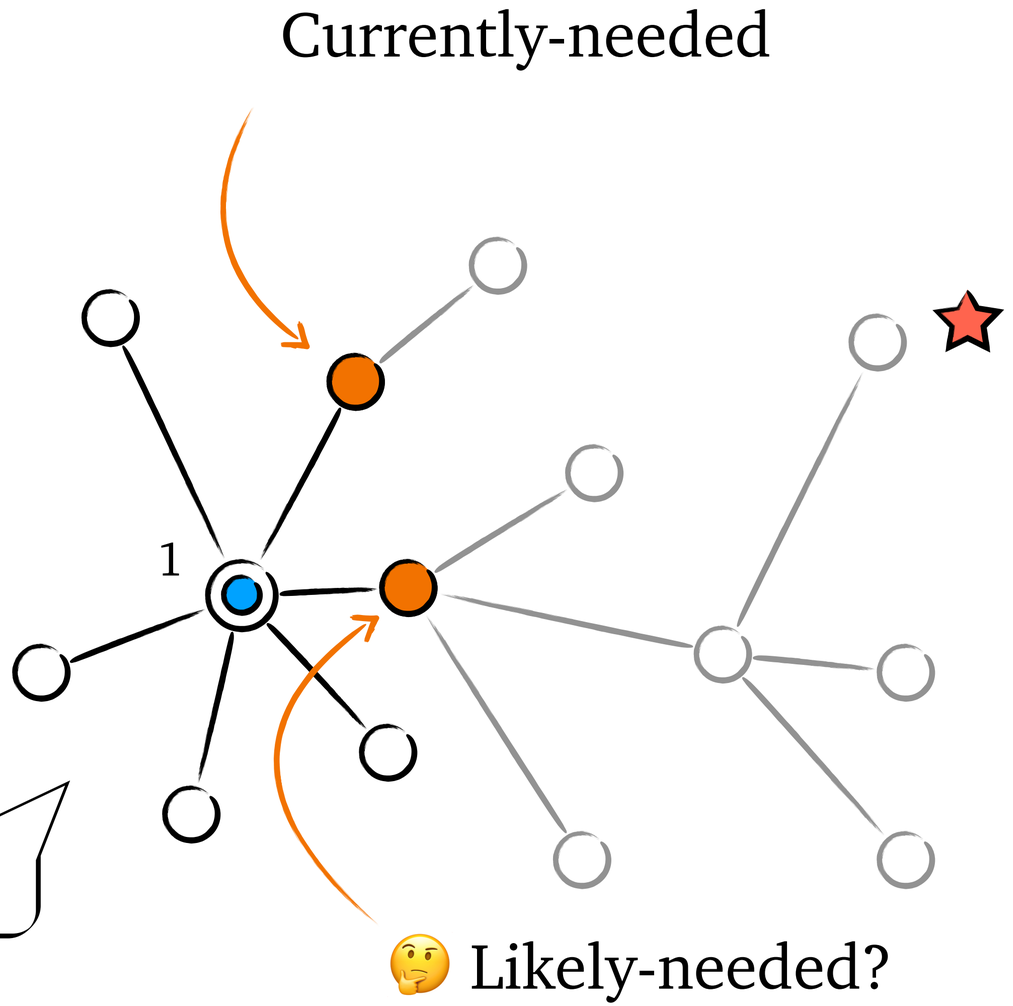
Speculative Neighbor Prefetch

Speculatively retrieves *likely-needed* nodes alongside the *currently-needed* node



Speculative Neighbor Prefetch

Speculatively retrieves *likely-needed* nodes alongside the *currently-needed* node



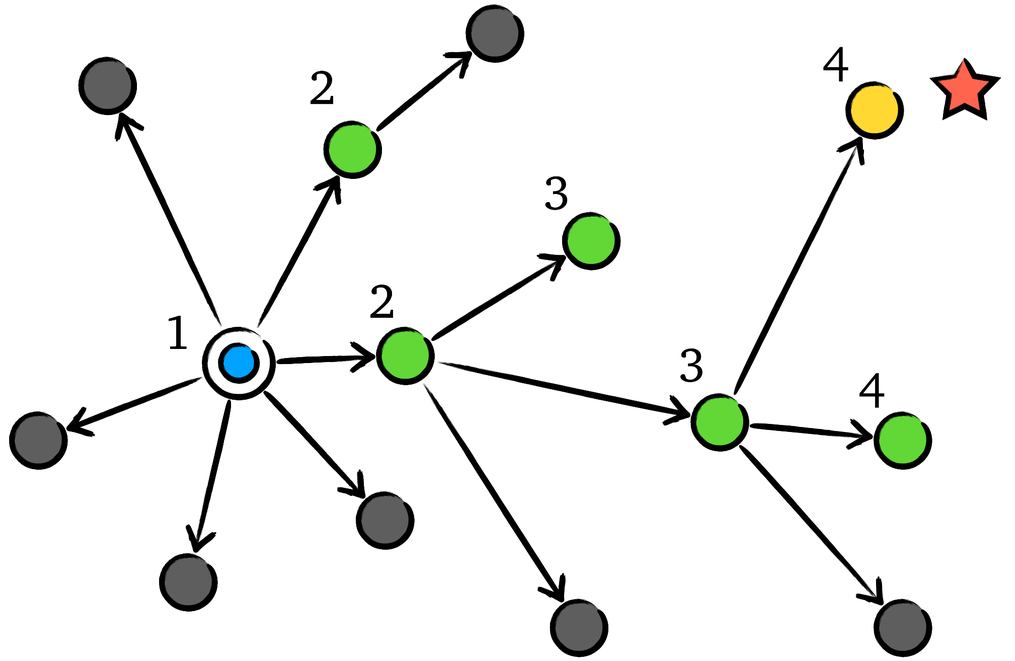
How to guess the likely-needed nodes?

The local candidate set!

Speculative Neighbor Prefetch

Example

- Select two candidate nodes per step

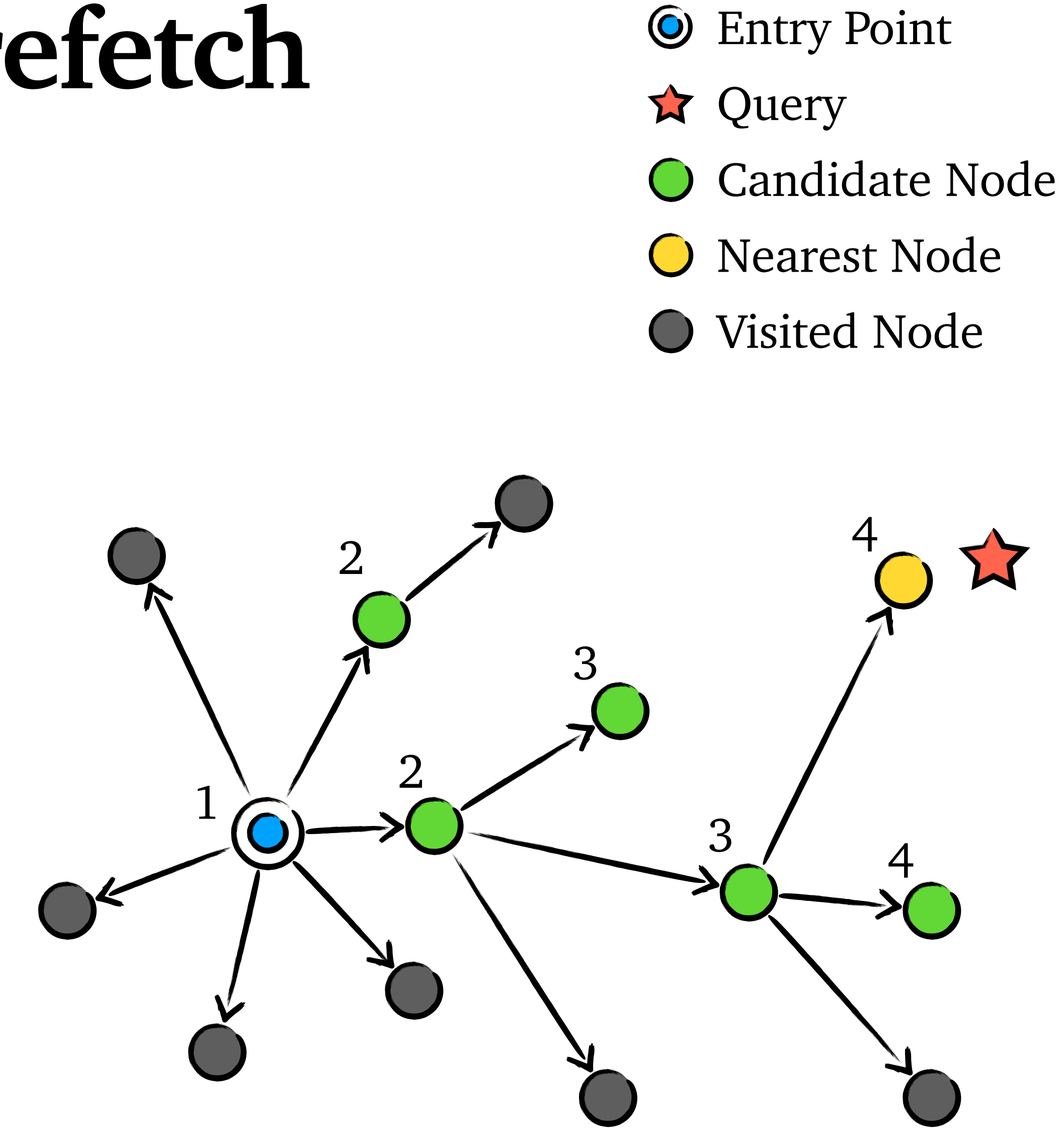


Speculative Neighbor Prefetch

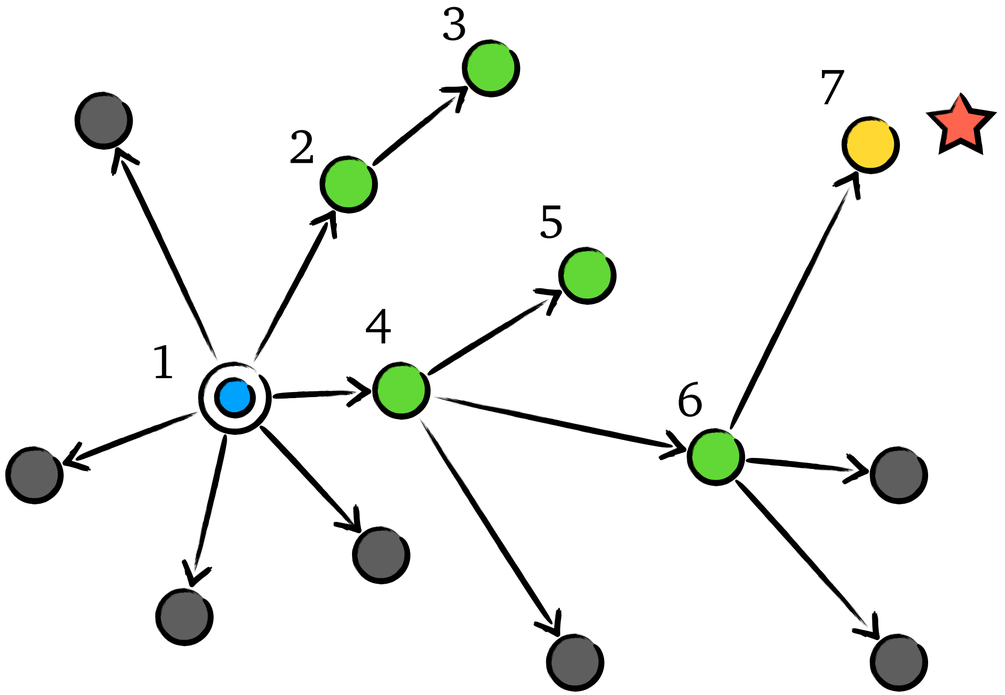
Example

- Select two candidate nodes per step
- Cost with this technique
  - 4 search steps and **13** visited nodes
- w/o this technique:
  - **7** search steps and **13** visited nodes

Empirical savings: 2 - 16×

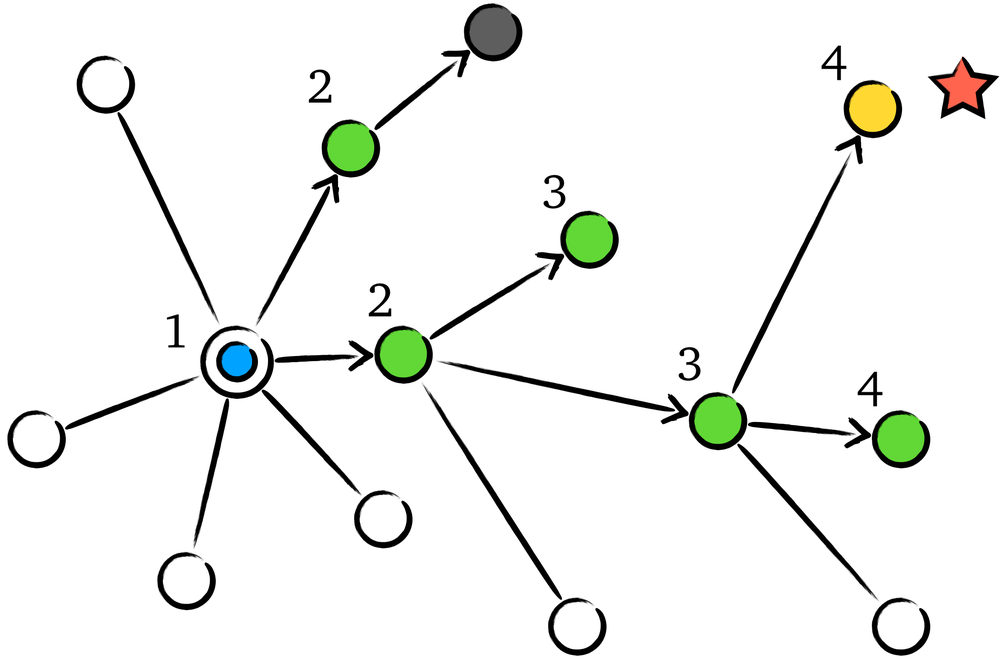


Put it all together



Classical

7 steps, 13 nodes



Compass

4 steps, 7 nodes

Empirical savings: 10×, 2-16×

Graph traversal-tailored ORAM

We make a white-box adoption of Ring ORAM [RFK+15] based on two observations:

#1: Requests to neighboring nodes within the same search step can be batched.

- Less network round trips.

#2: Nodes are visited at most once during a search request. Evictions can be deferred until the end of the query.

- Better user-perceived latency.

Please check our paper for malicious server protection, stash analysis and security proof!

Evaluation Setup

- Client: 8vCPUs and 32GB memory
- Server: 64vCPUs and 512GB memory
- Network:
  - Same-region: 3Gbps, 1ms RTT
  - Cross-region: 400Mbps, 80 ms RTT

Datasets

- Text-based:
  - **TripClick:** 1.5M 768-dimensional vectors
  - **MS MARCO:** 9M 768-dimensional vectors
- Image-based:
  - **SIFT1M:** 1M 128-dimensional vectors
  - **LAION:** 100K 512-dimensional vectors

Baselines

- Inv-ORAM (for text-based datasets only):
  - Use keyword search (TF-IDF) with inverted index.
  - Hide access pattern using ORAM (w/ batching).
- HE-Cluster:
  - Use HE to compute similarity score.
  - Reduce communication cost through clustering.

Accuracy

MRR@10

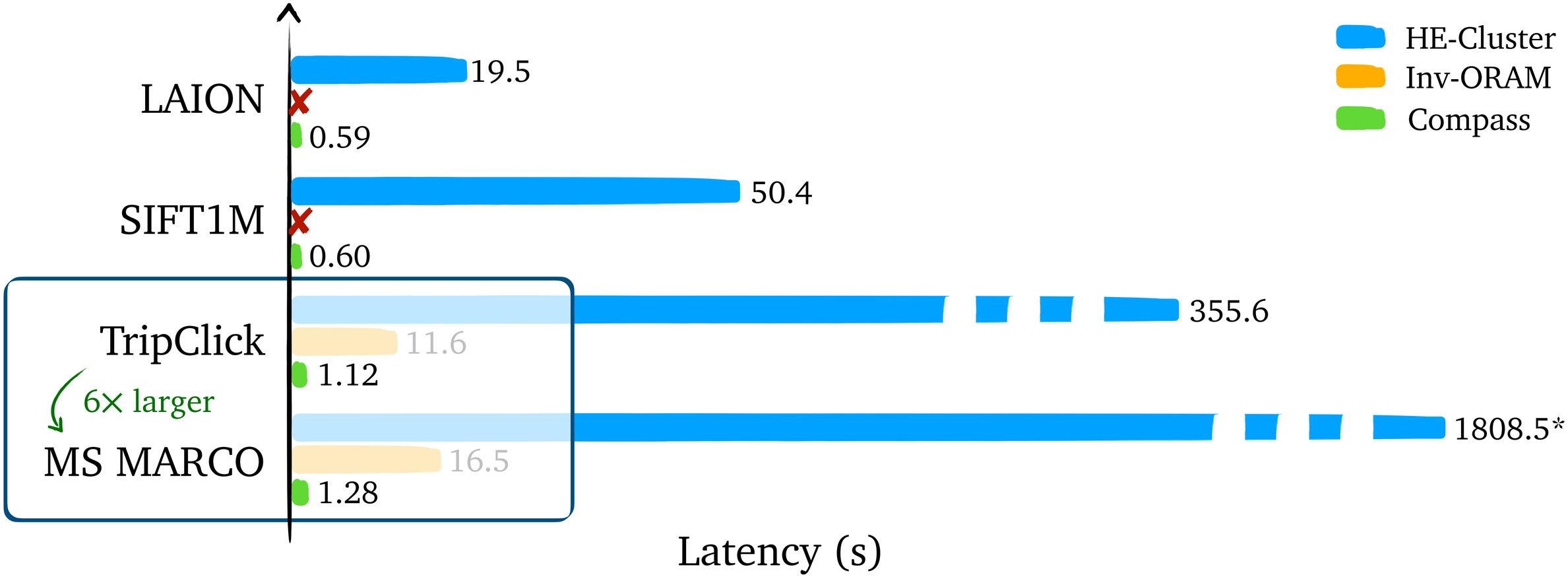
	Inv-ORAM	HE-Cluster	Compass	Vanilla HNSW	Optimal
LAION	N/A	0.99	0.97	0.99	1.0
SIFT1M	N/A	0.46	0.95	0.97	1.0
TripClick	0.24	0.14	0.30	0.30	0.31
MS MARCO	0.07	0.12	0.35	0.36	0.37

Compass's search quality matches the plaintext vanilla HNSW.

* indicates results are extrapolated due to OOM

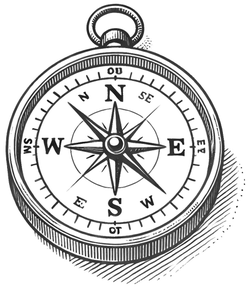
Latency

Over cross-region network



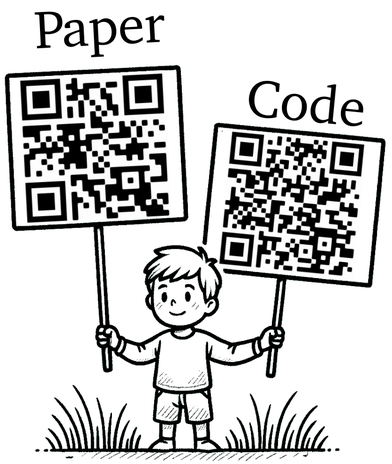
Compass achieves orders of magnitude lower latency than baselines

Conclusion



Compass is a semantic search system over encrypted embeddings that

1. matches the accuracy of state-of-the-art plaintext search algorithms,
2. achieves practical user-perceived latency,
3. provides strong security against malicious servers.



Thanks!

jinhao.zhu@berkeley.edu