Private web search

Alexandra Henzinger
MIT

Emma Dauterman
UC Berkeley

Henry Corrigan-Gibbs
MIT

Nickolai Zeldovich
MIT

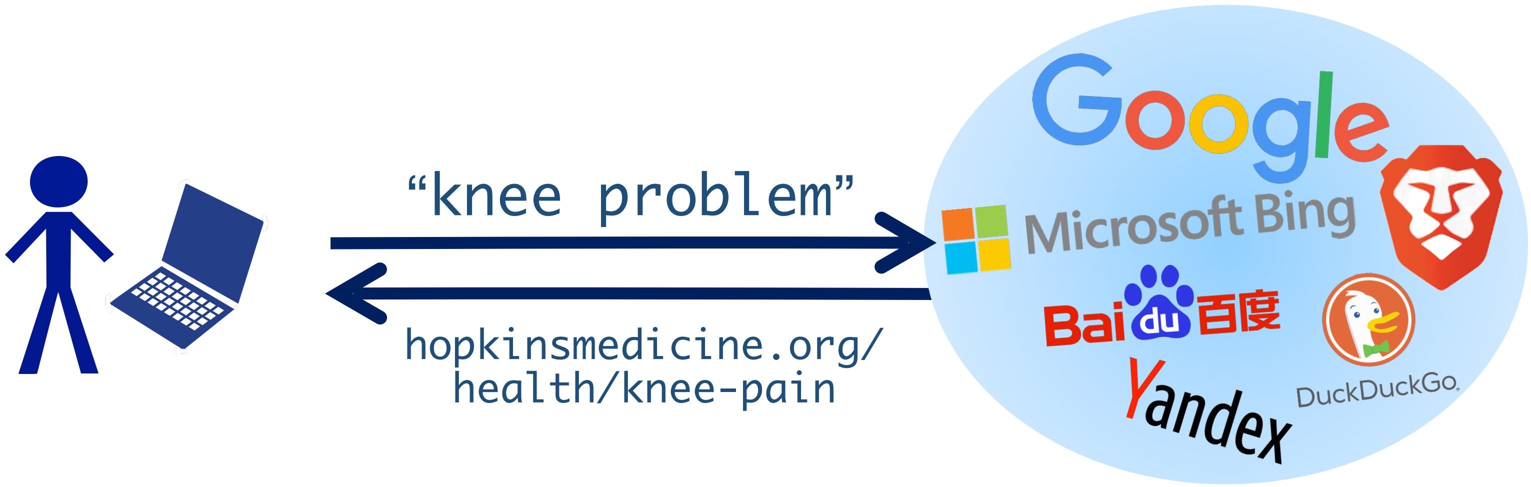
Appeared at SOSP 2023

Web-search queries reveal our sensitive data

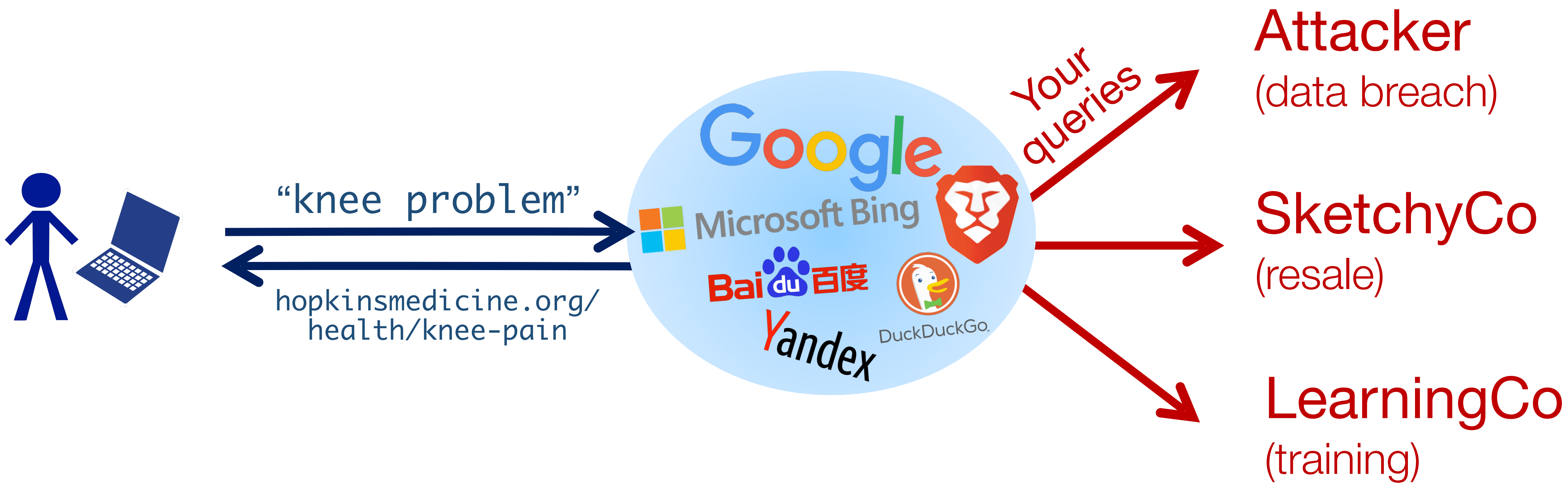
Health	`ballet knee problem`
Finances	`job opportunities in west palm beach`
Religion	`african american churches in norfolk va`
Citizenship	`application forms us citizen`

https://trec.nist.gov/data/million.query07.html

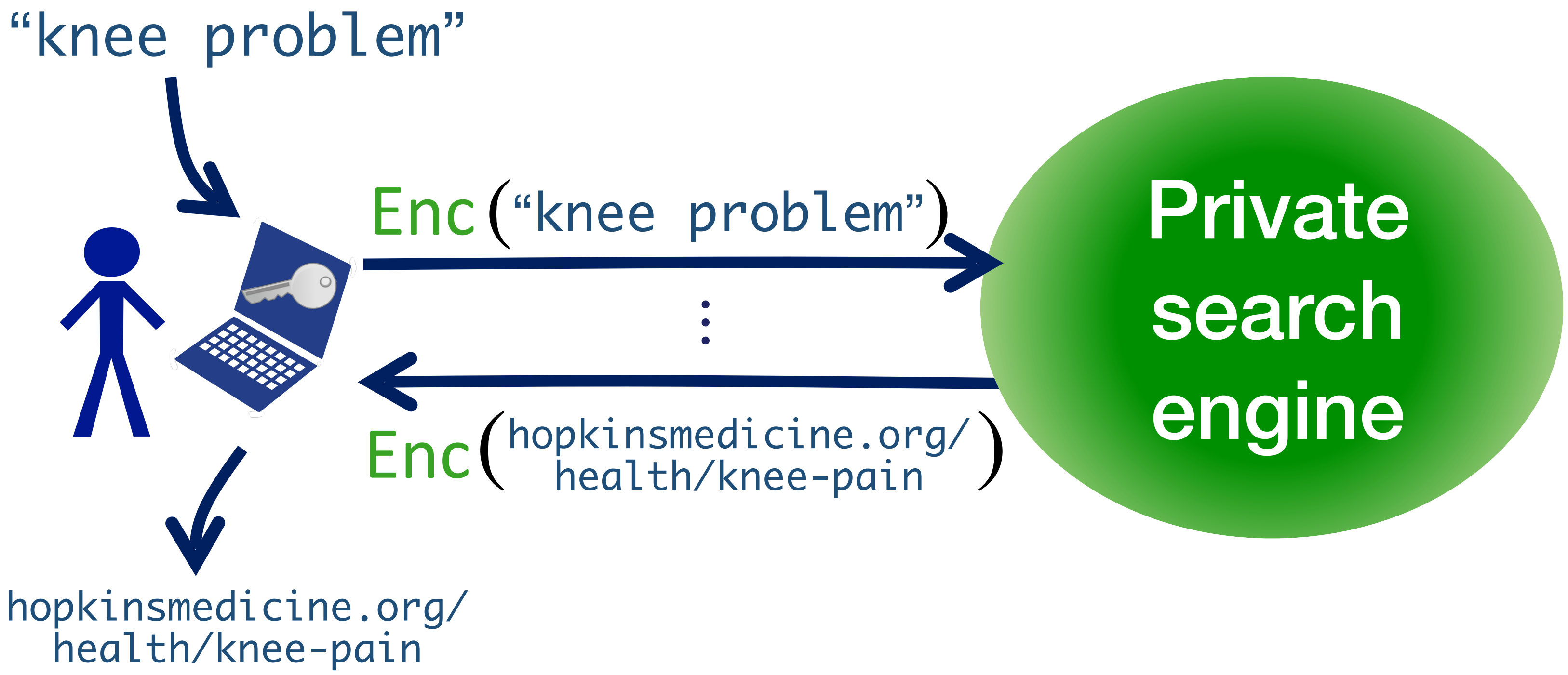
Today: Search engines learn our queries



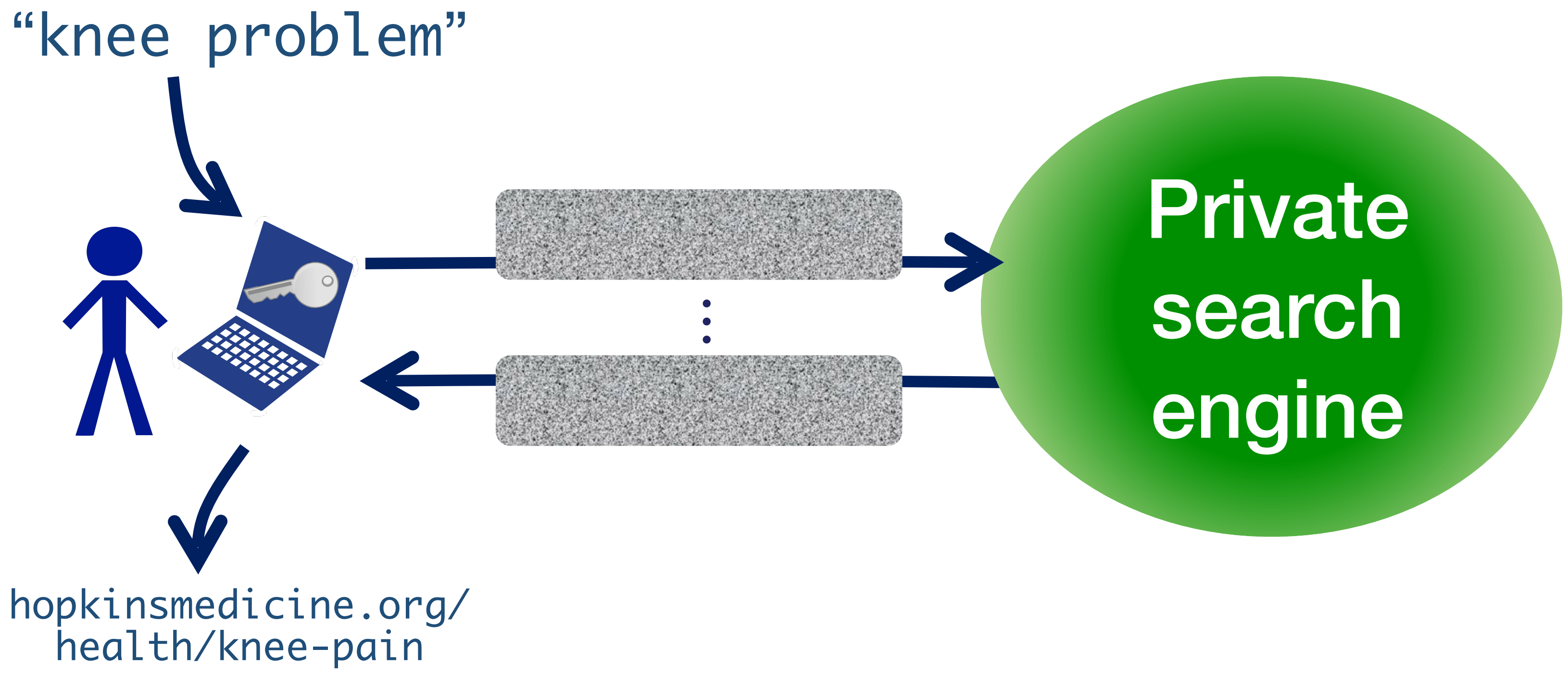
Today: Search engines learn our queries



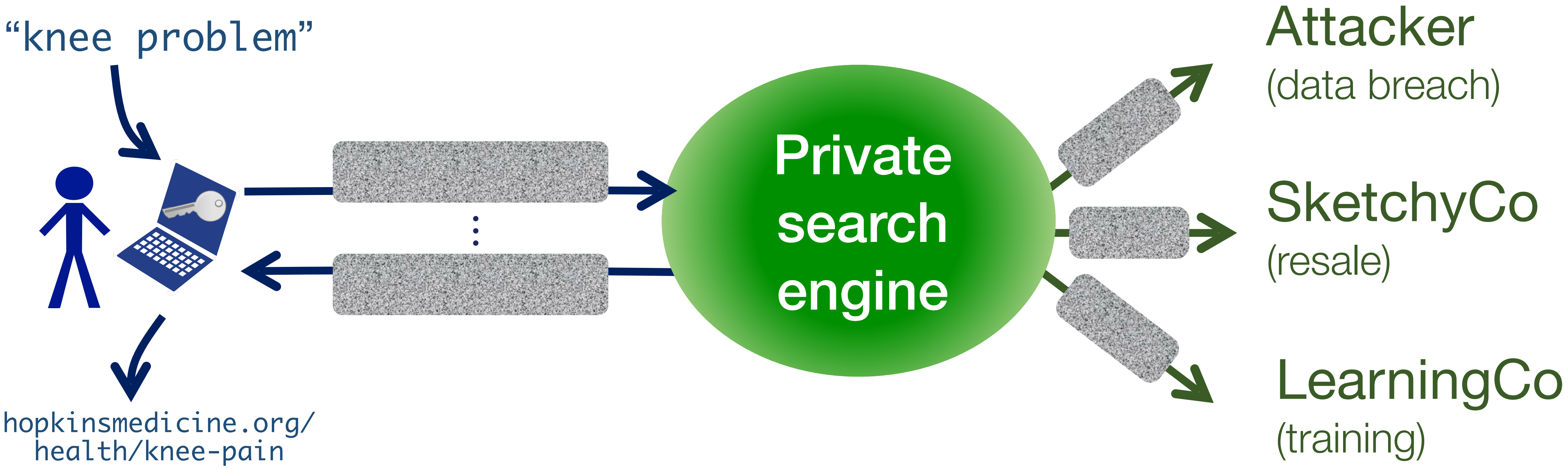
Goal: Search without revealing query



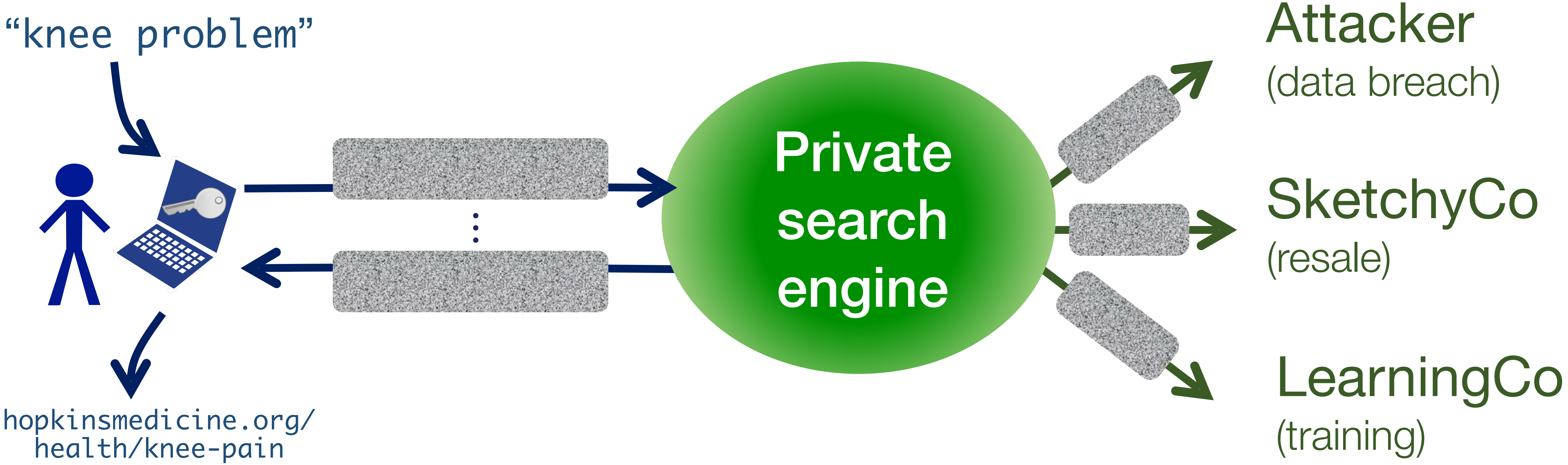
Goal: Search without revealing query



Goal: Search without revealing query

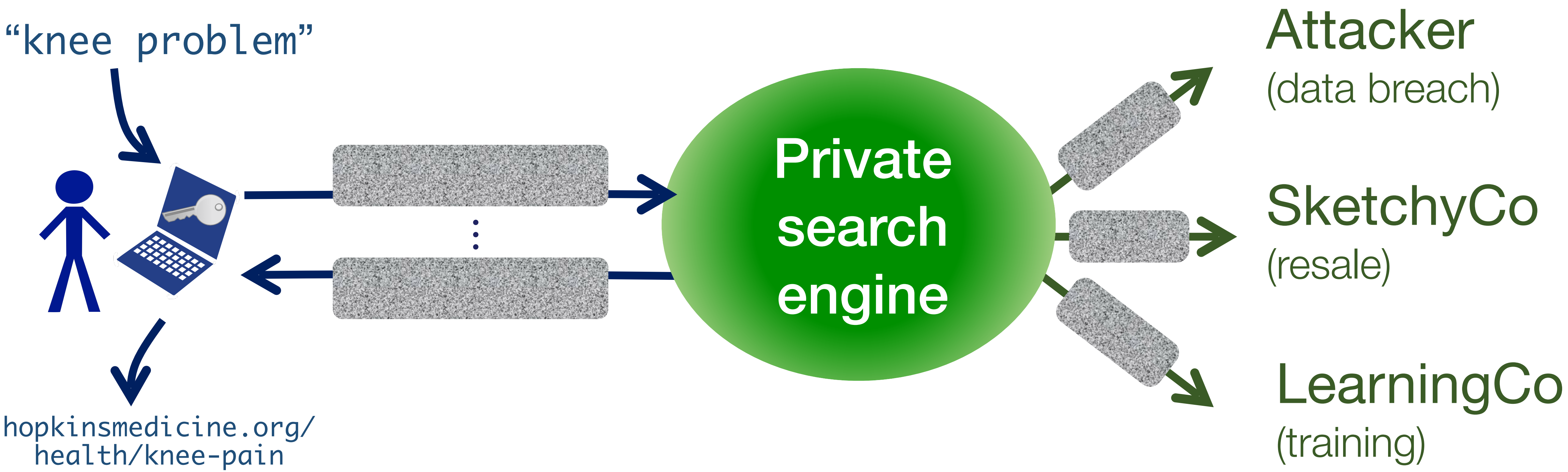


Goal: Search without revealing query



Non-goals:
- does not hide *when* the client makes searches
- does not guarantee integrity of search results
- does not hide subsequent HTTP(S) requests

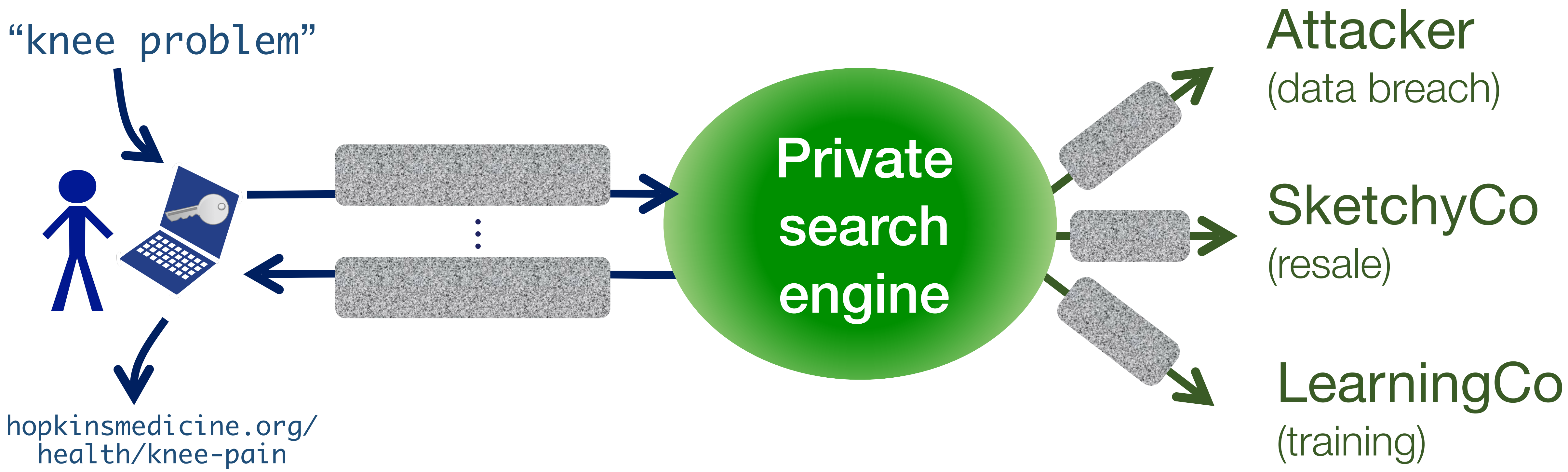
Goal: Search without revealing query



Theoretically possible: Fully homomorphic encryption [RAD’78, Gen’09]

But, classic search algorithms are very expensive to express as circuits

Goal: Search without revealing query



This work: Linearly homomorphic encryption suffices

Modern ML turns messy search computations into cheap, linear ones

Tiptoe: A private search engine

+ Search engine learns no information about the client's queries

  i.e., semantic security relying on LWE and ring-LWE

+ Supports text & image search

+ Searches over public web crawl (364M pages) in 2.7s of latency

  with 145 core-s of compute, 57 MiB of traffic, and 0.3 GiB of client storage

- Search results not yet as good as with non-private search engines

Private search on private data

- Searchable Encryption
  [SWP'00, CGKO'11, CryptDB'11, SPS'14, …]

- Oblivious RAM
  [GO'96, O'90, SVSRYD'13, Dory'20, …]

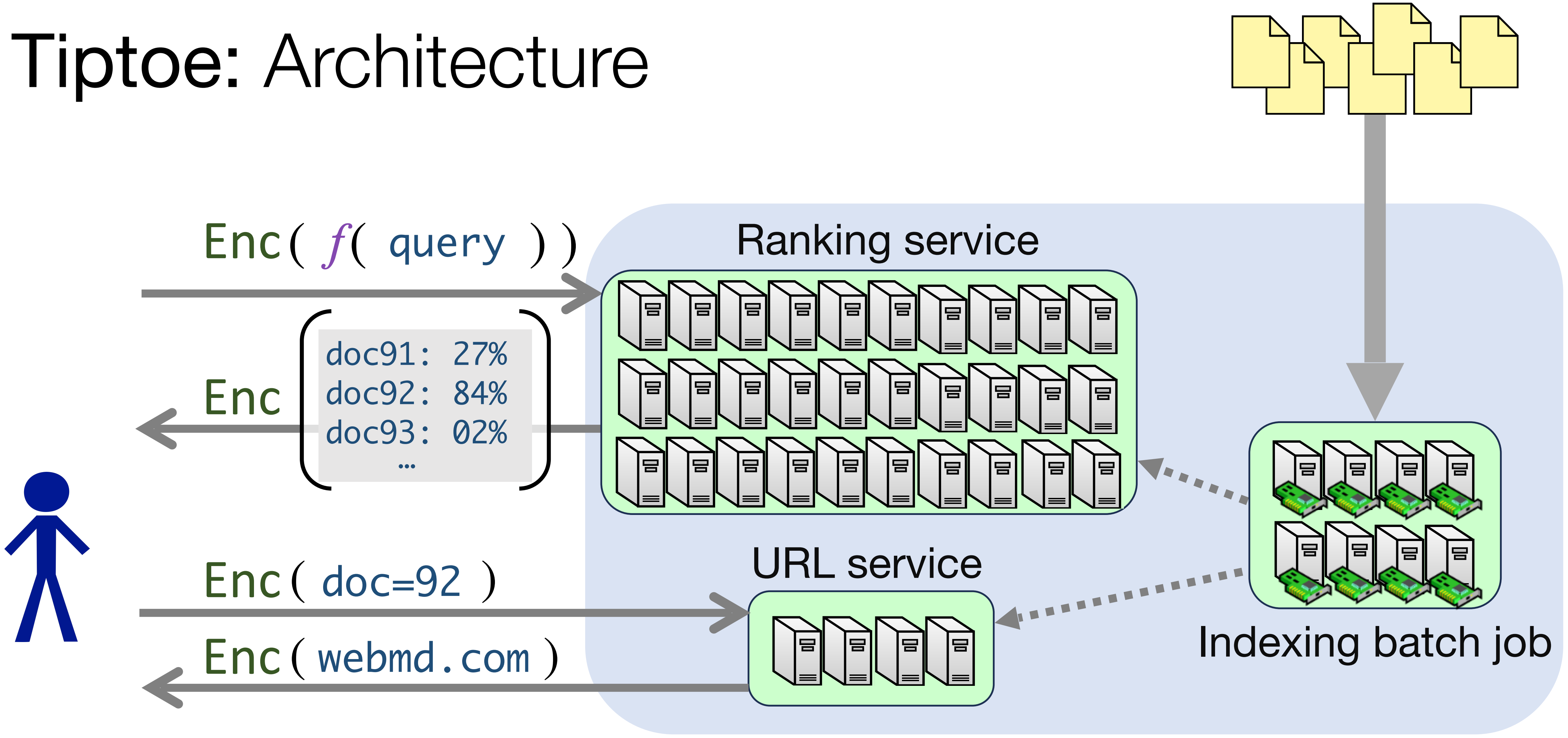
Private search on public data

- Private information retrieval [CGKS'95, KO'97, Splinter'17, …]
  only key-value lookups

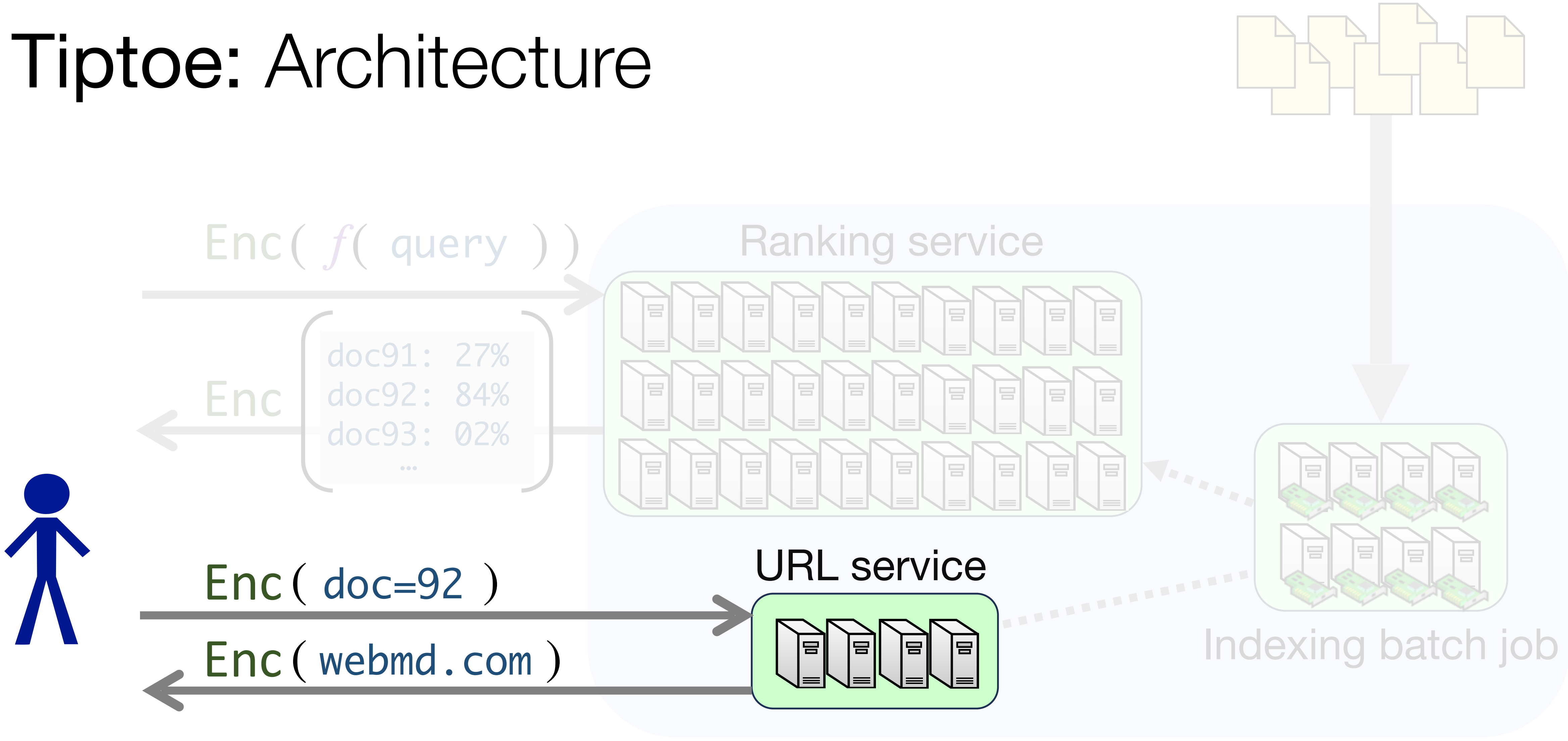
- Google over Tor [DMS'04]
  leaks query contents

→ Query-private search:
Tiptoe, Coeus [ASAEG'21]
expressive queries, hides query contents

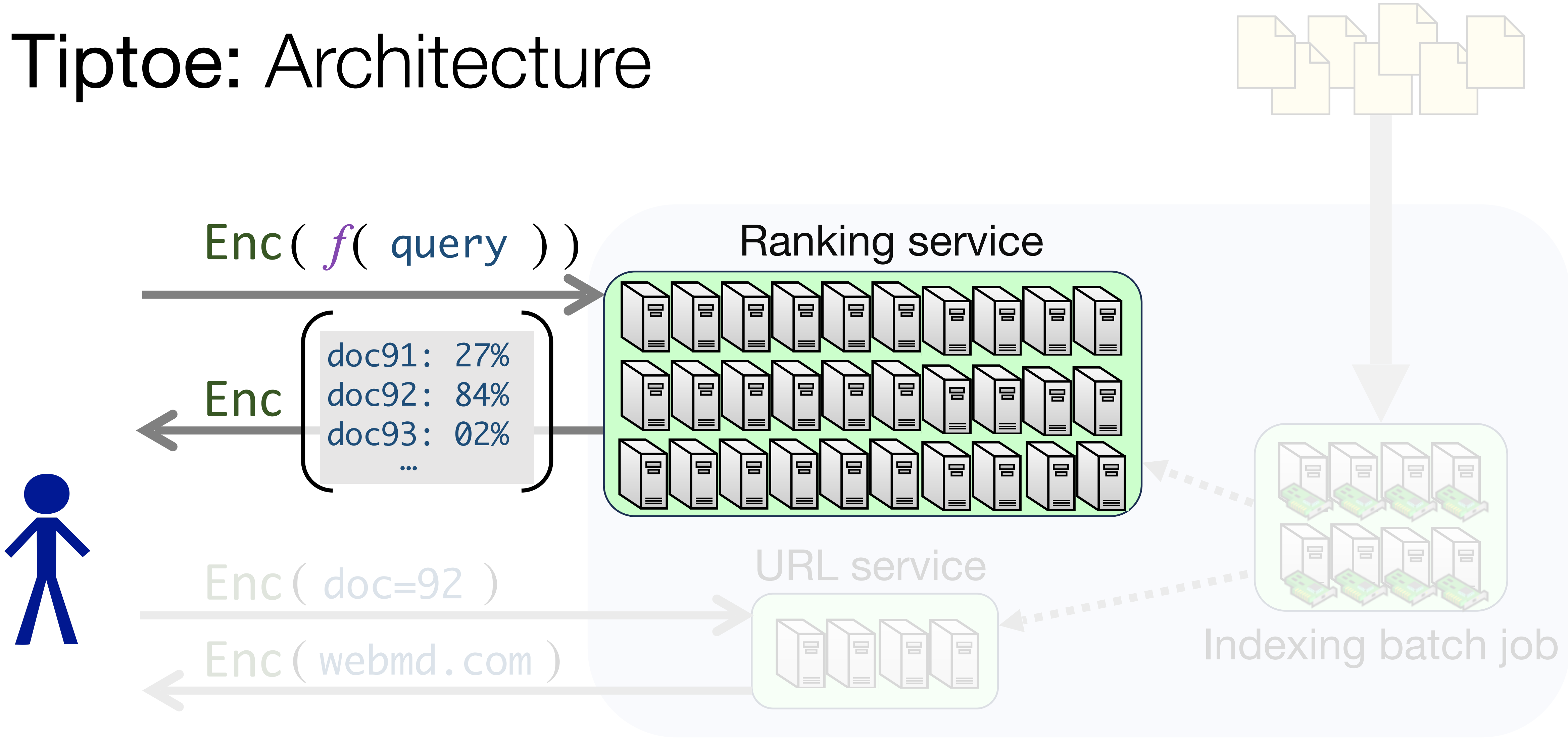
Tiptoe: Architecture



Tiptoe: Architecture



Tiptoe: Architecture



Tiptoe: Design steps

1. **Standard technique:** Reduce text search to nearest-neighbor search

   Key tool: Semantic embeddings [Osgood'57, …]

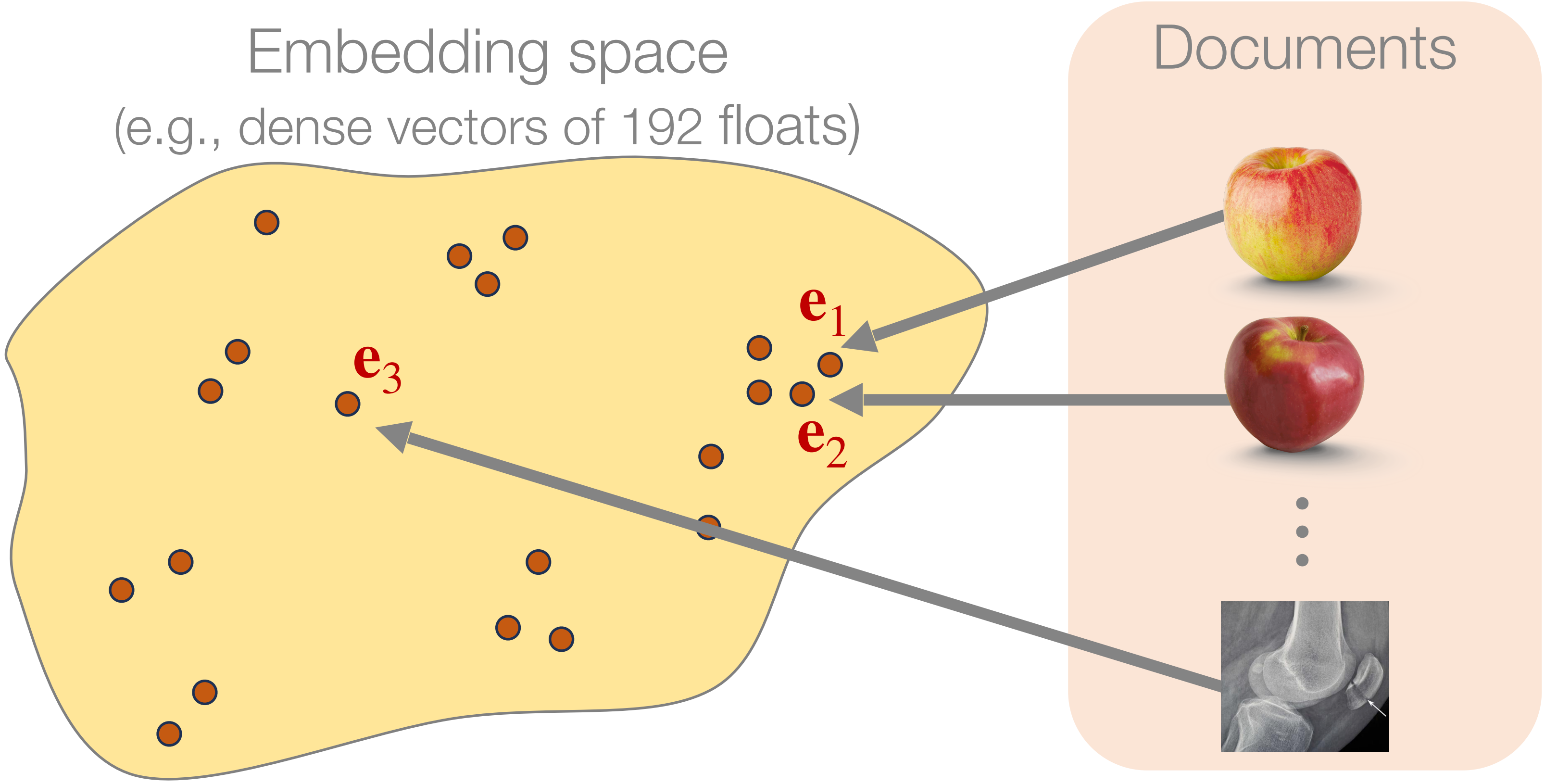
2. **Our contribution:** Fast <u>private</u> nearest-neighbor search

   Key tools: Clustering to reduce communication

   \+ Linearly homomorphic encryption with preprocessing to shrink the computation [SimplePIR'23]

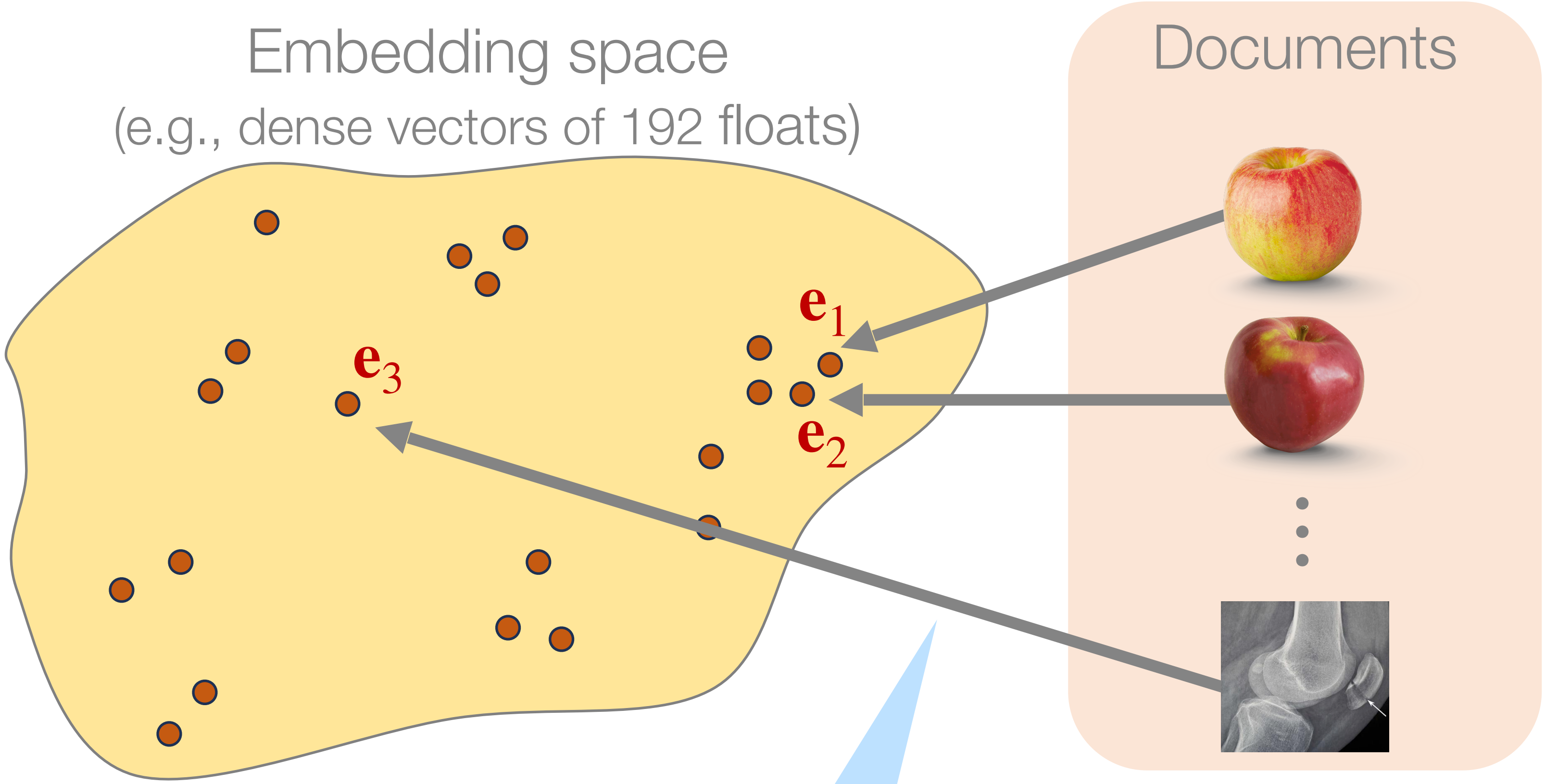
Represent documents and queries using semantic embeddings

[DC’19, MYCG’19, YYZL’19, SKPZ’22, …]



Represent documents and queries using semantic embeddings

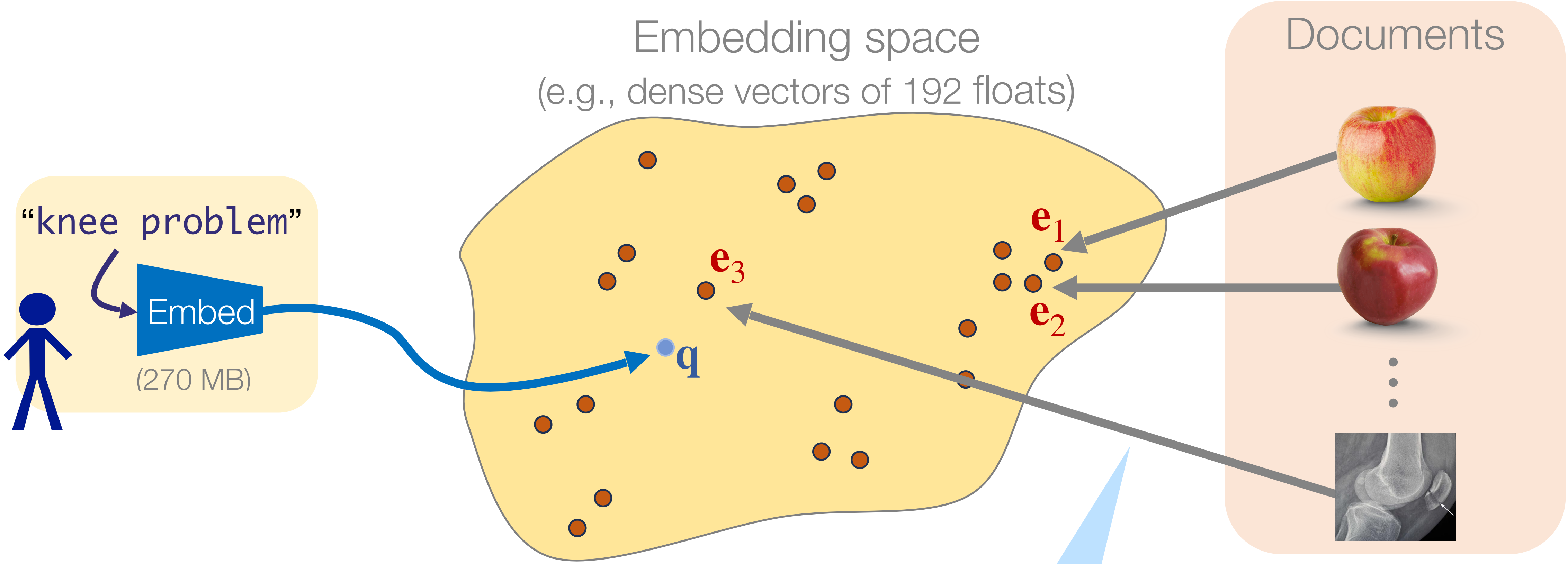
[DC'19, MYCG'19, YYZL'19, SKPZ'22, …]



Required property: when doc 1 and doc 2 are "similar" in meaning, their embedding inner-product score $\langle \mathbf{e}_1, \mathbf{e}_2 \rangle$ is large.

Represent documents and queries using semantic embeddings

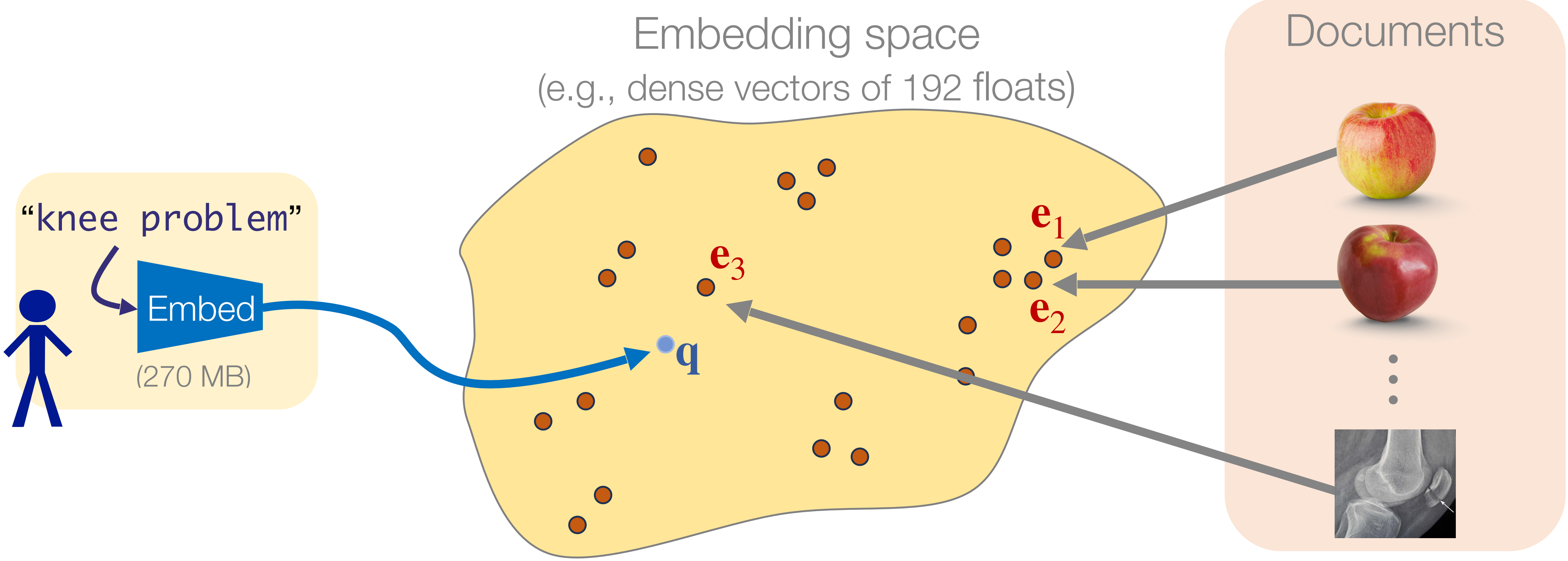
[DC'19, MYCG'19, YYZL'19, SKPZ'22, …]



Required property: when doc 1 and doc 2 are "similar" in meaning, their embedding inner-product score $\langle \mathbf{e_1}, \mathbf{e_2} \rangle$ is large.

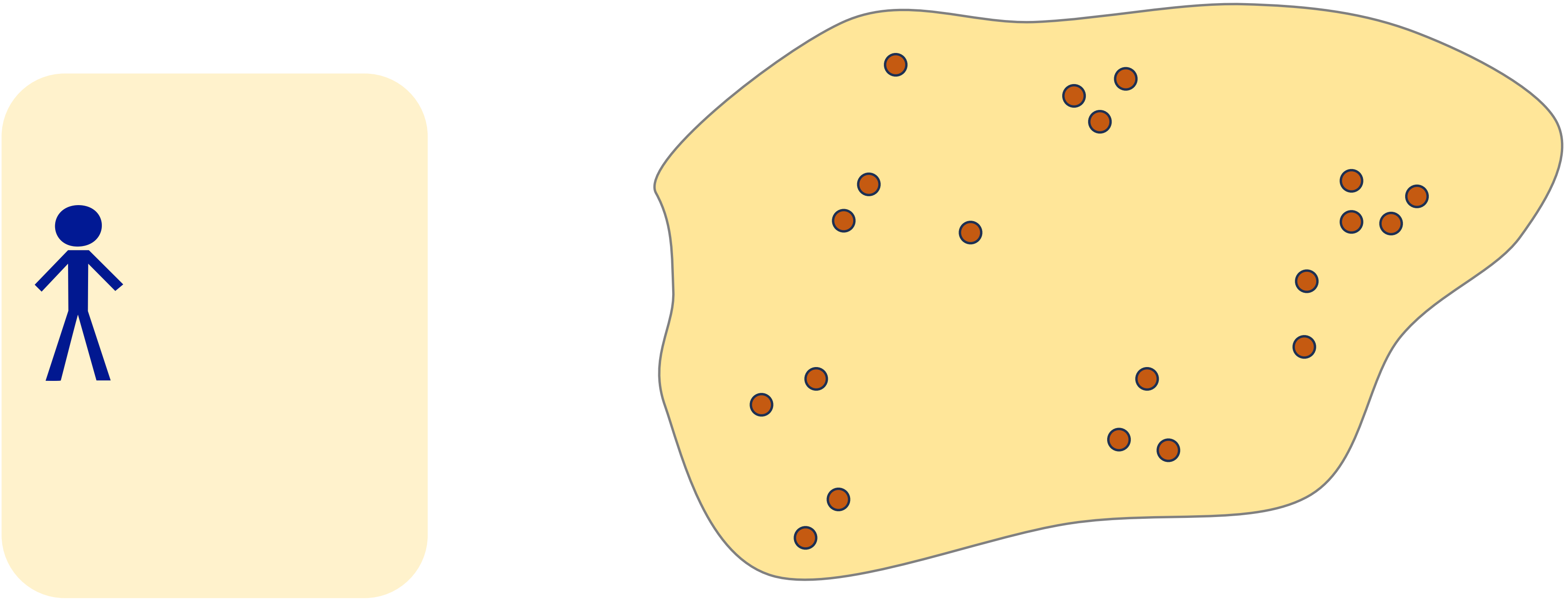
Represent documents and queries using semantic embeddings

[DC'19, MYCG'19, YYZL'19, SKPZ'22, ...]



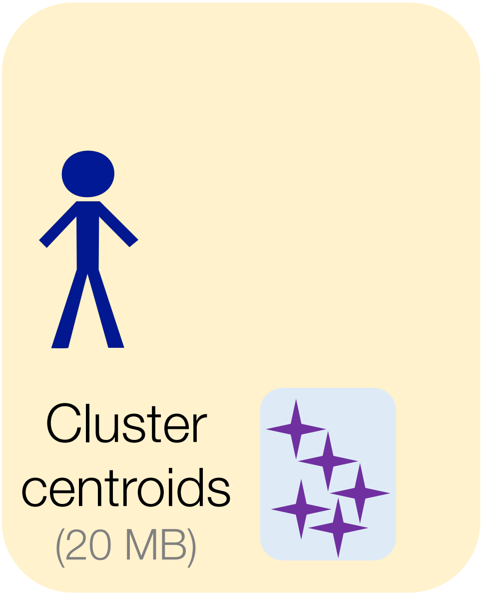
↳ **Goal:** privately find the doc that maximizes the score $\langle \mathbf{q}, \mathbf{e} \rangle$

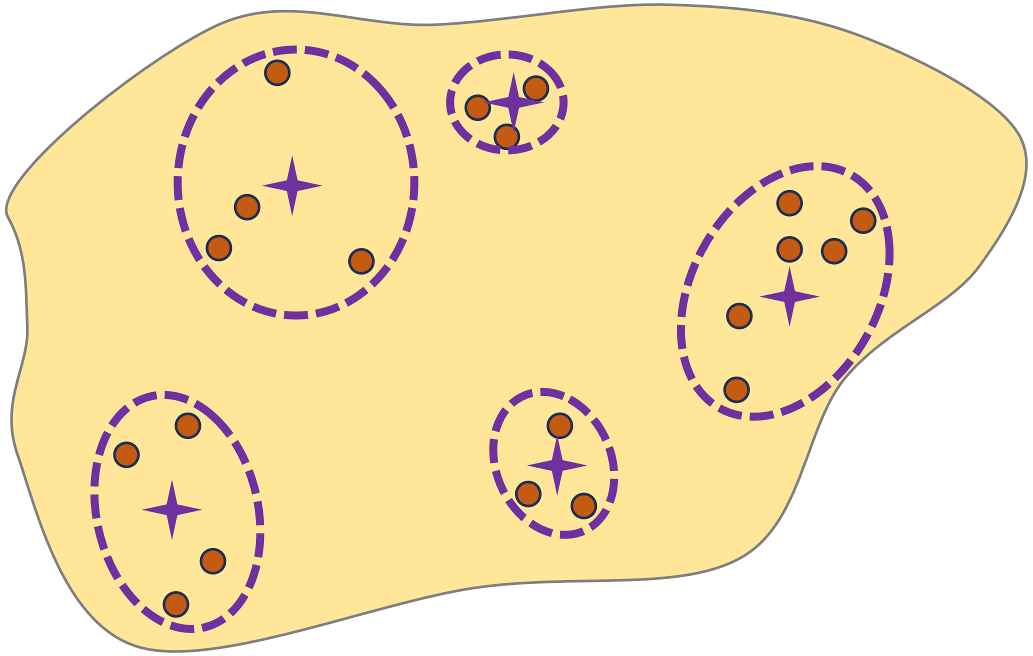
Perform coarse nearest-neighbor search locally on the client



Perform coarse nearest-neighbor search locally on the client

Ahead of time: Server groups the N docs into $\sqrt{N}$ clusters

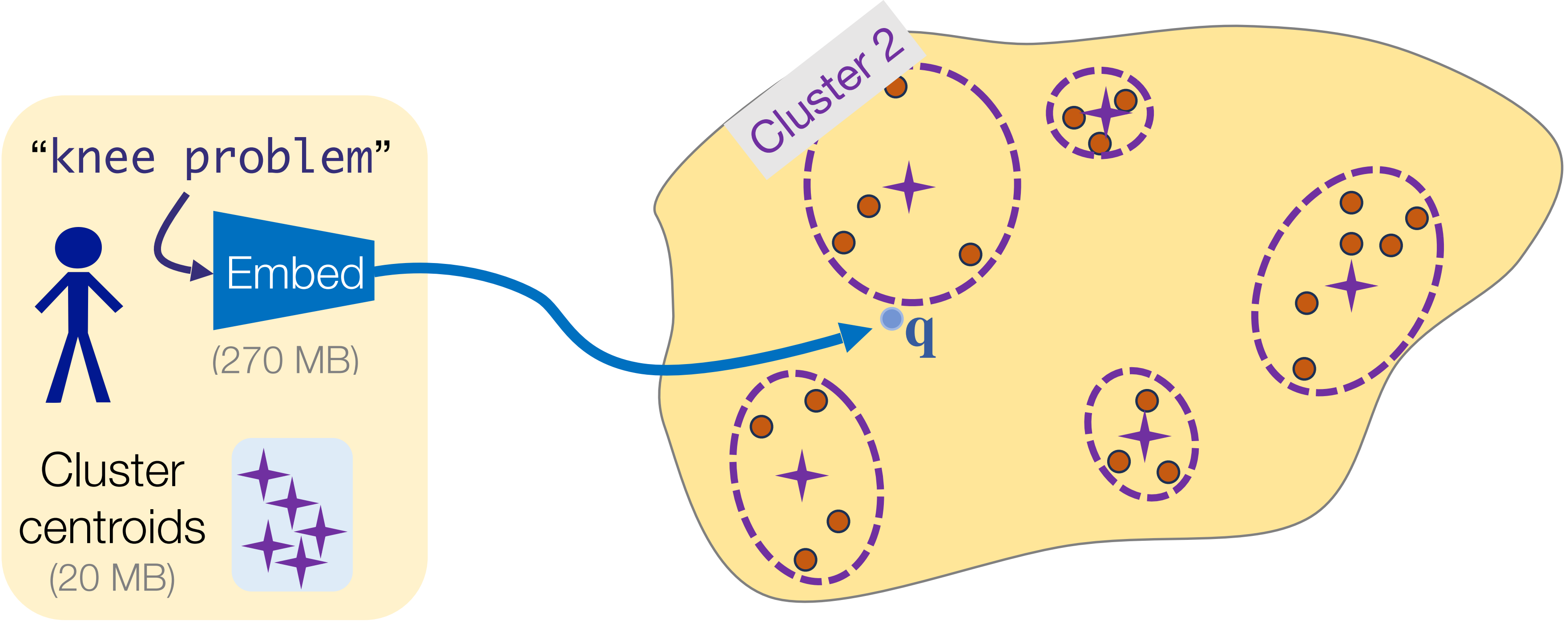




Perform coarse nearest-neighbor search locally on the client

Ahead of time: Server groups the N docs into $\sqrt{N}$ clusters

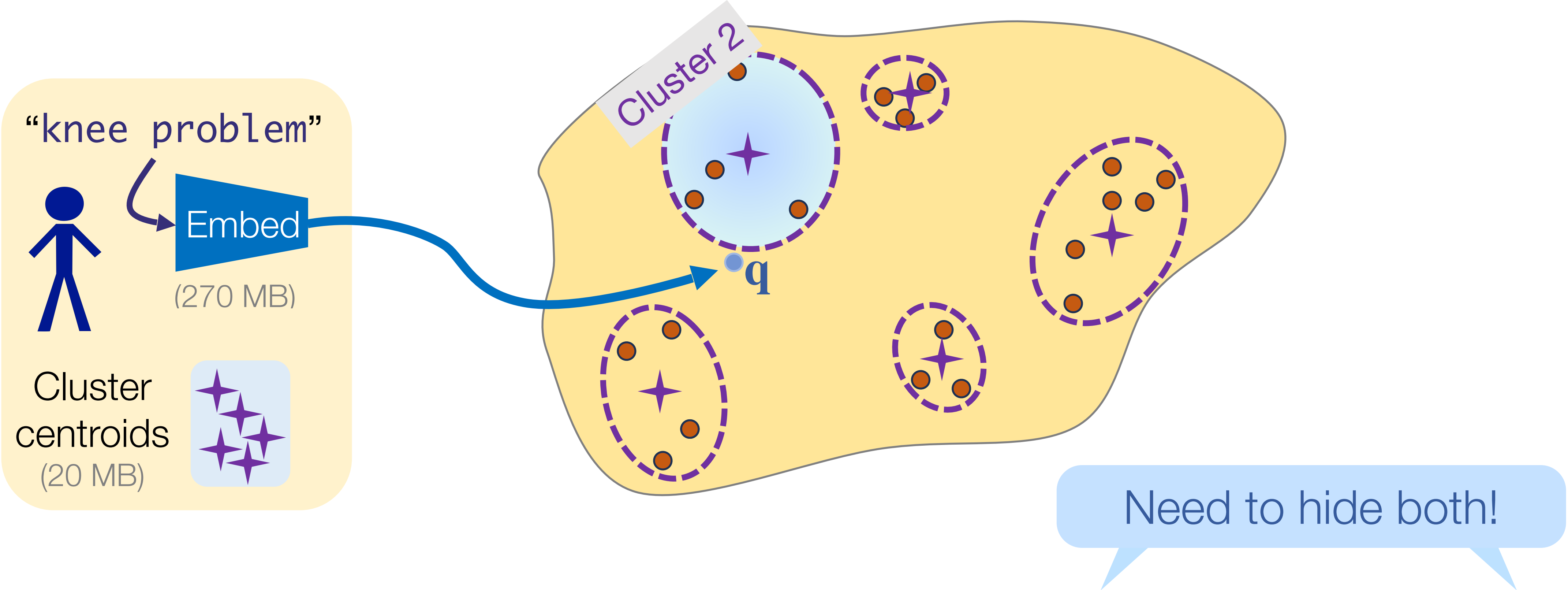
At query time: Client uses local list of centroids to find the closest cluster



Perform coarse nearest-neighbor search locally on the client

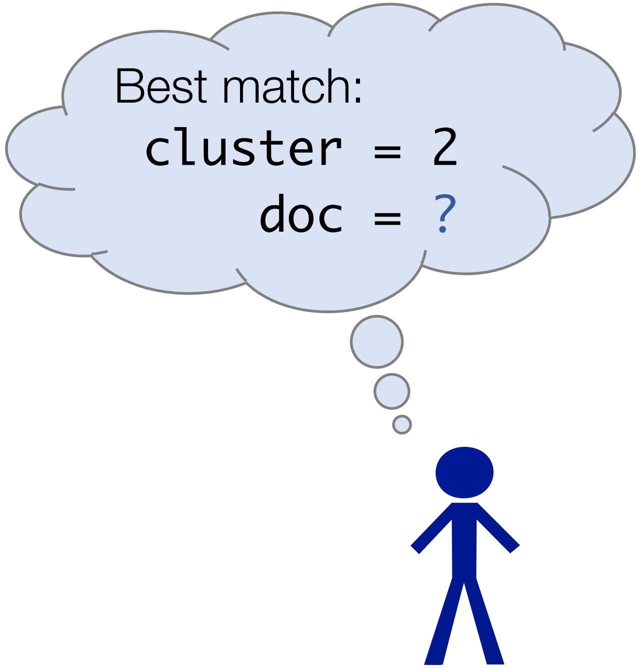
Ahead of time: Server groups the N docs into $\sqrt{N}$ clusters

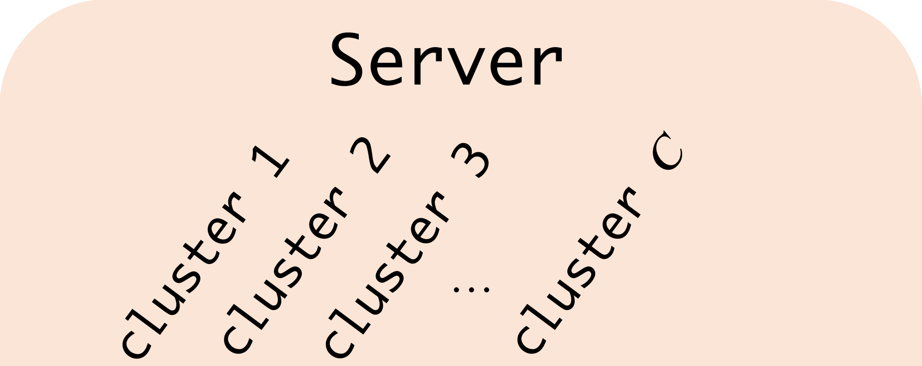
At query time: Client uses local list of centroids to find the closest cluster



↳ **Goal:** privately fetch inner-product scores $\langle \mathbf{q}, \mathbf{e} \rangle$ for docs in Cluster 2

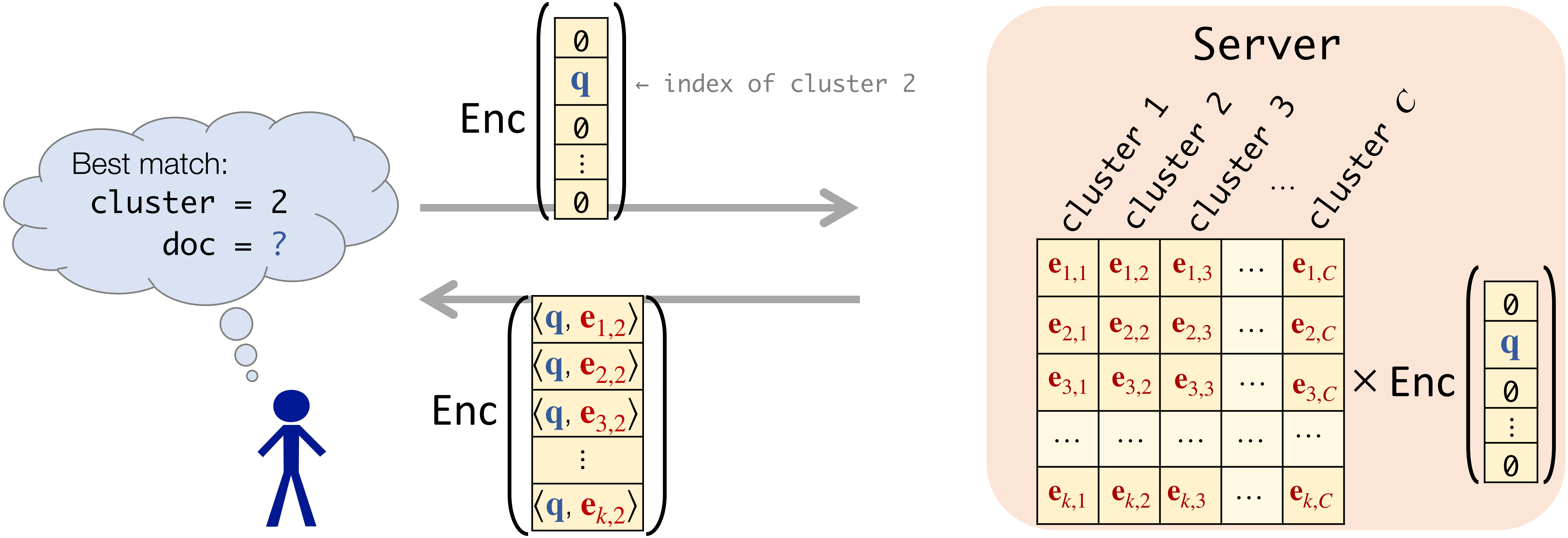
Perform exact search of the closest cluster under encryption



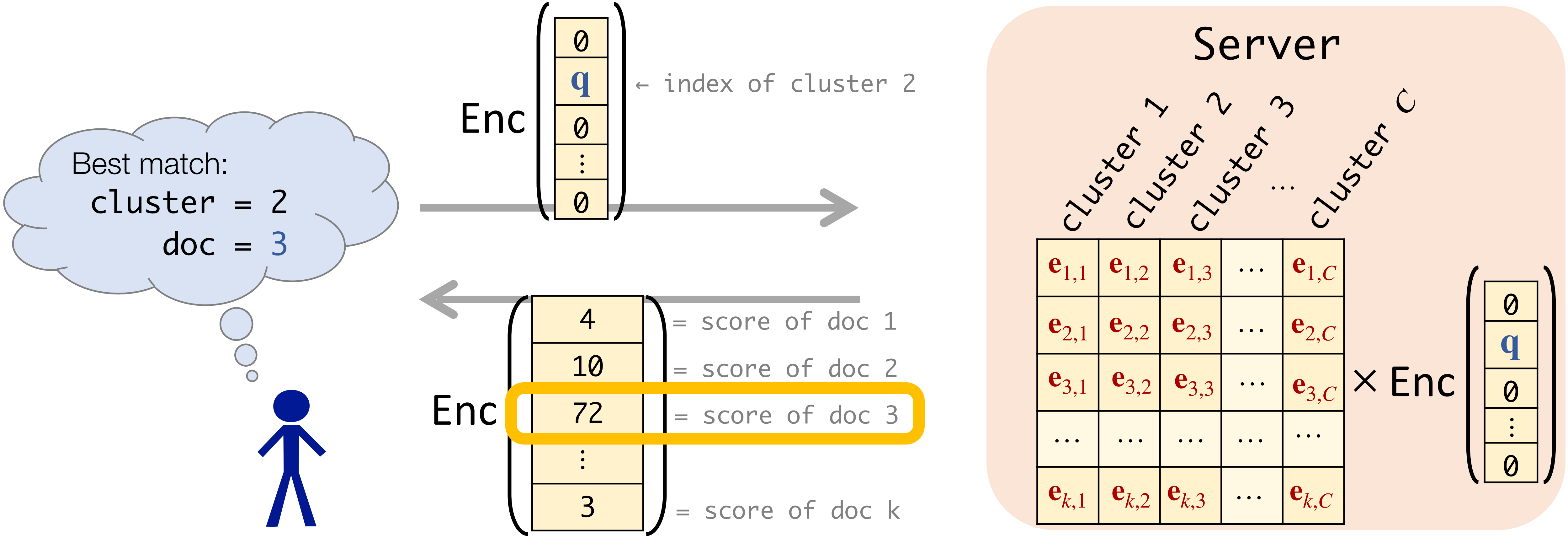


$\mathbf{e}_{1,1}$	$\mathbf{e}_{1,2}$	$\mathbf{e}_{1,3}$	…	$\mathbf{e}_{1,C}$
$\mathbf{e}_{2,1}$	$\mathbf{e}_{2,2}$	$\mathbf{e}_{2,3}$	…	$\mathbf{e}_{2,C}$
$\mathbf{e}_{3,1}$	$\mathbf{e}_{3,2}$	$\mathbf{e}_{3,3}$	…	$\mathbf{e}_{3,C}$
…	…	…	…	…
$\mathbf{e}_{k,1}$	$\mathbf{e}_{k,2}$	$\mathbf{e}_{k,3}$	…	$\mathbf{e}_{k,C}$

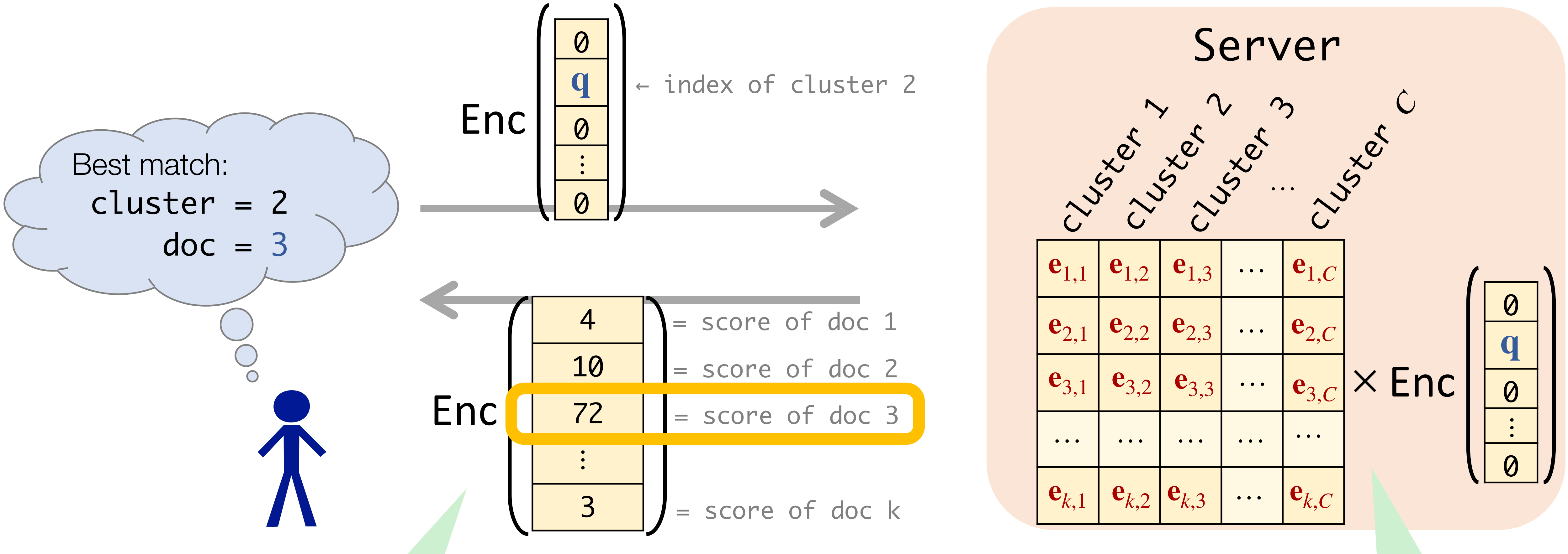
Perform exact search of the closest cluster under encryption



Perform exact search of the closest cluster under encryption



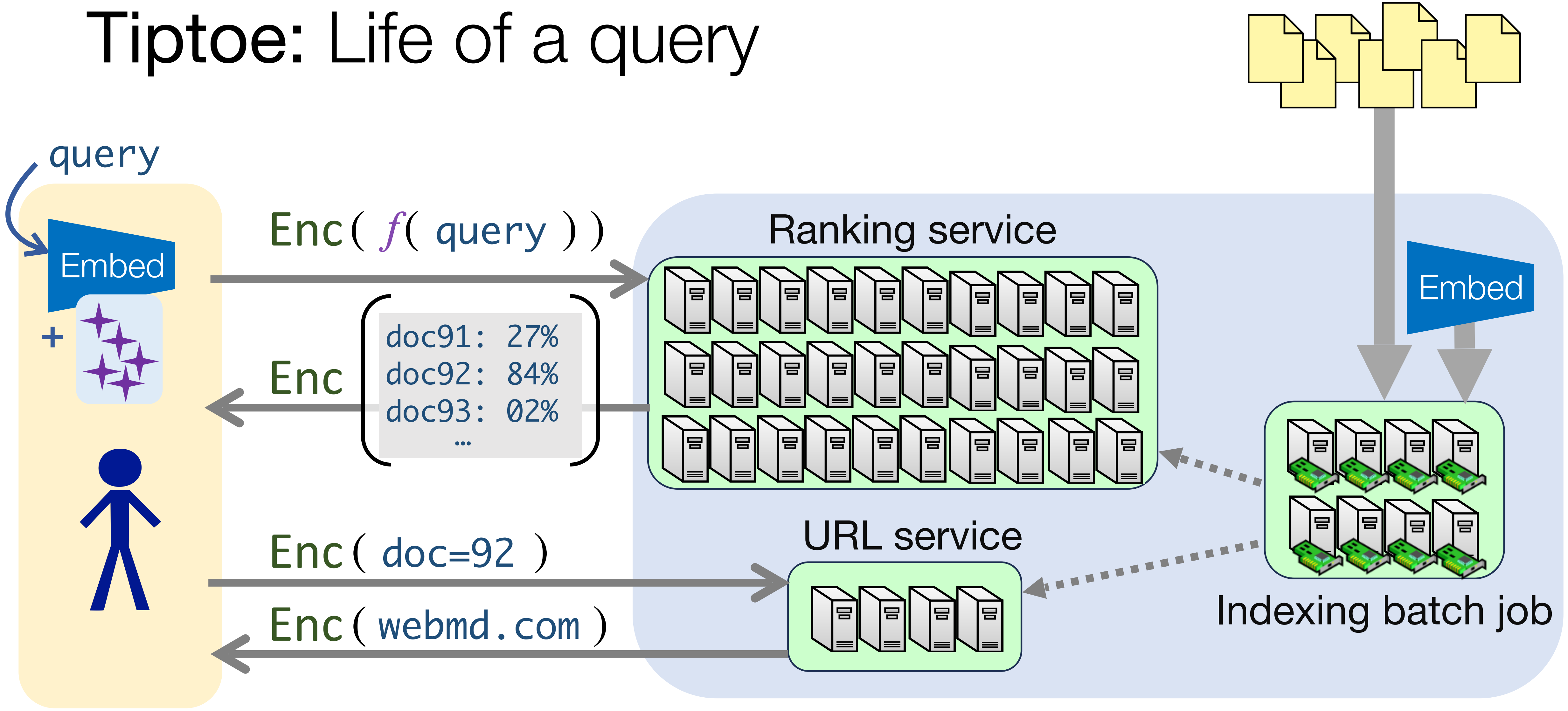
Perform exact search of the closest cluster under encryption



Communication: $O(\sqrt{Nd})$,
on N docs and embedding length d

Server work: fast with **SimplePIR**
($2d$ 64-bit operations per doc)

Tiptoe: Life of a query



Text embeddings: `msmarco-distilbert-base-tas-b`

Tiptoe is cheaper than state-of-the-art private search

	Coeus (SOSP'21)	Tiptoe	Gain
Docs searched	5 million	364 million	72 ×
Client storage	-	0.3 GiB	– ∞ ×
Server compute (per million docs)	2,580 core-s	0.4 core-s	6,450 ×
Communication (per million docs)	10 MiB	0.16 MiB	62 ×
End-to-end latency	-	2.7 s	

Tiptoe is cheaper than state-of-the-art private search

	Coeus (SOSP'21)	Tiptoe	Gain
Docs searched	5 million	364 million	72 ×
Client storage	-		$-\infty$ ×
Server compute (per million docs)	2,580 core-s	0.4 core-s	6,450 ×
Communication (per million docs)			62 ×
End-to-end latency	-	2.7 s	

Semantic embeddings: $100\times$ smaller doc representations

SimplePIR: $10\times$ less computation

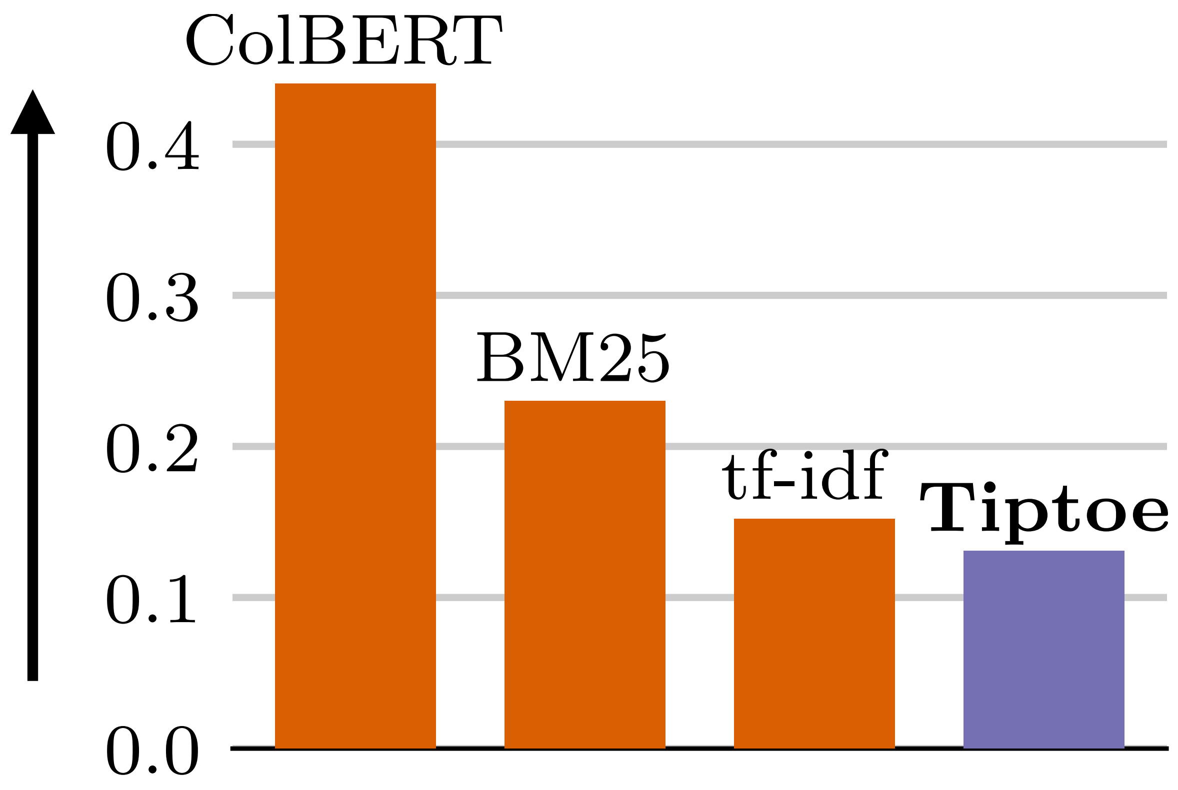
Clustering: communication sublinear in N

Tiptoe is cheaper than state-of-the-art private search

	Coeus (SOSP'21)	Tiptoe	Gain
Docs searched	5 million	364 million	72 ×
Client storage	-	0.3 GiB	– ∞ ×
Server compute (per million docs)	2,580 core-s	0.4 core-s	6,450 ×
Communication (per million docs)	10 MiB	0.16 MiB	62 ×
End-to-end latency	-	2.7 s	

Tiptoe's search quality is acceptable

Better search quality (MRR@100)



Best non-private:
Top result on average ranked 2.3 of 100

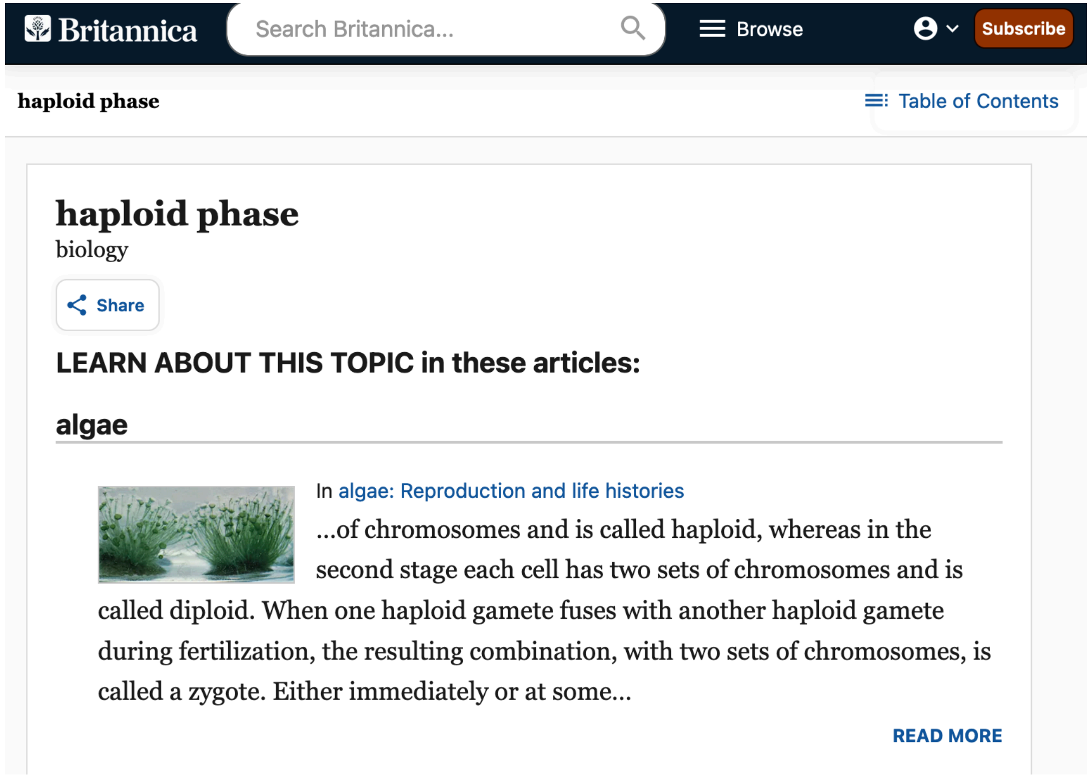
Private:
Top result on average ranked 7.7 of 100

On the MS MARCO doc-rank “dev” data set

Examples: Tiptoe works best on conceptual queries

`how long before eagles get feathers`

`the meaning of haploid cell`



… but Tiptoe's exact-string search could improve

`77 Massachusetts Avenue`

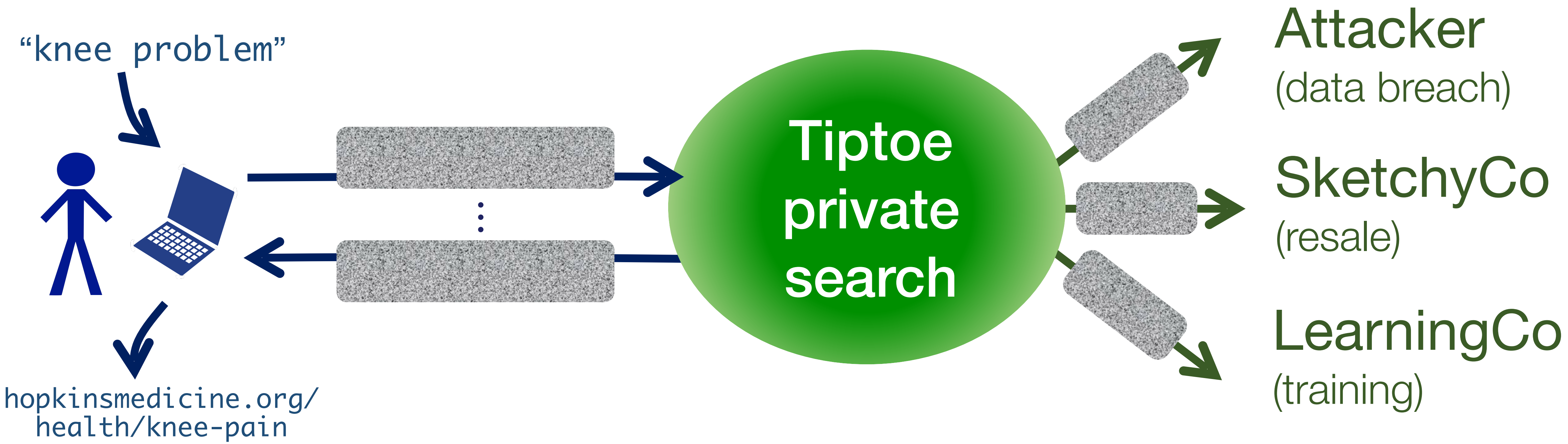
Private search is within reach… what's next?

Many directions for improvement

- Improve quality: run more powerful search under encryption?
- Reduce cost: shrink communication? increase throughput?

Many applications of private nearest-neighbor search

- Tiptoe can search over products, ads, feeds, and more



Paper

Code

Alexandra Henzinger

Code: github.com/ahenzinger/tiptoe

Paper: eprint.iacr.org/2023/1438

Demo: come talk to me!