

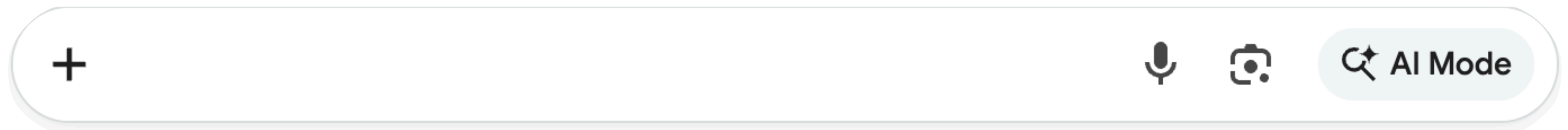
Cryptography at Scale: Designing a Private Web Search Engine

Alexandra Henzinger

Committee: Henry Corrigan-Gibbs, Yael Kalai, Vinod Vaikuntanathan, and Nickolai Zeldovich

We trust web servers with our secrets

Google



We trust web servers with our secrets



+

how to file|

×

🔍 AI Mode

🔍 how to file **a restraining order**

🔍 how to file **for unemployment**

🔍 how to file **for divorce**

Google Search

I'm Feeling Lucky

We trust web servers with our secrets



+ why did my |



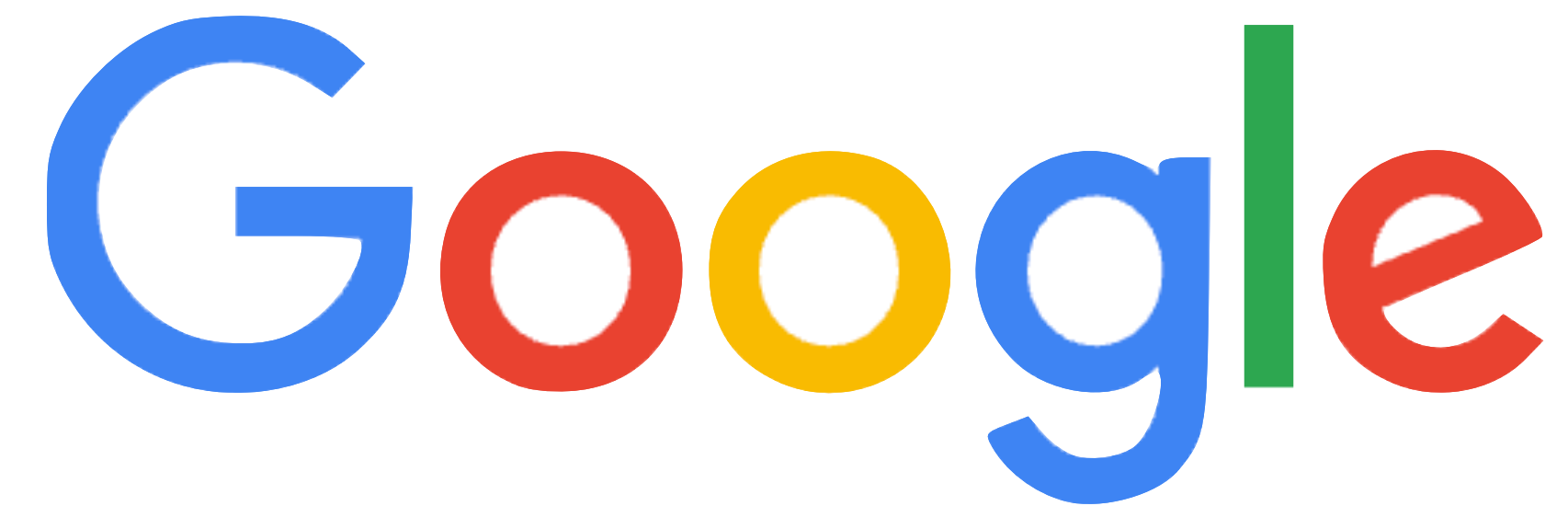
AI Mode

- 🔍 why did my **credit score drop**
- 🔍 why did my **period come early**
- 🔍 why did my **insurance go up**

Google Search

I'm Feeling Lucky

We trust web servers with our secrets



+ how do i know if |



AI Mode

how do i know if **i have diabetes**

how do i know if **i have adhd**

how do i know if **i have a concussion**

Google Search

I'm Feeling Lucky

Web servers **leak** our secrets

Breach

**Unsecured Microsoft Bing Server Exposed Users'
Search Queries and Location**

The Hacker News, Sep 2020

Web servers **leak** our secrets

Breach

Unsecured Microsoft Bing Server Exposed Users' Search Queries and Location

The Hacker News, Sep 2020

Reuse

Last chance to stop Instagram training Meta's AI with your data

Yahoo!Tech, May 2025

Web servers **leak** our secrets

Breach

Unsecured Microsoft Bing Server Exposed Users' Search Queries and Location

The Hacker News, Sep 2020

Reuse

Last chance to stop Instagram training Meta's AI with your data

Yahoo!Tech, May 2025

Sale

Health Insurers Are Vacuuming Up Details About You — And It Could Raise Your Rates

NPR, July 2018

Dream: Web servers **never see** our secrets

Breach

Unsecured Microsoft Bing Server Exposed Users' Search Queries and Location

The Hacker News, Sep 2020

Reuse

Last chance to stop Instagram training Meta's AI with your data

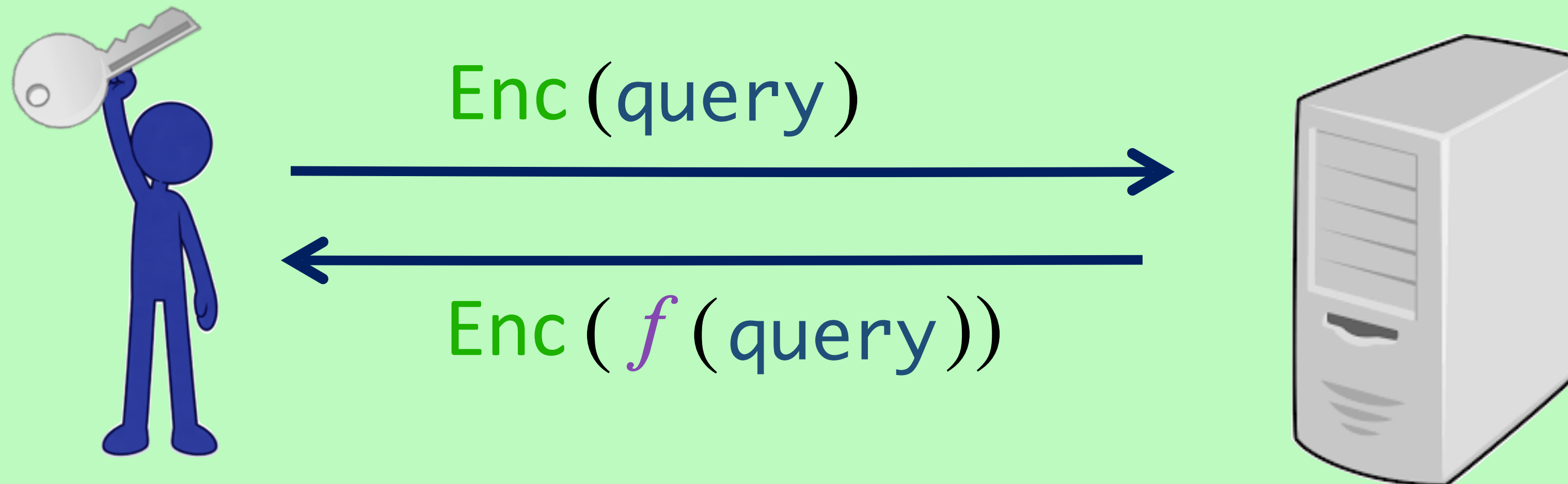
Yahoo!Tech, May 2025

Sale

Health Insurers Are Vacuuming Up Details About You — And It Could Raise Your Rates

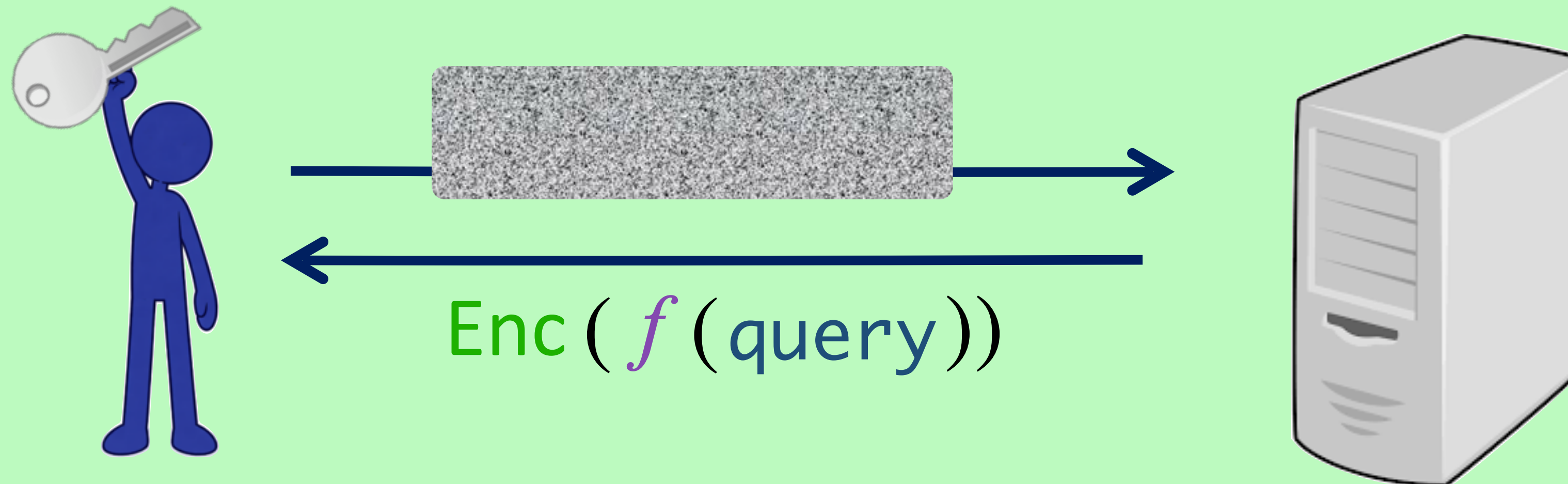
NPR, July 2018

Dream: Web servers **never see** our secrets



Since 2009: we have the algorithms to compute on encrypted data
... why aren't we using them?

Dream: Web servers **never see** our secrets



Since 2009: we have the algorithms to compute on encrypted data
... why aren't we using them?

Dream: Web servers **never see** our secrets

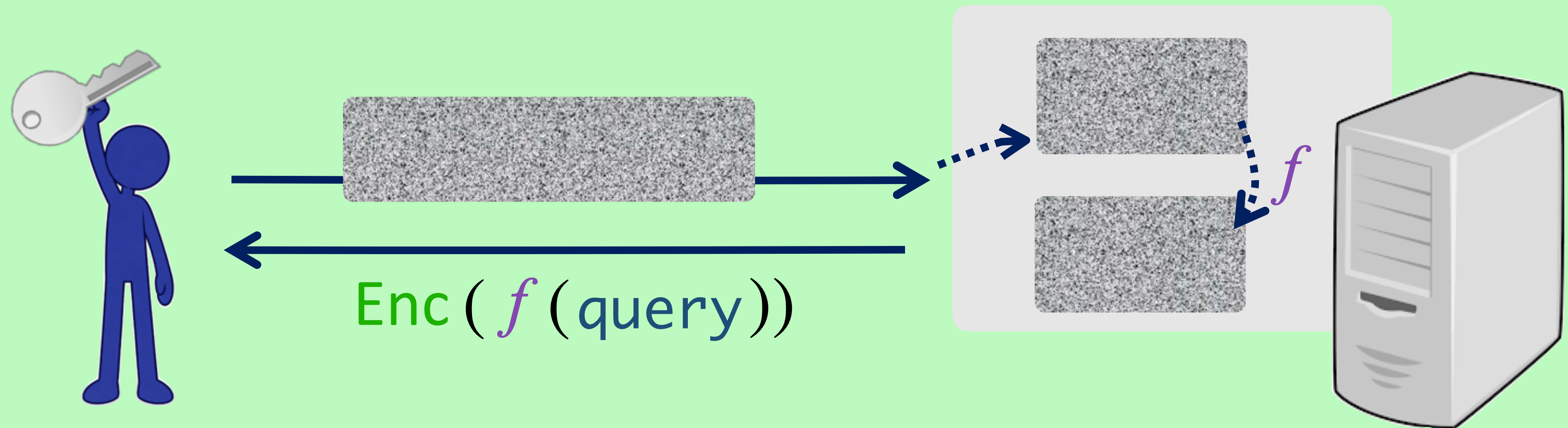


$\text{Enc}(f(\text{query}))$



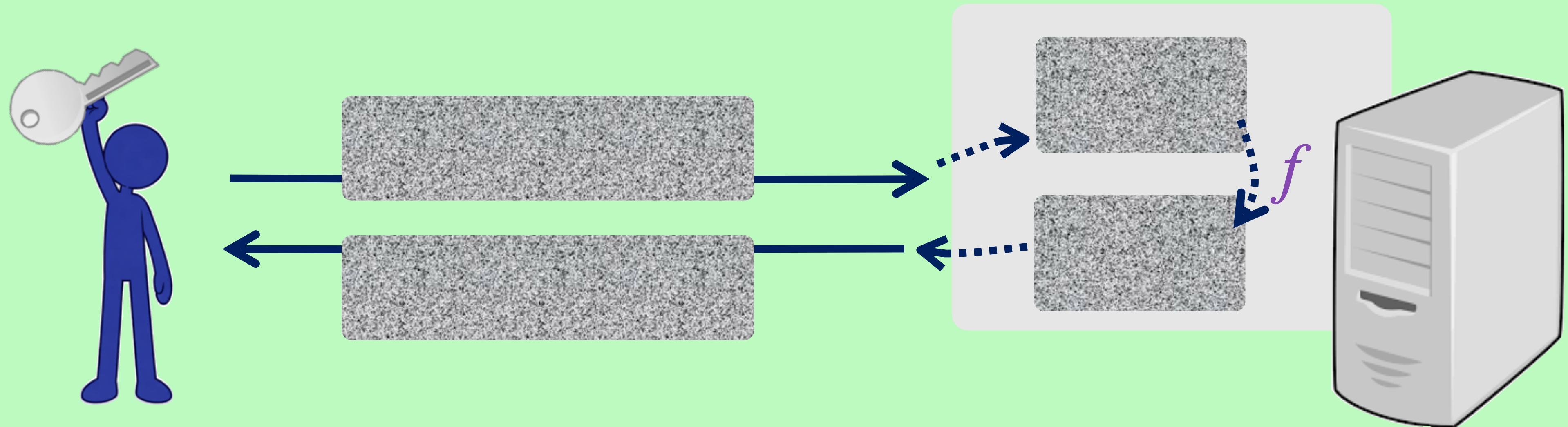
Since 2009: we have the algorithms to compute on encrypted data
... why aren't we using them?

Dream: Web servers **never see** our secrets



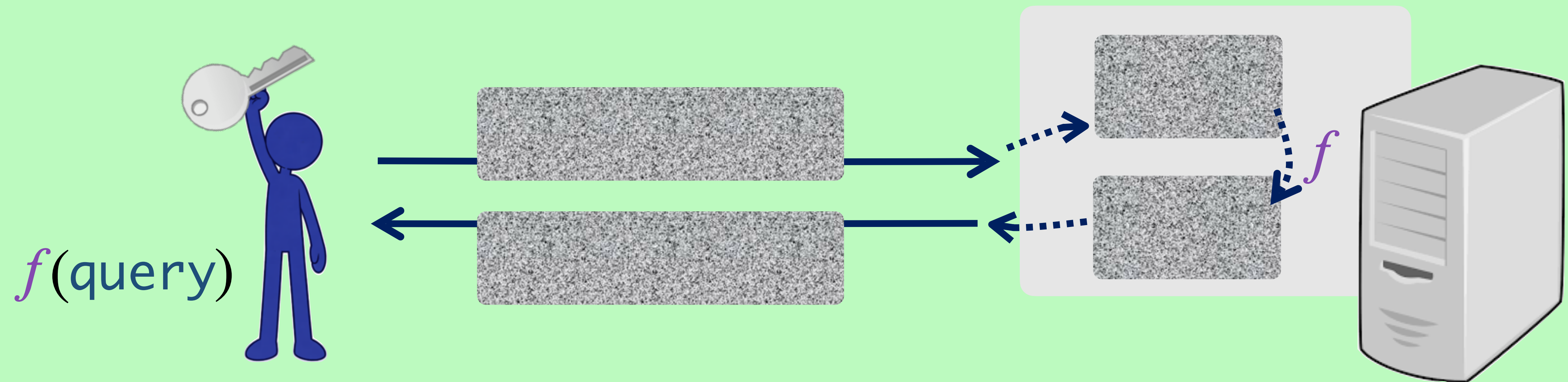
Since 2009: we have the algorithms to compute on encrypted data
... why aren't we using them?

Dream: Web servers **never see** our secrets



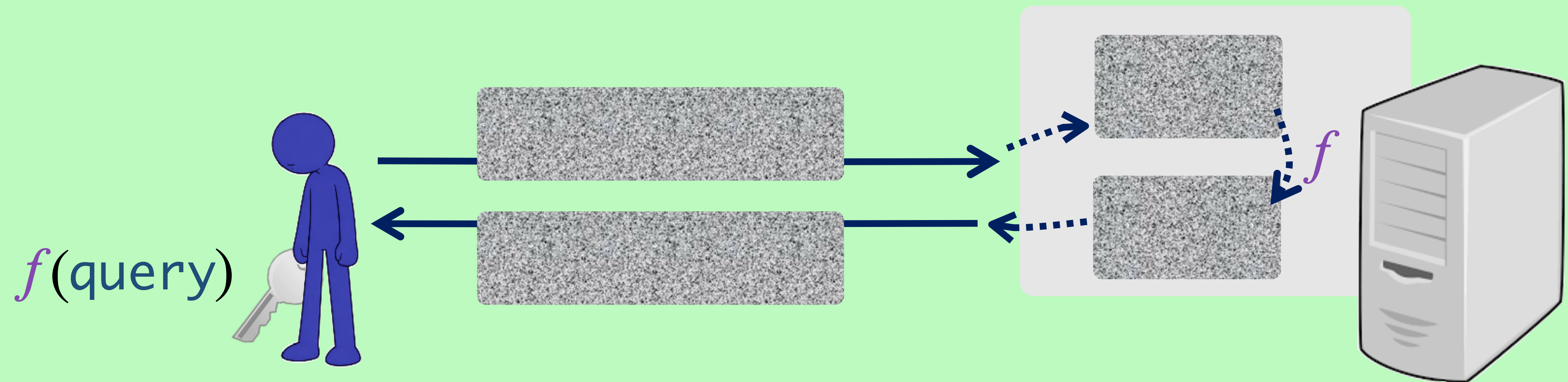
Since 2009: we have the algorithms to compute on encrypted data
... why aren't we using them?

Dream: Web servers **never see** our secrets



Since 2009: we have the algorithms to compute on encrypted data
... why aren't we using them?

Dream: Web servers **never see** our secrets



Since 2009: we have the algorithms to compute on encrypted data
... why aren't we using them?

The **cryptography** we have:

e.g., add and multiply on encrypted data

$$\text{Enc}(x) + \text{Enc}(y) = \text{Enc}(x + y)$$

$$\text{Enc}(x) \times \text{Enc}(y) = \text{Enc}(x \times y)$$

“Fully homomorphic encryption”

[RAD78, Gen09, BV11]

The **cryptography** we have:

e.g., add and multiply on encrypted data

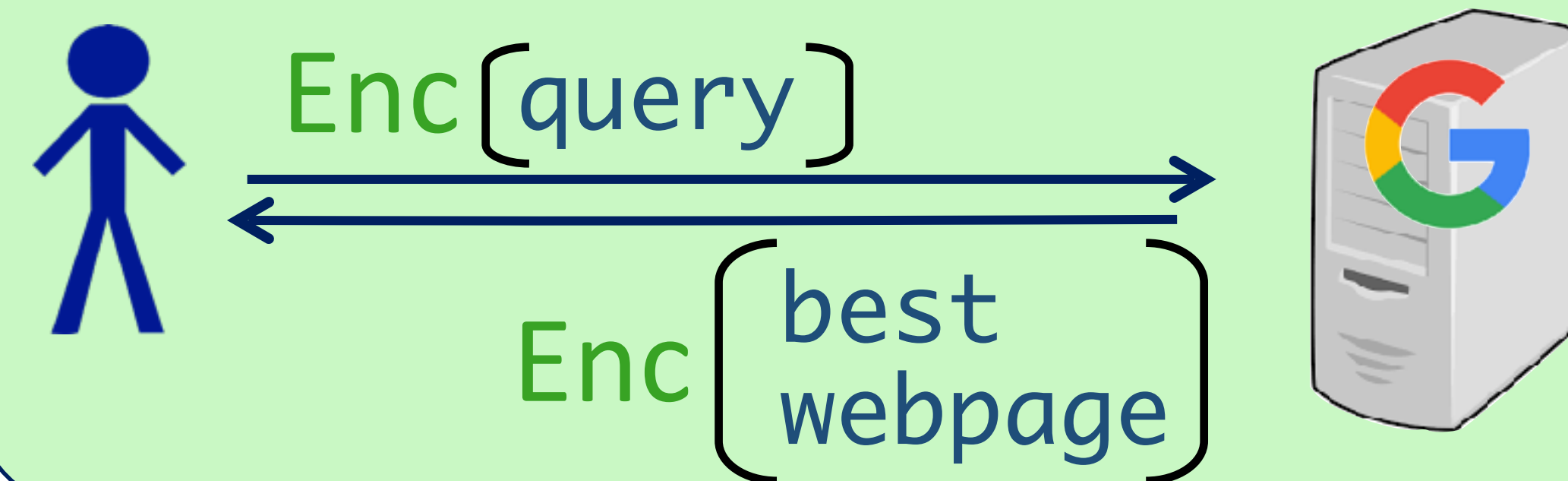
$$\text{Enc}(x) + \text{Enc}(y) = \text{Enc}(x + y)$$

$$\text{Enc}(x) \times \text{Enc}(y) = \text{Enc}(x \times y)$$

“Fully homomorphic encryption”
[RAD78, Gen09, BV11]

The **systems** we want:

e.g., web search on encrypted data



The **cryptography** we have:

e.g., add and multiply on encrypted data

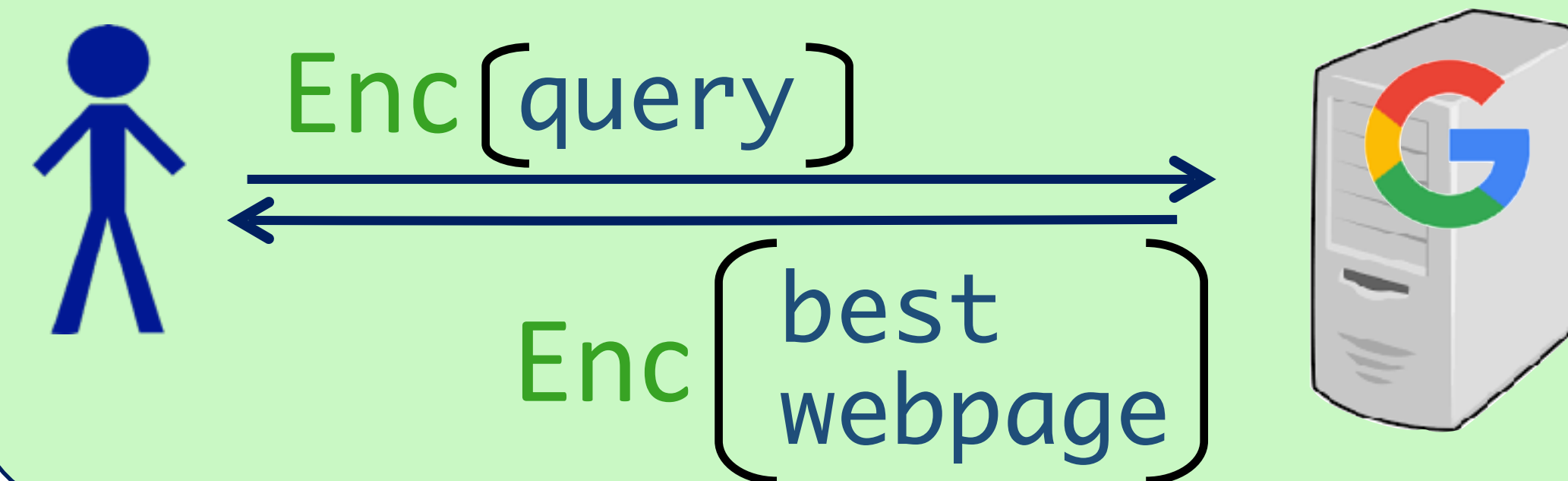
$$\text{Enc}(x) + \text{Enc}(y) = \text{Enc}(x + y)$$

$$\text{Enc}(x) \times \text{Enc}(y) = \text{Enc}(x \times y)$$

“Fully homomorphic encryption”
[RAD78, Gen09, BV11]

The **systems** we want:

e.g., web search on encrypted data



Wildly intractable:

Compile  down to $+$ and $\times$?



The **cryptography** we have:

e.g., add and multiply on encrypted data

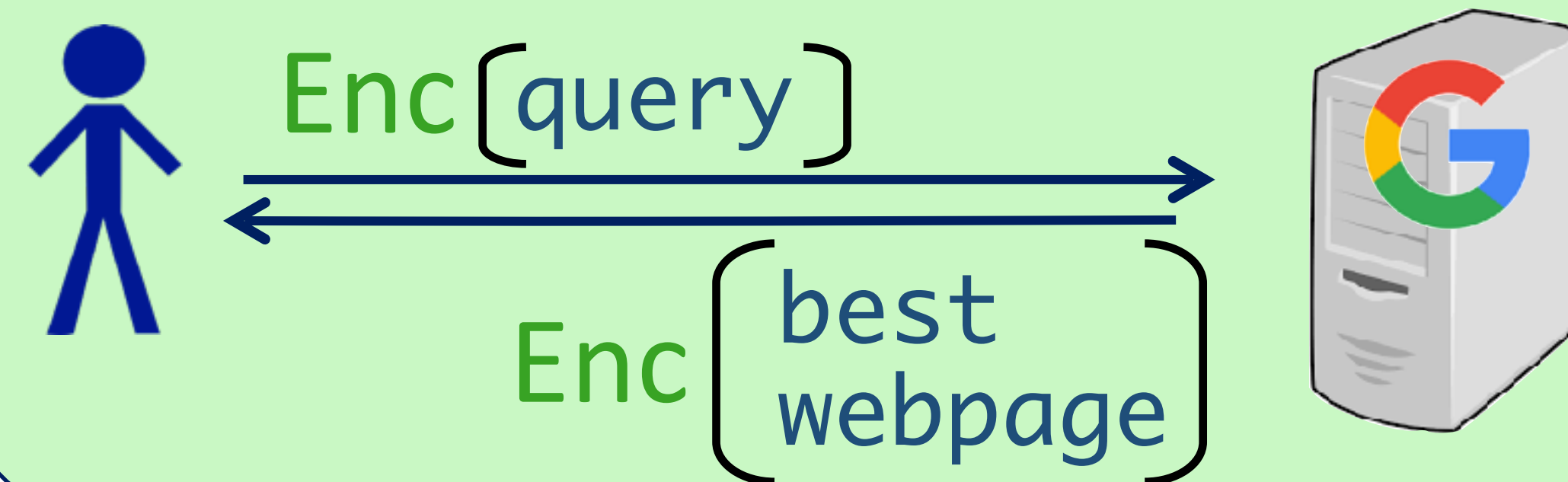
$$\text{Enc}(x) + \text{Enc}(y) = \text{Enc}(x + y)$$

$$\text{Enc}(x) \times \text{Enc}(y) = \text{Enc}(x \times y)$$

“Fully homomorphic encryption”
[RAD78, Gen09, BV11]

The **systems** we want:

e.g., web search on encrypted data



This Thesis: Build from both ends

The **cryptography** we have:

e.g., add and multiply on encrypted data

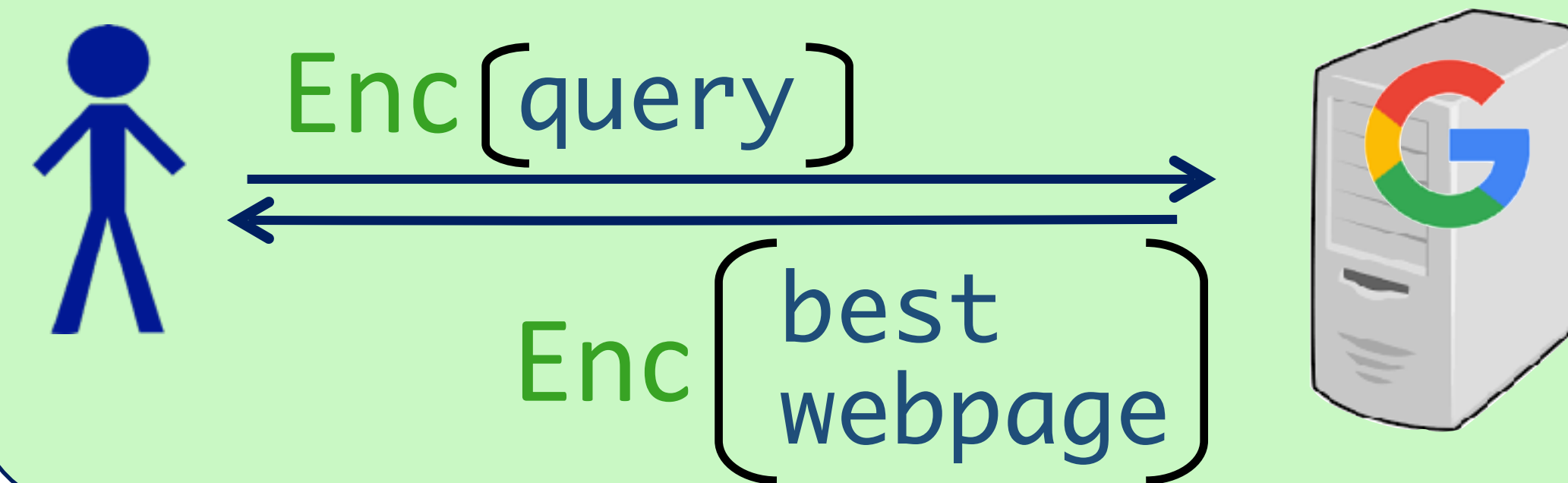
$$\text{Enc}(x) + \text{Enc}(y) = \text{Enc}(x + y)$$

$$\text{Enc}(x) \times \text{Enc}(y) = \text{Enc}(x \times y)$$

“Fully homomorphic encryption”
[RAD78, Gen09, BV11]

The **systems** we want:

e.g., web search on encrypted data



This Thesis: Build from both ends

The **cryptography** we have:

e.g., add and multiply on encrypted data

$$\text{Enc}(x) + \text{Enc}(y) = \text{Enc}(x + y)$$

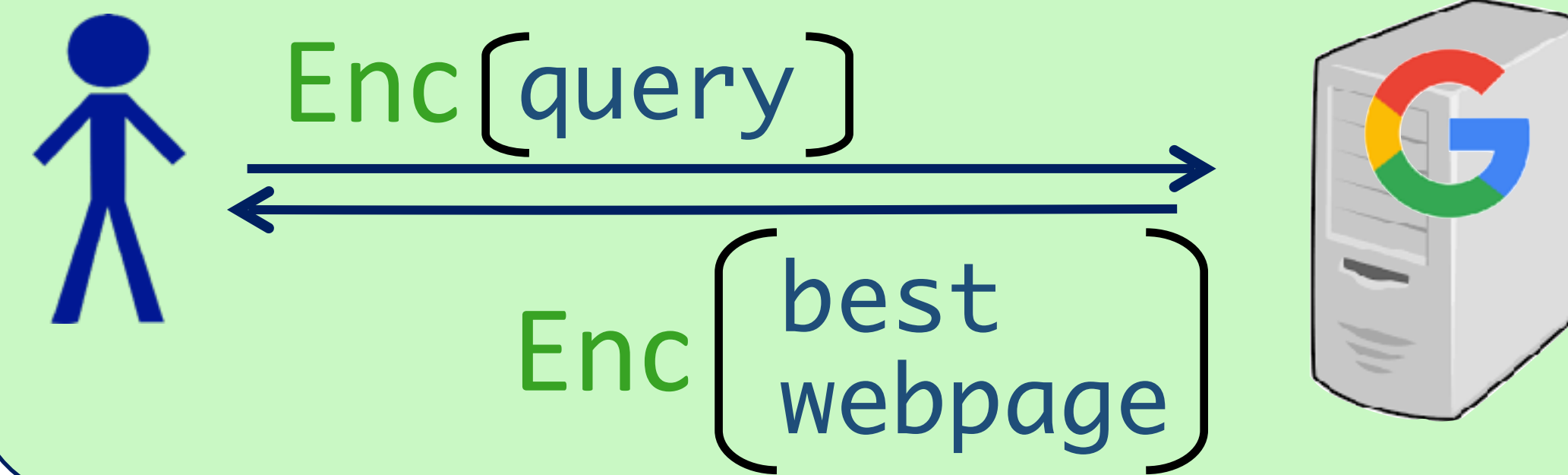
$$\text{Enc}(x) \times \text{Enc}(y) = \text{Enc}(x \times y)$$

“Fully homomorphic encryption”
[RAD78, Gen09, BV11]

Design fast
cryptography

The **systems** we want:

e.g., web search on encrypted data



This Thesis: Build from both ends

The **cryptography** we have:

e.g., add and multiply on encrypted data

$$\text{Enc}(x) + \text{Enc}(y) = \text{Enc}(x + y)$$

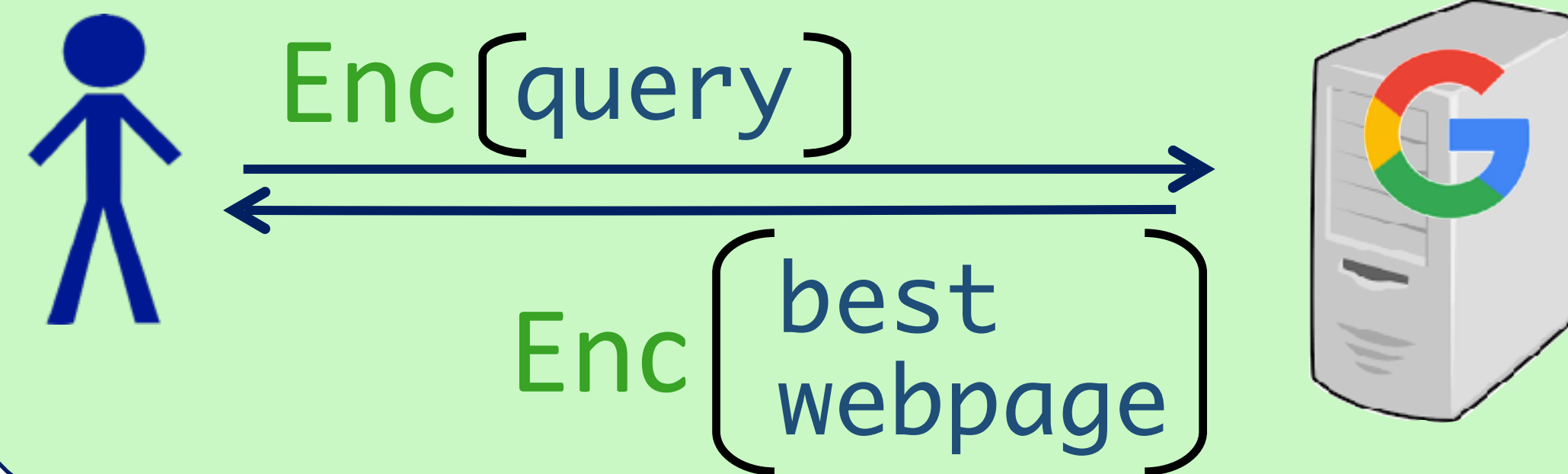
$$\text{Enc}(x) \times \text{Enc}(y) = \text{Enc}(x \times y)$$

“Fully homomorphic encryption”
[RAD78, Gen09, BV11]

Design fast
cryptography

The **systems** we want:

e.g., web search on encrypted data



Re-architect
systems

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems

- ▶ Protect high-impact applications

Libraries

Algorithms on encrypted data

- ▶ Reorganize computation and data in a way that scales

Cryptographic
operations

Fast cryptography

- ▶ Design efficient encrypted operations

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems

- ▶ Protect high-impact applications

Libraries

Algorithms on encrypted data

- ▶ Reorganize computation and data in a way that scales

Cryptographic
operations

Fast cryptography

- ▶ Design efficient encrypted operations

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems

- ▶ Protect high-impact applications

Libraries

Algorithms on encrypted data

- ▶ Reorganize computation and data in a way that scales

Cryptographic
operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- **Private matrix multiplication**
- **Private information retrieval**
- **Homomorphic encryption**

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems

- ▶ Protect high-impact applications

Libraries

Algorithms on encrypted data

- ▶ Reorganize computation and data in a way that scales

Cryptographic
operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems

- Protect high-impact applications

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- **Private nearest-neighbor search**
- **Private matrix factorization**

Cryptographic
operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- **Private matrix multiplication**
- **Private information retrieval**
- **Homomorphic encryption**

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems

- ▶ Protect high-impact applications

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- Private matrix factorization

Cryptographic
operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- Private recommendations

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- Private matrix factorization

Cryptographic operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- Private recommendations

Systems

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- Private matrix factorization

Security

Cryptographic
operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

Cryptography

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems (SOSP'23, USENIX Sec'26) **Systems**

- Private web search
- Private recommendations

ML

Distributed computing

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25) **Security**

- Private nearest-neighbor search
- Private matrix factorization

Data mining

Differential privacy

Cryptographic
operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26) **Cryptography**

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

Complexity

Coding theory

Data structures

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- Private recommendations

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search → In your iPhone: “Enhanced visual search” in the photos app [ABGM+24]
- Private matrix factorization

Cryptographic operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

This Talk: The design of a private search engine

Applications

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- Private recommendations

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- Private matrix factorization

In your iPhone:
“Enhanced visual search”
in the photos app [ABGM+24]

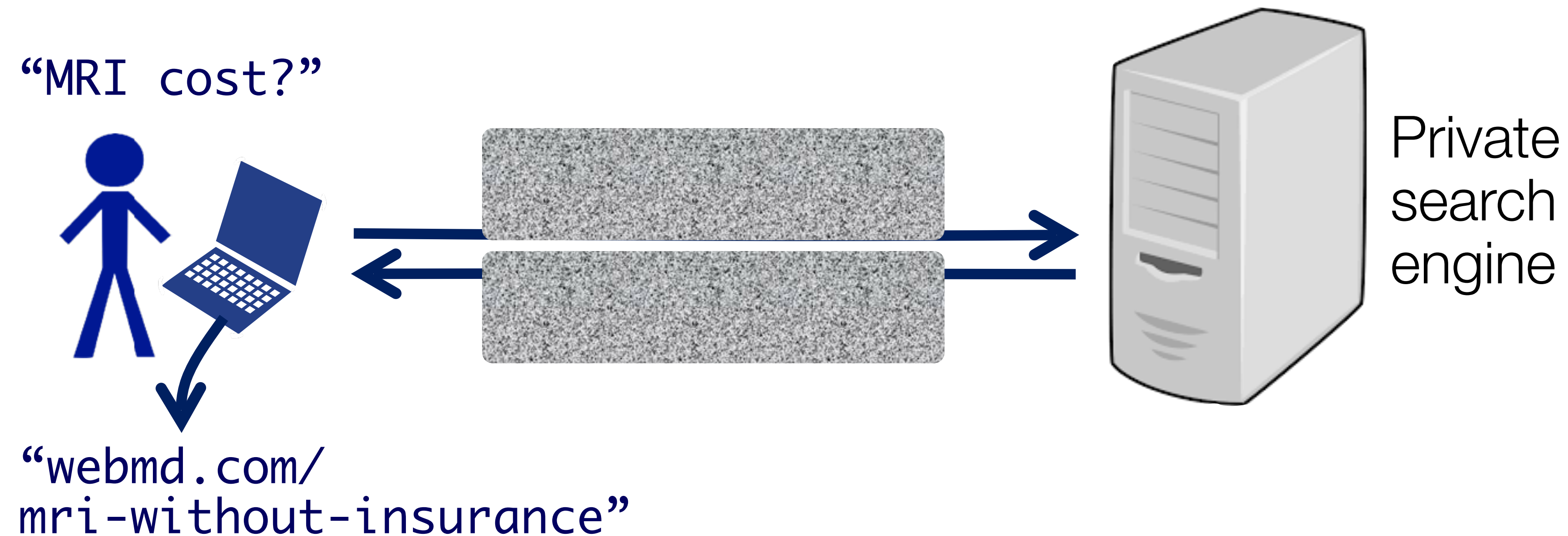


Cryptographic
operations

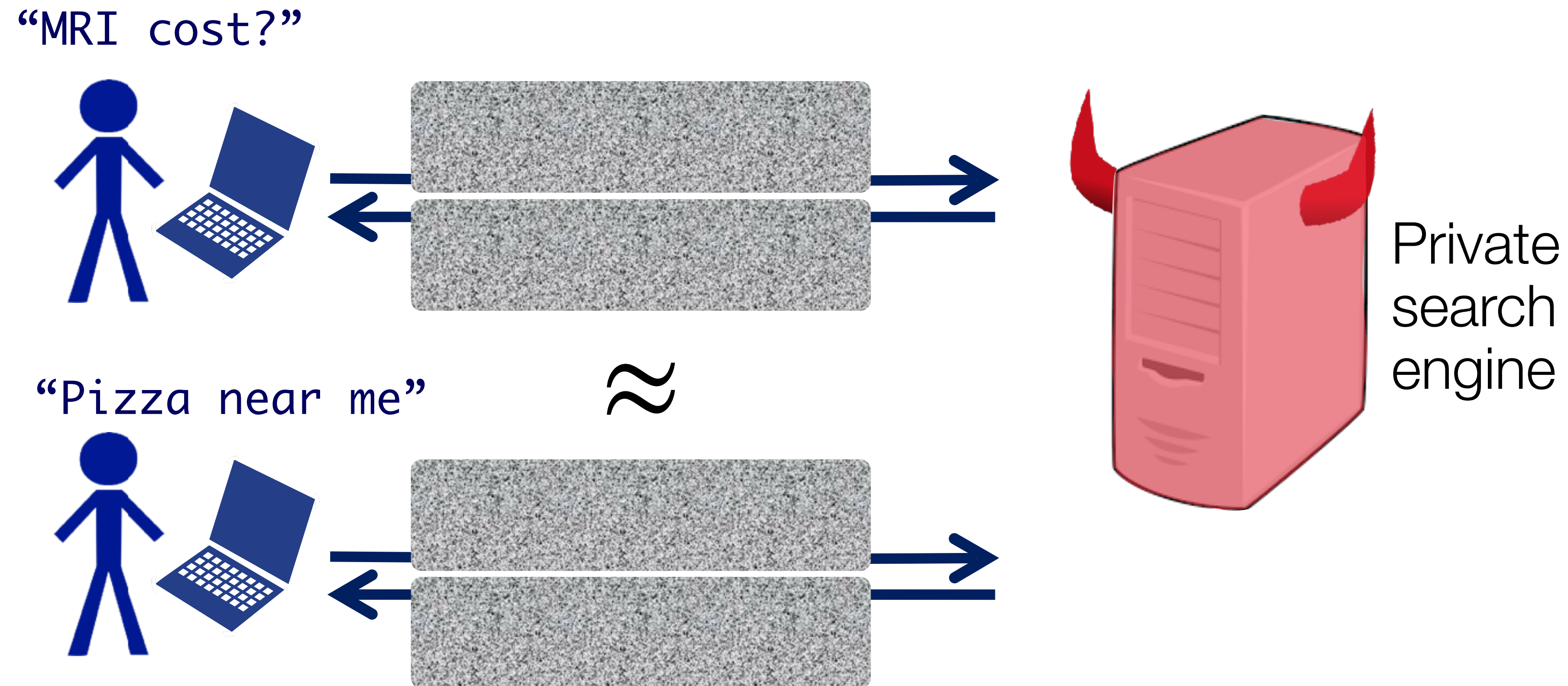
Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

Privacy definition: Search without revealing query

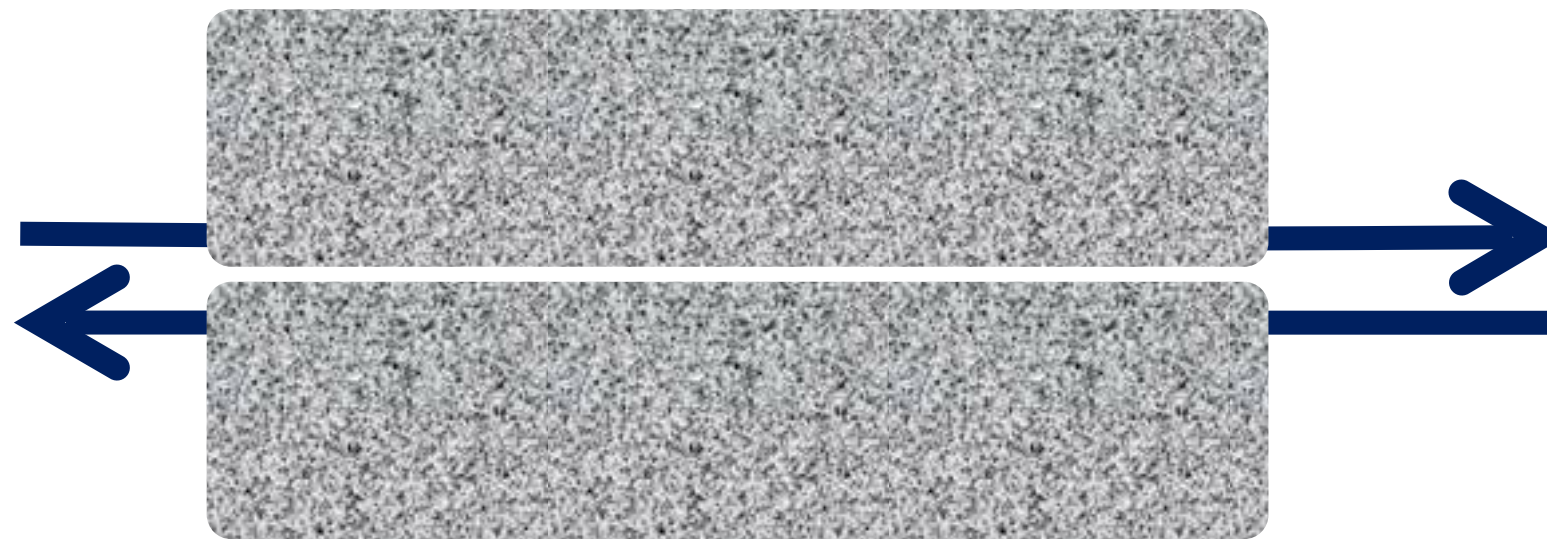


Privacy definition: Search without revealing query



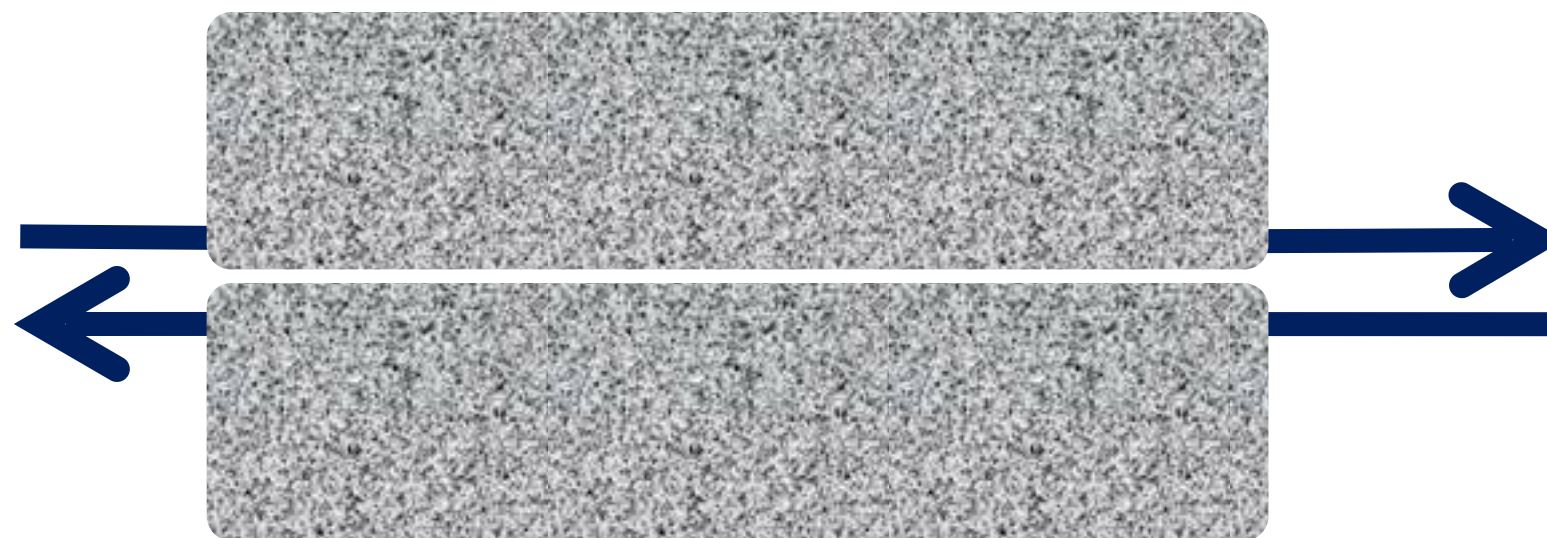
Privacy definition: Search without revealing query

“MRI cost?”



Private
search
engine

“Pizza near me”



≈

Limitations, for now:

- won't hide *when* the client searches
- won't guarantee integrity
- won't hide HTTP(S) requests after search



Tiptoe: A private search engine

Henzinger, Dauterman, Corrigan-Gibbs, and Zeldovich (SOSP'23)

Privacy: Search engine learns no information about queries

Scale: Searches over public web crawl (360M pages) in 2.7s of latency
using 145 core-seconds of compute and 57 MiB of traffic per query

Quality: Ranks the best search result in the top 10, on average
though less accurate than best non-private search

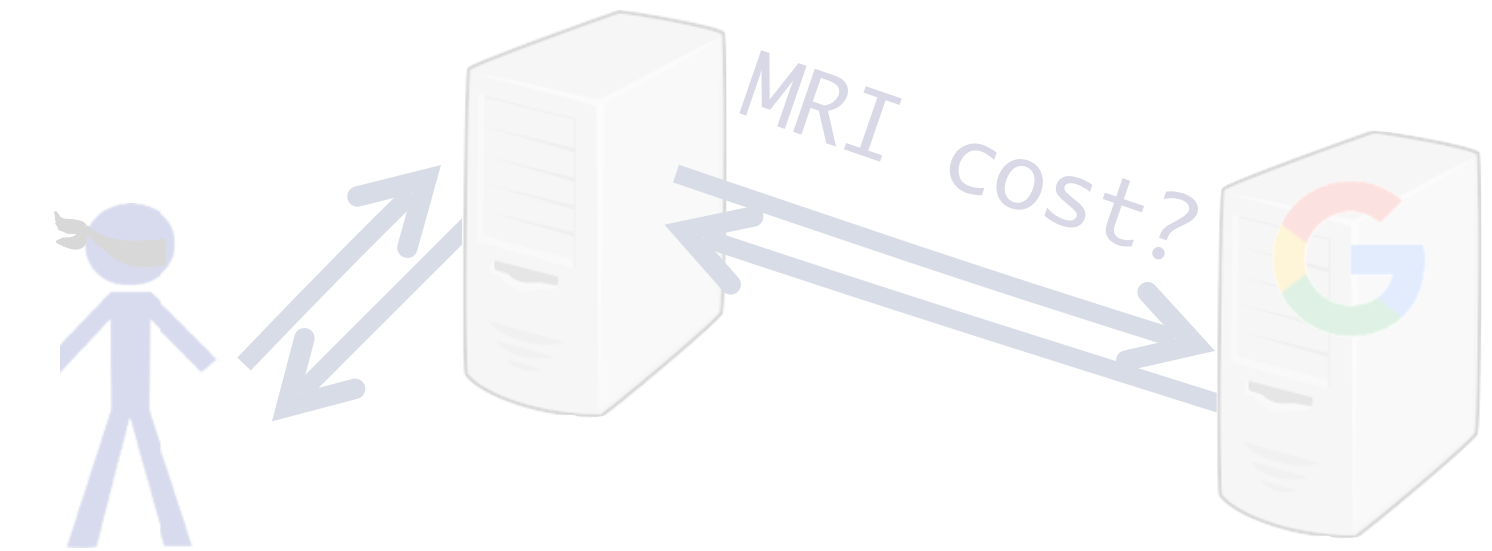
The first private search engine to work at this scale

Prior work in private search:

Relaxed privacy guarantees

- Anonymize queries

[DMS04, MC09, YWP09, BTD12, GSSCL14, ADE15]



- Obfuscate queries

[MC09, ND09, YWPC09, RF10, ADE15, BTD12, PCMBK15]



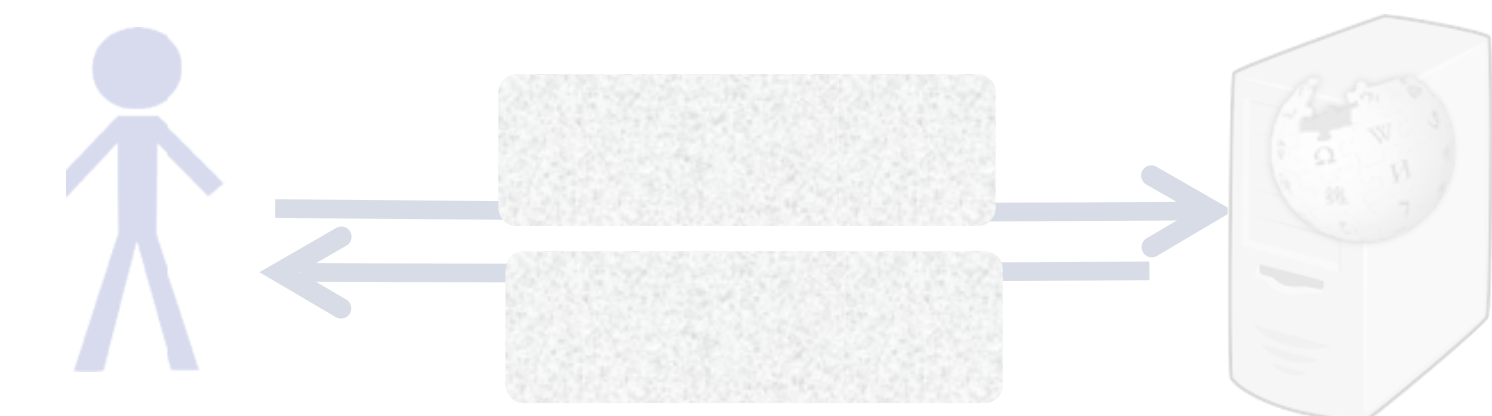
- Trust hardware (SGX)

[MBPFFPS17, PGMBBFKPS18, L7ZTXAC21]



Limited scale

- Coeus (SOSP'21) hides queries to Wikipedia

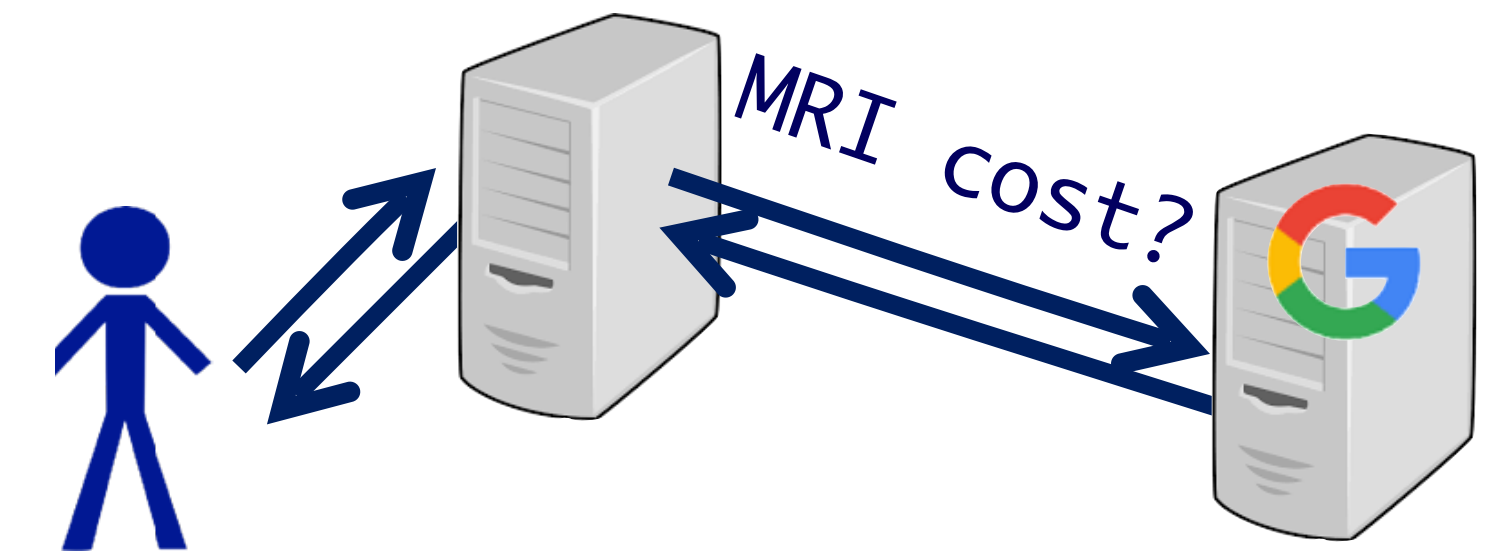


Prior work in private search:

Relaxed privacy guarantees

- Anonymize queries

[DMS04, MC09, YWP09, BT12, GSSCL14, ADE15]



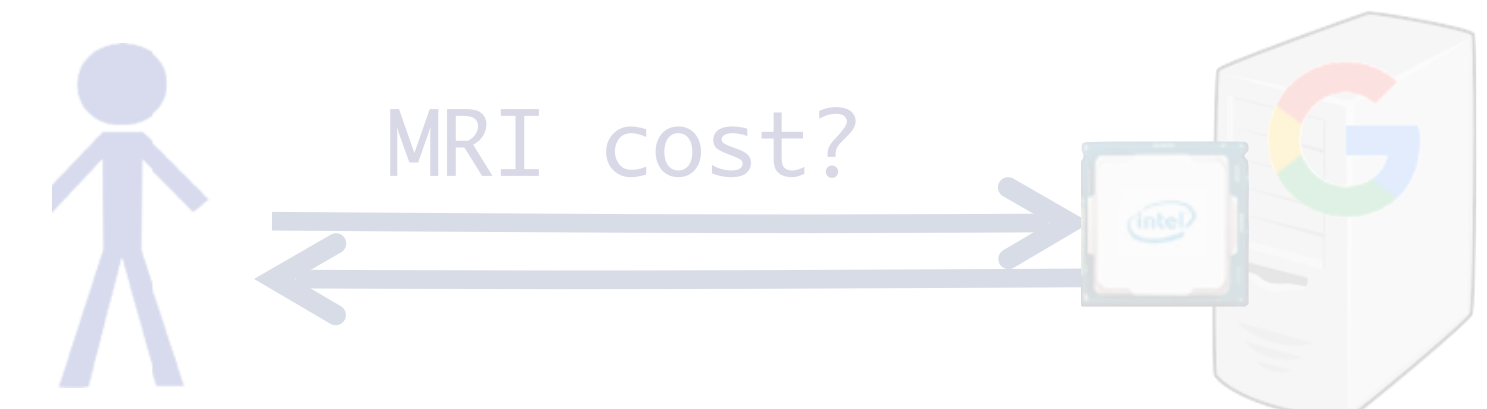
- Obfuscate queries

[MC09, ND09, YWPC09, RF10, ADE15, BT12, PCMBK15]



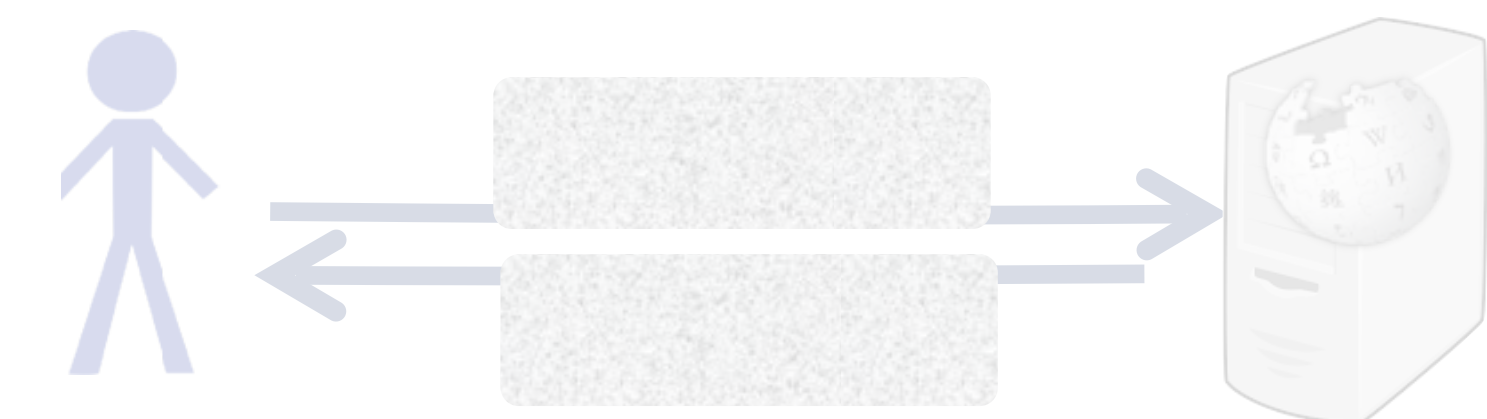
- Trust hardware (SGX)

[MBPFFPS17, PGMBBFKPS18, L7ZTXAC21]



Limited scale

- Coeus (SOSP'21) hides queries to Wikipedia

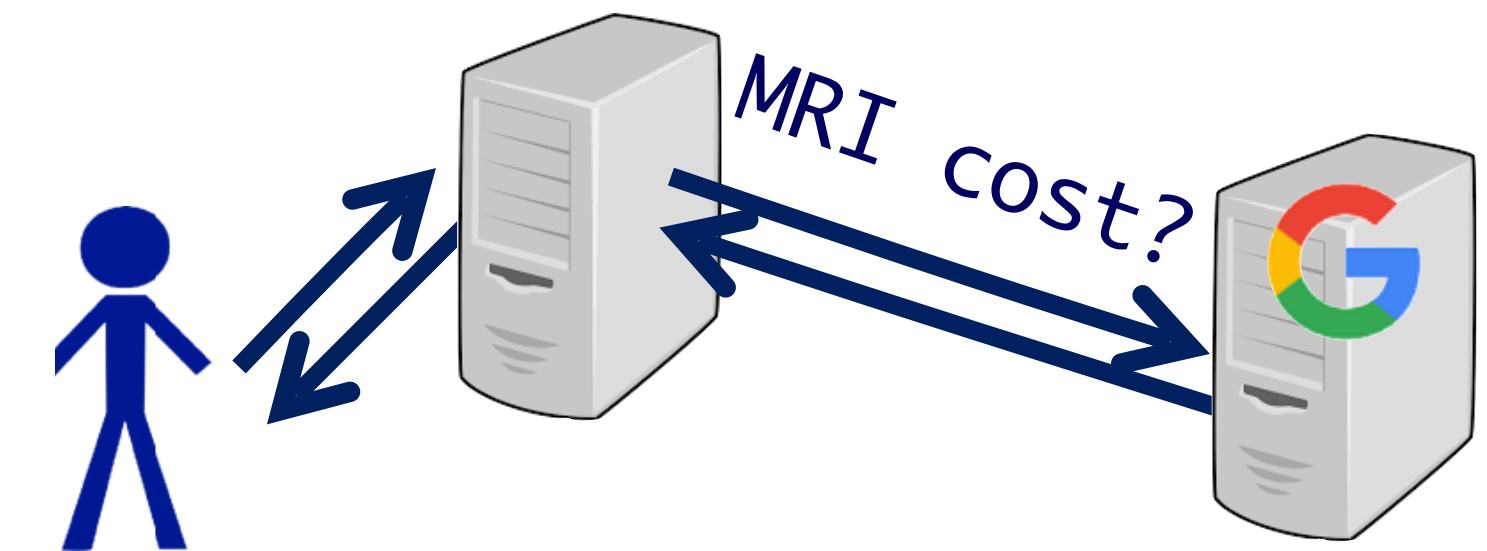


Prior work in private search:

Relaxed privacy guarantees

- Anonymize queries

[DMS04, MC09, YWP09, BT12, GSSCL14, ADE15]



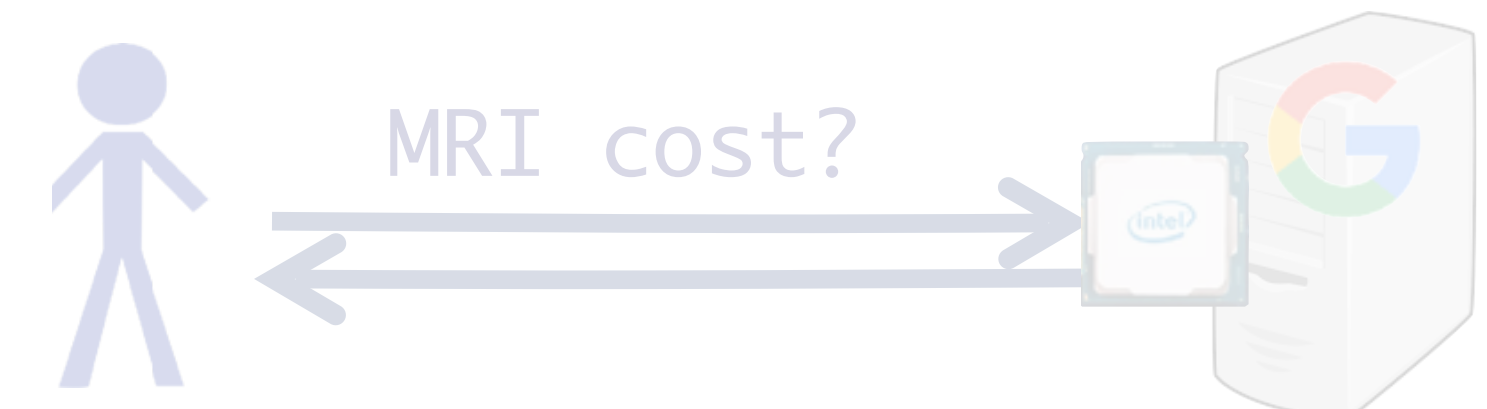
- Obfuscate queries

[MC09, ND09, YWPC09, RF10, ADE15, BT12, PCMBK15]



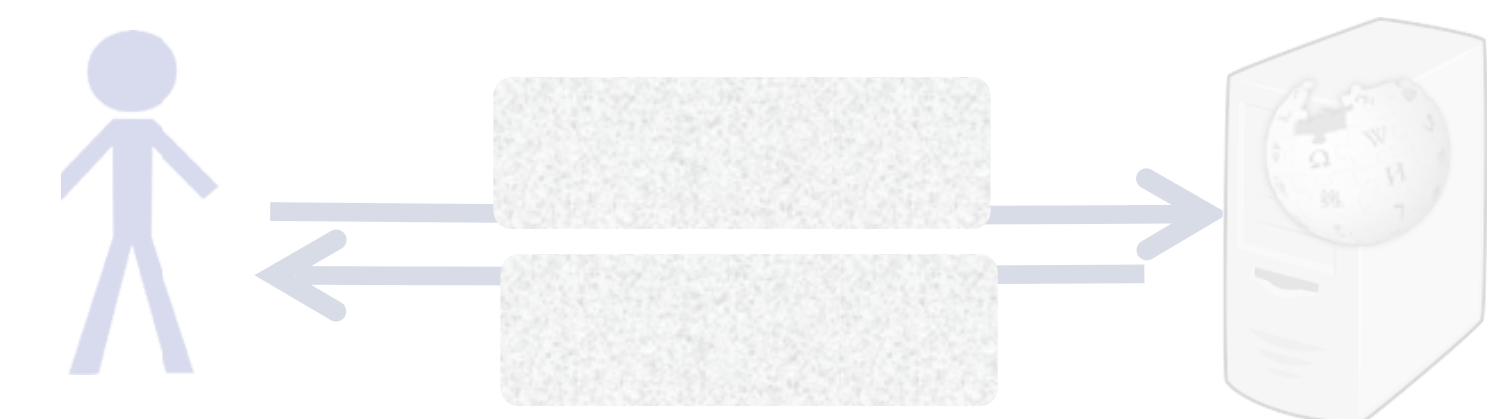
- Trust hardware (SGX)

[MBPFFPS17, PGMBBFKPS18, L7ZTXAC21]



Limited scale

- Coeus (SOSP'21) hides queries to Wikipedia

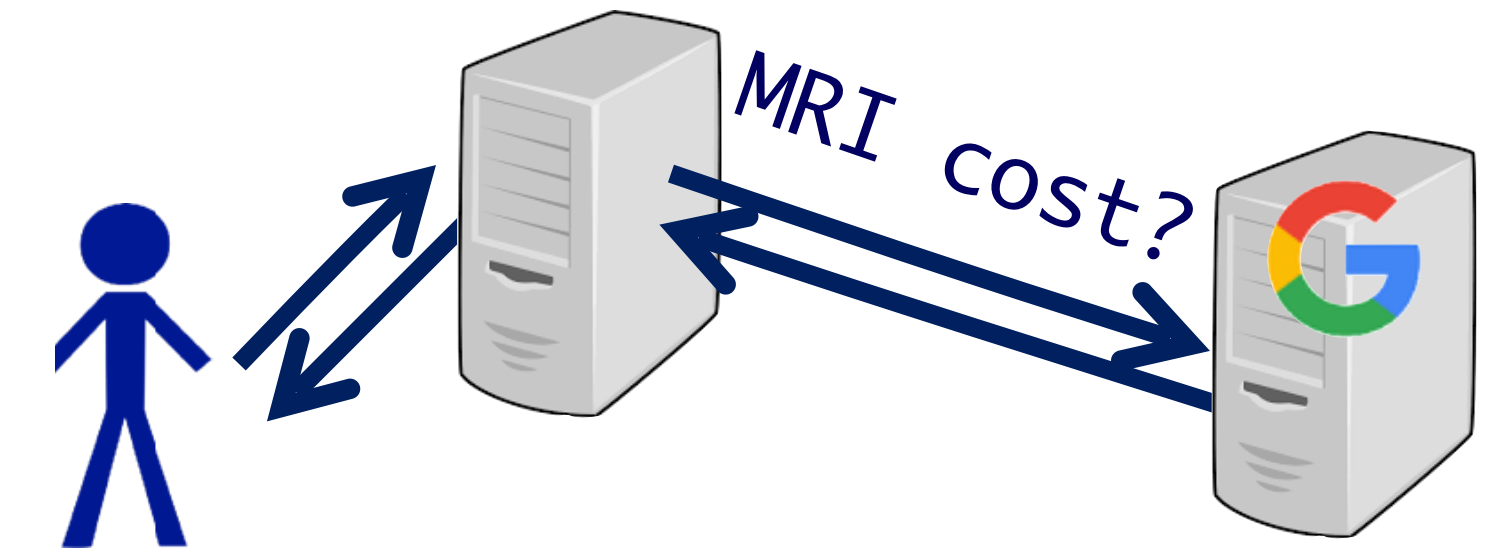


Prior work in private search:

Relaxed privacy guarantees

- Anonymize queries

[DMS04, MC09, YWP09, BTD12, GSSCL14, ADE15]



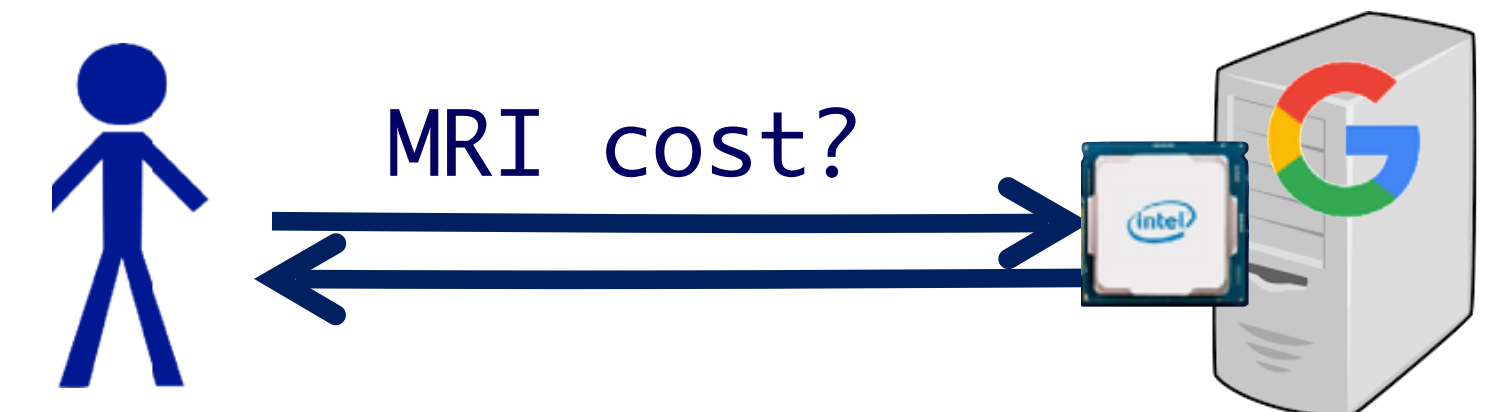
- Obfuscate queries

[MC09, ND09, YWPC09, RF10, ADE15, BTD12, PCMBK15]



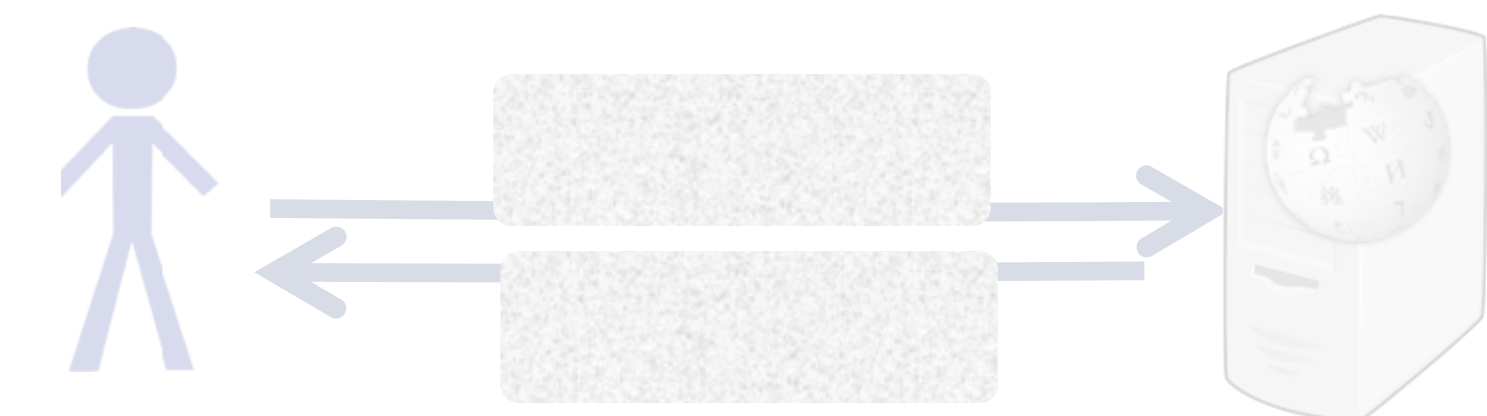
- Trust hardware (SGX)

[MBPFFPS17, PGMBBFFKPS18, L7ZTXAC21]



Limited scale

- Coeus (SOSP'21) hides queries to Wikipedia

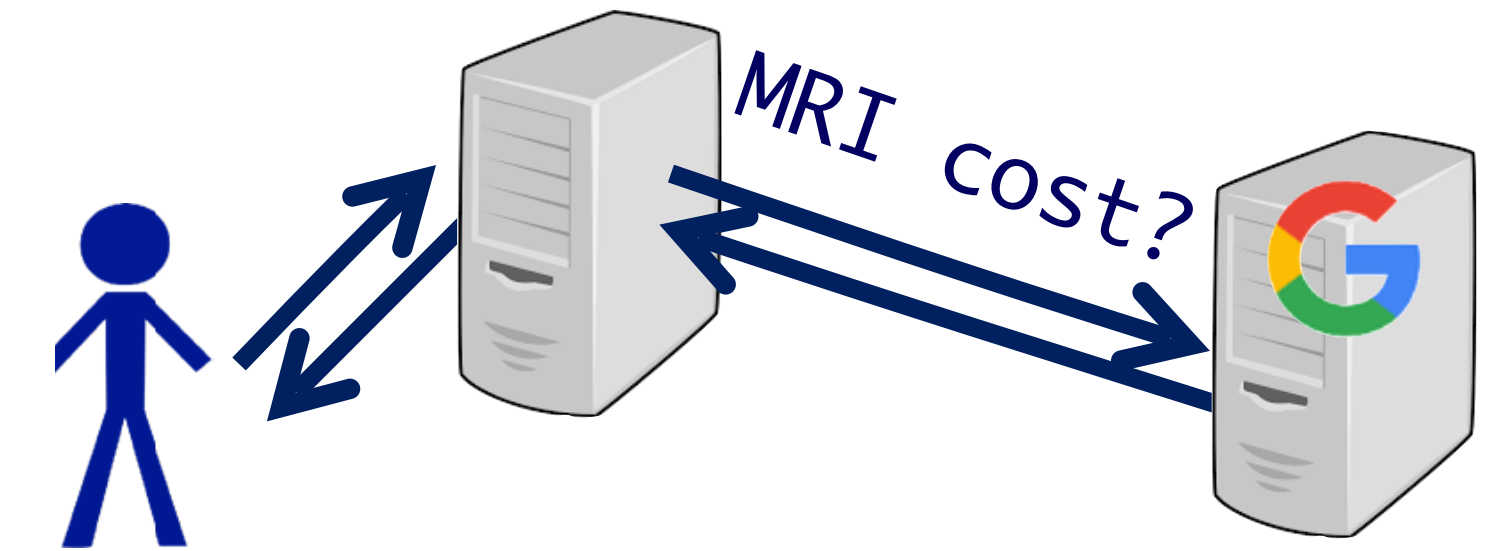


Prior work in private search:

Relaxed privacy guarantees

- Anonymize queries

[DMS04, MC09, YWP09, BTD12, GSSCL14, ADE15]



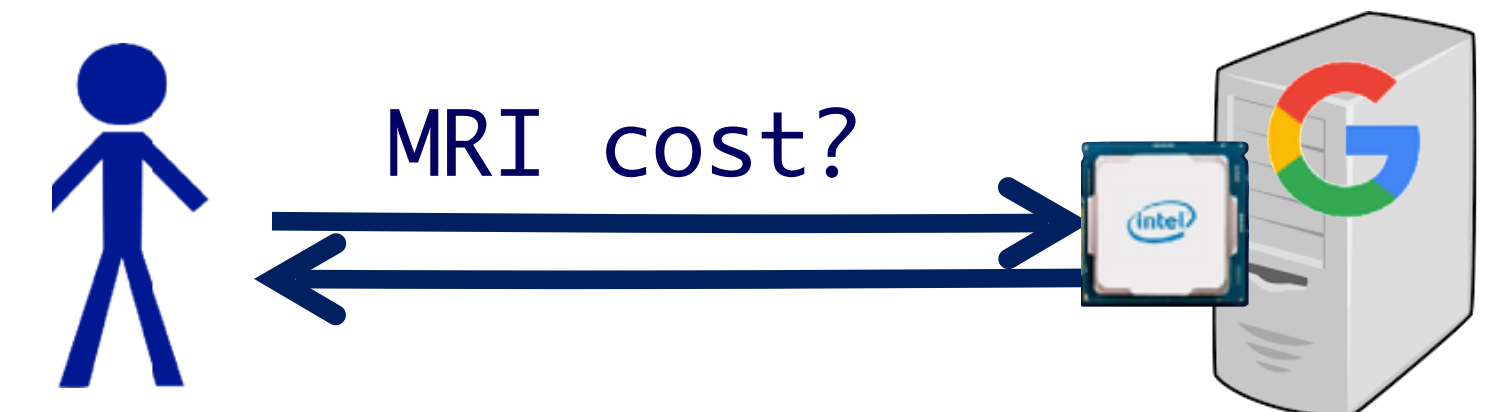
- Obfuscate queries

[MC09, ND09, YWPC09, RF10, ADE15, BTD12, PCMBK15]



- Trust hardware (SGX)

[MBPFFPS17, PGMBBFFKPS18, L7ZTXAC21]



Limited scale

- [Coeus](#) (SOSP'21) hides queries to Wikipedia

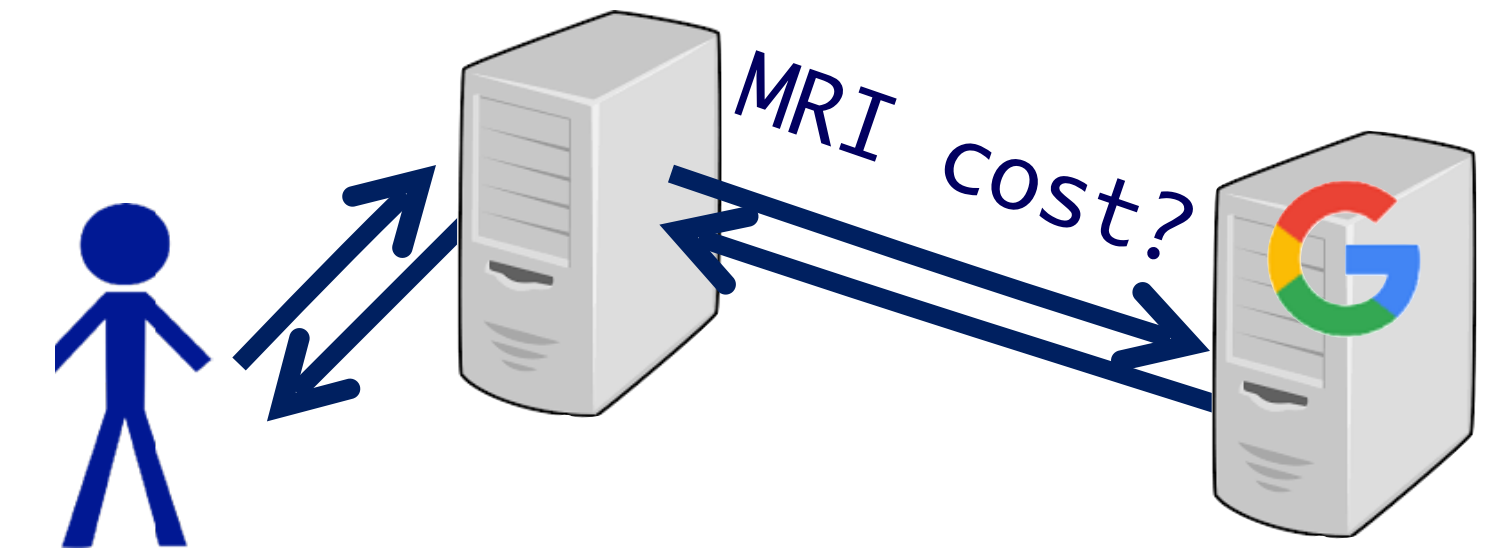


Prior work in private search:

Relaxed privacy guarantees

- Anonymize queries

[DMS04, MC09, YWP09, BTD12, GSSCL14, ADE15]



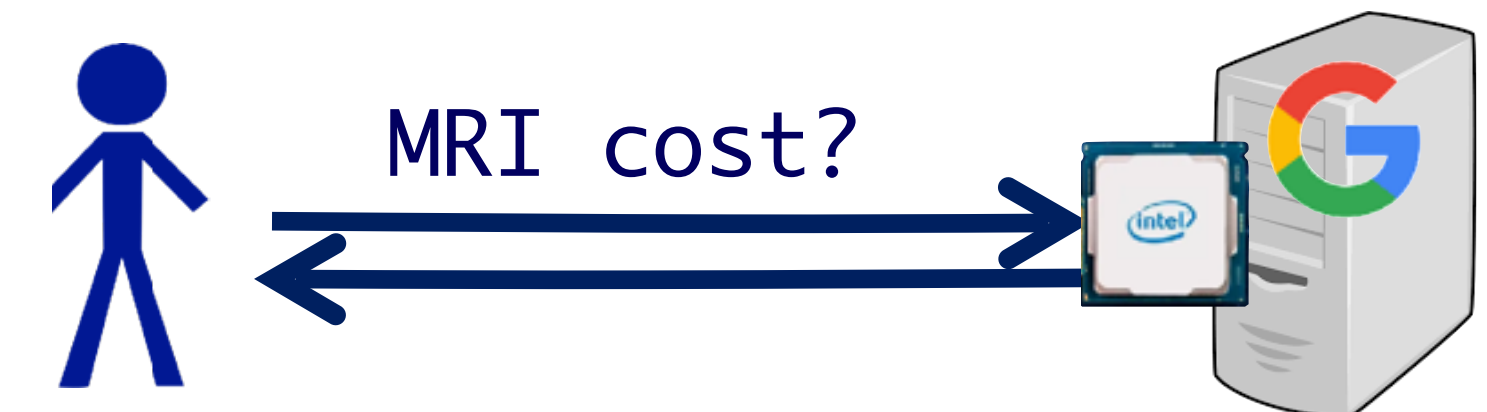
- Obfuscate queries

[MC09, ND09, YWPC09, RF10, ADE15, BTD12, PCMBK15]



- Trust hardware (SGX)

[MBPFFPS17, PGMBBFFKPS18, L7ZTXAC21]



Limited scale

Why?

- Coeus (SOSP'21) hides queries to Wikipedia

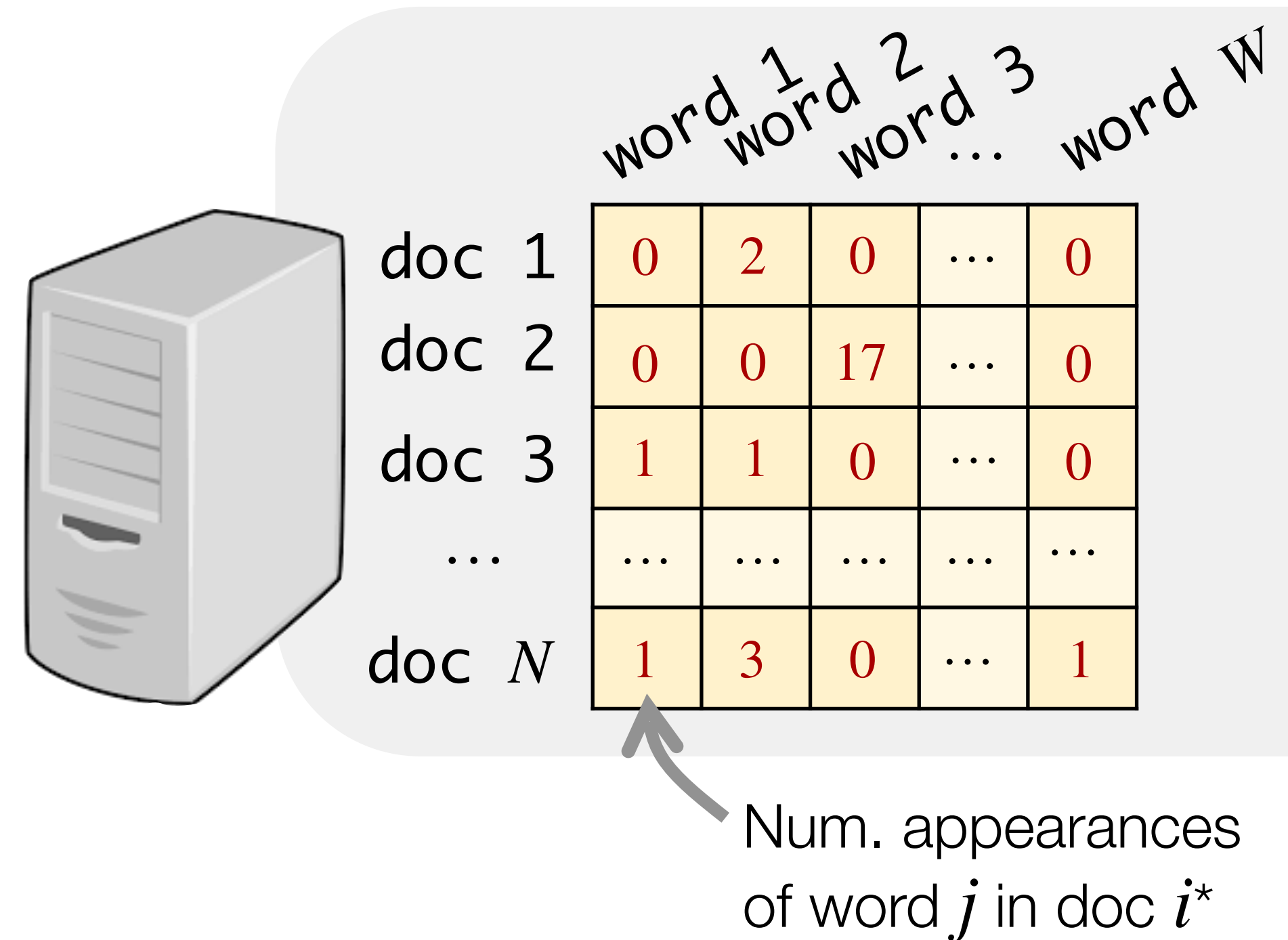


Challenge: Build search from encrypted ADD and MUL

Coeus (SOSP'21): count appearances of query words in each doc* (TF-IDF search) [RS76,SM86]

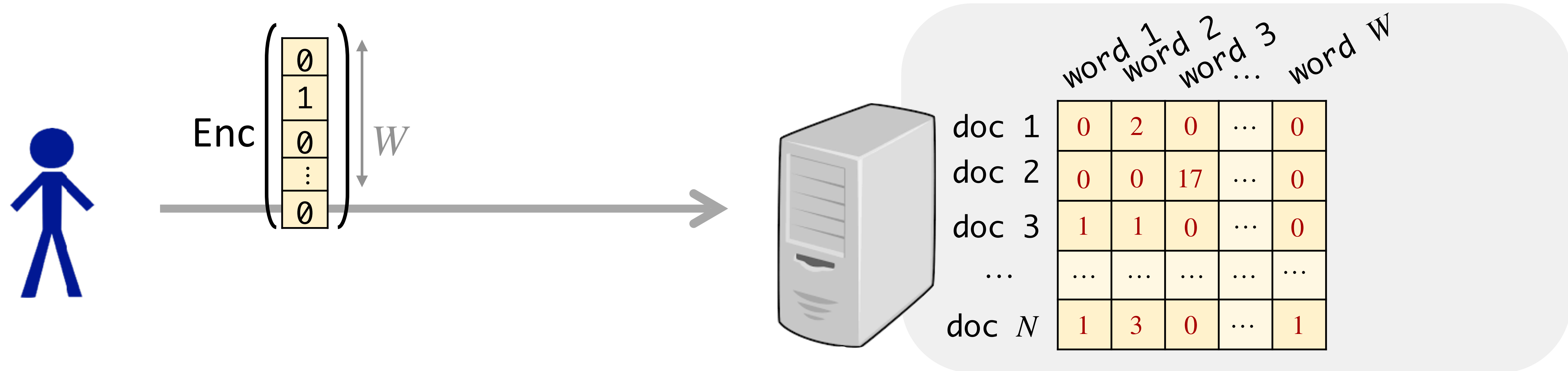
Challenge: Build search from encrypted ADD and MUL

Coeus (SOSP'21): count appearances of query words in each doc* (TF-IDF search) [RS76, SM86]



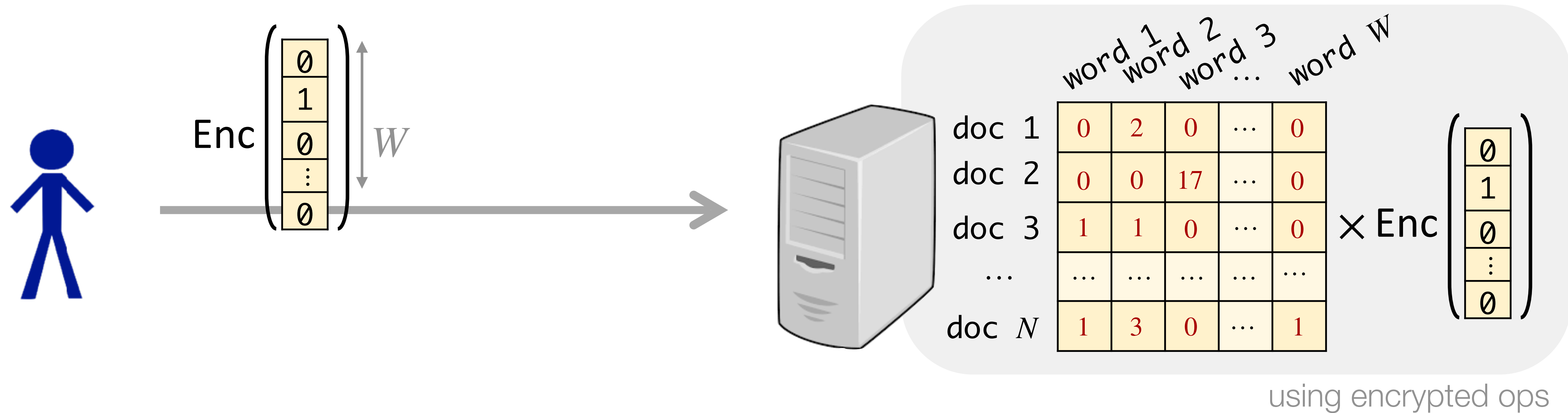
Challenge: Build search from encrypted ADD and MUL

Coeus (SOSP'21): count appearances of query words in each doc* (TF-IDF search) [RS76, SM86]



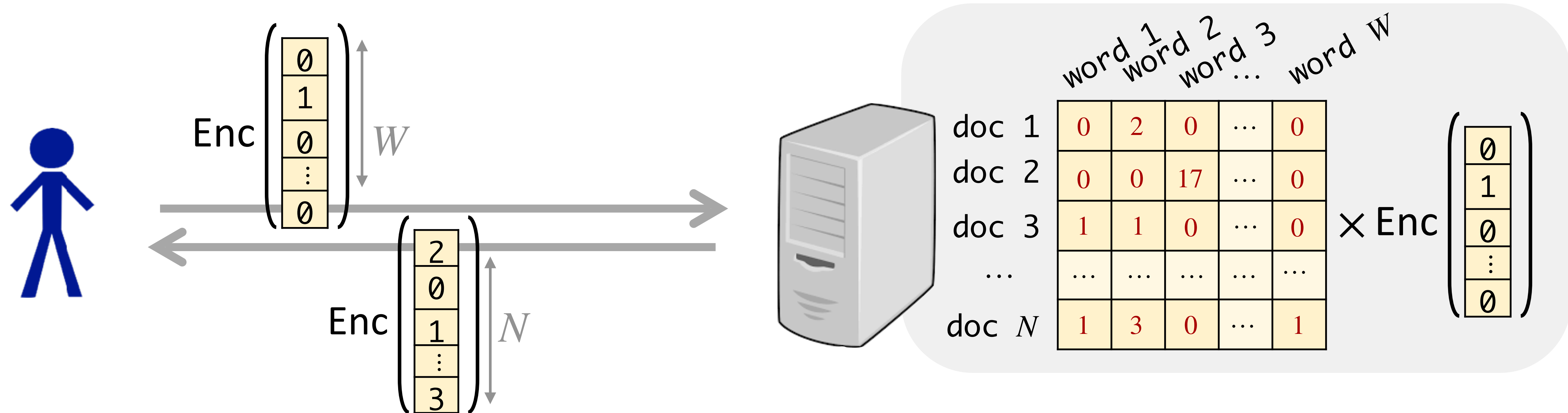
Challenge: Build search from encrypted ADD and MUL

Coeus (SOSP'21): count appearances of query words in each doc* (TF-IDF search) [RS76, SM86]



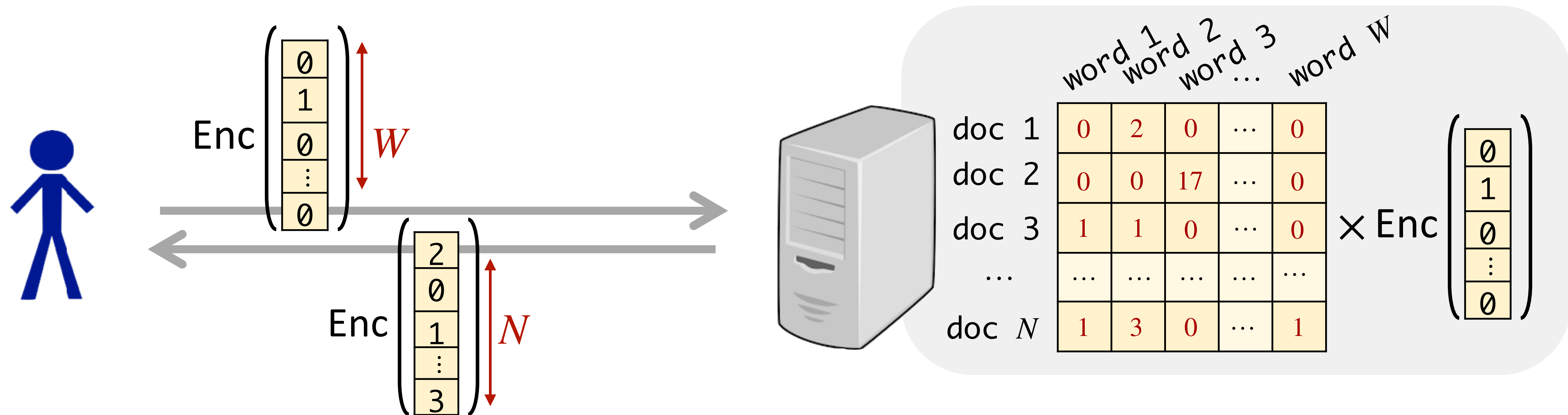
Challenge: Build search from encrypted ADD and MUL

Coeus (SOSP'21): count appearances of query words in each doc* (TF-IDF search) [RS76, SM86]



Challenge: Build search from encrypted ADD and MUL

Coeus (SOSP'21): count appearances of query words in each doc* (TF-IDF search) [RS76, SM86]

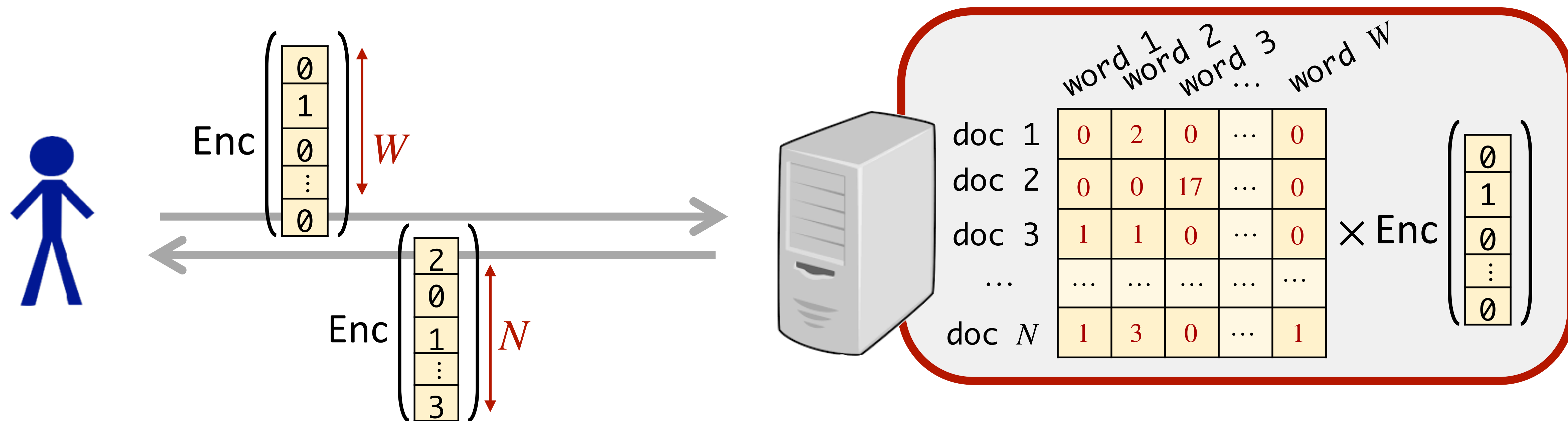


On every query,

- communication: $\geq N + W$ bytes

Challenge: Build search from encrypted ADD and MUL

Coeus (SOSP'21): count appearances of query words in each doc* (TF-IDF search) [RS76, SM86]

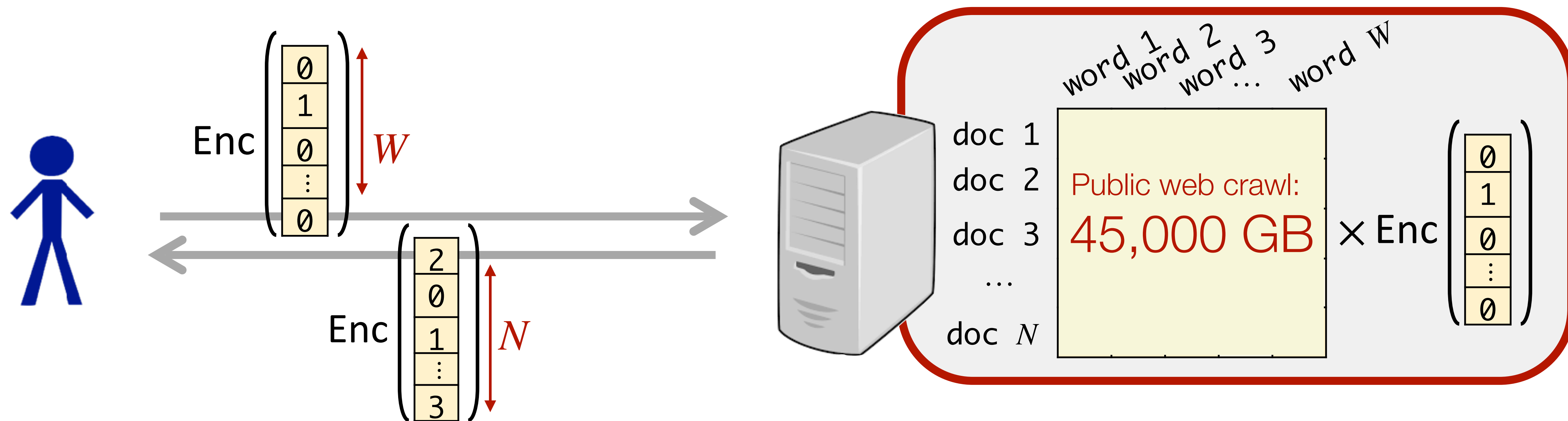


On every query,

- communication: $\geq N + W$ bytes
- computation: ≥ 1 encrypted operation for every word in every doc

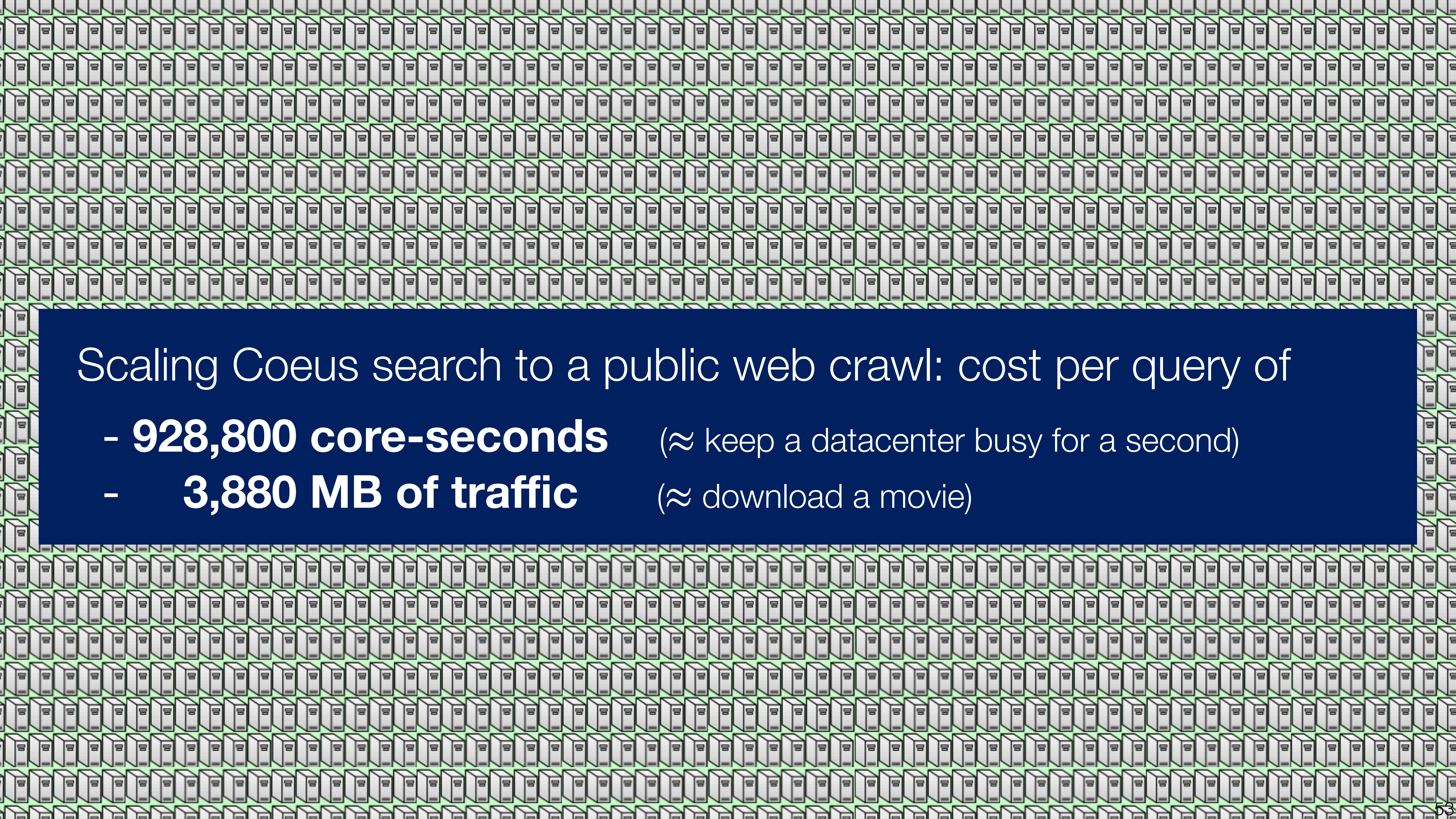
Challenge: Build search from encrypted ADD and MUL

Coeus (SOSP'21): count appearances of query words in each doc* (TF-IDF search) [RS76, SM86]



On every query,

- communication: $\geq N + W$ bytes
- computation: ≥ 1 encrypted operation for every word in every doc

The background of the slide is a repeating pattern of server racks. Each rack is depicted as a light gray rectangular unit with a dark gray front panel featuring a small square indicator. The racks are arranged in a dense, staggered grid, creating a sense of depth and scale. The pattern covers the entire slide area, with a dark blue rectangular box overlaid in the center to contain the text.

Scaling Coeus search to a public web crawl: cost per query of

- **928,800 core-seconds** ($\approx$ keep a datacenter busy for a second)
- **3,880 MB of traffic** ($\approx$ download a movie)

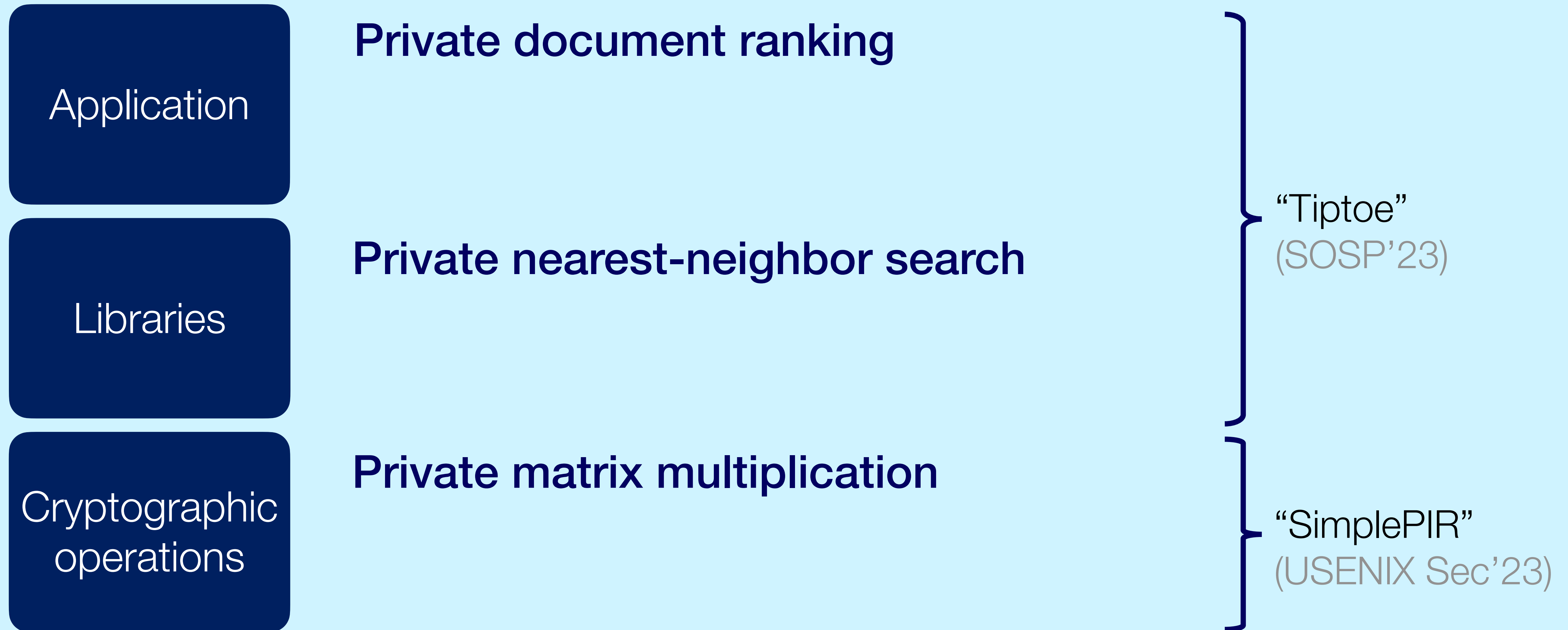
Scaling Coeus search to a public web crawl: cost per query of

- **928,800 core-seconds** ($\approx$ keep a datacenter busy for a second)
- **3,880 MB of traffic** ($\approx$ download a movie)

$\Rightarrow$ Need new system design

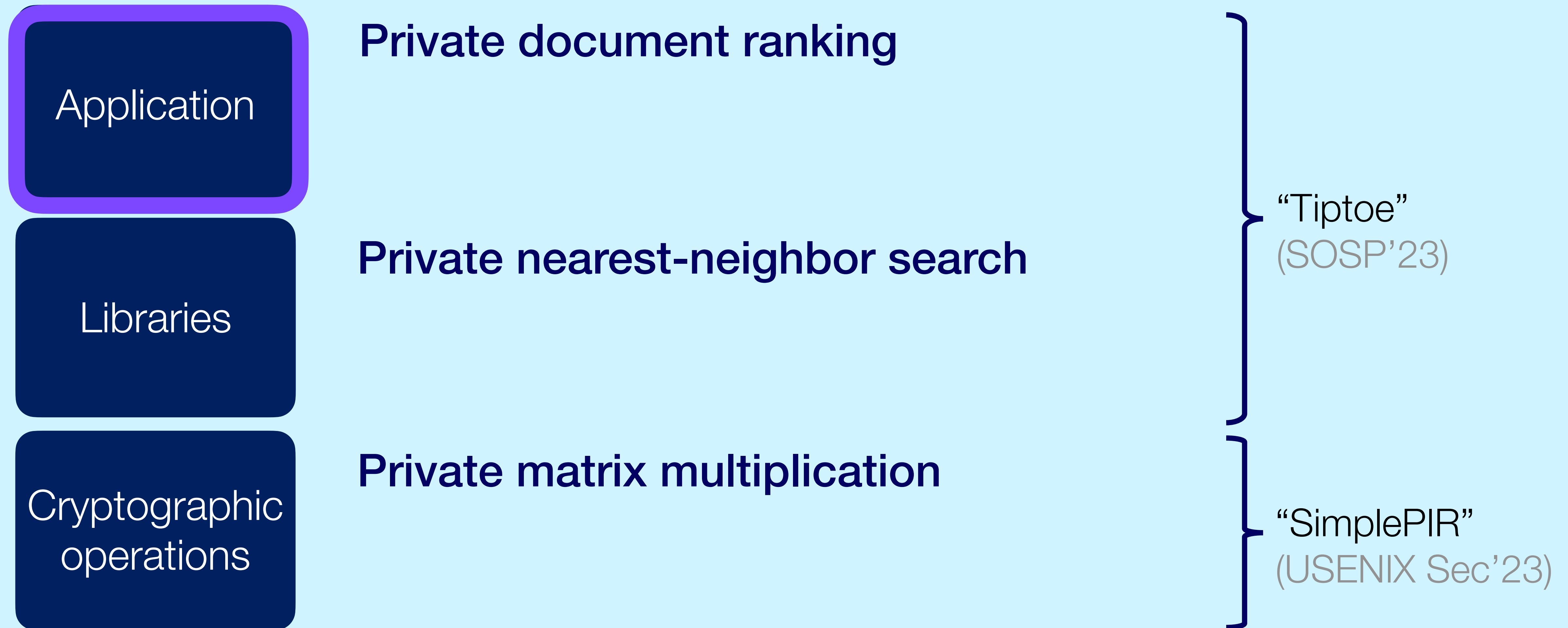


Tiptoe: Design in layers





Tiptoe: Design in layers



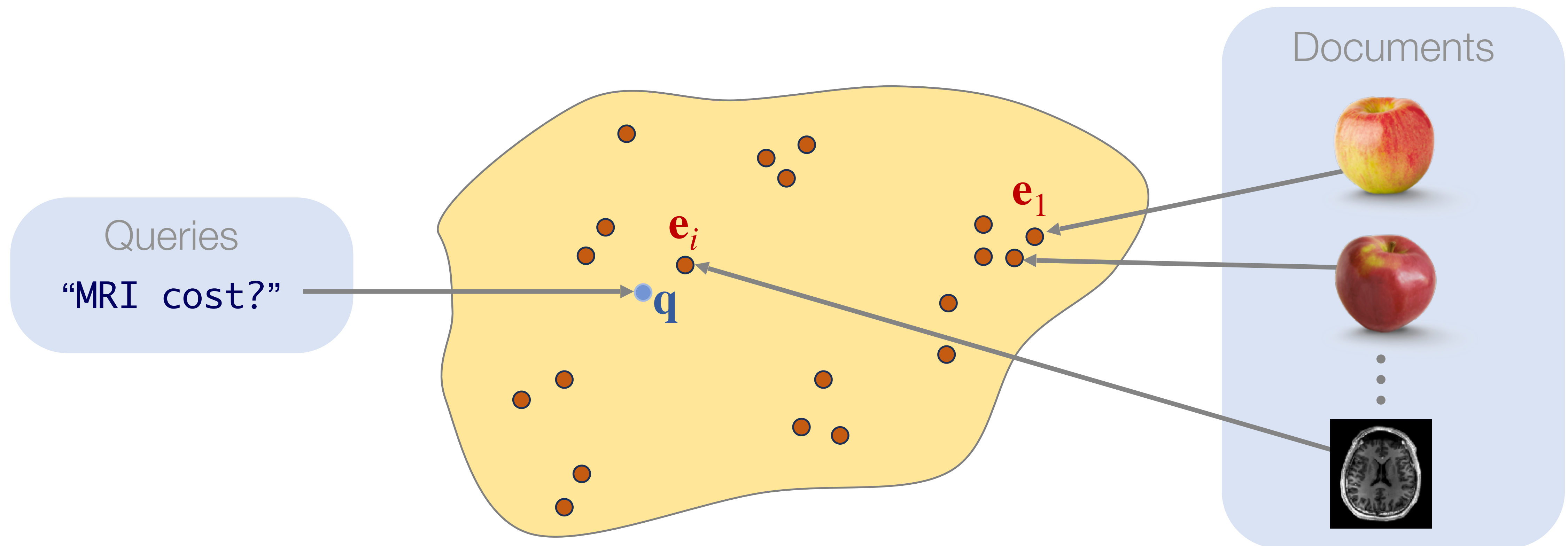


Tiptoe: Design in layers



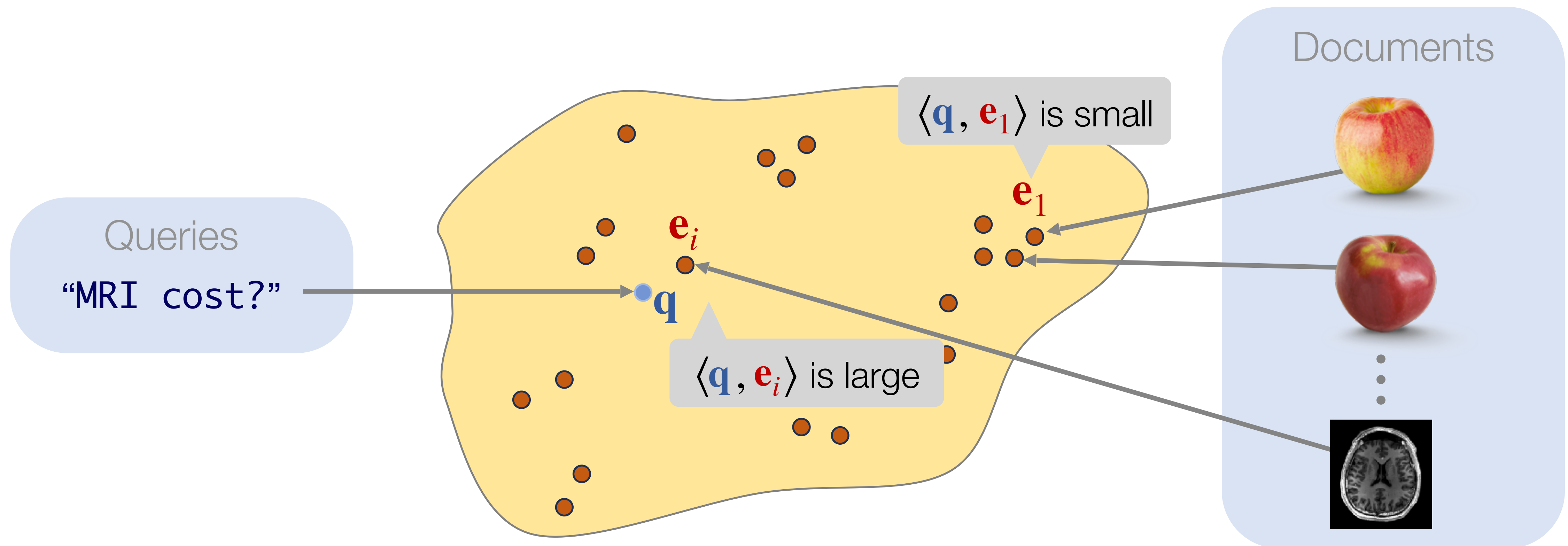
New approach: Private search via semantic embeddings

- Used widely in non-private search (Google, Spotify, YouTube) [Osgood'57, DC'19, MYCG'19, YYZL'19, ...]
- Embeddings map inputs into $\mathbb{R}^d$ such that “similar” inputs are “close”
- This talk: inner-product distance for “closeness” and $d \approx 192$

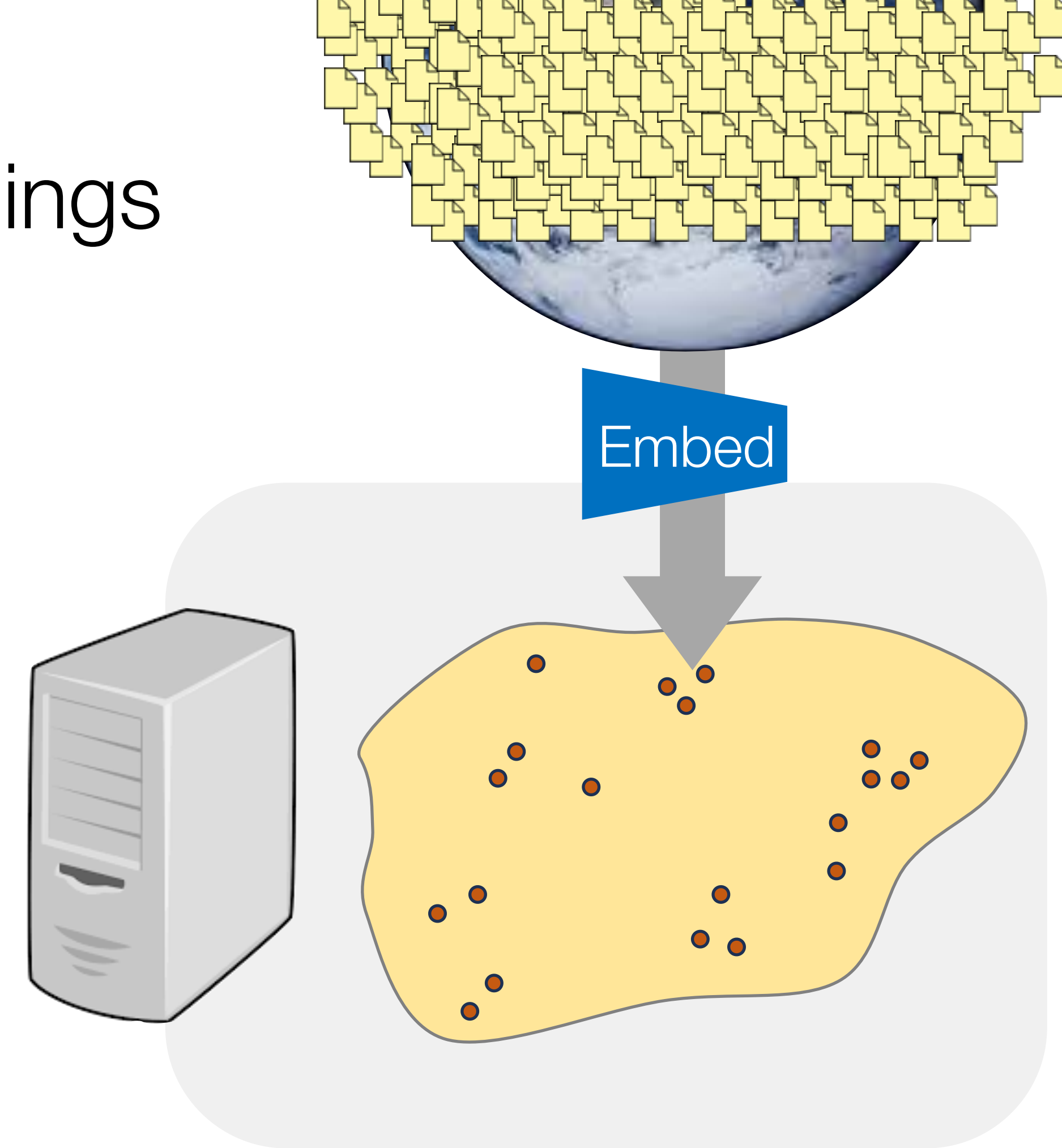


New approach: Private search via semantic embeddings

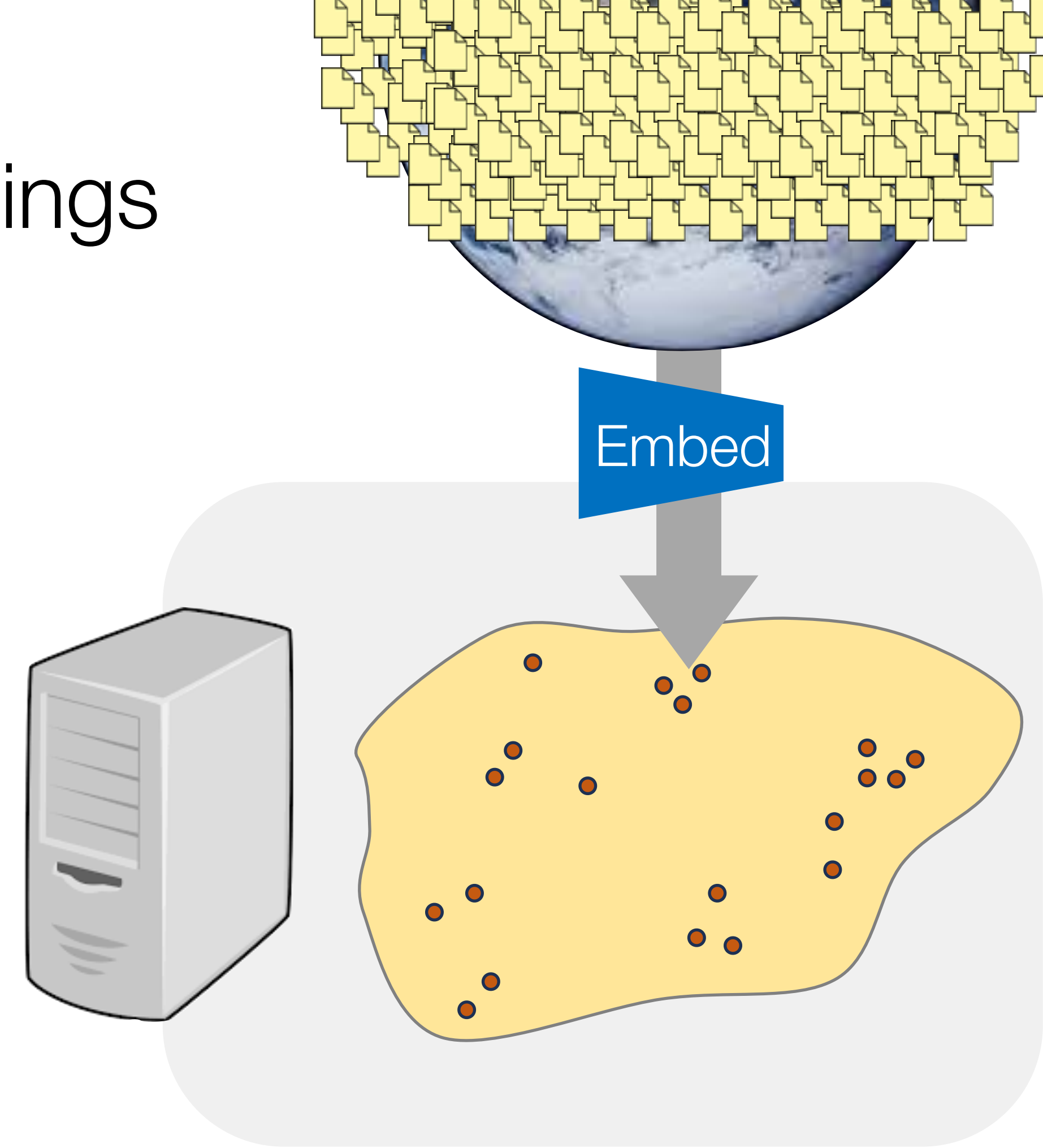
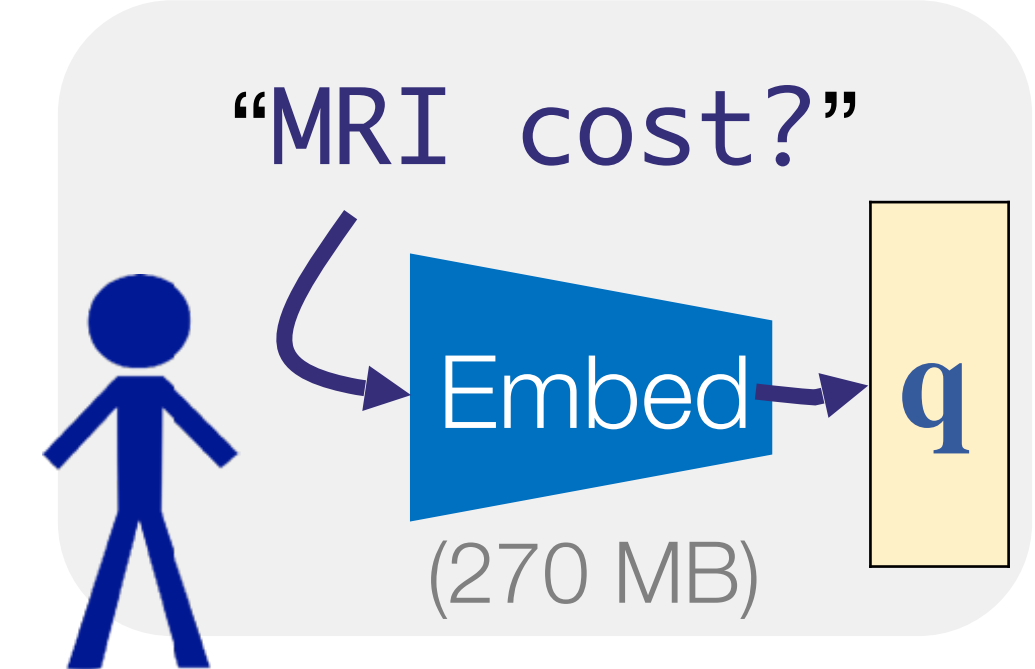
- Used widely in non-private search (Google, Spotify, YouTube) [Osgood'57, DC'19, MYCG'19, YYZL'19, ...]
- Embeddings map inputs into $\mathbb{R}^d$ such that “similar” inputs are “close”
- This talk: inner-product distance for “closeness” and $d \approx 192$



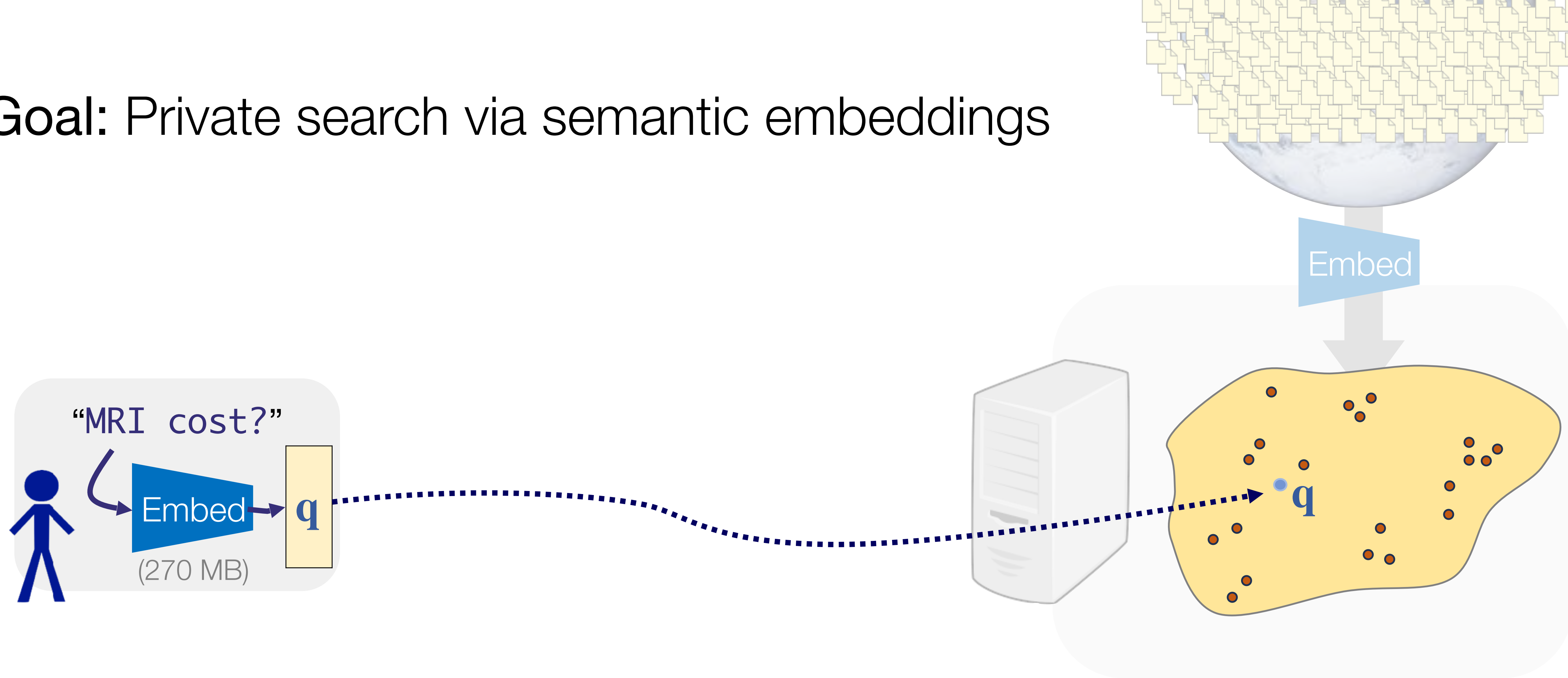
Goal: Private search via semantic embeddings



Goal: Private search via semantic embeddings



Goal: Private search via semantic embeddings



↳ Updated goal: private nearest-neighbor search
i.e., privately find the server-held vector with max inner-product $\langle \mathbf{q}, \mathbf{e} \rangle$

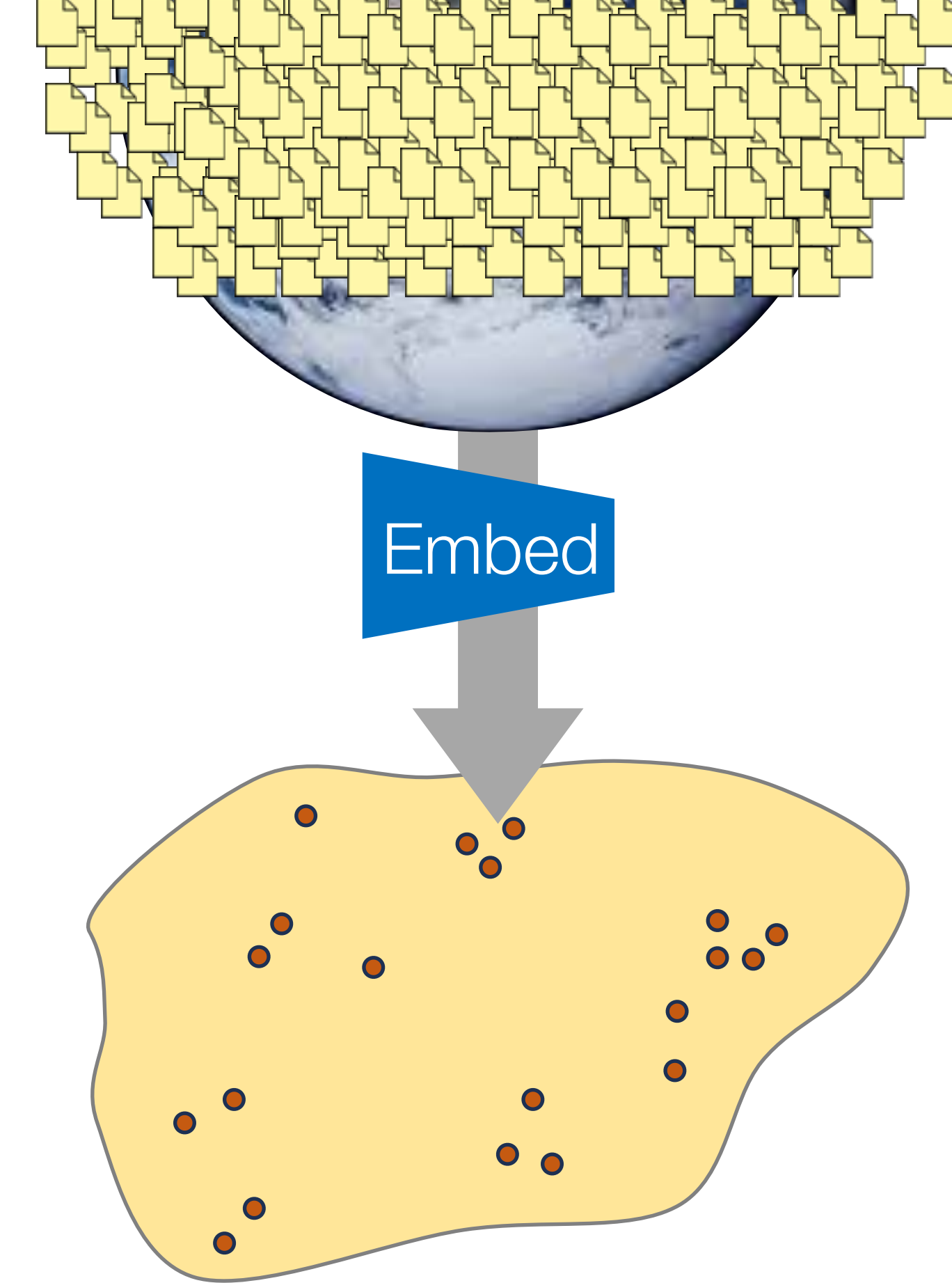
Why embeddings are useful

As in non-private search,

- + **Expressive**: can perform semantic search
- + **General**: exist for text, code, images, ...

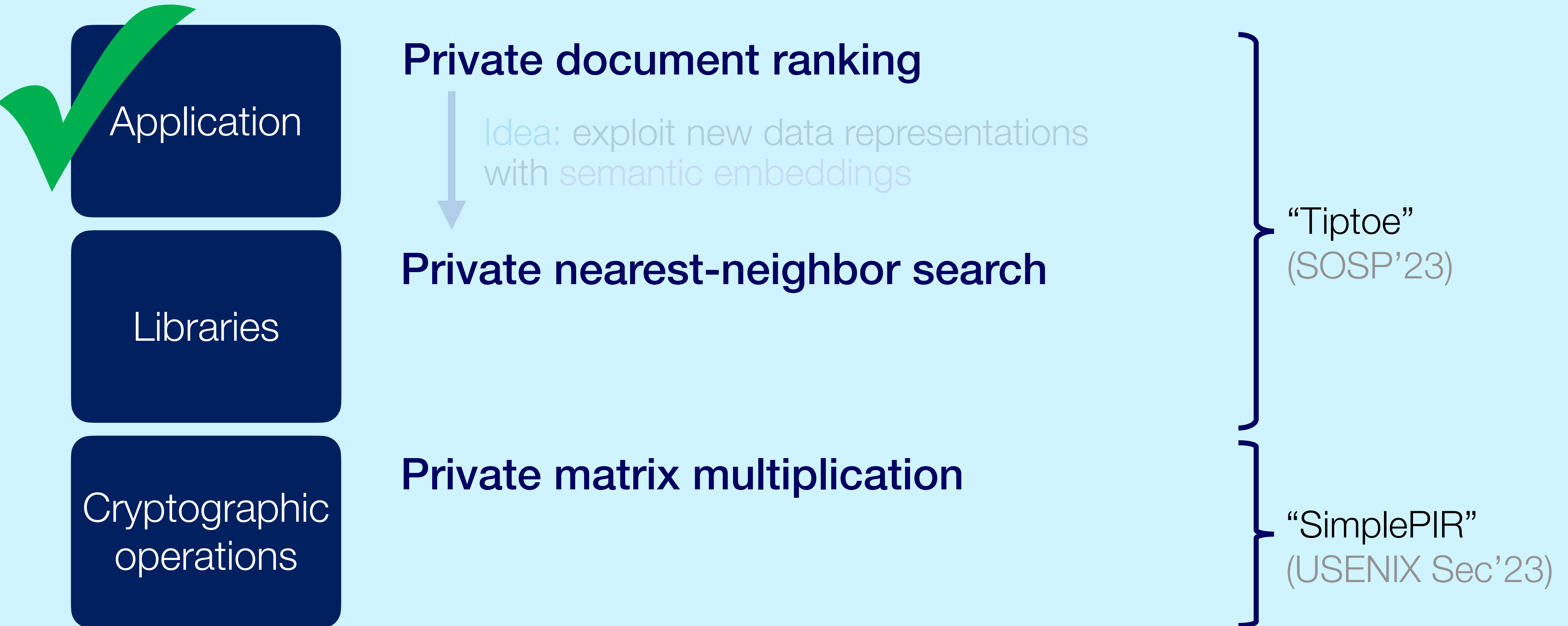
In private search, a **key benefit to scalability**:

- + Embeddings “**linearize**” the computation:
produce crypto-friendly operations, without large doc representations



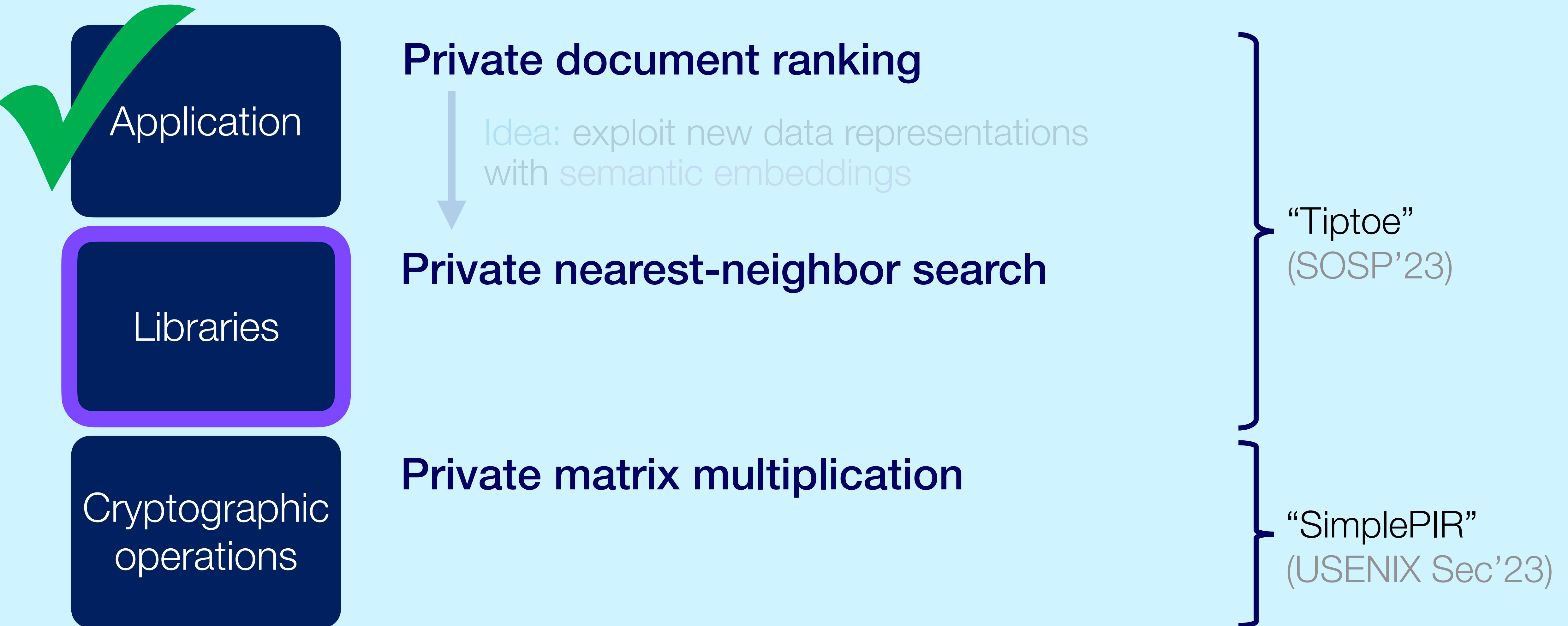


Tiptoe: Design in layers



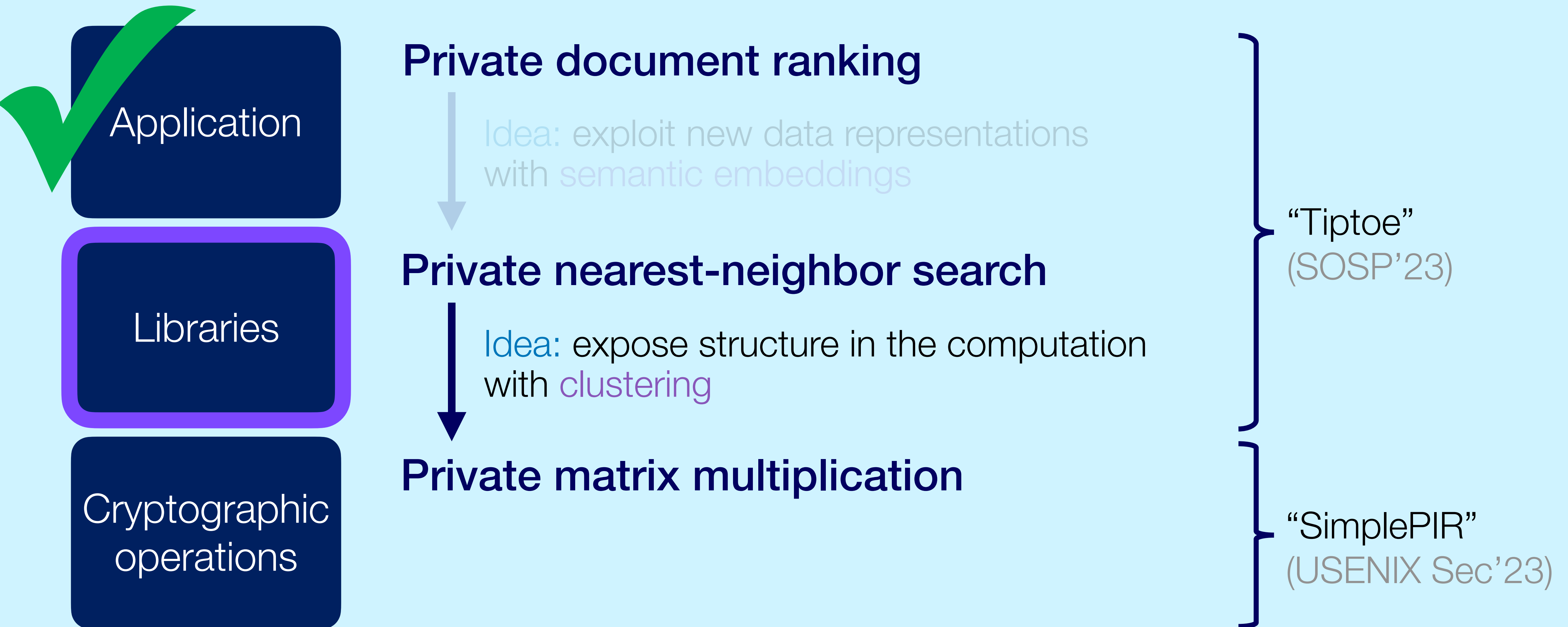


Tiptoe: Design in layers

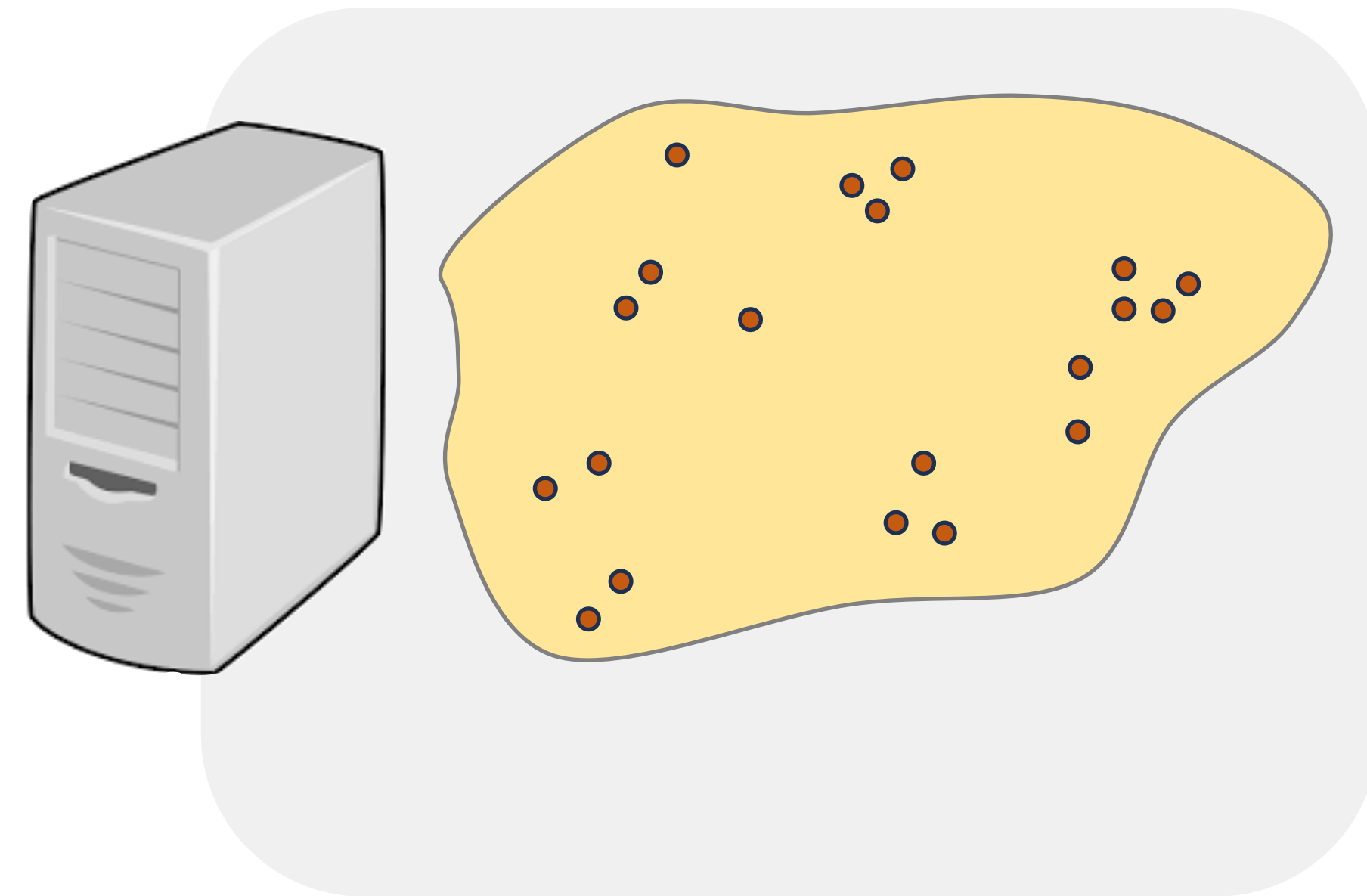
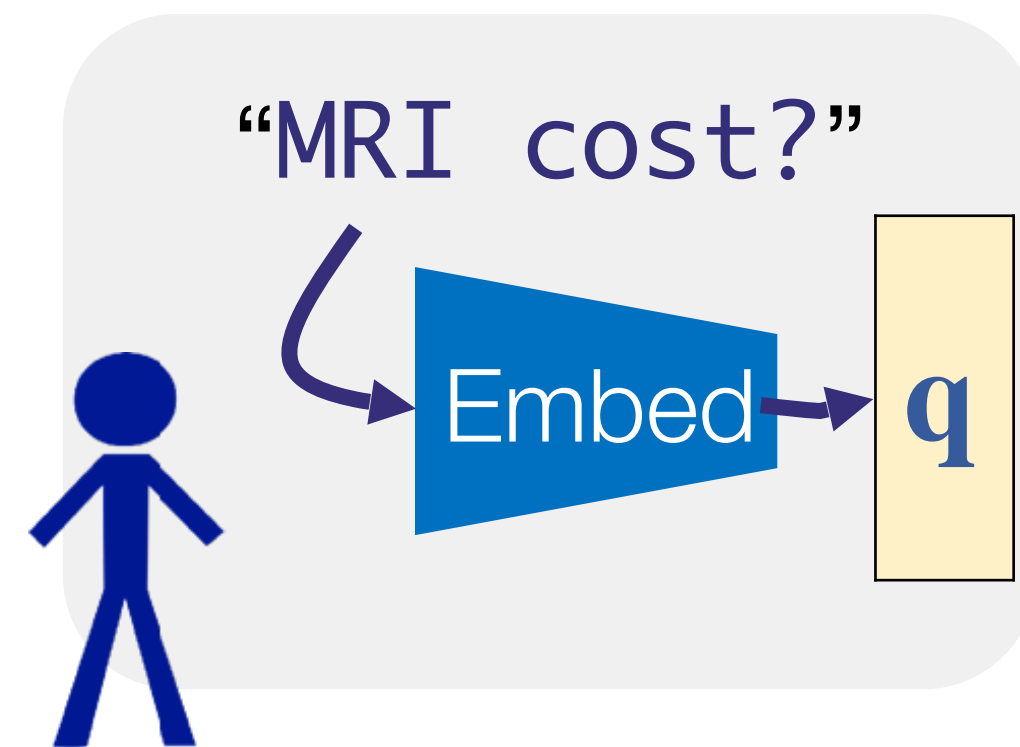




Tiptoe: Design in layers

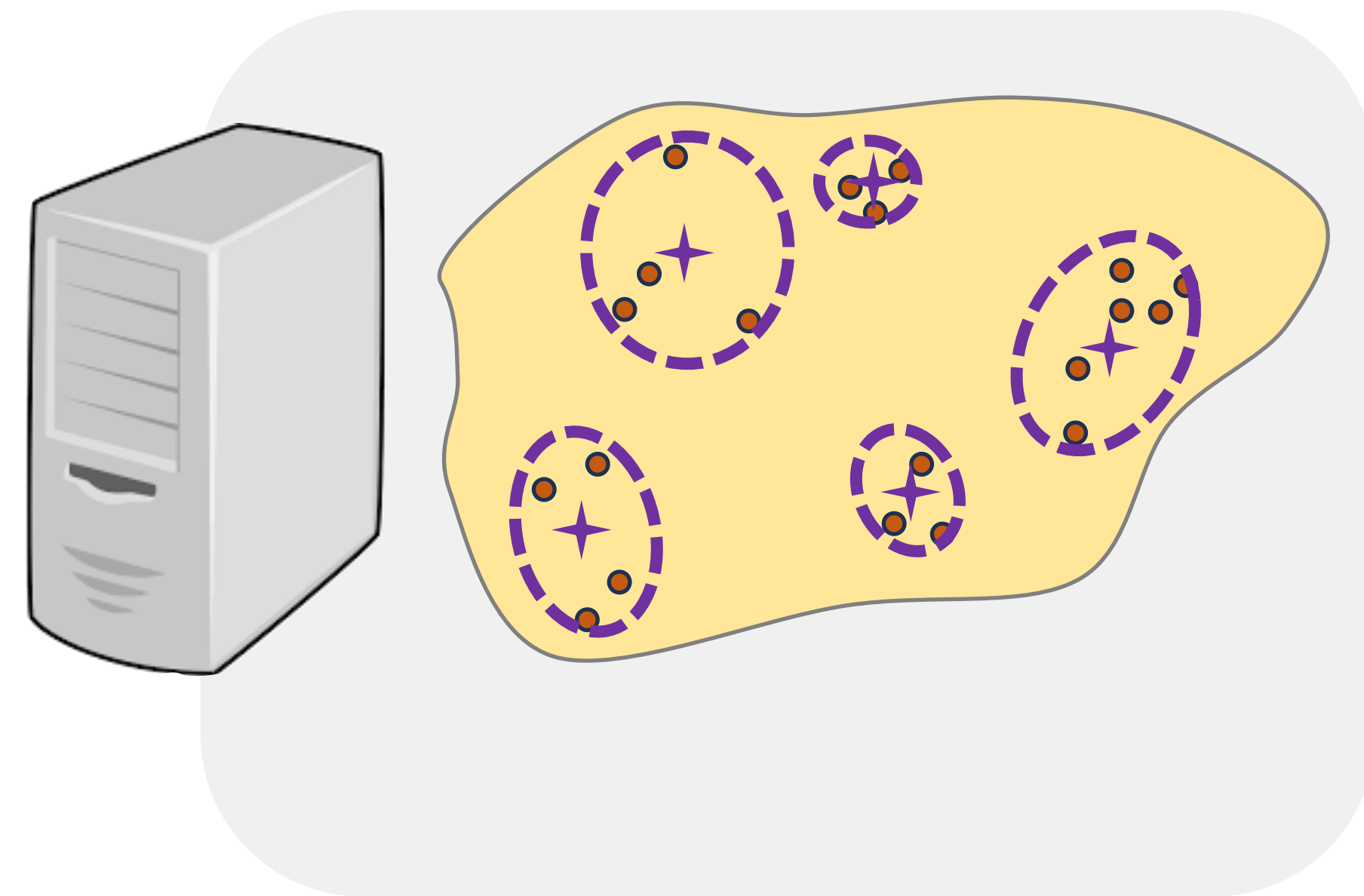
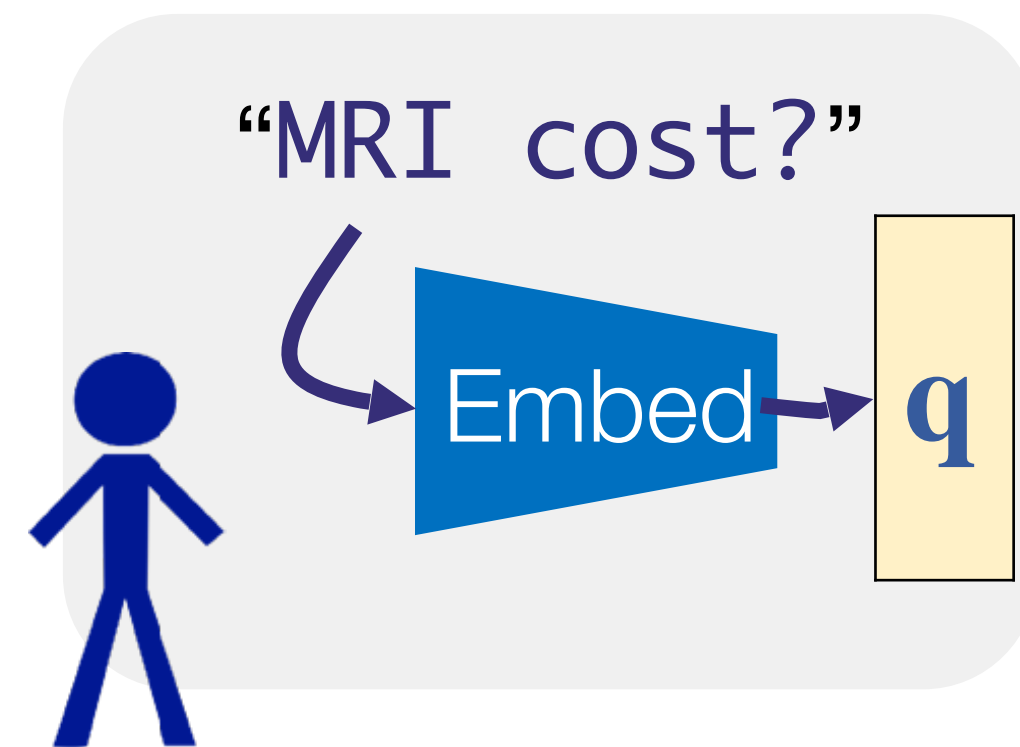


Goal: Private nearest-neighbor search



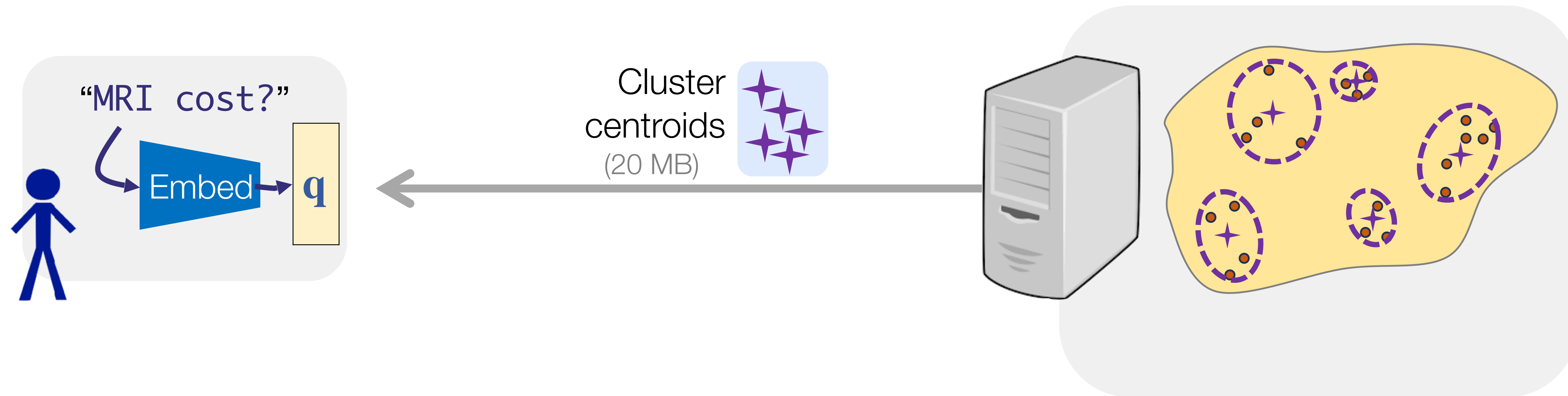
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



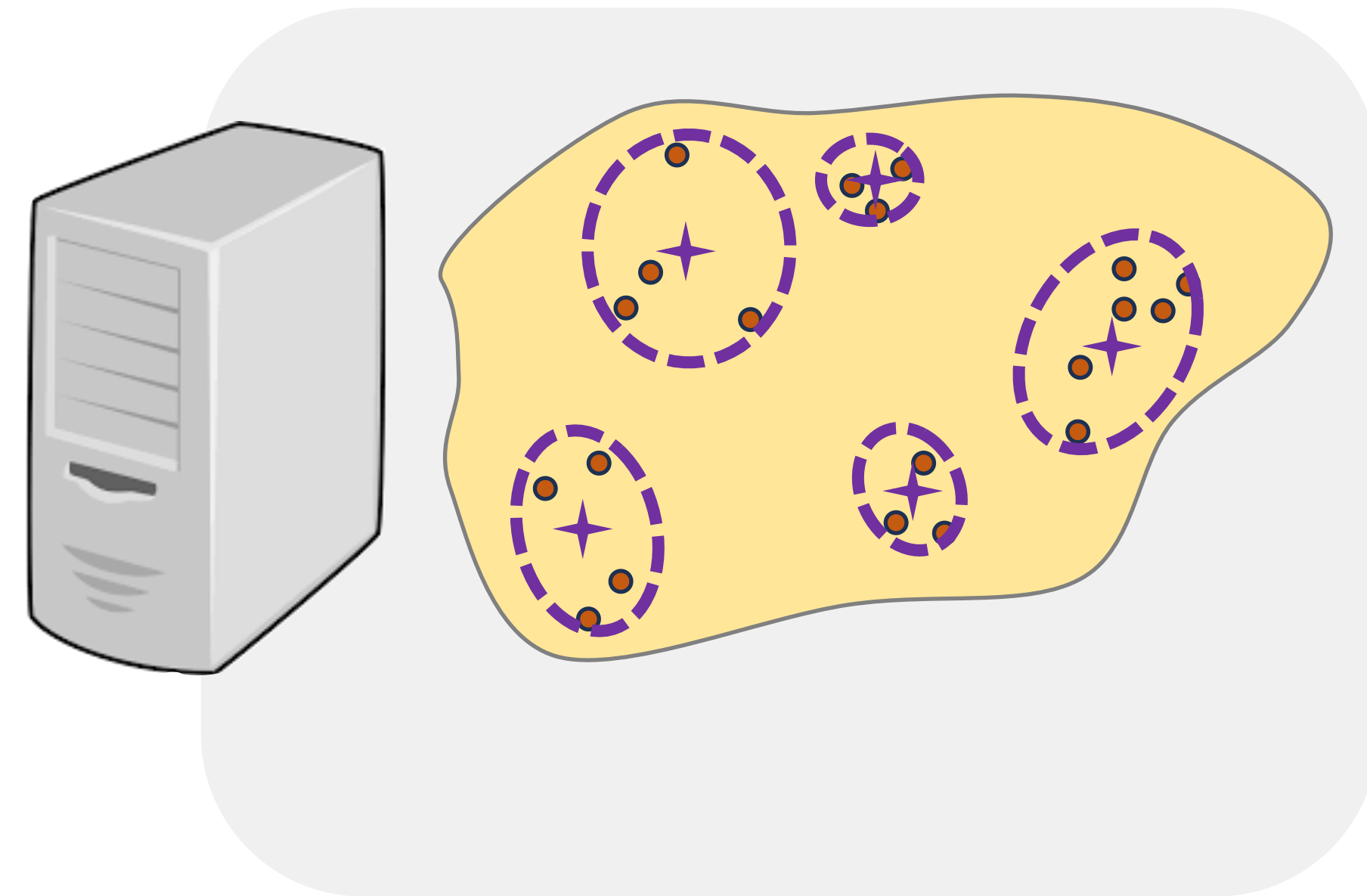
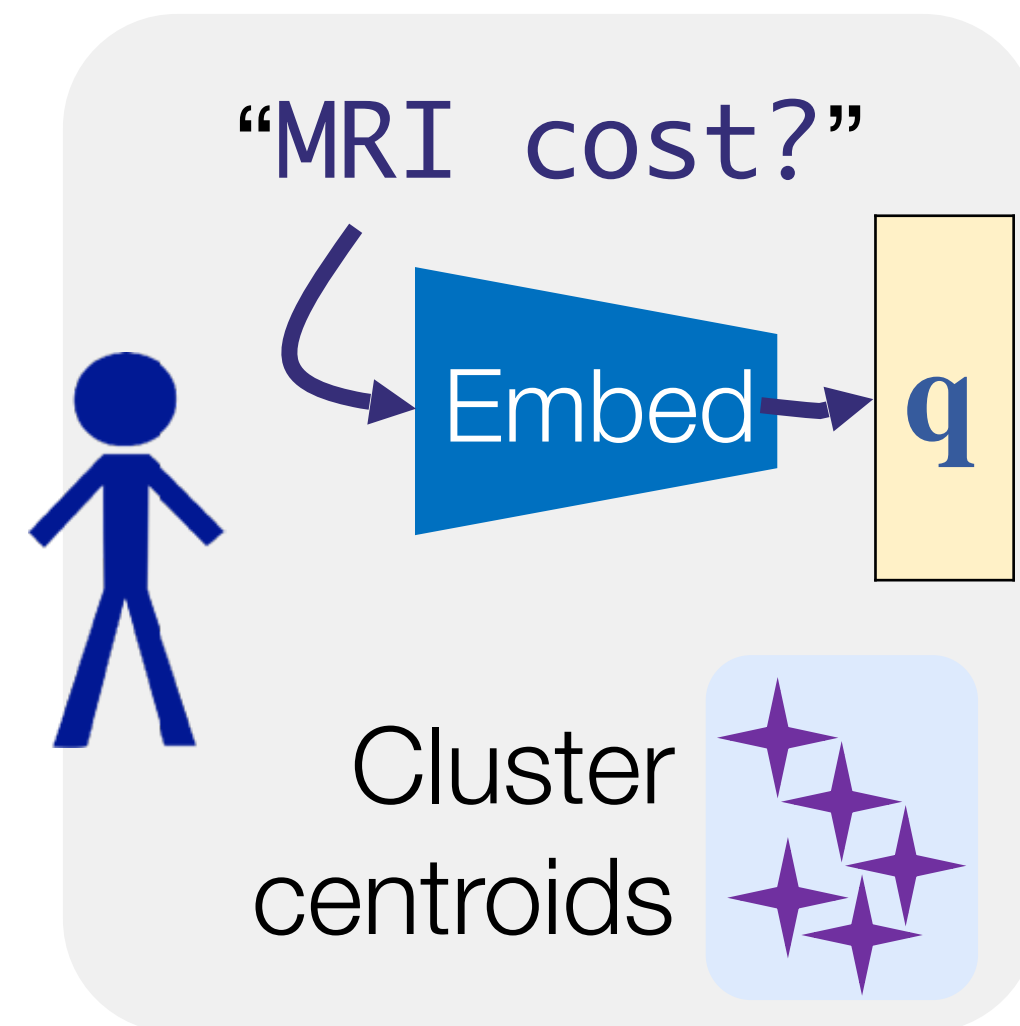
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



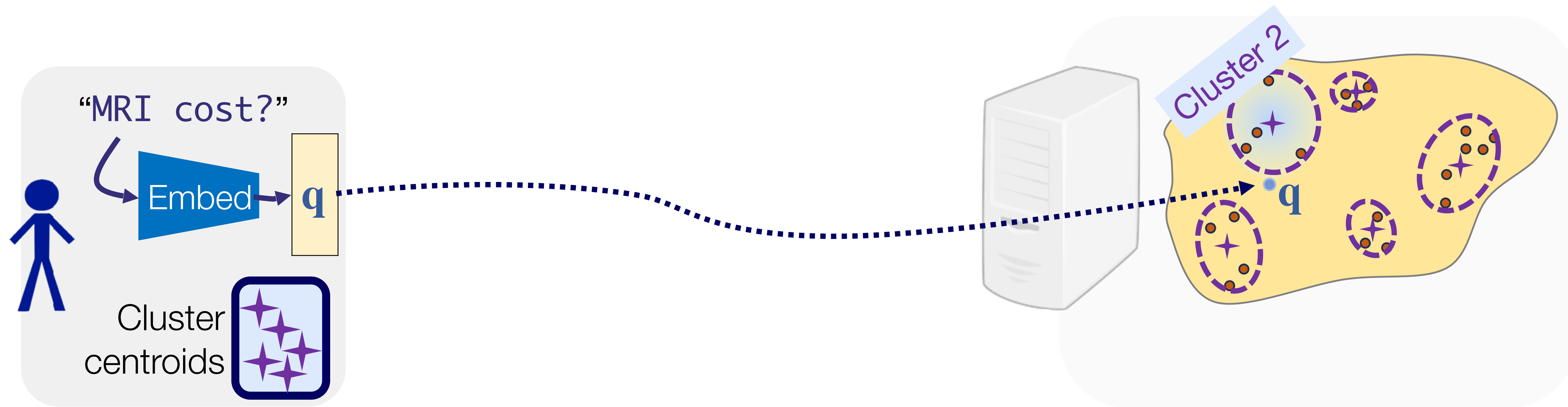
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



Goal: Private nearest-neighbor search

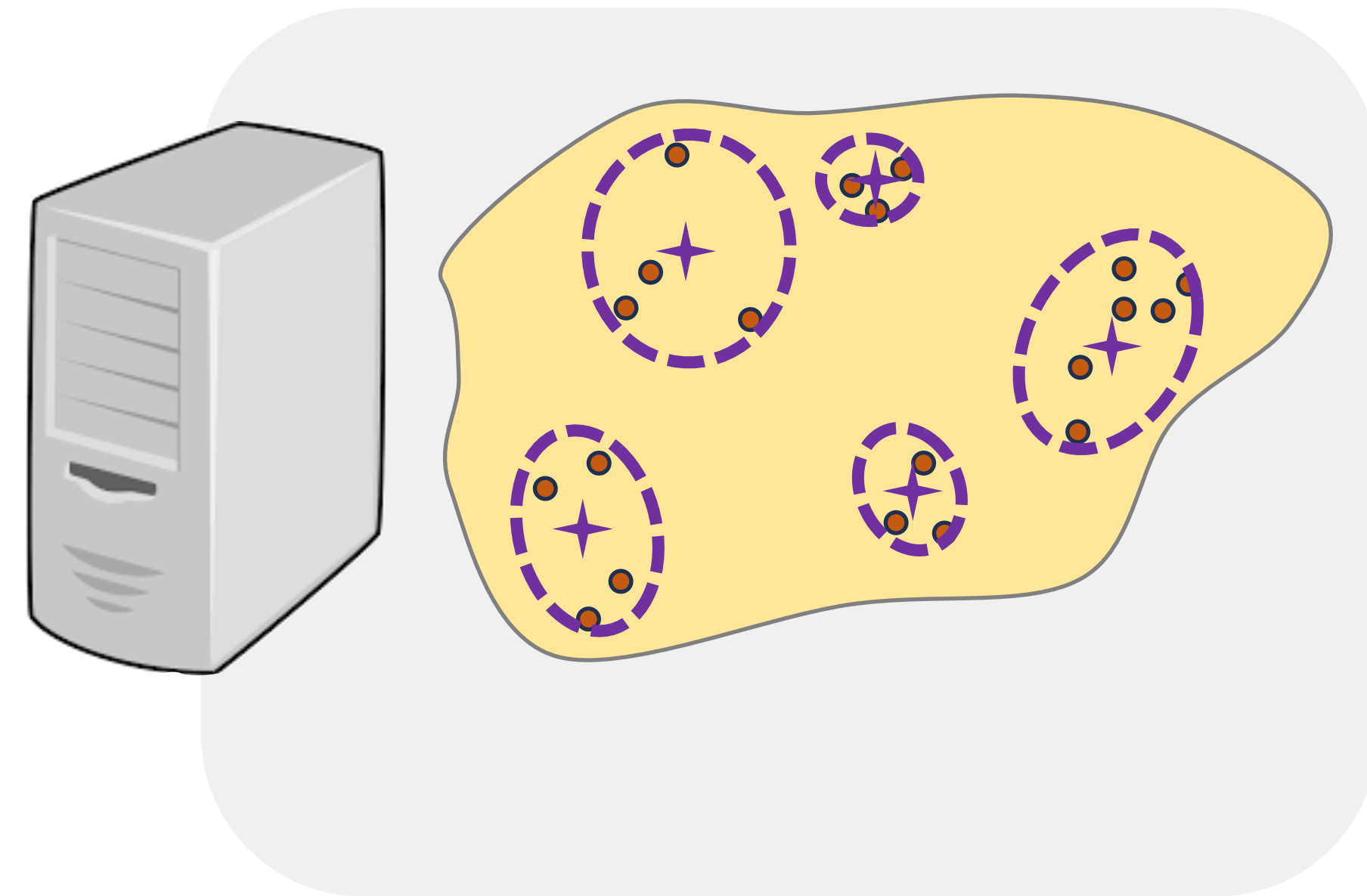
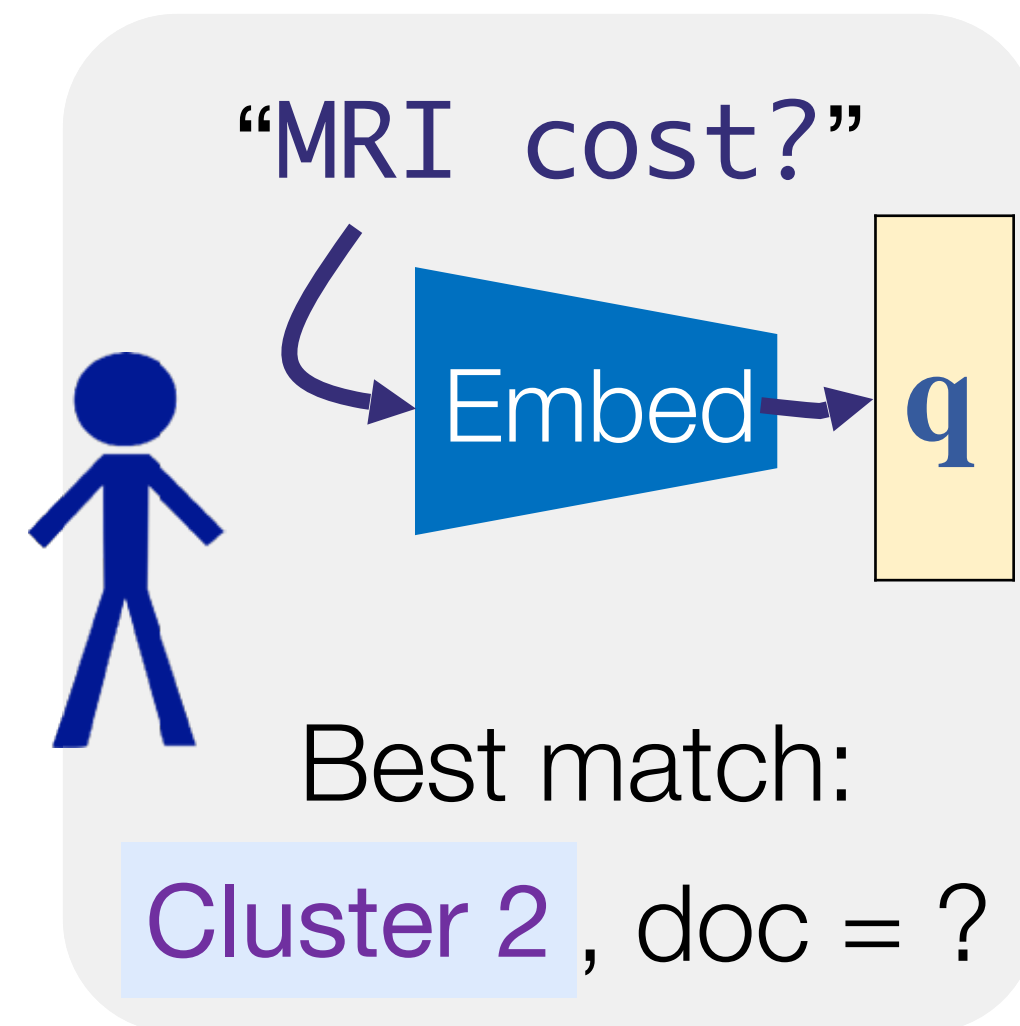
Plan: Structure the user's search with clustering



↳ Updated goal: exact search over Cluster 2
i.e., privately fetch inner-product scores $\langle \mathbf{q}, \mathbf{e} \rangle$ for docs in Cluster 2

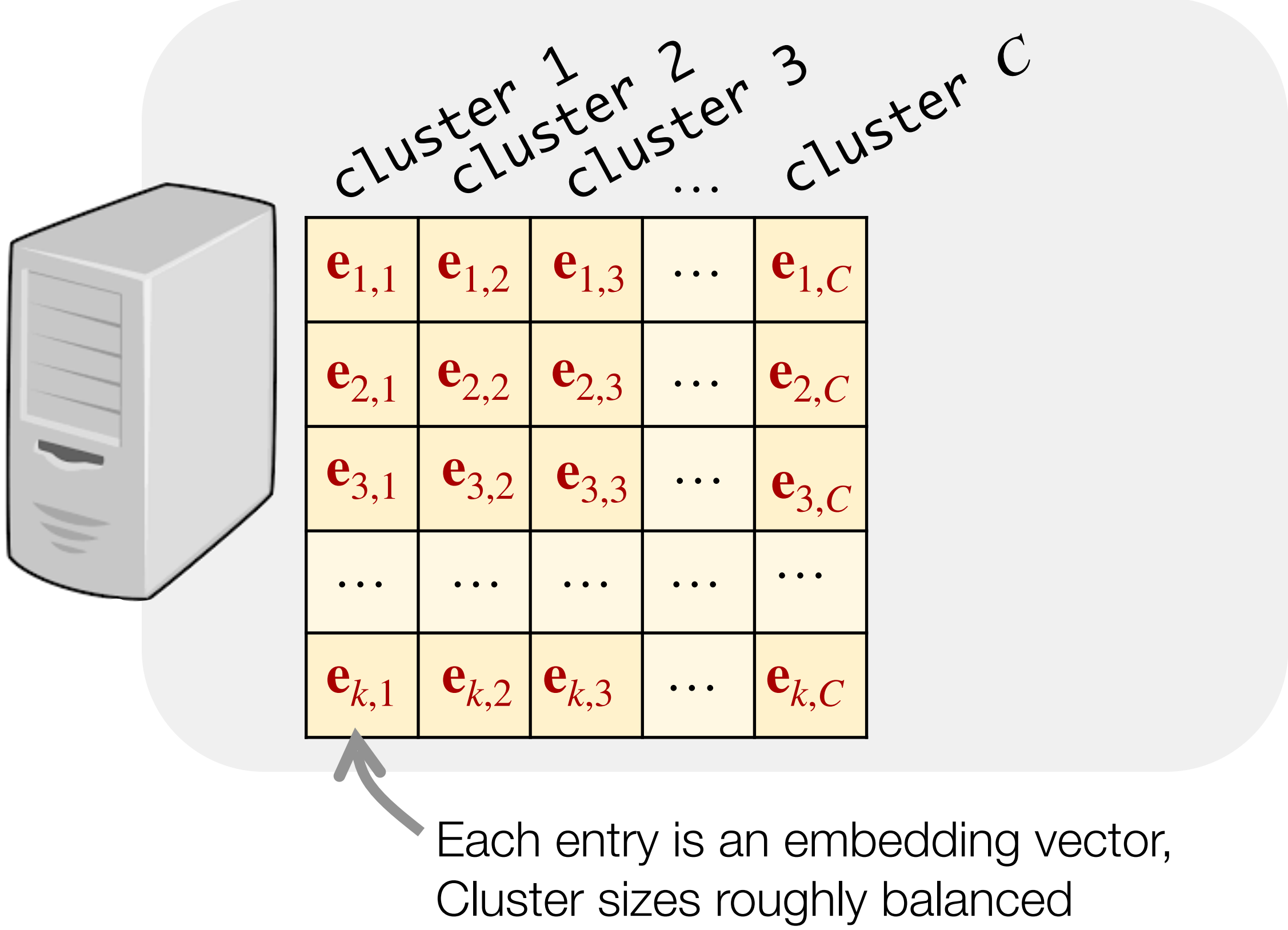
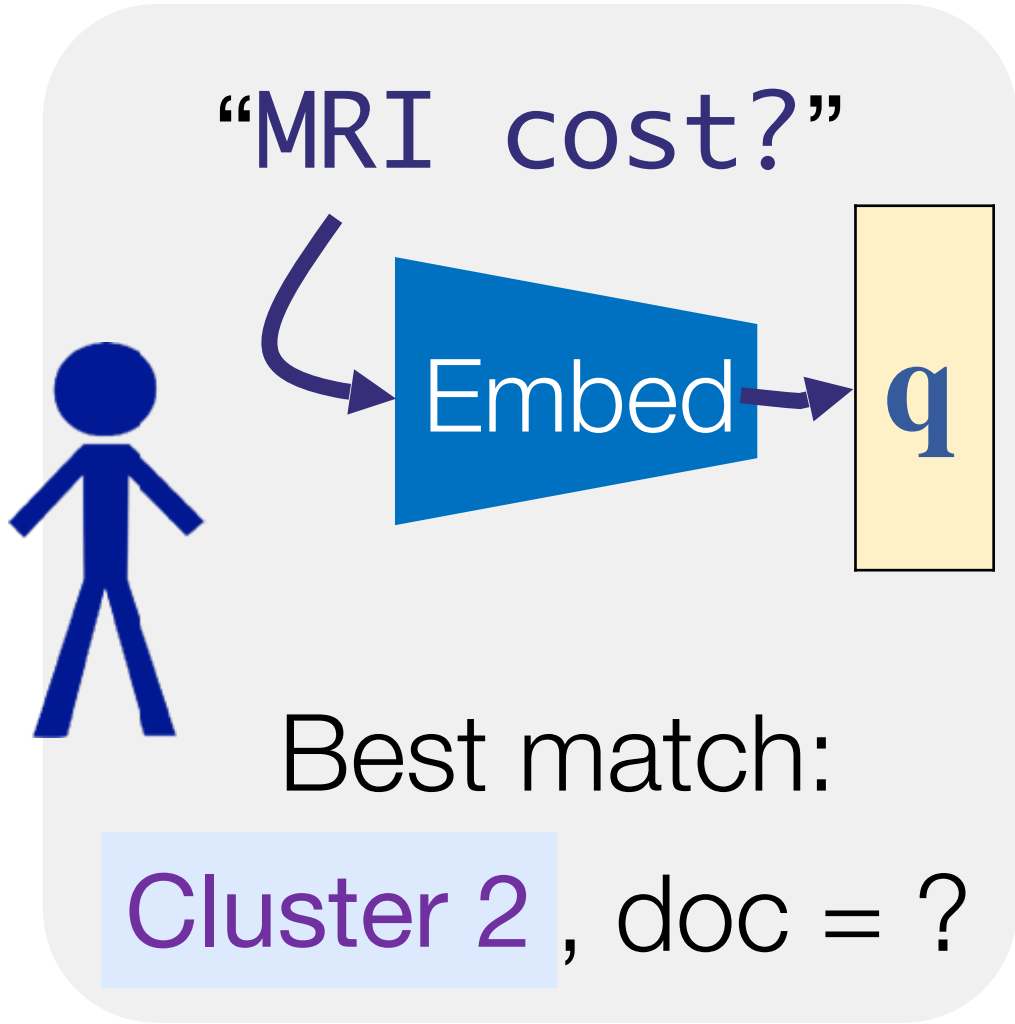
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



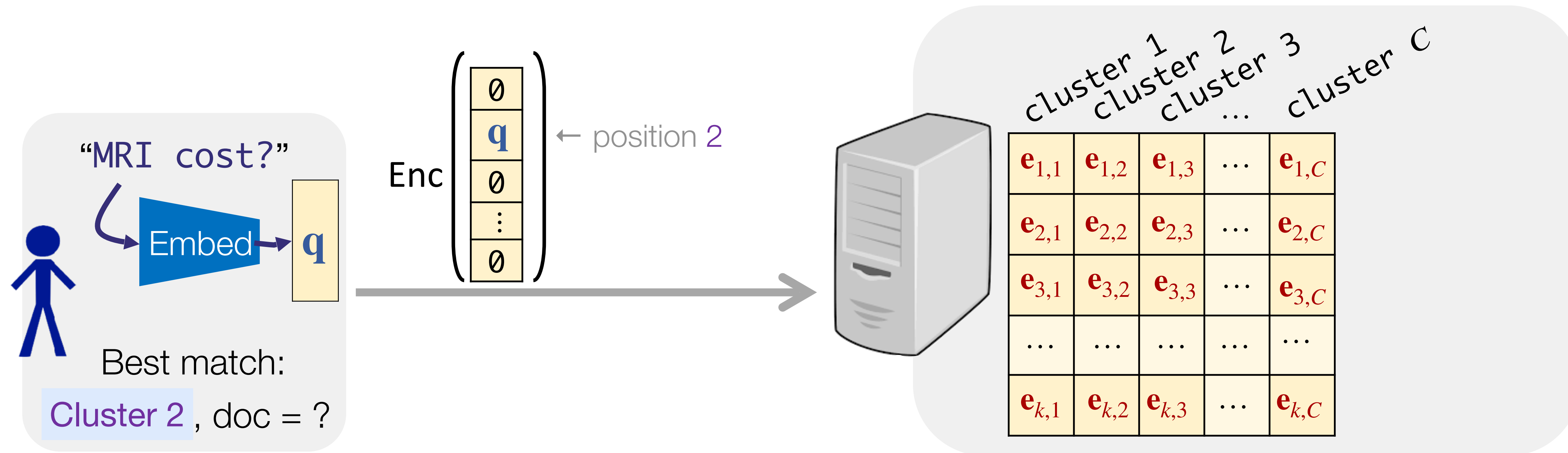
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



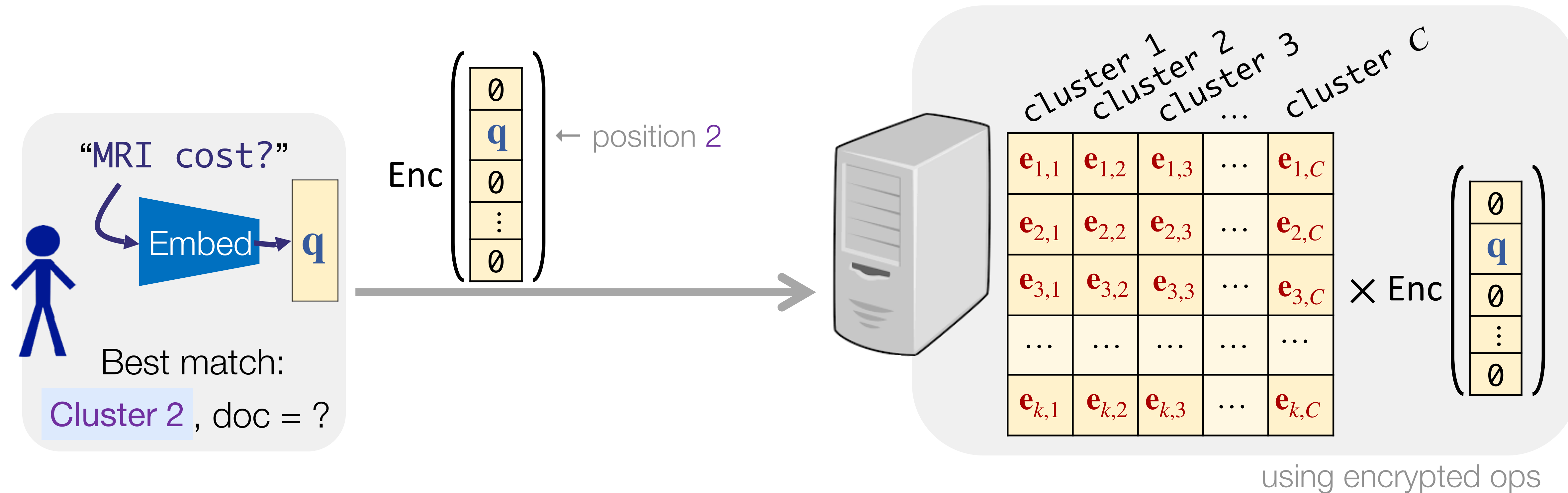
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



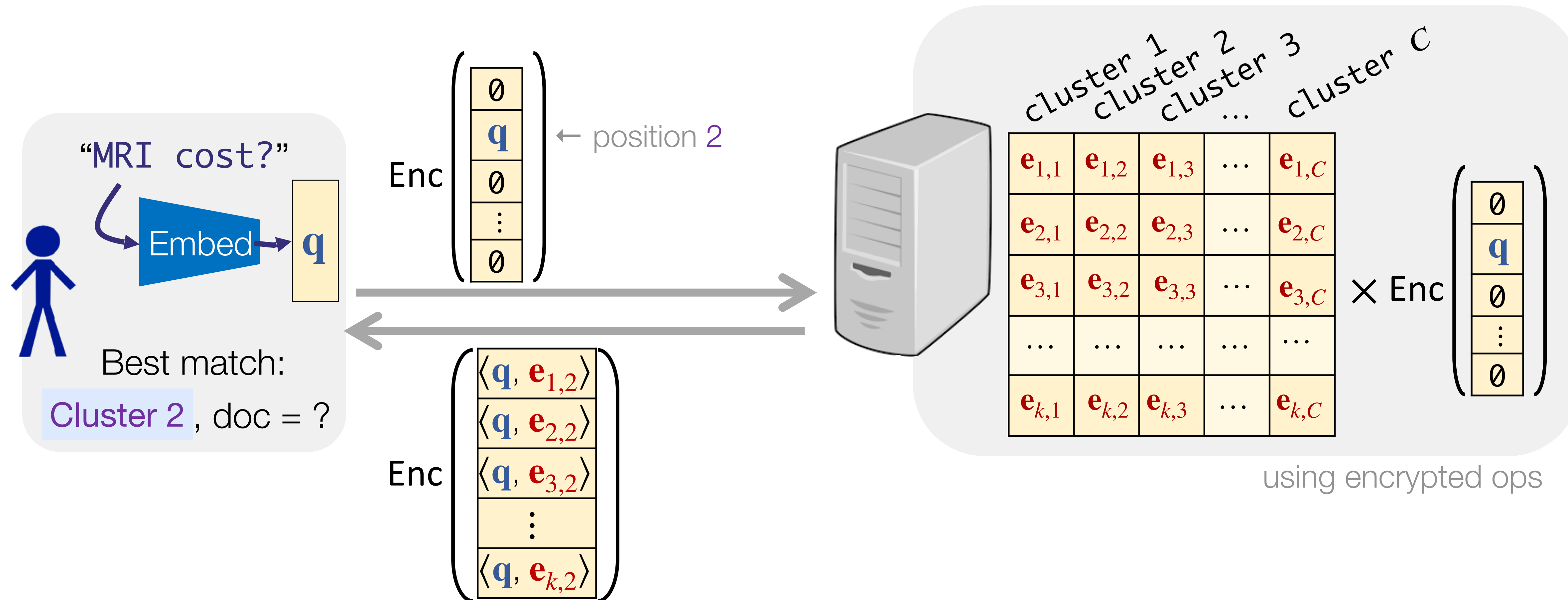
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



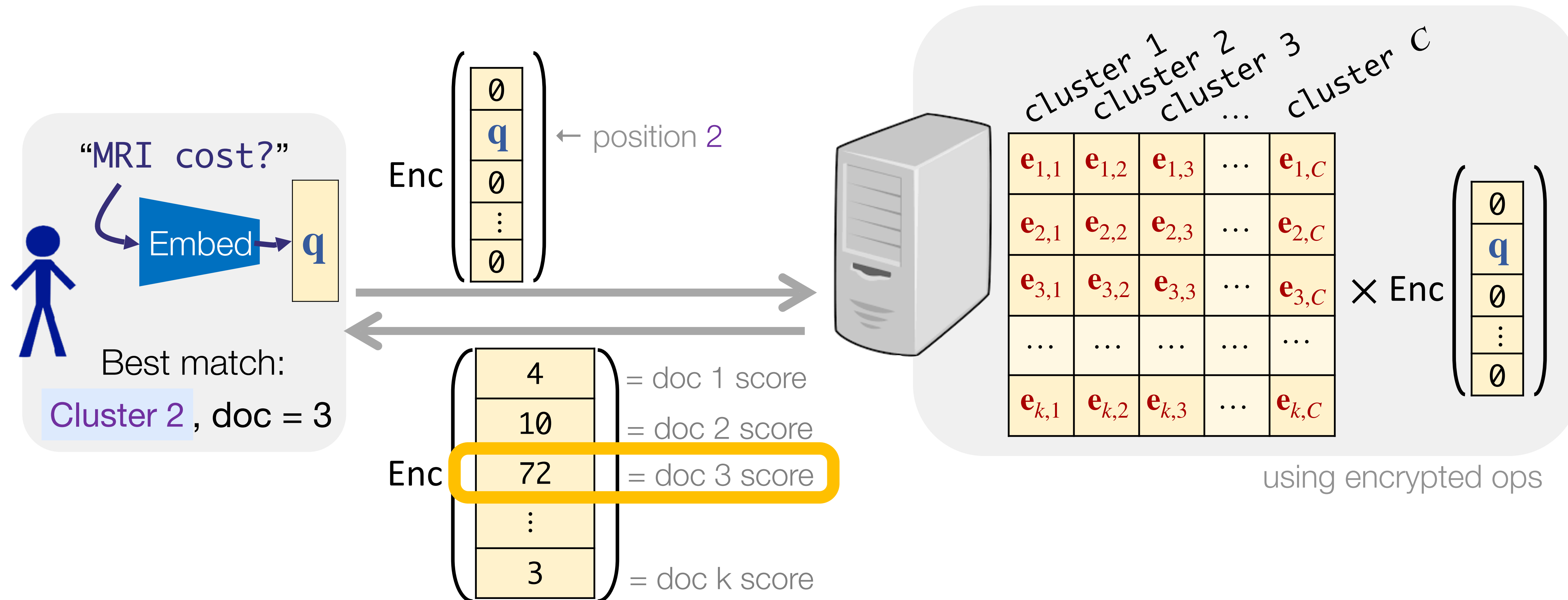
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



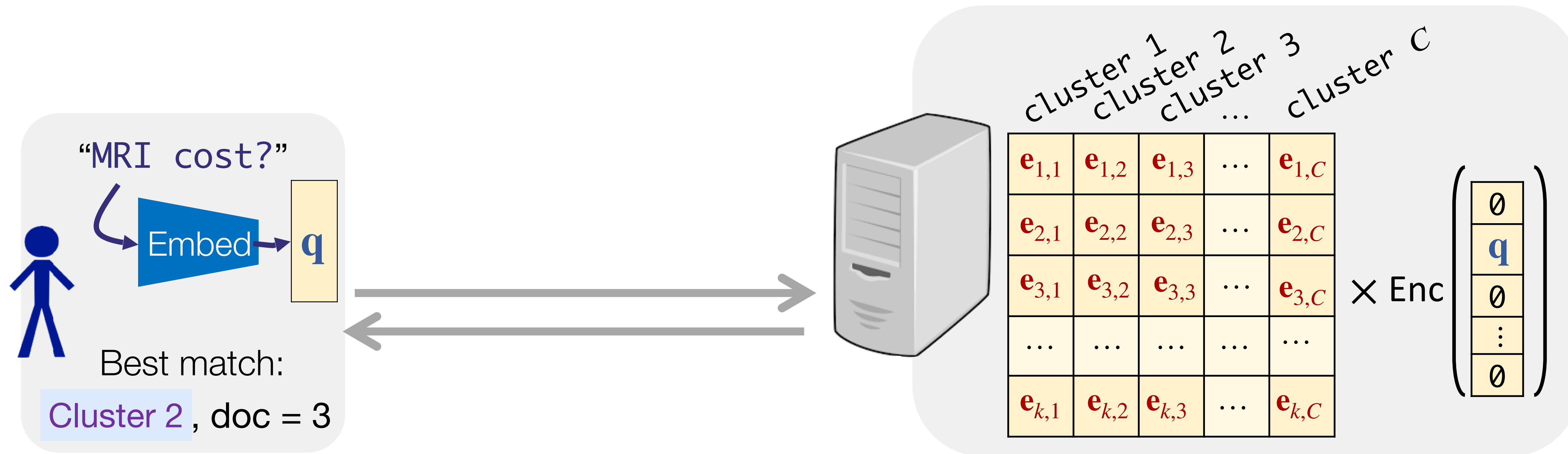
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering



Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering

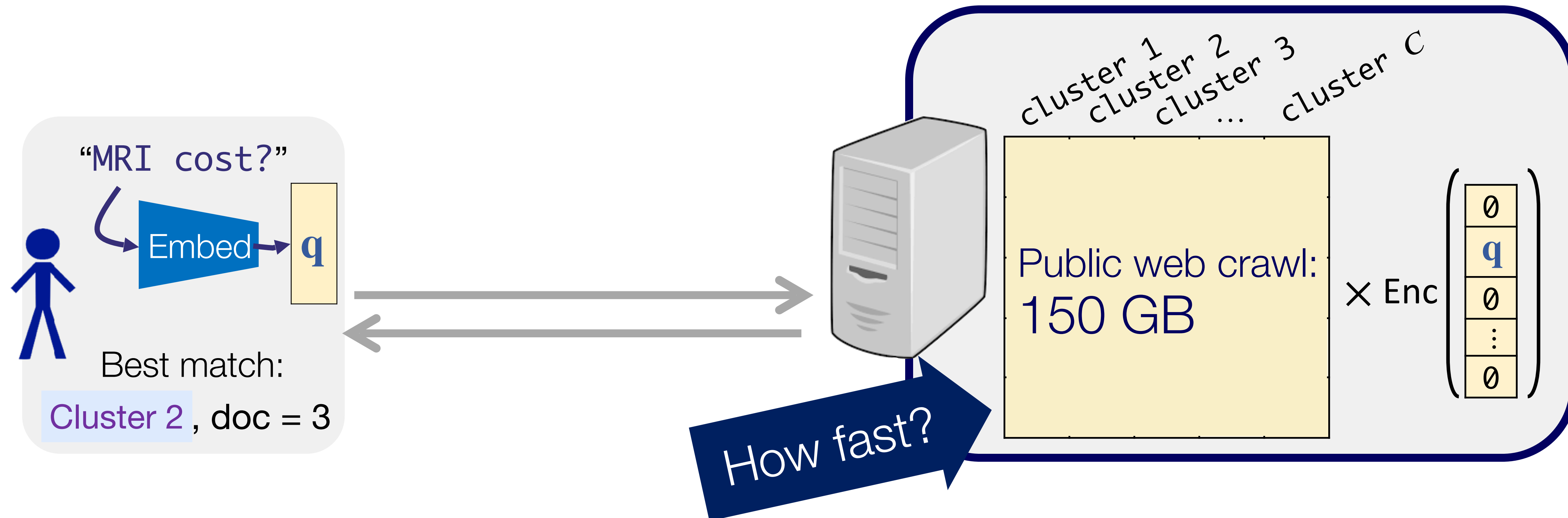


On every query,

- communication: $O(\sqrt{Nd})$ with N docs, dim- d embeddings, and $\sqrt{N/d}$ clusters

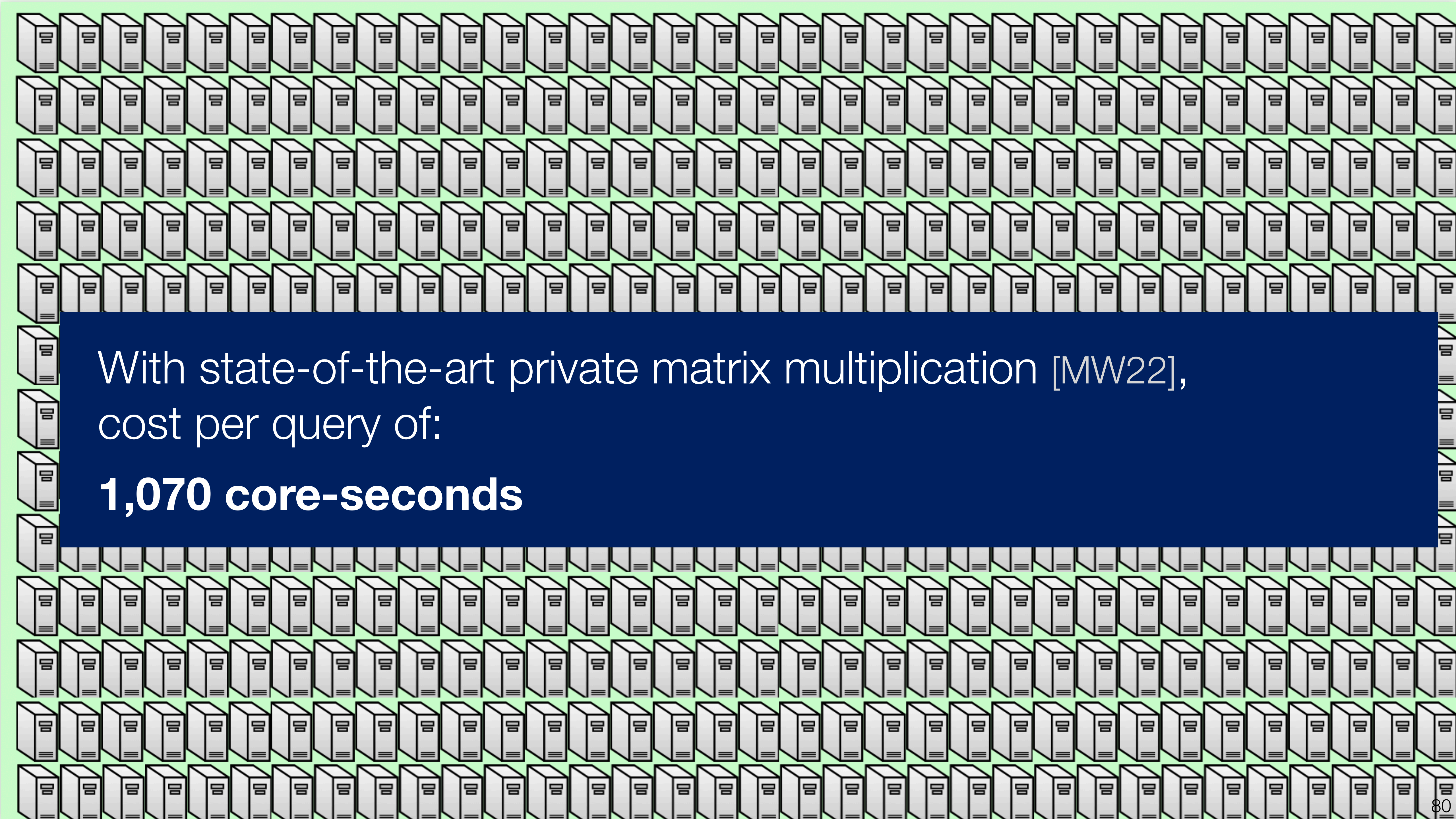
Goal: Private nearest-neighbor search

Plan: Structure the user's search with clustering

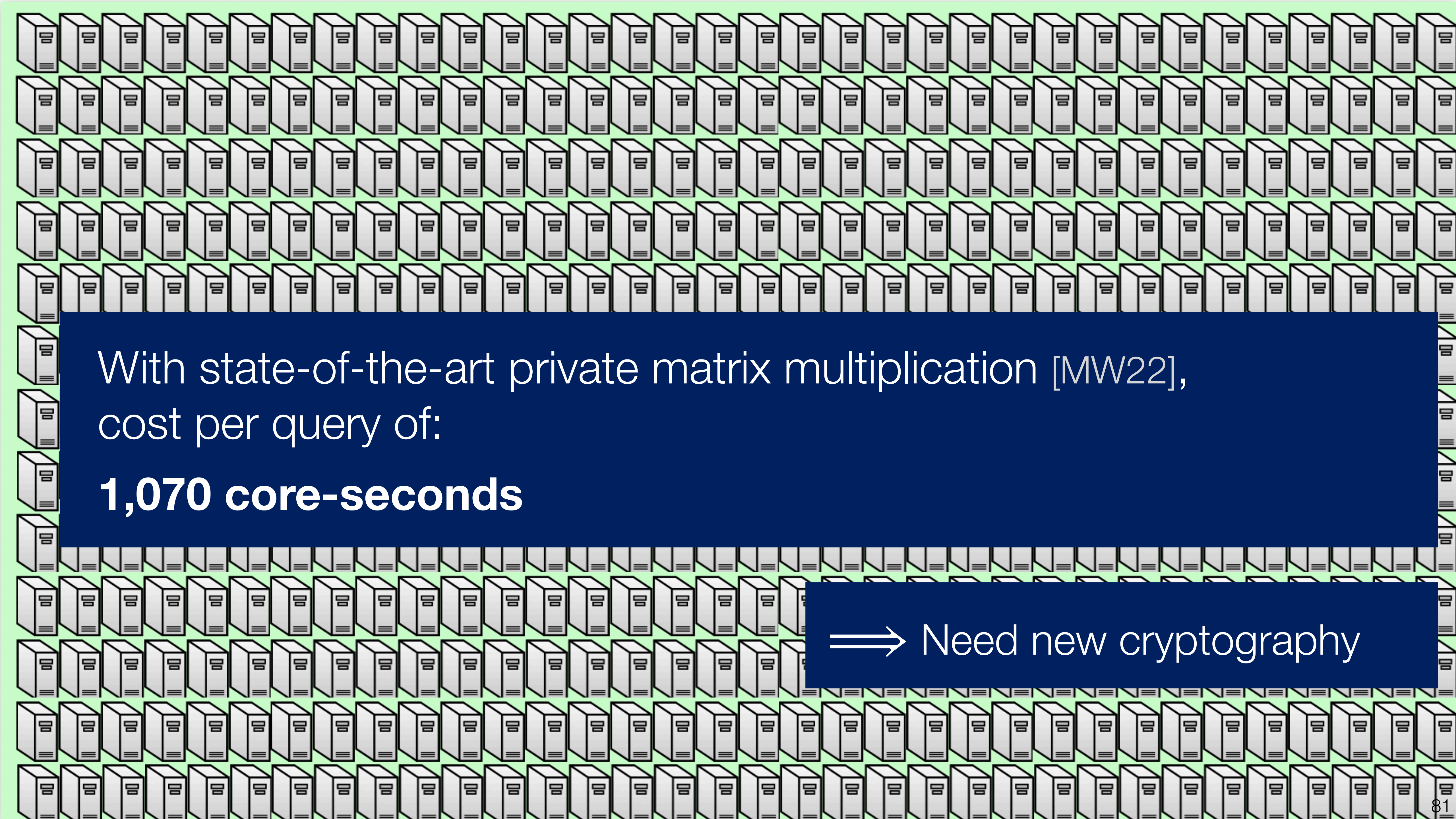


On every query,

- communication: $O(\sqrt{Nd})$ with N docs, dim- d embeddings, and $\sqrt{N/d}$ clusters
- computation: one large private matrix multiplication

The background of the slide is a repeating pattern of server racks. The racks are depicted in a light gray color with black outlines, arranged in a grid. The top half of the slide has a light green background, while the bottom half has a light blue background. A dark blue rectangular box is positioned in the center, containing white text.

With state-of-the-art private matrix multiplication [MW22],
cost per query of:
1,070 core-seconds

The background of the slide is a repeating pattern of server racks. Each rack is a light gray 3D box with a small black rectangle on its front face, representing a display or indicator. They are arranged in a grid, with some racks partially obscured by the text boxes.

With state-of-the-art private matrix multiplication [MW22],
cost per query of:

1,070 core-seconds

⇒ Need new cryptography



Tiptoe: Design in layers



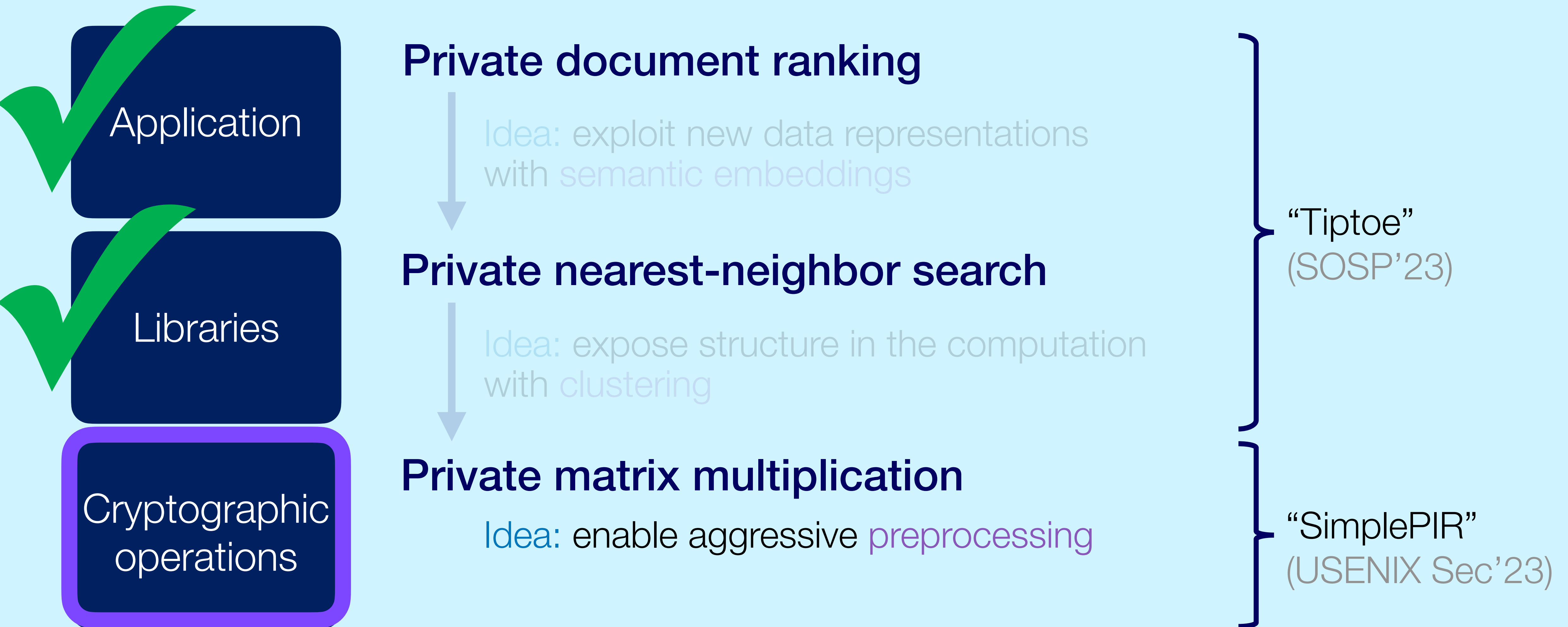


Tiptoe: Design in layers



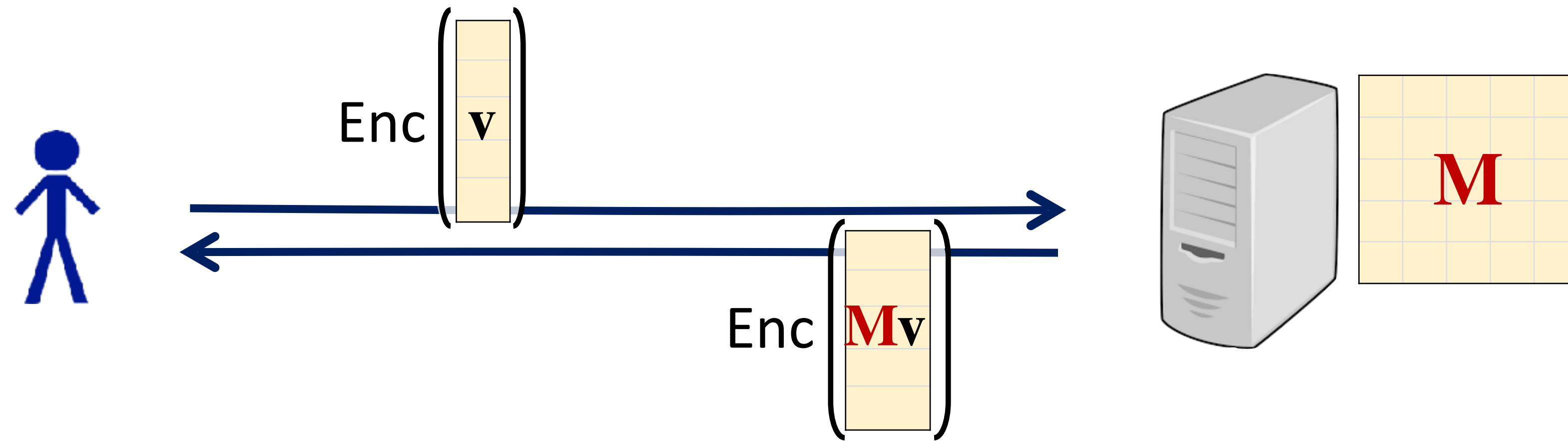


Tiptoe: Design in layers



SimplePIR: Fast private matrix multiplication

Henzinger, Hong, Corrigan-Gibbs, Meiklejohn, and Vaikuntanathan (USENIX Sec'23)



+ New encryption that supports preprocessing any matrix $\mathbf{M}$ for fast multiplication

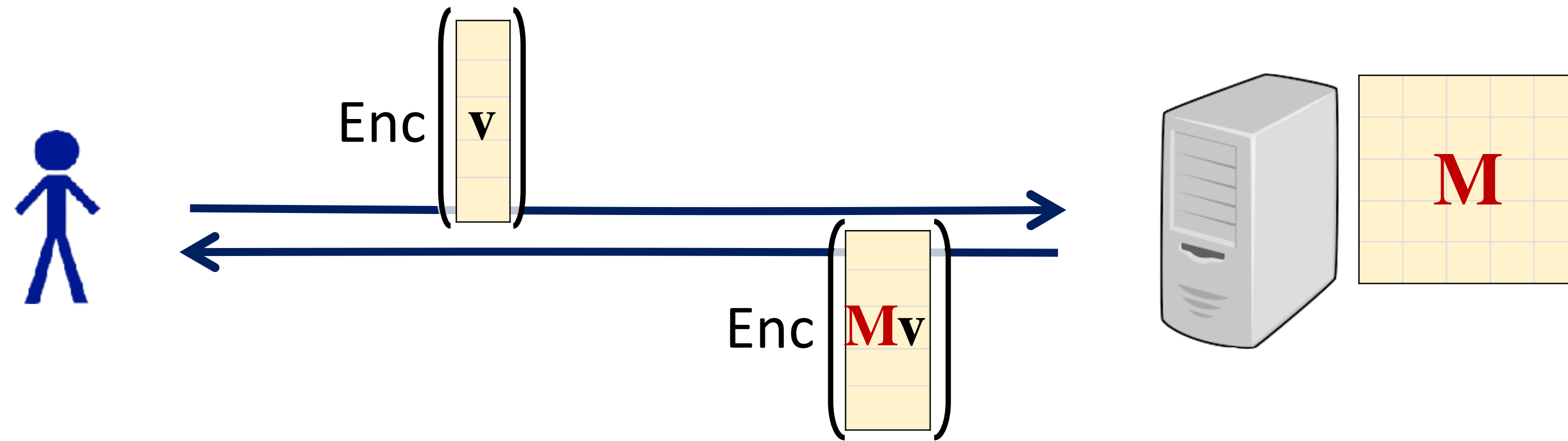
+ Per-query server work (on n -by- n matrix of 16-bit ints):

No privacy: n^2 16-bit adds and muls

Our scheme: n^2 64-bit adds and muls

SimplePIR: Fast private matrix multiplication

Henzinger, Hong, Corrigan-Gibbs, Meiklejohn, and Vaikuntanathan (USENIX Sec'23)



+ New encryption that supports preprocessing any matrix $\mathbf{M}$ for fast multiplication

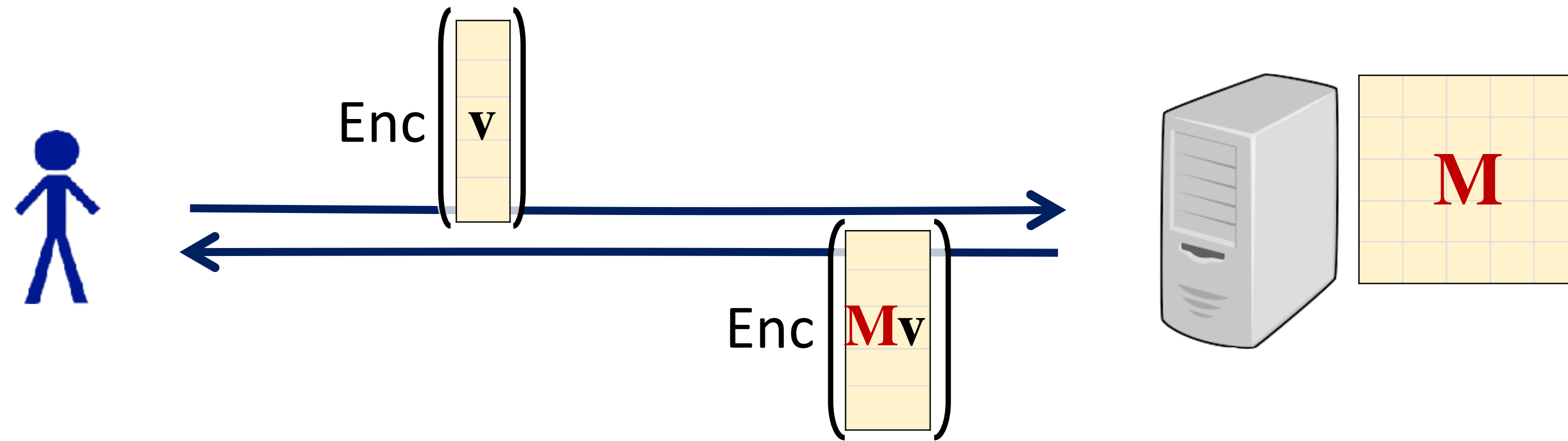
+ Per-query server work (on n -by- n matrix of 16-bit ints):

No privacy: n^2 16-bit adds and muls

Our scheme: n^2 64-bit adds and muls

SimplePIR: Fast private matrix multiplication

Henzinger, Hong, Corrigan-Gibbs, Meiklejohn, and Vaikuntanathan (USENIX Sec'23)



+ New encryption that supports preprocessing any matrix $\mathbf{M}$ for fast multiplication

+ Per-query server work (on n -by- n matrix of 16-bit ints):

No privacy: n^2 16-bit adds and muls

→ throughput: 12 GB / core-s

Our scheme: n^2 64-bit adds and muls

→ throughput: 10 GB / core-s

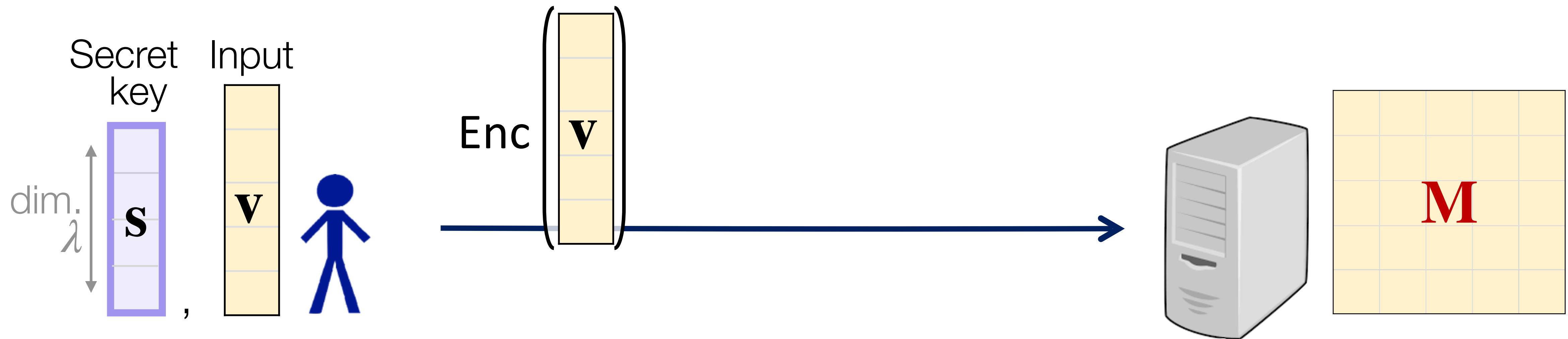
Background: Encryption hides data in a hard problem with a secret key

Hard problem: “Learning with errors” [Reg’09]



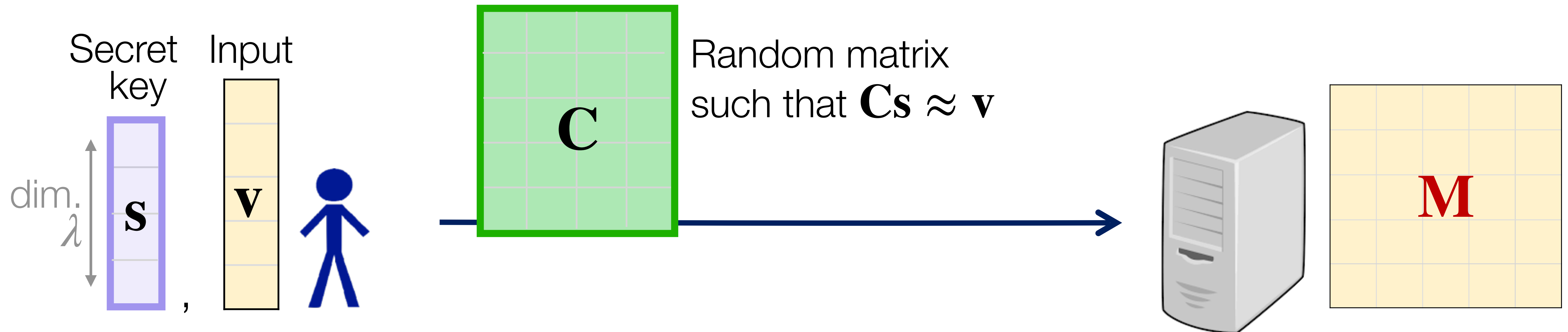
Background: Encryption hides data in a hard problem with a secret key

Hard problem: “Learning with errors” [Reg’09]



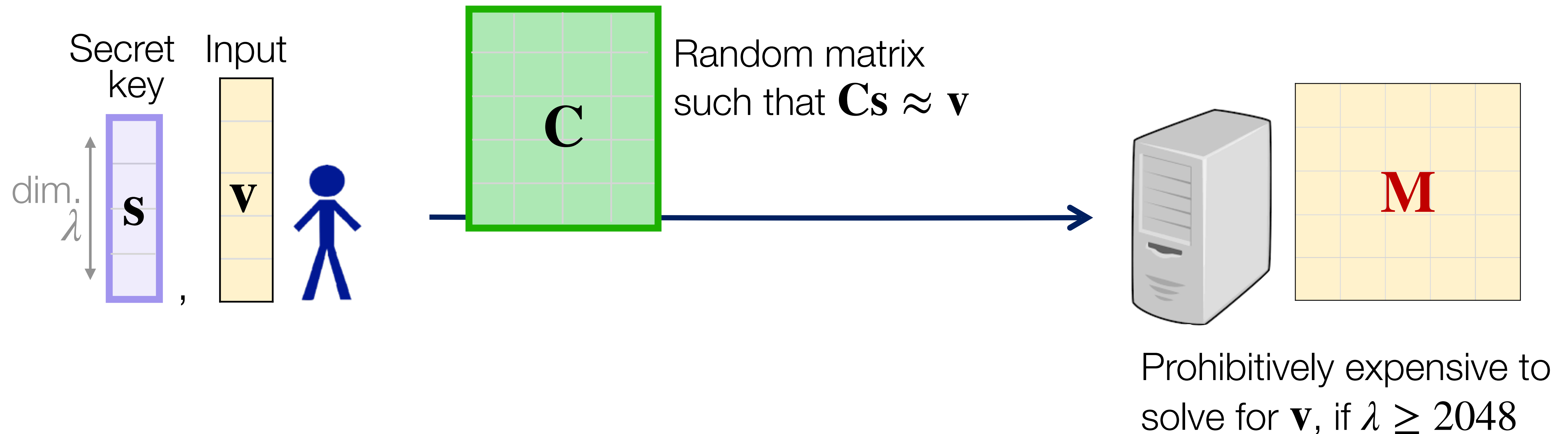
Background: Encryption hides data in a hard problem with a secret key

Hard problem: “Learning with errors” [Reg’09]



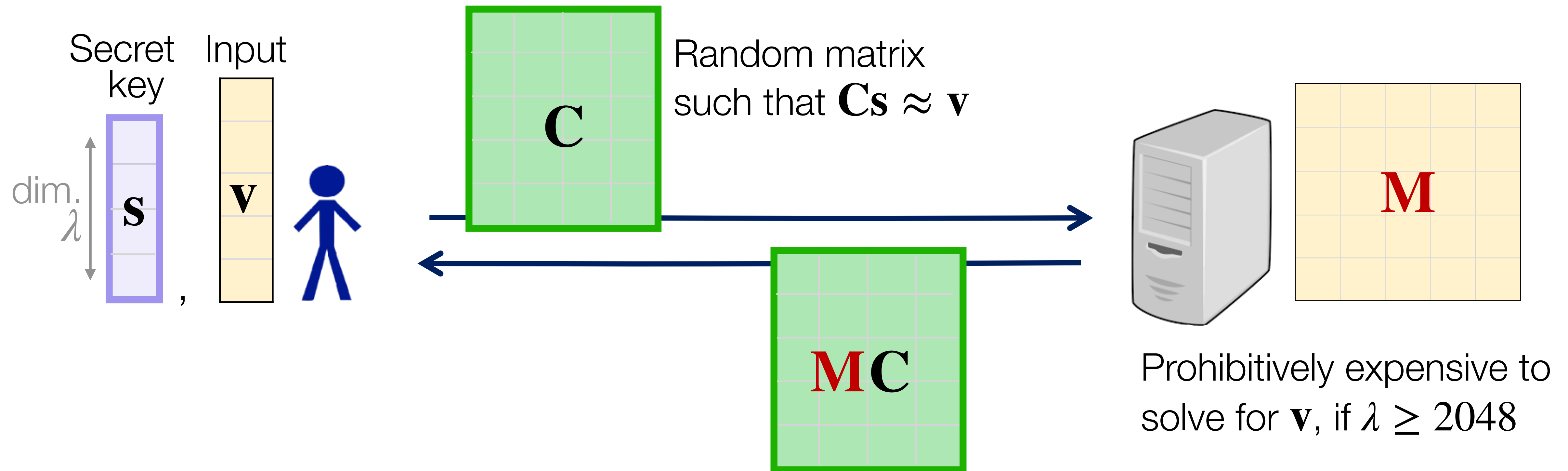
Background: Encryption hides data in a hard problem with a secret key

Hard problem: “Learning with errors” [Reg’09]



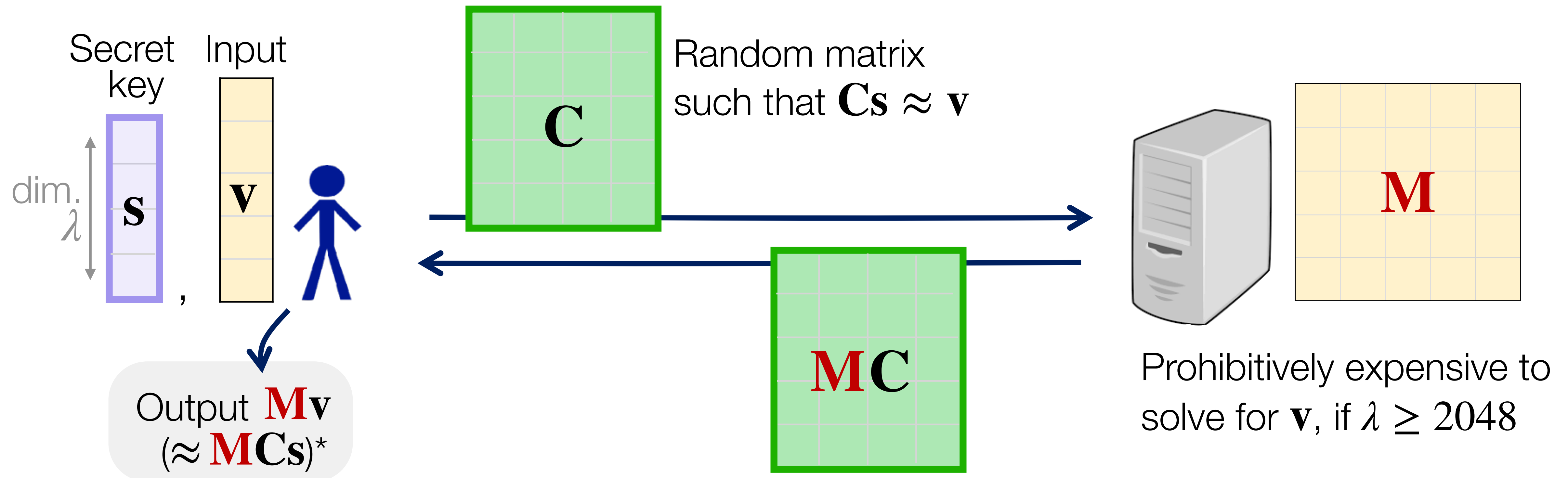
Background: Encryption hides data in a hard problem with a secret key

Hard problem: “Learning with errors” [Reg’09]



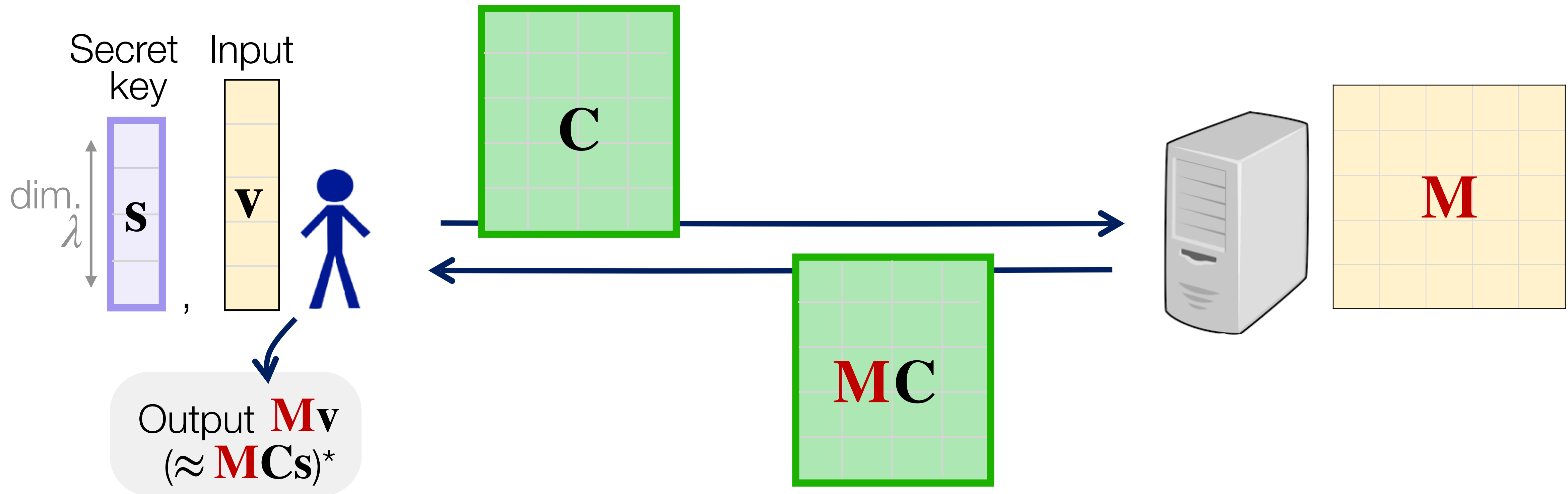
Background: Encryption hides data in a hard problem with a secret key

Hard problem: “Learning with errors” [Reg’09]



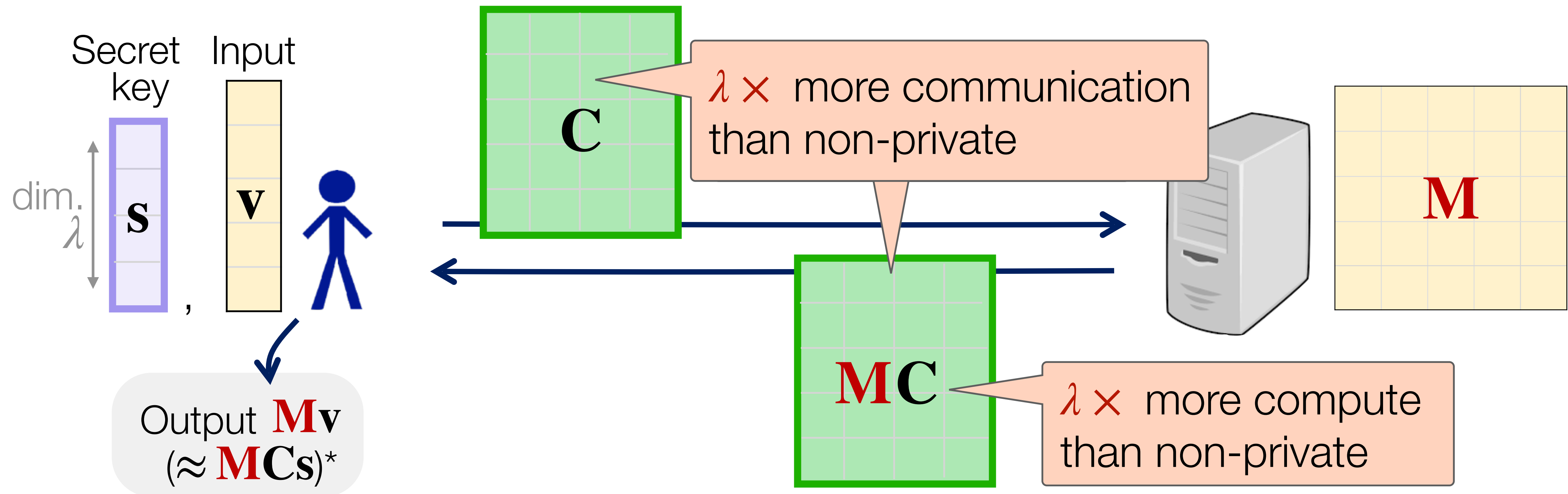
Result: Private matrix multiplication ... at high cost

For security: $\lambda \geq 2048$

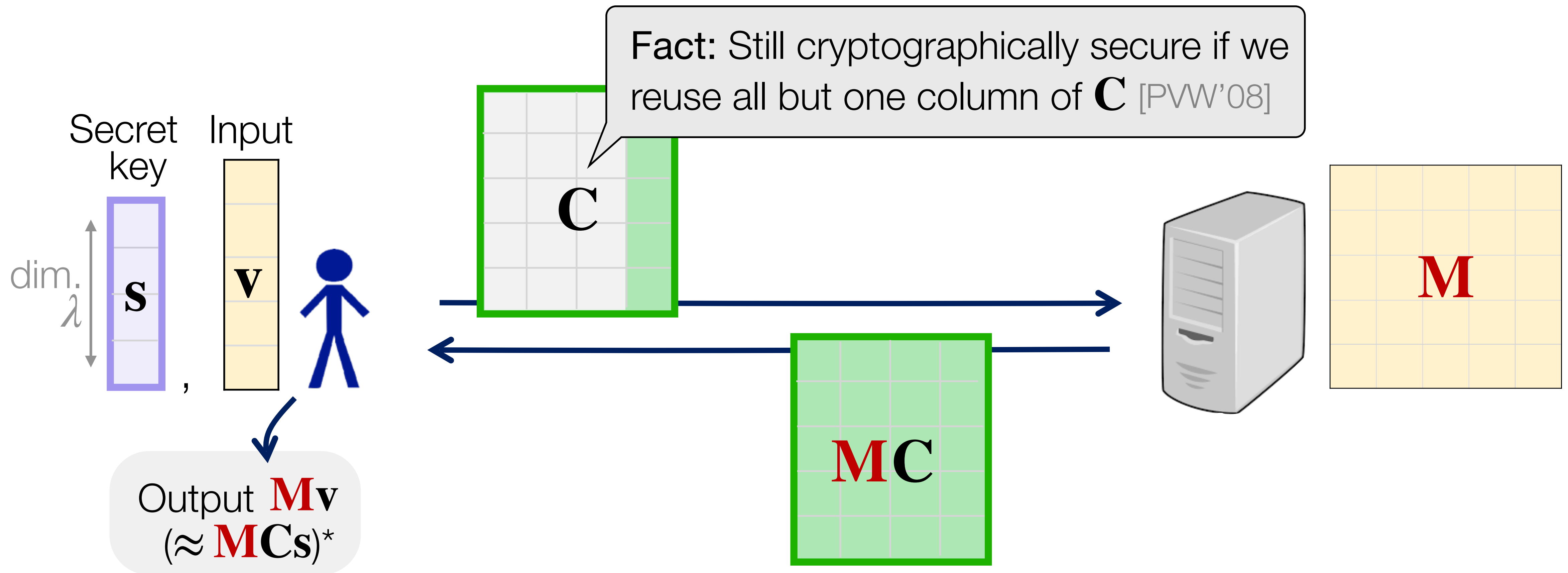


Result: Private matrix multiplication ... at high cost

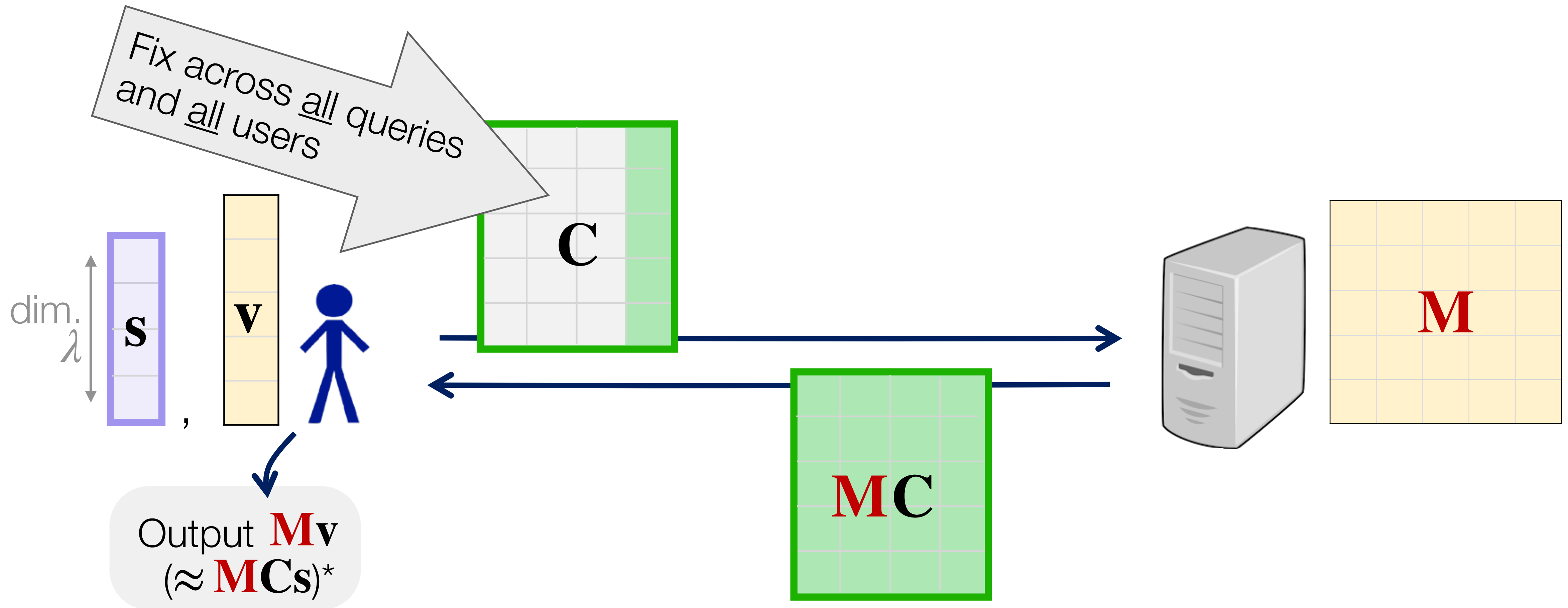
For security: $\lambda \geq 2048$



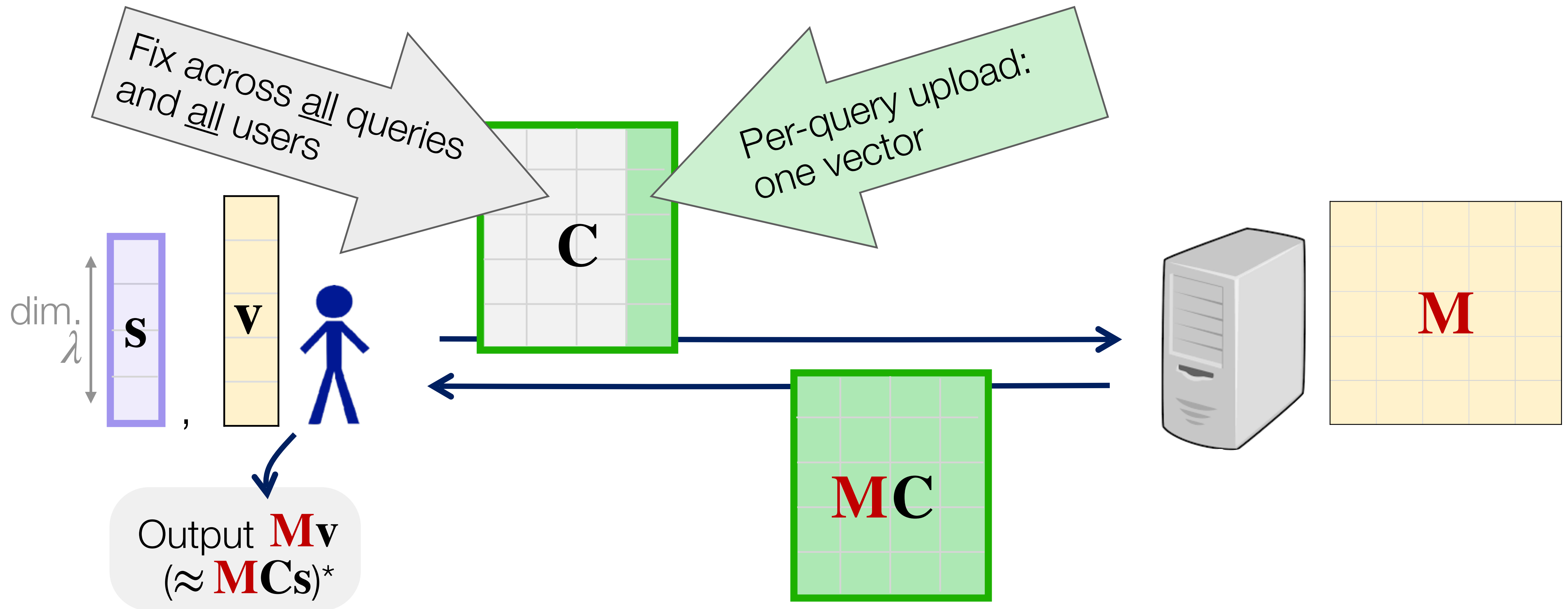
Idea: Expose useful structure for preprocessing



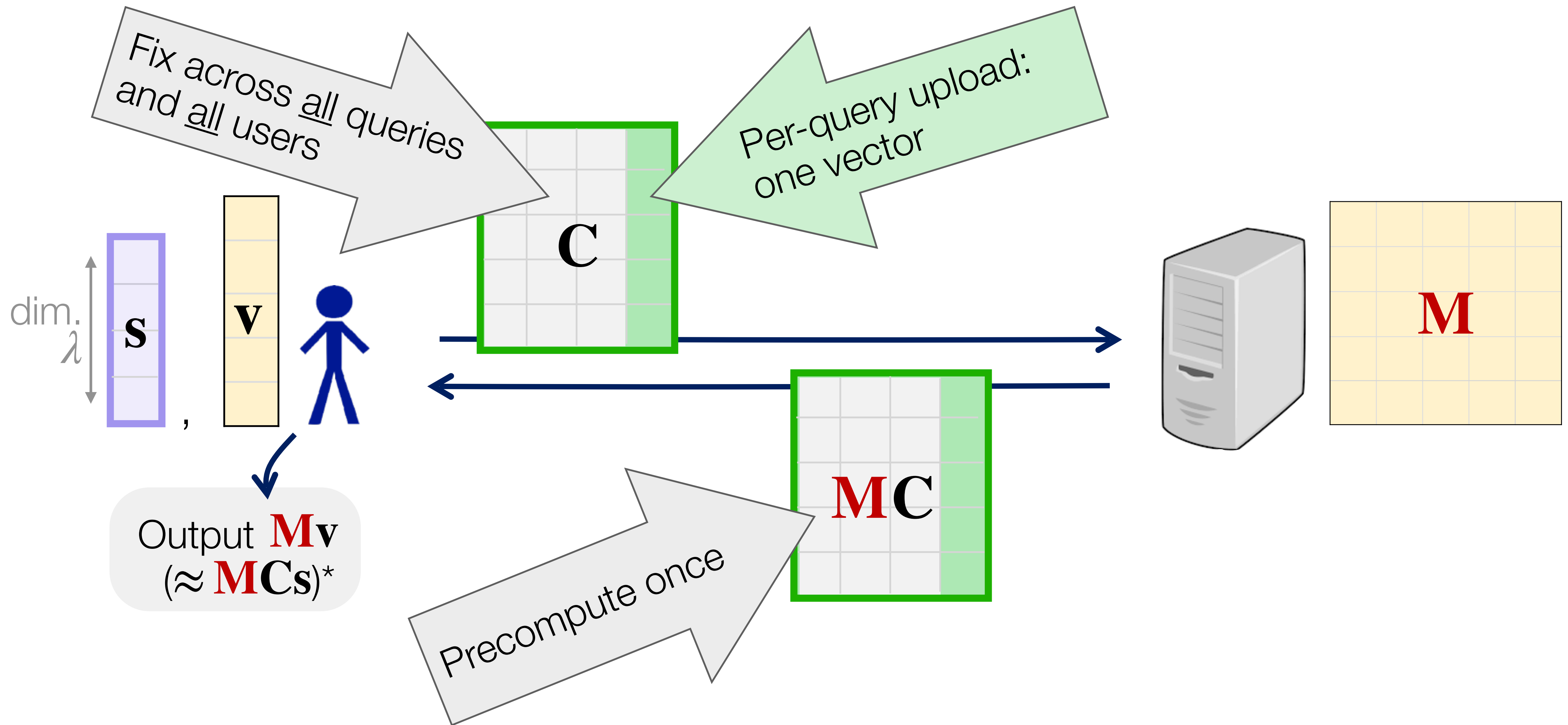
Idea: Expose useful structure for preprocessing



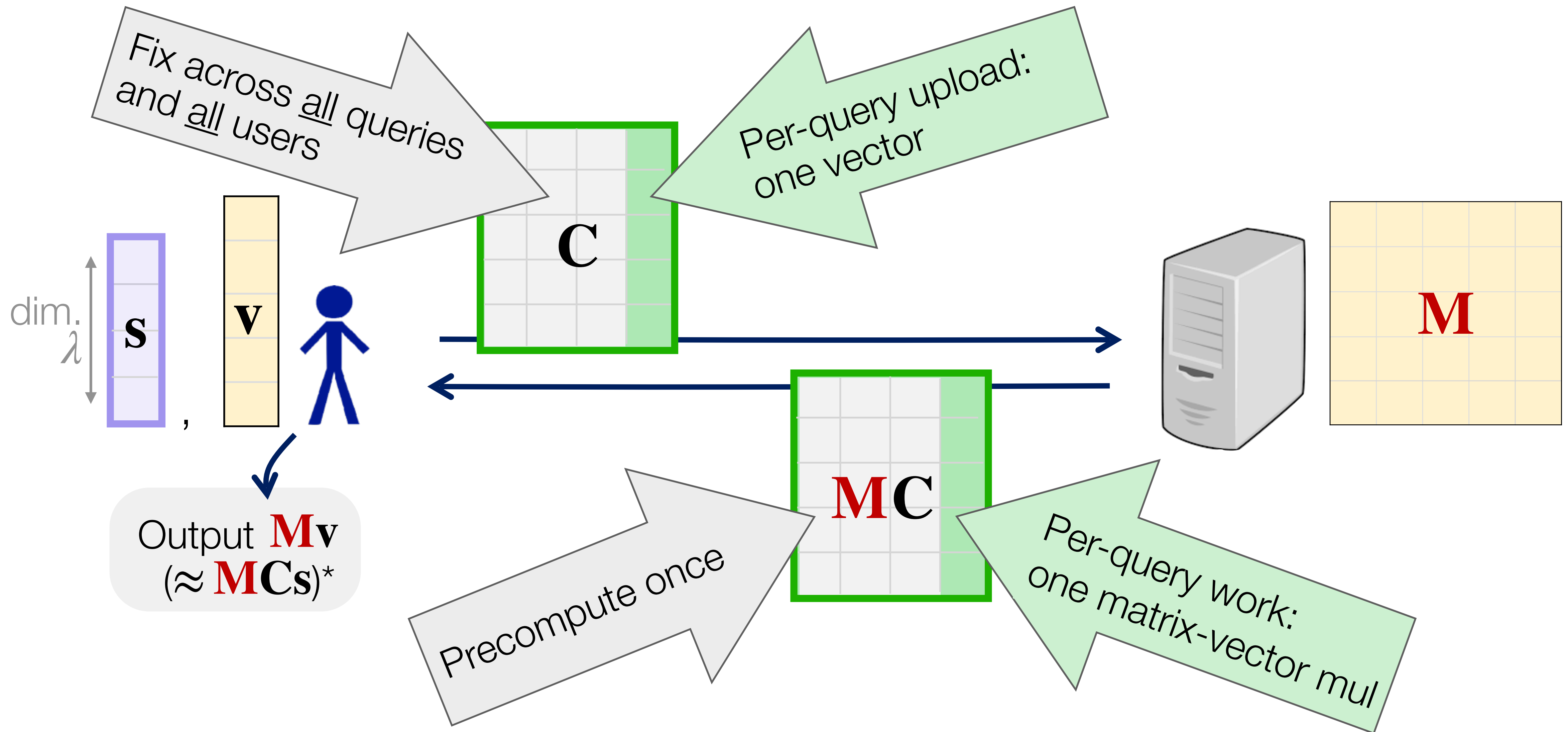
Idea: Expose useful structure for preprocessing



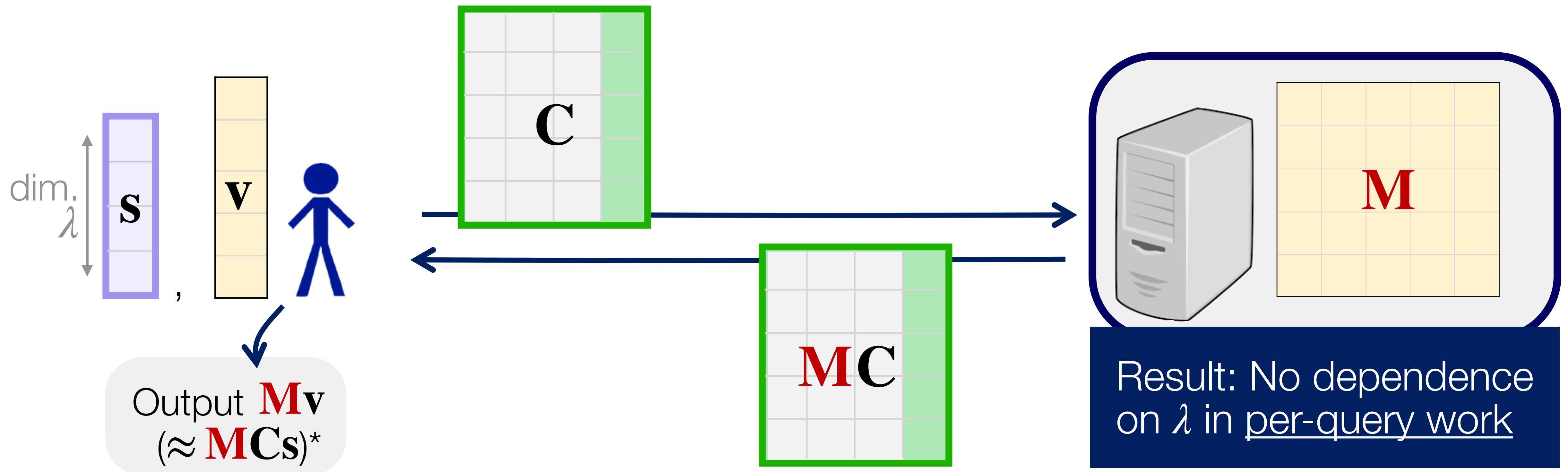
Idea: Expose useful structure for preprocessing



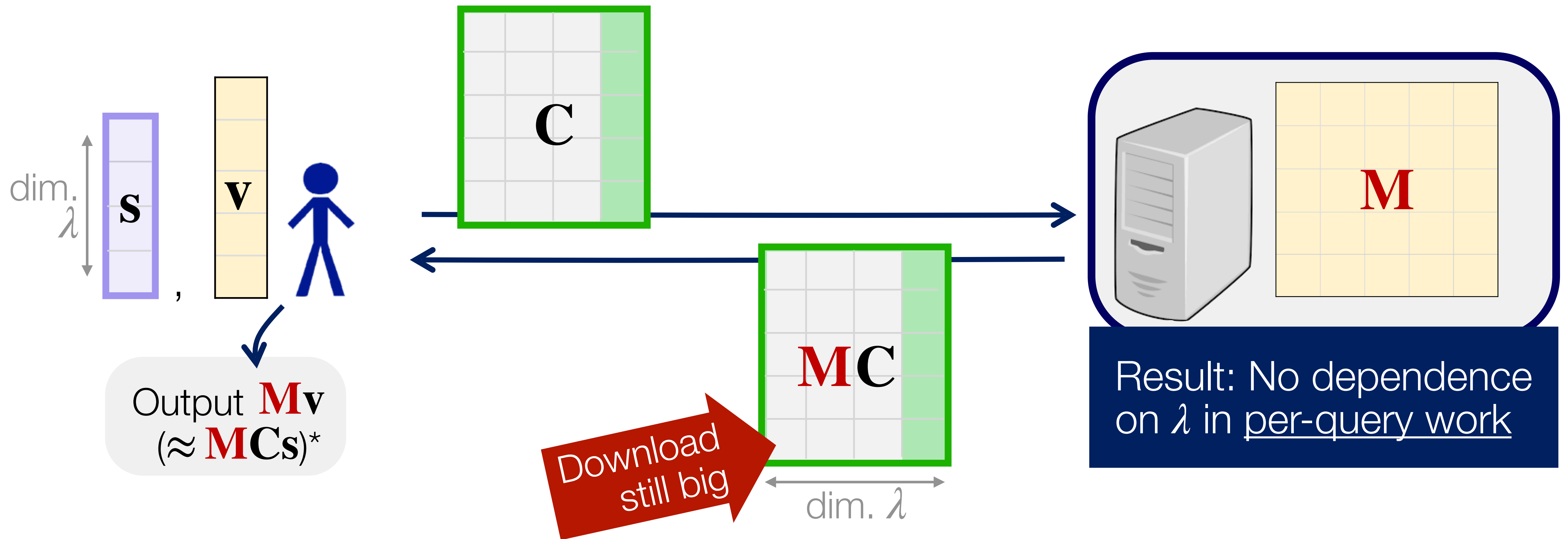
Idea: Expose useful structure for preprocessing



Idea: Expose useful structure for preprocessing

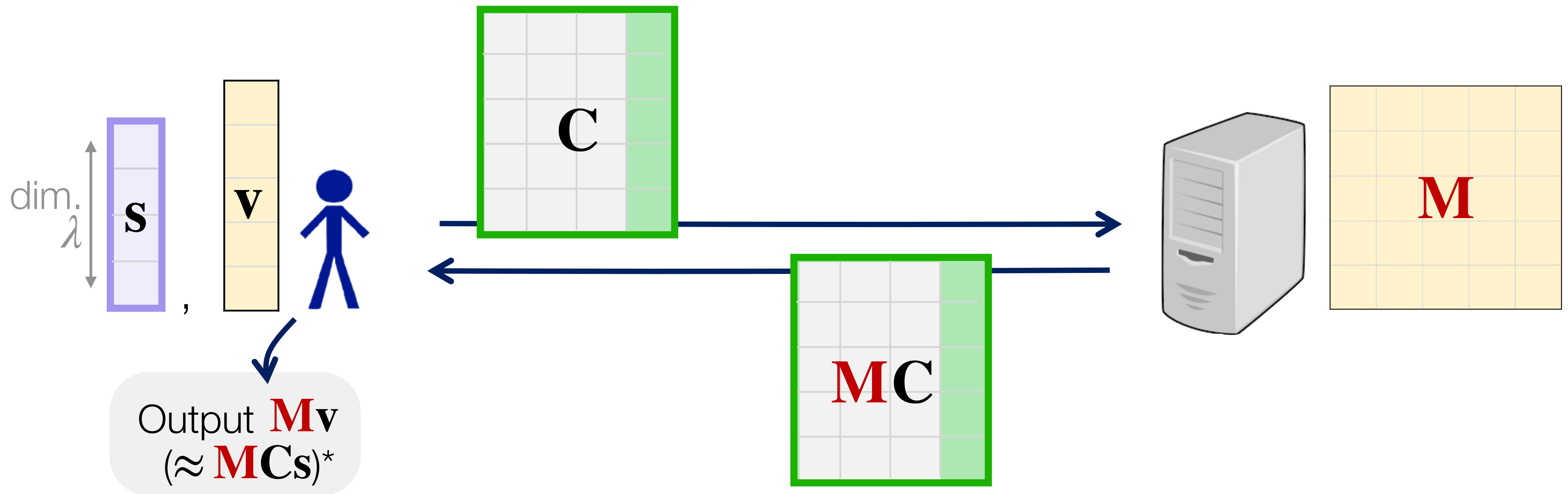


Idea: Expose useful structure for preprocessing



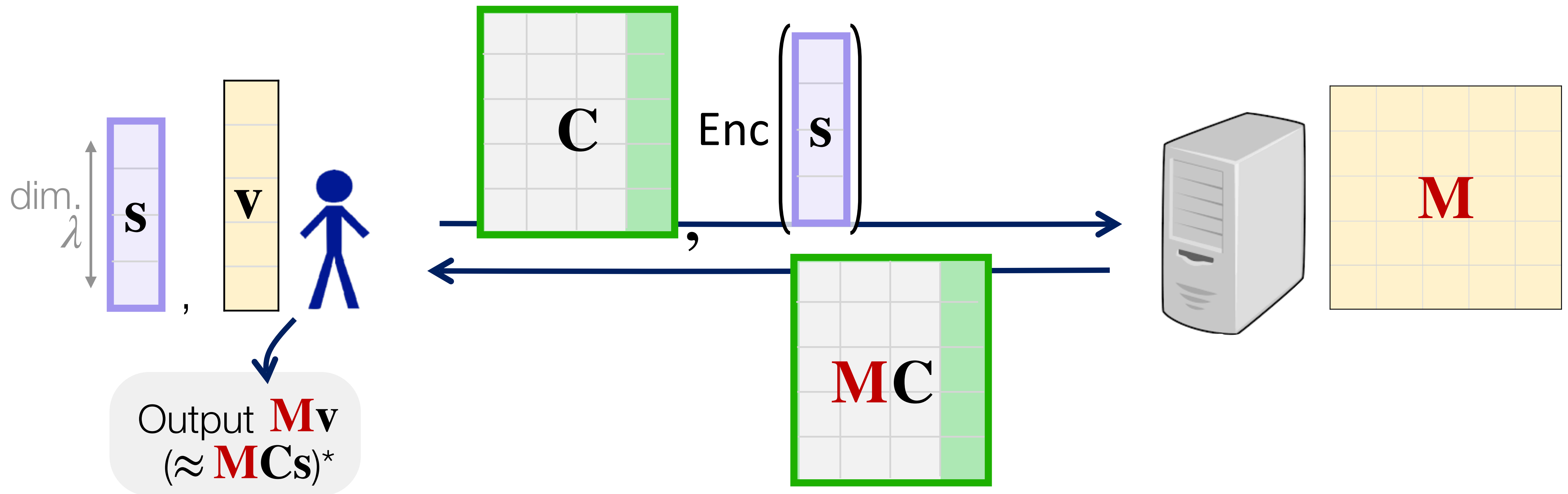
Extension: Repeat to shrink communication

Strategy: decrypt under encryption, inspired by fully-homomorphic schemes [Gen09, BV11]



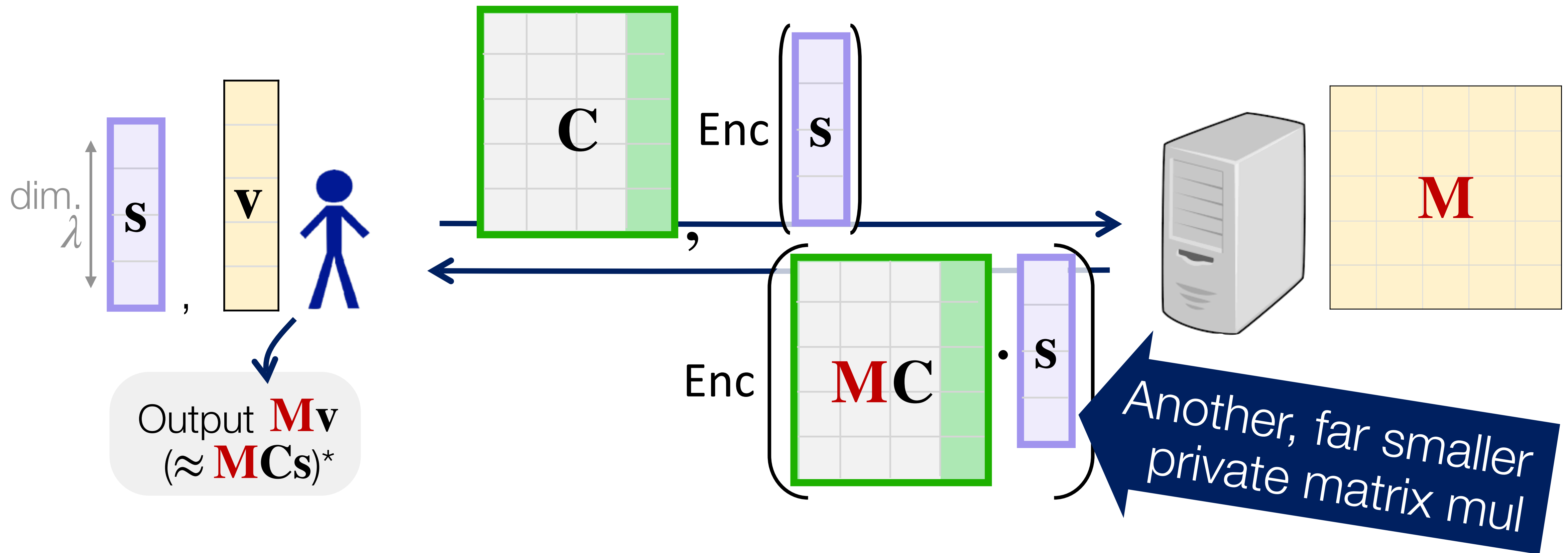
Extension: Repeat to shrink communication

Strategy: decrypt under encryption, inspired by fully-homomorphic schemes [Gen09, BV11]



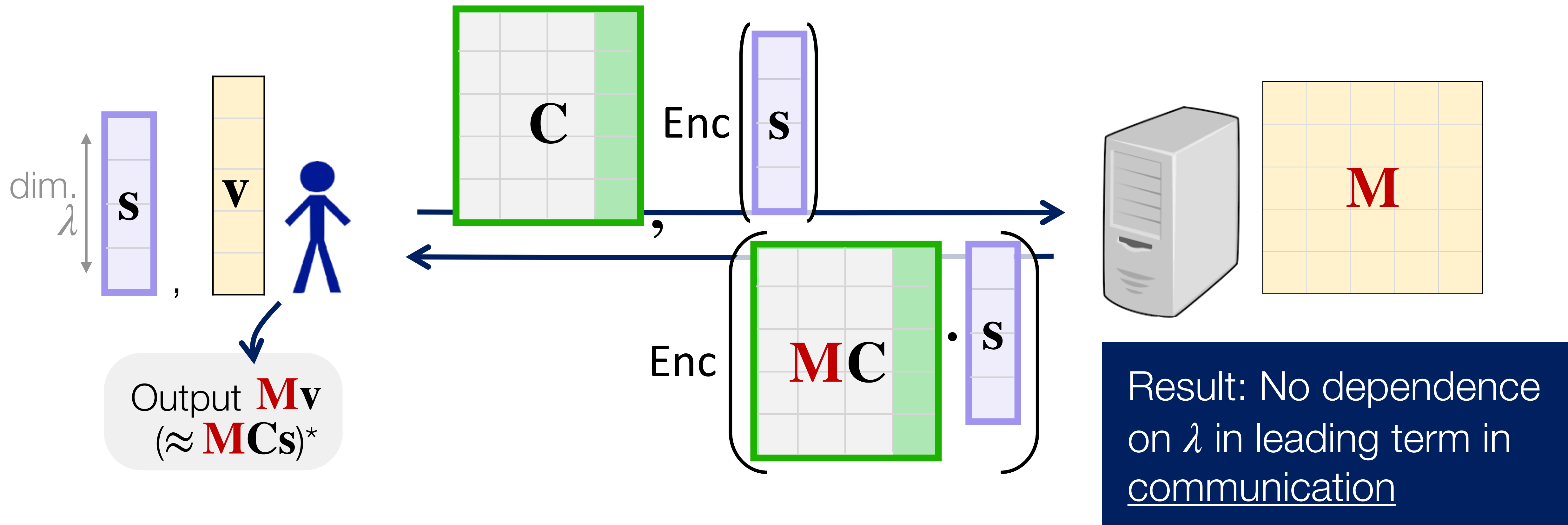
Extension: Repeat to shrink communication

Strategy: decrypt under encryption, inspired by fully-homomorphic schemes [Gen09, BV11]



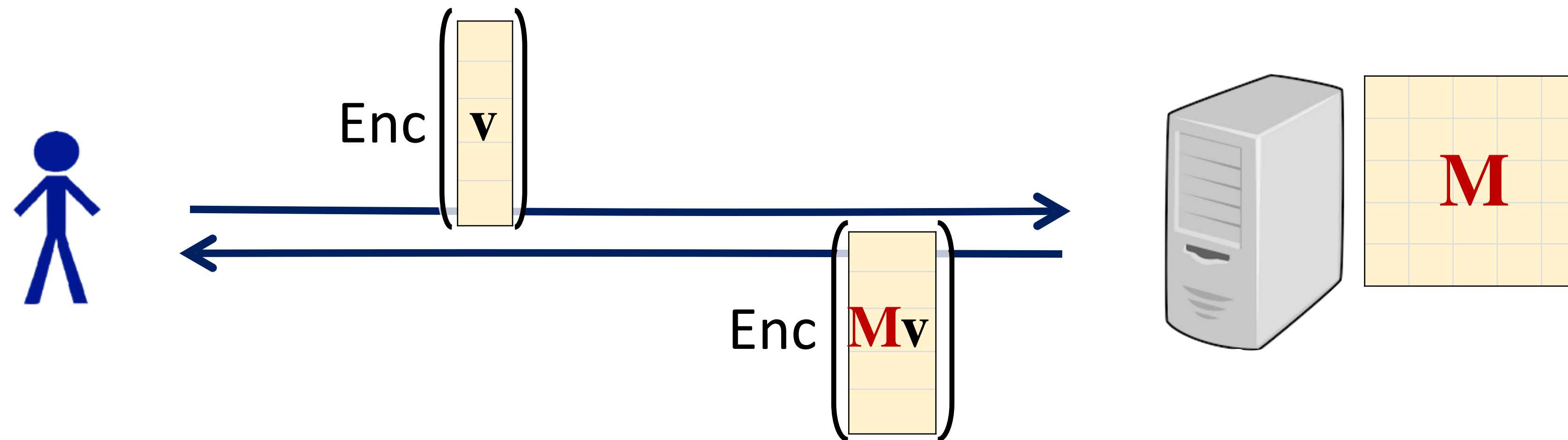
Extension: Repeat to shrink communication

Strategy: decrypt under encryption, inspired by fully-homomorphic schemes [Gen09, BV11]



SimplePIR: The bottom line

With preprocessing, can multiply a matrix by encrypted vector at near-unencrypted speed



Protocol extensions

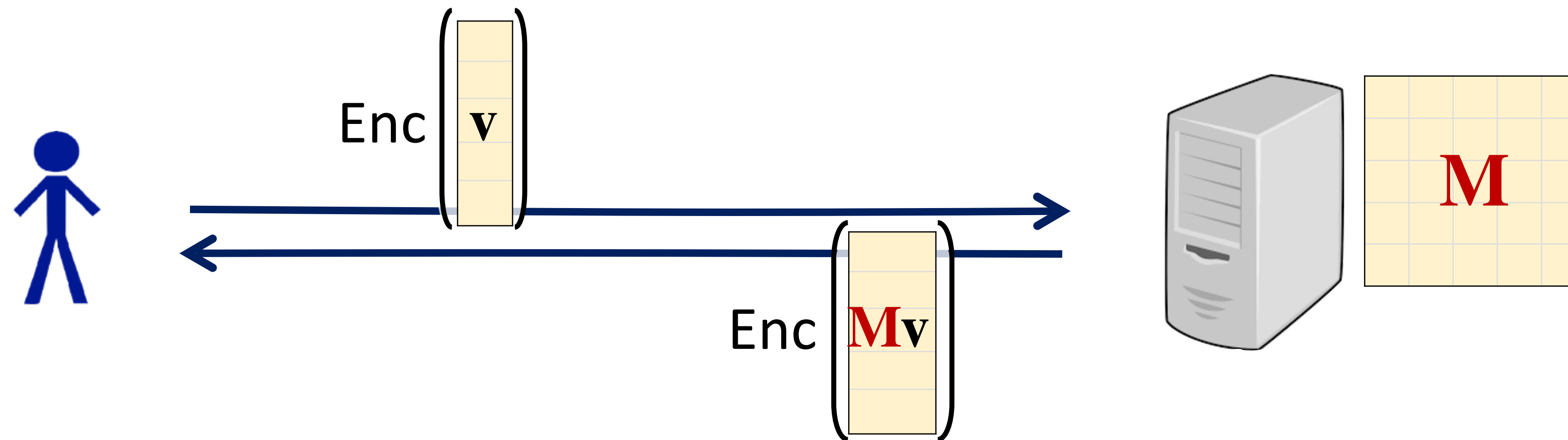
- VeriSimplePIR [CL24, RLP26]
- HintlessPIR [LMRS24]
- YPIR [MW24], InspPIRe [MPSY26]
- “Barely doubly efficient” SimplePIR [Lee25]
- Run on GPUs [LHC25]

Applications

- Private information retrieval
- Private key-value lookups [CD24, HLPZ+25]
- Private retrieval-augmented-generation [Z24, WLZZ25]
- Private set intersection [LLMT25]
- Private certificate auditing [HPW25]

SimplePIR: The bottom line

With preprocessing, can multiply a matrix by encrypted vector at near-unencrypted speed



Protocol extensions

- VeriSimplePIR [CL24, RLP26]
- HintlessPIR [LMRS24]
- YPIR [MW24], InspPIRe [MPSY26]
- “Barely doubly efficient” SimplePIR [Lee25]
- Run on GPUs [LHC25]

Applications

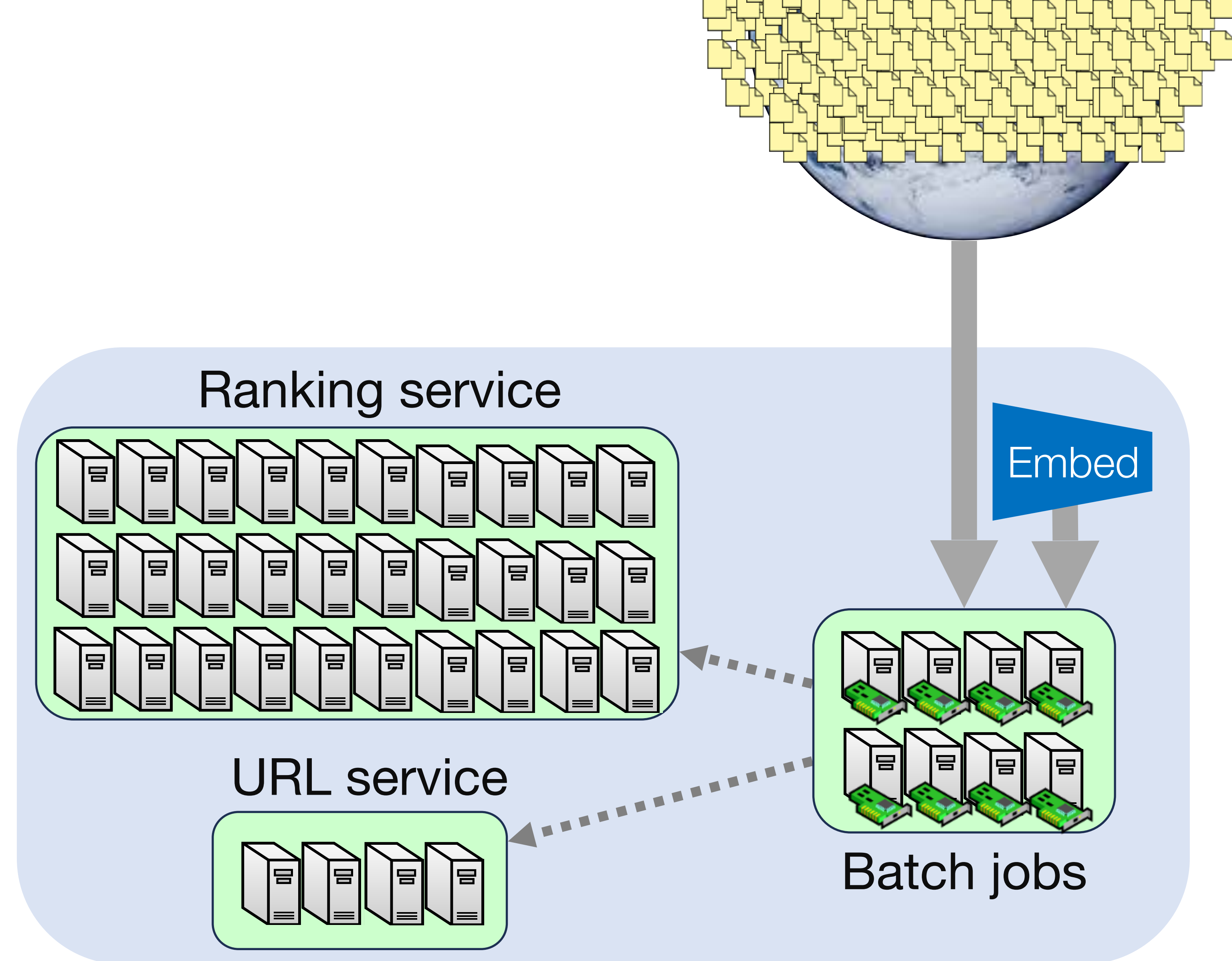
- Private information retrieval
- Private key-value lookups [CD24, HLPZ+25]
- Private retrieval-augmented-generation [Z24, WLZZ25]
- Private set intersection [LLMT25]
- Private certificate auditing [HPW25]



Tiptoe: Design in layers



Tiptoe: The full system

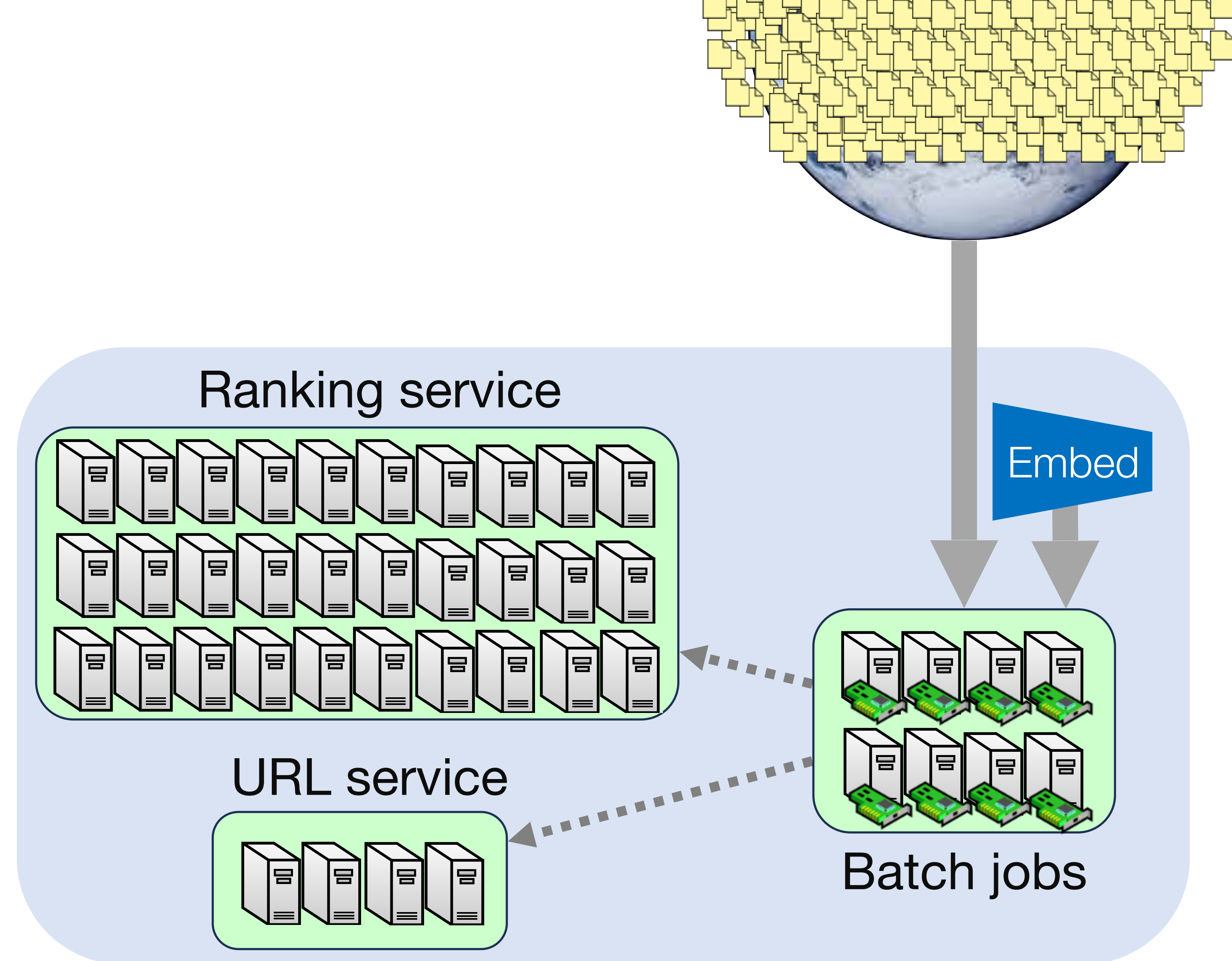


github.com/ahenzinger/tiptoe
github.com/ahenzinger/simplepir

Tiptoe: The full system

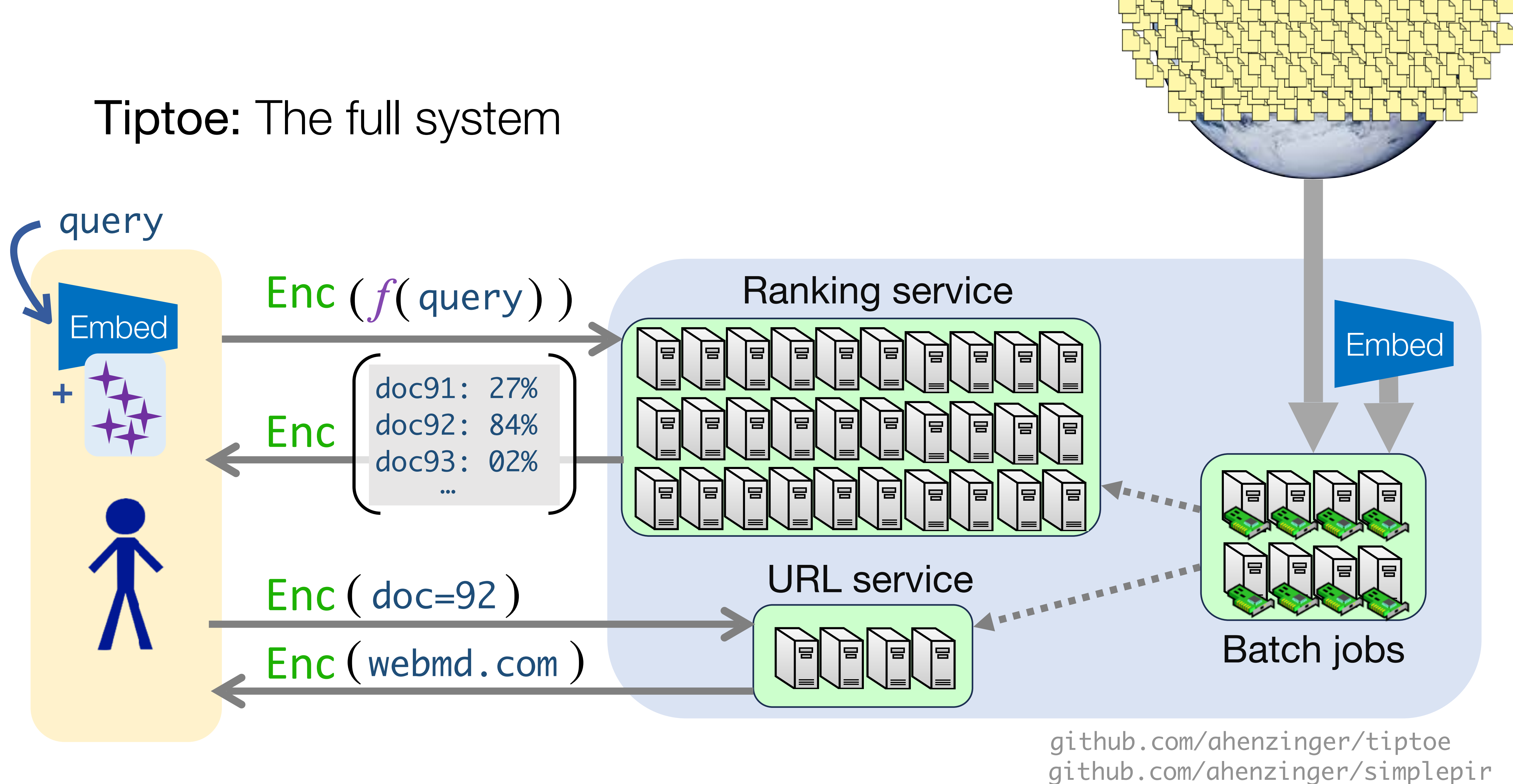
Batch jobs:

1. Embed documents
2. Cluster embeddings
3. Preprocess crypto



github.com/ahenzinger/tiptoe
github.com/ahenzinger/simplepir

Tiptoe: The full system



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

On English Common crawl with 364M pages

	Coeus, scaled up (SOSP'21)	Tiptoe (SOSP'23)	Gain
Client storage	-	0.3 GiB	
Server compute (per query)	928,800 core-s	145 core-s	6,400 ×
Communication (per query)	3,880 MiB	57 MiB	62 ×
Latency (per query)	-	2.7 s	

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

On English Common crawl with 364M pages

	Coeus, scaled up (SOSP'21)	Tiptoe (SOSP'23)	Gain
Client storage	SimplePIR: 10 × less computation		
Server compute (per query)	Semantic embeddings: 100 × smaller than a bag-of-words representation		6,400 ×
Communication (per query)	Clustering: communication sublinear in N		62 ×
Latency (per query)	-	2.7 s	

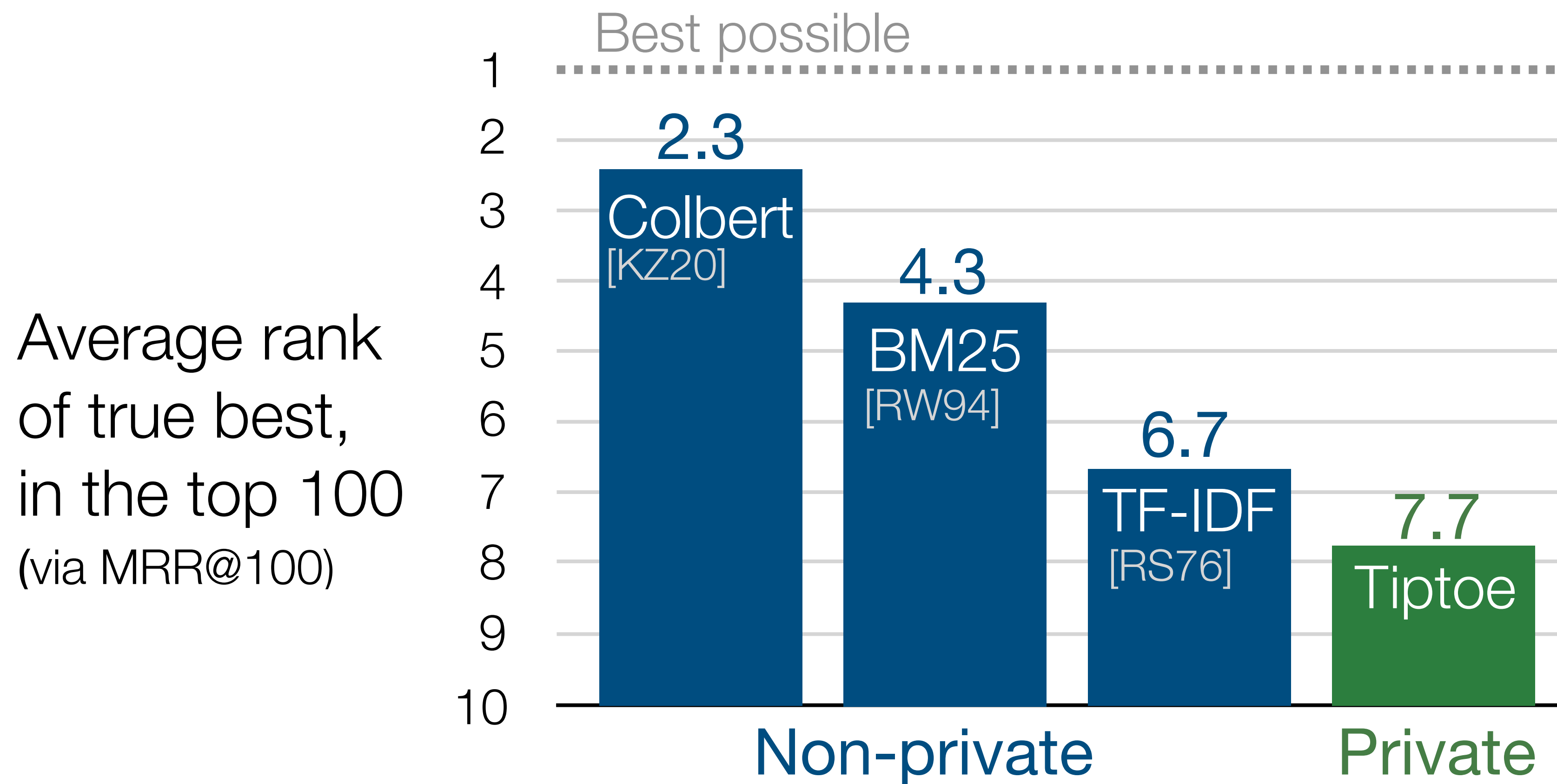
Tiptoe is cheaper than state-of-the-art private search

On English Common crawl with 364M pages

	Coeus, scaled up (SOSP'21)	Tiptoe (SOSP'23)	Gain
Client storage	-	0.3 GiB	
Server compute (per query)	928,800 core-s	145 core-s	6,400 ×
Communication (per query)	3,880 MiB	57 MiB	62 ×
Latency (per query)	-	2.7 s	

Tiptoe ranks the best result in the top 10, on average

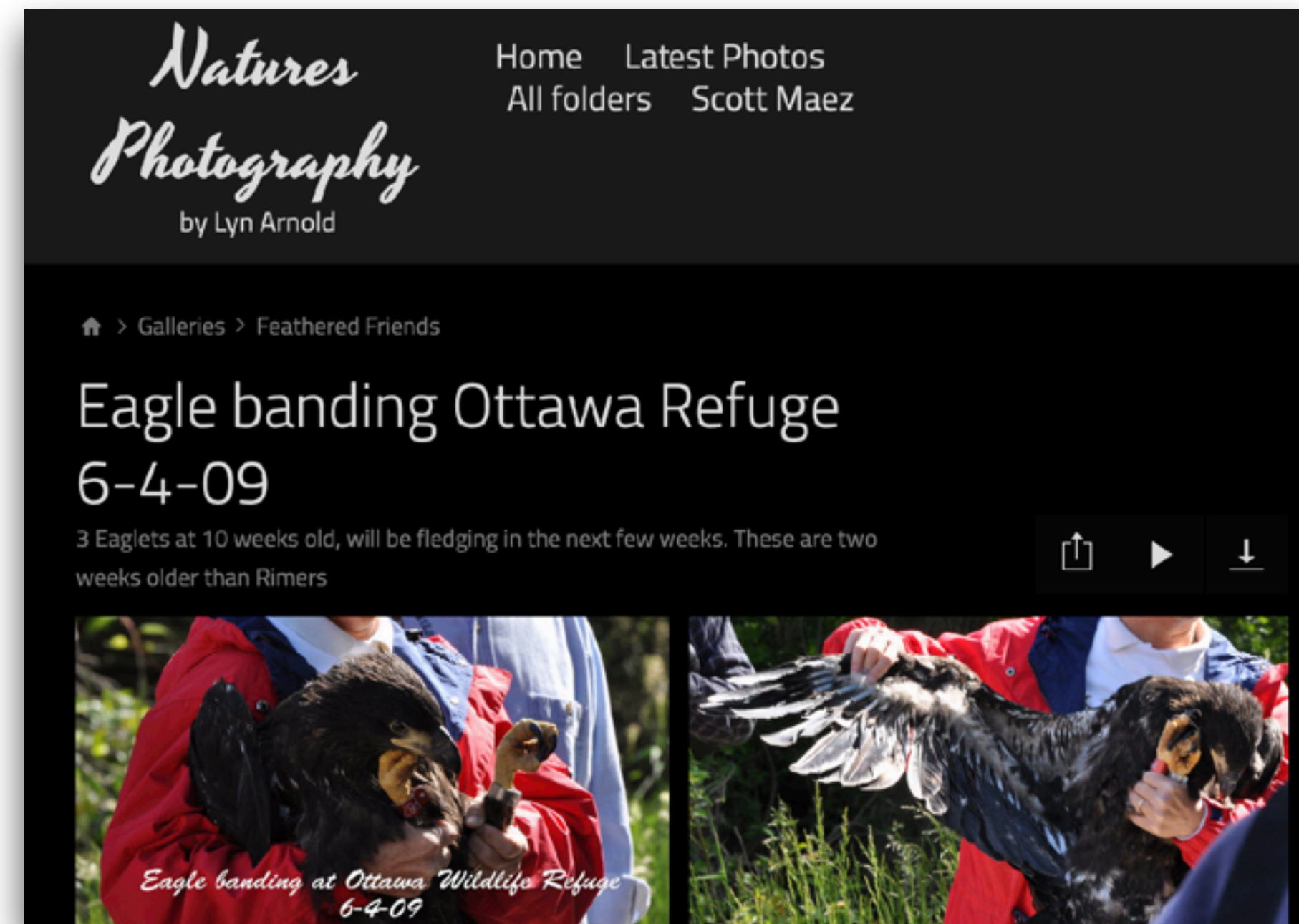
On MS MARCO document ranking data set



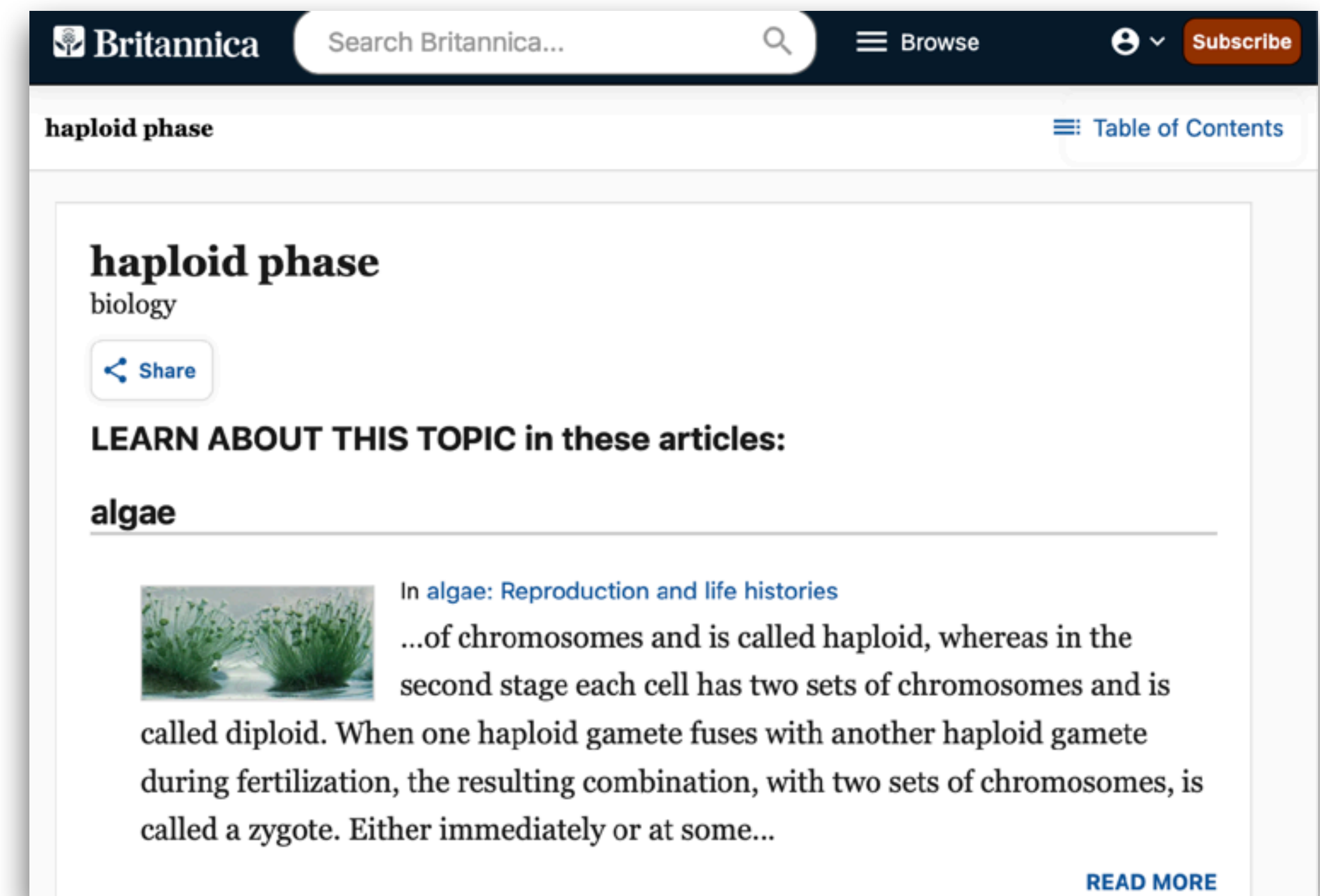
Embedding model: msmarco-distilbert-base-tas-b

Examples: Tiptoe's text search

Q: how long before eagles get feathers



Q: the meaning of haploid cell



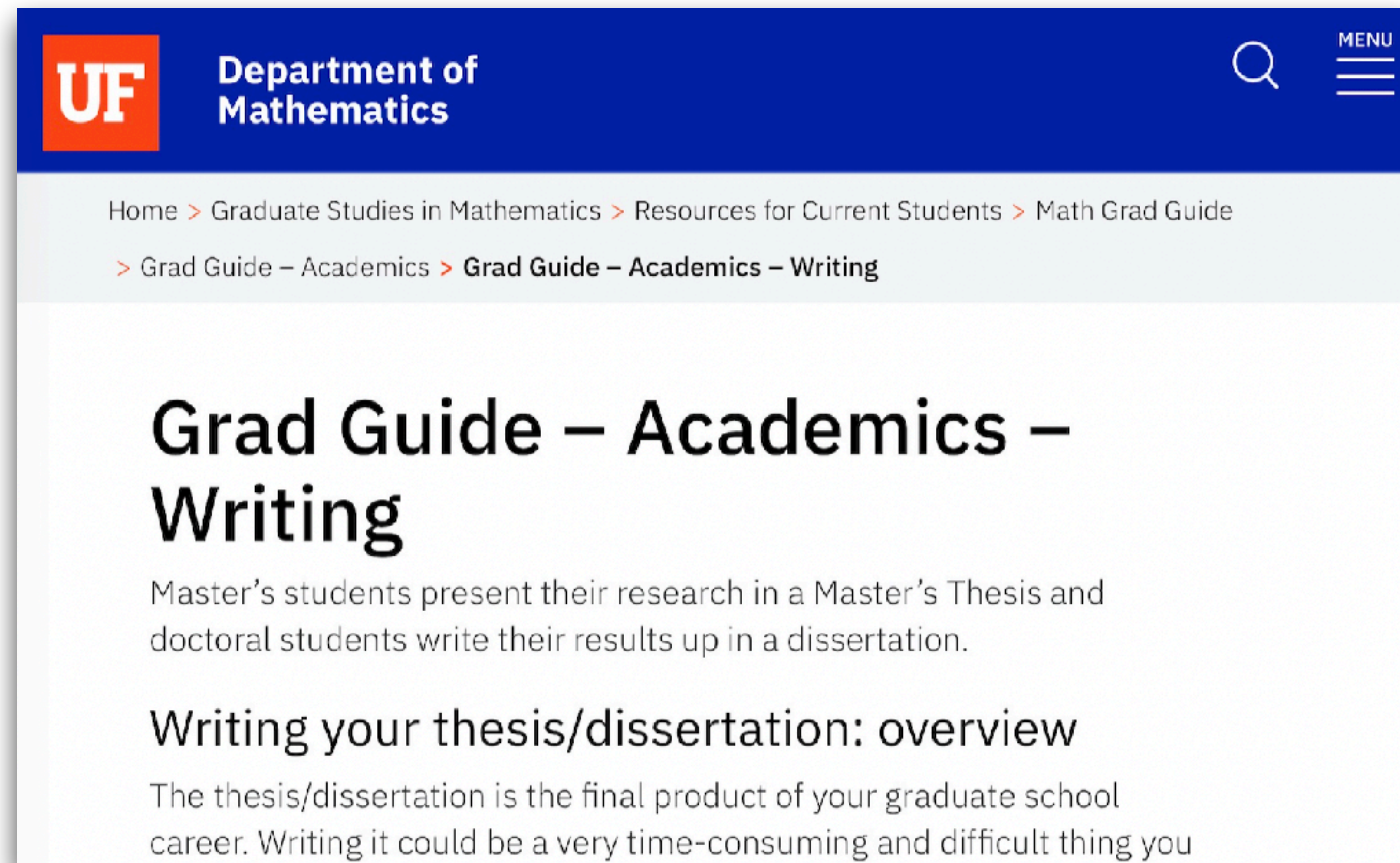
On the Common crawl data set

Examples: Tiptoe's text search

Q: Advice on writing a doctoral thesis

Examples: Tiptoe's text search

Q: Advice on writing a doctoral thesis



The screenshot shows the top navigation bar of the University of Florida Department of Mathematics website. It includes the UF logo, the department name, a search icon, and a menu icon. Below the navigation bar is a breadcrumb trail: Home > Graduate Studies in Mathematics > Resources for Current Students > Math Grad Guide > Grad Guide – Academics > Grad Guide – Academics – Writing. The main heading is "Grad Guide – Academics – Writing". The text below the heading states: "Master's students present their research in a Master's Thesis and doctoral students write their results up in a dissertation." followed by a subheading "Writing your thesis/dissertation: overview" and a paragraph: "The thesis/dissertation is the final product of your graduate school career. Writing it could be a very time-consuming and difficult thing you".

UF Department of Mathematics

Home > Graduate Studies in Mathematics > Resources for Current Students > Math Grad Guide > Grad Guide – Academics > Grad Guide – Academics – Writing

Grad Guide – Academics – Writing

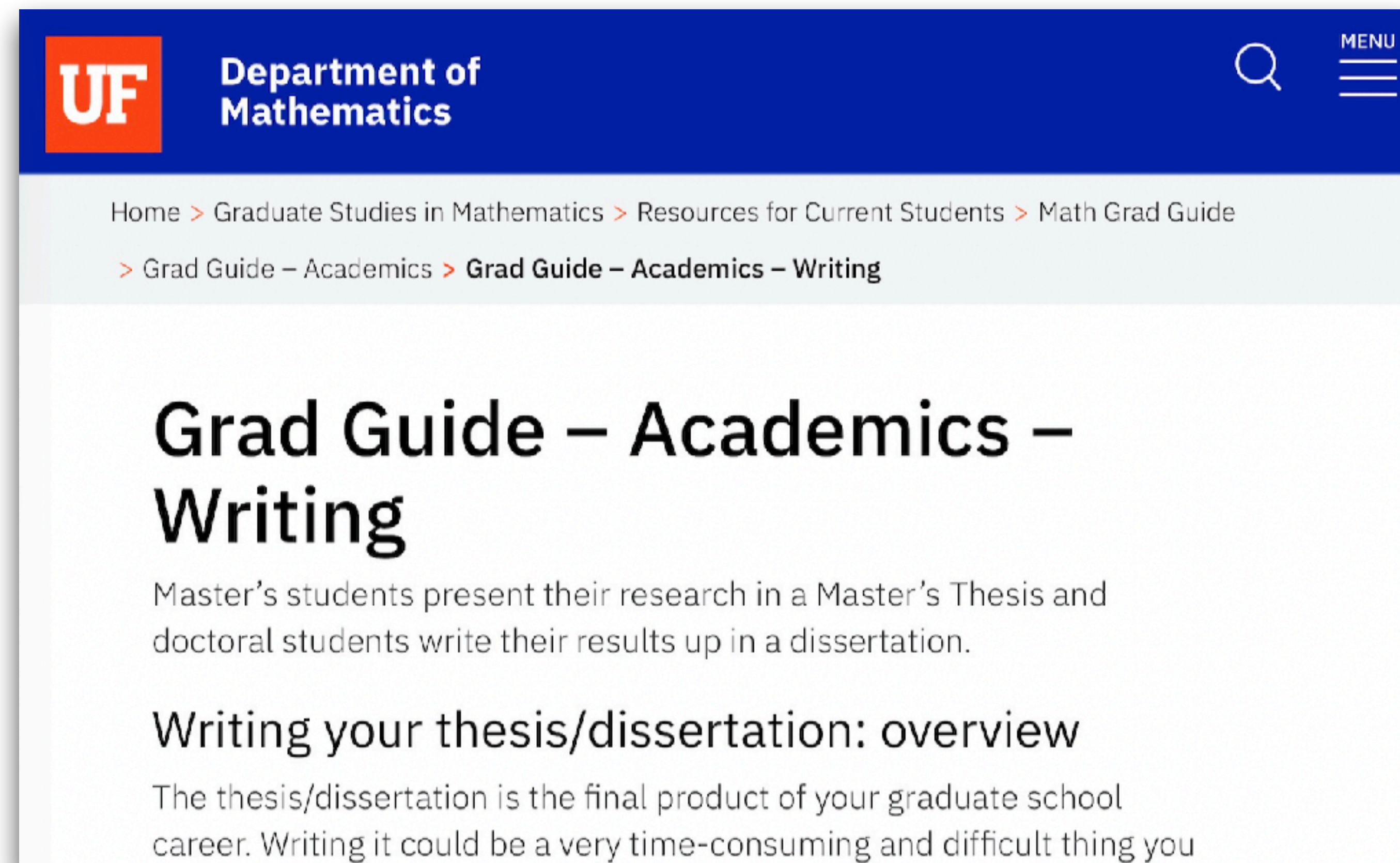
Master's students present their research in a Master's Thesis and doctoral students write their results up in a dissertation.

Writing your thesis/dissertation: overview

The thesis/dissertation is the final product of your graduate school career. Writing it could be a very time-consuming and difficult thing you

Examples: Tiptoe's text search

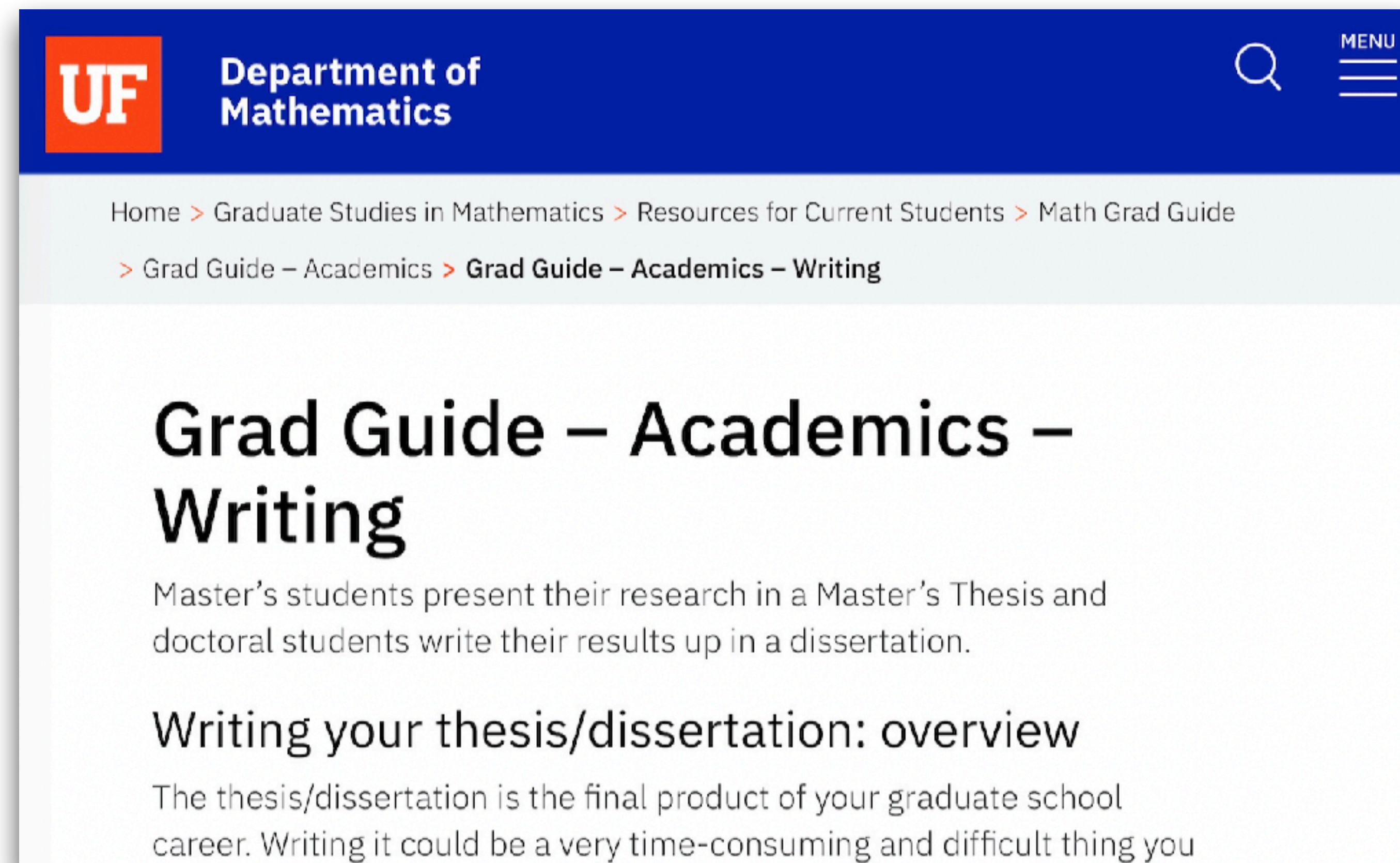
Q: Advice on writing a doctoral thesis



Q: Advice on writing a doctoral thesis in Computer Science

Examples: Tiptoe's text search

Q: Advice on writing a doctoral thesis

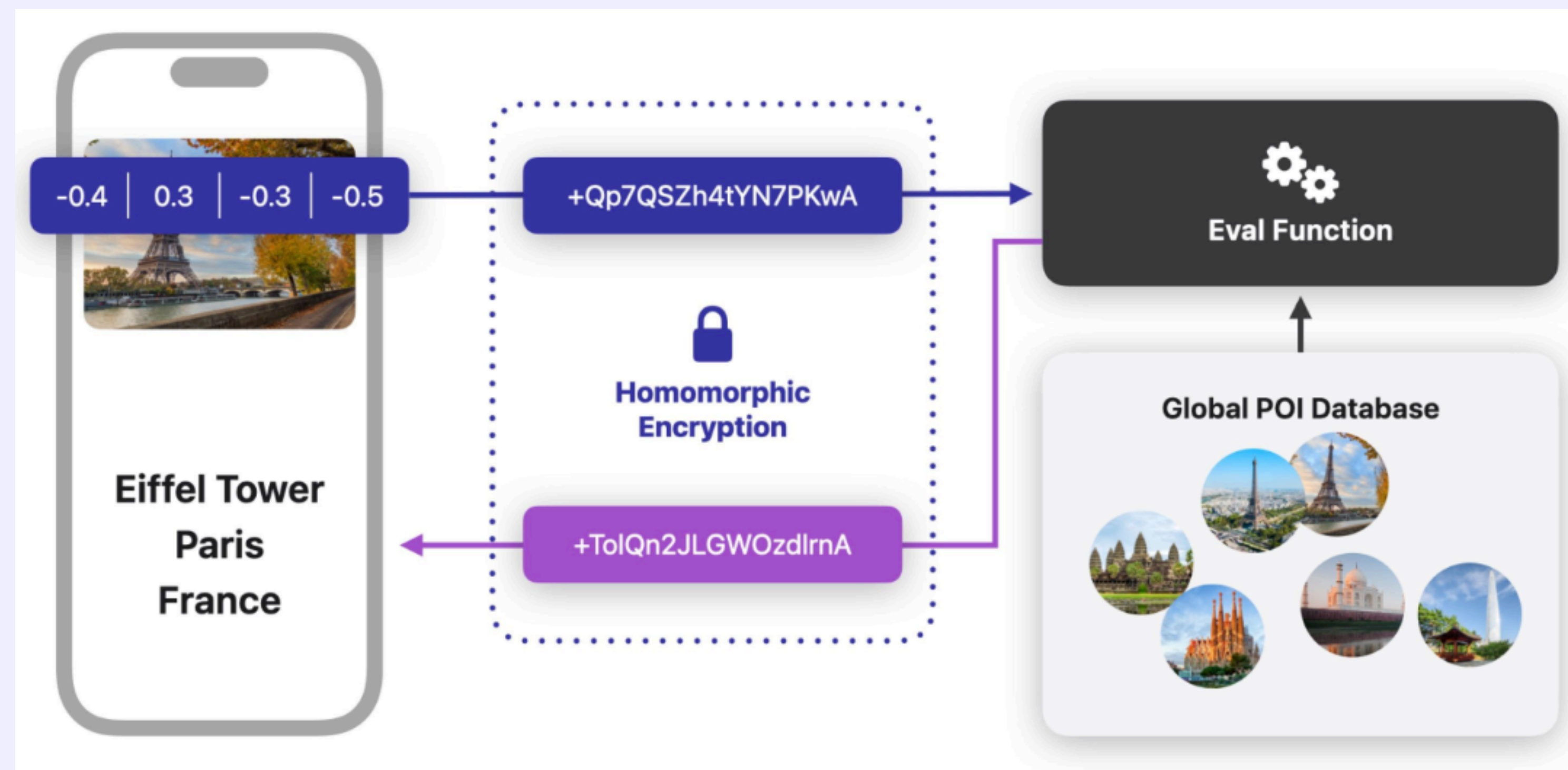


Q: Advice on writing a doctoral thesis in Computer Science

A: buydissertations.com/computer-science.html

Apple adopted Tiptoe's private nearest-neighbor search in iOS 18 and macOS “Enhanced Visual Search”

Goal: label landmarks in photos without revealing the photos to Apple servers



machinelearning.apple.com/research/wally-search
machinelearning.apple.com/research/homomorphic-encryption

My research: A privacy-preserving software stack

Applications

Libraries

Cryptographic
operations

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- Private recommendations

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- Private matrix factorization

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

My research: A privacy-preserving software stack

Applications

Libraries

Cryptographic
operations

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- **Private recommendations** — Train lightweight recommender models on private data

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- **Private matrix factorization**

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- Private recommendations

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- Private matrix factorization

Cryptographic operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- Private information retrieval
- **Homomorphic encryption** ——— New ways to add and multiply on encrypted data

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- Private recommendations

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- Private matrix factorization

Cryptographic operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

- Private matrix multiplication
- **Private information retrieval** — “Doubly-efficient” schemes via new time-space tradeoffs
- Homomorphic encryption

My research: A privacy-preserving software stack

Applications

Privacy-preserving systems (SOSP'23, USENIX Sec'26)

- Private web search
- Private recommendations

Libraries

Algorithms on encrypted data (USENIX Sec'23, '25)

- Private nearest-neighbor search
- Private matrix factorization

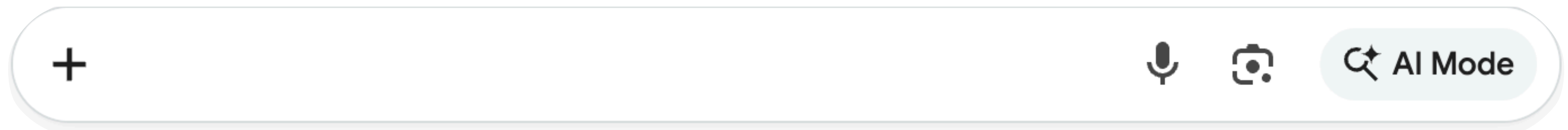
Cryptographic operations

Fast cryptography (EUROCRYPT'22, '25, '26, '26)

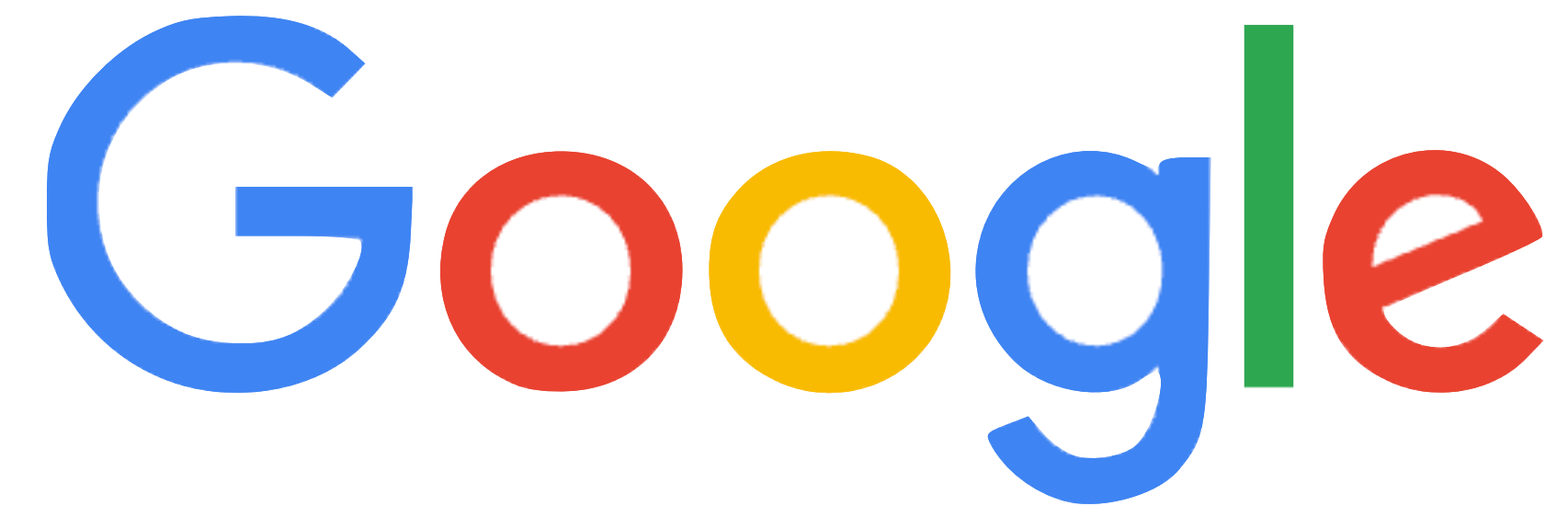
- Private matrix multiplication
- Private information retrieval
- Homomorphic encryption

Outsourcing computation has always required revealing our data

Google



Outsourcing computation has always required revealing our data



+ how do i know if |



AI Mode

how do i know if **i have diabetes**

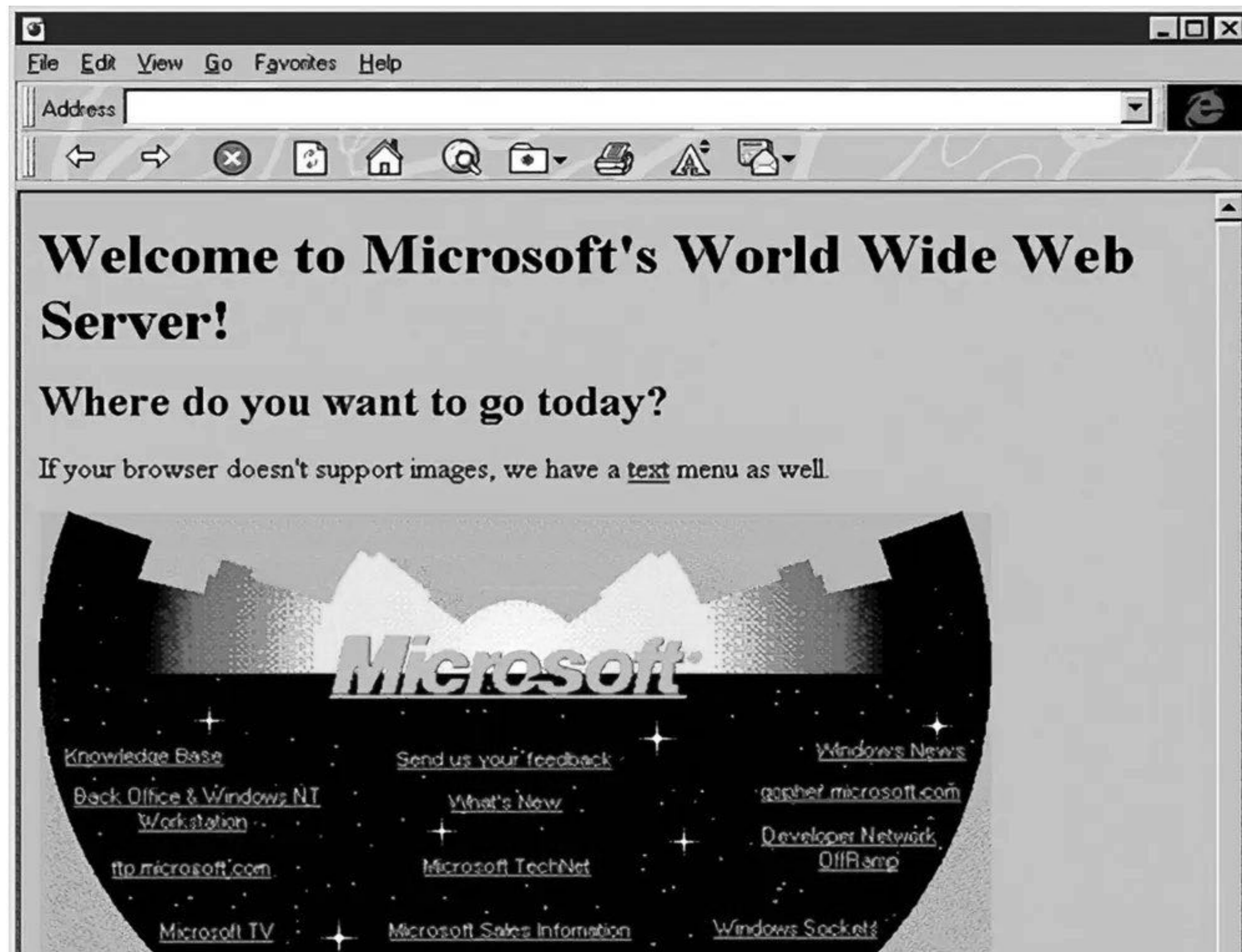
how do i know if **i have adhd**

how do i know if **i have a concussion**

Google Search

I'm Feeling Lucky

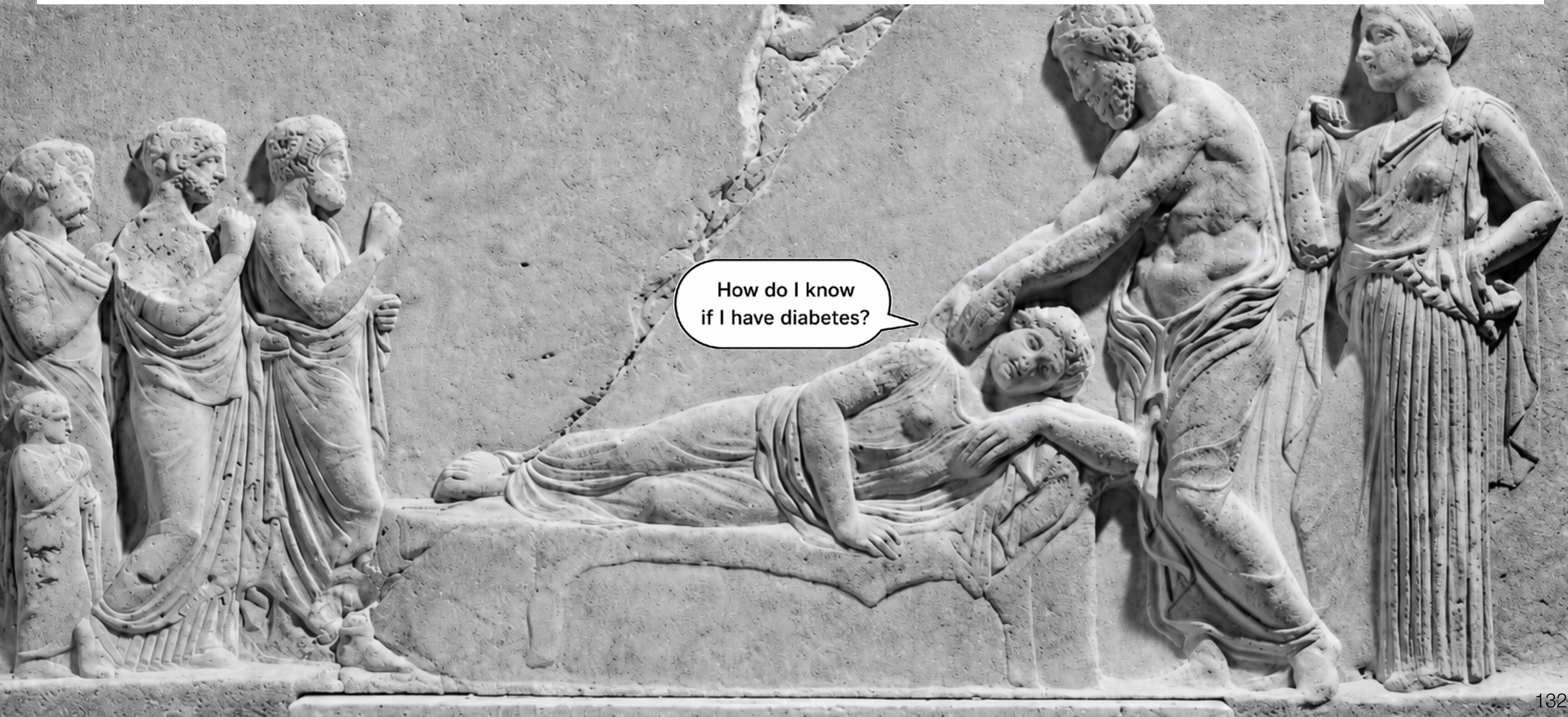
Outsourcing computation has always required revealing our data



Outsourcing computation has always required revealing our data

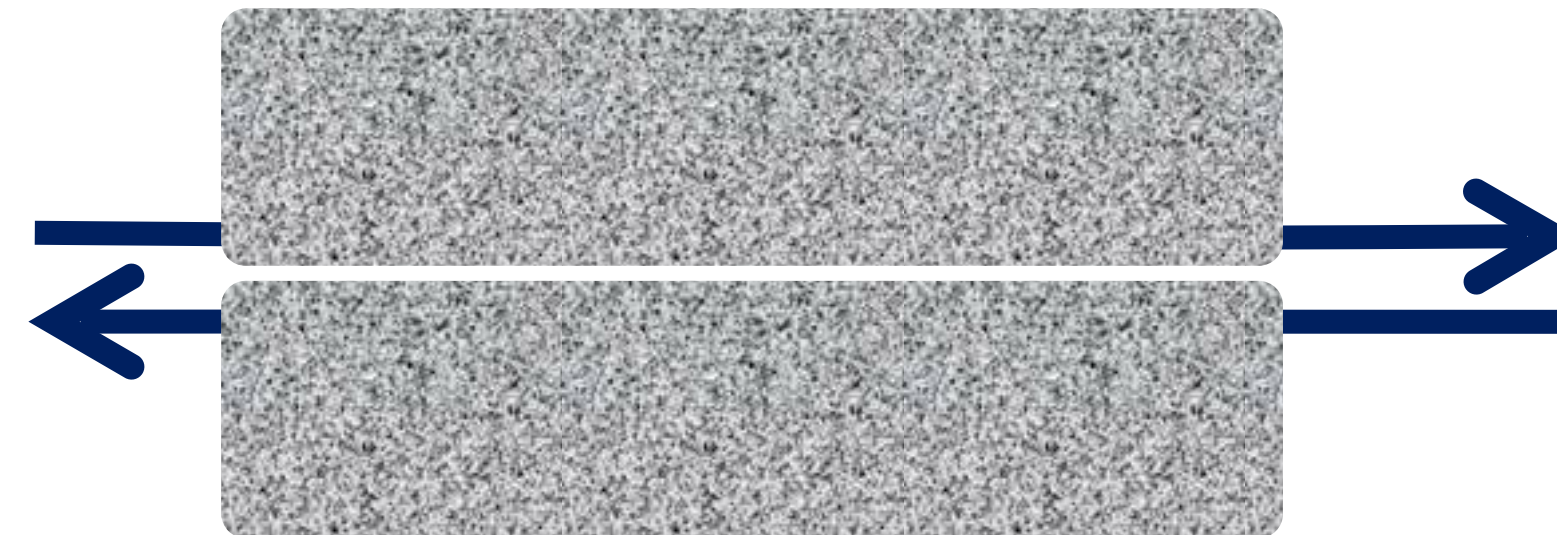


Outsourcing computation has always required revealing our data



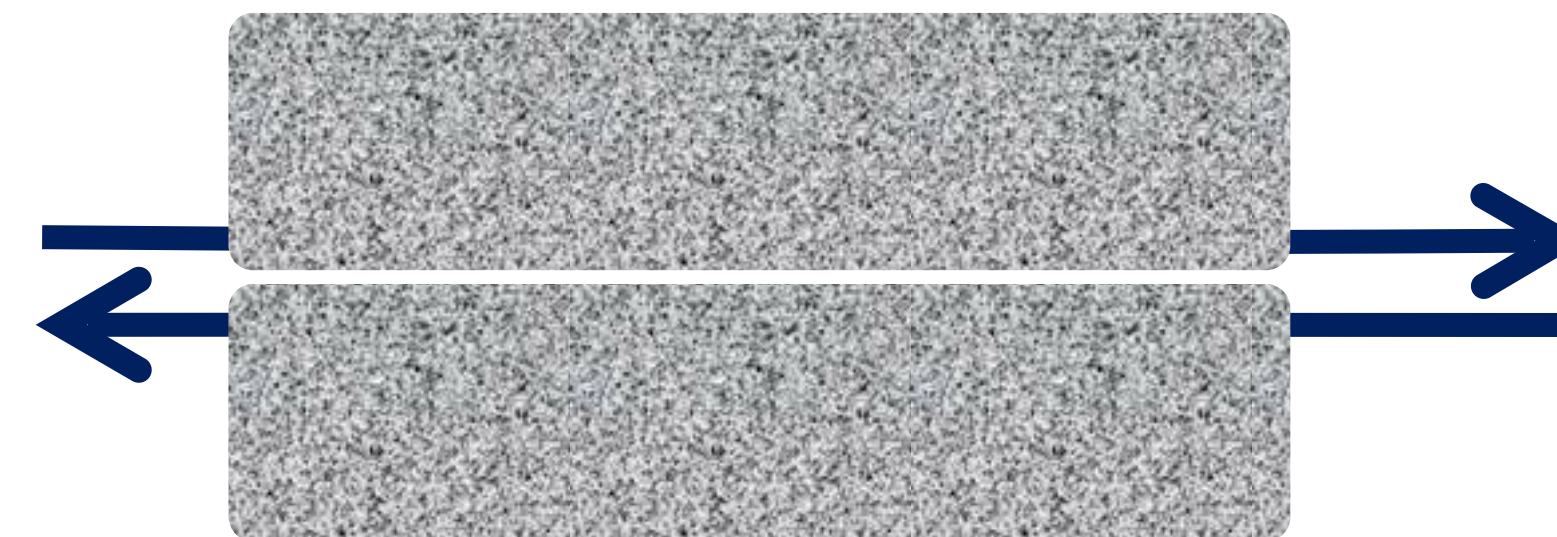
Computer science can give us more: **functionality and privacy**

“How do I know if I have diabetes?”



≈

“Pizza near me”



Private
search
engine

Conclusion

Today, computer systems see our secrets. They don't have to.

We can build privacy-protecting systems by **re-thinking the stack**:

- new system architectures
- new algorithms on encrypted data
- new cryptography

Privacy at scale is in reach, if we design it right.

Thank you!

Acknowledgements

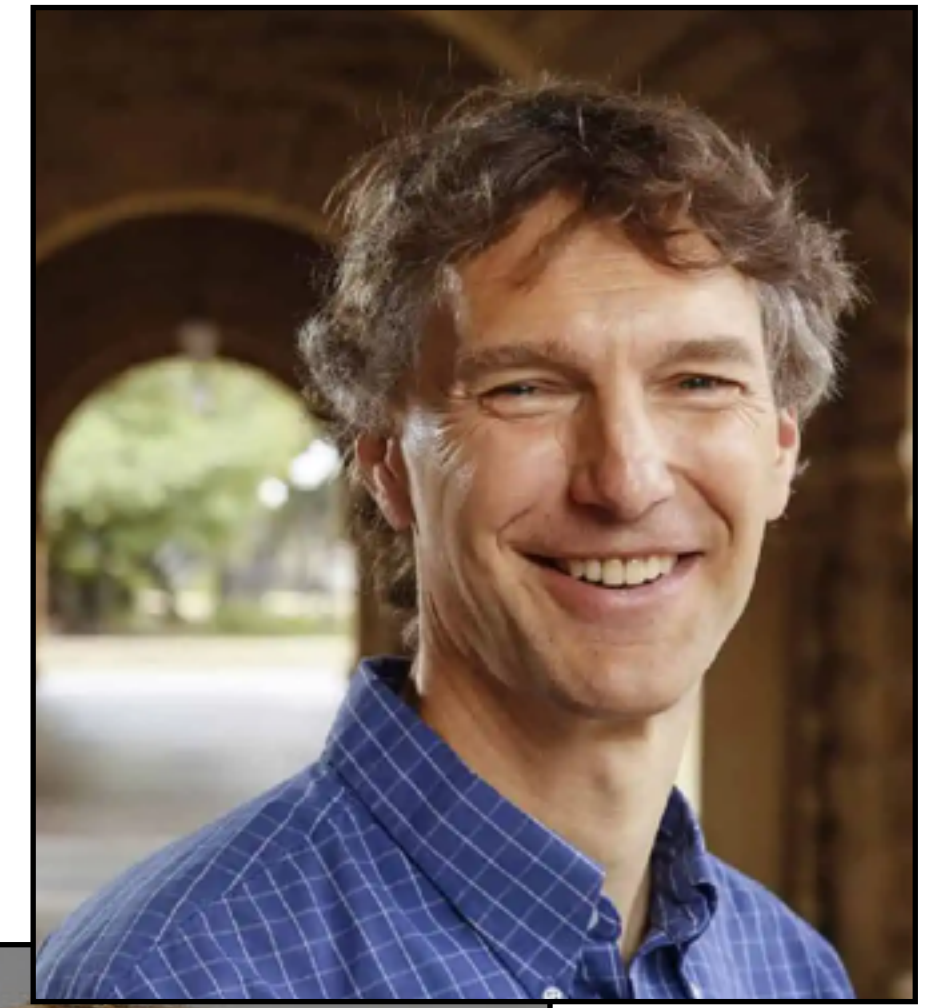
My advisor



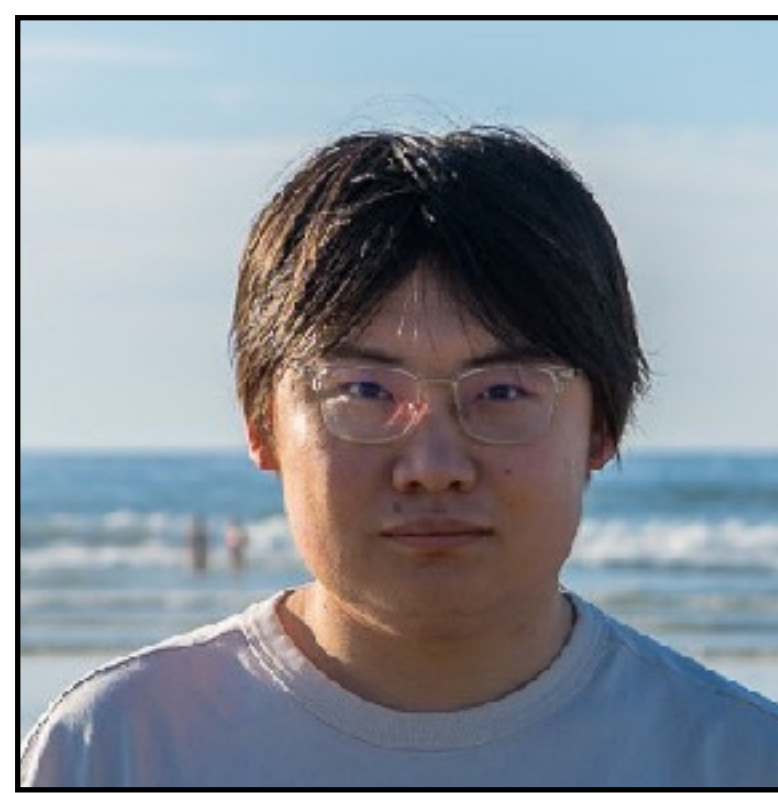
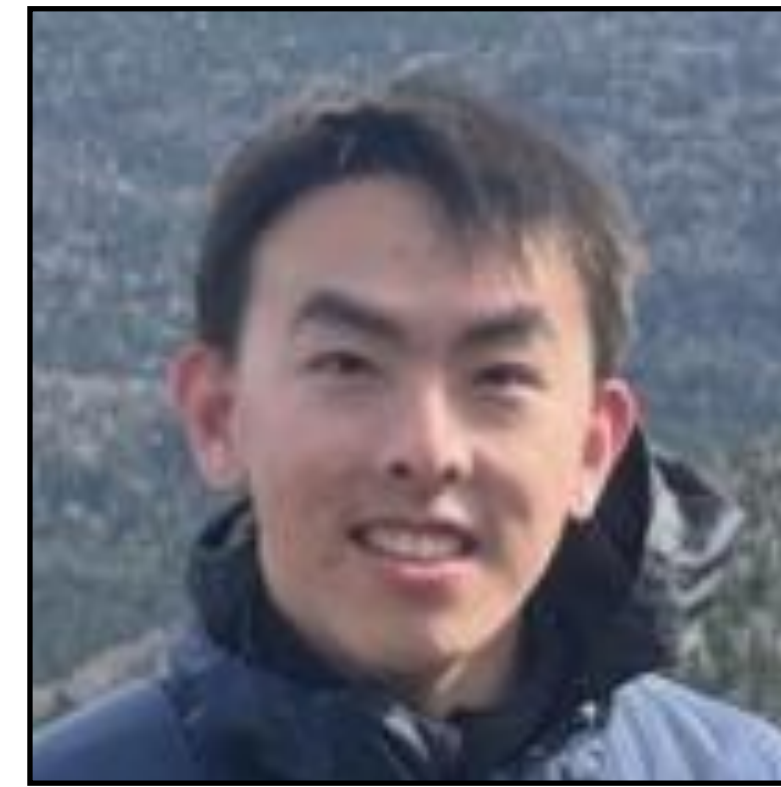
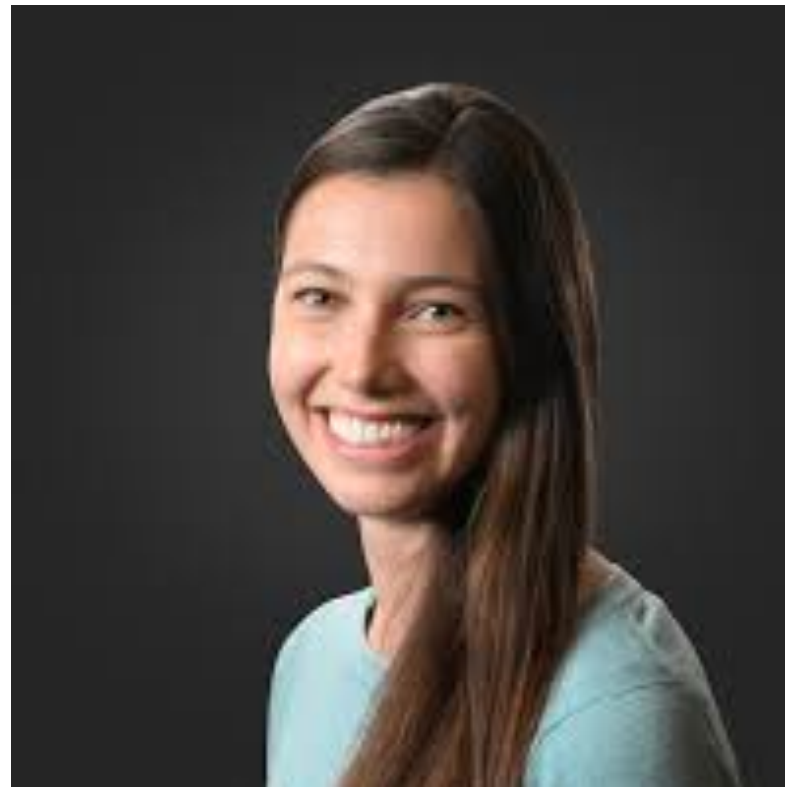
My committee



Trusted Oracles



My collaborators



PDOS past & present



PDOS past & present



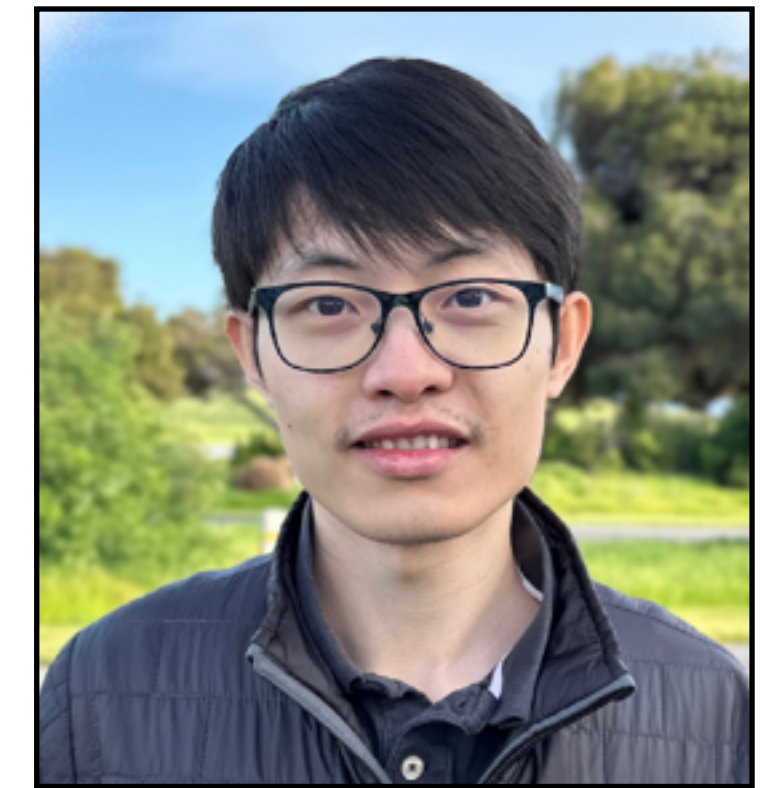
PDOS: The security sub-group



The crypto group

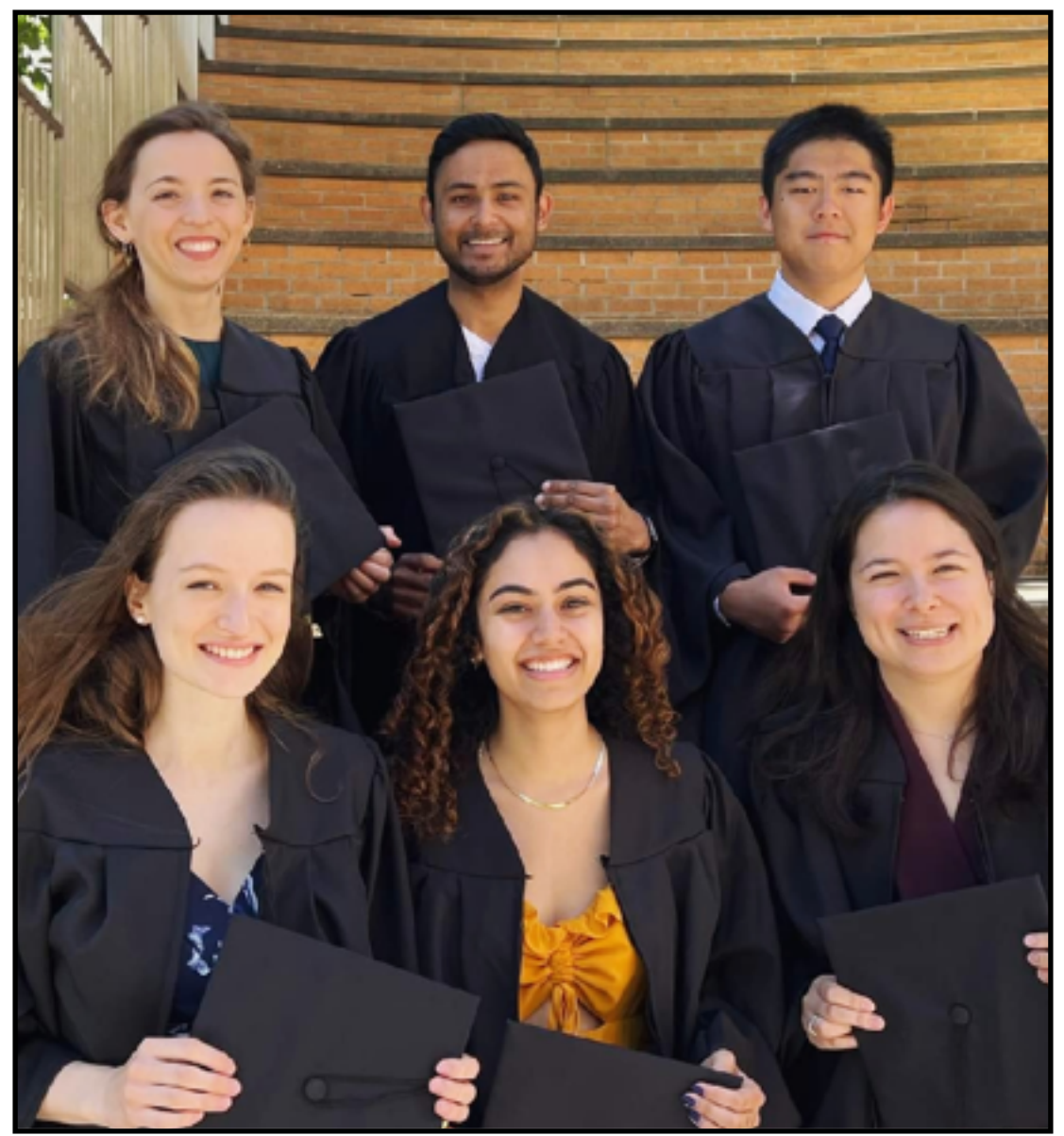


The Stanford applied crypto group



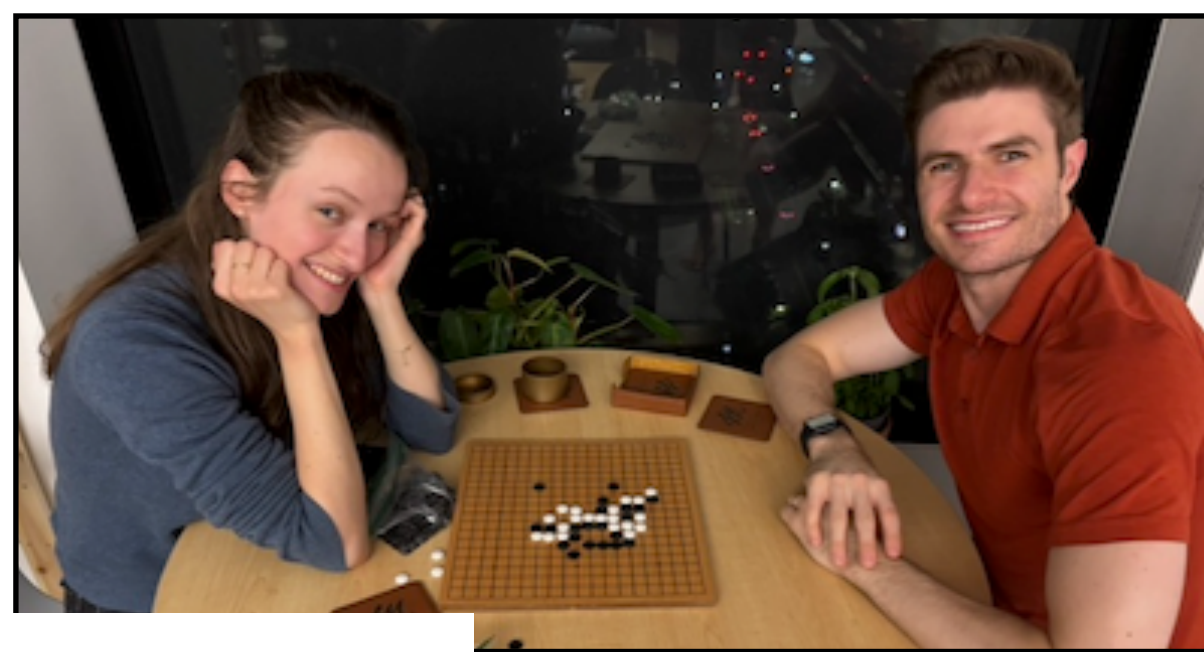
“My students”





Friends from grad school





Friends outside grad school



My soon-to-be family



My sisters



My parents



Nate



Celebration to follow!

Cake and snacks on 9th floor of the G-tower,
after the private portion of the defense.