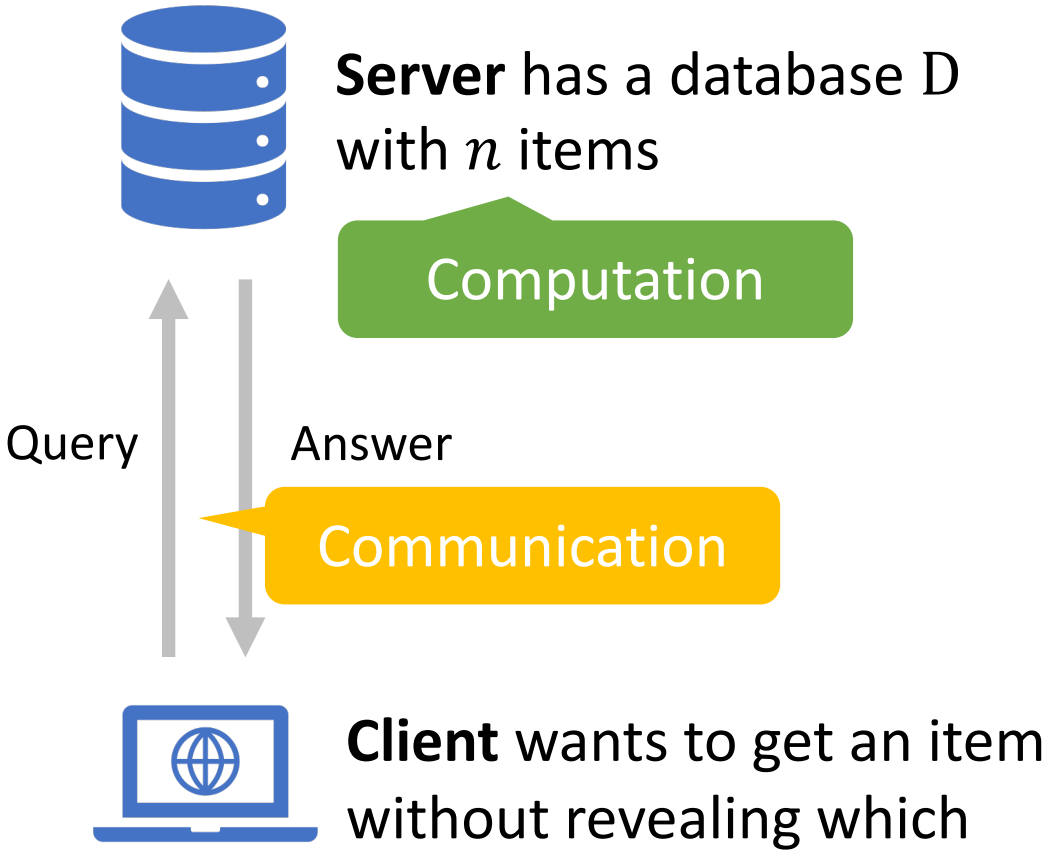
Incremental Offline/Online PIR

Yiping Ma[1] Ke Zhong[1] Tal Rabin[1,2] Sebastian Angel[1,3]

[1]University of Pennsylvania [2]Algorand Foundation [3]Microsoft Research

Private Information Retrieval (PIR)

[CGKS95, KO97]



Applications:

- Anonymous communication
  [PIR-Tor, Sec11], [Pung, OSDI16], [Addra, OSDI21]
- Private reading
  [Popcorn, NSDI16]
- Private search
  [DORY, OSDI20], [Checklist, Sec21]
- …

Private Information Retrieval (PIR)

[CGKS95, KO97]

Server has a database D with n items

Trivial PIR: download D

Prohibitively high cost

Client wants to get an item without revealing which

Private Information Retrieval (PIR)

[CGKS95, KO97]

Server has a database D with n items

Query

Answer

Communication

Sublinear in n [CGKS95, KO97, CG97, CMS99, BI01, …]

Client wants to get an item without revealing which

Private Information Retrieval (PIR)

[CGKS95, KO97]

Server has a database D with n items

Computation

A lower bound [BIM00]:
At least n if no "preprocessing"

Query

Answer

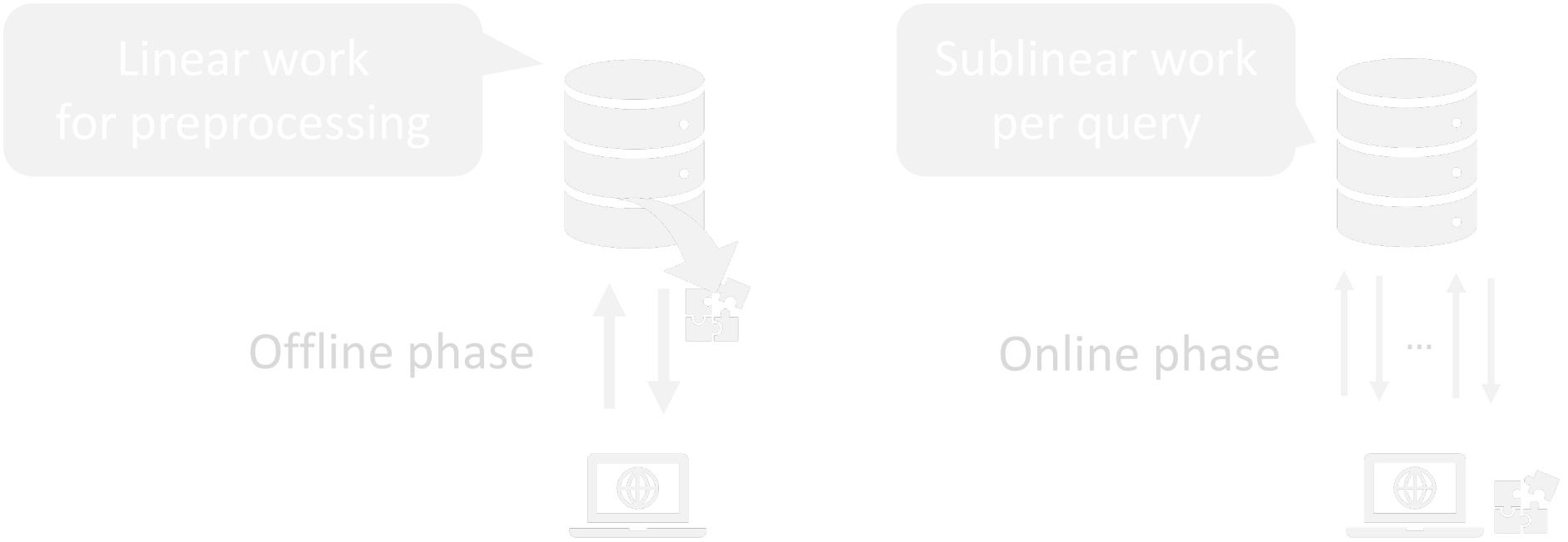
Client wants to get an item without revealing which

PIR with preprocessing

[BIM00, IKOS04, CHR17, BIPW17, HOWW18, PPY18, CK20, SACM21, KC22, CHK22, …]

Precompute answers, encode database, …

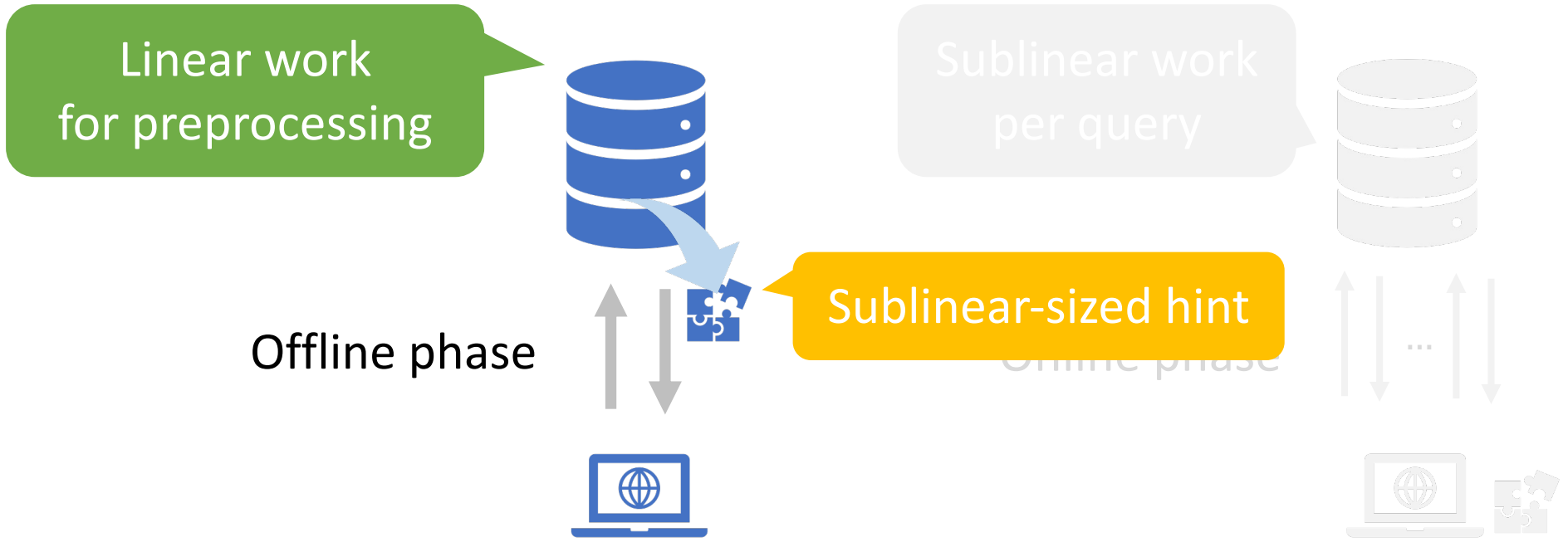
Offline/Online PIR



PIR with preprocessing

[BIM00, IKOS04, CHR17, BIPW17, HOWW18, PPY18, CK20, SACM21, KC22, CHK22, …]

Offline/Online PIR

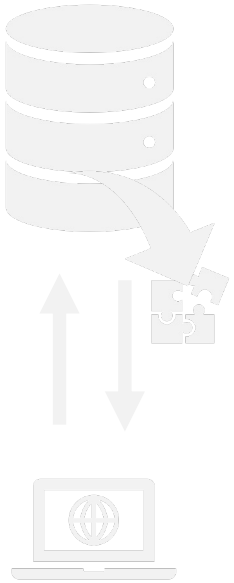


PIR with preprocessing

[BIM00, IKOS04, CHR17, BIPW17, HOWW18, PPY18, CK20, SACM21, KC22, CHK22, …]

Offline/Online PIR

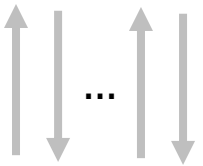
Linear work for preprocessing



Offline phase

Sublinear work per query

Online phase



…

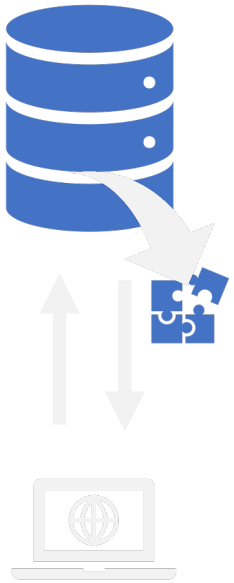
Each query sublinear

PIR with preprocessing

[BIM00, IKOS04, CHR17, BIPW17, HOWW18, PPY18, CK20, SACM21, KC22, CHK22, …]

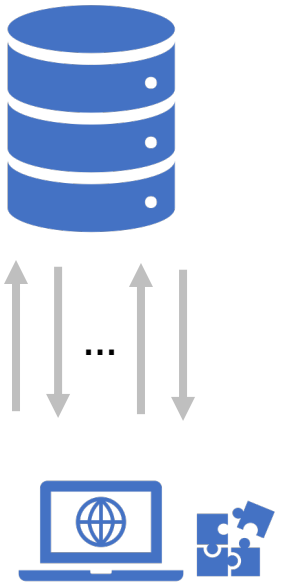
Offline/Online PIR

Linear work
for preprocessing



Offline phase

Sublinear work
per query



Online phase

PIR with preprocessing

[BIM00, IKOS04, CHR17, BIPW17, HOWW18, PPY18, CK20, SACM21, KC22, CHK22, …]

Offline/Online PIR

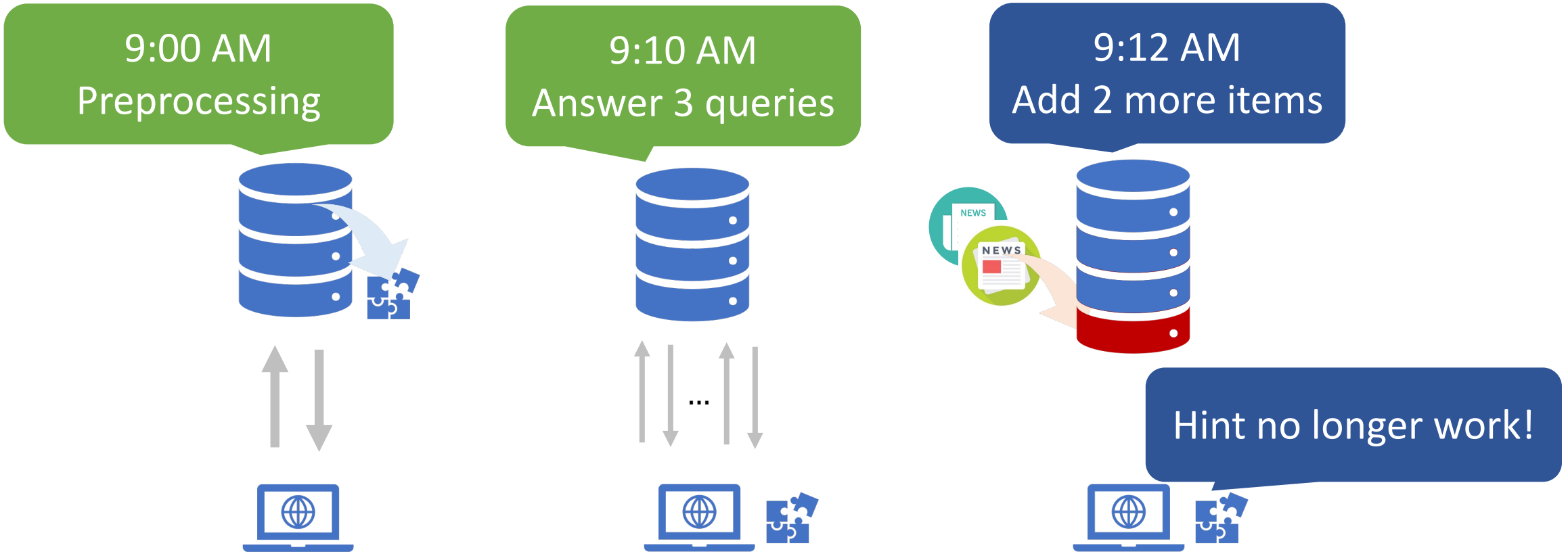
Linear work
for preprocessing

Sublinear work
per query

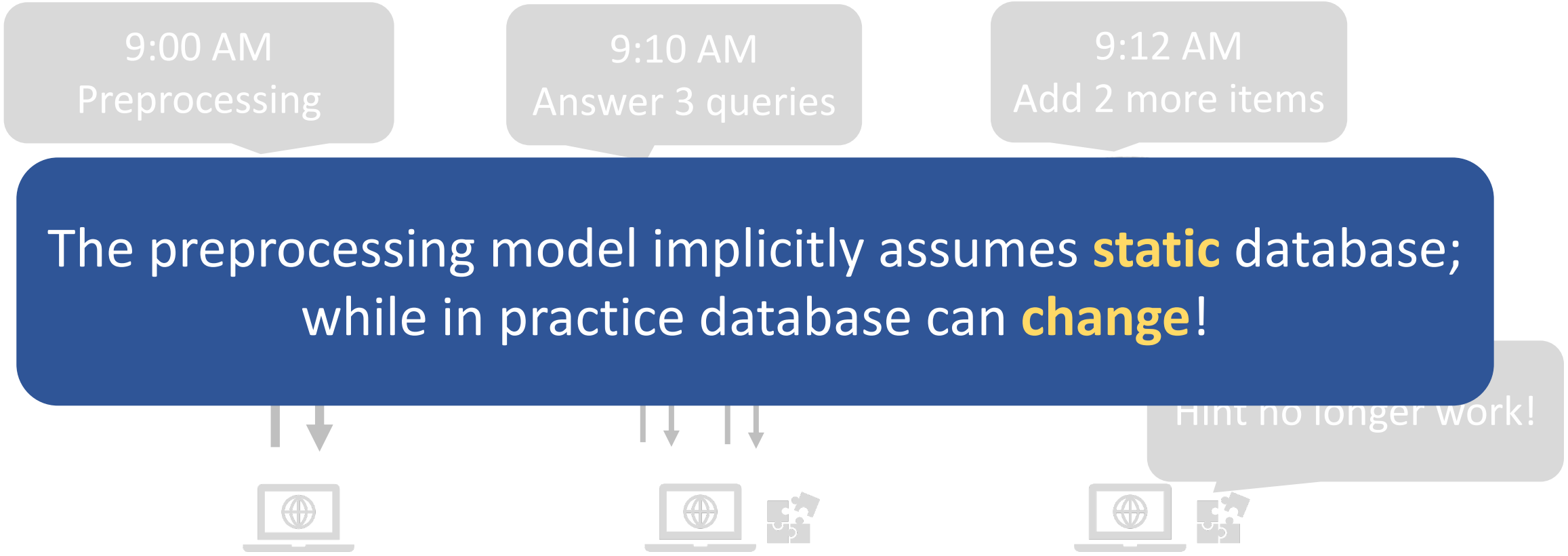
Sublinear communication and sublinear computation!
Are we done?

PIR with preprocessing in applications

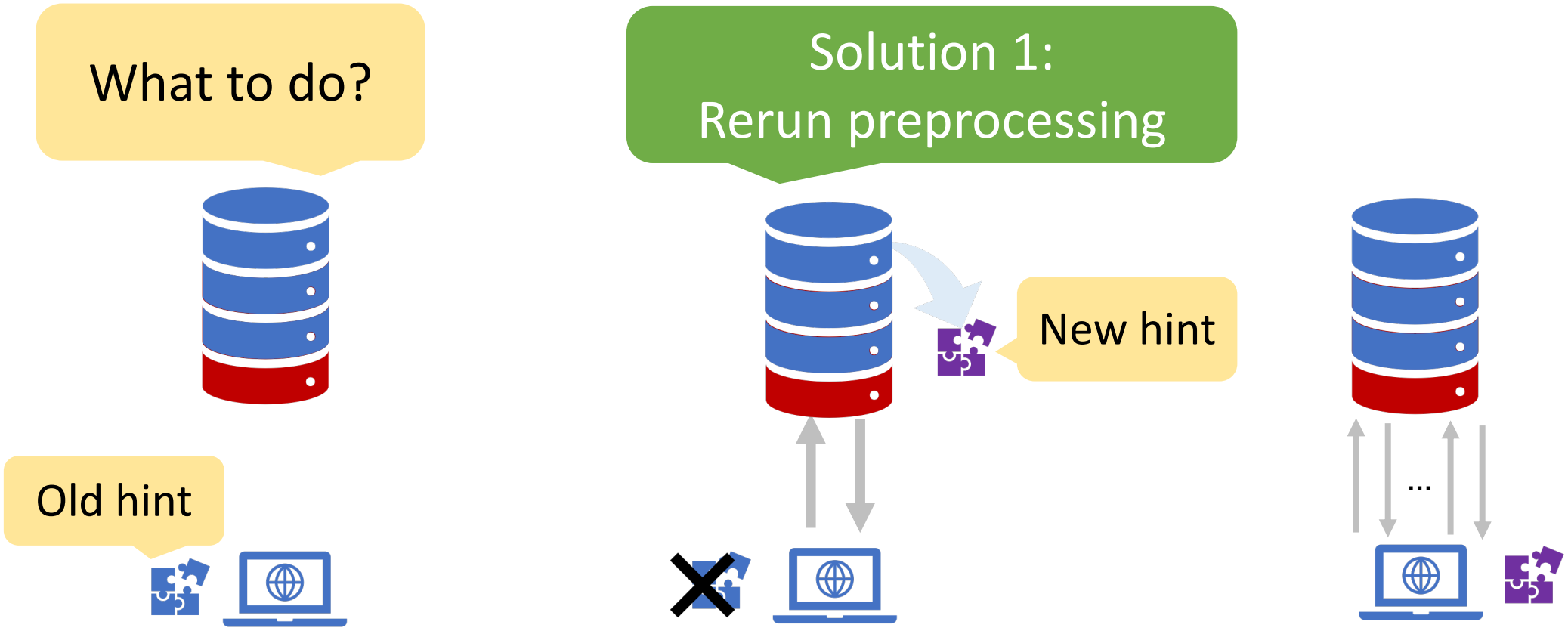
PIR with preprocessing in applications



PIR with preprocessing in applications



PIR with preprocessing in applications



PIR with preprocessing in applications

What to do?

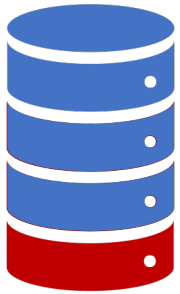
Solution 1:
Rerun preprocessing

For **each** time of change (even small):
server work **linear** in the database size!

Old hint

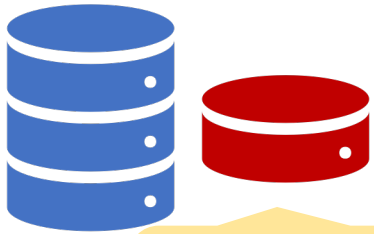
PIR with preprocessing in applications

What to do?



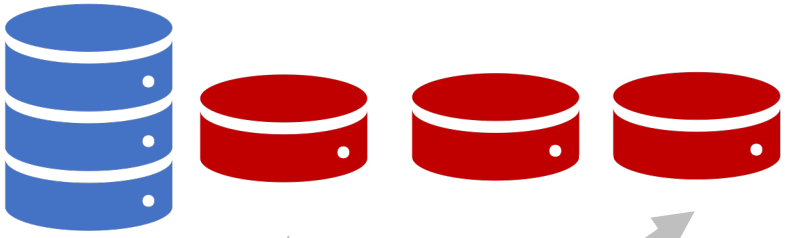
Old hint

Solution 2:
Separate databases



Only linear work
to #additions

Server manages
multiple databases

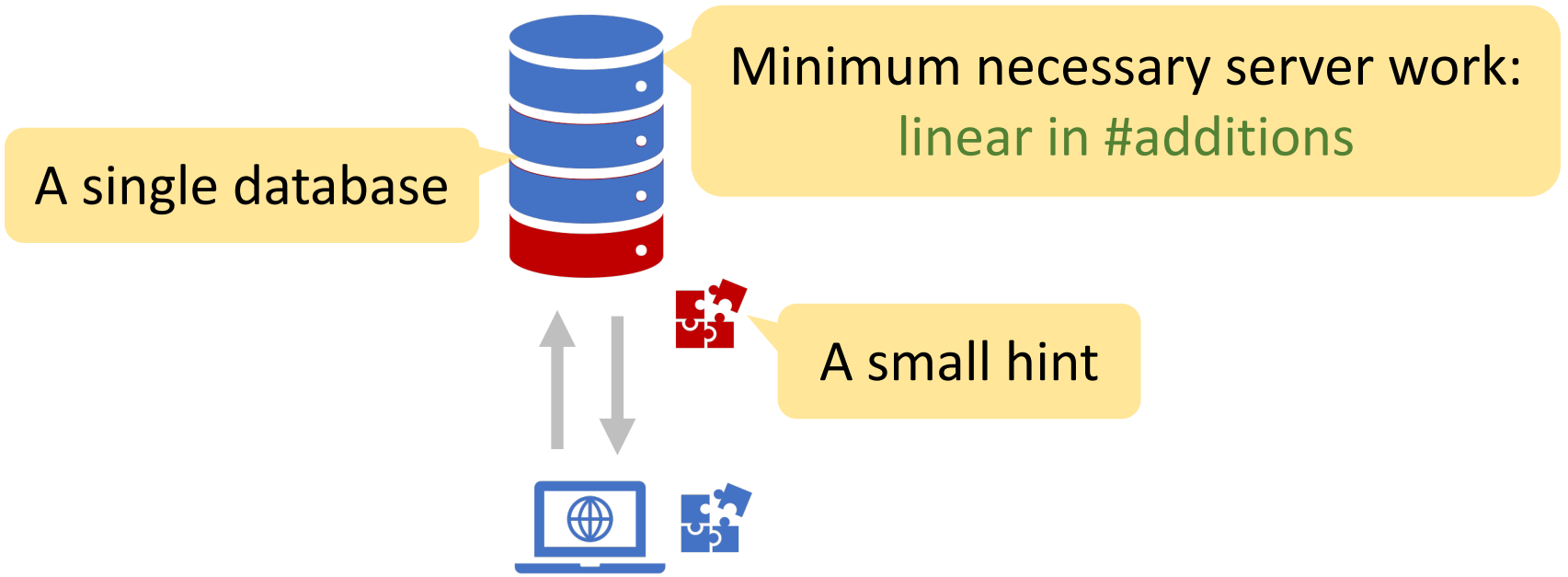


PIR PIR PIR PIR

Client issues many
queries for one item

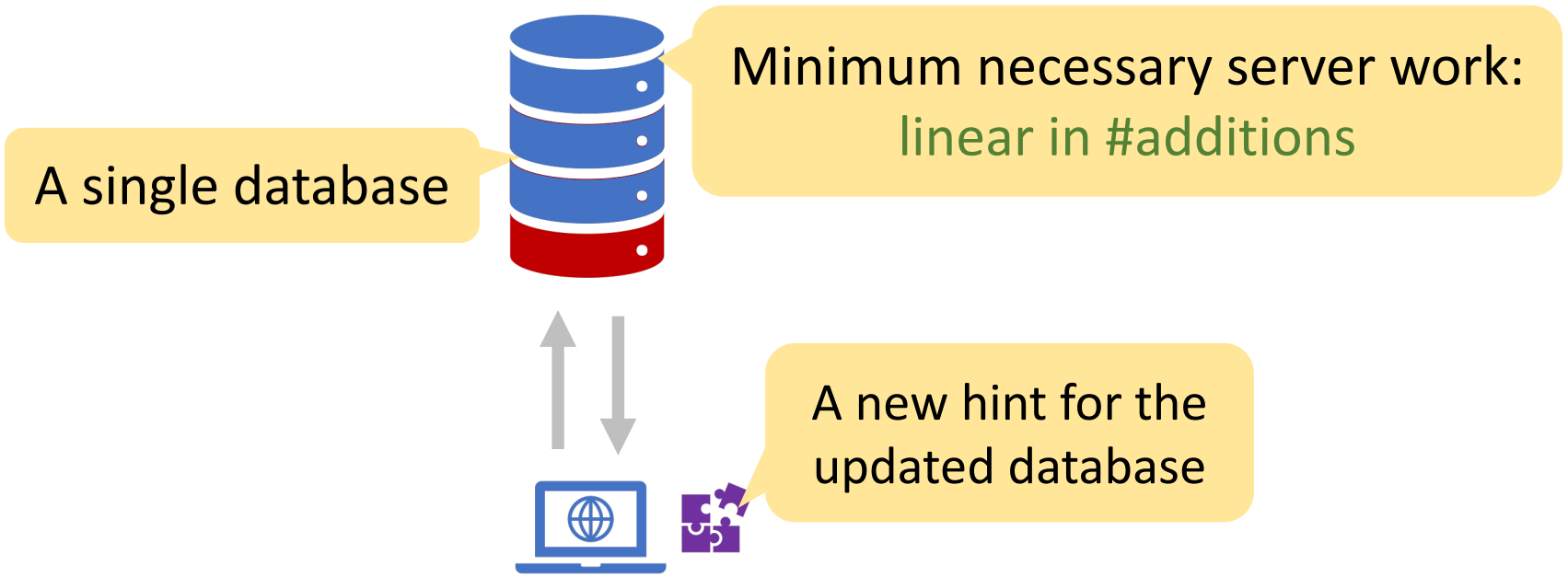
Mutable preprocessing in offline/online model

Our approach to handle dynamic database
preserves all the properties of the solutions for the static database



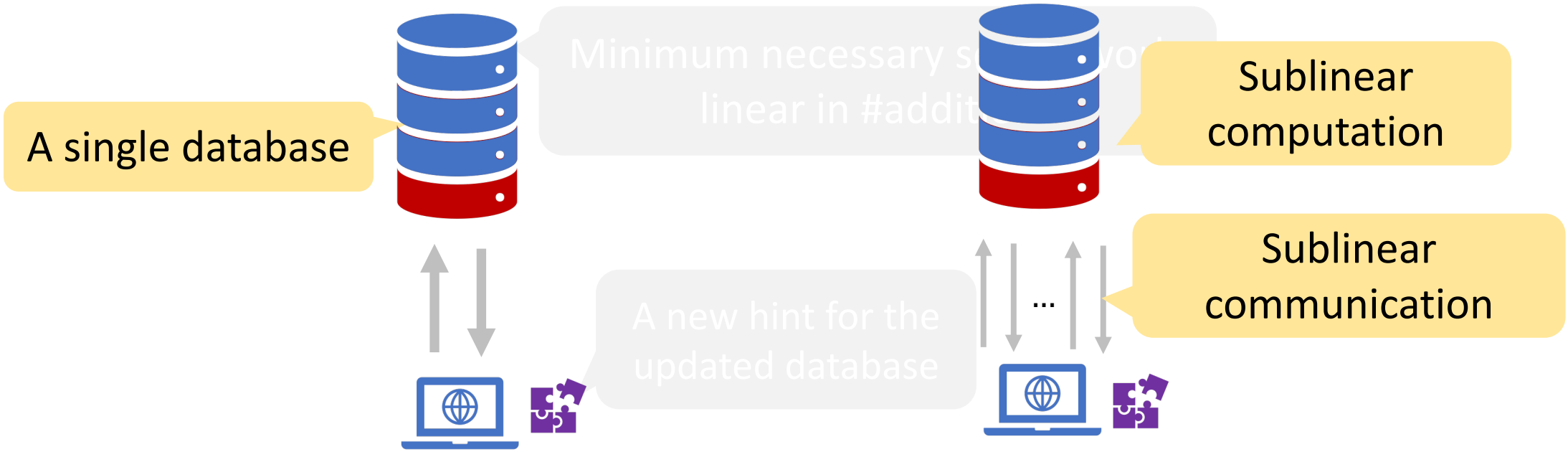
Mutable preprocessing in offline/online model

Our approach to handle dynamic database
preserves all the properties of the solutions for the static database



Mutable preprocessing in offline/online model

Our approach to handle dynamic database
preserves all the properties of the solutions for the static database



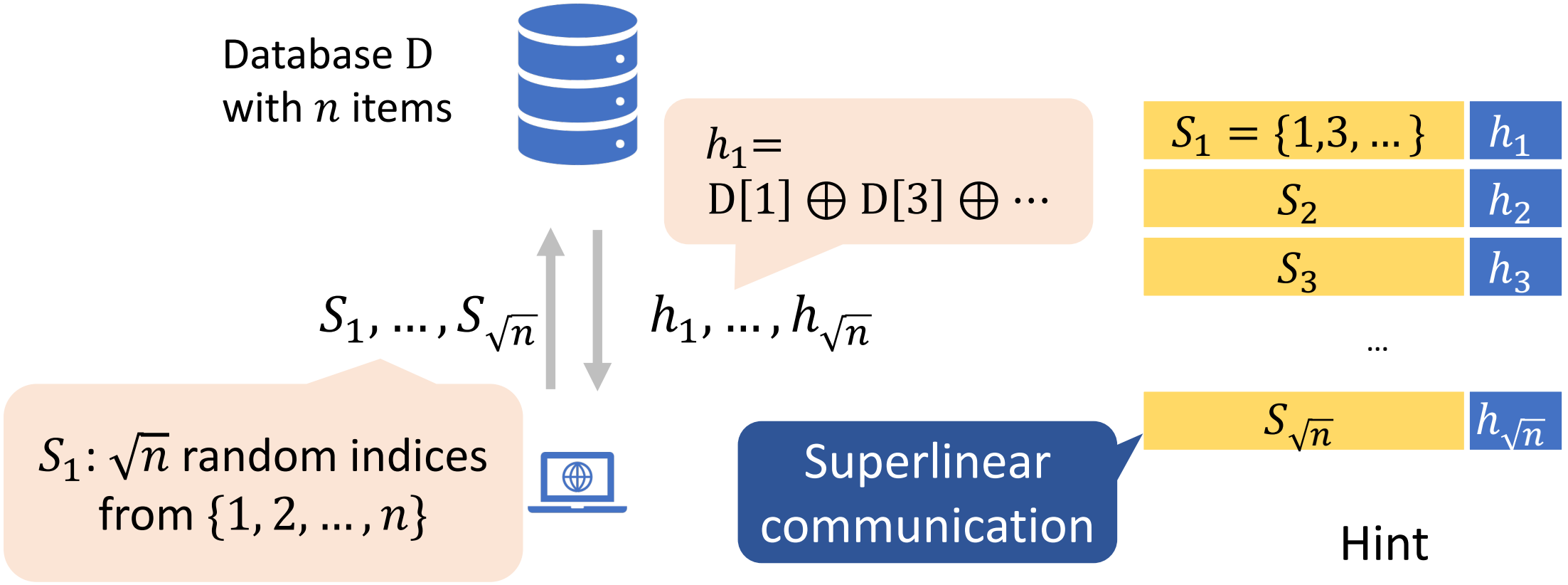
Rest of this talk

- Background on offline/online PIR [CK20] [SACM21]
- Our solution for supporting mutable preprocessing
- Experimental evaluation

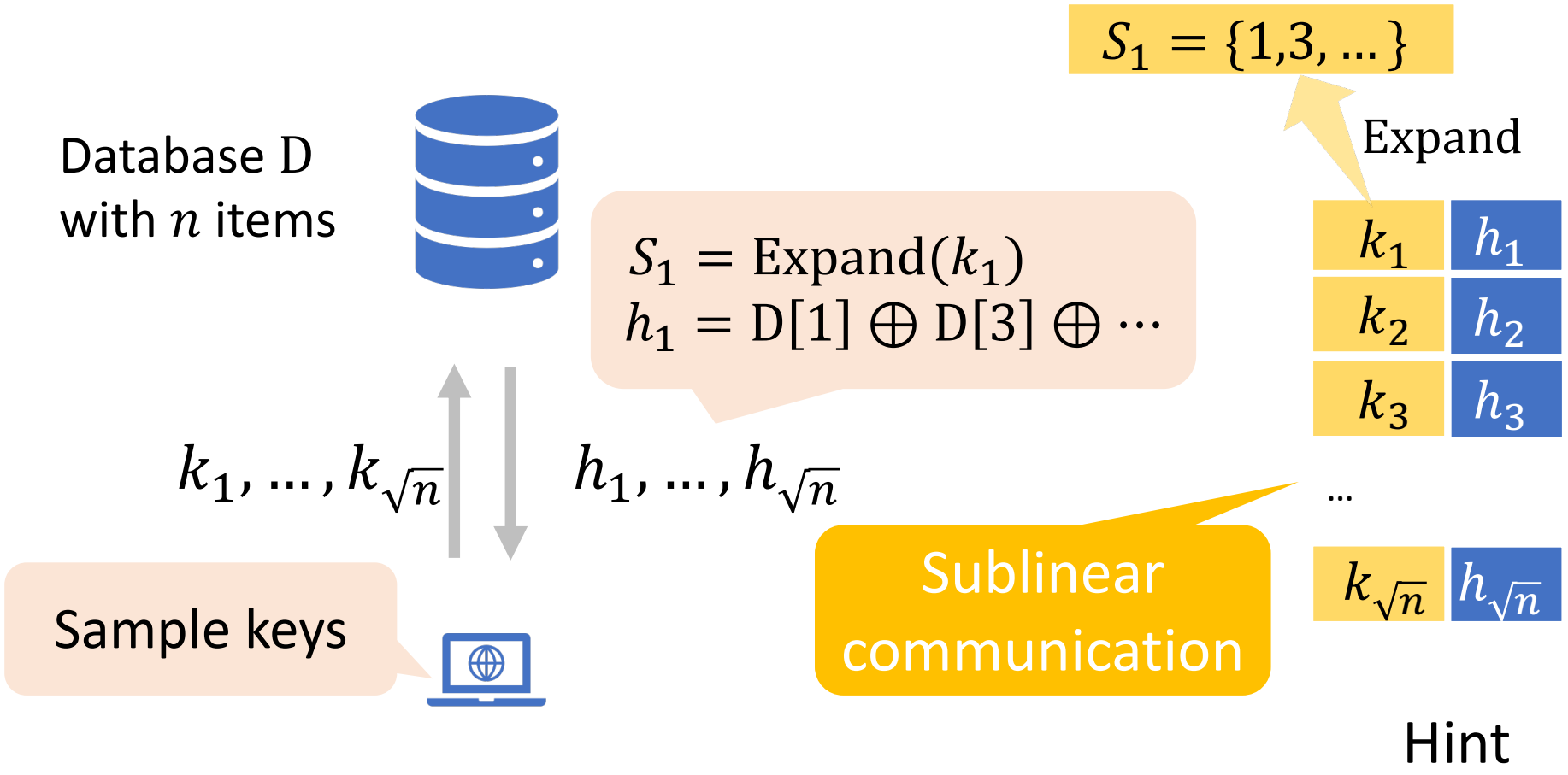
Rest of this talk

- Background on offline/online PIR [CK20] [SACM21]
- Our solution for supporting mutable preprocessing
- Experimental evaluation

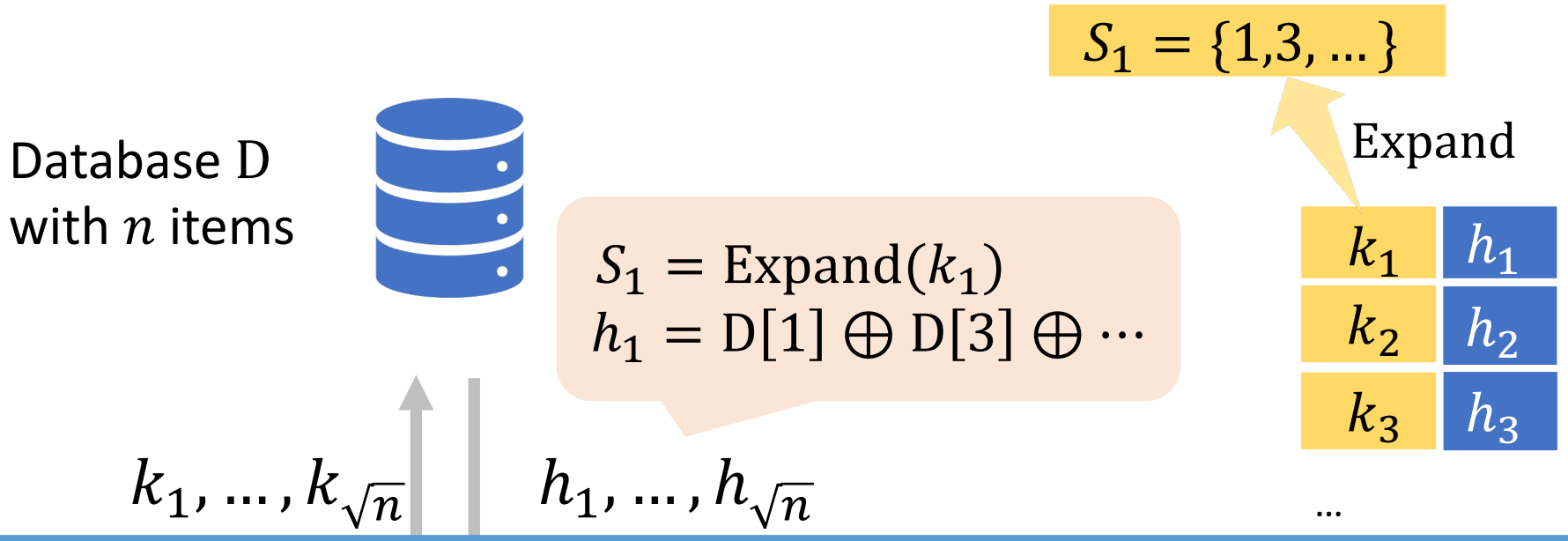
Background on offline/online PIR [CK20]



Background on offline/online PIR [CK20]



Background on offline/online PIR [CK20]

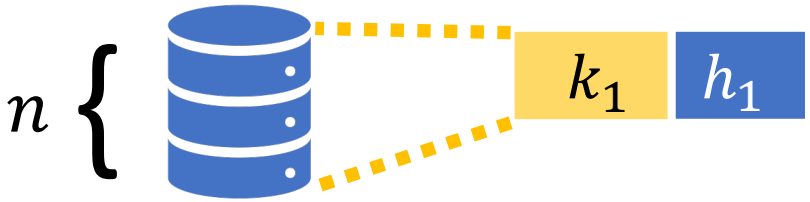


To ensure the scheme is correct and private,

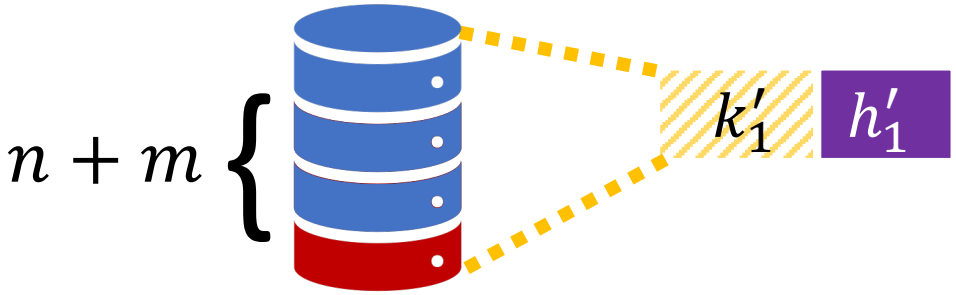
- Each set should be random over all indices of the database
- The parity should match the set

Mutable preprocessing

Offline phase
(preprocessing for D)

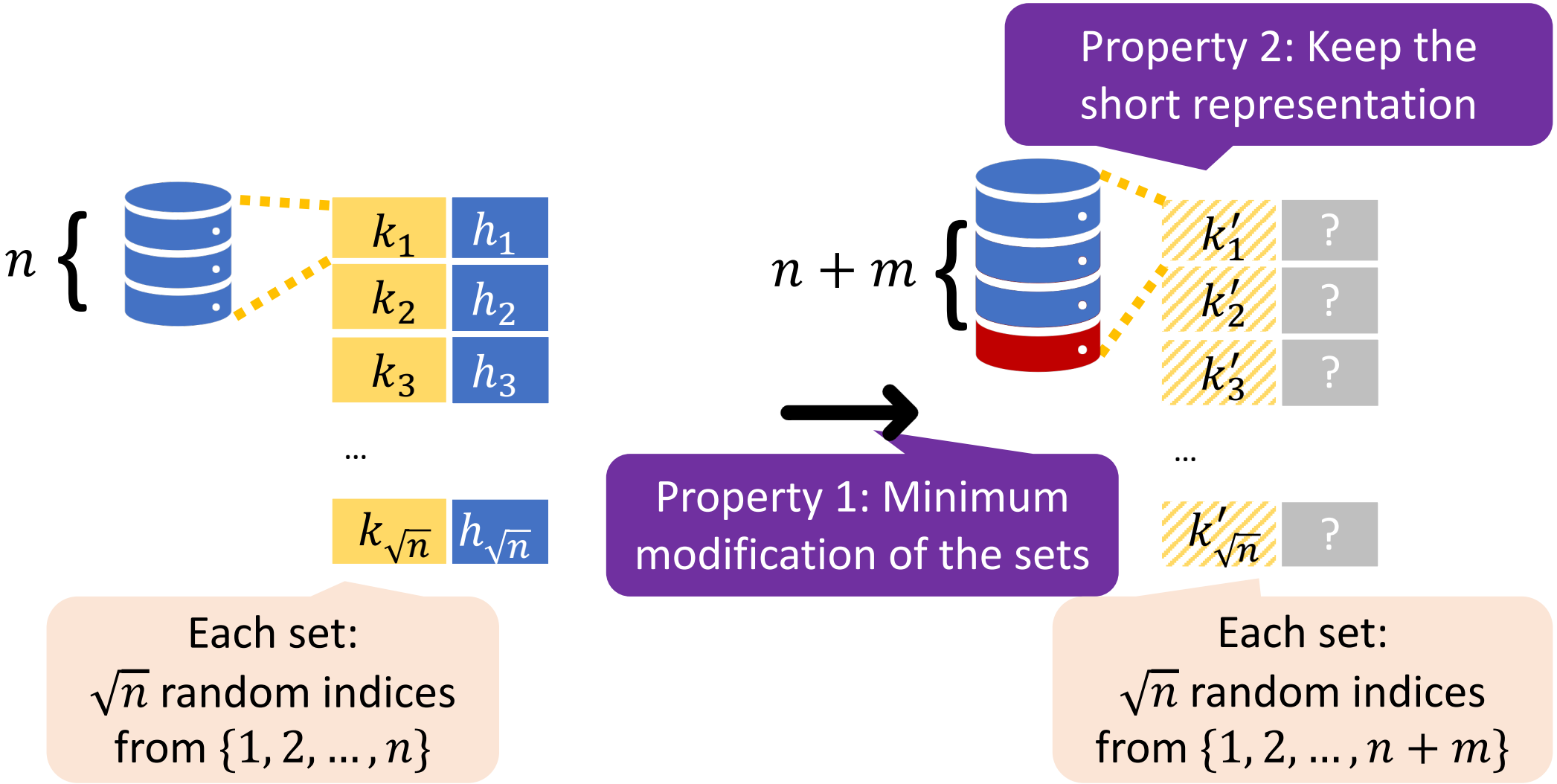


Hint updates
from D to D′

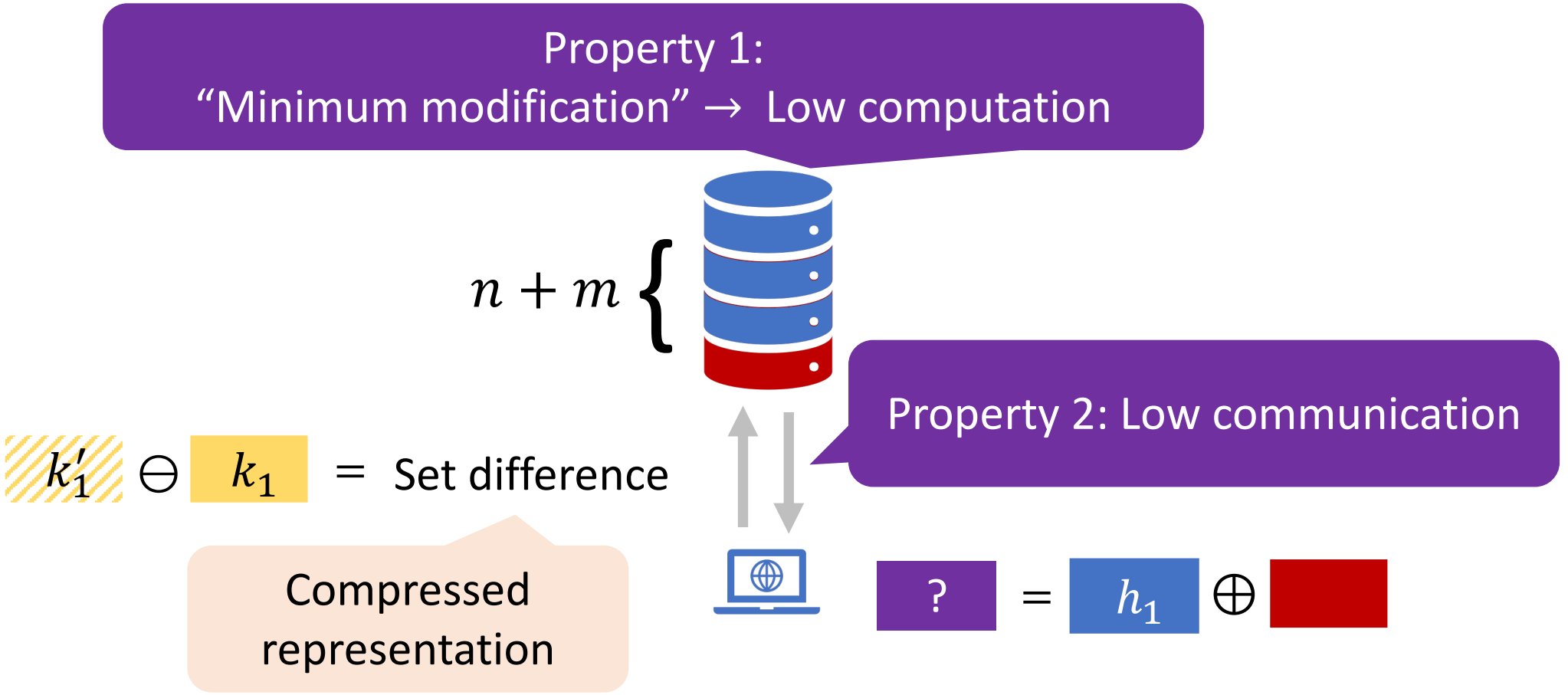


- How to update each **key (set)** to be random over the new range?
- How to update each **parity** to match the key?

Contribution 1: a randomized algorithm to update keys



Contribution 2: a procedure to update the parities

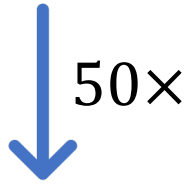


Evaluation

How does our construction save **server cost**?

Results for adding 1% data:

Database* size	2^{16}	2^{18}	2^{20}
Offline phase (sec)	3.64	14.52	58.67
Hint update phase (sec)	0.07	0.25	1.03



*Each data item 32 bytes, results run on a machine with 2 GHz processor and 64 GB RAM, single thread

Takeaways

- Preprocessing model assumes static database:
  - Don’t work well when database can change
- Techniques to make preprocessing mutable for offline/online PIR
  - Server cost proportional to #changes
  - Work for applications where changes are frequent but small
- More evaluation results for Tor application in the paper

Paper is available at https://eprint.iacr.org/2021/1438
Code is available at https://github.com/eniac/incpir

Thank you!