SECRECY: SECURE COLLABORATIVE ANALYTICS IN UNTRUSTED CLOUDS

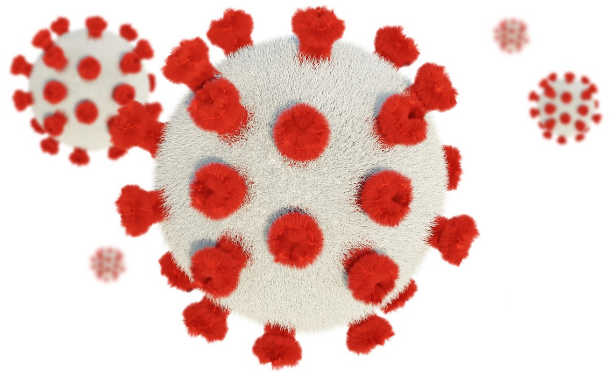
John Liagouris, Vasiliki Kalavri, Muhammad Faisal, Mayank Varia
Boston University

NSDI'23

18 April 2023

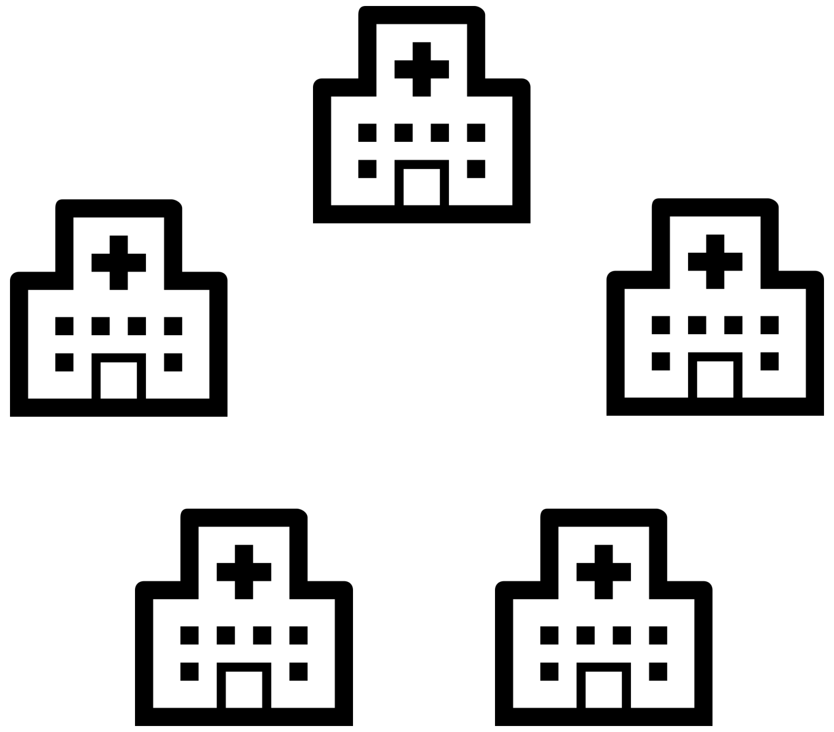
MOTIVATION: SECURE COLLABORATIVE ANALYTICS

Medical Studies

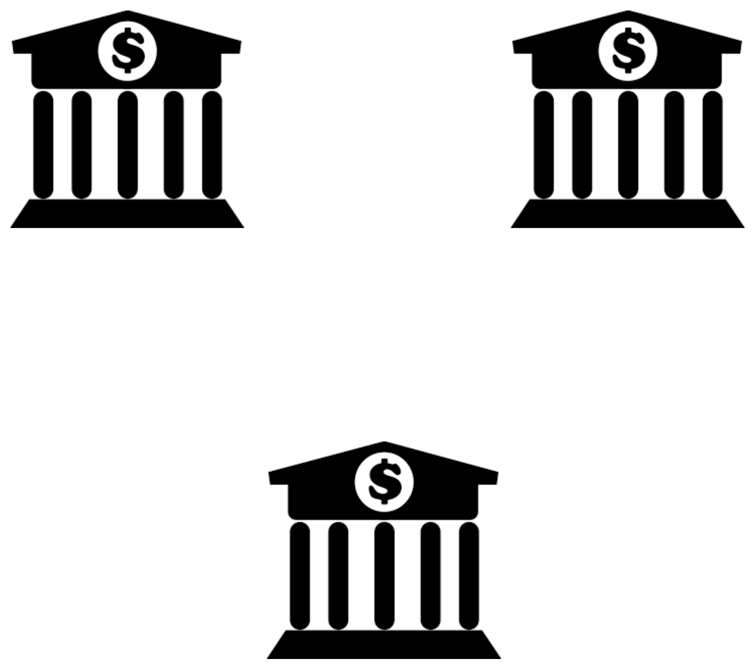


Market Analyses

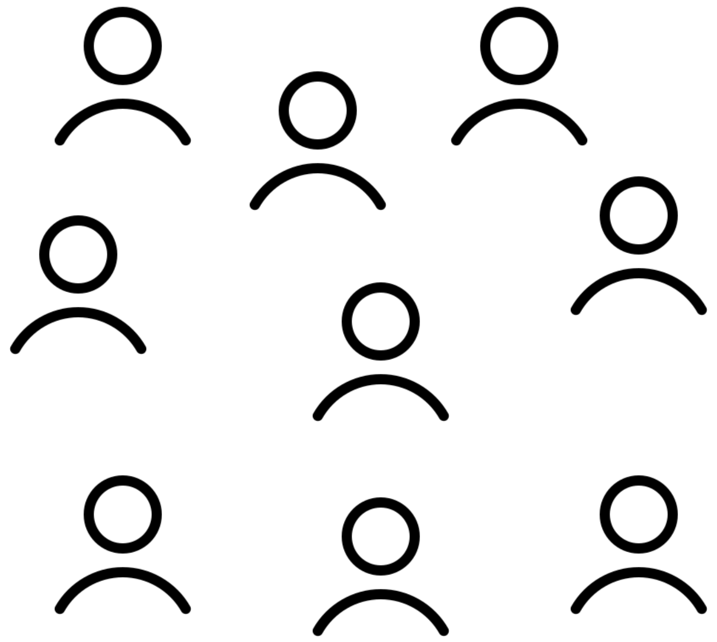
Privacy-preserving advertising



Healthcare providers



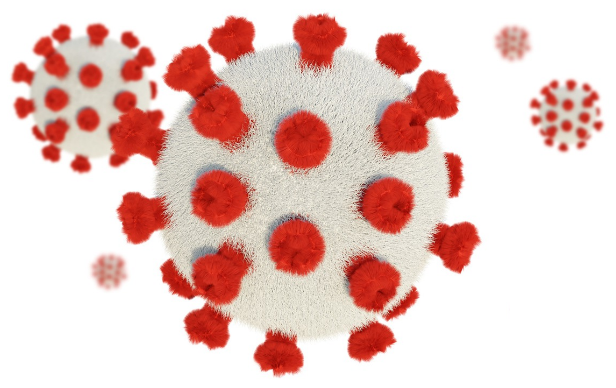
Credit score agencies

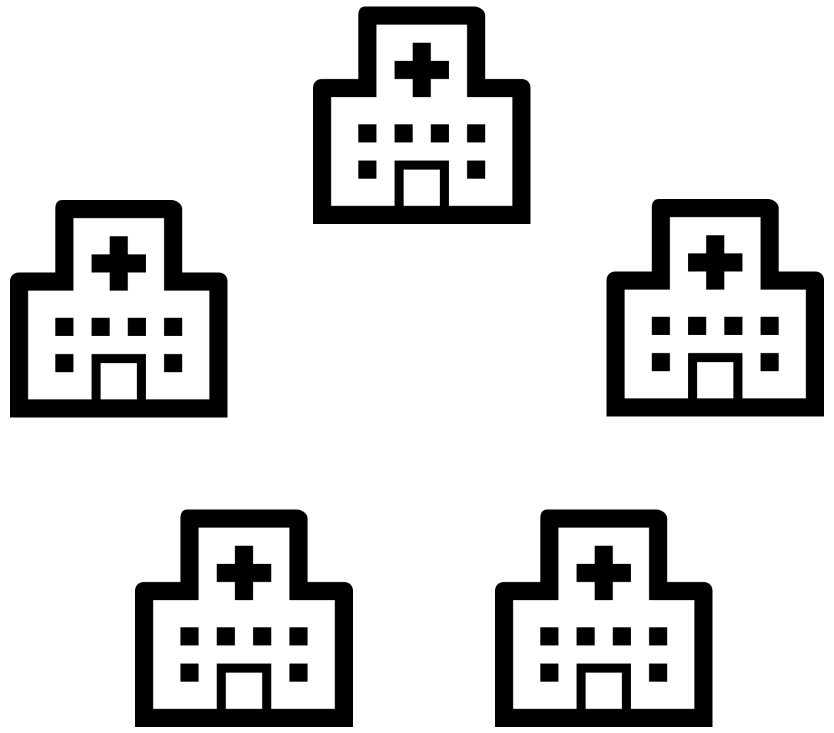


Web users

MOTIVATION: SECURE COLLABORATIVE ANALYTICS

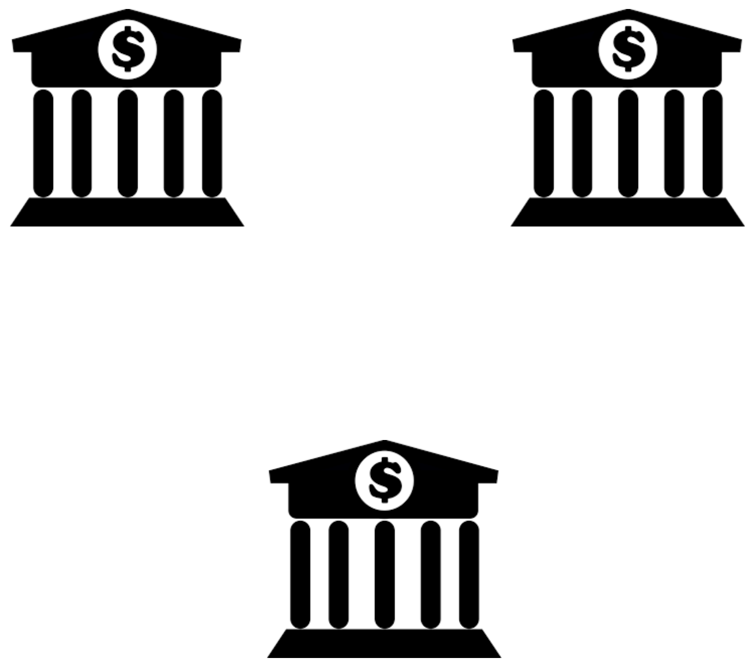
Medical Studies





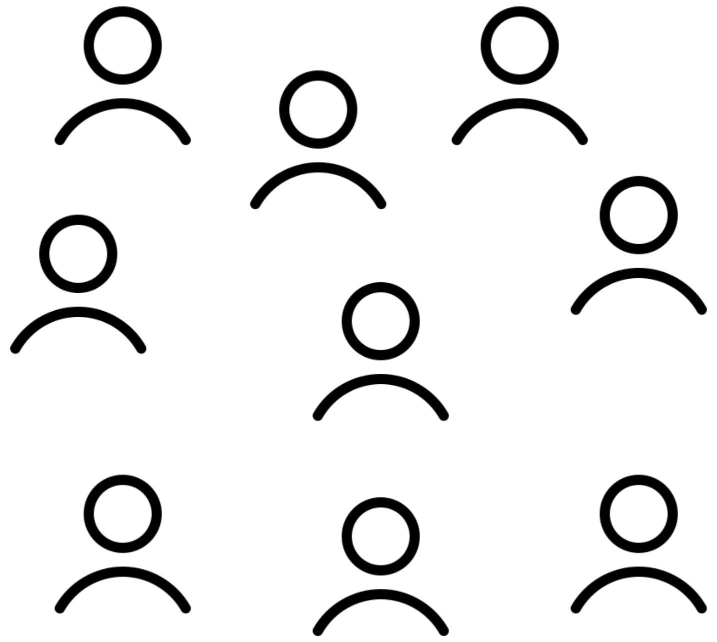
Healthcare providers

Market Analyses



Credit score agencies

Privacy-preserving advertising



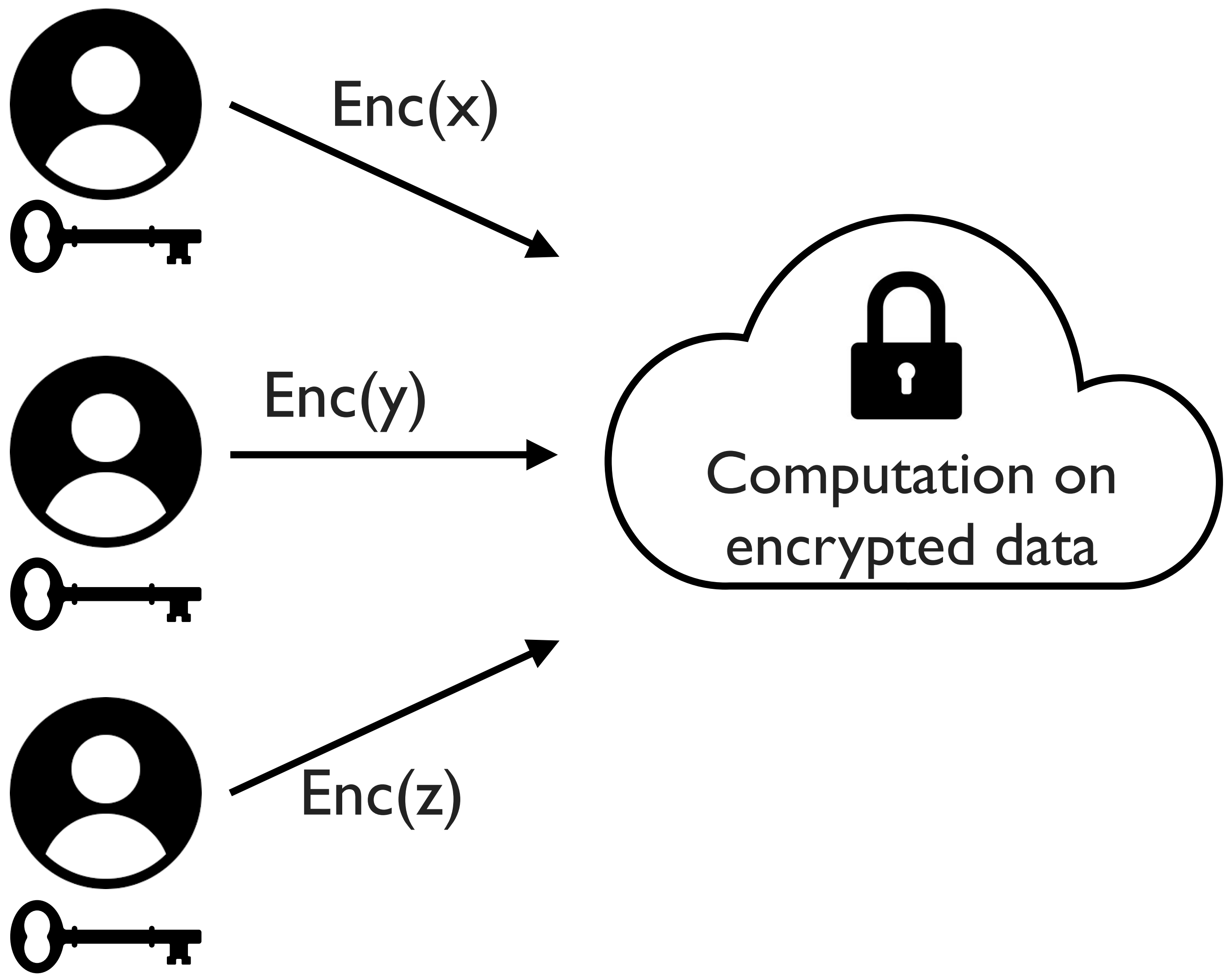
Web users

Requirements:

- ***No information leakage to untrusted entities***
- ***No reliance on trusted resources***
- ***Relational analytics***
- ***Practical performance***

APPROACHES TO SECURE COLLABORATIVE ANALYTICS

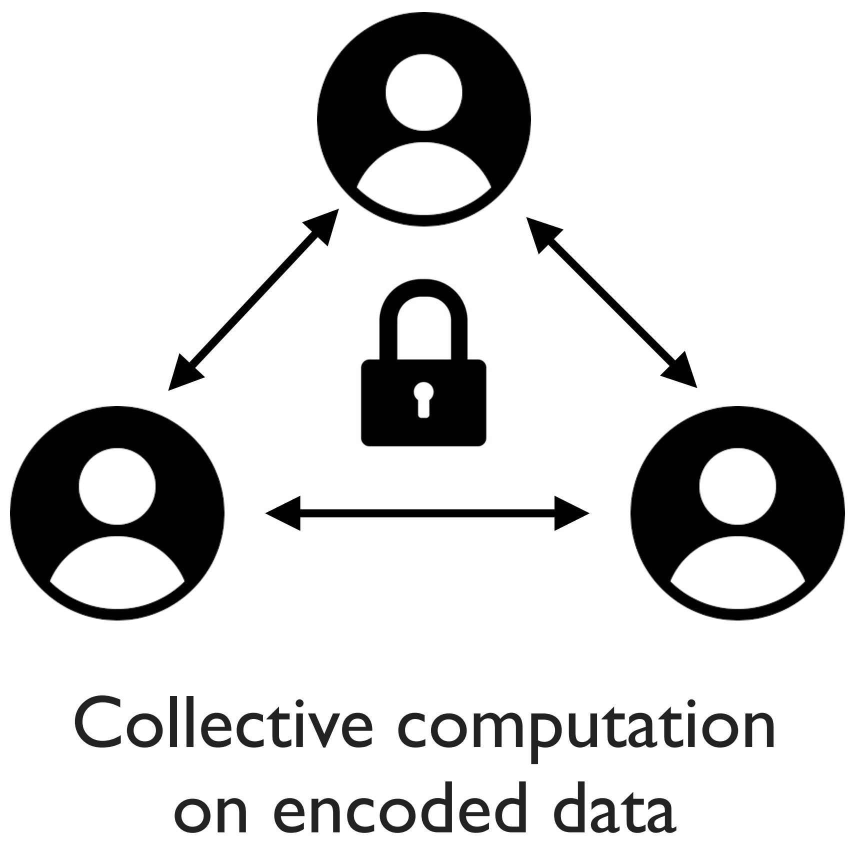
Fully Homomorphic Encryption (FHE)



Security via homomorphic encryption

(very high computational cost)

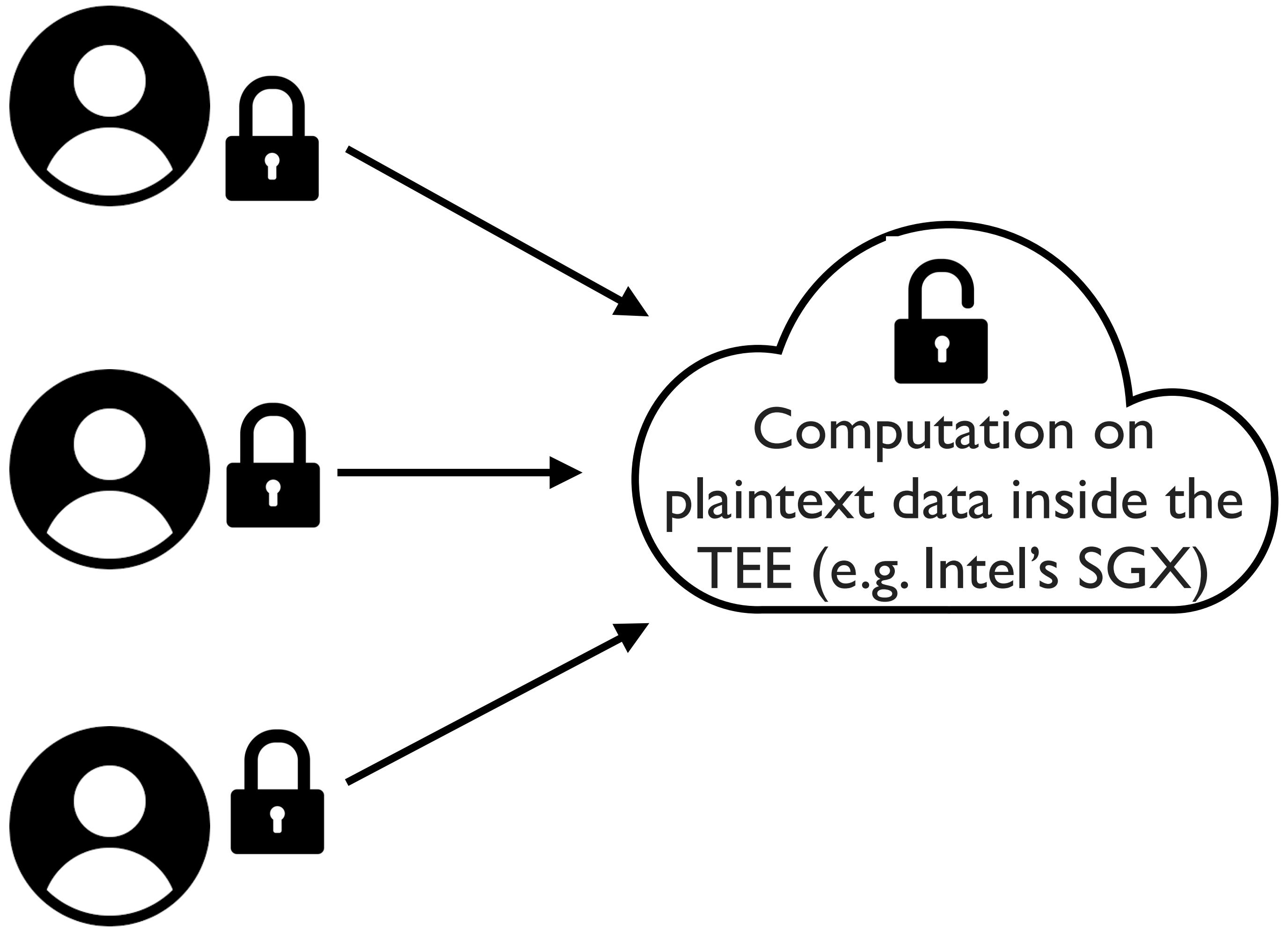
Secure Multi-Party Computation (MPC)



Security via decentralized trust

(high communication cost)

Trusted Execution Environments (TEEs)



Security via physically protected HW

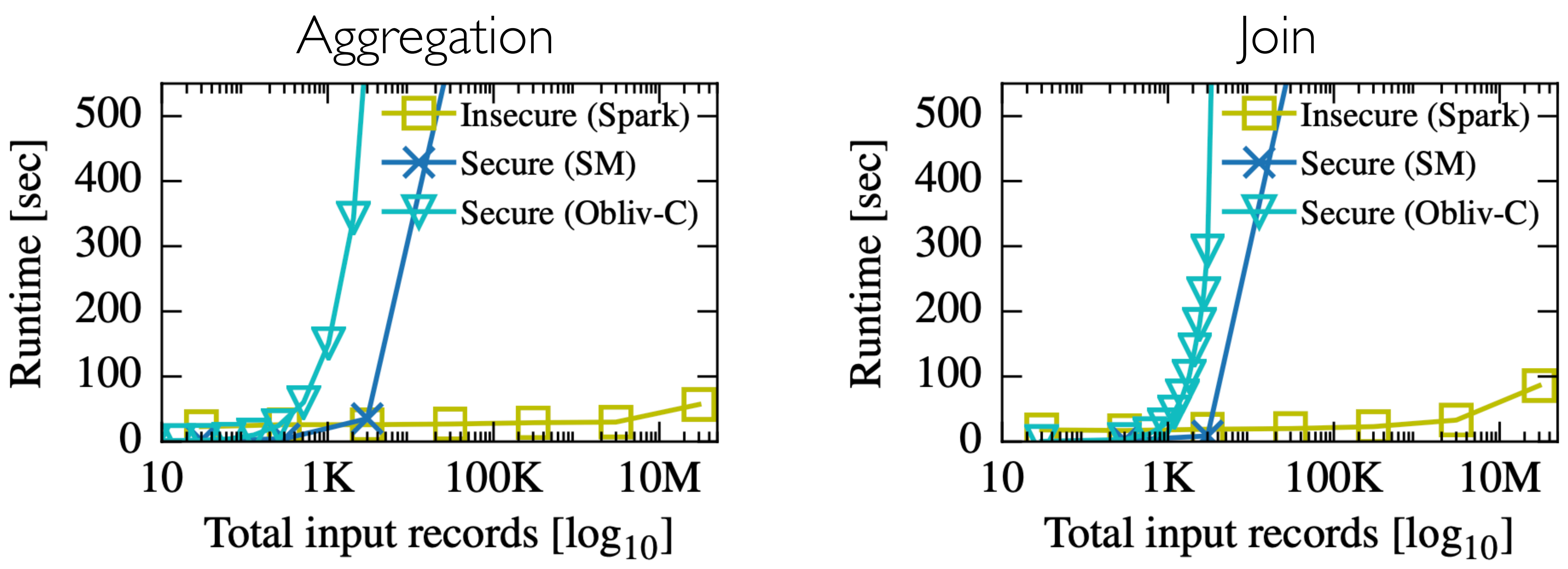
(prone to side-channel attacks)

CHALLENGE: HOW TO REDUCE THE MPC COST?

"Running the query entirely under MPC […] ***fails to scale beyond 3,000 total records****…"*

"Computing a function f on millions of client inputs […] could potentially take an ***astronomical amount of time*** *in a full MPC."*

"The primary source of the slowdown arises from their join operators that have ***hundreds of input tuples****…"*



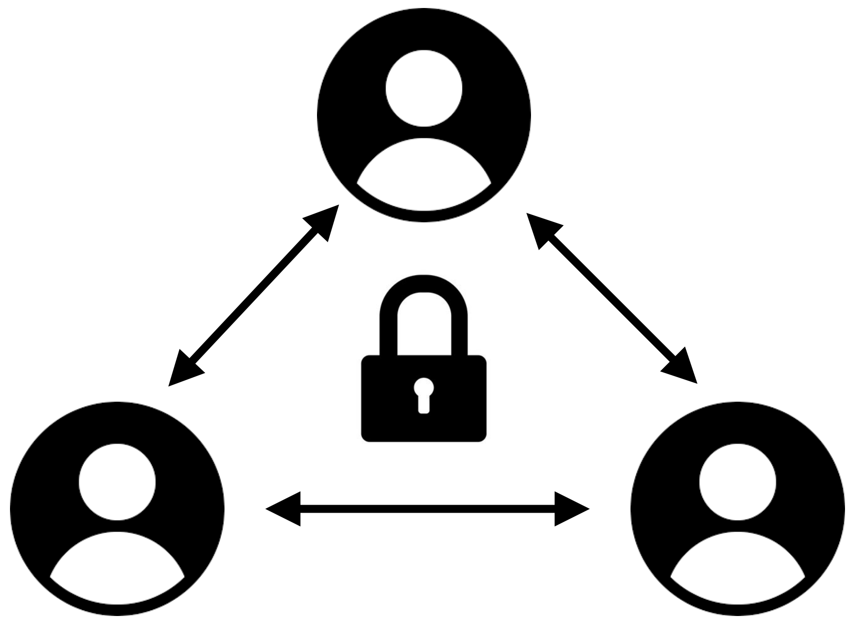
Plaintext	Secure	Slowdown
158	253,894	1,609X
165	159,145	967X
193	8,195,317	43,337X

[1] N. Volgushev, M. Schwarzkopf, B. Getchell, M. Varia, A. Lapets, and A. Bestavros. *Conclave: secure multi-party computation on big data*. EuroSys, 2019.

[2] J. Bater, G. Elliott, C. Eggen, S. Goel, A. N. Kho, and J. Rogers. *SMCQL: secure querying for federated databases*. PVLDB, 10(6):673–684, 2017.

[3] H. Corrigan-Gibbs and D. Boneh. *Prio: Private, Robust, and Scalable Computation of Aggregate Statistics*, NSDI, 2017.

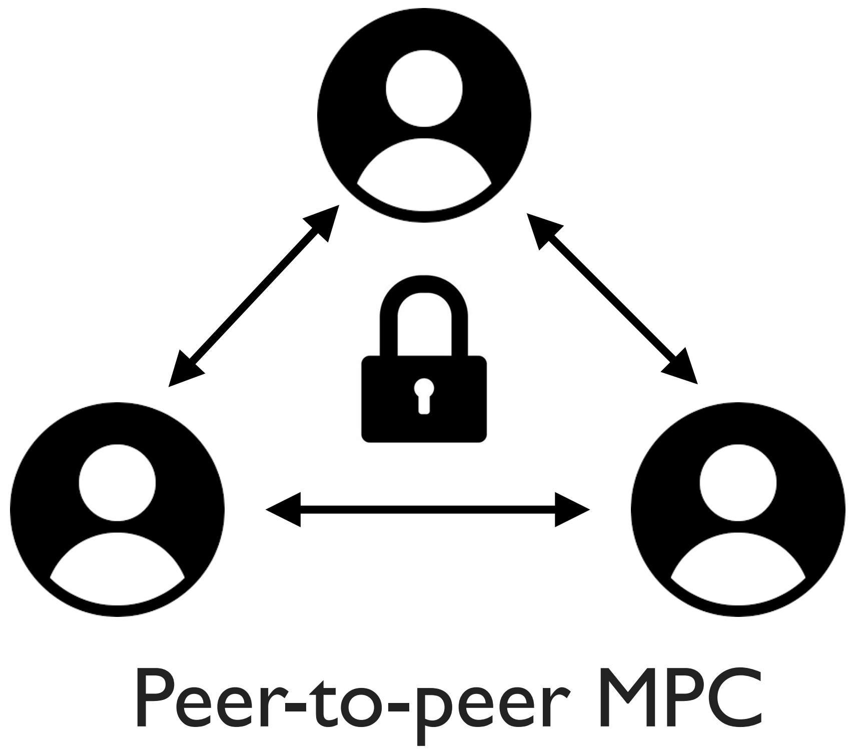
PRIOR WORK ON RELATIONAL MPC



Data owners act as computing parties using trusted resources

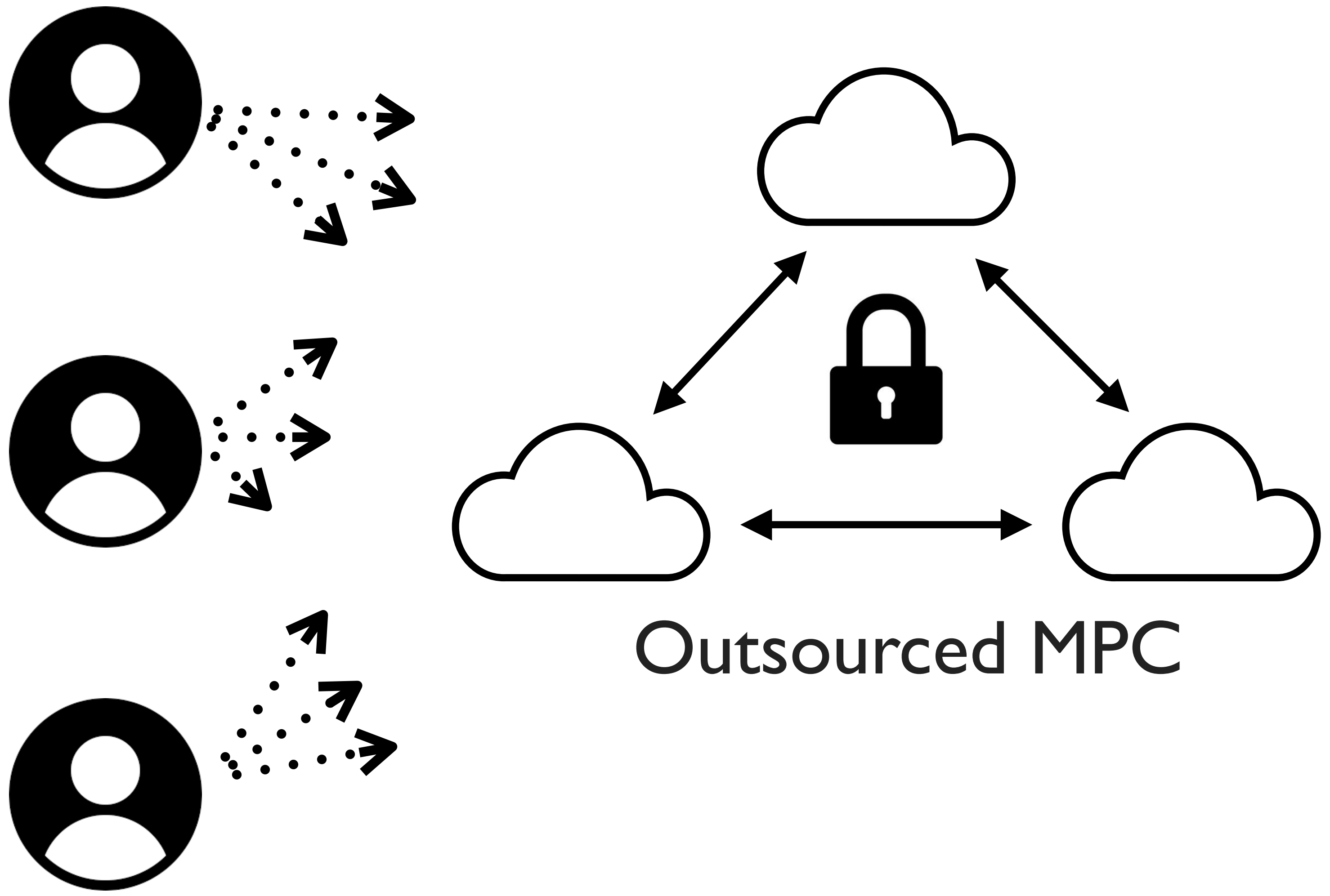
- Data owners may not have domain expertise or private infrastructure
- MPC does not scale well with the number of data owners

OUR FOCUS: OPTIMIZE MPC IN THE CLOUD



Data owners act as computing parties using trusted resources

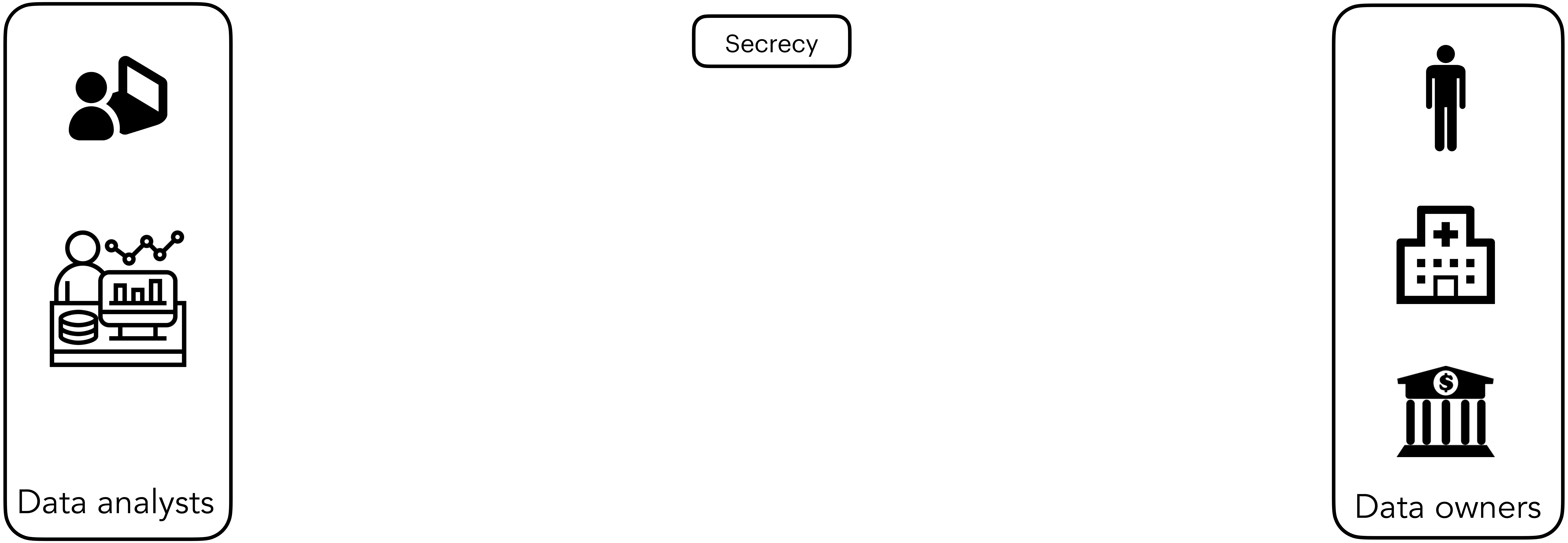
- Data owners may not have domain expertise or private infrastructure
- MPC does not scale well with the number of data owners



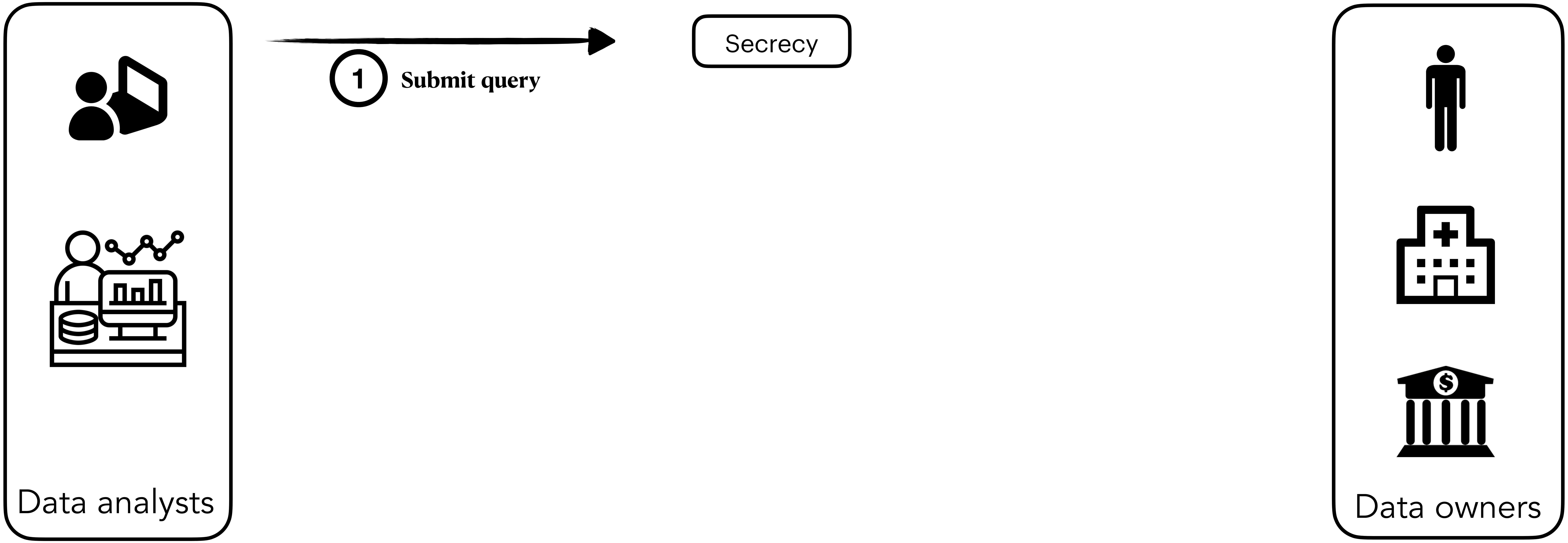
Data owners outsource secret shares of their data to untrusted third parties

- Data owners can use untrusted cloud resources on demand
- A small number of third parties can support a large number of data owners

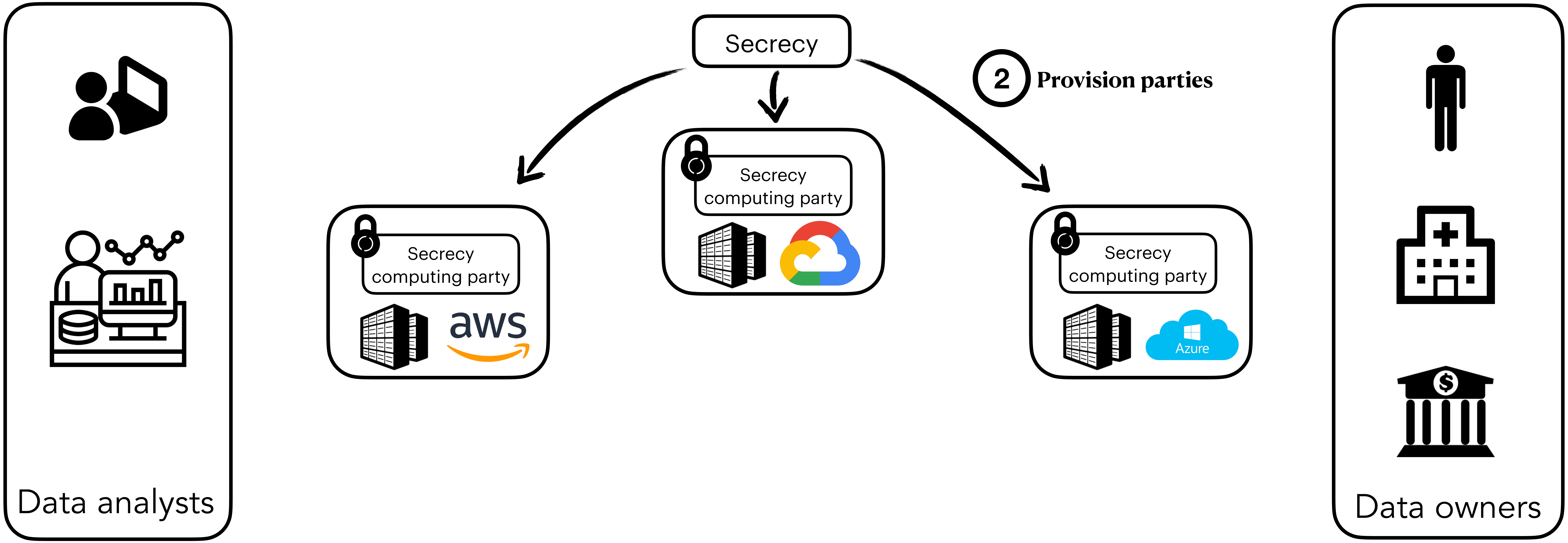
SECRECY AS A SERVICE



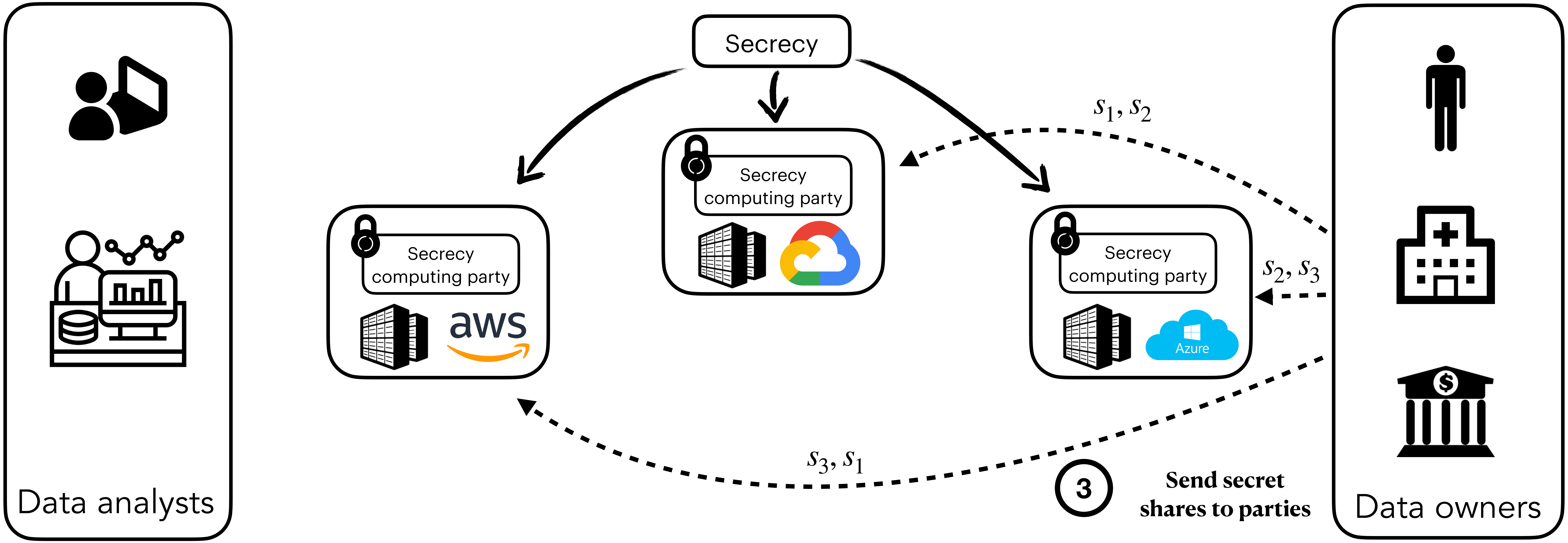
SECRECY AS A SERVICE



SECRECY AS A SERVICE



SECRECY AS A SERVICE



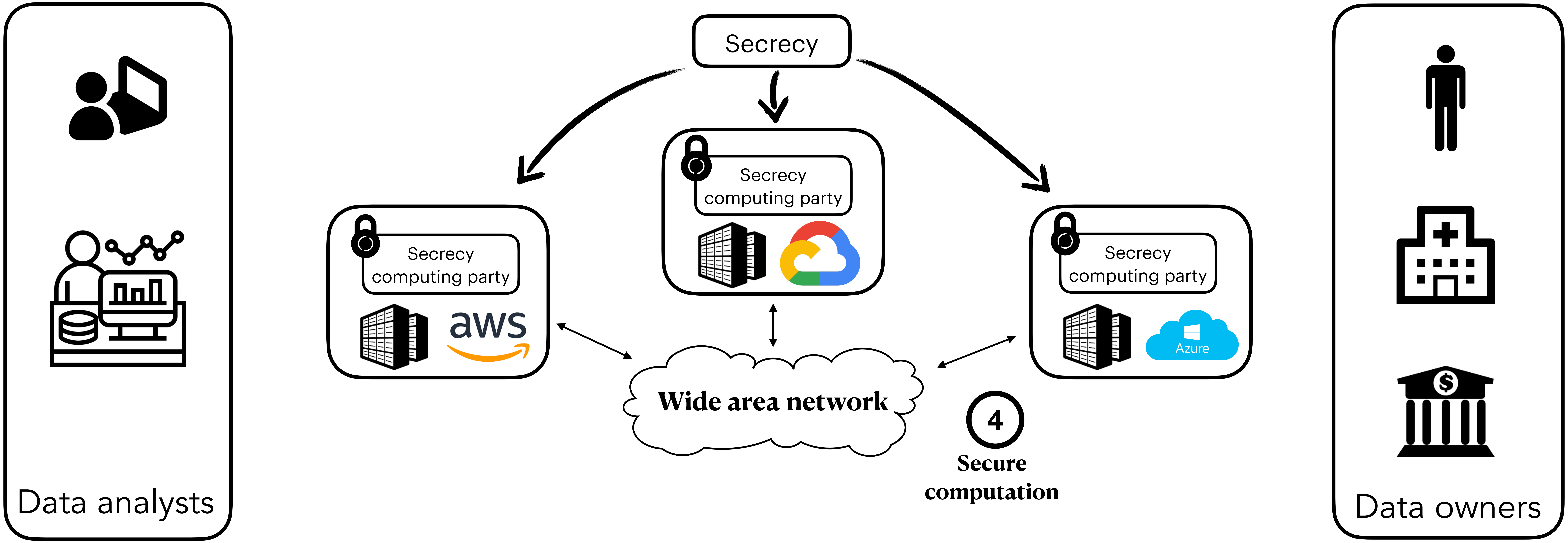
Arithmetic sharing: $s = s_1 + s_2 + s_3 \pmod{2^k}$

(for k-bit integers)

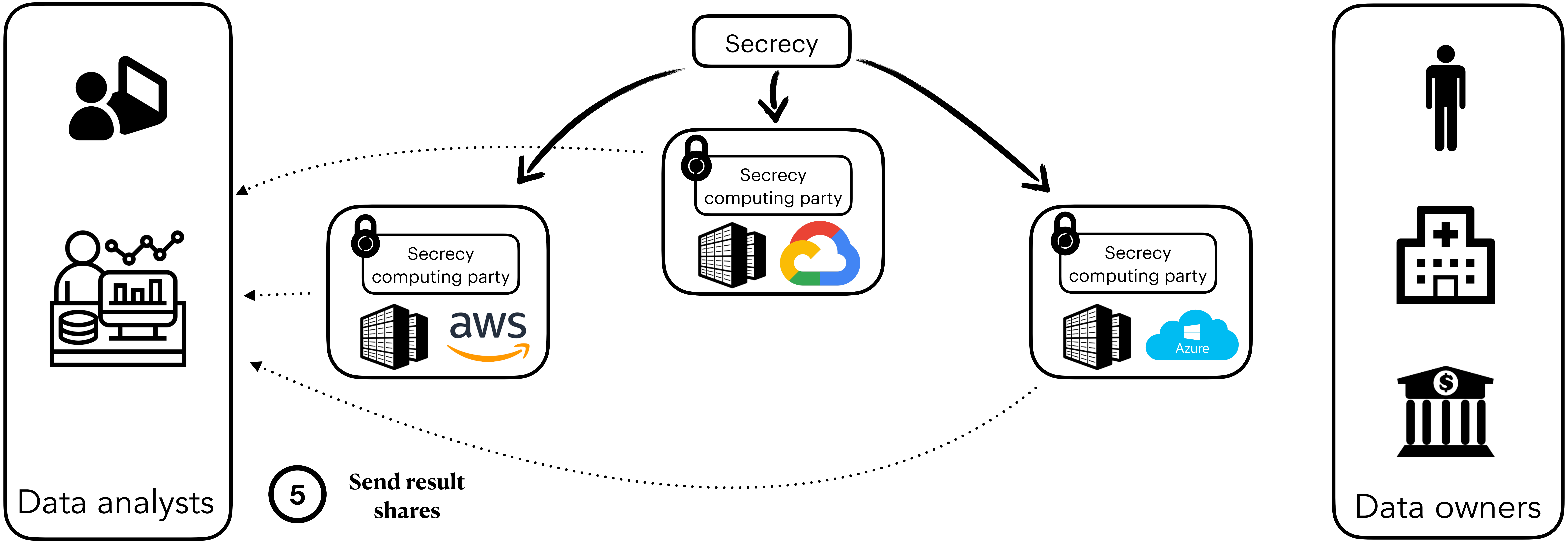
Boolean sharing: $s = s_1 \oplus s_2 \oplus s_3$

(for k-bit strings)

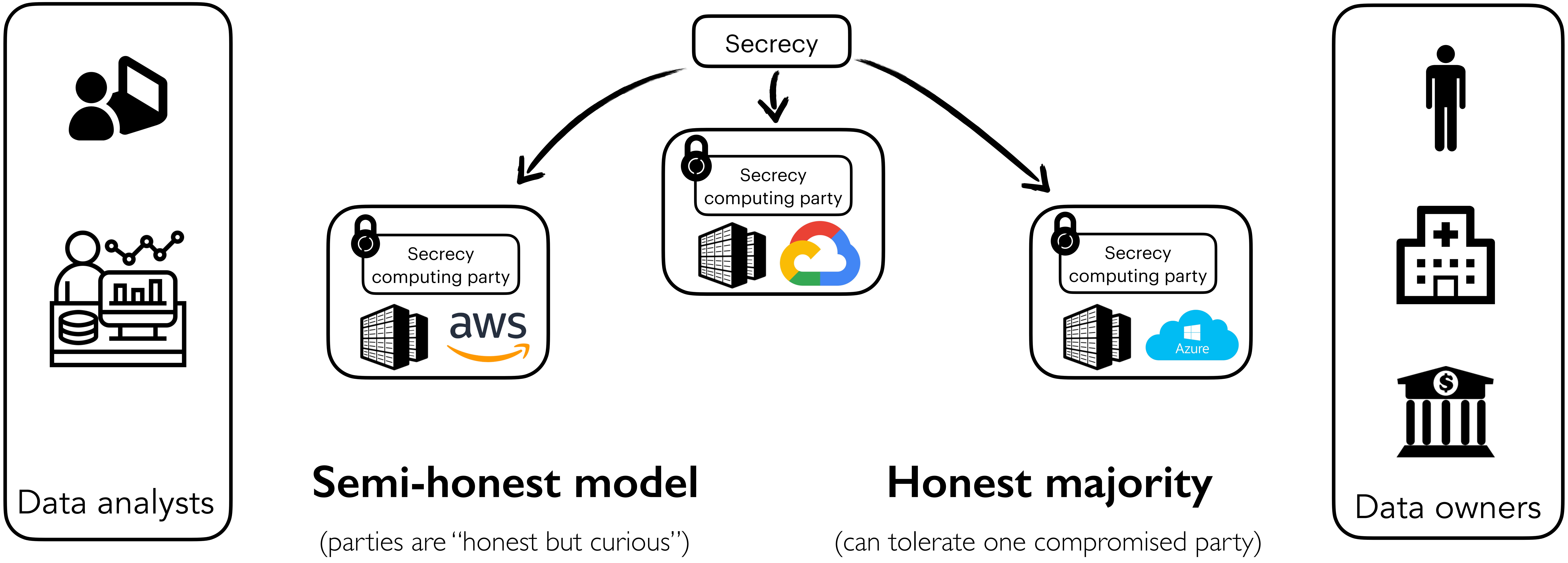
SECRECY AS A SERVICE



SECRECY AS A SERVICE

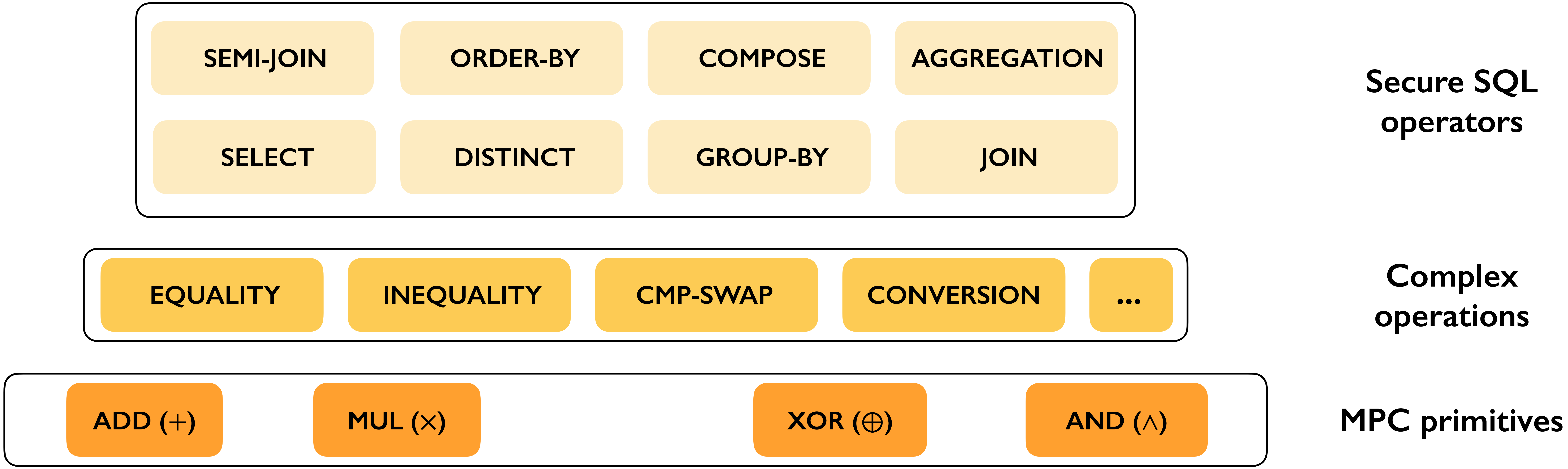


SECRECY AS A SERVICE



T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara. *High-Throughput Semi-Honest Secure Three-Party Computation with an Honest Majority*. CCS, 2016.

FROM SECURE MPC PRIMITIVES TO RELATIONAL ANALYTICS



Arithmetic sharing: $s = s_1 + s_2 + s_3 \pmod{2^k}$ Boolean sharing: $s = s_1 \oplus s_2 \oplus s_3$

SECRECY's CORE CONTRIBUTIONS

1. **Relational MPC primitives**
   - Amortize network I/O
   - Make secret-sharing competitive in WAN

EXAMPLE: MESSAGE BATCHING IN SECRECY

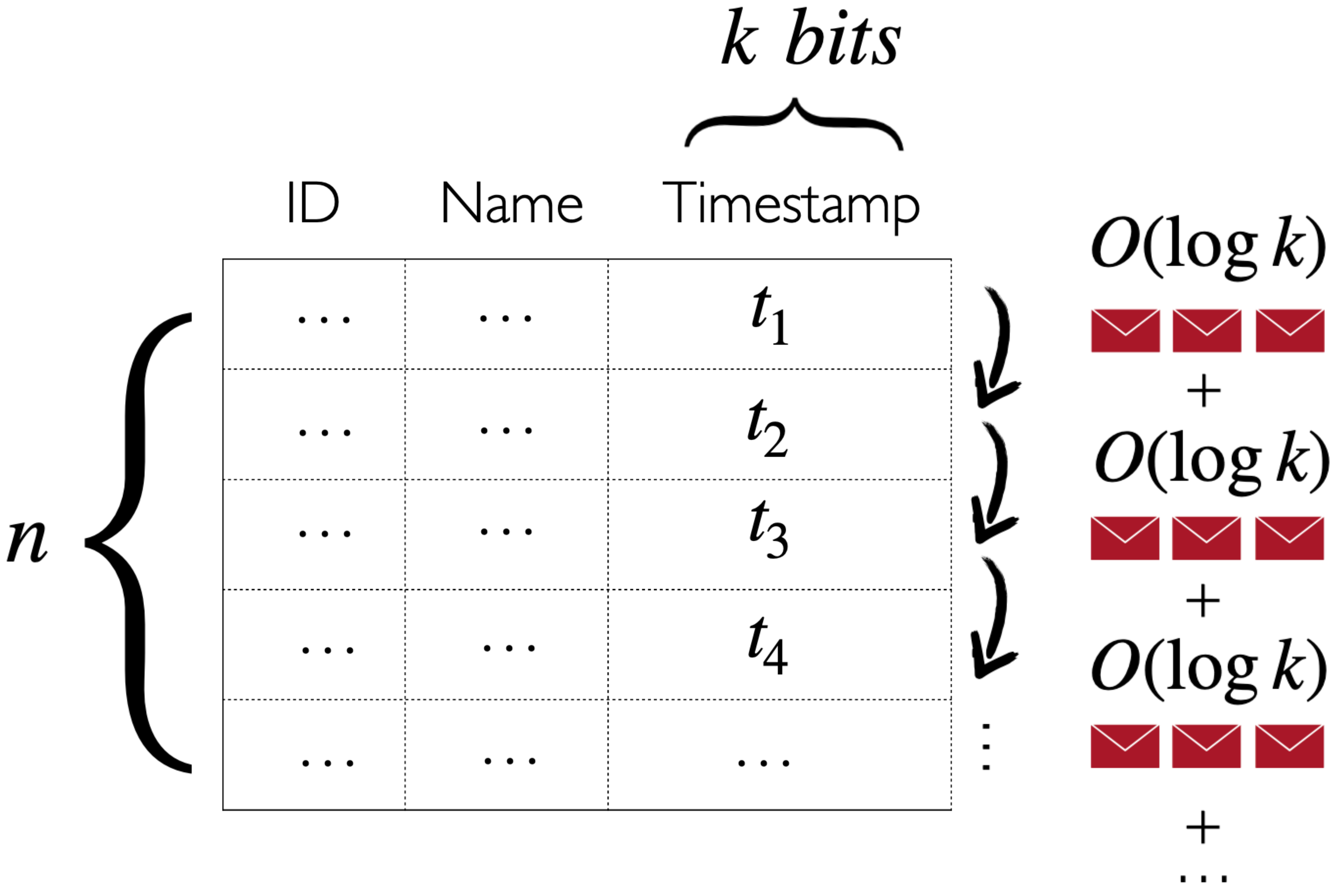
k bits

ID	Name	Timestamp
$\cdots$	$\cdots$	t_1
$\cdots$	$\cdots$	t_2
$\cdots$	$\cdots$	t_3
$\cdots$	$\cdots$	t_4
$\cdots$	$\cdots$	$\cdots$

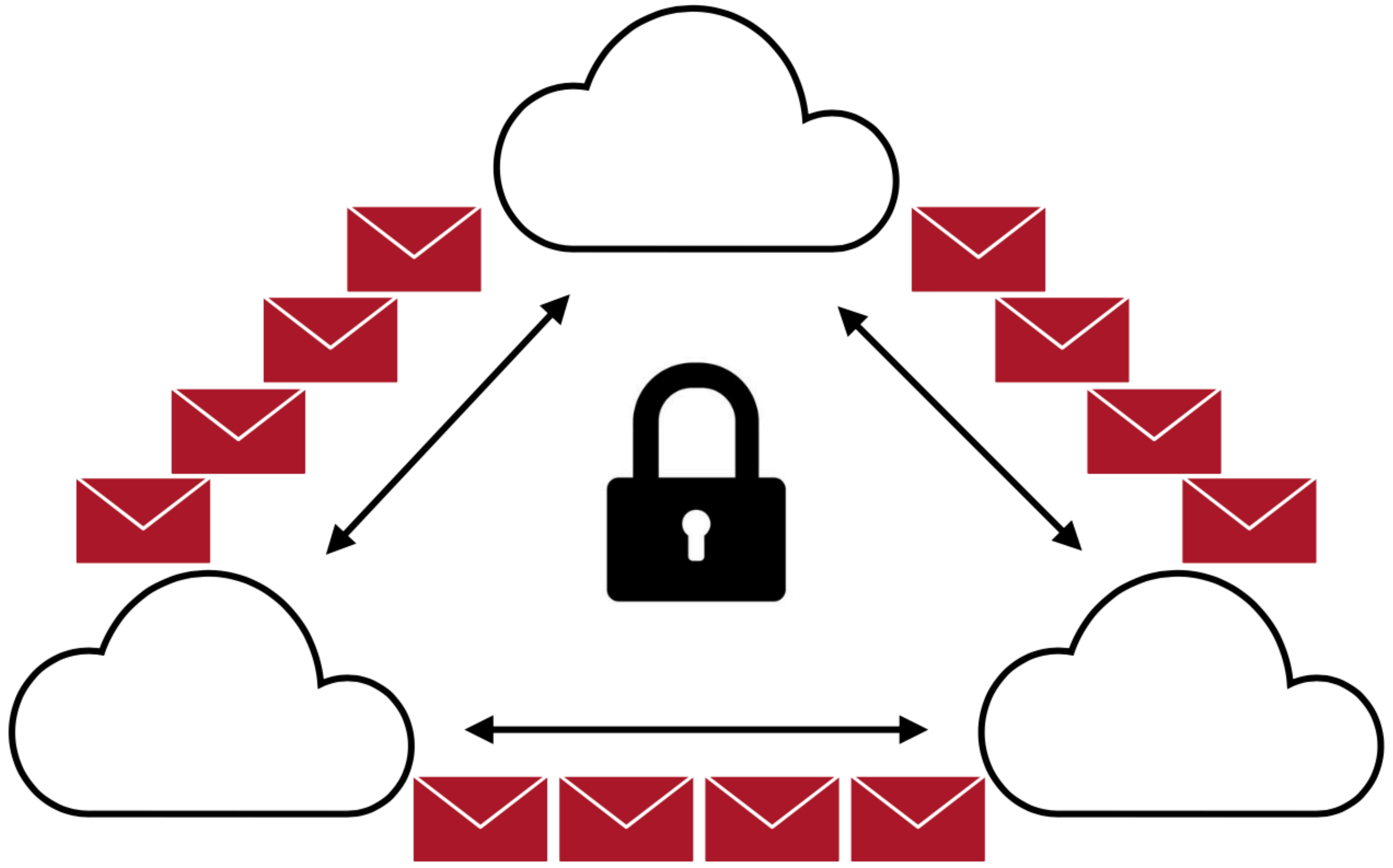
n

"Select all records with timestamp $t > \mathtt{T}$*"*

EXAMPLE: MESSAGE BATCHING IN SECRECY



"*Select all records with timestamp* $t > \mathtt{T}$"



Secret-sharing protocols exchange many small messages between parties

Each inequality requires $O(\log k)$ communication rounds under MPC

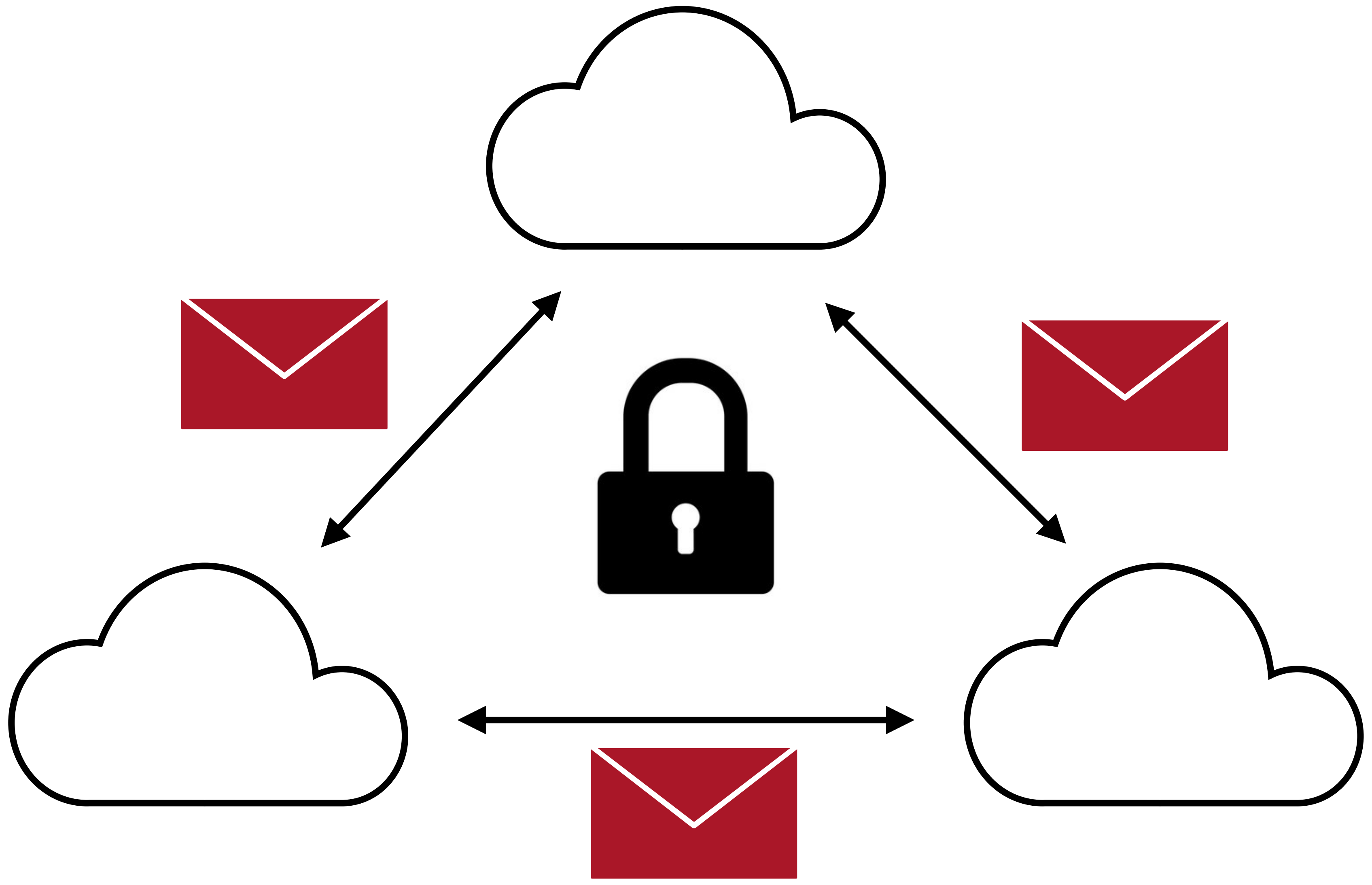
EXAMPLE: MESSAGE BATCHING IN SECRECY

k *bits*

n

ID	Name	Timestamp
$\cdots$	$\cdots$	t_1
$\cdots$	$\cdots$	t_2
$\cdots$	$\cdots$	t_3
$\cdots$	$\cdots$	t_4
$\cdots$	$\cdots$	$\cdots$

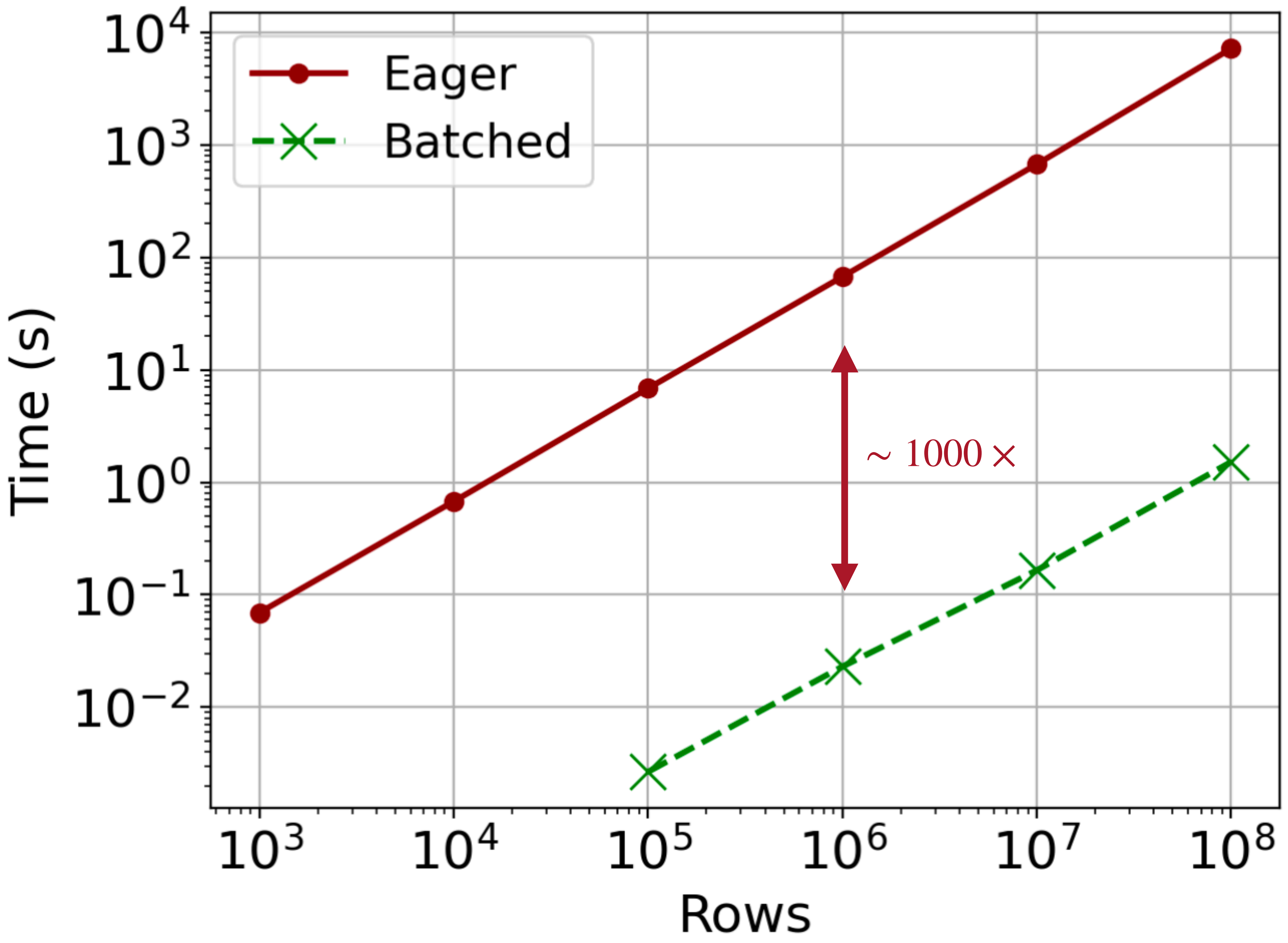
"Select all records with timestamp $t > \mathtt{T}$"



Secrecy requires $\log k + 1$ communication rounds for the entire data table

(independent of the number of records)

EFFECT OF MESSAGE BATCHING (LAN)



- Eager: Message batching disabled (one network I/O per row)
- Batched: Message batching enabled

Lower is better

* Secrecy servers deployed on AWS EC2 r5.xlarge instances (us-east-2)

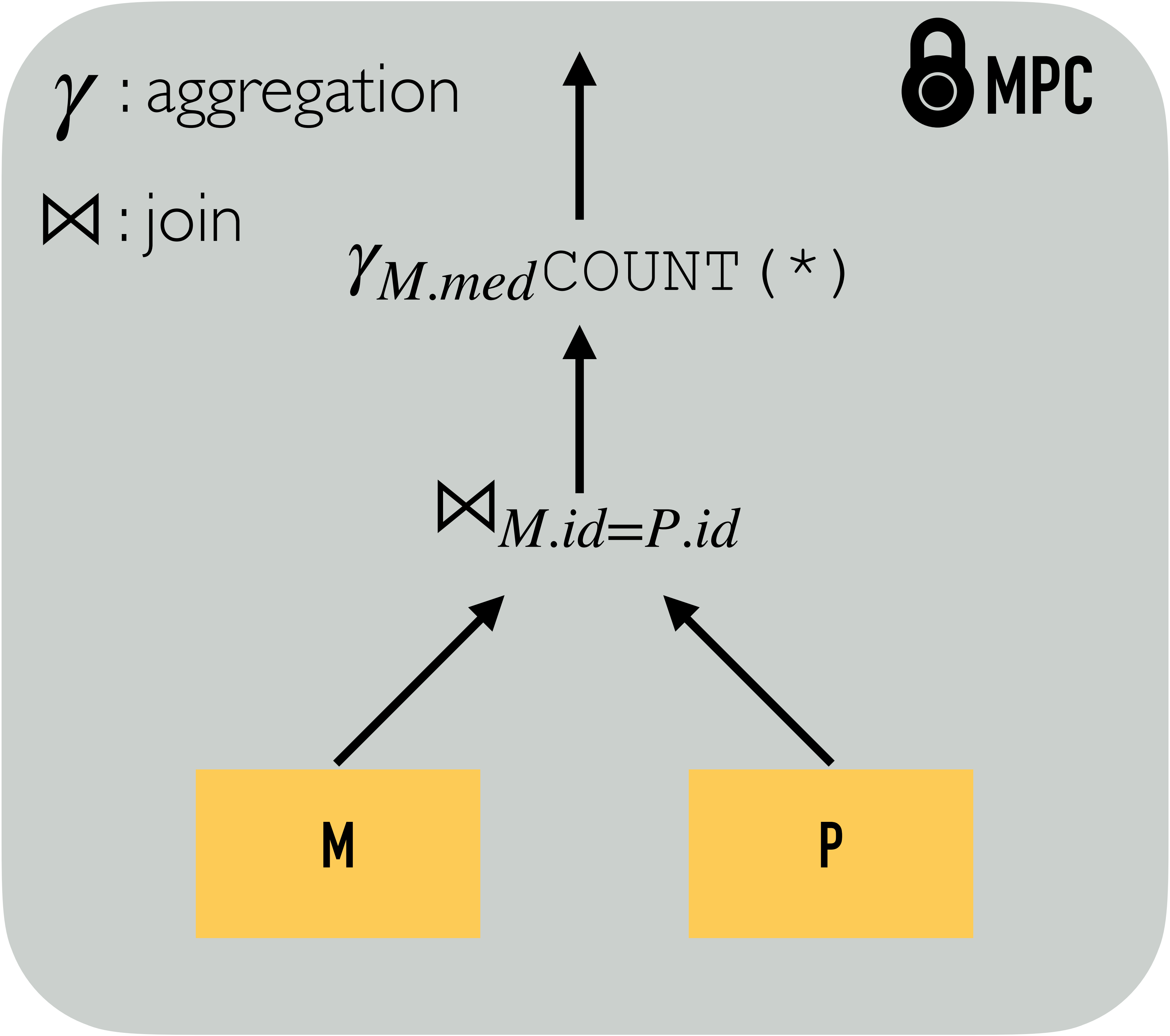
SECRECY's CORE CONTRIBUTIONS

1. **Relational MPC primitives**
   - Amortize network I/O
   - Make secret-sharing competitive in WAN
2. **Analytical cost model for MPC**

SECRECY's CORE CONTRIBUTIONS

1. **Relational MPC primitives**
   - Amortize network I/O
   - Make secret-sharing competitive in WAN
2. **Analytical cost model for MPC**
   - Operation cost
   - Synchronization cost
   - Composition cost

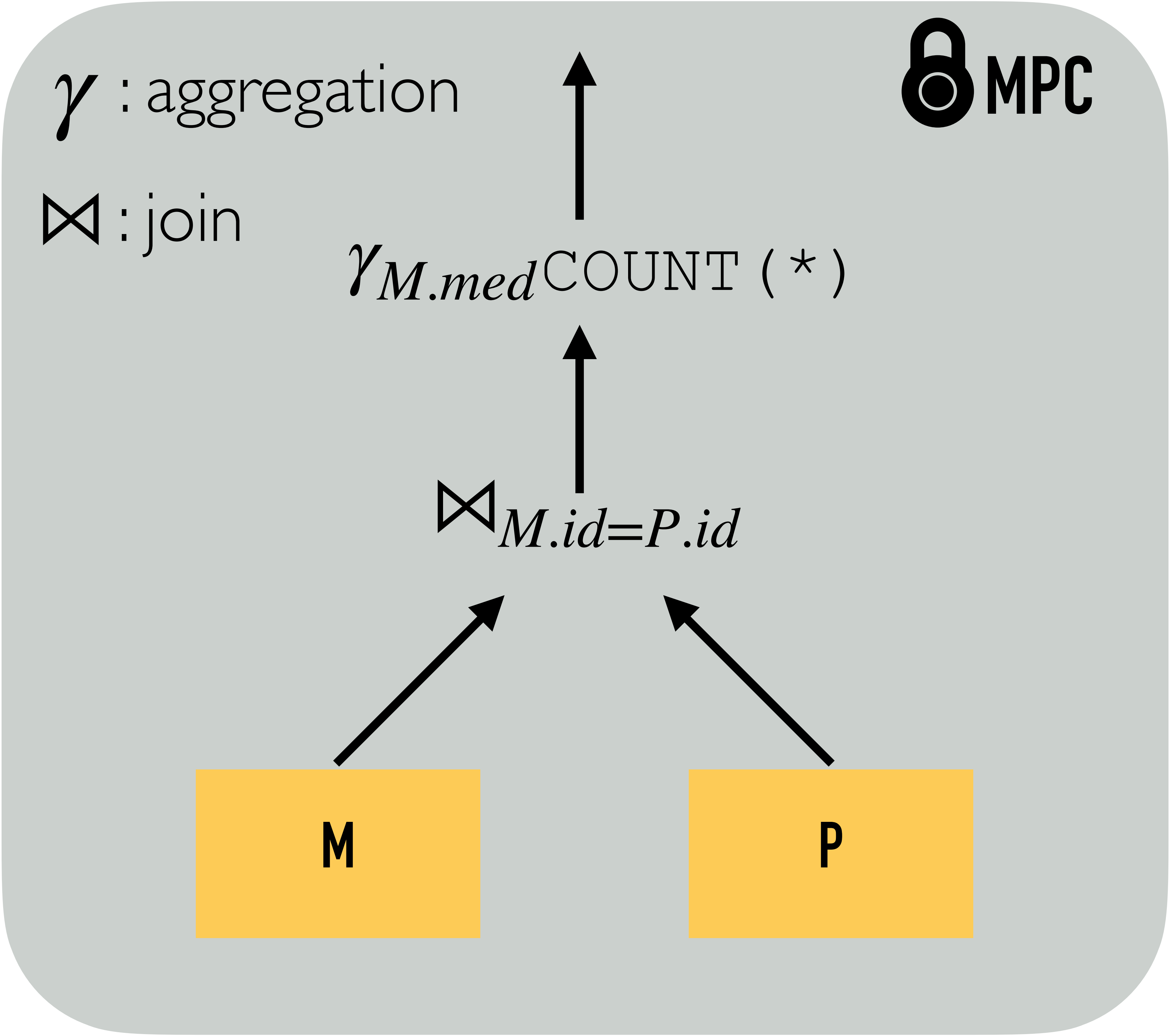
EXAMPLE: OPERATOR DECOMPOSITION IN SECRECY




```
SELECT M.med, COUNT(*)
FROM Medication as M, Patients as P
WHERE M.id = P.id
GROUP-BY M.med
```

"Count the number of patients per prescribed medication"

EXAMPLE: OPERATOR DECOMPOSITION IN SECRECY



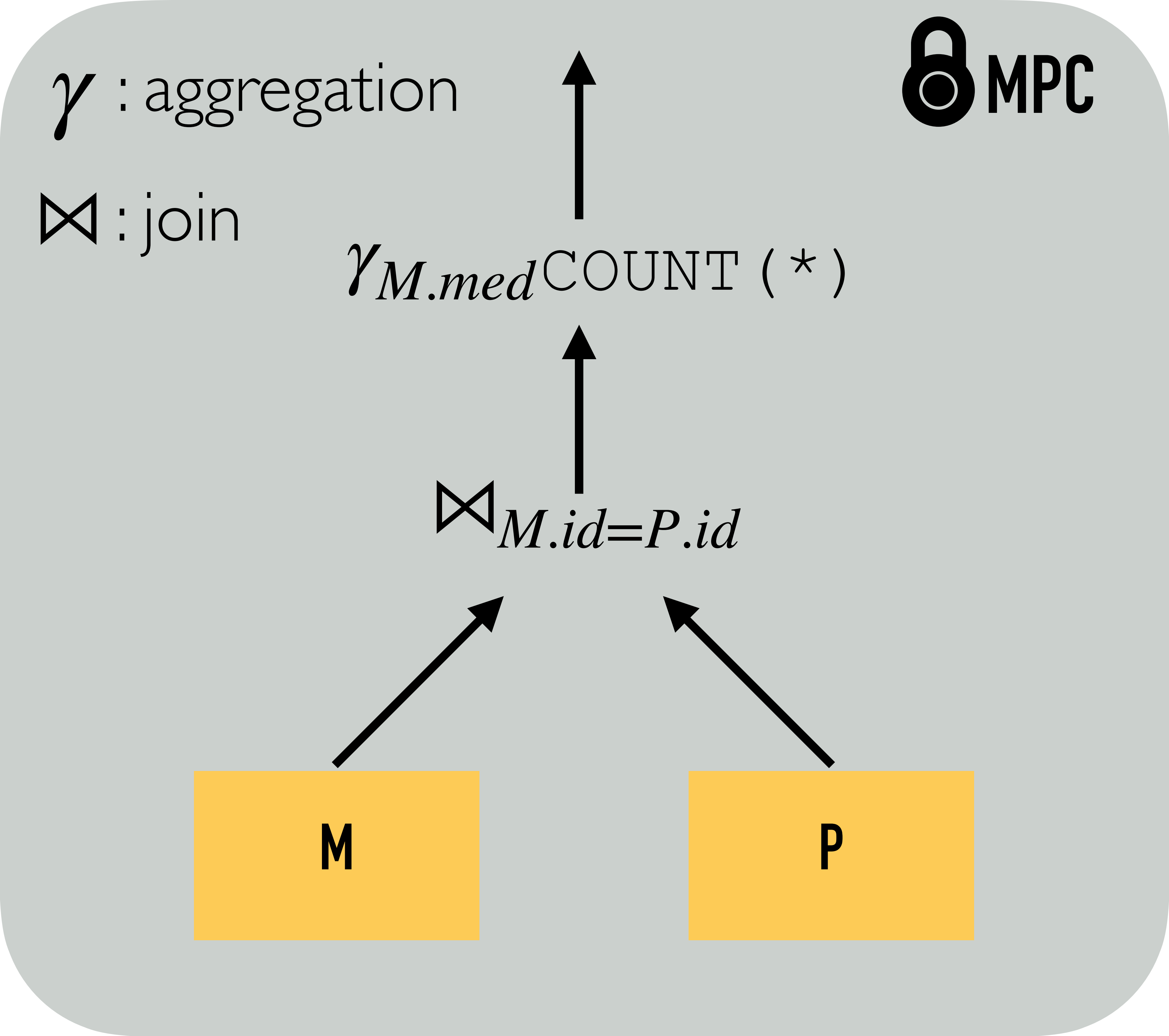

```
SELECT M.med, COUNT(*)
FROM Medication as M, Patients as P
WHERE M.id = P.id
GROUP-BY M.med
```

"Count the number of patients per prescribed medication"

Applying `GROUP-BY` *after the join will require materializing the cartesian product* $M \times P$

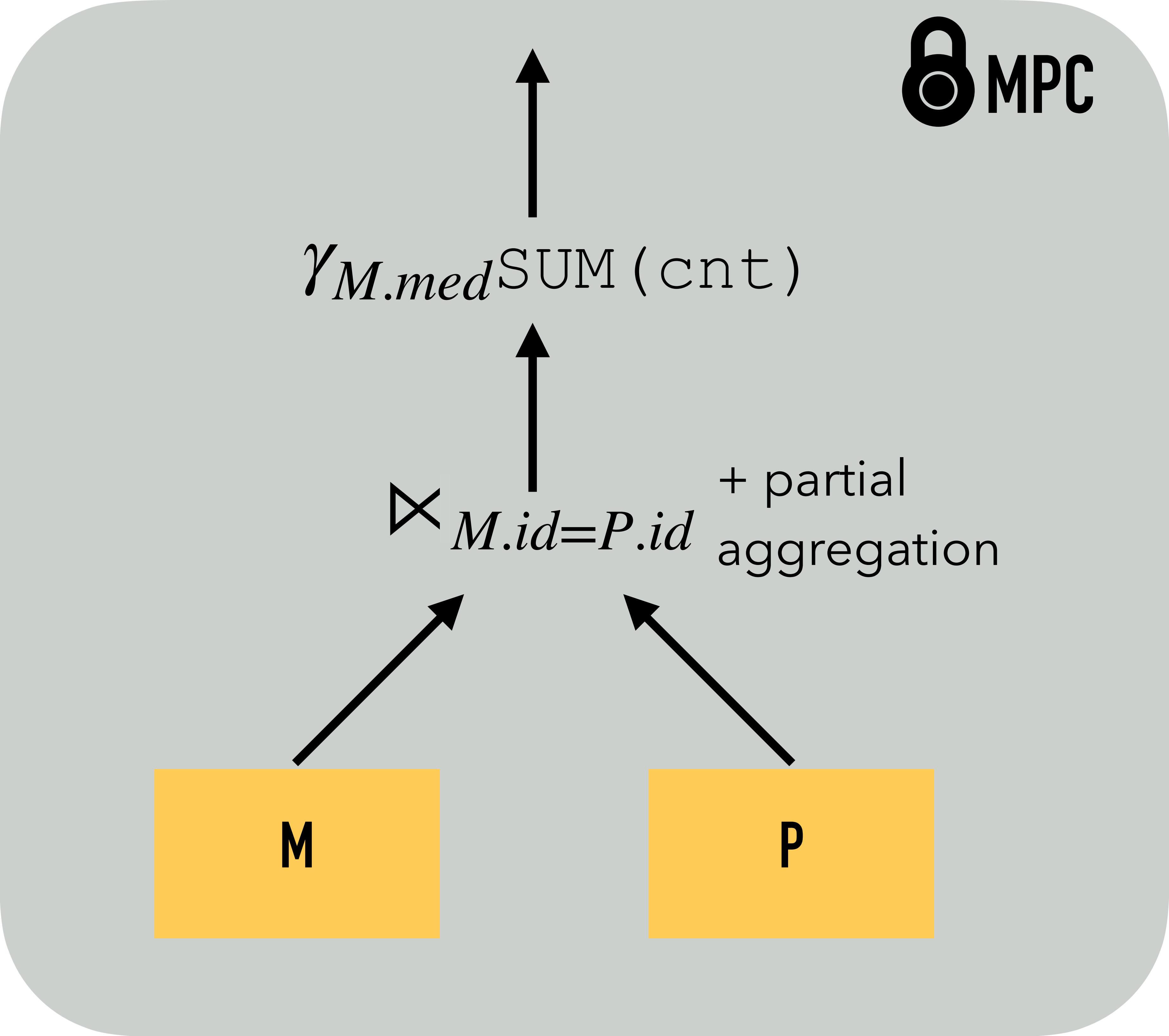
EXAMPLE: OPERATOR DECOMPOSITION IN SECRECY

* Assuming the group-by operator is based on an odd-even circuit



$O(n^2 \log^2 n)$ *operations / messages*

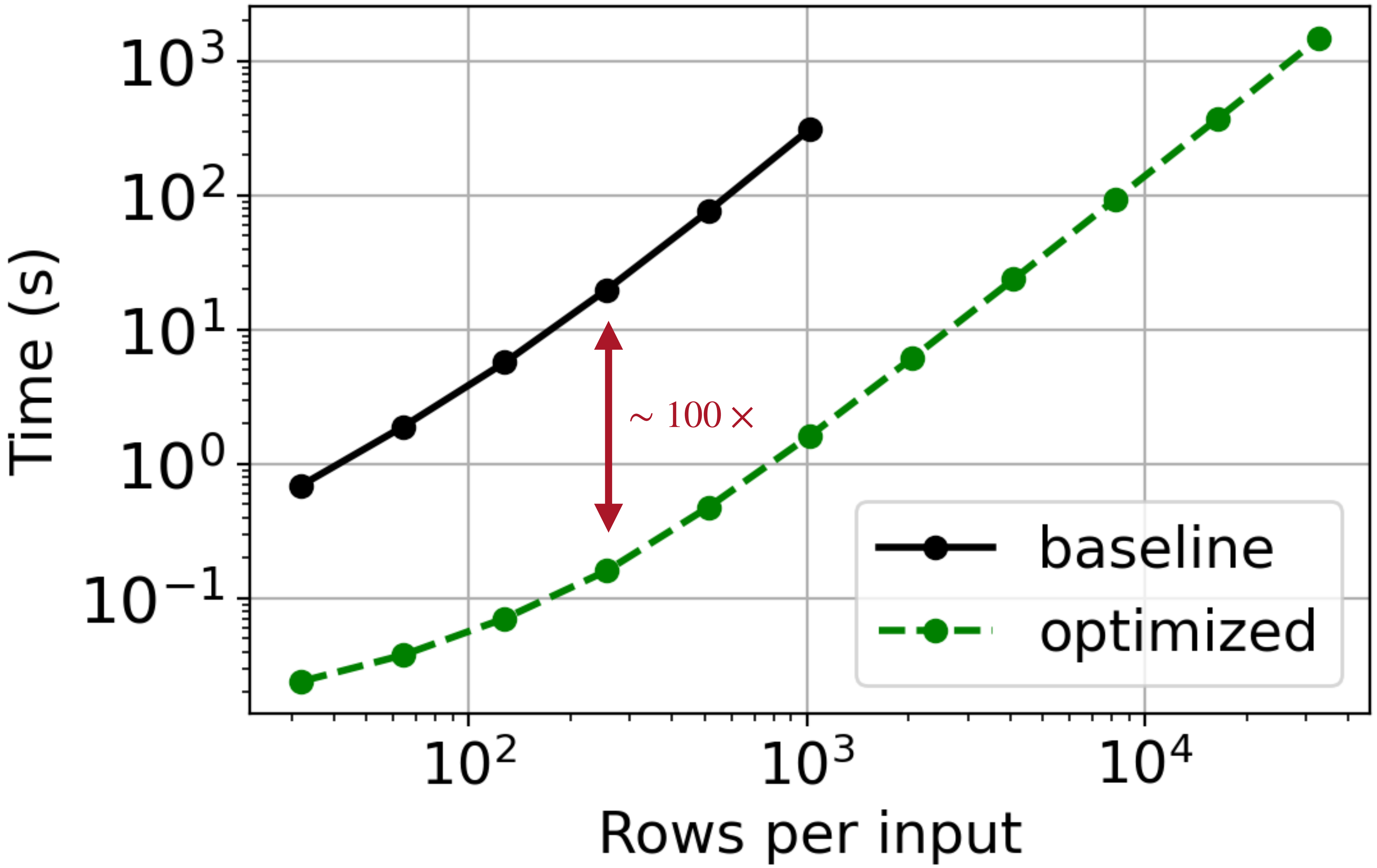
$O(\log^2 n)$ *rounds* $O(n^2)$ *space*



$O(n^2)$ operations / messages

$\sim 4\times$ fewer rounds $O(n)$ *space*

EFFECT OF JOIN-AGGREGATION DECOMPOSITION (LAN)



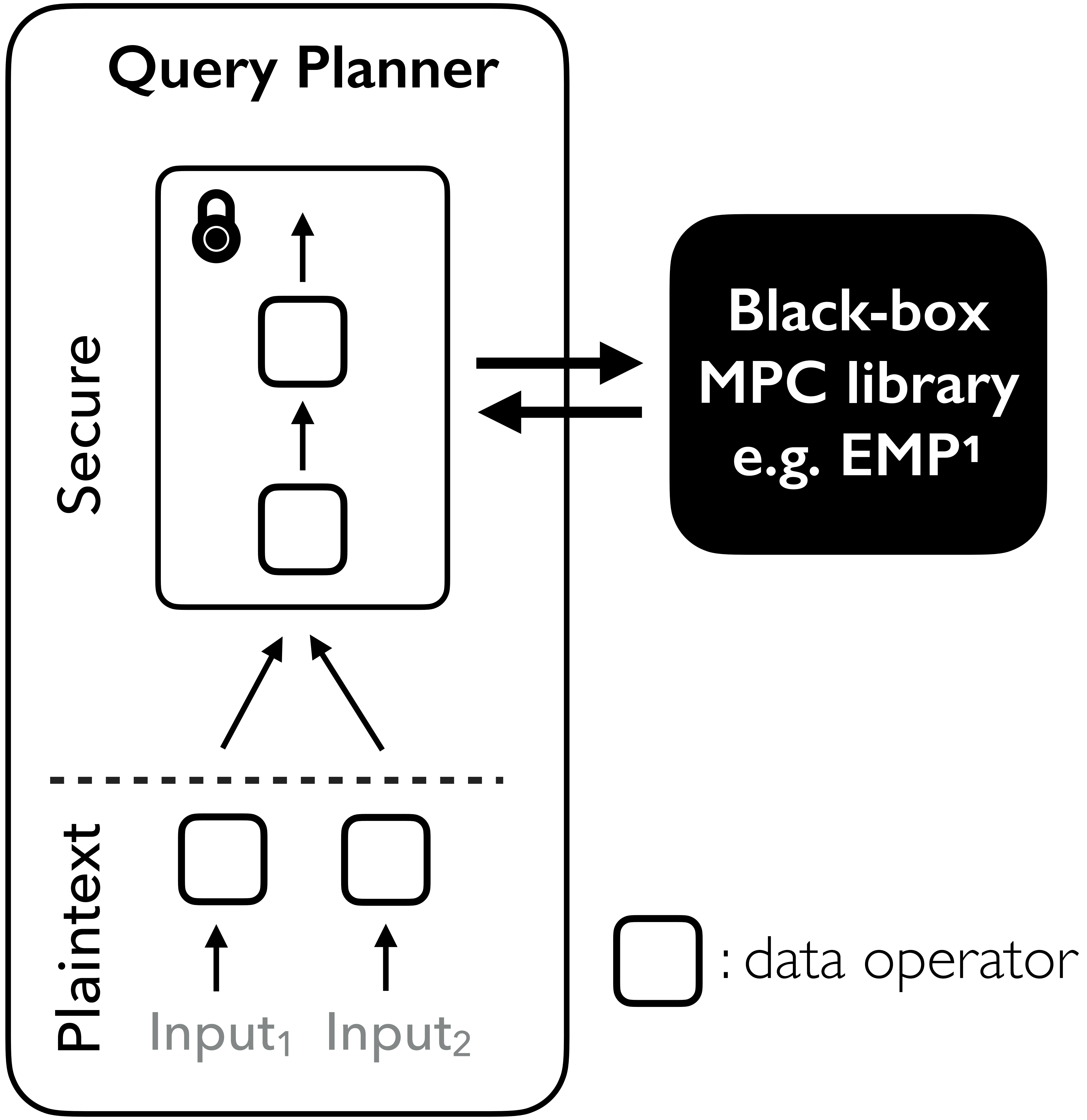
Lower is better

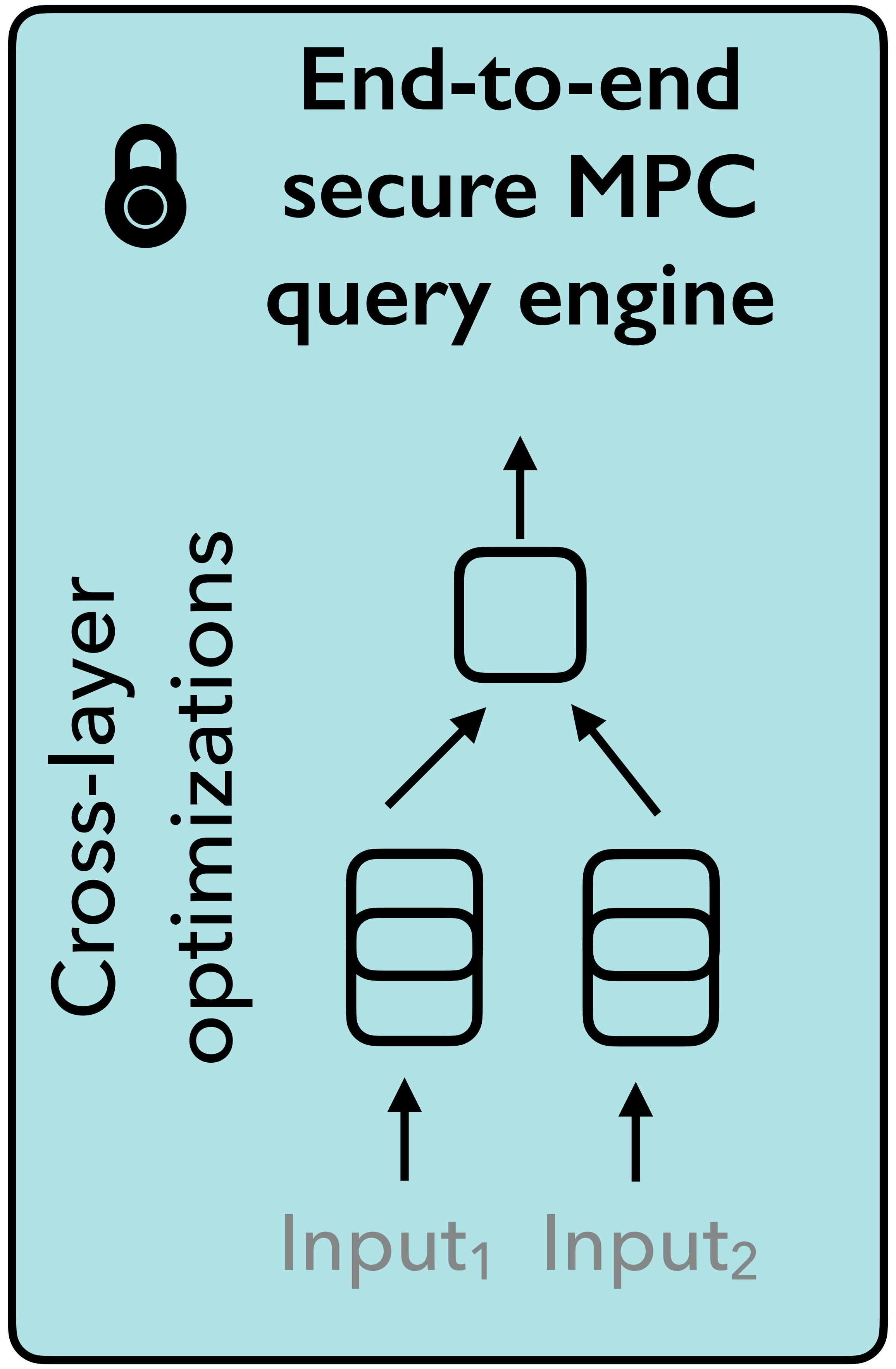
* Secrecy servers deployed on AWS EC2 r5.xlarge instances (us-east-2)

SECRECY's CORE CONTRIBUTIONS

1. **Relational MPC primitives**
   - Amortize network I/O
   - Make secret-sharing competitive in WAN
2. **Analytical cost model for MPC**
   - Operation cost
   - Synchronization cost
   - Composition cost
3. **Vectorized MPC query processor**
   - Logical optimizations
   - System optimizations
   - Protocol-specific optimizations

OPENING THE MPC BLACK BOXES





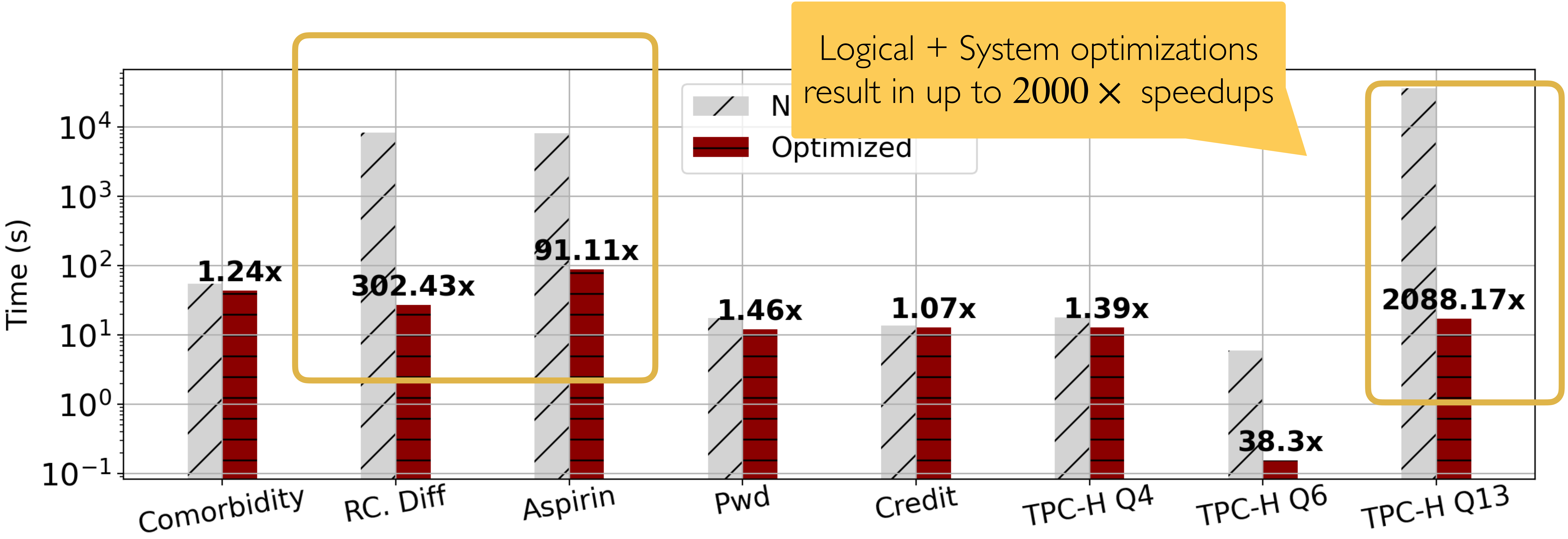
The Secrecy Framework

Supported optimizations:

- Logical (e.g. operator decomposition)
- Physical (e.g. message batching, operator fusion)
- Protocol-specific (e.g. dual sharing)

[1] X. Wang, A. J. Malozemoff, and J. Katz. *EMP-toolkit: Efficient MultiParty computation toolkit*, 2016. https://github.com/emp-toolkit

EFFECT OF SECRECY OPTIMIZATIONS ON REAL QUERIES (MULTI-CLOUD)

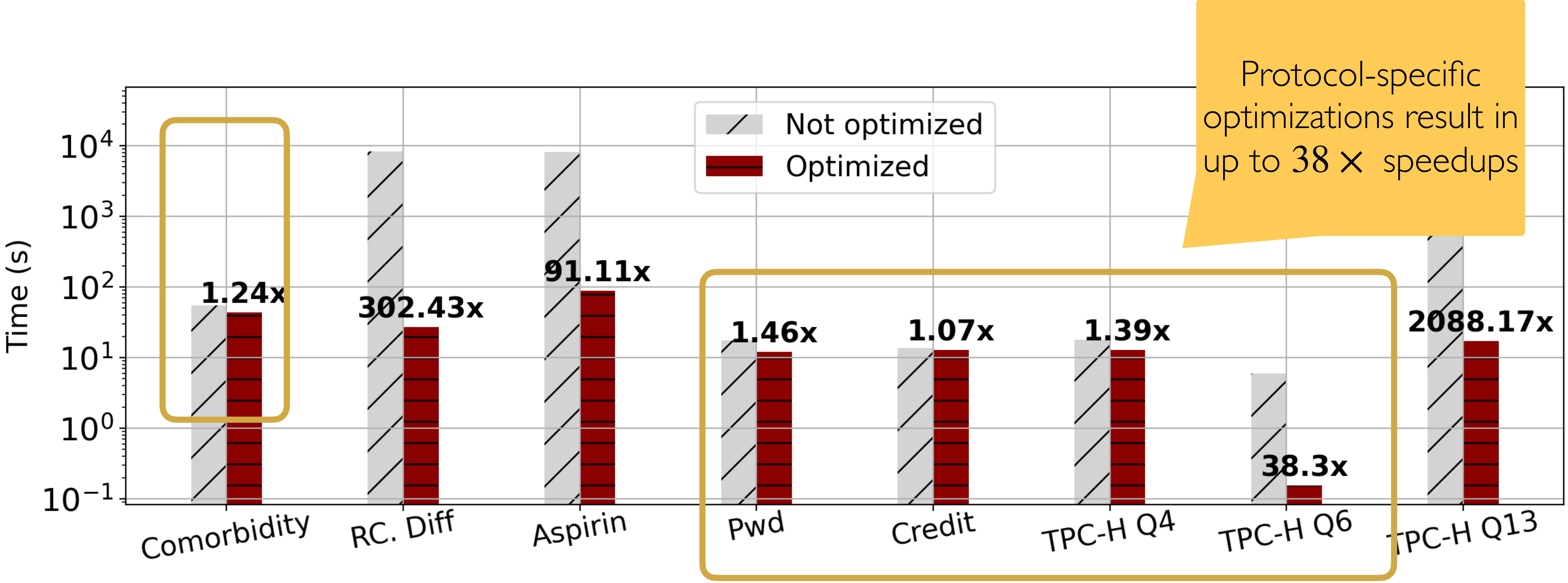


* Secrecy servers deployed in three clouds: AWS (Ohio), GCP (South Carolina), and Azure (Virginia)

* Reported times are for 1000 rows per input table

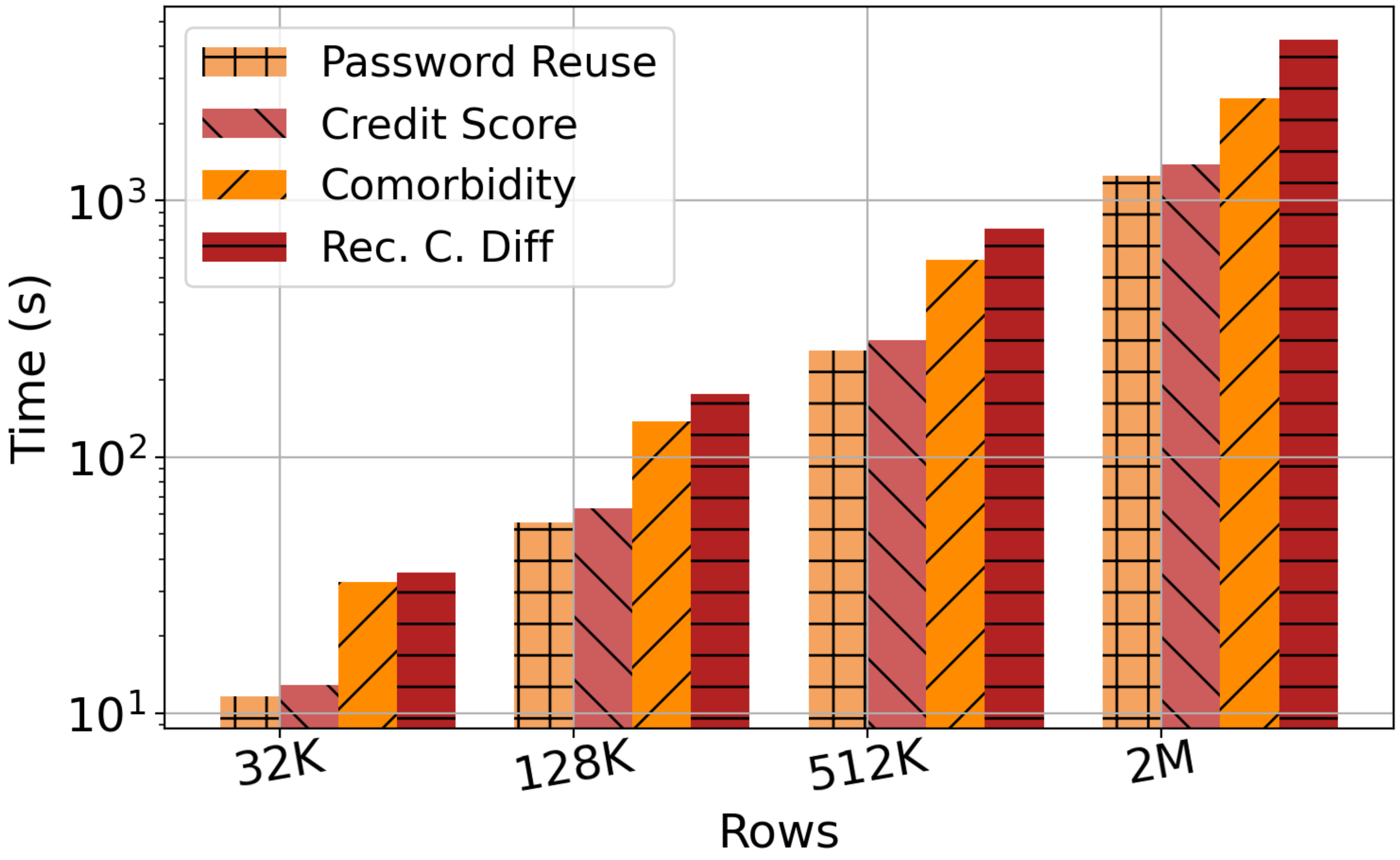
* Not optimized plans use message batching too (otherwise the cost of MPC is prohibitive)

EFFECT OF SECRECY OPTIMIZATIONS ON REAL QUERIES (MULTI-CLOUD)



* Secrecy servers deployed in three clouds: AWS (Ohio), GCP (South Carolina), and Azure (Virginia)
* Reported times are for 1000 rows per input table
* Not optimized plans use message batching too (otherwise the cost of MPC is prohibitive)

SECRECY's SCALING BEHAVIOR (LAN)



Rec. C. Diff scales to 2 million rows in ~1.2h

```
WITH rcd AS (
   SELECT pid, time, row_no() OVER
   (PARTITION BY pid ORDER BY time)
   FROM diagnosis
   WHERE diag=cdiff)
 SELECT DISTINCT pid
 FROM rcd r1 JOIN rcd r2 ON r1.pid = r2.pid
 WHERE r2.time - r1.time >= 15 DAYS
 AND r2.time - r1.time <= 56 DAYS
 AND r2.row_no = r1.row_no + 1
```

"Find the distinct ids of patients who have been diagnosed with `cdiff` *and have two consecutive infections between 15 and 56 days apart"*

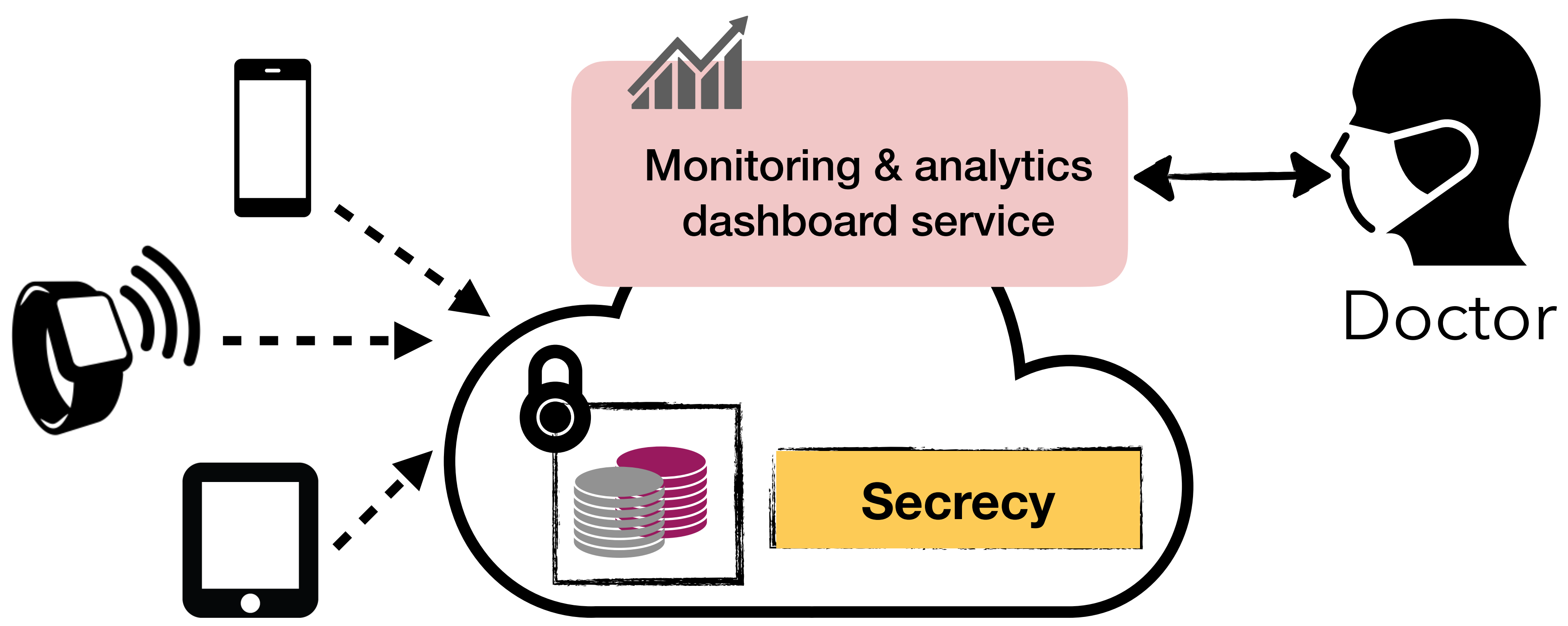
* Secrecy servers deployed on AWS EC2 r5.xlarge instances (us-east-2)

* Each server uses a single vCPU

REAL-WORLD SECRECY USE CASES

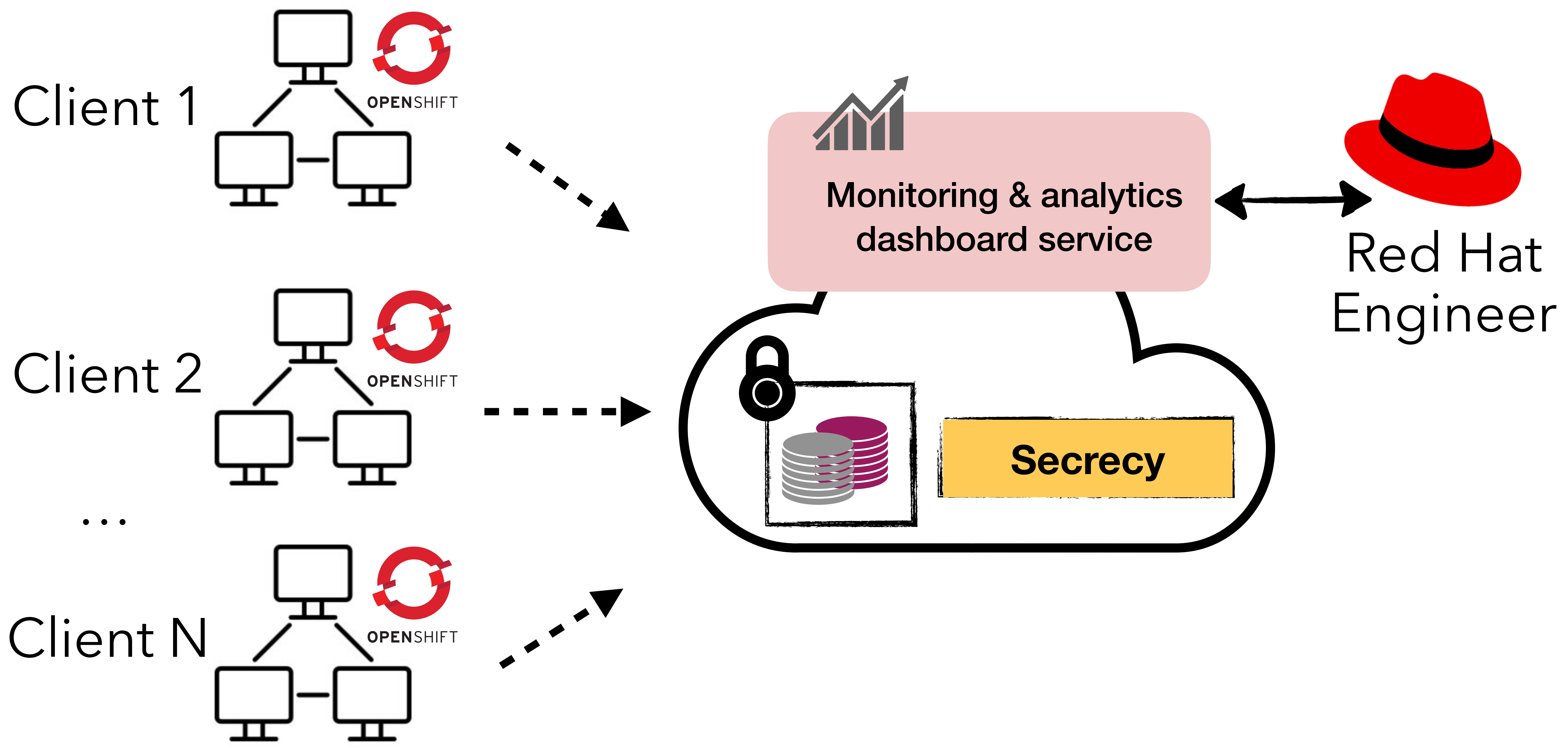
Secure digital health analytics[1]

(BU Medical & Hariri Institute for Computing)



Secure cross-site analytics on OpenShift logs[2]

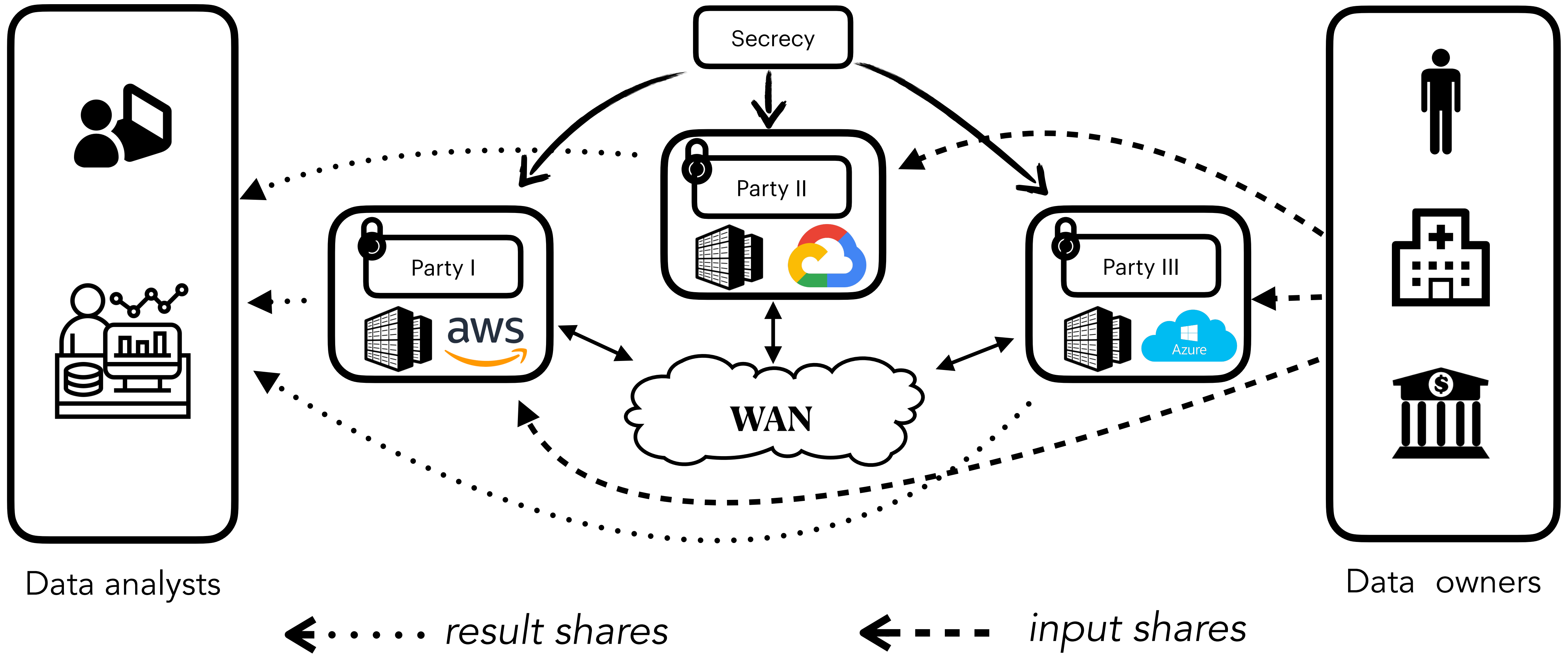
(BU Red Hat Collaboratory)

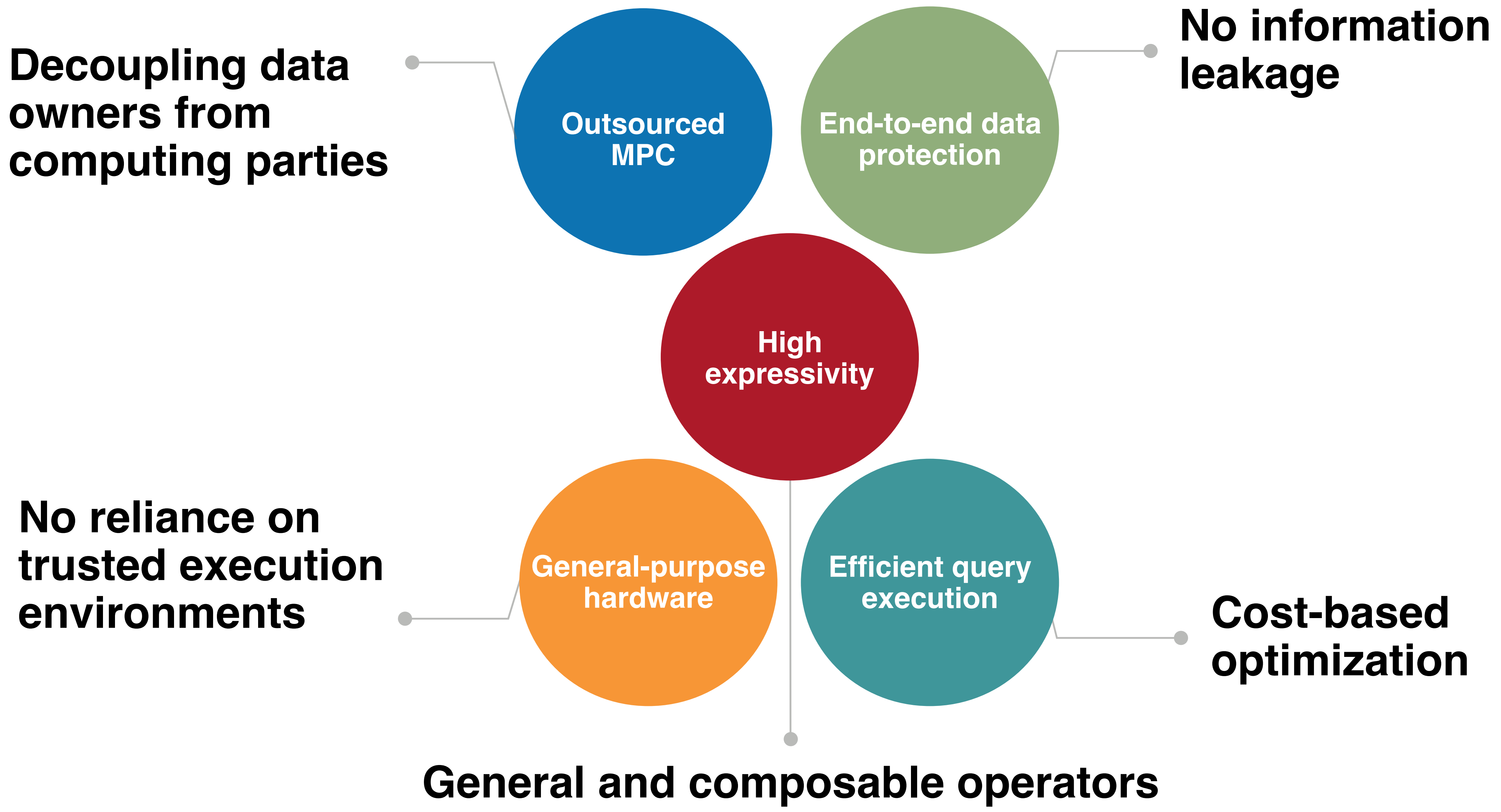


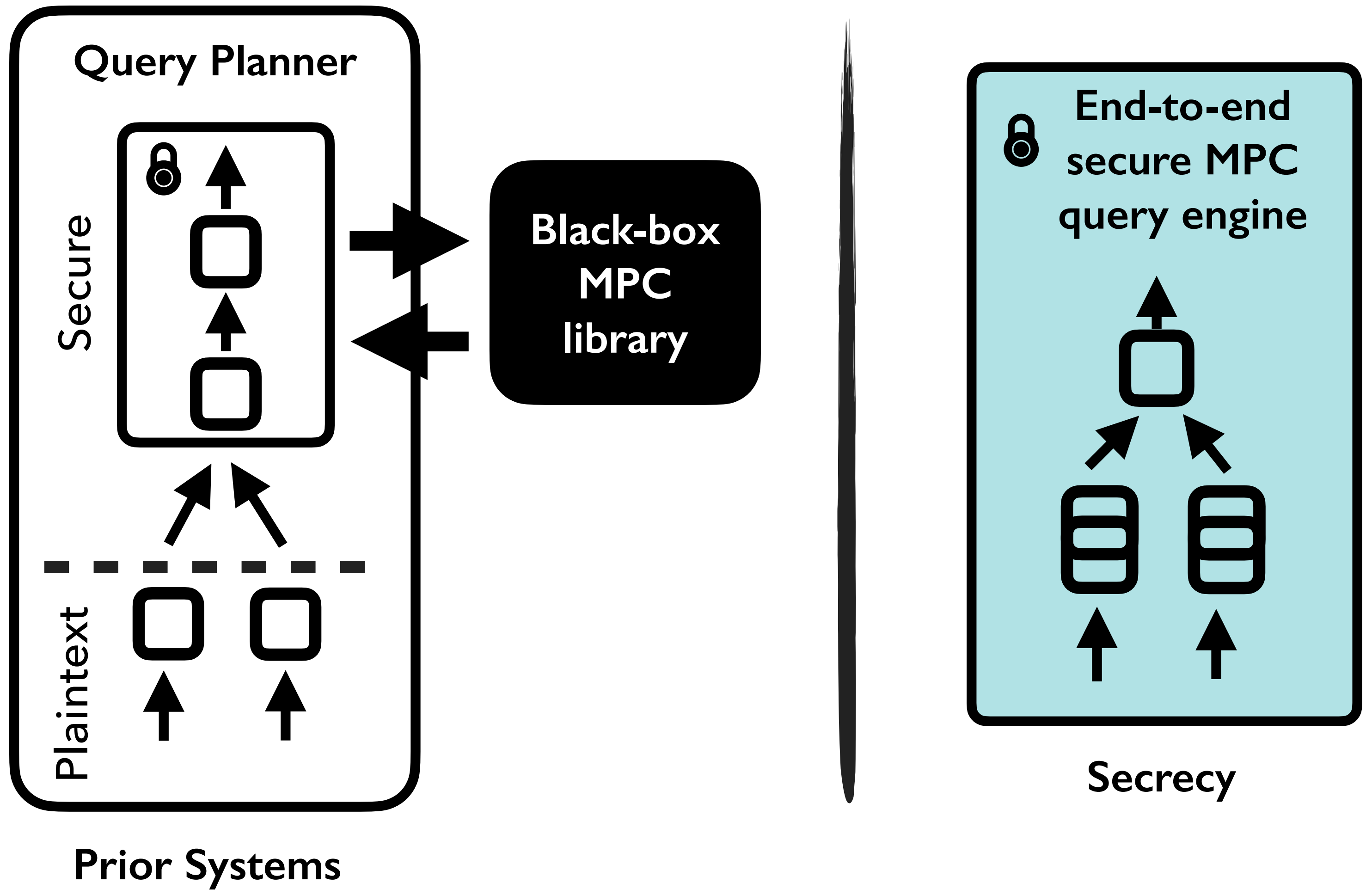
1 https://www.bu.edu/hic/research/focused-research-programs/continuous-analysis-of-mobile-health-data-among-medically-vulnerable-populations/

2 https://www.bu.edu/rhcollab/projects/security-privacy/secure-cross-site-analytics-on-openshift-logs/

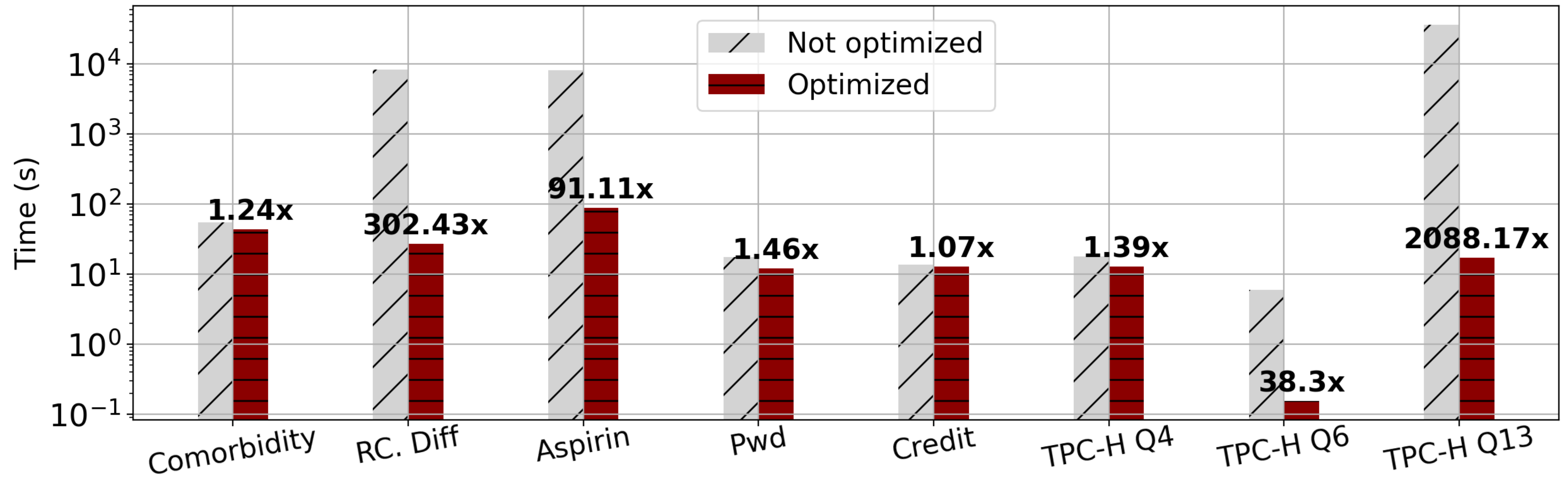
SECRECY SUMMARY



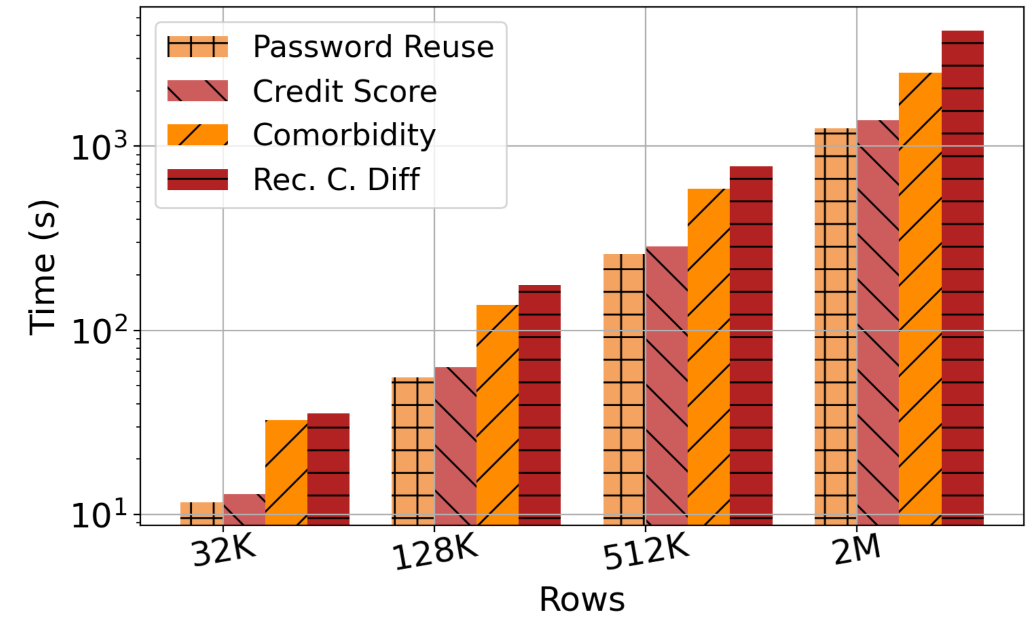




Up to 2000x speedups for real and synthetic workloads



Millions of input records entirely under MPC



Source code: https://github.com/CASP-Systems-BU/Secrecy