Making Privacy Technology Accessible: Benchmarks and Platforms

Michael Hay, Colgate University

The opinions expressed in this talk are my own and not those of the U.S. Census Bureau.

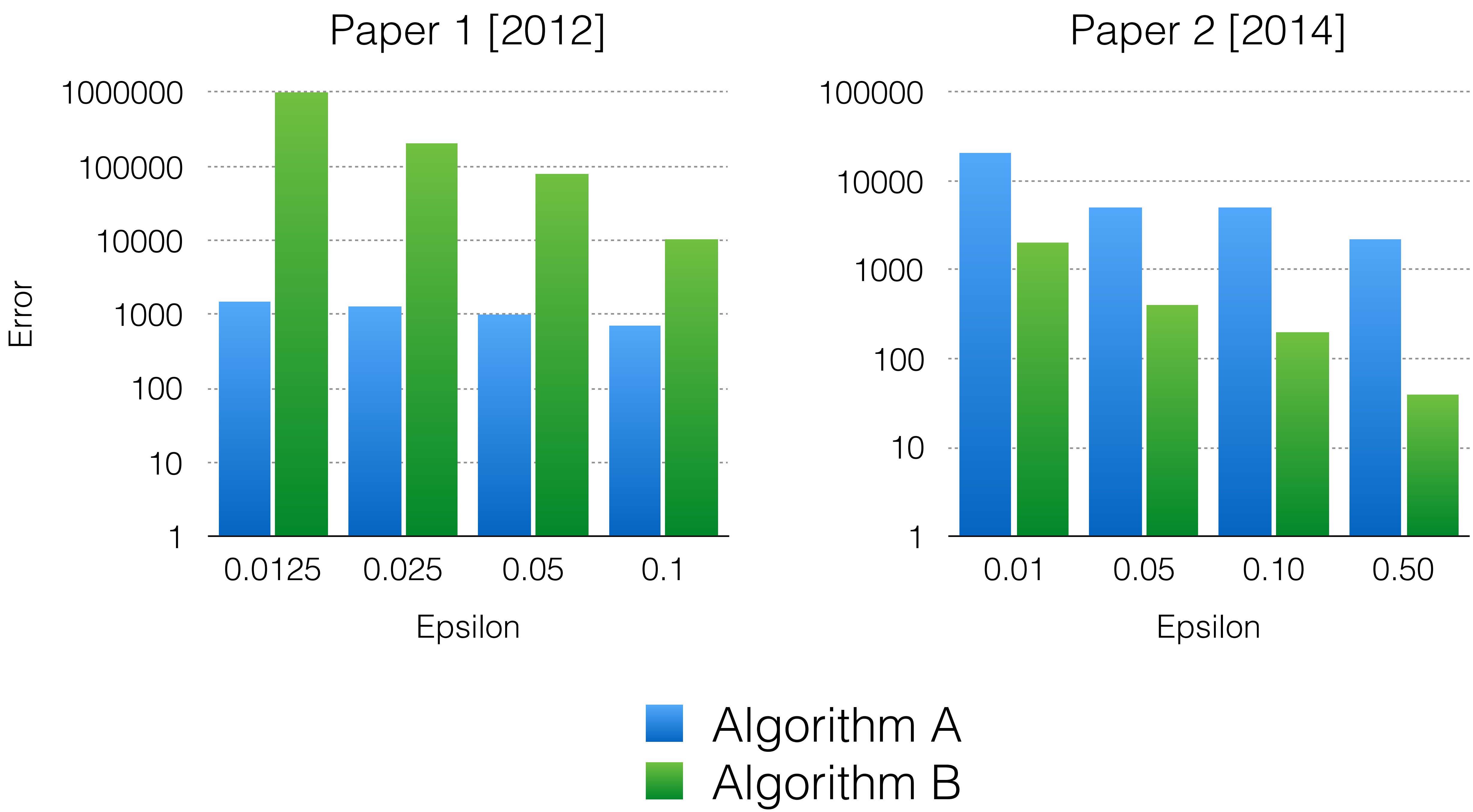
Simons Workshop on “Data Privacy: From Foundations to Applications” March 2019

Illustrative Example

age	child race	household size	race householder
12	white	3	white
9	asian	4	white
…	…	…	…

Goal: Produce 2-way marginal between race of child and race of householder, computed under DP

Which DP algorithm should I use?



Analysis & Implementation

- Query: 2-way marginal between race of child and race of householder

- Analyst calculates sensitivity

age	race	household size	race householde
12	white	3	white
29	asian	4	white
…	…	…	…

- Analysts finds Laplace RNG

    - Friend (DP expert) warns, "Watch out for floating-point precision attack." [Mironov CCS12]

Challenges to deployment

- Conflicting empirical results
- Lack of reference implementations
- Risk of subtle bugs (analysis + implementation)

Today’s talk

- Introduction
- DPBench: principled empirical evaluations of accuracy
- Ektelo: framework for private computation
- PrivateSQL: differentially private SQL query engine

Sound evaluation is hard

- Factors affecting performance: setting of epsilon, “amount” of data, tunable algorithm parameters, data pre-processing (cleaning, representation)

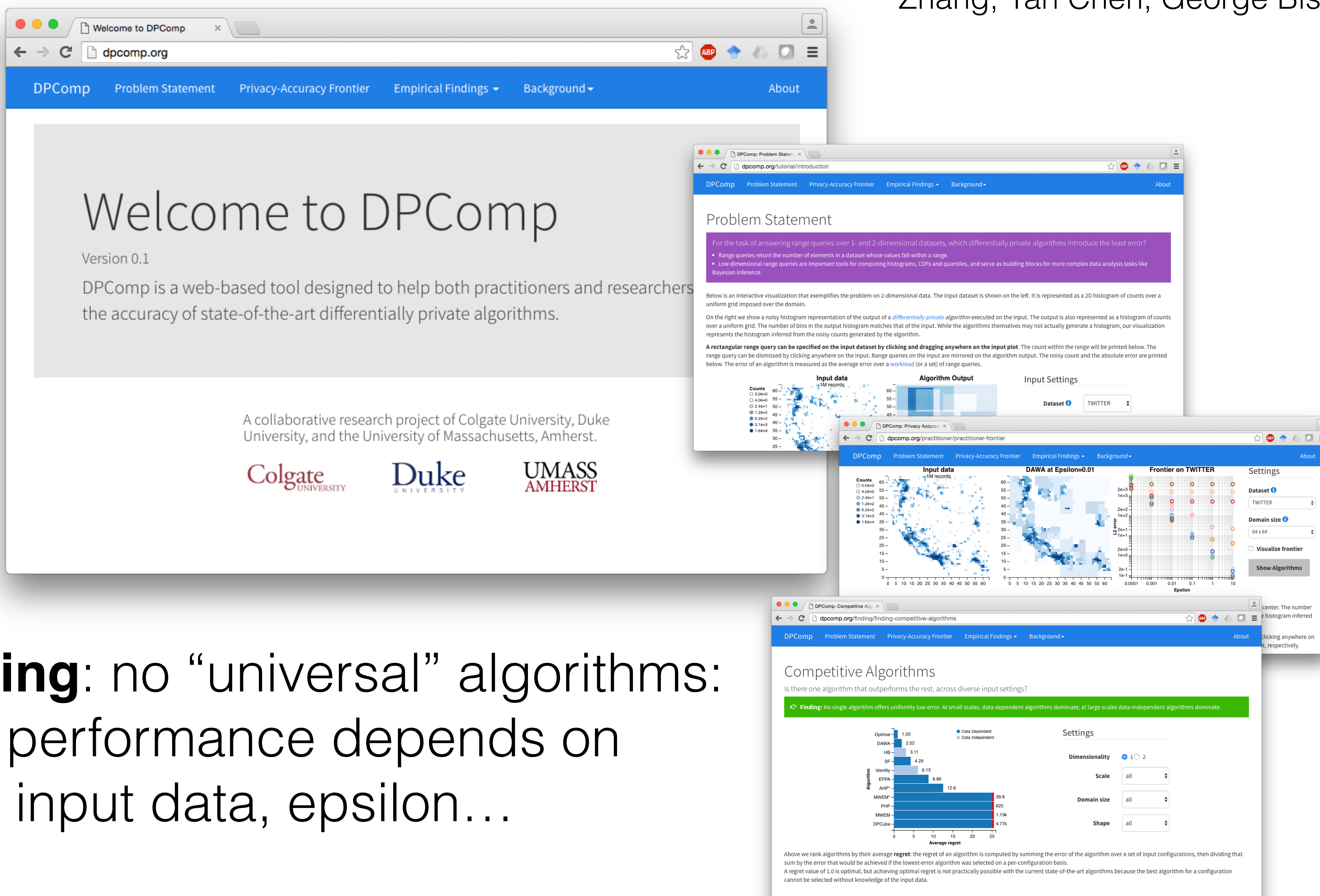
- Algorithms can be **data-dependent** because they *adapt* or *introduce statistical bias*.

    - Examples: smooth sensitivity [Nissim STOC 2007], DAWA [Li VLDB 2014], Adaptive Grid [Qardaji ICDE 2013], StructureFirst [Xu VLDBJ 2013]

Principled evaluation of DP algorithms [SIGMOD16]

Companion website**:** dpcomp.org

Joint work with Gerome Miklau, Ashwin Machanavajhalla, Dan Zhang, Yan Chen, George Bissias



Finding: no "universal" algorithms: best performance depends on task, input data, epsilon…

Sound evaluation is important!

- How to incentivize community participation?
  - Benchmarks
    Successful in other communities TPC-H, Trec, MNIST
  - Contests
    NIST Differential Privacy Synthetic Data Challenge
  - Reproducibility requirements

Outline

- Introduction
- DPBench: principled empirical evaluations of accuracy
- Ektelo: framework for private computation
- PrivateSQL: differentially private SQL query engine

Challenges of DP Deployment

- Successful deployments have required a team of privacy experts.
- Limited resources available
  - Few libraries, reference implementations or re-usable tools.
  - Frameworks like PINQ ensure privacy safe computation, but little guidance on accuracy
  - Implementations often start from scratch in arbitrary PL.
- Difficult for privacy non-experts to contribute.

Challenges of DP Deployment

- Privacy: Many points of failure
  ➡ Code must be carefully vetted.

- Accuracy: Sophisticated algorithms needed
  ➡ Need to think in new ways to get optimal error

- Context: data analysis workflows are *ad hoc*
  ➡ Need toolkits, not monolithic algorithms.

Joint work with Gerome Miklau, Ashwin Machanavajhalla,
Dan Zhang, Ryan McKenna, Ios Kotsogiannis

εktelo execution framework

- Goal: simplify and accelerate development of *efficient* and *accurate* differentially private algorithms
- Ektelo supports a library of vetted **operators**.
- Operators encode (some) best practices from literature
- Differentially private computation expressed as a ***plan:*** a sequence of operator calls

```
persons = ProtectedDataSource(persons_uri)
for level in geo_levels:
  geo_regions = SelectPartition(level)
  splits = persons.SplitByPartition(geo_regions)
  for persons_in_region in splits:
    x = persons_in_region.Vectorize()
    M = SelectMeasurementsHDMM(W)
    y = x.LaplaceMeasure(M, eps)
    x_hat = LeastSquares(M, y)
    ... additional post-processing ...
```

Top Down algorithm*
implemented as Ektelo plan.

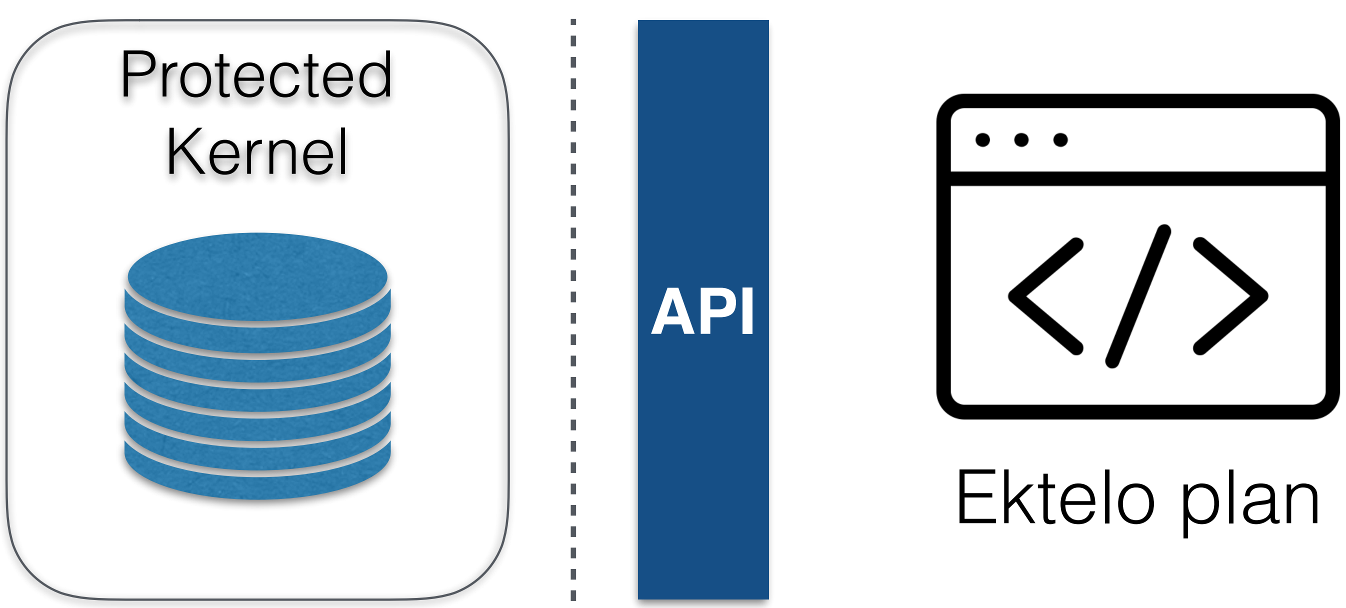
(Artistic rendering)

* Dan Kifer's presentation "Consistency with External Knowledge: The TopDown Algorithm"

```
persons = ProtectedDataSource(persons_uri)
for level in geo_levels:
  geo_regions = SelectPartition(level)
  splits = persons.SplitByPartition(geo_regions)
  for persons_in_region in splits:
    x = persons_in_region.Vectorize()
    M = SelectMeasurementsHDMM(W)
    y = x.LaplaceMeasure(M, eps)
    x_hat = LeastSquares(M, y)
    ... additional post-processing ...
```

Top Down algorithm*
implemented as Ektelo plan.

(Artistic rendering)



Plan executed by *client*, with calls to *protected kernel* that manages sensitive data

```
persons = ProtectedDataSource(persons_uri)
for level in geo_levels:
  geo_regions = SelectPartition(level)
  splits = persons.SplitByPartition(geo_regions)
  for persons_in_region in splits:
    x = persons_in_region.Vectorize()
    M = SelectMeasurementsHDMM(W)
    y = x.LaplaceMeasure(M, eps)
    x_hat = LeastSquares(M, y)
    ... additional post-processing ...
```

Top Down algorithm* implemented as Ektelo plan.

(Artistic rendering)

Transformations

```
persons = ProtectedDataSource(persons_uri)
for level in geo_levels:
  geo_regions = SelectPartition(level)
  splits = persons.SplitByPartition(geo_regions)
  for persons_in_region in splits:
    x = persons_in_region.Vectorize()
    M = SelectMeasurementsHDMM(W)
    y = x.LaplaceMeasure(M, eps)
    x_hat = LeastSquares(M, y)
    ... additional post-processing ...
```

Top Down algorithm* implemented as Ektelo plan.

(Artistic rendering)

Measurement Selection

```
persons = ProtectedDataSource(persons_uri)
for level in geo_levels:
  geo_regions = SelectPartition(level)
  splits = persons.SplitByPartition(geo_regions)
  for persons_in_region in splits:
    x = persons_in_region.Vectorize()
    M = SelectMeasurementsHDMM(W)
    y = x.LaplaceMeasure(M, eps)
    x_hat = LeastSquares(M, y)
    ... additional post-processing ...
```

Top Down algorithm* implemented as Ektelo plan.

(Artistic rendering)

Measurement

```
persons = ProtectedDataSource(persons_uri)
for level in geo_levels:
  geo_regions = SelectPartition(level)
  splits = persons.SplitByPartition(geo_regions)
  for persons_in_region in splits:
    x = persons_in_region.Vectorize()
    M = SelectMeasurementsHDMM(W)
    y = x.LaplaceMeasure(M, eps)
    x_hat = LeastSquares(M, y)
    ... additional post-processing ...
```

Top Down algorithm* implemented as Ektelo plan.

(Artistic rendering)

Inference (and other post-processing)

```
persons = ProtectedDataSource(persons_uri)
for level in geo_levels:
  geo_regions = SelectPartition(level)
  splits = persons.SplitByPartition(geo_regions)
  for persons_in_region in splits:
    x = persons_in_region.Vectorize()
    M = SelectMeasurementsHDMM(W)
    y = x.LaplaceMeasure(M, eps)
    x_hat = LeastSquares(M, y)
    ... additional post-processing ...
```

Top Down algorithm* implemented as Ektelo plan.

(Artistic rendering)

Runs in trusted environment. Impacts sensitivity

Releases noisy measurements; Consumes privacy loss budget

Client-side; no impact on privacy

Operator classes

`Transform`

Filter, project, group, etc.

`Query`

Laplace mechanism

`Query selection`

Strategically choose query sets

`Inference`

Reconcile inconsistencies in noisy answers

`Partition selection`

Dimensionality reduction

Operator classes and instances

Transform	
TV	T-Vectorize
TP	V-SplitByPartition
TR	V-ReduceByPartition

Query	
LM	Vector Laplace

Theorem: if ***red*** and ***orange*** operators are vetted, then any Ektelo plan satisfies DP

Query selection	
SI	Identity
ST	Total
SP	Privelet
SH2	H2
SHB	HB
SG	Greedy-H
SU	UniformGrid
SA	AdaptiveGrids
SQ	Quadtree
SW	Worst-approx
SPB	PrivBayes select

Inference	
LS	Least squares
NLS	Nneg Least squares
MW	Mult Weights
HR	Thresholding

Partition selection	
PA	AHPpartition
PG	Grid
PD	Dawa
PW	Workload-based
PS	Stripe(attr)
PM	Marginal(attr)

Algorithms as Ektelo plans

Operators

Transform	
TV	T-Vectorize
TP	V-SplitByPartition
TR	V-ReduceByPartition

Query	
LM	Vector Laplace

Query selection	
SI	Identity
ST	Total
SP	Privelet
SH2	H2
SHB	HB
SG	Greedy-H
SU	UniformGrid
SA	AdaptiveGrids
SQ	Quadtree
SW	Worst-approx
SPB	PrivBayes select

Inference	
LS	Least squares
NLS	Nneg Least squares
MW	Mult Weights
HR	Thresholding

Partition selection	
PA	AHPpartition
PG	Grid
PD	Dawa
PW	Workload-based
PS	Stripe(attr)
PM	Marginal(attr)

ID	Cite	Algorithm name	Plan signature
1	[8]	Identity	SI LM
2	[39]	Privelet	SP LM LS
3	[17]	Hierarchical (H2)	SH2 LM LS
4	[34]	Hierarchical Opt (HB)	SHB LM LS
5	[22]	Greedy-H	SG LM LS
6	-	Uniform	ST LM LS
7	[15]	MWEM	I:(SW LM MW)
8	[42]	AHP	PA TR SI LM LS
9	[22]	DAWA	PD TR SG LM LS
10	[6]	Quadtree	SQ LM LS
11	[33]	UniformGrid	SU LM LS
12	[33]	AdaptiveGrid	SU LM LS TP[SA LM] LS
13	NEW	DAWA-Striped	PS TP[PD TR SG LM] LS
14	NEW	HB-Striped	PS TP[SHB LM] LS
15	NEW	PrivBayesLS	SPB LM LS
16	NEW	MWEM variant b	I:(SW SH2 LM MW)
17	NEW	MWEM variant c	I:(SW LM NLS)
18	NEW	MWEM variant d	I:(SW SH2 LM NLS)

Algorithms from DPBench [SIGMOD 16]

Novel algorithm variants

Benefits

- Reuse: existing algorithms implemented with reusable operators
- Reduces code verification effort
- Improved operator implementations
- New variants of algorithm easy to construct (improved accuracy!)

Architecture for Private Computation?

- Separate concerns:
  - Transformations
  - Measurement selection
  - Measurement
  - Post-processing (consistency, synthetic data, inference)
- Benefits of modularity:
  - Reduce scope of privacy verification
  - Diverse contributors: relevant expertise differs by component

Outline

- Introduction
- DPBench: principled empirical evaluations of accuracy
- Ektelo: framework for private computation
- PrivateSQL: differentially private SQL query engine

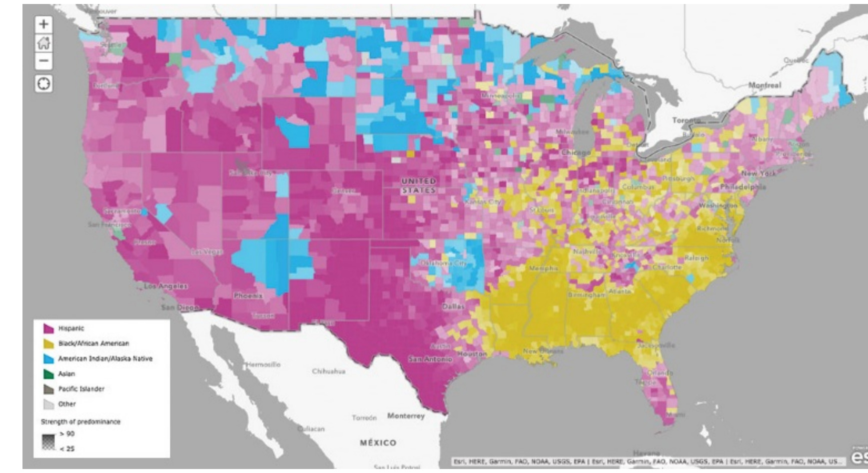
Motivations for Private SQL

- Towards a declarative interface for query answering
- **Complex queries over multi-relational data**
- Privacy at multiple resolutions

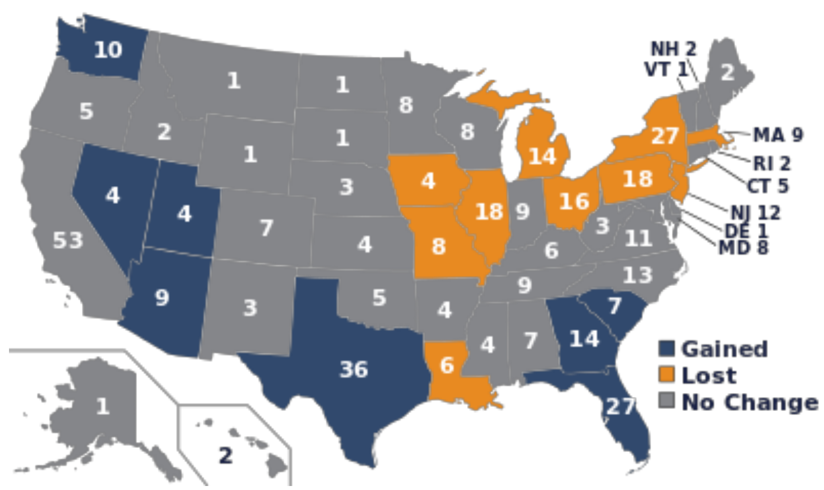
Joint work with Gerome Miklau, Ashwin Machanavajhalla, Ios Kotsogiannis, Yuchao Tao, Xi He, Maryam Fanaeepour

Population

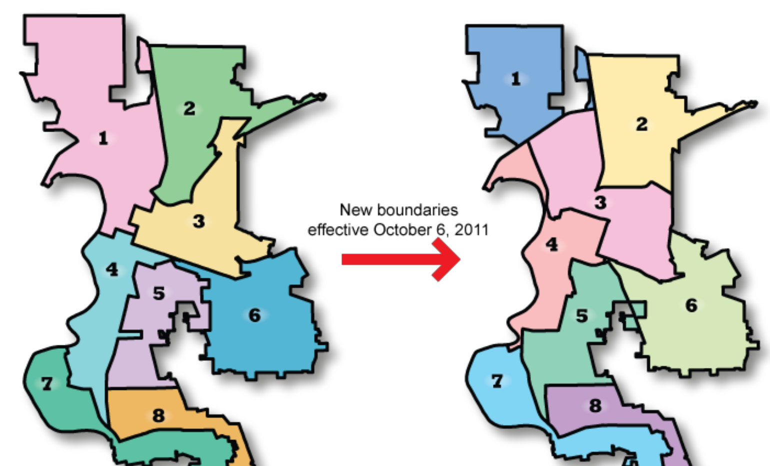
United States Census Bureau



Statistics



Congressional Apportionment



Redistricting

Fund allocations to schools

Minority Language Voting Rights

Statistics Released by US Census Bureau

Person

ID	Sex	…	HID
122	M	…	H6
123	F	…	H6
124	M	…	H7
125	M	…	H8
126	F	…	H8

Household

HID	…	Geo
H6	…	CA
H7	…	FL
H8	…	NC

Census Summary File 1 (SF-1)

- "Number of males between 18 and 21 years old", …
- "Number of people living in owned houses of size 3 where the householder is a married Hispanic male", …

At all levels of geography (state, county, tract, block)

Complex Queries

- Linear queries on households

```
SELECT COUNT(*)
FROM ( SELECT hid, COUNT(*) AS CNT
          FROM Persons p, (SELECT hid
                                     FROM Persons p1, Persons p2
                                     WHERE p1.hid = p2.hid
                                          AND p1.Rel = ‘householder’
                                          AND p2.Rel = ‘spouse’
                                          AND ( (p1.sex= ‘M’ AND p2.sex = ‘F’)
                                          OR (p1.sex= ‘F’ AND p2.sex = ‘M’))
                                     GROUP BY hid) AS h
          WHERE p.hid = h.hid AND p.Rel = ‘child’
               AND p.Age < 18
          GROUP BY hid)
WHERE CNT >= 1
```

Complex Queries

- Linear queries on households

```
SELECT COUNT(*)
FROM ( SELECT hid, COUNT(*) AS CNT
```

Count of the number of households
where the householder age in [15..64]
AND it's a husband-wife family
AND there is at least one related child under 18.

```
                                    OR (p1.sex= 'F' AND p2.sex = 'M'))
                               GROUP BY hid) AS h
        WHERE p.hid = h.hid AND p.Rel = 'child'
            AND p.Age < 18
        GROUP BY hid)
WHERE CNT >= 1
```

Complex Queries

- Queries on people living in households

```
SELECT COUNT(*)
FROM Person p
Where p.Age < 18 AND
        p.hID in (SELECT hID
                  FROM Person p
                  WHERE p.Rel = "householder"
                        AND p.Race = "Asian")
```

Complex Queries

- Queries on people living in households

```
SELECT COUNT(*)
FR
W
       p.hID in (SELECT hID
                 FROM Person p
                 WHERE p.Rel = "householder"
                       AND p.Race = "Asian")
```

Count of the number of people under 18
living in households with an Asian householder

Complex queries

- Degree distribution query or count of count histogram

```
SELECT cnt, COUNT(*)
FROM (SELECT hID, COUNT(*) as cnt
          FROM Person p
         GROUP BY hID)
GROUPBY cnt
ORDER BY cnt
```

Complex queries

- Degree distribution query or count of count histogram

```
SELECT cnt, COUNT(*)
FROM (SELECT hID, COUNT(*) as cnt
        GROUP BY hID)
GROUPBY cnt
ORDER BY cnt
```

For every household size,
release the number of households of that size

Motivations for Private SQL

- Complex queries over multi-relational data

- **Privacy at multiple resolutions**

Privacy requirement

- Title 13 Section 9

Neither the secretary nor any officer or employee …
… make any publication whereby the data furnished by any particular establishment or individual under this title can be identified …

- In some data products, only properties of people need to be hidden, and in other products, properties of households also need to be hidden.

Privacy at multiple resolutions

Person-privacy: hide properties of people

Household-privacy: hide properties of households and the people within them.

Person

ID	Sex	…	HID
122	M	…	H6
123	F	…	H6
124	M	…	H7
125	M	…	H8
126	F	…	H8

Household

HID	…	Geo
H6	…	CA
H7	…	FL
H8	…	NC

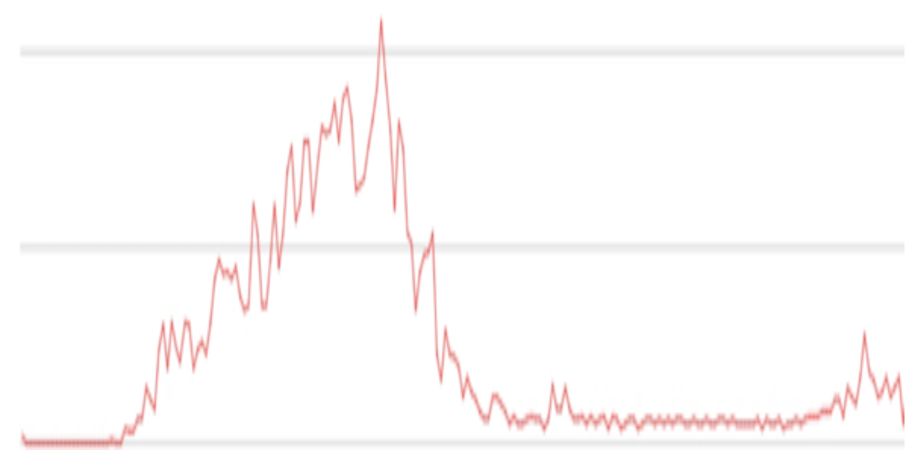
Edge-privacy: hide the presence of an edge

Node-privacy: hide the presence of a node and all edges incident to it.

Event-privacy: hide sensor reading

Window-privacy: hide readings in (t-w, t]

User-privacy: hide all sensor readings



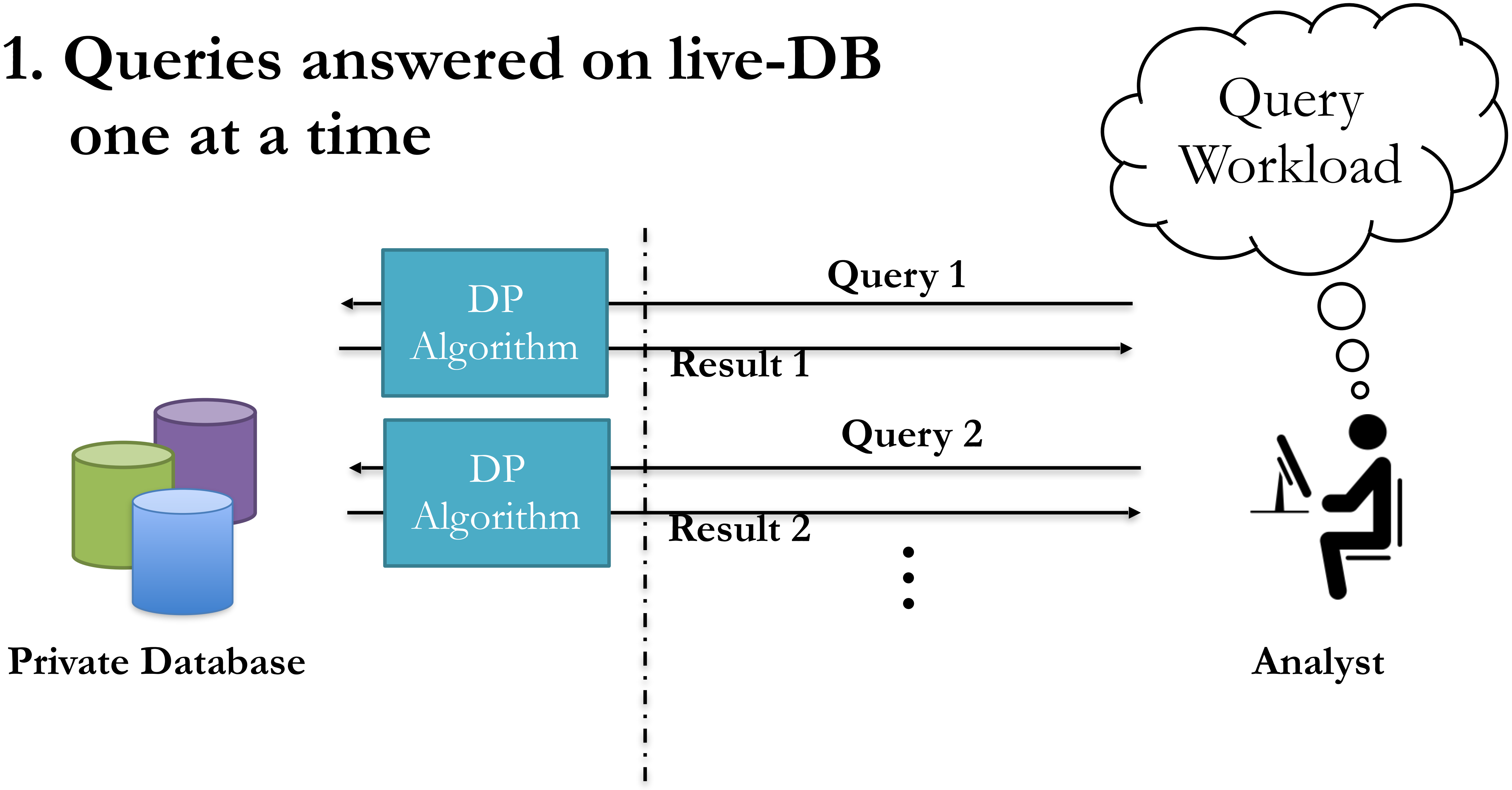
Goals of Private SQL

- *Automatically* generates differentially private code to accurately answer the queries specified in a high level language (SQL)

- Ensures a *fixed privacy budget* across all queries posed by the analyst.

- Enables privacy to be specified at *multiple resolutions*.

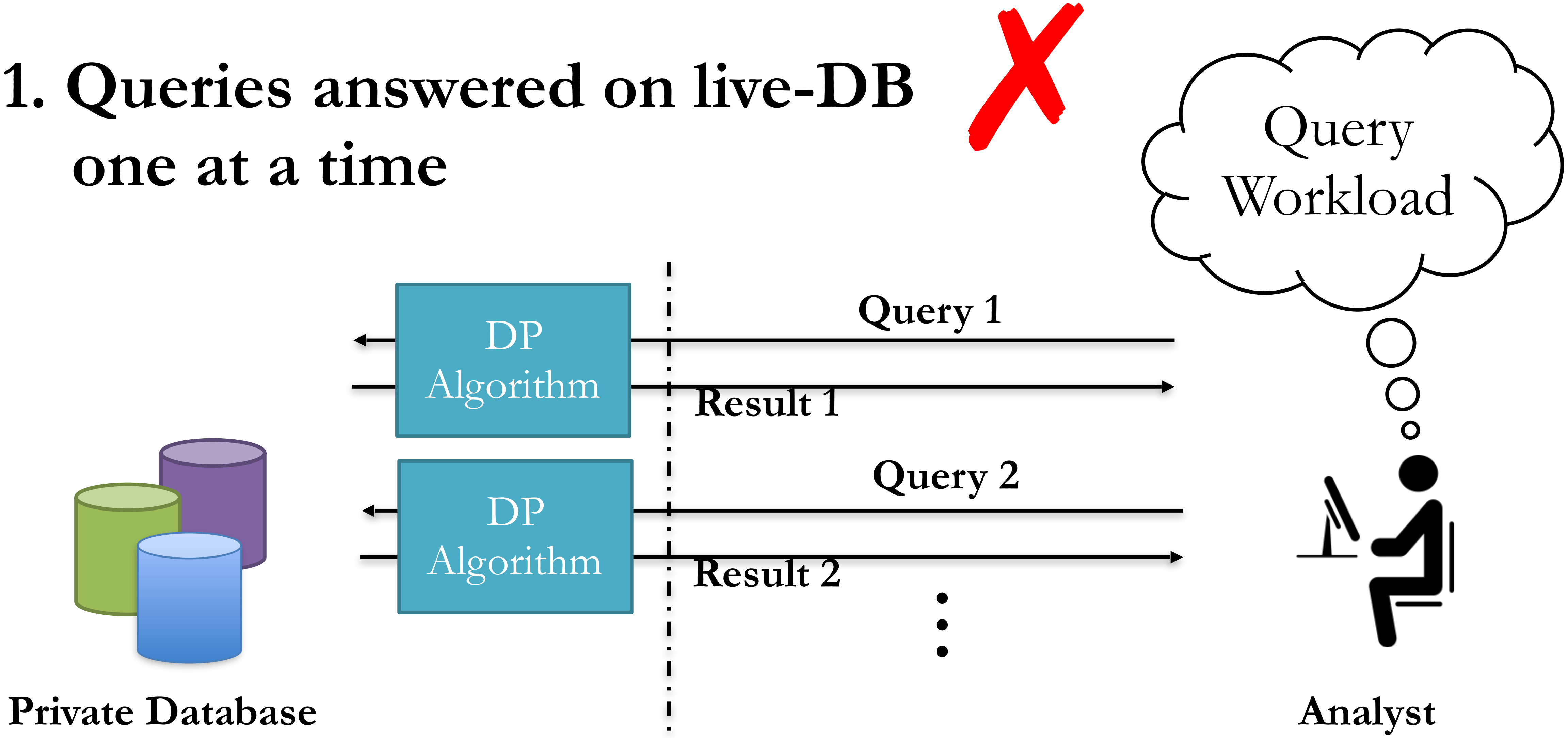
1. Queries answered on live-DB one at a time



Example: FLEX [VLDB18]

- Deployed at Uber.

1. Queries answered on live-DB one at a time



Unbounded Privacy Loss

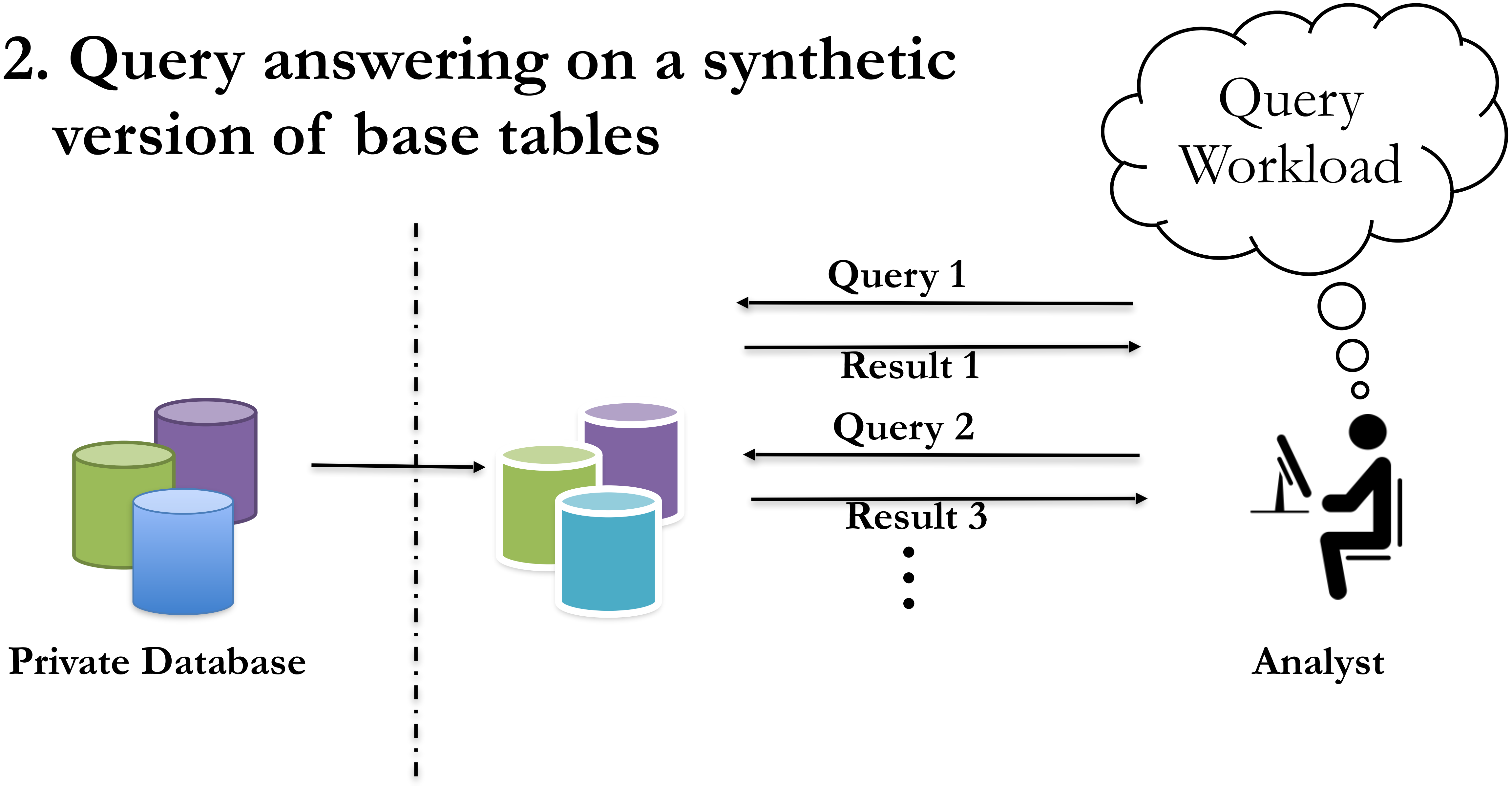
- Unless the system decides to shut off future queries, the privacy loss keeps increasing.

Inflexible privacy semantics (for Flex specifically)

- Hides any row in DB, but this may not align with privacy in particular context.

Other concerns: inconsistency between answers, side channel attacks

2. Query answering on a synthetic version of base tables



Examples:

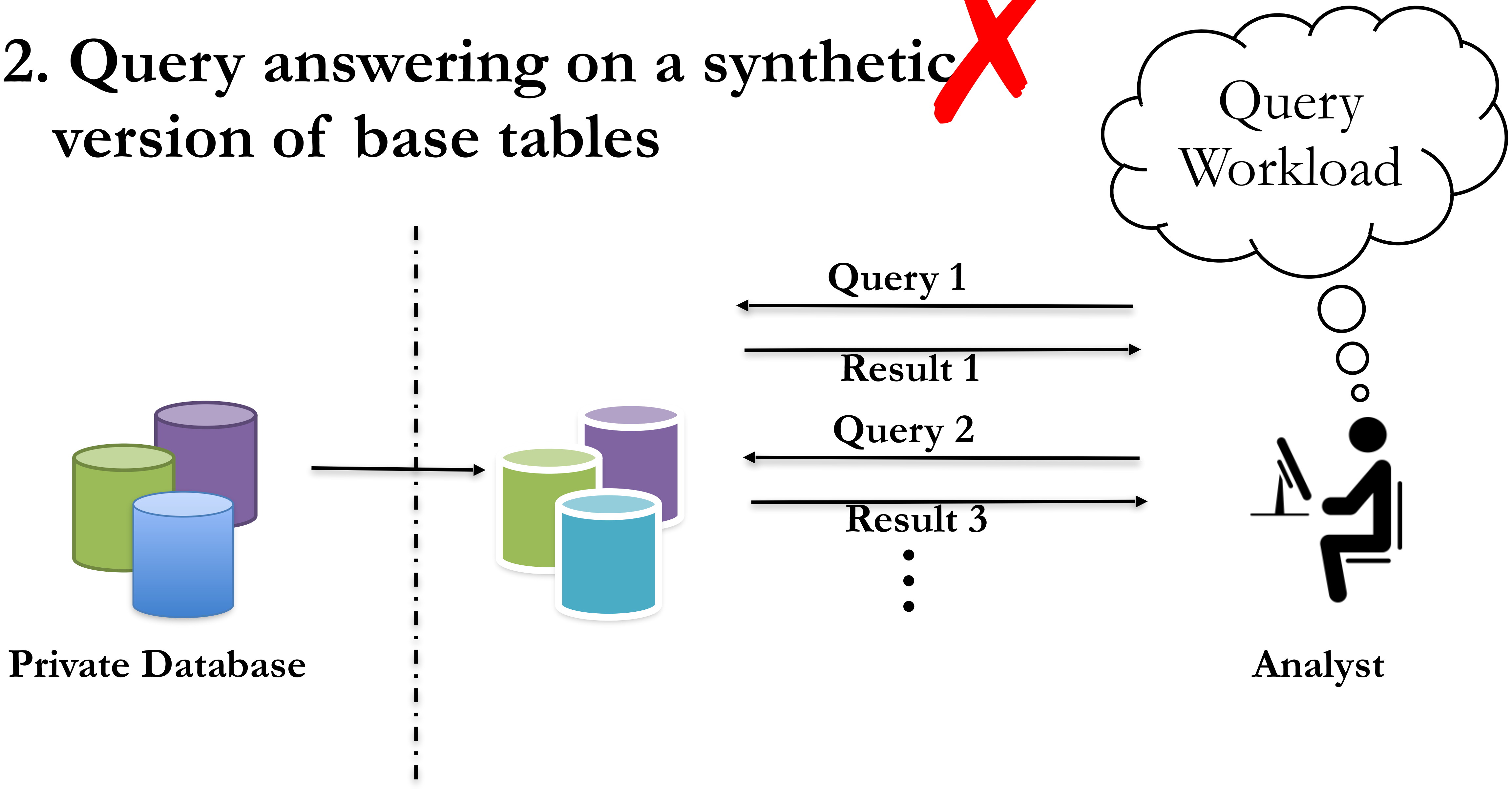
HDMM [VLDB18], MWEM [NIPS12] …

- Output a histogram tunes to query workload

PrivBayes [SIGMOD14], Private Synthetic Data using GANs [NIST Challenge 18]

- Generates a synthetic database in the same schema as input

2. Query answering on a synthetic version of base tables



No support for multi-relational tables

Joins computed on synthetic tables have very high error.

Defining privacy at multiple resolutions

Edge-privacy: hide the presence of an edge

Node-privacy: hide the presence of a node and all edges incident to it.

Person-privacy: hide properties of people

Household-privacy: hide properties of households and the people within them.

Person

ID	Sex	…	HID
122	M	…	H6
123	F	…	H6
124	M	…	H7
125	M	…	H8
126	F	…	H8

Household

HID	…	Geo
H6	…	CA
H7	…	FL
H8	…	NC

Multi-resolution privacy in PrivateSQL

- **Policy:** A specification of the base relation that is the *primary private object*.

- **Neighboring Databases:**
  - Add or remove a row *r* in the primary private relation
  - Add or remove all rows in other tables that *transitively refer* to the row *r* in the primary private relation

Multi-resolution privacy in PrivateSQL

Person

ID	Sex	…	HID
122	M	…	H6
123	F	…	H6
124	M	…	H7
125	M	…	H8
126	F	…	H8

Household

HID	…	Geo
H6	…	CA
H7	…	FL
H8	…	NC

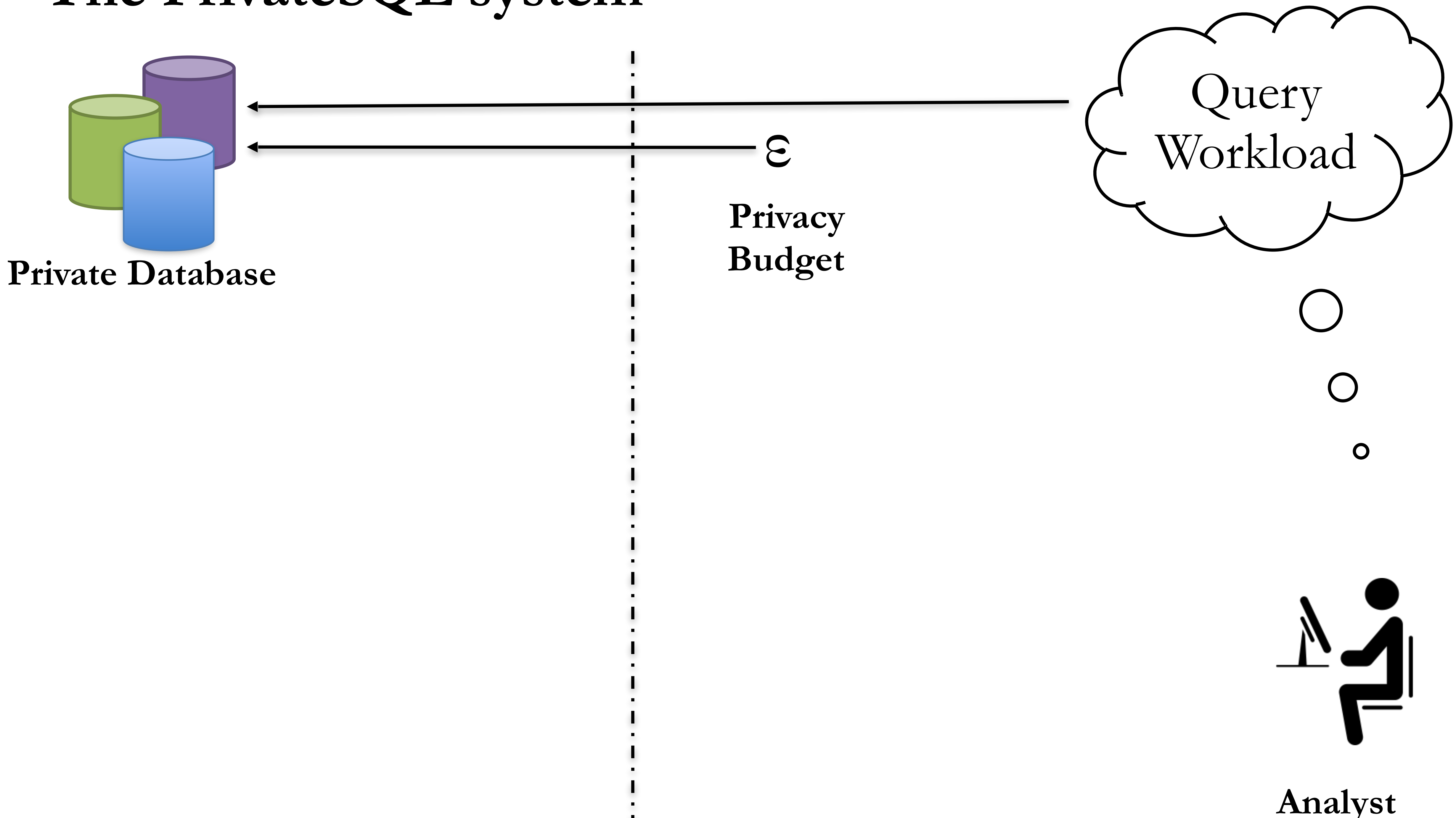
Person-privacy:

- Person is the primary private relation
- Adding or removing an person record does not affect the household table.

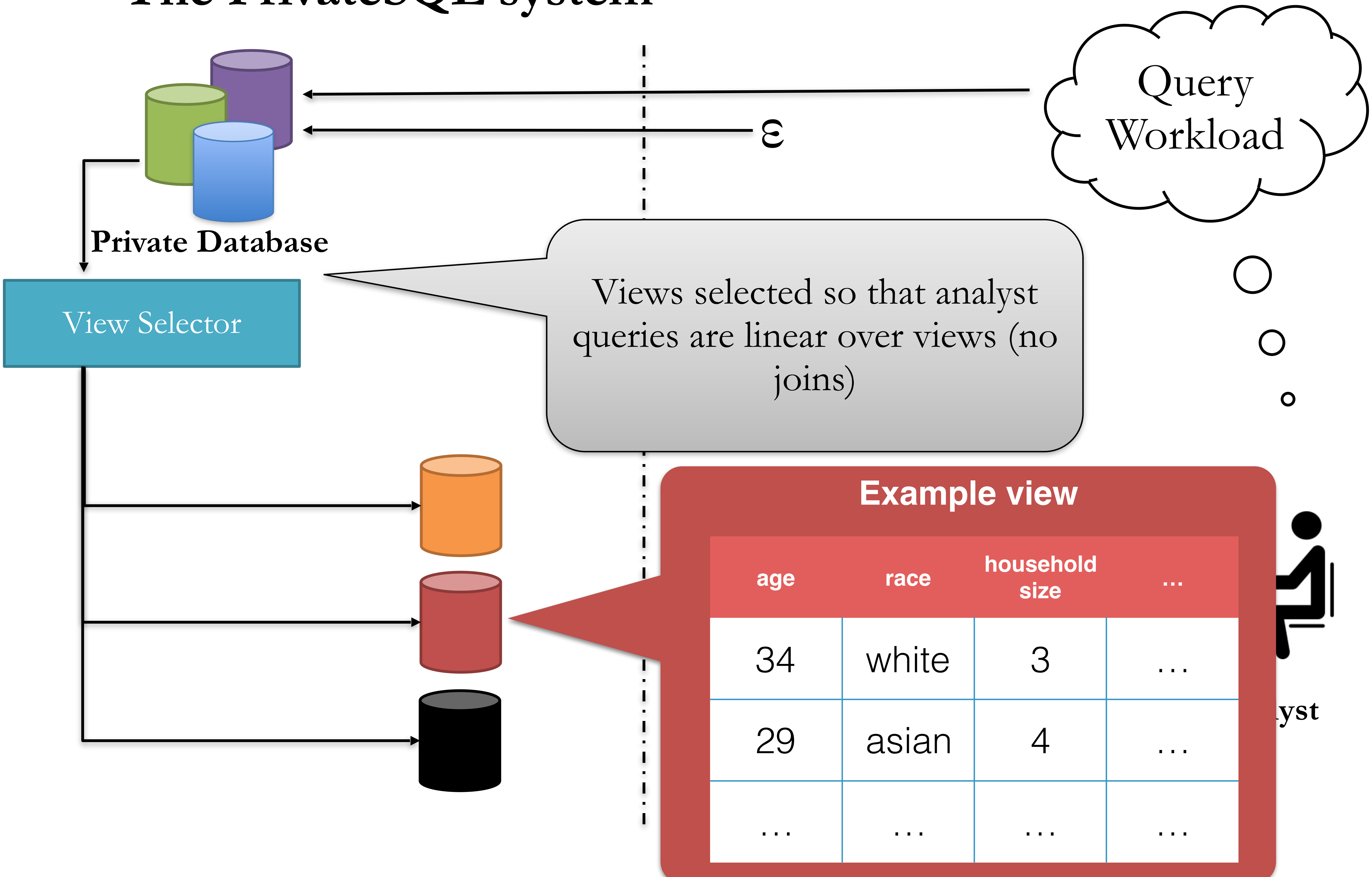
Household-privacy:

- Household is the primary private relation
- Adding or removing a row *r* from household removes all rows in person that refer to *r* in household table.

The PrivateSQL system

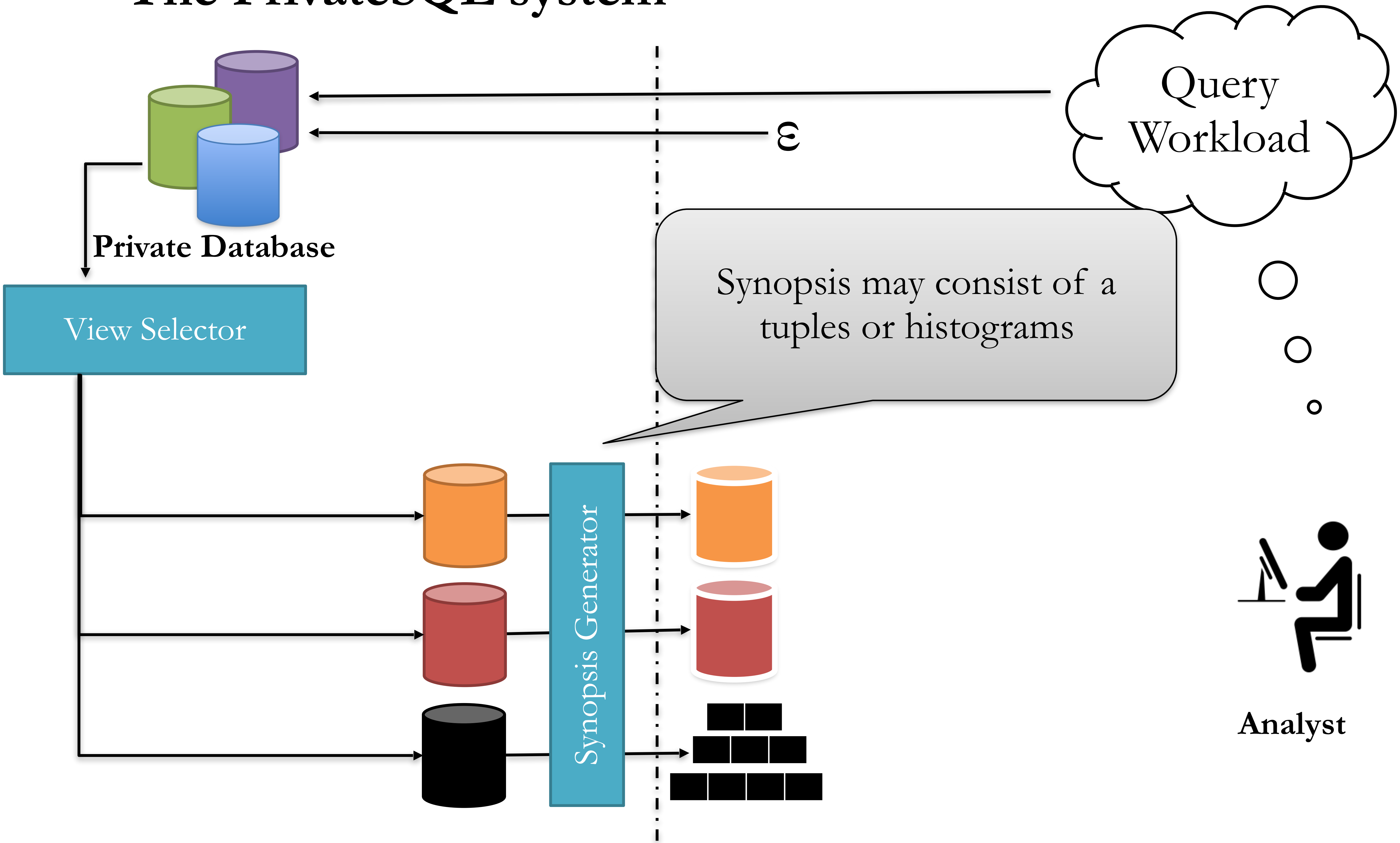


The PrivateSQL system

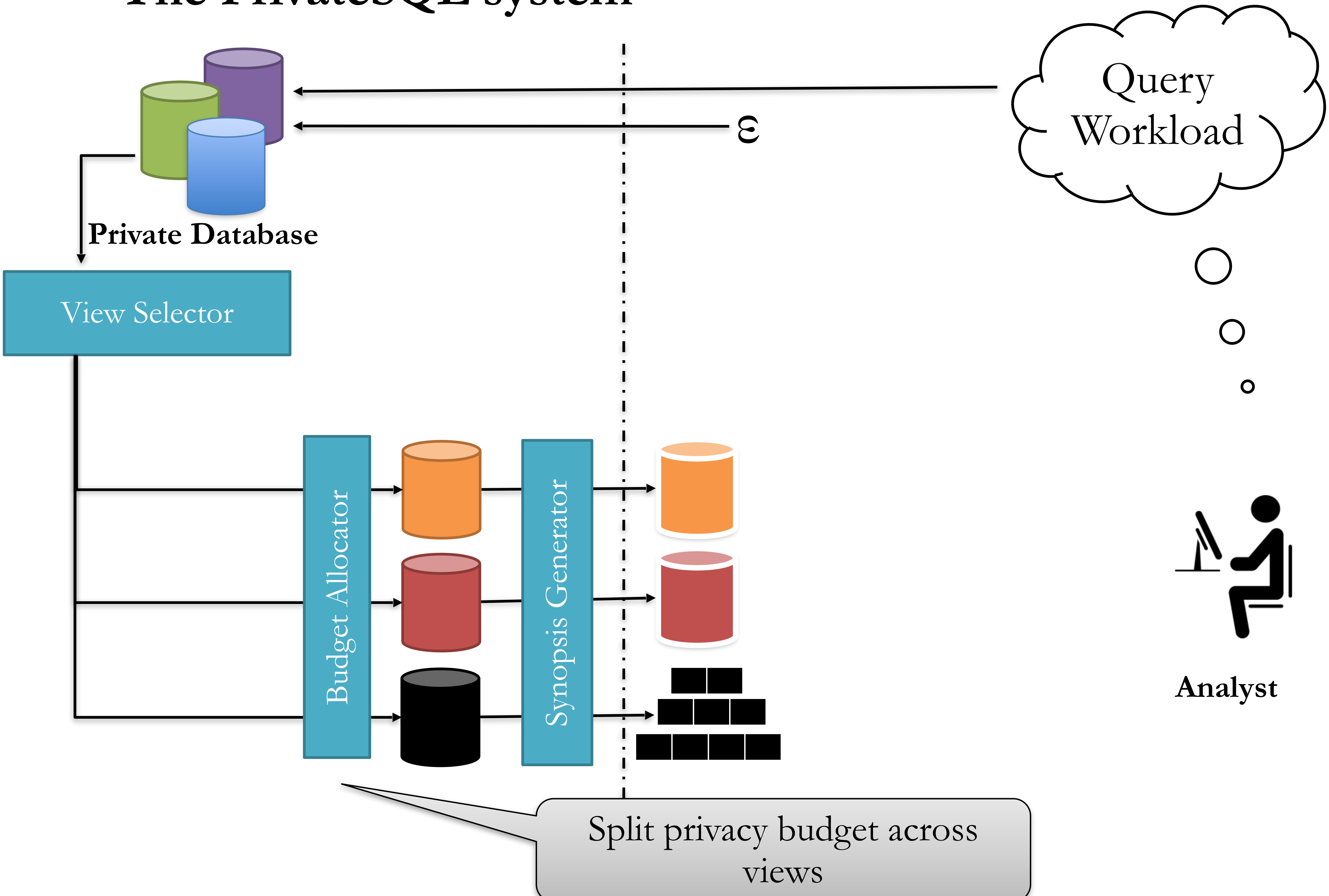


age	race	household size	…
34	white	3	…
29	asian	4	…
…	…	…	…

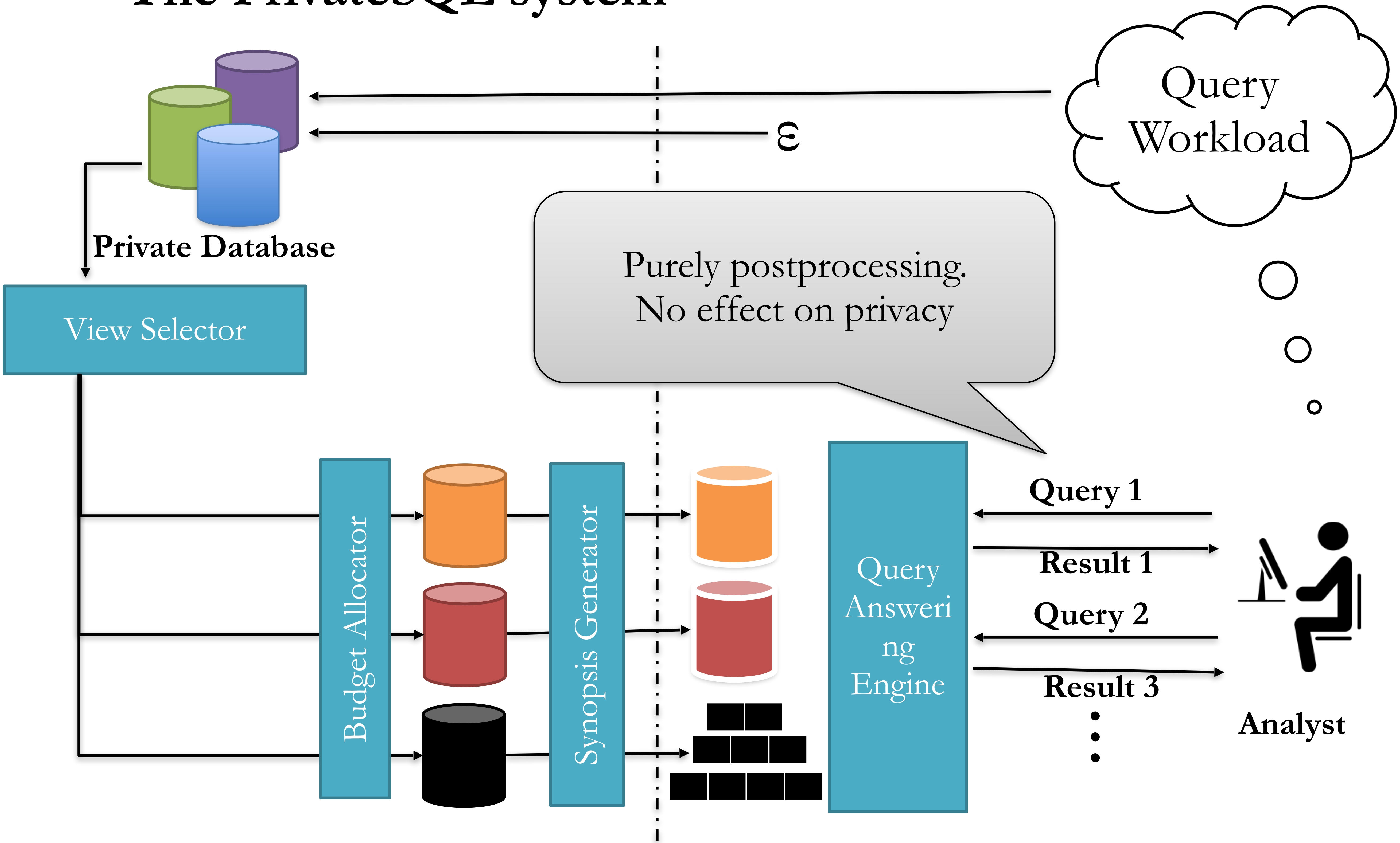
The PrivateSQL system



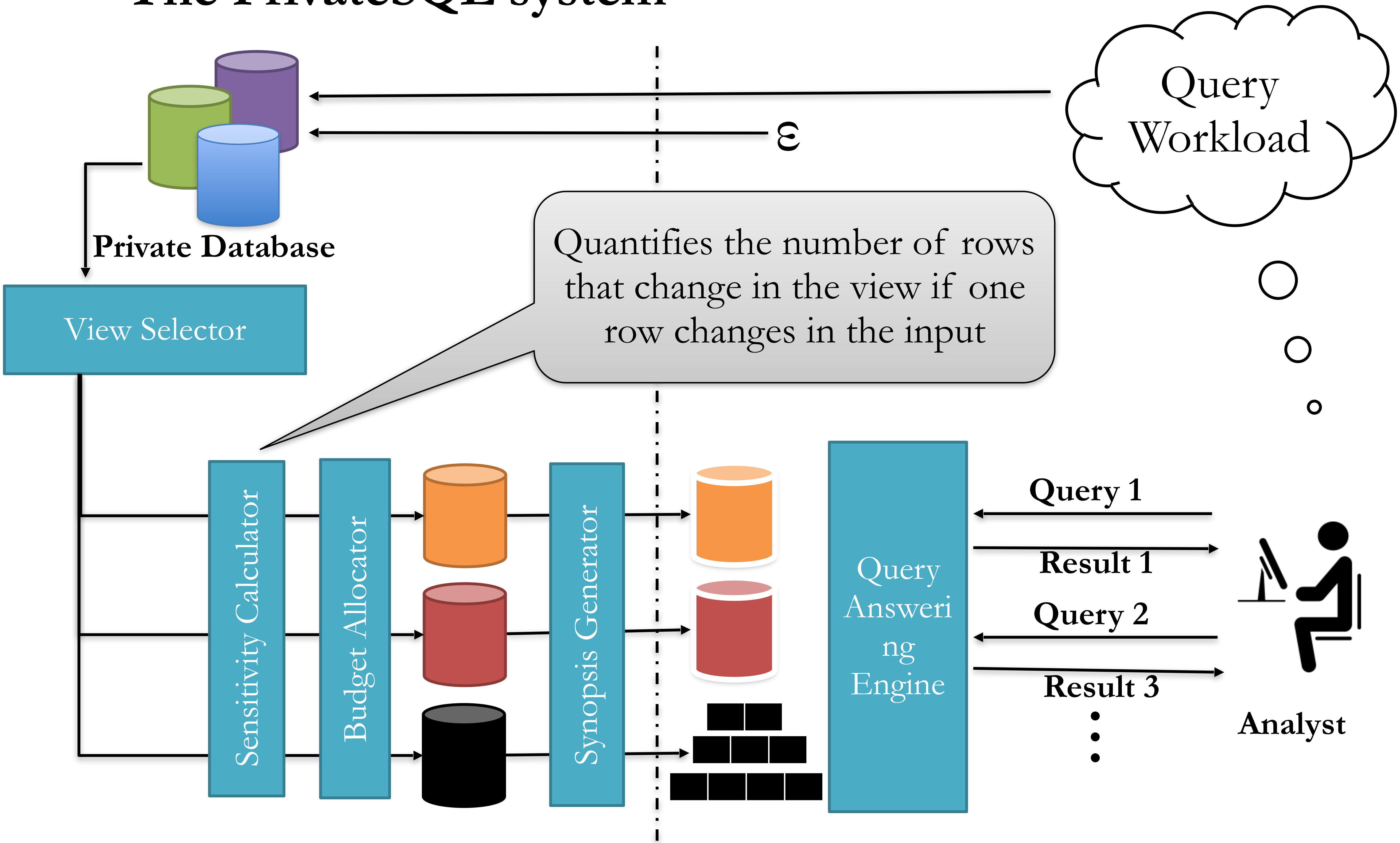
The PrivateSQL system



The PrivateSQL system



The PrivateSQL system



Addressing view sensitivity

- View is complex SQL query; evaluation is hard [Arapinis et al. ICALP16]

➡ Rule-based sensitivity bound calculator
(builds on PINQ, Flex, with new rules: joins on keys)

Addressing view sensitivity

- View is complex SQL query; evaluation is hard [Arapinis et al. ICALP16]

→ Rule-based sensitivity bound calculator
(builds on PINQ, Flex, with new rules: joins on keys)

- Global sensitivity may be high / unbounded

Example view

age	race	household size	…
34	white	3	…
29	asian	4	…
…	…	…	…

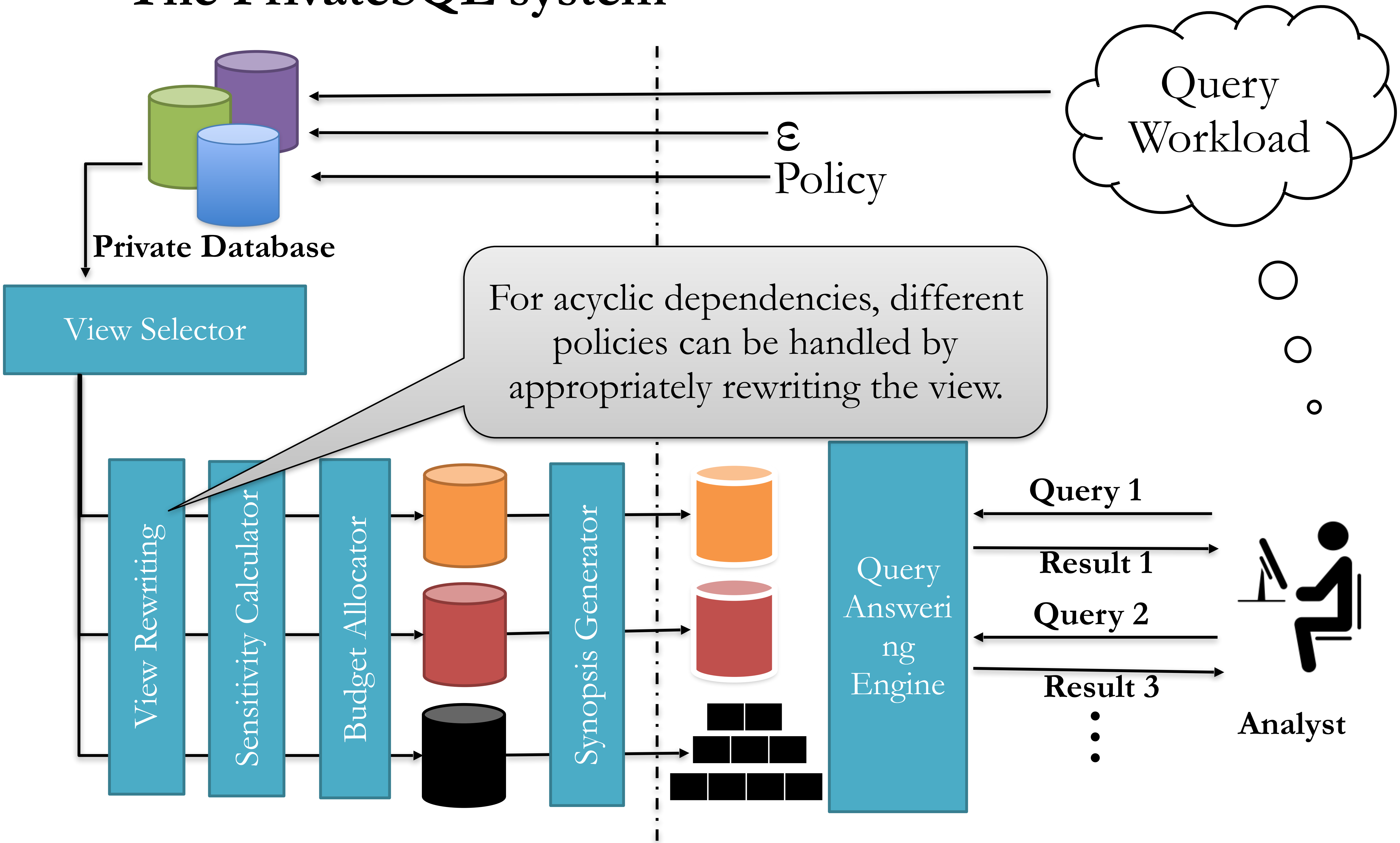
Addressing view sensitivity

- View is complex SQL query; evaluation is hard [Arapinis et al. ICALP16] ➡ Rule-based sensitivity bound calculator (builds on PINQ, Flex, with new rules: joins on keys)

- Global sensitivity may be high / unbounded ➡ Truncate "outliers"

- Calculation depends on privacy resolution level (e.g., person vs. household) ➡ View rewriting

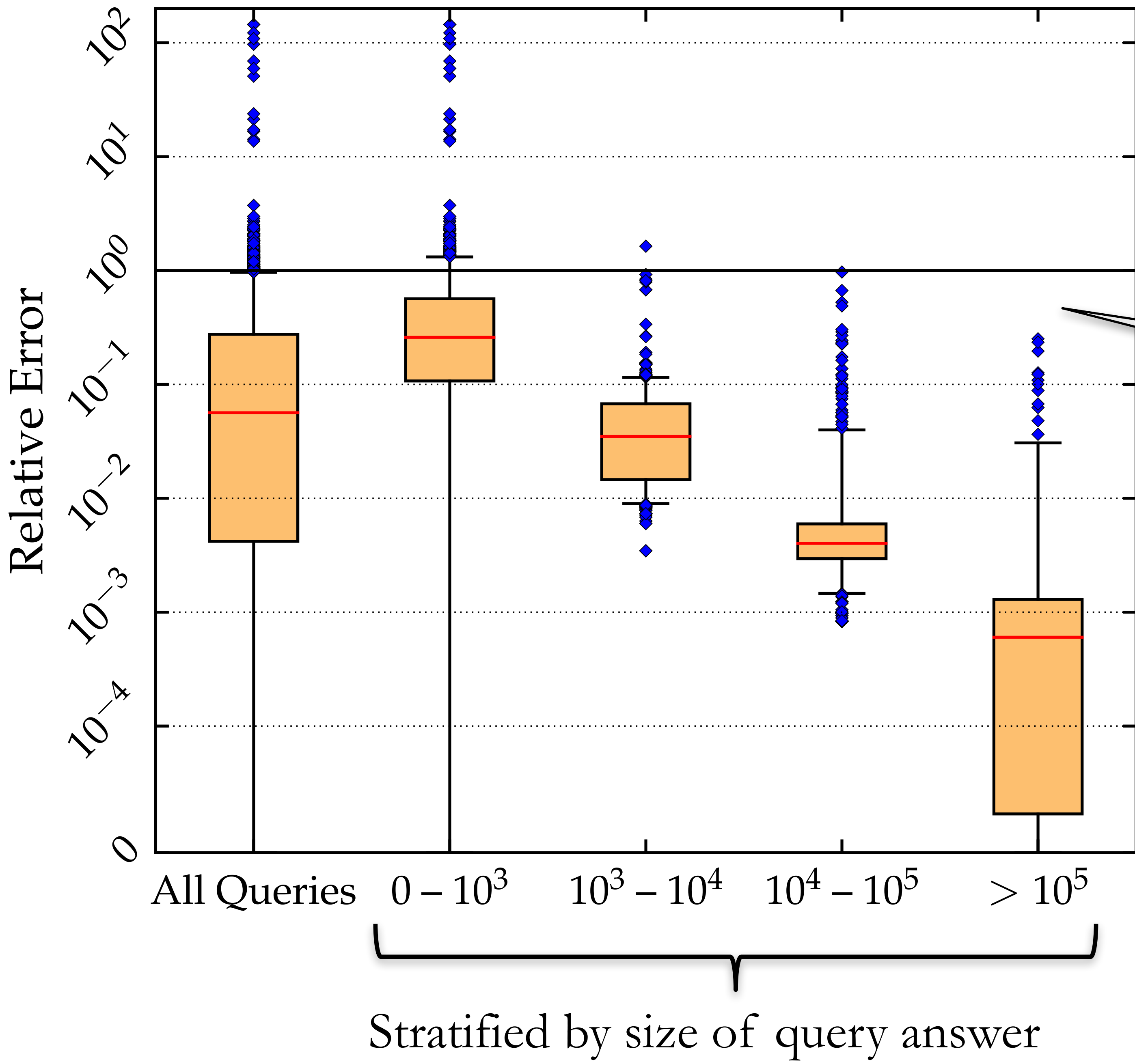
The PrivateSQL system



Empirical evaluation

- Dataset: A synthetic census dataset
  - person(id, sex, gender, age, race, relationship, hid) and household(hid, location)
  - Restricted to the state of NC
  - 5.4 million people and 2.7 million households
- Queries: 3493 counting queries from the 2010 Summary file 1.
  - "Number of males between 18 and 21 years old."
  - "Number of people living in owned houses of size 3 where the householder is a married Hispanic male."
- Views: PrivateSQL generated 17 views

Overall Error



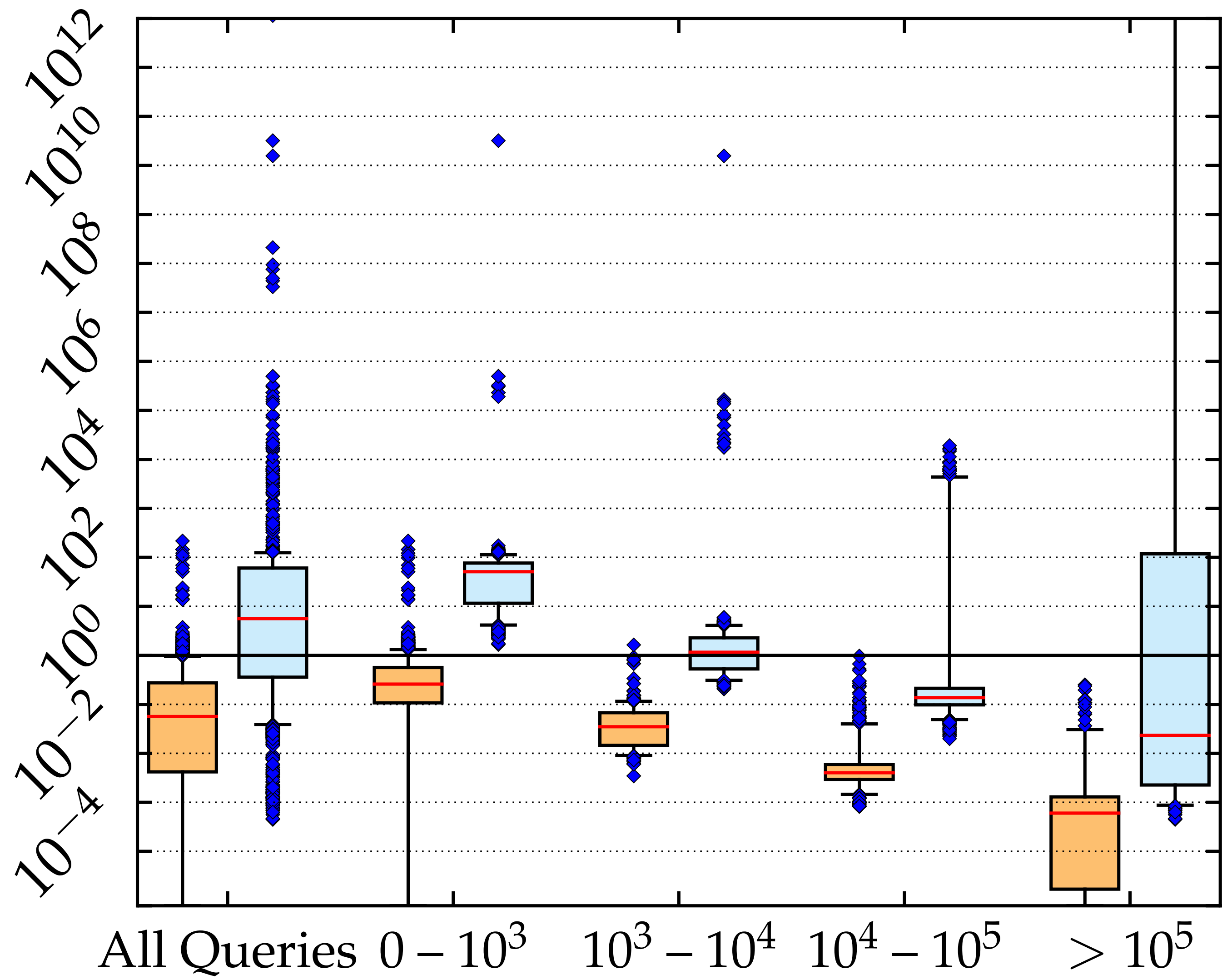
Privacy Budget: 1.0

Policy: Hiding a row in person table.

Outputting 0 for all queries gives relative error 1.

For queries with sufficiently large answers, the relative error is small.

Comparison to one-query-at-a-time approach



Stratified by size of query answer

Privacy Budget: 1.0

Competitor: A baseline based on FLEX [VLDB18]

Improvement over FLEX can be attributed to:

- Tighter sensitivity bounds
- Truncation instead of smoothing
- Better composition (across queries sharing view)

Key highlights of PrivateSQL

- *View Selection + Synopsis Generation* gets us away from one query at a time answering
  - Bounded privacy loss, consistent answers, avoids some side channel attacks
- *Privacy can be defined at multiple resolutions*
  - Able to specify a rich set of policies, and automatically rewrite views based on policy
- *Computing sensitivity for complex SQL queries* is challenging
  - Our techniques give an order of magnitude tighter bounds on sensitivity than prior work.
- *Modular architecture allows independent innovation in each component*

Some Open Questions

- More sophisticated truncation [Raskhodnikova FOCS 16; Chen, SIGMOD13]

- Theoretical characterization of bias-variance tradeoff of truncation

- Quantifying error in the answers

Summary

- Benchmarks can provide valuable insight and focus research community
- Modular architectures like Ektelo can simplify and accelerate algorithm development.
- PrivateSQL towards declarative interface for complex queries over multi-relational data

Thanks

[SIGMOD16] Hay et al, "Principled Evaluation of Differentially Private Algorithms using DPBench" https://www.dpcomp.org/

[SIGMOD18] Zhang et al, "Ektelo: A Framework for Defining Differentially-Private Computations" https://ektelo.github.io/

[CIDR19] Kotsogiannis et al, "Architecting a Differentially Private SQL Engine"

Other related work:

[SIGMOD09] McSherry, "Privacy Integrated Queries"

[NIPS12] Hardt et al, "A Simple and Practical Algorithm for Differentially Private Data Release"

[TODS17] Zhang et al, "PrivBayes: Private Data Release via Bayesian Networks"

[VLDB19] Johnson et al, "Towards Practical Differential Privacy for SQL Queries"

[JPC17] Ebadi and Sands. Featherweight PINQ.

[FOCS16] Raskhodnikova and Smith, "Lipschitz Extensions for Node-Private Graph Statistics and the Generalized Exponential Mechanism"

[SIGMOD13] Chen and Zhou, "Recursive mechanism: towards node differential privacy and unrestricted joins"