Differentially-Private Batch Query Answering

Exploiting the Workload vs. Exploiting the Data

Gerome Miklau

University of Massachusetts, Amherst

DIMACS Workshop on Recent Work on Differential Privacy across Computer Science • October 2012

Batch (non-interactive) query answering

- **Given:** a fixed set of queries
  - complex data analysis task into simpler queries.
  - multiple users each issuing one or more queries.
  - uncertainty about the eventual query answers needed--design workload to include all queries possibly of interest.

Batch (non-interactive) query answering

- **Given:** a fixed set of queries
  - complex data analysis task into simpler queries.
  - multiple users each issuing one or more queries.
  - uncertainty about the eventual query answers needed--design workload to include all queries possibly of interest.

the “workload”

Batch (non-interactive) query answering

- **Given:** a fixed set of queries

the “workload”

  - complex data analysis task into simpler queries.
  - multiple users each issuing one or more queries.
  - uncertainty about the eventual query answers needed--design workload to include all queries possibly of interest.

- **Goal**: release answers to all queries under ε- or (ε, δ)-differential privacy.

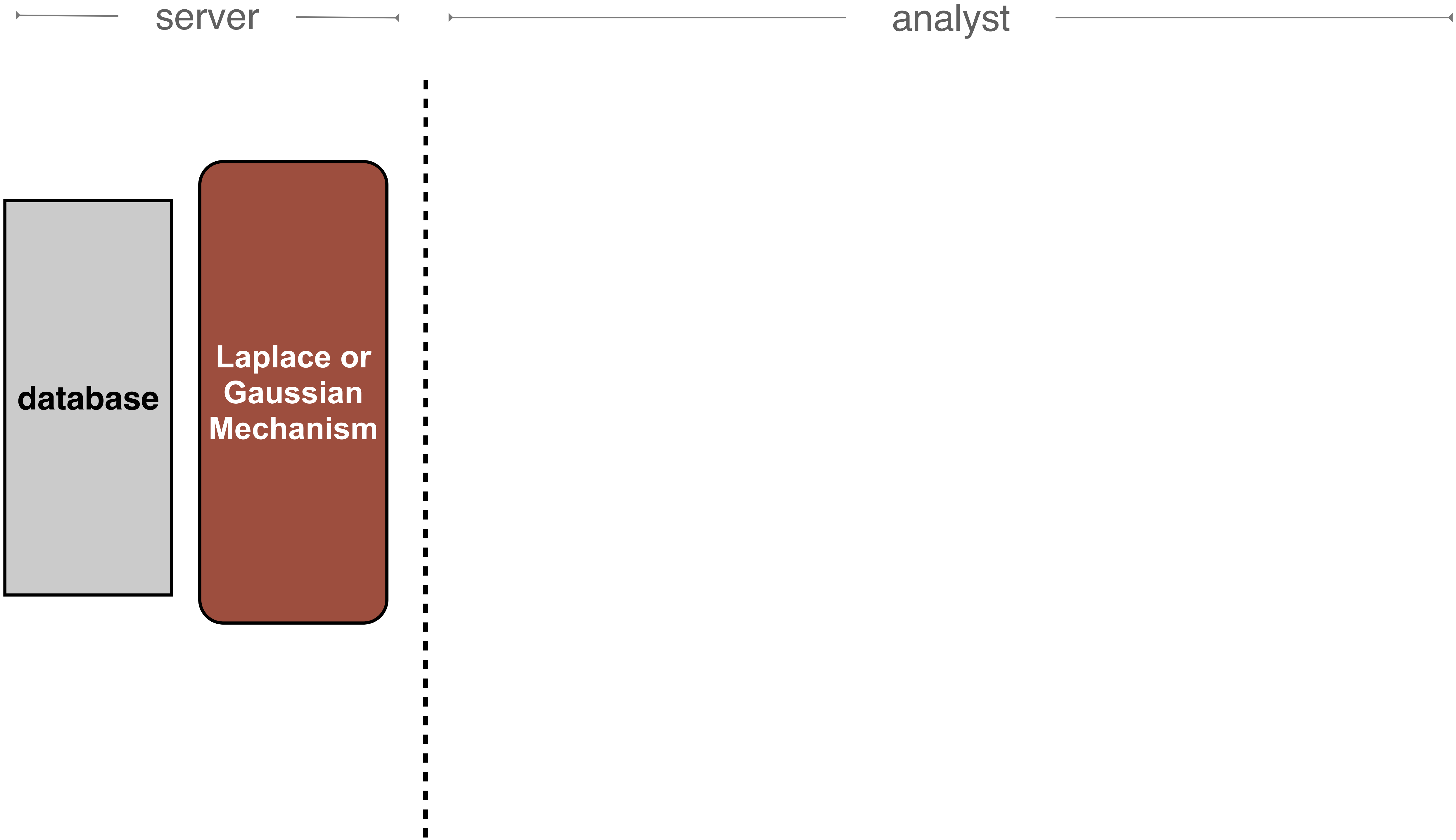
Batch (non-interactive) query answering

- **Given:** a fixed set of queries

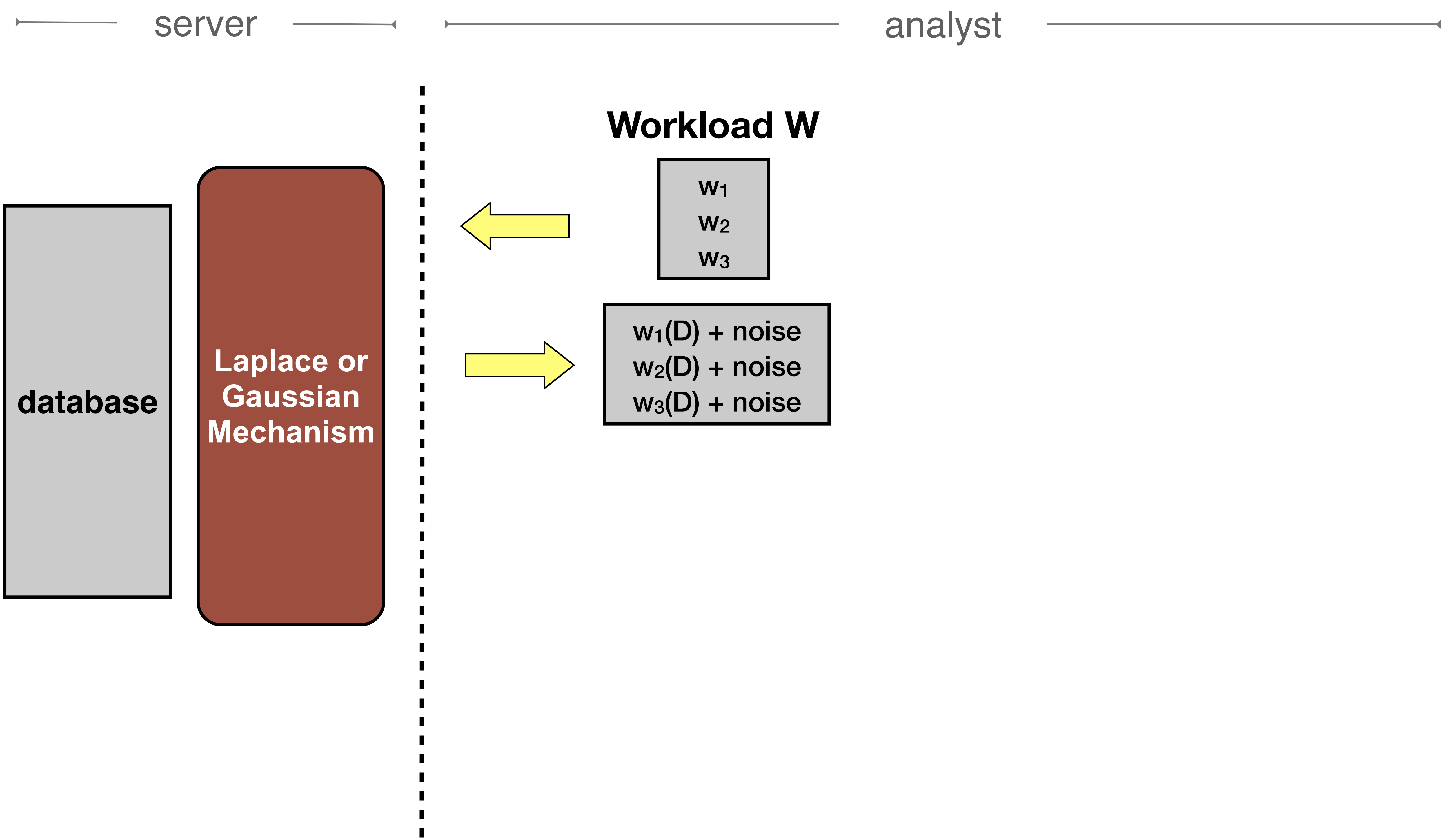
  the “workload”

  - complex data analysis task into simpler queries.
  - multiple users each issuing one or more queries.
  - uncertainty about the eventual query answers needed--design workload to include all queries possibly of interest.
- **Goal**: release answers to all queries under ε- or (ε, δ)-differential privacy.
- Linear counting queries
  - includes predicate counting queries, spatial queries, multi-dimensional range queries, marginals, data cubes, etc.

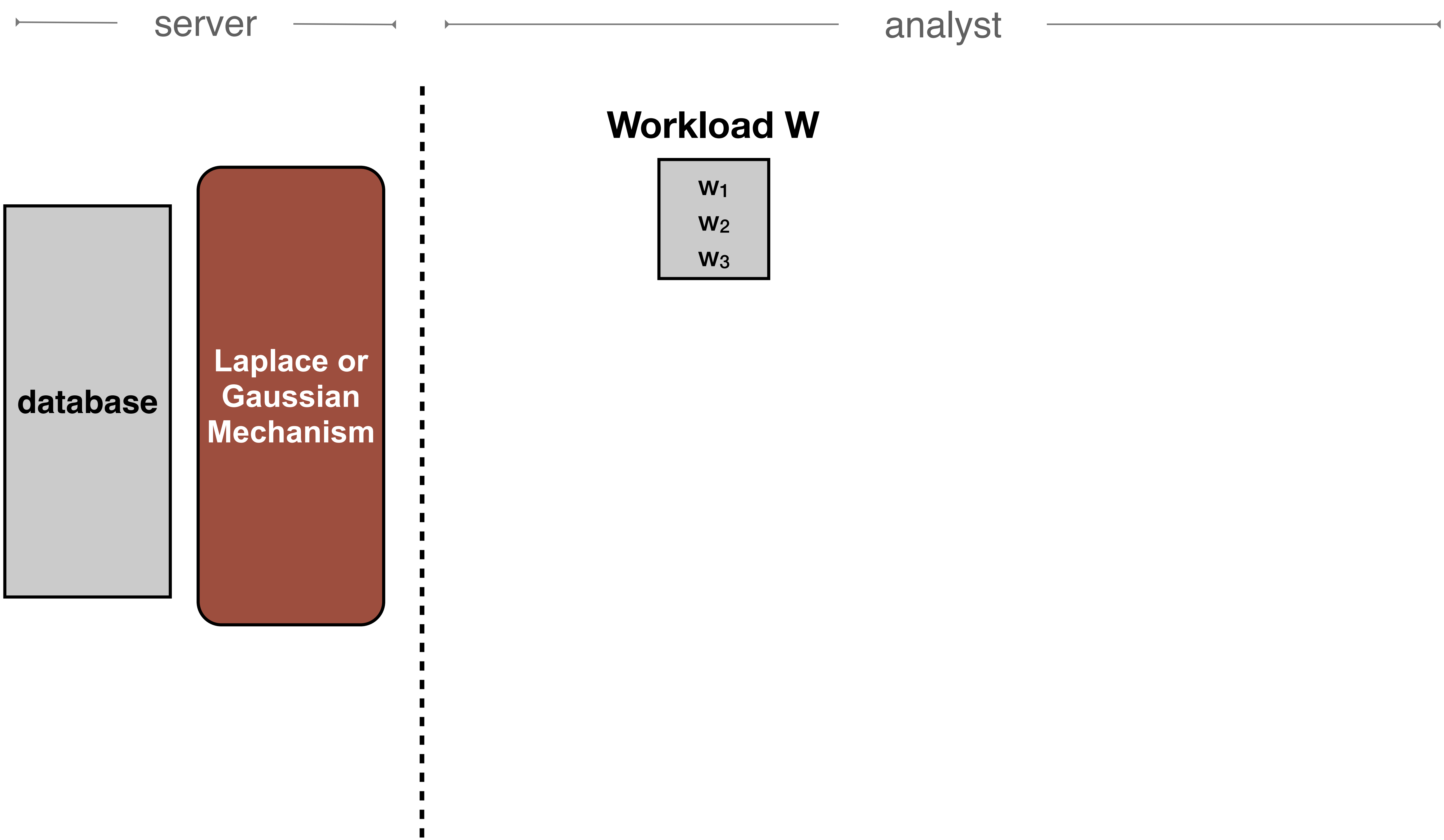
Approach 1: workload-aware mechanisms



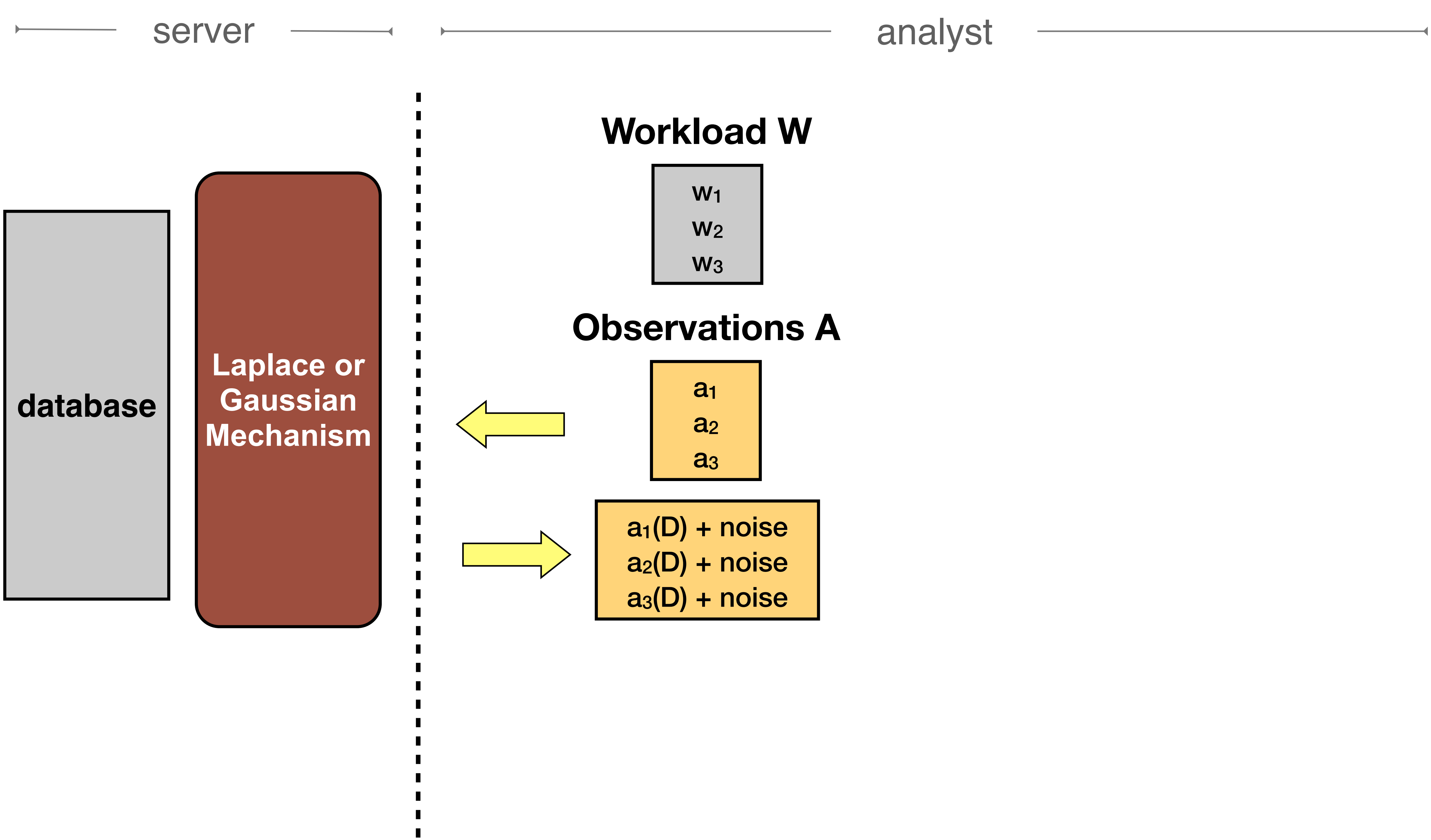
Approach 1: workload-aware mechanisms



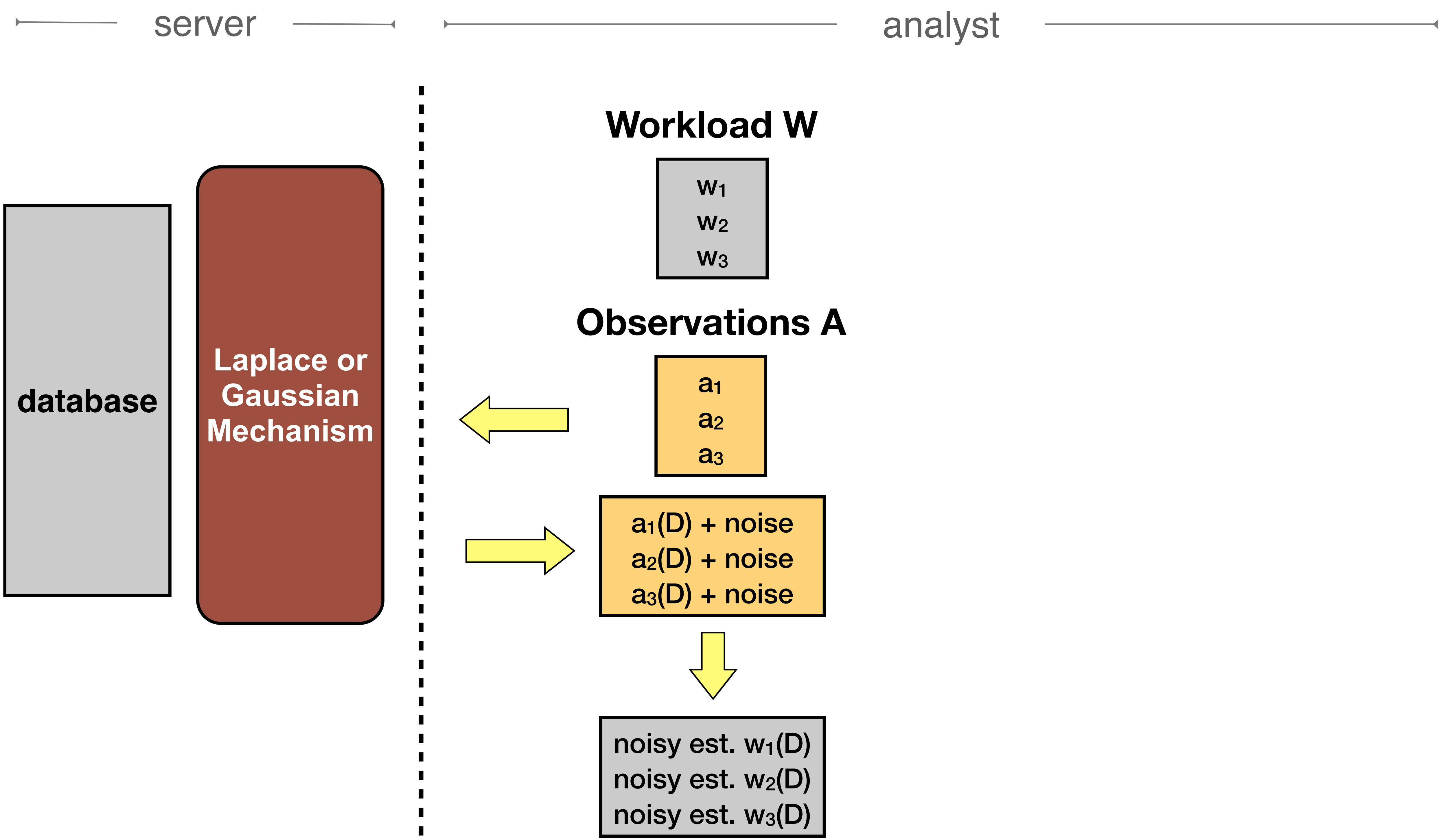
Approach 1: workload-aware mechanisms



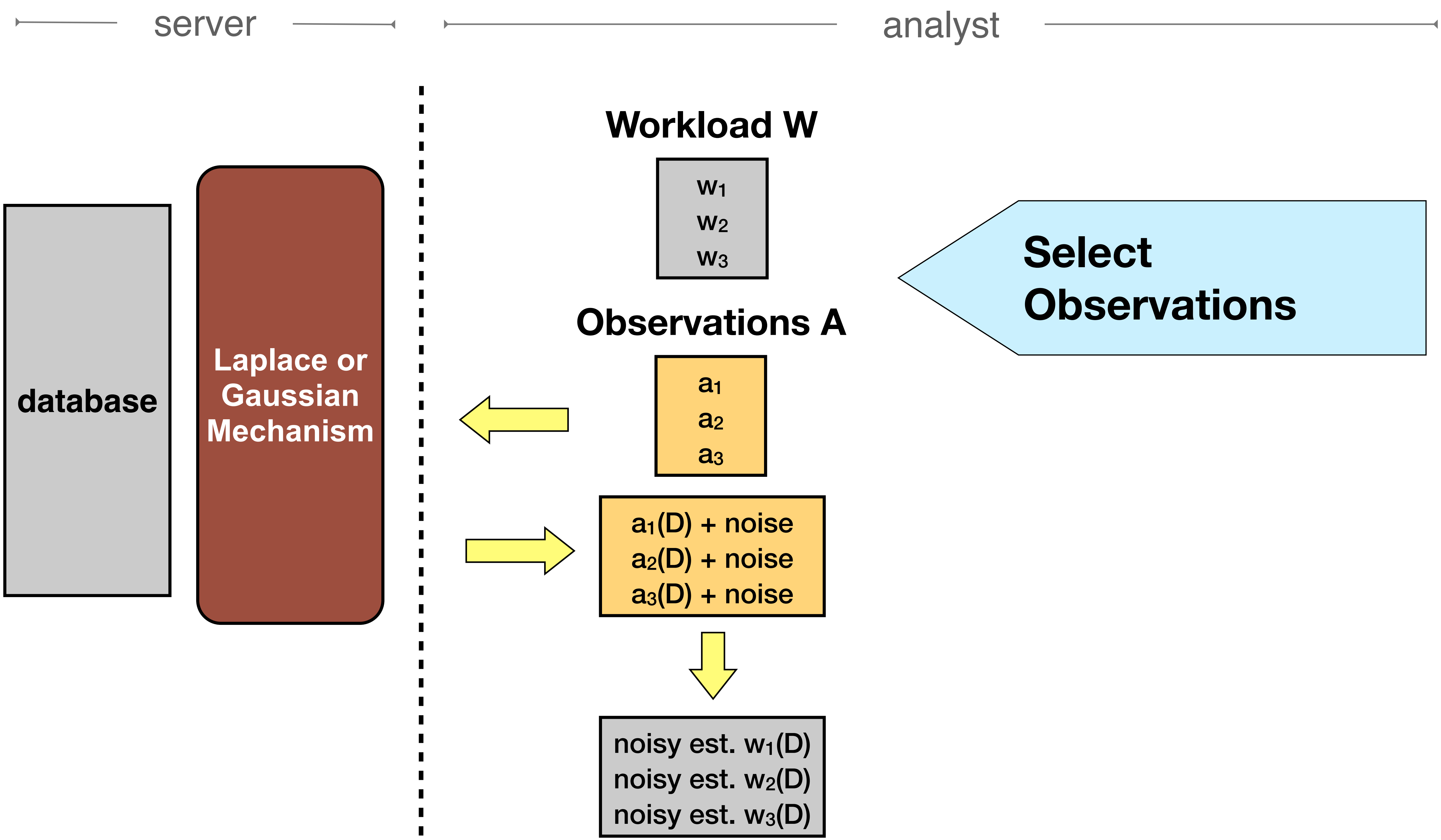
Approach 1: workload-aware mechanisms



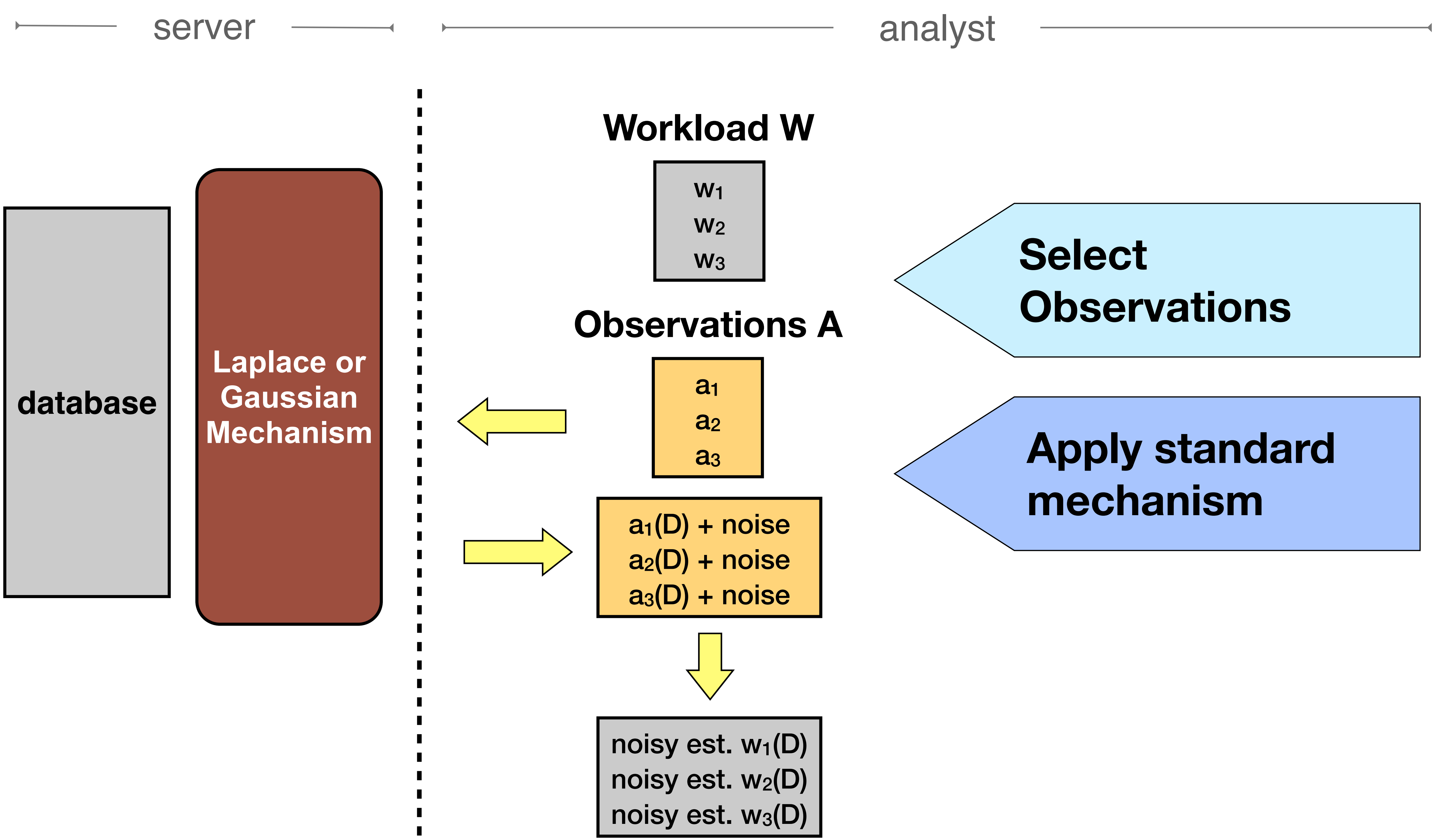
Approach 1: workload-aware mechanisms



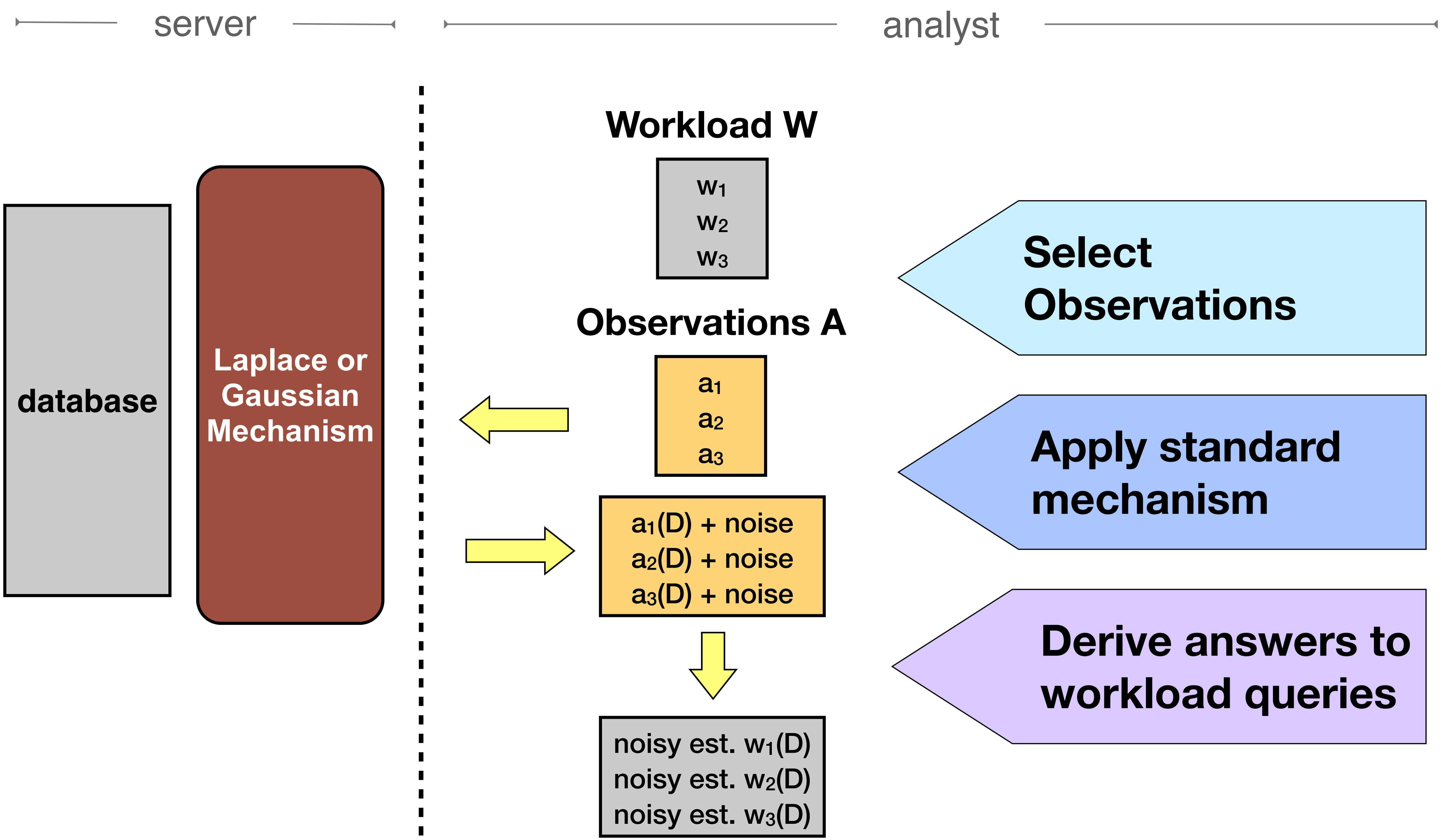
Approach 1: workload-aware mechanisms



Approach 1: workload-aware mechanisms



Approach 1: workload-aware mechanisms

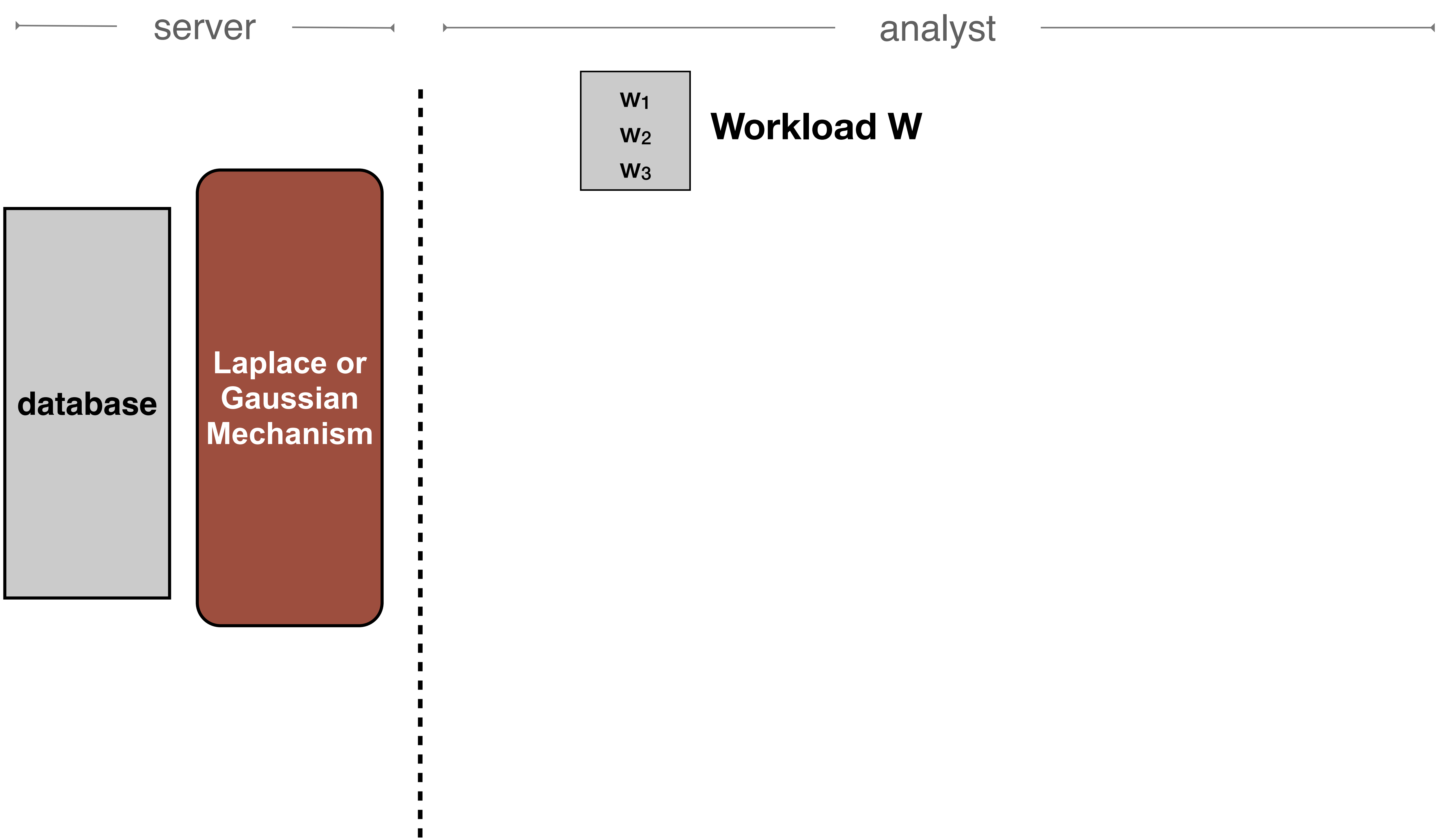


Workload-aware mechanisms

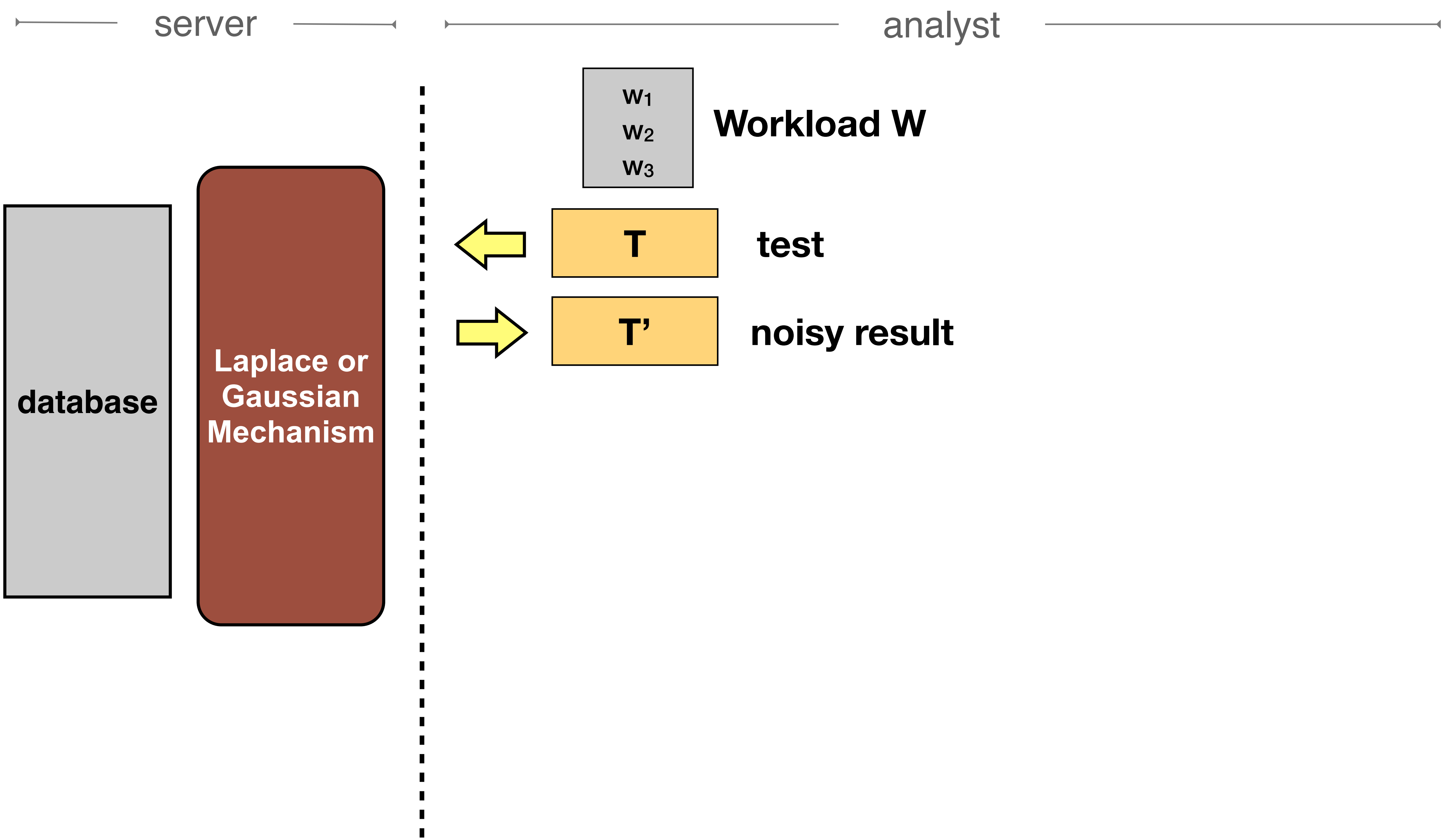
- Observations selected to match (only) the workload.

Workload	Observations		Citation
low-order marginals	Fixed	Fourier basis queries	[Barak, PODS ‘07]
all one-dim range queries		Hierarchical ranges	[Hay, PVLDB ‘10]
all (multi-dim) range queries		Haar wavelet queries	[Xiao, ICDE ‘10]
2-dim range queries		Quad-tree queries	[Cormode, ICDE ’12]
sets of data cubes	Optimized	sets of data cubes	[Ding, SIGMOD ’11]
set of linear queries		set of linear queries	[Li, PODS ‘10] [Li, PVLDB ‘12]
set of linear queries		set of linear queries	[Yuan, VLDB ’12]

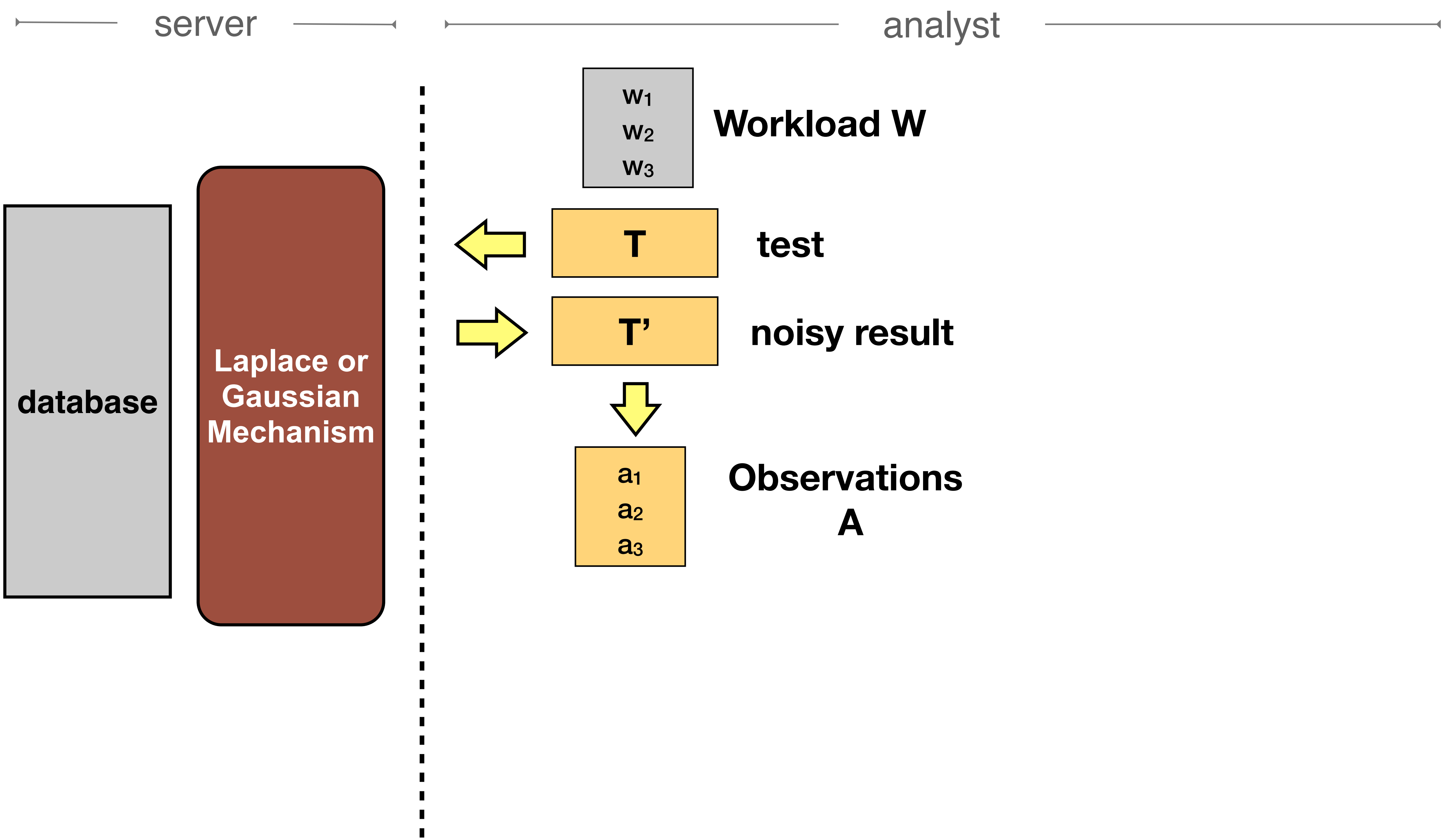
Approach 2: data-aware mechanisms



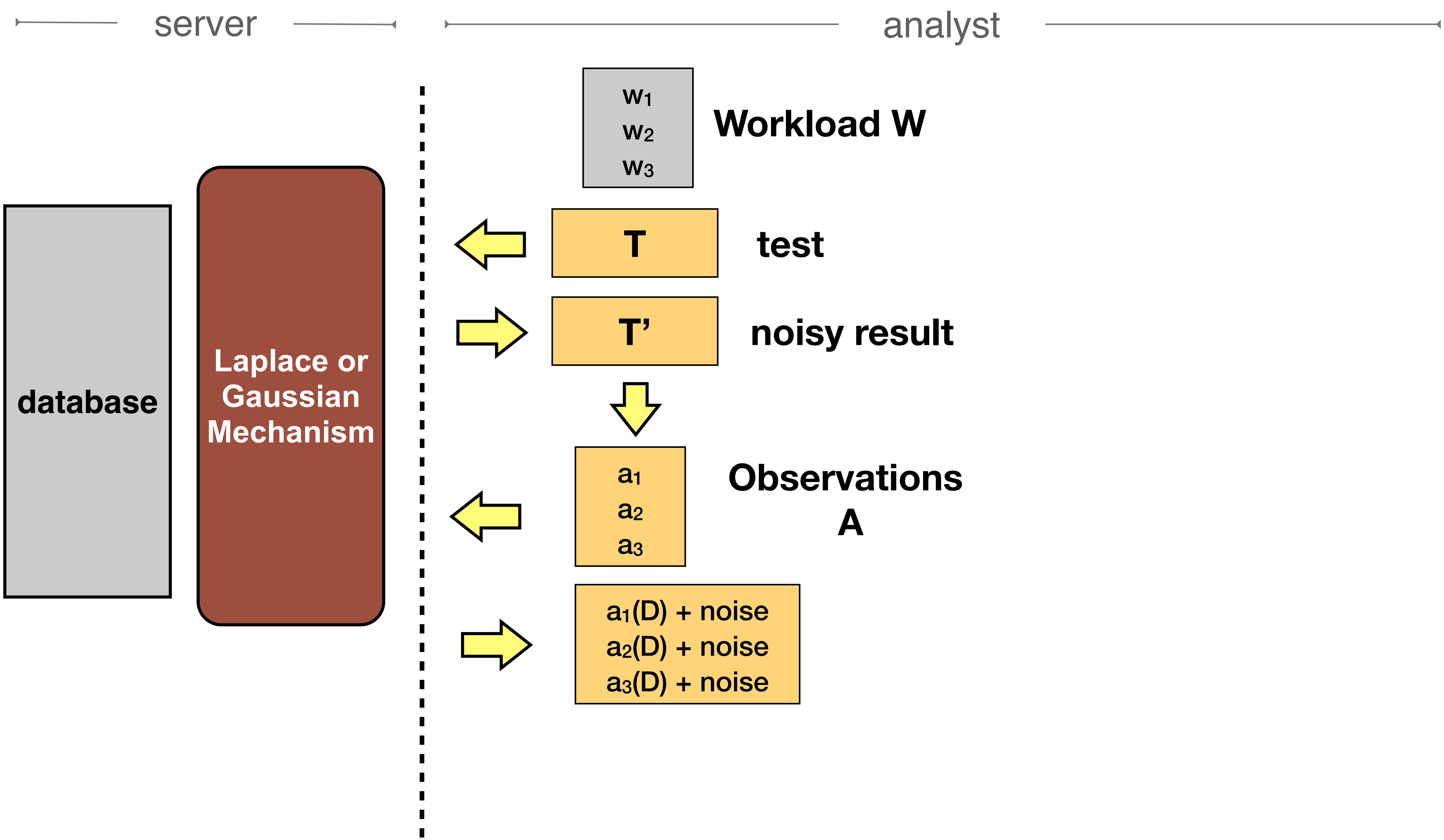
Approach 2: data-aware mechanisms



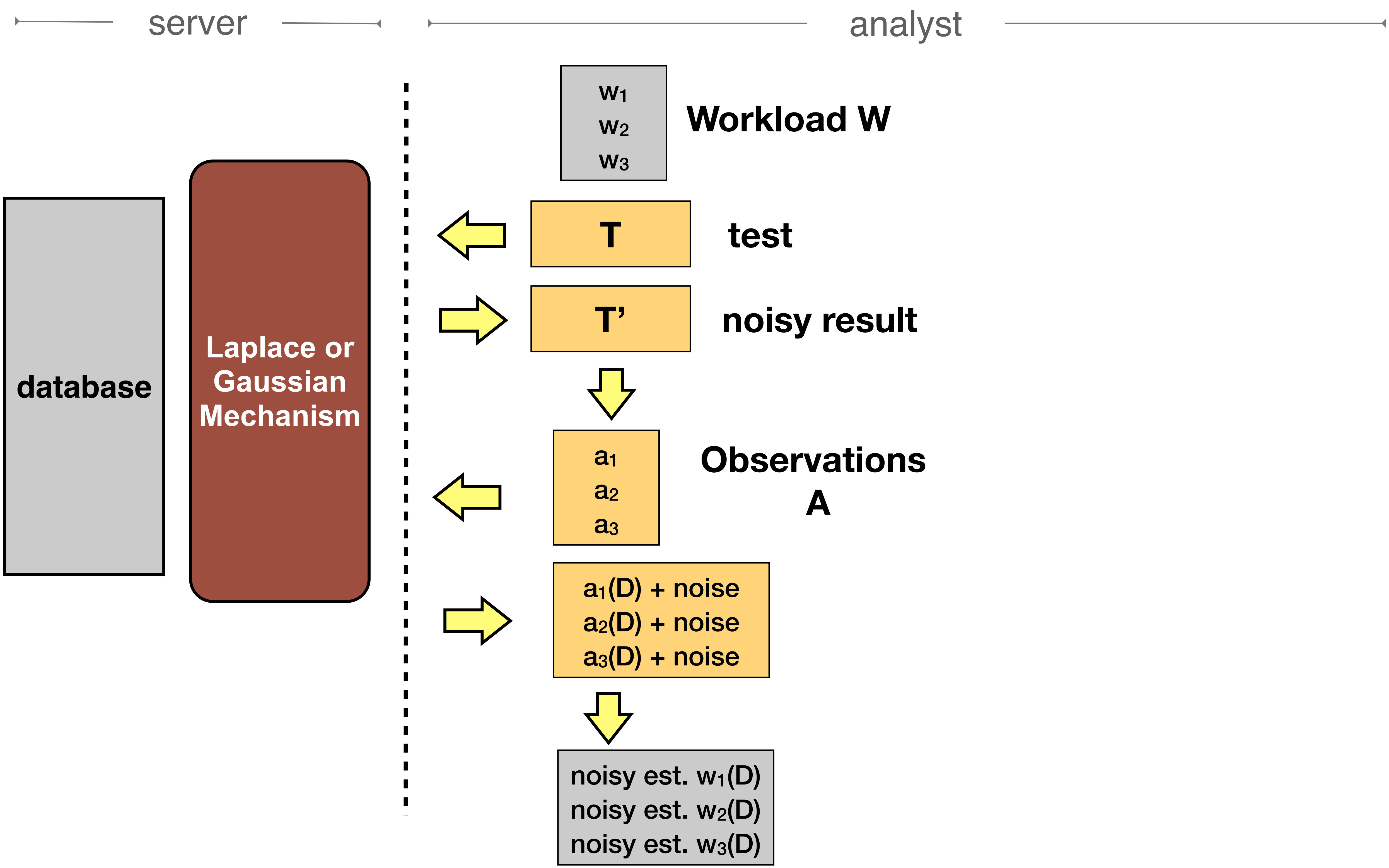
Approach 2: data-aware mechanisms



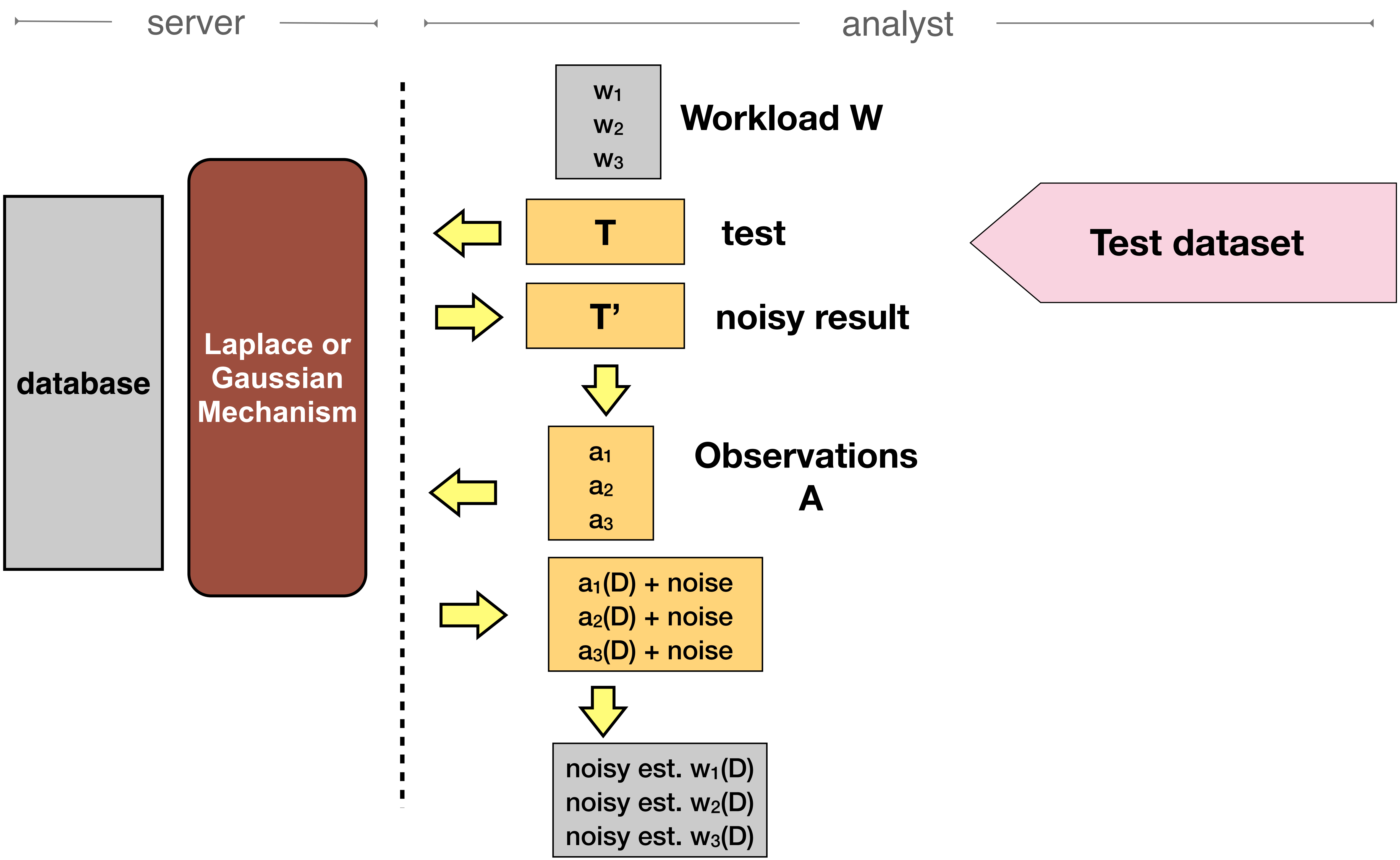
Approach 2: data-aware mechanisms



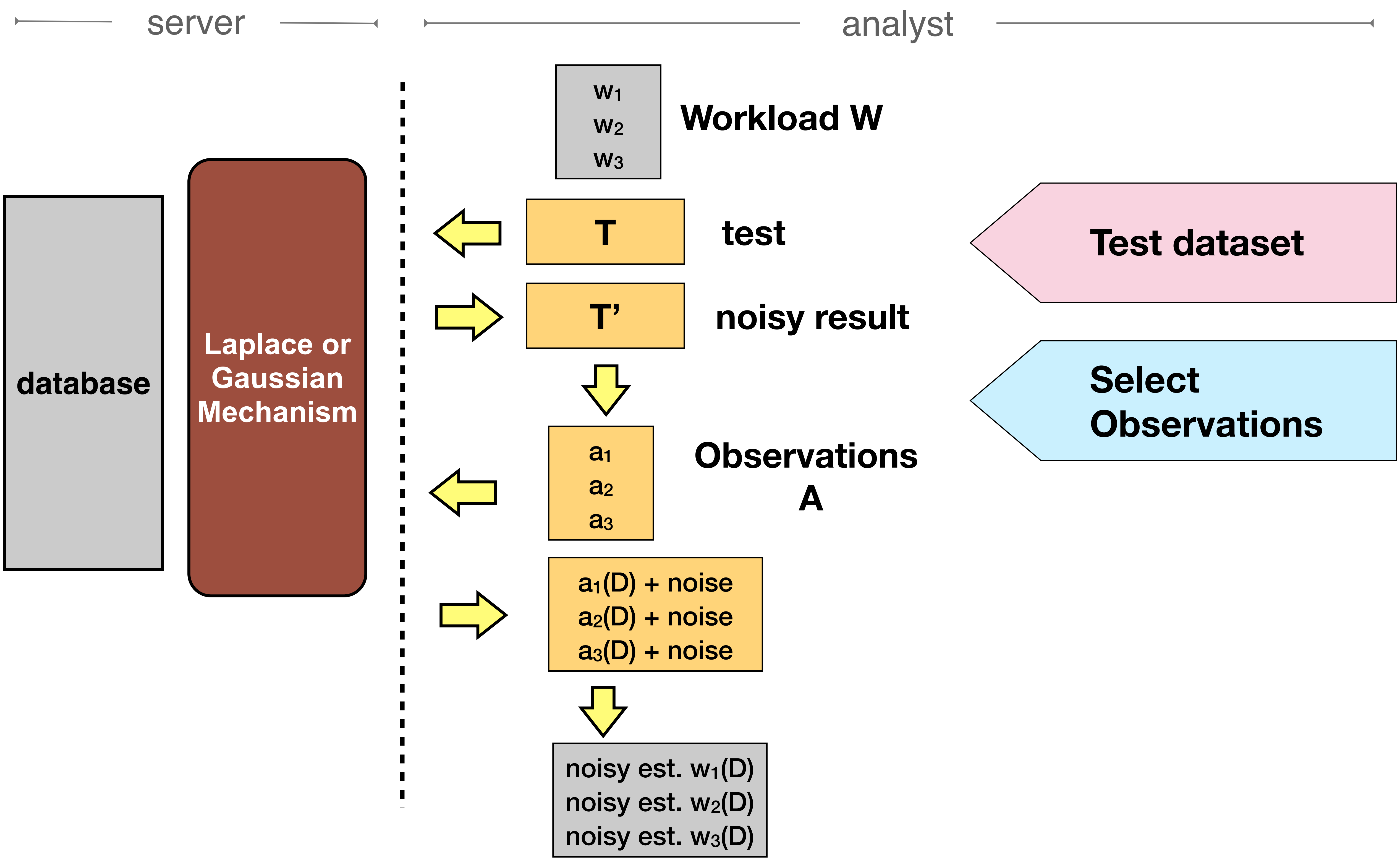
Approach 2: data-aware mechanisms



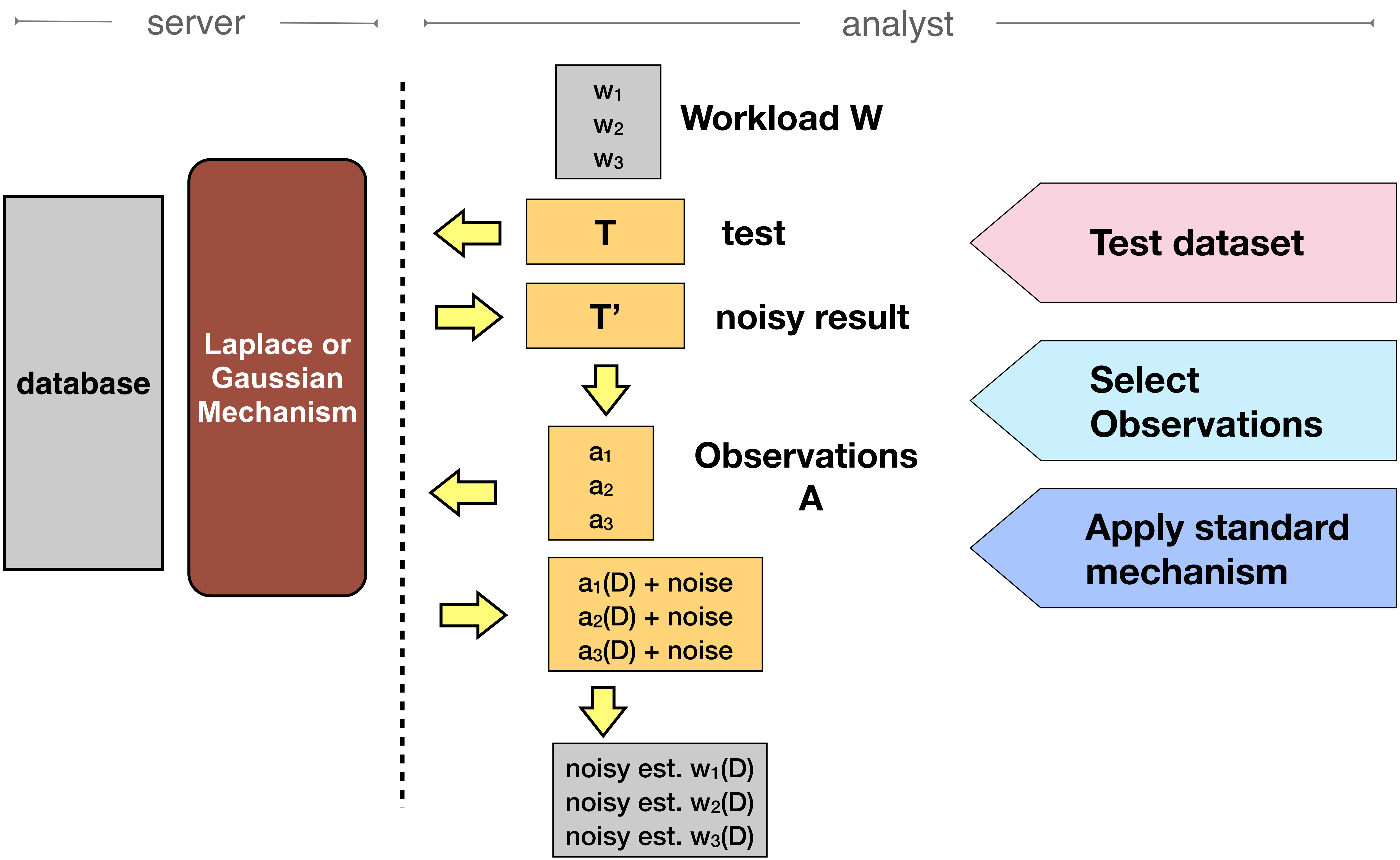
Approach 2: data-aware mechanisms



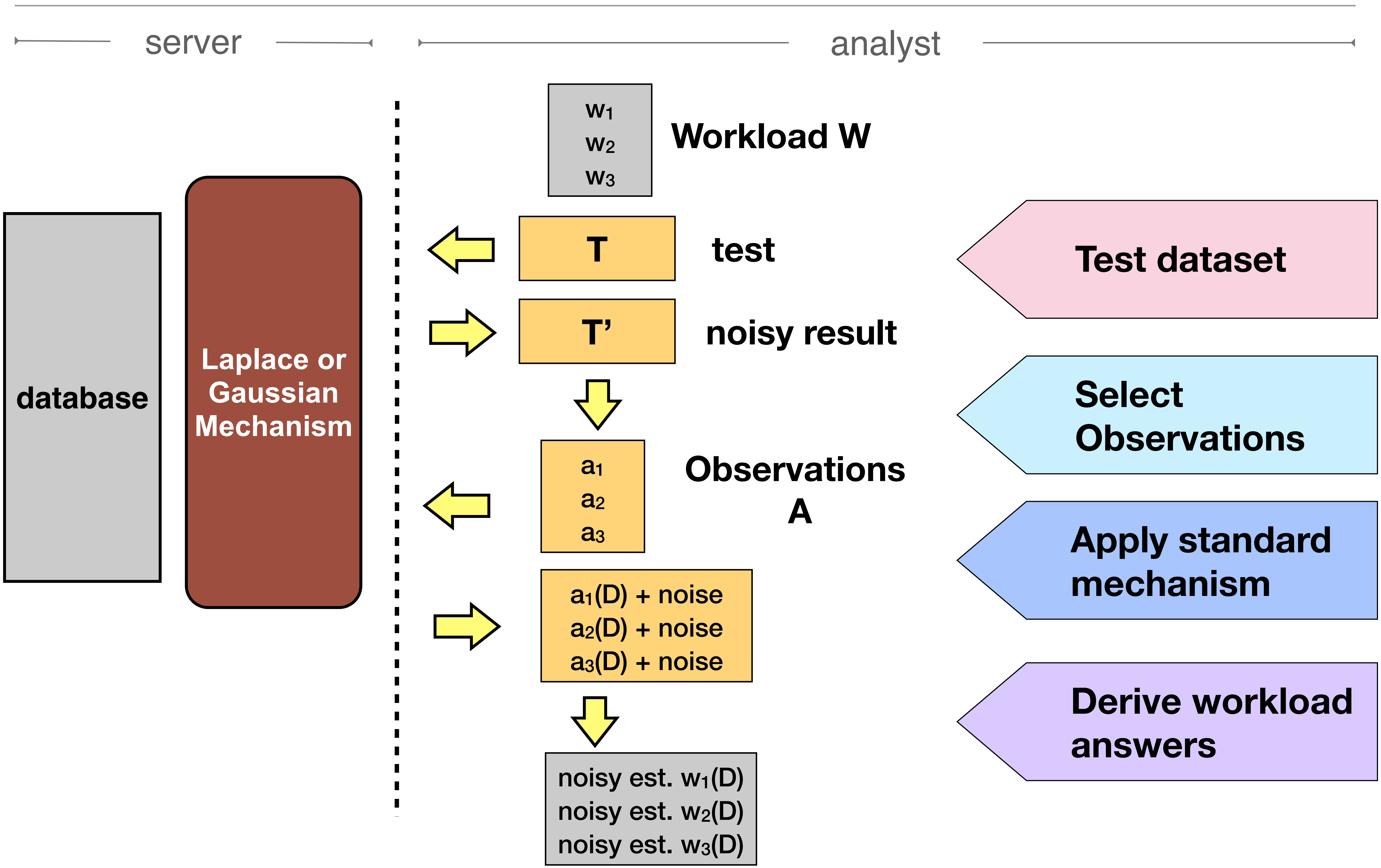
Approach 2: data-aware mechanisms



Approach 2: data-aware mechanisms



Approach 2: data-aware mechanisms



Data-aware mechanisms

- Observations selected to match properties of the database.

Workload	Observations	Citation
1D range queries	approx. v-optimal histogram	[Xu, ICDE ’12]
2D range queries	kd-tree queries	[Xiao, SDM ‘10]
2D range queries	hybrid kd-tree queries	[Cormode, ICDE ’12]
Marginals	scaled workload queries	[Xiao, SIGMOD ’11]
Linear queries	subset of workload	[Hardt, NIPS ’12]

Outline

1. Preliminaries
2. Approach 1: workload-aware
   - Fixed Observations
   - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

Frequency representation of the database

name	gender	grade
Alice	Female	91
Bob	Male	84
Carl	Male	82
Dave	Male	97
Edwina	Female	88
Faith	Female	78
Ghita	Female	85
...	...	...

Relational database

{gender, grade}

gender	grade	count
Male	100	10
Male	99	13
Male	98	5
Male	97	7
...	...	...
Female	100	15
Female	99	21
Female	98	4
Female	97	14
Female	96	9

Frequency vector

x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
...
x_n

$\mathbf{x}$

Frequency representation of the database

name	gender	grade
Alice	Female	91
Bob	Male	84
Carl	Male	82
Dave	Male	97
Edwina	Female	88
Faith	Female	78
Ghita	Female	85
...	...	...

Relational database

Frequency vector $\mathbf{x}$

Frequency representation of the database

name	gender	grade
Alice	Female	91
Bob	Male	84
Carl	Male	82
Dave	Male	97
Edwina	Female	88
Faith	Female	78
Ghita	Female	85
...	...	...

Relational database

{grade}

grade	count
90-100	10
80-90	23
70-80	16
60-70	3

Frequency vector

x_1
x_2
x_3
x_4

$\mathbf{x}$

Linear counting queries

A **linear counting query** $\mathbf{w}$ computes a linear combination of the frequency vector counts:

$$\mathbf{w(D)} = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n \qquad \text{each } w_i \in R$$

Linear counting queries

A **linear counting query** $\mathbf{w}$ computes a linear combination of the frequency vector counts:

$$\mathbf{w(D)} = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n$$

each $w_i \in R$

... as a length n row vector:

$$\mathbf{w} = [w_1, w_2, w_3 \ldots w_n]$$

The query result is:

$$\mathbf{wx}$$

Linear counting queries

A **linear counting query** $\mathbf{w}$ computes a linear combination of the frequency vector counts:

$$\mathbf{w(D)} = w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

each $w_i \in R$

... as a length n row vector:

$$\mathbf{w} = [w_1, w_2, w_3 \ldots w_n]$$

The query result is:

$$\mathbf{wx}$$

a set of linear counting queries is a matrix:

$$\mathbf{W} \quad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

The query result is:

$$\mathbf{Wx}$$

Queries and workloads

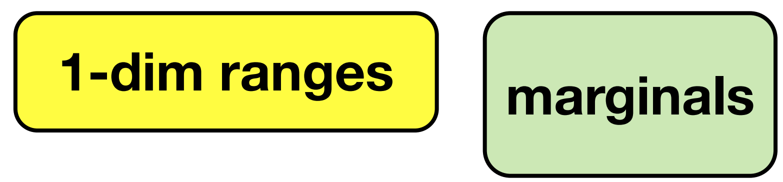
- **1-dimensional range queries**: intervals
- **Marginals / data cube queries / contingency tables**: aggregate over excluded dimensions.
- **k-dimensional range queries**: axis-aligned rectangles
- **Predicate counting queries**: only 0 or 1 coefficients
- **Linear counting queries**: arbitrary coefficients

Queries and workloads

1-dim ranges

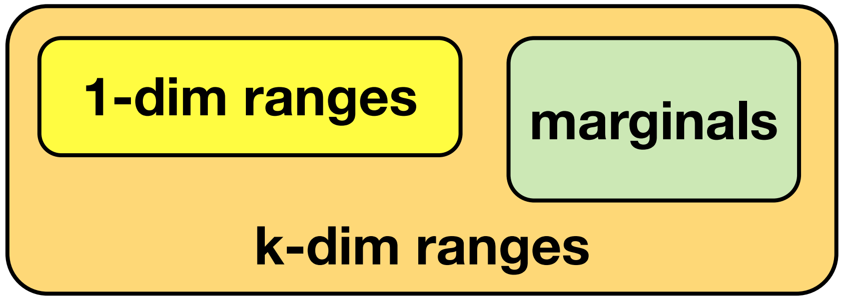
- **1-dimensional range queries**: intervals
- **Marginals / data cube queries / contingency tables**: aggregate over excluded dimensions.
- **k-dimensional range queries**: axis-aligned rectangles
- **Predicate counting queries**: only 0 or 1 coefficients
- **Linear counting queries**: arbitrary coefficients

Queries and workloads



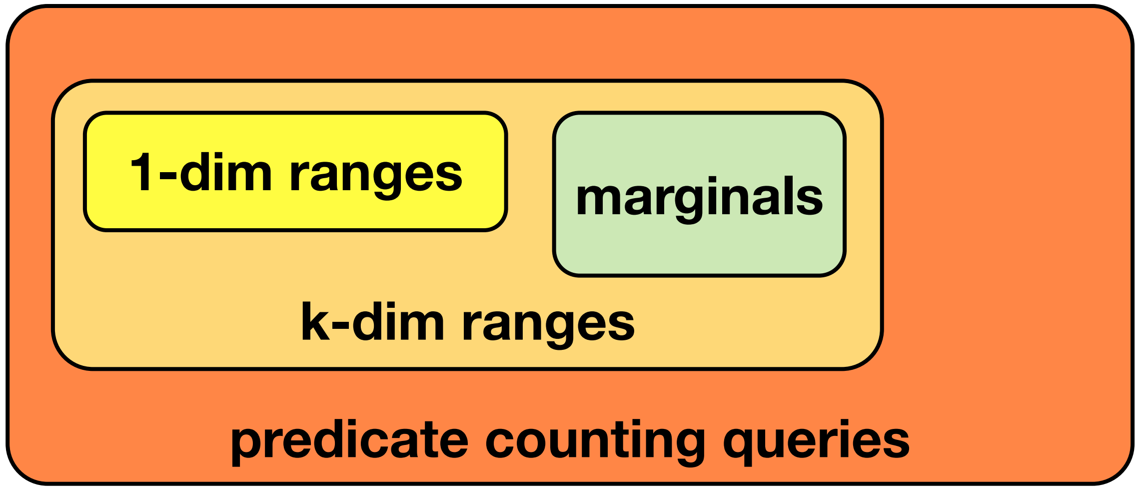
- **1-dimensional range queries**: intervals
- **Marginals / data cube queries / contingency tables**: aggregate over excluded dimensions.
- **k-dimensional range queries**: axis-aligned rectangles
- **Predicate counting queries**: only 0 or 1 coefficients
- **Linear counting queries**: arbitrary coefficients

Queries and workloads



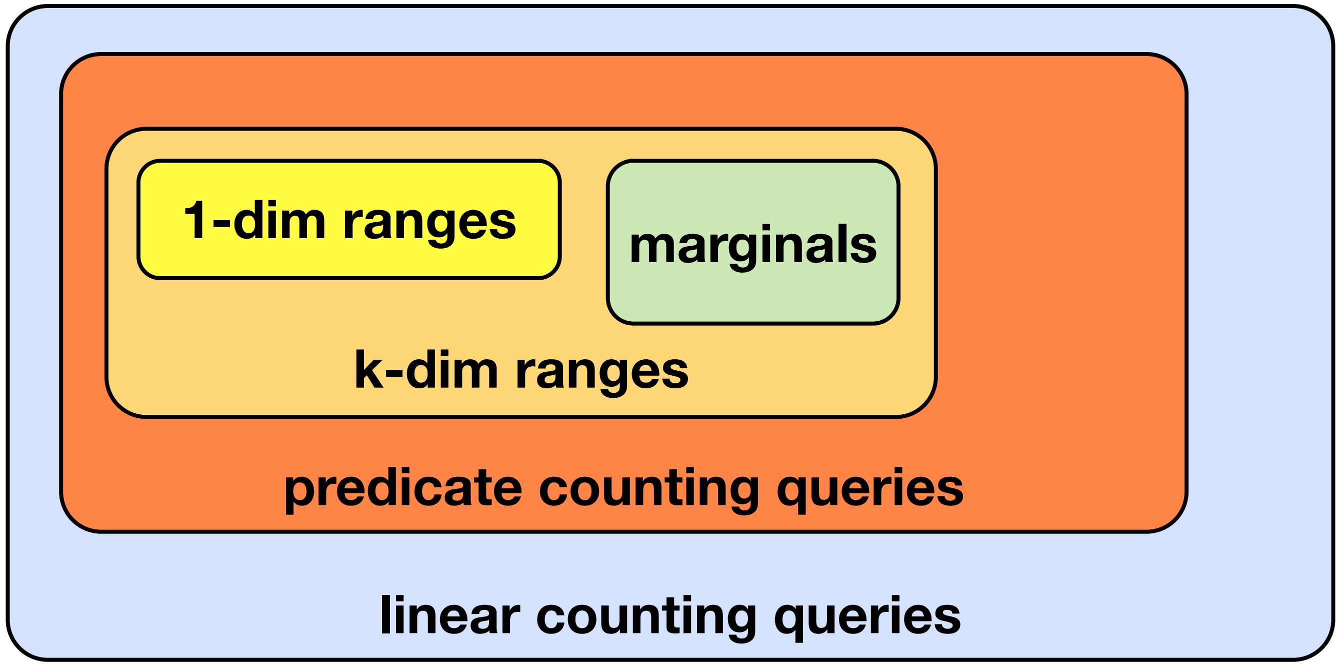
- **1-dimensional range queries**: intervals
- **Marginals / data cube queries / contingency tables**: aggregate over excluded dimensions.
- **k-dimensional range queries**: axis-aligned rectangles
- **Predicate counting queries**: only 0 or 1 coefficients
- **Linear counting queries**: arbitrary coefficients

Queries and workloads



- **1-dimensional range queries**: intervals
- **Marginals / data cube queries / contingency tables**: aggregate over excluded dimensions.
- **k-dimensional range queries**: axis-aligned rectangles
- **Predicate counting queries**: only 0 or 1 coefficients
- **Linear counting queries**: arbitrary coefficients

Queries and workloads



- **1-dimensional range queries**: intervals
- **Marginals / data cube queries / contingency tables**: aggregate over excluded dimensions.
- **k-dimensional range queries**: axis-aligned rectangles
- **Predicate counting queries**: only 0 or 1 coefficients
- **Linear counting queries**: arbitrary coefficients

Privacy definitions & mechanisms

- Differential privacy

A randomized algorithm A provides (ε,δ)-**differential privacy** if:
for all neighboring databases D and D', and
for any set of outputs *S*:

$$Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} Pr[\mathcal{A}(D') \in S] + \delta$$

- if δ=0, standard ε-differential privacy:
  - **Laplace(0,b) noise where $b=||q||_1/\varepsilon$**
- if δ>0, approximate (ε,δ)-differential privacy:
  - **Gaussian(0,σ) noise where $\sigma= ||q||_2 \, (2\ln(2/\delta))^{1/2}/\varepsilon$**
- Multi-query Laplace/Gaussian mechanism adds independent noise to **each** query answer.
- Exponential mechanism

The sensitivity of a query matrix

- For two neighboring databases D and D', their frequency vectors x and x' will differ in one position, by exactly 1.

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}}_{\text{answers}} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}}_{\text{query matrix } \mathbf{W}} \text{ x } \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \quad +1 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}}_{\mathbf{x'}}$$

The sensitivity of a query matrix

- For two neighboring databases D and D', their frequency vectors x and x' will differ in one position, by exactly 1.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \quad +1 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}$$

answers — query matrix $\mathbf{W}$ — $\mathbf{x'}$

The sensitivity of a query matrix

- For two neighboring databases D and D', their frequency vectors x and x' will differ in one position, by exactly 1.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & \boxed{1} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \boxed{1} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \boxed{1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & \boxed{1} & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \quad +1 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}$$

answers

query matrix $\mathbf{W}$

$\mathbf{x'}$

$\boxed{||\mathbf{W}||_1 = 4}$

The sensitivity of a query matrix

- For two neighboring databases D and D', their frequency vectors x and x' will differ in one position, by exactly 1.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & \boxed{1} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \quad +1 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}$$

answers query matrix $\mathbf{W}$ $\mathbf{x'}$

$$||\mathbf{W}||_1 = 4$$

The $\mathbf{L_1}$ **sensitivity** of a query matrix is:
the maximum L1 norm of the columns.

The sensitivity of a query matrix

- For two neighboring databases D and D', their frequency vectors x and x' will differ in one position, by exactly 1.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 1 & \boxed{1} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \quad +1 \\ x_8 \\ x_9 \\ x_{10} \end{bmatrix}$$

answers   query matrix $\mathbf{W}$ $\mathbf{x'}$

$$||\mathbf{W}||_1 = 4$$

The $\mathbf{L_1}$ **sensitivity** of a query matrix is:
the maximum L1 norm of the columns.

The $\mathbf{L_2}$ **sensitivity** of a query matrix is:
the maximum L2 norm of the columns.

Outline

1. Preliminaries
2. Approach 1: workload-aware
   - Fixed Observations
   - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

Outline

1. Preliminaries
2. Approach 1: workload-aware
    - Fixed Observations
    - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

Answering all range queries

Goal: answer all **range-count queries** over $\mathbf{x}$

$$\text{AllRange} = \{\, \mathbf{w} \mid \mathbf{w} = \mathbf{x_i} + \ldots + \mathbf{x_j} \quad \text{for } 1 \leq i \leq j \leq n \,\}$$

workload W

w_1	range(x_1,x_4)	x_1	+	x_2	+	x_3	+	x_4
w_2	range(x_1,x_3)	x_1	+	x_2	+	x_3		
w_3	range(x_2,x_4)			x_2	+	x_3	+	x_4
w_4	range(x_1,x_2)	x_1	+	x_2				
w_5	range(x_2,x_3)			x_2	+	x_3		
w_6	range(x_3,x_4)					x_3	+	x_4
w_7	range(x_1,x_1)	x_1						
w_8	range(x_2,x_2)			x_2				
w_9	range(x_3,x_3)					x_3		
w_{10}	range(x_4,x_4)							x_4

Answering all range queries

Goal: answer all range-count queries over x

$$\text{AllRange} = \{\ \mathbf{w} \mid \mathbf{w} = \mathbf{x_i} + \ldots + \mathbf{x_j} \quad \text{for } 1 \le i \le j \le n\ \}$$

workload W

w_1	range(x_1,x_4)	x_1	+	x_2	+	x_3	+	x_4
w_2	range(x_1,x_3)	x_1	+	x_2	+	x_3		
w_3	range(x_2,x_4)			x_2	+	x_3	+	x_4
w_4	range(x_1,x_2)	x_1	+	x_2				
w_5	range(x_2,x_3)			x_2	+	x_3		
w_6	range(x_3,x_4)					x_3	+	x_4
w_7	range(x_1,x_1)	x_1						
w_8	range(x_2,x_2)			x_2				
w_9	range(x_3,x_3)					x_3		
w_{10}	range(x_4,x_4)							x_4

x=

10		23		16		3

Answering all range queries

Goal: answer all **range-count queries** over **x**

$$\text{AllRange} = \{\, \mathbf{w} \mid \mathbf{w} = \mathbf{x_i} + \ldots + \mathbf{x_j} \quad \text{for } 1 \le i \le j \le n \,\}$$

workload W

w_1	range(x_1,x_4)	x_1	+	x_2	+	x_3	+	x_4	w_1	52
w_2	range(x_1,x_3)	x_1	+	x_2	+	x_3			w_2	49
w_3	range(x_2,x_4)			x_2	+	x_3	+	x_4	w_3	42
w_4	range(x_1,x_2)	x_1	+	x_2					w_4	33
w_5	range(x_2,x_3)			x_2	+	x_3			w_5	39
w_6	range(x_3,x_4)					x_3	+	x_4	w_6	19
w_7	range(x_1,x_1)	x_1							w_7	10
w_8	range(x_2,x_2)			x_2					w_8	23
w_9	range(x_3,x_3)					x_3			w_9	16
w_{10}	range(x_4,x_4)							x_4	w_{10}	3

x=

10		23		16		3

Method 1: basic Laplace mechanism

Workload queries

w_1	x_1	+	x_2	+	x_3	+	x_4
w_2	x_1	+	x_2	+	x_3		
w_3			x_2	+	x_3	+	x_4
w_4	x_1	+	x_2				
w_5			x_2	+	x_3		
w_6					x_3	+	x_4
w_7	x_1						
w_8			x_2				
w_9					x_3		
w_{10}							x_4

$\mathbf{W}$

$||\mathbf{W}||_1 = 6$

Laplace noise

$$+ \ (6/\varepsilon) \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \\ b_{10} \end{bmatrix} \Rightarrow$$

private output

$$\begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \\ w'_5 \\ w'_6 \\ w'_7 \\ w'_8 \\ w'_9 \\ w'_{10} \end{bmatrix}$$

Method 1: basic Laplace mechanism

Workload queries | **Laplace noise** | **private output**

W

w_1	x_1	+	x_2	+	x_3	+	x_4	**52**
w_2	x_1	+	x_2	+	x_3			**49**
w_3			x_2	+	x_3	+	x_4	**42**
w_4	x_1	+	x_2					**33**
w_5			x_2	+	x_3			**39**
w_6					x_3	+	x_4	**19**
w_7	x_1							**10**
w_8			x_2					**23**
w_9					x_3			**16**
w_{10}							x_4	**3**

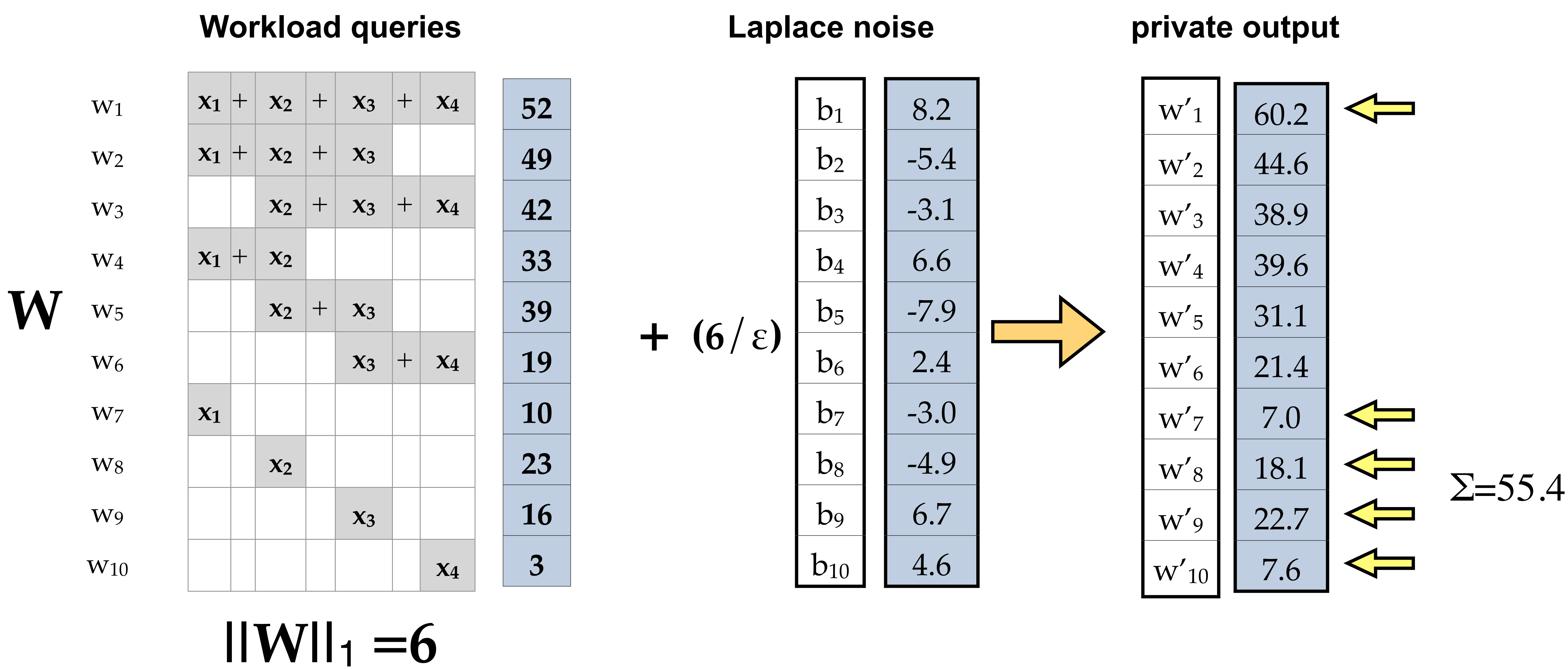
$$\|\mathbf{W}\|_1 = 6$$

+ $(6/\varepsilon)$

b_1	8.2
b_2	-5.4
b_3	-3.1
b_4	6.6
b_5	-7.9
b_6	2.4
b_7	-3.0
b_8	-4.9
b_9	6.7
b_{10}	4.6

w'_1	60.2
w'_2	44.6
w'_3	38.9
w'_4	39.6
w'_5	31.1
w'_6	21.4
w'_7	7.0
w'_8	18.1
w'_9	22.7
w'_{10}	7.6

Method 1: basic Laplace mechanism



Method 1: basic Laplace mechanism

W

	Workload queries			Laplace noise		private output	
w_1	$x_1 + x_2 + x_3 + x_4$	**52**		b_1	8.2	w'_1	60.2
w_2	$x_1 + x_2 + x_3$	**49**		b_2	-5.4	w'_2	44.6
w_3	$x_2 + x_3 + x_4$	**42**		b_3	-3.1	w'_3	38.9
w_4	$x_1 + x_2$	**33**		b_4	6.6	w'_4	39.6
w_5	$x_2 + x_3$	**39**	**+ (6 / ε)**	b_5	-7.9	w'_5	31.1
w_6	$x_3 + x_4$	**19**		b_6	2.4	w'_6	21.4
w_7	x_1	**10**		b_7	-3.0	w'_7	7.0
w_8	x_2	**23**		b_8	-4.9	w'_8	18.1
w_9	x_3	**16**		b_9	6.7	w'_9	22.7
w_{10}	x_4	**3**		b_{10}	4.6	w'_{10}	7.6

$\Sigma = 55.4$

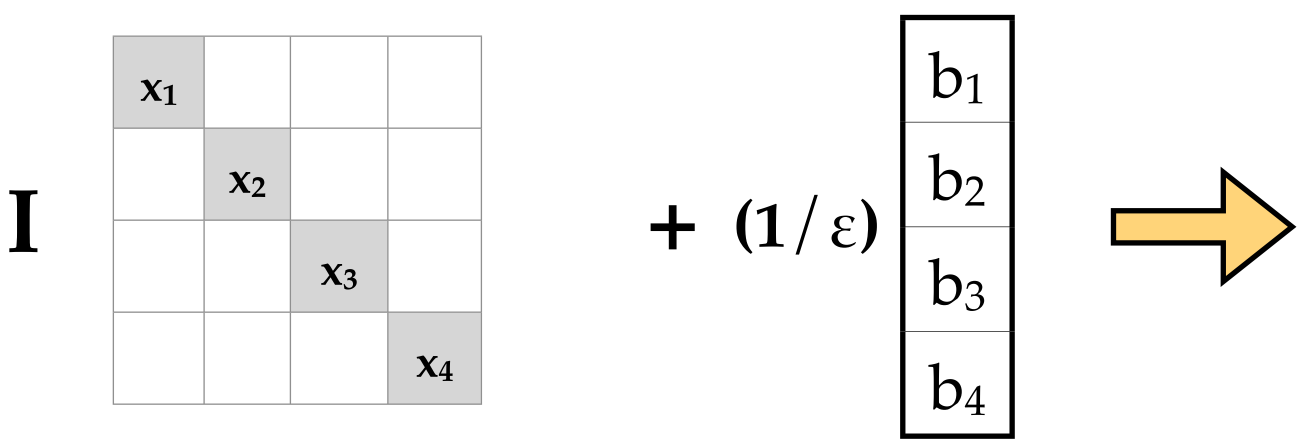
$\|W\|_1 = 6$

	n=4	n
Sensitivity $\|W\|_1$	6	$O(n^2)$
Error per query	$2(\|W\|_1/\varepsilon)^2 = 72/\varepsilon^2$	$2(\|W\|_1/\varepsilon)^2 = O(n^4)/\varepsilon^2$

Method 2: noisy frequency counts

Use Laplace mechanism to get noisy estimates for each x_i.

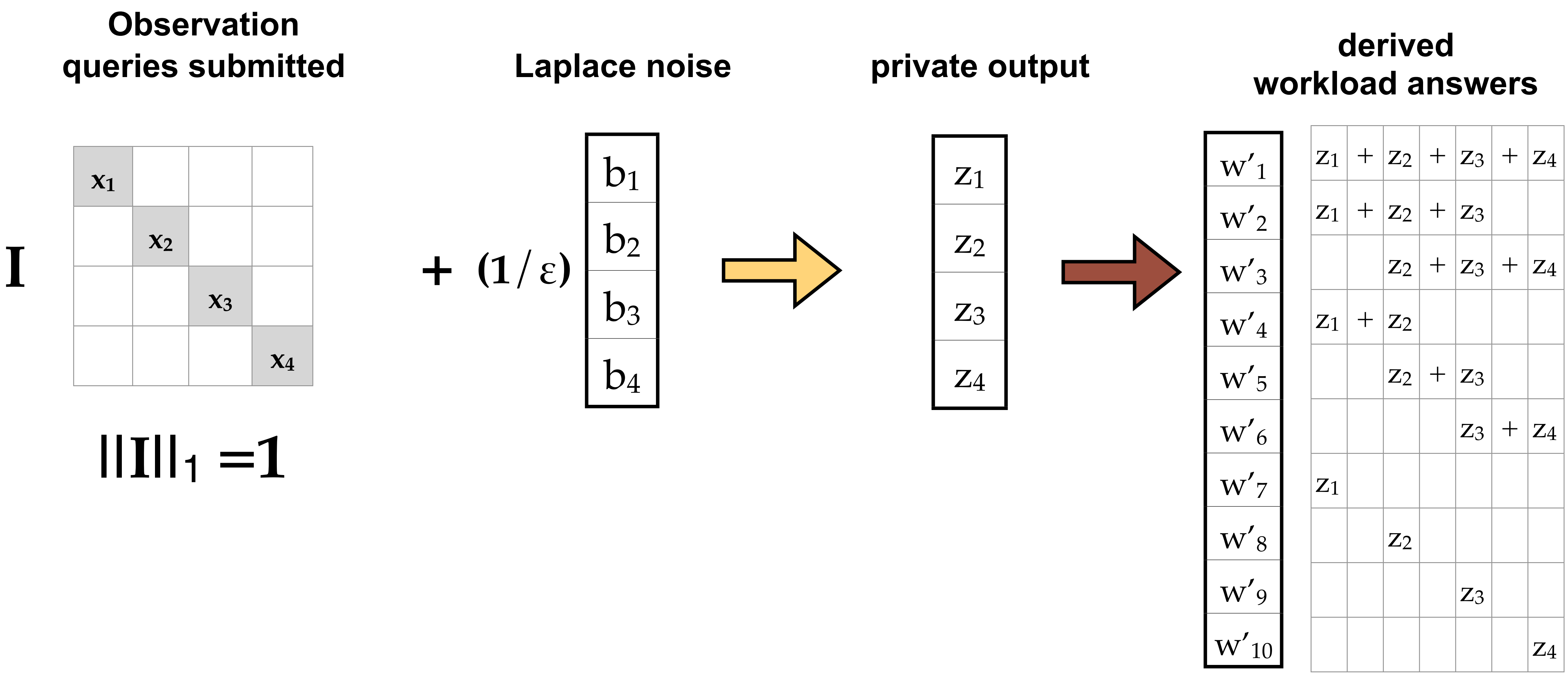
Observation queries submitted | **Laplace noise** | **private output**



$$||\mathbf{I}||_1 = 1$$

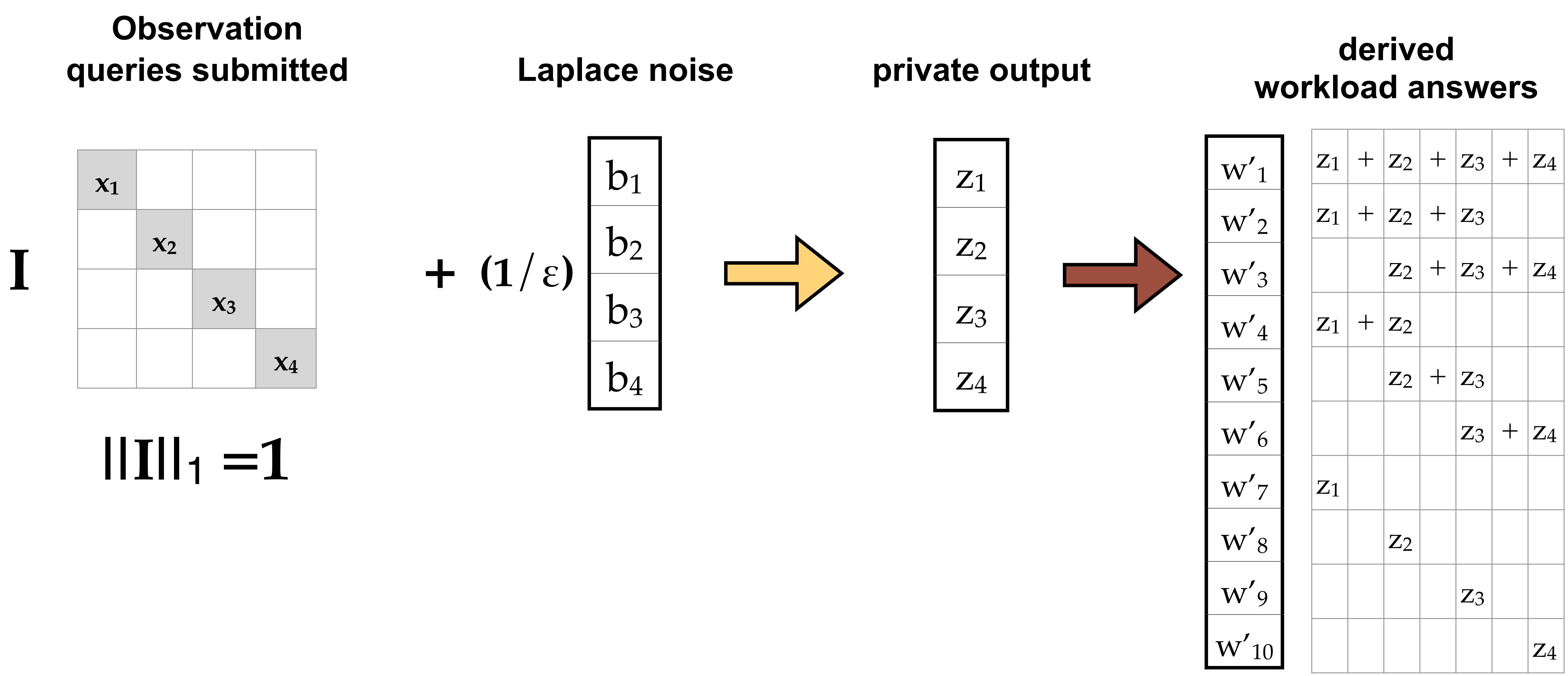
Method 2: noisy frequency counts

Use Laplace mechanism to get noisy estimates for each $\mathbf{x_i}$.



Method 2: noisy frequency counts

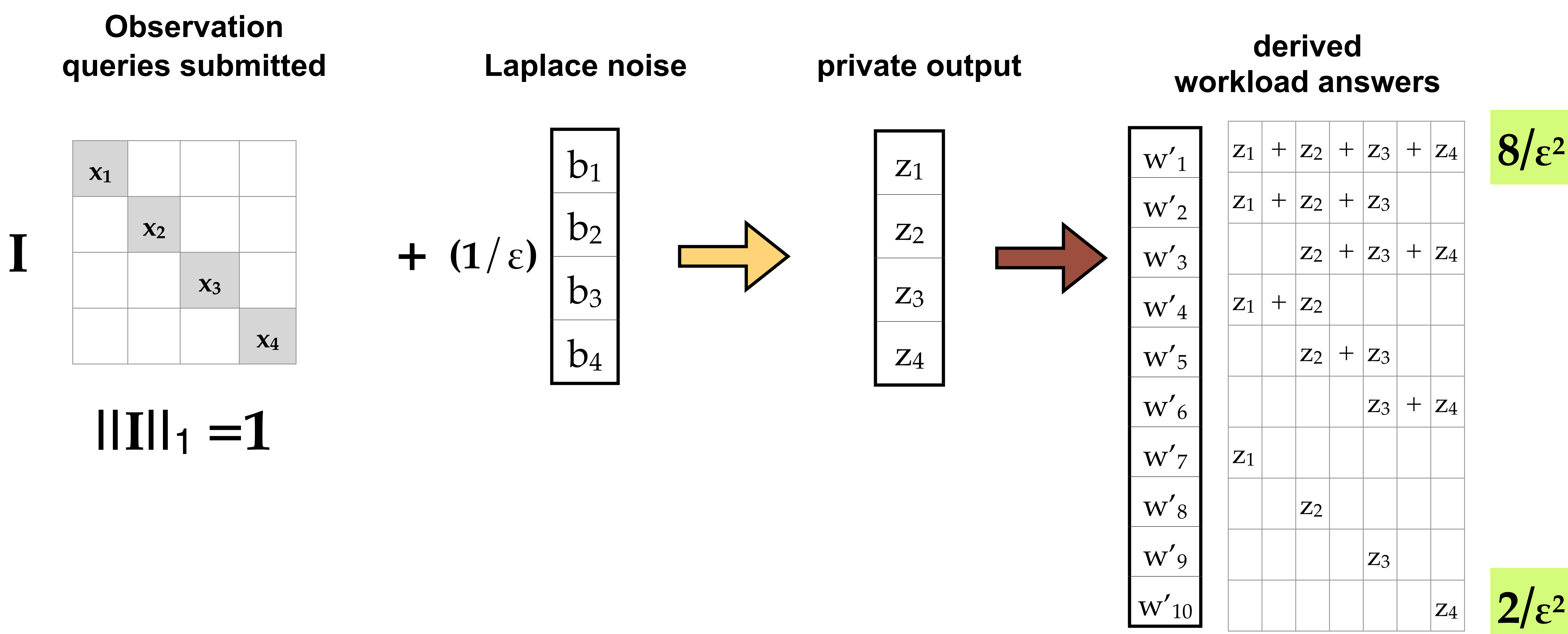
Use Laplace mechanism to get noisy estimates for each $\mathbf{x_i}$.



For $\mathbf{w=range(x_i,x_j)}$ $\mathrm{Error(w)} = 2(j\text{-}i+1)/\varepsilon^2$

Method 2: noisy frequency counts

Use Laplace mechanism to get noisy estimates for each x_i.



For **w=range(x_i,x_j)** Error(w)= $2(j-i+1) / \varepsilon^2$

Method 3: hierarchical observations

Hierarchical queries: recursively partition the domain, computing sums of each interval.

Observation queries submitted | **Laplace noise** | **private output**

H

x_1	+	x_2	+	x_3	+	x_4
x_1	+	x_2				
				x_3	+	x_4
x_1						
		x_2				
				x_3		
						x_4

$$+\ (3/\varepsilon)\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \Rightarrow \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix}$$

$$||\mathbf{H}||_1 = 3$$

$$= \log n + 1$$

Method 3: hierarchical observations

Hierarchical queries: recursively partition the domain, computing sums of each interval.

Observation queries submitted | **Laplace noise** | **private output** | **derived workload answers**

$$\mathbf{H}\quad \begin{bmatrix} x_1+x_2+x_3+x_4 \\ x_1+x_2 \\ x_3+x_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + (3/\varepsilon) \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \Rightarrow \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} \Rightarrow \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \\ w'_5 \\ w'_6 \\ w'_7 \\ w'_8 \\ w'_9 \\ w'_{10} \end{bmatrix} \quad ?$$

$$\|\mathbf{H}\|_1 = 3$$
$$= \log n + 1$$

Method 3: hierarchical observations

Hierarchical queries: recursively partition the domain, computing sums of each interval.

Observation queries submitted | **Laplace noise** | **private output** | **derived workload answers**

$$\mathbf{H}\quad \begin{array}{|c|c|c|c|c|c|c|} \hline x_1 & + & x_2 & + & x_3 & + & x_4 \\ \hline x_1 & + & x_2 & & & & \\ \hline & & & & x_3 & + & x_4 \\ \hline x_1 & & & & & & \\ \hline & & x_2 & & & & \\ \hline & & & & x_3 & & \\ \hline & & & & & & x_4 \\ \hline \end{array} \;+\; (3/\varepsilon) \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \qquad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} \qquad \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \\ w'_5 \\ w'_6 \\ w'_7 \\ w'_8 \\ w'_9 \\ w'_{10} \end{bmatrix} \;?$$

$$\|\mathbf{H}\|_1 = 3$$
$$= \log n + 1$$

Possible estimates for query $\mathrm{range}(x_2, x_3) = x_2 + x_3$

[Hay, PVLDB 10]

Method 3: hierarchical observations

Hierarchical queries: recursively partition the domain, computing sums of each interval.

Observation queries submitted | **Laplace noise** | **private output** | **derived workload answers**

H

x_1	+	x_2	+	x_3	+	x_4
x_1	+	x_2				
				x_3	+	x_4
x_1						
		x_2				
				x_3		
						x_4

$$+ \ (3/\varepsilon) \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \Rightarrow \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} \Rightarrow \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \\ w'_5 \\ w'_6 \\ w'_7 \\ w'_8 \\ w'_9 \\ w'_{10} \end{bmatrix} \ ?$$

$$||\mathbf{H}||_1 = 3$$

$$= \log n + 1$$

Possible estimates for query $\mathrm{range}(x_2, x_3) = x_2 + x_3$

$z_5 + z_6$

Method 3: hierarchical observations

Hierarchical queries: recursively partition the domain, computing sums of each interval.

Observation queries submitted | **Laplace noise** | **private output** | **derived workload answers**

$$\mathbf{H}: \begin{bmatrix} x_1 + x_2 + x_3 + x_4 \\ x_1 + x_2 \\ x_3 + x_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + (3/\varepsilon) \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \Rightarrow \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} \Rightarrow \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \\ w'_5 \\ w'_6 \\ w'_7 \\ w'_8 \\ w'_9 \\ w'_{10} \end{bmatrix} \; ?$$

$$||\mathbf{H}||_1 = 3 = \log n + 1$$

Possible estimates for query $\text{range}(x_2, x_3) = x_2 + x_3$

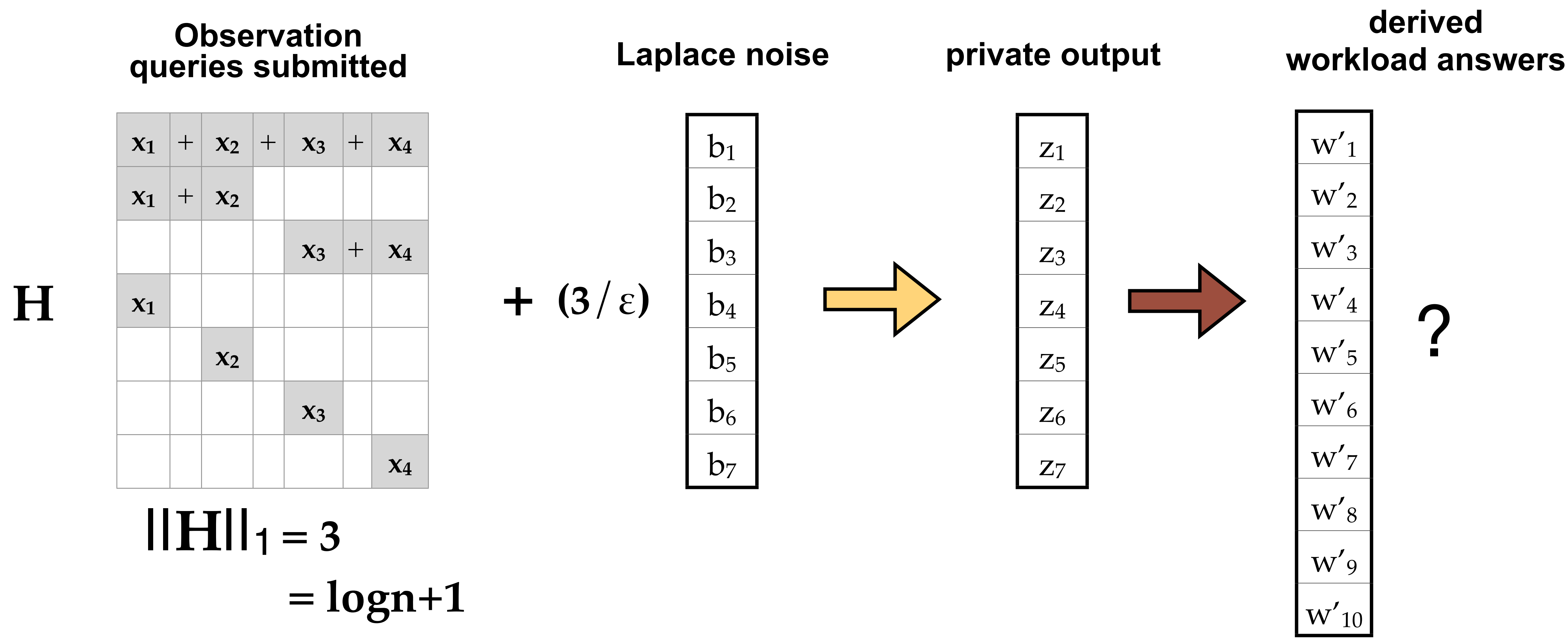
$z_5 + z_6$

$z_2 - z_4 + z_6$

[Hay, PVLDB 10]

Method 3: hierarchical observations

Hierarchical queries: recursively partition the domain, computing sums of each interval.



Possible estimates for query $\mathrm{range}(x_2, x_3) = x_2 + x_3$

$z_5 + z_6$ | $z_2 - z_4 + z_6$ | $z_1 - z_4 - z_7$

Method 3: hierarchical observations

Hierarchical queries: recursively partition the domain, computing sums of each interval.

Observation queries submitted | **Laplace noise** | **private output** | **derived workload answers**

H

x_1	+	x_2	+	x_3	+	x_4
x_1	+	x_2				
				x_3	+	x_4
x_1						
		x_2				
				x_3		
						x_4

$$+ \ (3/\varepsilon) \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \Rightarrow \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \\ z_7 \end{bmatrix} \Rightarrow \begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \\ w'_5 \\ w'_6 \\ w'_7 \\ w'_8 \\ w'_9 \\ w'_{10} \end{bmatrix} \ ?$$

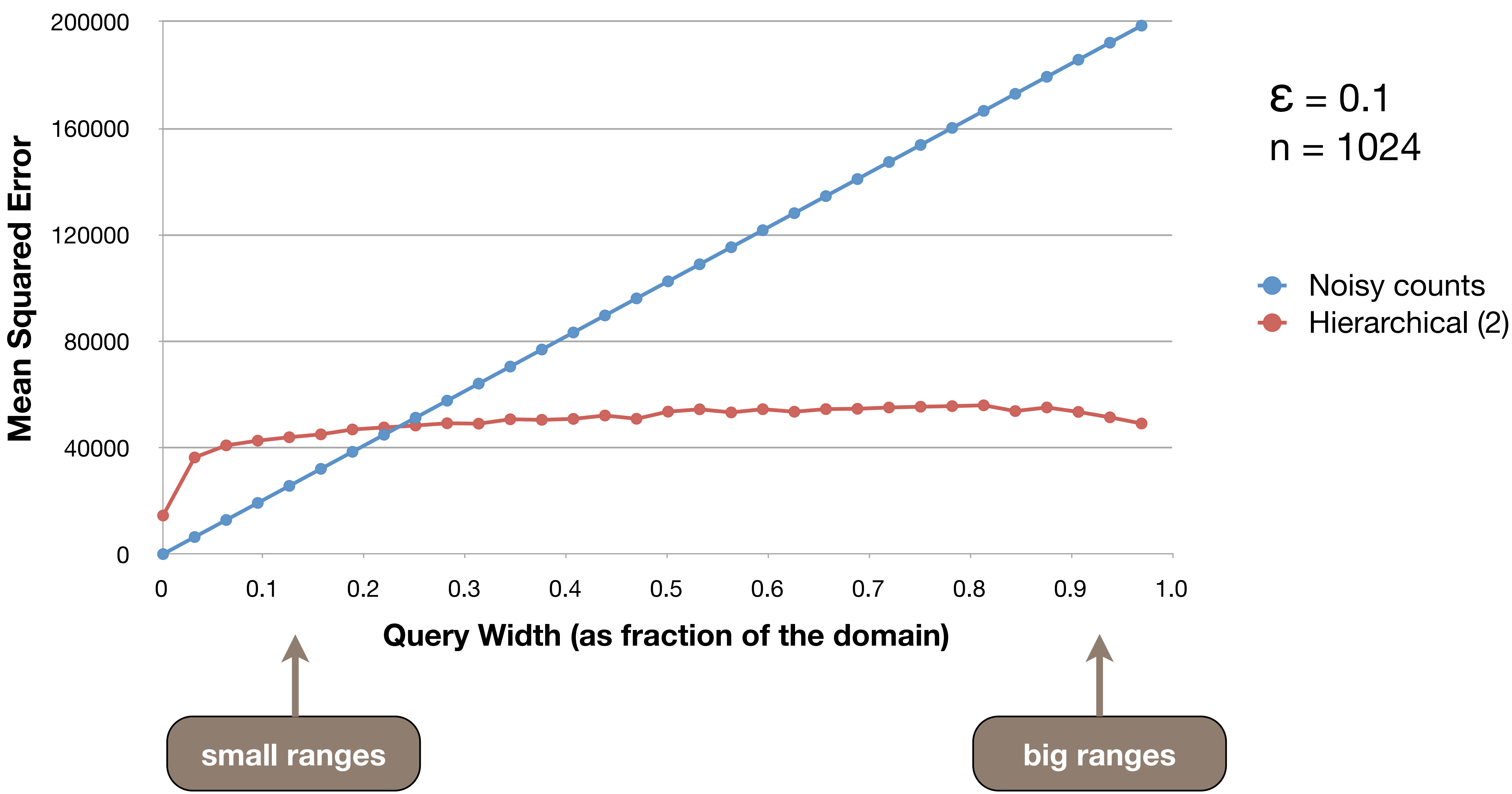
$$||\mathbf{H}||_1 = 3$$
$$= \log n + 1$$

Possible estimates for query $\mathrm{range}(x_2, x_3) = x_2 + x_3$

Least-squares estimate	$(6z_1 + 3z_2 + 3z_3 - 9z_4 + 12z_5 + 12z_6 - 9z_7)/21$

Error rates: workload of all range queries

ε-differential privacy



[Xiao, ICDE 10]

Method 4: wavelet queries

Wavelet: use Haar wavelet as observations.

Observation queries submitted | **Laplace noise** | **private output** | **derived workload answers**

$$
\mathbf{Y}\quad
\begin{bmatrix}
x_1 + x_2 + x_3 + x_4 \\
x_1 + x_2 - x_3 - x_4 \\
x_1 - x_2 \\
x_3 - x_4
\end{bmatrix}
+ (3/\varepsilon)
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}
\Rightarrow
\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}
\Rightarrow
\begin{bmatrix} w'_1 \\ w'_2 \\ w'_3 \\ w'_4 \\ w'_5 \\ w'_6 \\ w'_7 \\ w'_8 \\ w'_9 \\ w'_{10} \end{bmatrix}
\quad ?
$$

$$||\mathbf{Y}||_1 = 3$$
$$= \log n + 1$$

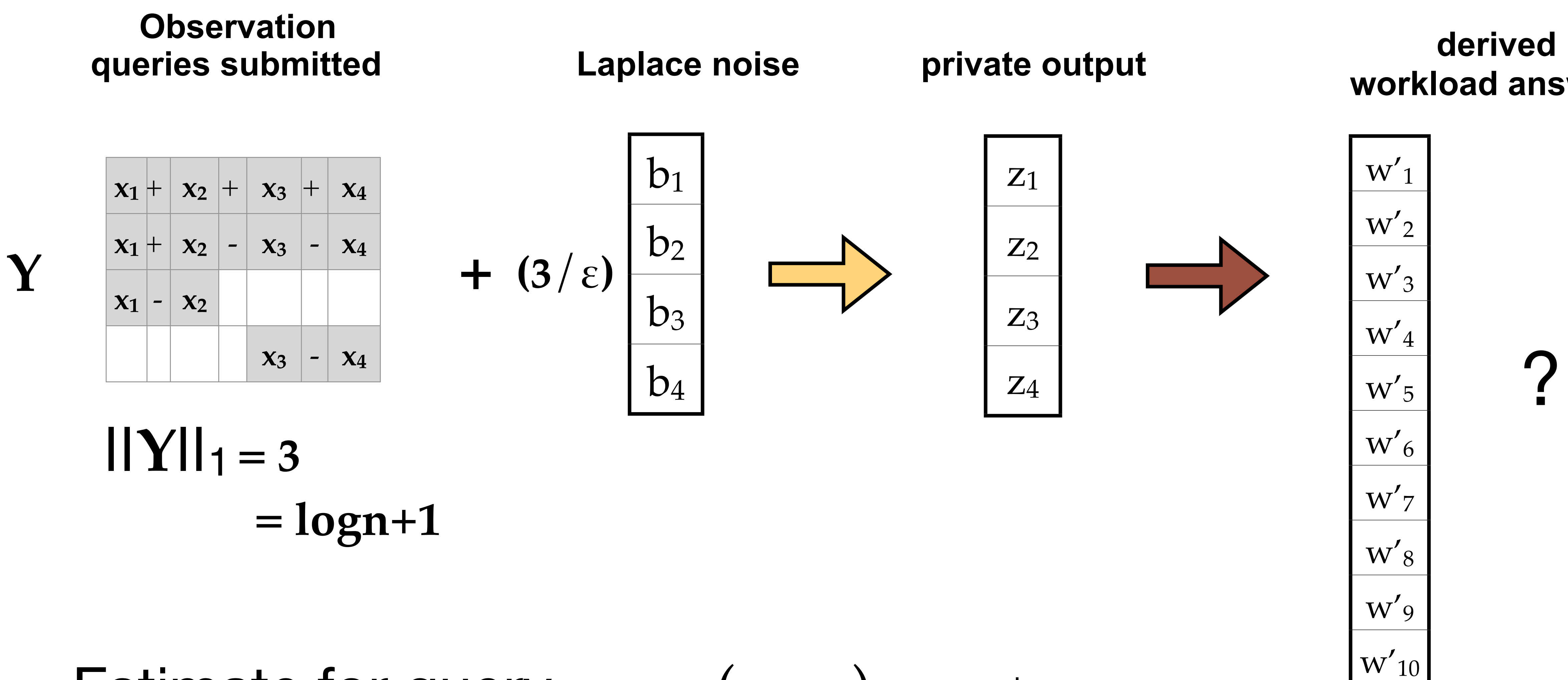
Estimate for query $\text{range}(x_2, x_3) = x_2 + x_3$

$$.5z_1 + 0z_2 - .5z_3 + .5z_4$$

[Xiao, ICDE 10]

Method 4: wavelet queries

Wavelet: use Haar wavelet as observations.



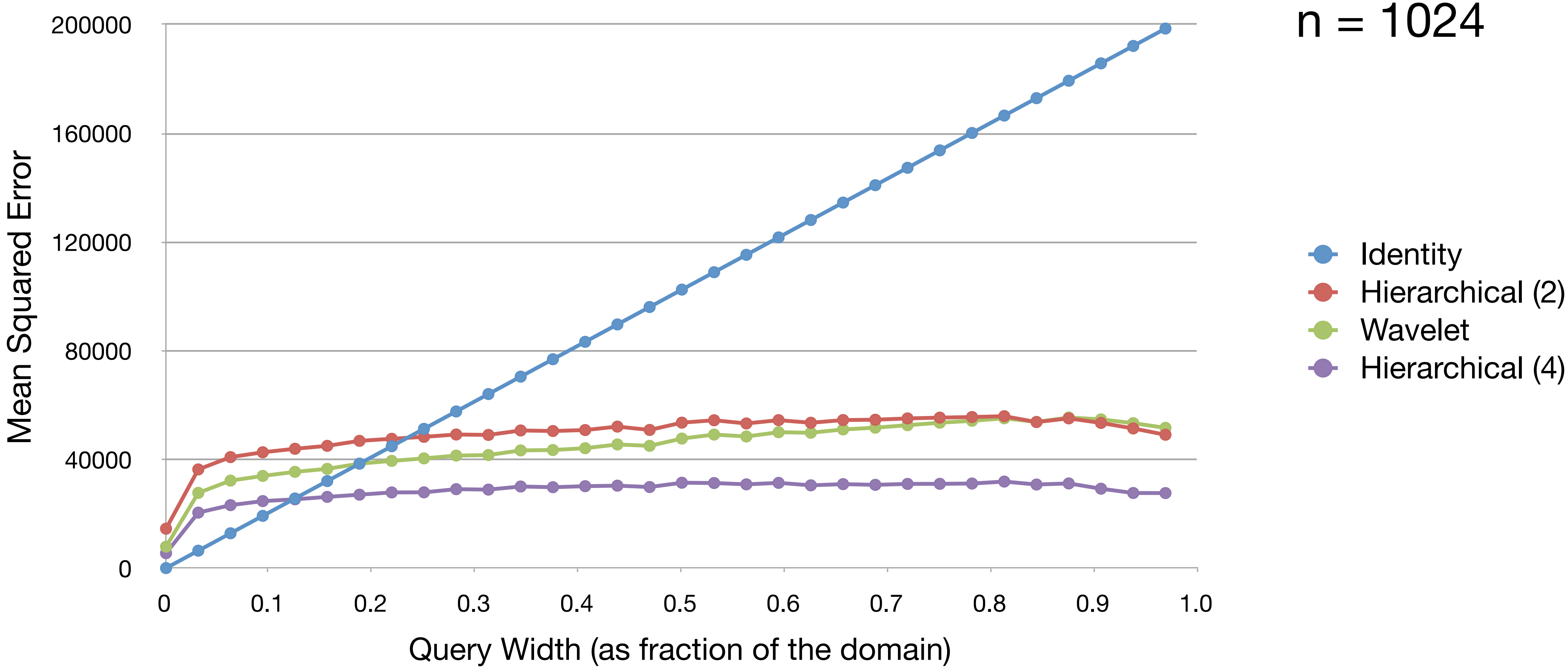
Estimate for query $\text{range}(x_2,x_3) = x_2 + x_3$

$$.5z_1 + 0z_2 - .5z_3 + .5z_4$$

Error: workload of all range queries

ε-differential privacy

$\varepsilon = 0.1$
$n = 1024$



Observations for the workload of all range queries

Noisy counts

x_1						
		x_2				
				x_3		
						x_4

I

Hierarchical

x_1	+	x_2	+	x_3	+	x_4
x_1	+	x_2				
				x_3	+	x_4
x_1						
		x_2				
				x_3		
						x_4

H

Wavelet

x_1	+	x_2	+	x_3	+	x_4
x_1	+	x_2	-	x_3	-	x_4
x_1	-	x_2				
				x_3	-	x_4

Y

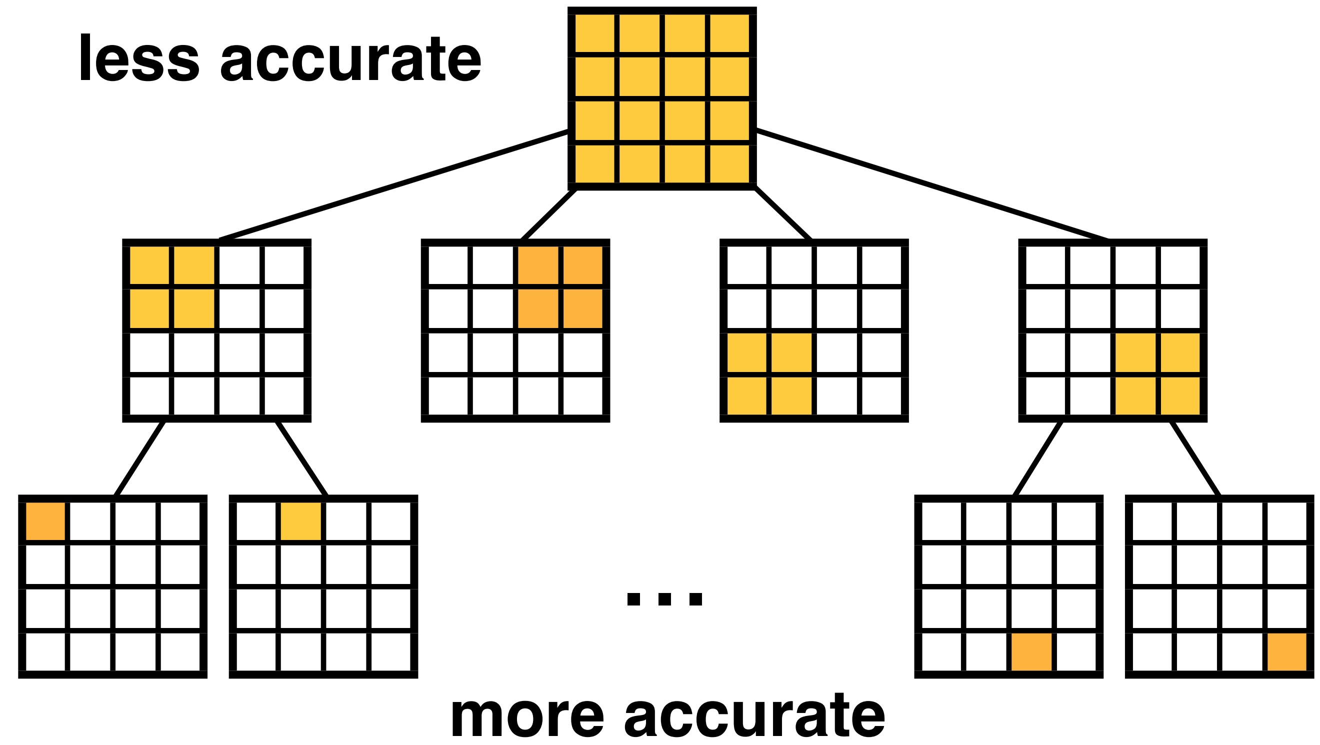
Very low sensitivity, but large ranges estimated badly.

Low sensitivity, and all range queries can be estimated using no more than logn output entries.

Max/Avg error	I		H	Y
	$O(n/\varepsilon^2)$	1-dim	$O(\log^3 n/\varepsilon^2)$	$O(\log^3 n/\varepsilon^2)$
		k-dim		$O(\log^{3k} n/\varepsilon^2)$

Observations for alternative workloads

- **Workload**: sets of 2D range queries
- **Observations:** [Cormode, ICDE '12]
  - Quad-tree queries
  - Geometrically increasing ε by level



- **Workload**: sets of low-order marginals
- **Observations:** [Barak, PODS '07]
  - Fourier basis queries

$$H_i = \begin{bmatrix} H_{i-1} & H_{i-1} \\ H_{i-1} & -H_{i-1} \end{bmatrix}$$

Questions raised

	Workload	Observations	Citation
Non-adaptive	low-order marginals	Fourier basis queries	**[Barak, PODS '07]**
	all one-dim range queries	Hierarchical ranges	**[Hay, PVLDB '10]**
	all (multi-dim) range queries	Haar wavelet queries	**[Xiao, ICDE '10]**
	2-dim range queries	Quad-tree queries	**[Cormode, ICDE '12]**

- Are these observations optimal for the targeted workloads?
- Which observations should we use for other custom workloads?

Questions raised

	Workload		Observations	Citation
Non-adaptive	low-order marginals	⟺	Fourier basis queries	[Barak, PODS ‘07]
	all one-dim range queries	⟺	Hierarchical ranges	[Hay, PVLDB ‘10]
	all (multi-dim) range queries	⟺	Haar wavelet queries	[Xiao, ICDE ‘10]
	2-dim range queries	⟺	Quad-tree queries	[Cormode, ICDE ’12]

- Are these observations optimal for the targeted workloads?
- Which observations should we use for other custom workloads?

Questions raised

	Workload	Observations	Citation
Non-adaptive	low-order marginals	Fourier basis queries	[Barak, PODS ‘07]
	all one-dim range queries	Hierarchical ranges	[Hay, PVLDB ‘10]
	all (multi-dim) range queries	Haar wavelet queries	[Xiao, ICDE ‘10]
	2-dim range queries	Quad-tree queries	[Cormode, ICDE ’12]

- Are these observations optimal for the targeted workloads?
- Which observations should we use for other custom workloads?

Adapt observations to workload

Outline

1. Preliminaries
2. Approach 1: workload-aware
    - Fixed Observations
    - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

Outline

1. Preliminaries
2. Approach 1: workload-aware
    - Fixed Observations
    - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

Laplace mechanism (matrix notation)

$$\mathbf{Laplace(W,x) = Wx + (||W||_1 / \varepsilon)b}$$

$\mathbf{W}$	m×n	workload
$\mathbf{x}$	n×1	database
$\|\|\mathbf{W}\|\|_1$	scalar	sensitivity
$\mathbf{b}$	m×1	noise: independent samples from Laplace(1)

Laplace mechanism (matrix notation)

$$\textbf{Laplace}(\mathbf{W},\mathbf{x}) = \mathbf{W}\mathbf{x} + (\|\mathbf{W}\|_1 / \varepsilon)\mathbf{b}$$

$\mathbf{W}$	m×n	workload
$\mathbf{x}$	n×1	database
$\|\mathbf{W}\|_1$	scalar	sensitivity
$\mathbf{b}$	m×1	noise: independent samples from Laplace(1)

$$\textbf{Error}(\mathbf{w}) = 2\,(\|\mathbf{W}\|_1 / \varepsilon)^2$$

The matrix mechanism: justification

The matrix mechanism: justification

❶ (**Select Observations**) Choose a (full rank) query matrix $\mathbf{A}$

The matrix mechanism: justification

❶ (**Select Observations**) Choose a (full rank) query matrix $\mathbf{A}$

❷ (**Apply Laplace**) Use the Laplace mechanism to answer $\mathbf{A}$

The matrix mechanism: justification

❶ (**Select Observations**) Choose a (full rank) query matrix $\mathbf{A}$

❷ (**Apply Laplace**) Use the Laplace mechanism to answer $\mathbf{A}$

$$\mathbf{z} = \mathbf{A}\mathbf{x} + (||\mathbf{A}||_1 / \varepsilon)\mathbf{b}$$

The matrix mechanism: justification

❶ (**Select Observations**) Choose a (full rank) query matrix $\mathbf{A}$

❷ (**Apply Laplace**) Use the Laplace mechanism to answer $\mathbf{A}$

$$\mathbf{z} = \mathbf{A}\mathbf{x} + (||\mathbf{A}||_1 / \varepsilon)\mathbf{b}$$

❸ (**Derive answers**) Compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ using answers $\mathbf{z}$.

The matrix mechanism: justification

❶ (**Select Observations**) Choose a (full rank) query matrix $\mathbf{A}$

❷ (**Apply Laplace**) Use the Laplace mechanism to answer $\mathbf{A}$

$$\mathbf{z} = \mathbf{A}\mathbf{x} + (||\mathbf{A}||_1 / \varepsilon)\mathbf{b}$$

❸ (**Derive answers**) Compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ using answers $\mathbf{z}$.

- compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ that minimizes squared error: $\| \mathbf{A}\underline{\mathbf{x}} - \mathbf{z} \|_2^2$

The matrix mechanism: justification

❶ (**Select Observations**) Choose a (full rank) query matrix $\mathbf{A}$

❷ (**Apply Laplace**) Use the Laplace mechanism to answer $\mathbf{A}$

$$\mathbf{z} = \mathbf{A}\mathbf{x} + (||\mathbf{A}||_1 / \varepsilon)\mathbf{b}$$

❸ (**Derive answers**) Compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ using answers $\mathbf{z}$.

- compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ that minimizes squared error: $||\mathbf{A}\underline{\mathbf{x}} - \mathbf{z}||_2^2$
- solution is the ordinary least squares estimator:

$$\underline{\mathbf{x}} = \mathbf{A}^+\mathbf{z} \qquad \text{where } \mathbf{A}^+ = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$$

The matrix mechanism: justification

❶ (**Select Observations**) Choose a (full rank) query matrix $\mathbf{A}$

❷ (**Apply Laplace**) Use the Laplace mechanism to answer $\mathbf{A}$

$$\mathbf{z} = \mathbf{Ax} + (||\mathbf{A}||_1 / \varepsilon)\mathbf{b}$$

❸ (**Derive answers**) Compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ using answers $\mathbf{z}$.

- compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ that minimizes squared error: $||\mathbf{A}\underline{\mathbf{x}} - \mathbf{z}||_2^2$
- solution is the ordinary least squares estimator:

$$\underline{\mathbf{x}} = \mathbf{A}^{+}\mathbf{z} \qquad \text{where } \mathbf{A}^{+} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}$$

Thm: $\underline{\mathbf{x}}$ is unbiased and has the least variance among all linear unbiased estimators.

The matrix mechanism: justification

❶ (**Select Observations**) Choose a (full rank) query matrix $\mathbf{A}$

❷ (**Apply Laplace**) Use the Laplace mechanism to answer $\mathbf{A}$

$$\mathbf{z} = \mathbf{A}\mathbf{x} + (\|\mathbf{A}\|_1 / \varepsilon)\mathbf{b}$$

❸ (**Derive answers**) Compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ using answers $\mathbf{z}$.

- compute estimate $\underline{\mathbf{x}}$ of $\mathbf{x}$ that minimizes squared error: $\| \mathbf{A}\underline{\mathbf{x}} - \mathbf{z} \|_2^2$
- solution is the ordinary least squares estimator:

$$\underline{\mathbf{x}} = \mathbf{A}^{+}\mathbf{z} \qquad \text{where } \mathbf{A}^{+} = (\mathbf{A}^{T}\mathbf{A})^{-1}\mathbf{A}^{T}$$

- Compute workload queries using estimate $\underline{\mathbf{x}}$:

$$\mathbf{W}\underline{\mathbf{x}}$$

Thm: $\underline{\mathbf{x}}$ is unbiased and has the least variance among all linear unbiased estimators.

The matrix mechanism

Given a workload $\mathbf{W}$, and any full-rank strategy matrix $\mathbf{A}$, the following randomized algorithm is ε-differentially private:

$$\mathbf{Matrix_A(W,x) = Wx + (||A||_1 / \varepsilon)\, WA^+ b}$$

$\mathbf{b = Lap(1)}$

The matrix mechanism

Given a workload $\mathbf{W}$, and any full-rank strategy matrix $\mathbf{A}$, the following randomized algorithm is ε-differentially private:

$$\boxed{\mathbf{Matrix_A(W,x) = Wx + (||A||_1 / \varepsilon)\, WA^+ b}}$$

$\mathbf{b=Lap(1)}$

instantiated with observations A

The matrix mechanism

Given a workload $\mathbf{W}$, and any full-rank strategy matrix $\mathbf{A}$, the following randomized algorithm is ε-differentially private:

$$\mathbf{Matrix_A(W,x) = Wx + (||A||_1 / \varepsilon)\, WA^+ b}$$

$\mathbf{b=Lap(1)}$

instantiated with observations A

true answer

The matrix mechanism

Given a workload $\mathbf{W}$, and any full-rank strategy matrix $\mathbf{A}$, the following randomized algorithm is ε-differentially private:

$$\boxed{\mathbf{Matrix_A(W,x) = Wx + (||A||_1 / \varepsilon)\, WA^+ b}}$$

$\mathbf{b = Lap(1)}$

instantiated with observations A

true answer

scaling by $||\mathbf{A}||_1$

The matrix mechanism

Given a workload $\mathbf{W}$, and any full-rank strategy matrix $\mathbf{A}$, the following randomized algorithm is ε-differentially private:

$$\mathbf{Matrix_A(W,x) = Wx + (||A||_1 / \varepsilon)\, WA^+ b} \qquad \mathbf{b=Lap(1)}$$

instantiated with observations A

true answer

scaling by $||\mathbf{A}||_1$

transformation by $\mathbf{WA}^+$

The matrix mechanism

Given a workload $\mathbf{W}$, and any full-rank strategy matrix $\mathbf{A}$, the following randomized algorithm is ε-differentially private:

$$\mathbf{Matrix_A(W,x) = Wx + (||A||_1 / \varepsilon)\, WA^+ b}$$

$\mathbf{b=Lap(1)}$

instantiated with observations A

true answer

scaling by $||\mathbf{A}||_1$

transformation by $\mathbf{WA^+}$

Compare with the Laplace mechanism:

$$\mathbf{Laplace(W,x) = Wx + (||W||_1 / \varepsilon) b}$$

Instances of the matrix mechanism

Given workload W of linear queries:

Observation Matrix A	Resulting mechanism
A = W	Never worse than Laplace -- sometimes better
A = Identity matrix	a common baseline
A = Haar wavelet	[Xiao, ICDE ‘10]
A = tree based	[Hay, PVLDB ‘10] [Cormode, ICDE ’12]
A = fourier basis	[Barak, PODS ‘07]

Observation matrices equivalent to wavelet

Wavelet Y

1	1	1	1
1	1	-1	-1
1	-1	0	0
0	0	1	-1

$\equiv$

Y′

1	1	0	0
0	0	1	1
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

$>$

Y″

1	1	0	0
0	0	1	1
$\sqrt{2}$	0	0	0
0	$\sqrt{2}$	0	0
0	0	$\sqrt{2}$	0
0	0	0	$\sqrt{2}$

$||Y||_1 = 3$

$||Y'||_1 = 3$

$||Y''||_1 = 2.414$

Observation matrices equivalent to wavelet

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} > \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \sqrt{2} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{bmatrix}$$

Equivalent error for all queries

Wavelet Y $\|Y\|_1 = 3$

Y′ $\|Y'\|_1 = 3$

Y″ $\|Y''\|_1 = 2.414$

Observation matrices equivalent to wavelet

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} > \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \sqrt{2} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{bmatrix}$$

Equivalent error for all queries

Lower error for all queries

Wavelet Y

$||Y||_1 = 3$

Y′

$||Y'||_1 = 3$

Y″

$||Y''||_1 = 2.414$

Observation matrices equivalent to wavelet

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} > \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \sqrt{2} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 \\ 0 & 0 & 0 & \sqrt{2} \end{bmatrix}$$

Equivalent error for all queries

Lower error for all queries

Wavelet Y — $||\mathbf{Y}||_1 = 3$

Y′ — $||\mathbf{Y}'||_1 = 3$

Y″ — $||\mathbf{Y}''||_1 = 2.414$

The haar wavelet observation matrix $\mathbf{Y}$ is **dominated** by alternative matrix $\mathbf{Y}''$.

Error of matrix mechanism

Given an observation matrix $\mathbf{A}$ and workload $\mathbf{W}$, the error under the mechanism $\mathbf{Matrix_A}$ is:

For a single query $\mathbf{w}$ in $\mathbf{W}$:

$$\mathbf{Error_A(w)} = (2/\varepsilon^2)(||\mathbf{A}||_1)^2\ \mathbf{w}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{w}^T$$

Total error for workload $\mathbf{W}$:

$$\mathbf{TotalError_A(w)} = (2/\varepsilon^2)(||\mathbf{A}||_1)^2\ \mathbf{trace}(\ \mathbf{W}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{W}^T\ )$$

Error independent of the input data

Optimal selection of observations

Objective: given workload $\mathbf{W}$, find the observation matrix $\mathbf{A}$ that minimizes the **total** error.

Optimal selection of observations

Objective: given workload $\mathbf{W}$, find the observation matrix $\mathbf{A}$ that minimizes the **total** error.

Privacy	Optimization Objective	Problem Type	Runtime

Optimal selection of observations

Objective: given workload $\mathbf{W}$, find the observation matrix $\mathbf{A}$ that minimizes the **total** error.

Privacy	Optimization Objective	Problem Type	Runtime
ε DP	Given $\mathbf{W}$ consisting of data cube queries, choose $\mathbf{A}$ consisting of data cube queries to minimize simplified error measure. **[Ding, SIGMOD '11]**	set-cover approx	O(n)

Optimal selection of observations

Objective: given workload $\mathbf{W}$, find the observation matrix $\mathbf{A}$ that minimizes the **total** error.

Privacy	Optimization Objective	Problem Type	Runtime
ε DP	Given $\mathbf{W}$ consisting of data cube queries, choose $\mathbf{A}$ consisting of data cube queries to minimize simplified error measure. **[Ding, SIGMOD '11]**	set-cover approx	O(n)
ε DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP w/ rank constraints	$O(n^8)$

Optimal selection of observations

Objective: given workload $\mathbf{W}$, find the observation matrix $\mathbf{A}$ that minimizes the **total** error.

Privacy	Optimization Objective	Problem Type	Runtime
ε DP	Given $\mathbf{W}$ consisting of data cube queries, choose $\mathbf{A}$ consisting of data cube queries to minimize simplified error measure. **[Ding, SIGMOD '11]**	set-cover approx	O(n)
ε DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP w/ rank constraints	$O(n^8)$
(ε,δ) DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP	$O(n^8)$

Optimal selection of observations

Objective: given workload $\mathbf{W}$, find the observation matrix $\mathbf{A}$ that minimizes the **total** error.

Privacy	Optimization Objective	Problem Type	Runtime
ε DP	Given $\mathbf{W}$ consisting of data cube queries, choose $\mathbf{A}$ consisting of data cube queries to minimize simplified error measure. **[Ding, SIGMOD '11]**	set-cover approx	O(n)
ε DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP w/ rank constraints	$O(n^8)$
(ε,δ) DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP	$O(n^8)$
ε DP	Given $\mathbf{W}$, choose $\mathbf{AB} \approx \mathbf{W}$ to minimize $\mathbf{TotalError_A(AB)}$ **[Yuan, VLDB '12]**	bi-convex opt	$O(n^4)$

Optimal selection of observations

Objective: given workload $\mathbf{W}$, find the observation matrix $\mathbf{A}$ that minimizes the **total** error.

Privacy	Optimization Objective	Problem Type	Runtime
ε DP	Given $\mathbf{W}$ consisting of data cube queries, choose $\mathbf{A}$ consisting of data cube queries to minimize simplified error measure. **[Ding, SIGMOD '11]**	set-cover approx	O(n)
ε DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP w/ rank constraints	$O(n^8)$
(ε,δ) DP	Given $\mathbf{W}$, choose $\mathbf{A}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PODS '10]**	SDP	$O(n^8)$
ε DP	Given $\mathbf{W}$, choose $\mathbf{AB}\approx\mathbf{W}$ to minimize $\mathbf{TotalError_A(AB)}$ **[Yuan, VLDB '12]**	bi-convex opt	$O(n^4)$
(ε,δ) DP	Given $\mathbf{W}$, choose optimal scaling of eigenvectors of $\mathbf{W}$ to minimize $\mathbf{TotalError_A(W)}$ **[Li, PVLDB '12]**	convex opt	$O(n^4)$

Approximately optimal selection of observations

Matrix Mechanism under (ε,δ)-Differential Privacy

- Given $\mathbf{W}$, choose a set of **basis queries** for the observations:
    - $\mathbf{v_1}, \mathbf{v_2}, \ldots \mathbf{v_n}$ (the eigenvectors of $\mathbf{W}$)
    - compute optimal scalars to minimize error $c_1, c_2, \ldots c_n$
- Resulting observation matrix is:

$$\mathbf{A} = \begin{bmatrix} c_1\mathbf{v_1} \\ c_2\mathbf{v_2} \\ \ldots \\ c_n\mathbf{v_n} \end{bmatrix}$$

- Algorithm running time: O(n rank($\mathbf{W}$)^3)
- Efficiently solvable and achieves optimal error rates in practice.

Representative experimental findings

- **Benefit of fixed observations**:
  - W={All Range Queries} can be reduced by a factor of 2-4 by using wavelet or hierarchical observations. **[Xiao, ICDE '10] [Hay, PVLDB '10]**
- **Benefit of optimized observations**:
  - ε-DP: Error reduced by 2-3 times compared with fixed observation methods. **[Yuan, VLDB '12]**
  - (ε,δ)-DP: Error reduced by 2-6 times on range and marginal workloads for which fixed observation methods were designed; up to 10 times reduction for ad hoc workloads. **[Li, PVLDB '12]**

Note 1: comparisons don't depend on input data or privacy parameters.*

Note 2: ratios based on **root** mean squared error.

Lower bound on error

- Given workload $\mathbf{W}$ with singular values $\lambda_1 > \ldots > \lambda_n$, the minimum total error of the matrix mechanism is greater than or equal to:

Privacy	Error Lower Bound
ε-DP	$(2/\varepsilon^2)(1/n)(\lambda_1 + \ldots + \lambda_n)^2$
(ε,δ)-DP	$(2\log(2/\delta)/\varepsilon^2)(1/n)(\lambda_1 + \ldots + \lambda_n)^2$

(tight)

Runtime complexity

- Answering **W** using Laplace/Gaussian mechanism takes O(|**W**|n) time.

Costs	Fixed Observations	Optimized Observations
1. Select observations	-	~ $O(n^4)$
2. Apply standard mechanism	O(|**A**|n)	O(|**A**|n)
3. Derive answers	O(|**W**|n)	O(|**W**|n^2)

- Because of data-independence, observation matrix can be preprocessed:
  - Given fixed workload **W** and observation matrix **A**, runtime is O(|**W**|n) after pre-computation of $\mathbf{WA}^+$: no worse than standard mechanisms

Summary: workload-aware mechanisms

- Methods can be seen as a generalization of Laplace/Gaussian mechanism, with error rates significantly reduced and independent of data.

Workload	Observations		Citation
low-order marginals	Fixed	Fourier basis queries	[Barak, PODS '07]
all one-dim range queries	Fixed	Hierarchical ranges	[Hay, PVLDB '10]
all (multi-dim) range queries	Fixed	Haar wavelet queries	[Xiao, ICDE '10]
2-dim range queries	Fixed	Quad-tree queries	[Cormode, ICDE '12]
sets of data cubes	Optimized	sets of data cubes	[Ding, SIGMOD '11]
set of linear queries	Optimized	set of linear queries	[Li, PODS '10] [Li, PVLDB '12]
set of linear queries	Optimized	low-order set of linear queries	[Yuan, VLDB '12]

Summary: workload-aware mechanisms

- **Benefits**
  - Independence of data makes error analysis easy, error rates publishable to analyst, and improves efficiency in some cases.
- **Limitations**
  - Computational dependence on domain size, n.
  - Error dependence on epsilon: $1/\varepsilon^2$
  - For some workloads, there is no set of observations that can help much.
- **Open questions**
  - Alternative derivation methods: e.g. non-negative least squares
  - Relationship with "universal" error lower bounds for DP.

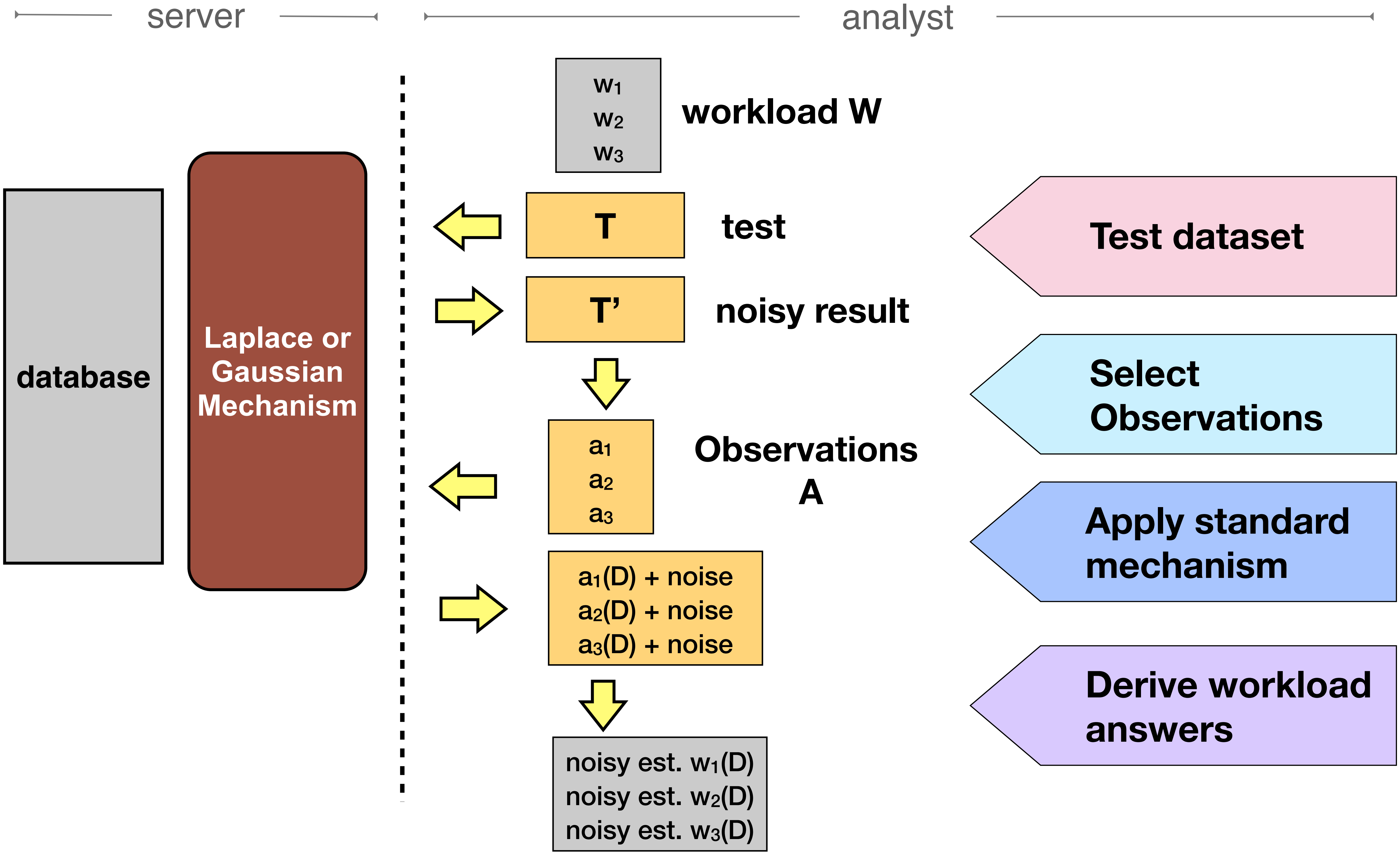
Outline

1. Preliminaries
2. Approach 1: workload-aware
    - Fixed Observations
    - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

Outline

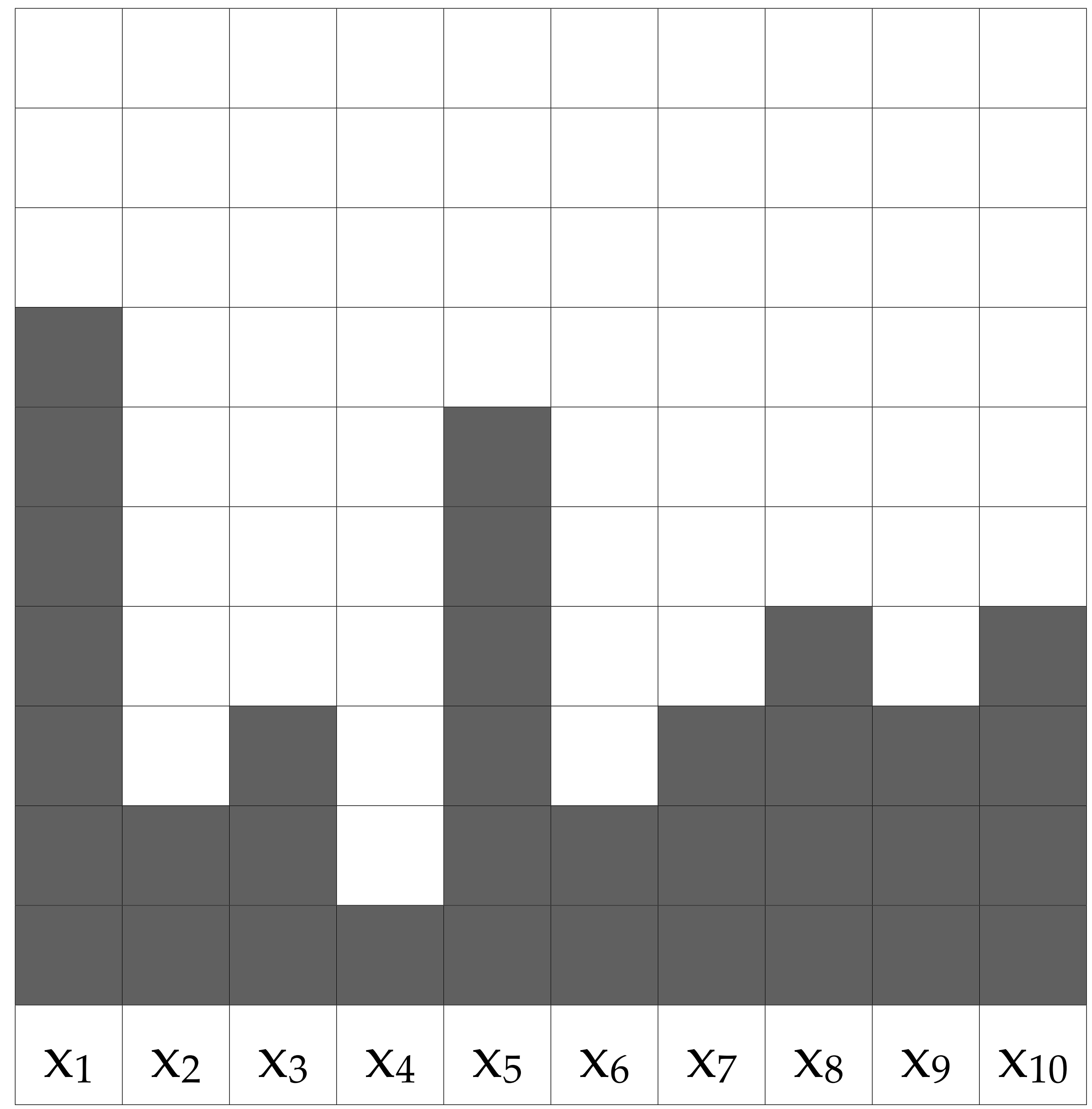
1. Preliminaries
2. Approach 1: workload-aware
   - Fixed Observations
   - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

(Recall) Approach 2: data-aware mechanisms



A basic intuition

- Detect when additional observations won't help much.
- Challenges:
  - Balance privacy budget between testing data and usable observations.
  - When possible, incorporate test observations into query answers.
  - Perturbation error vs. approximation error.



A basic intuition

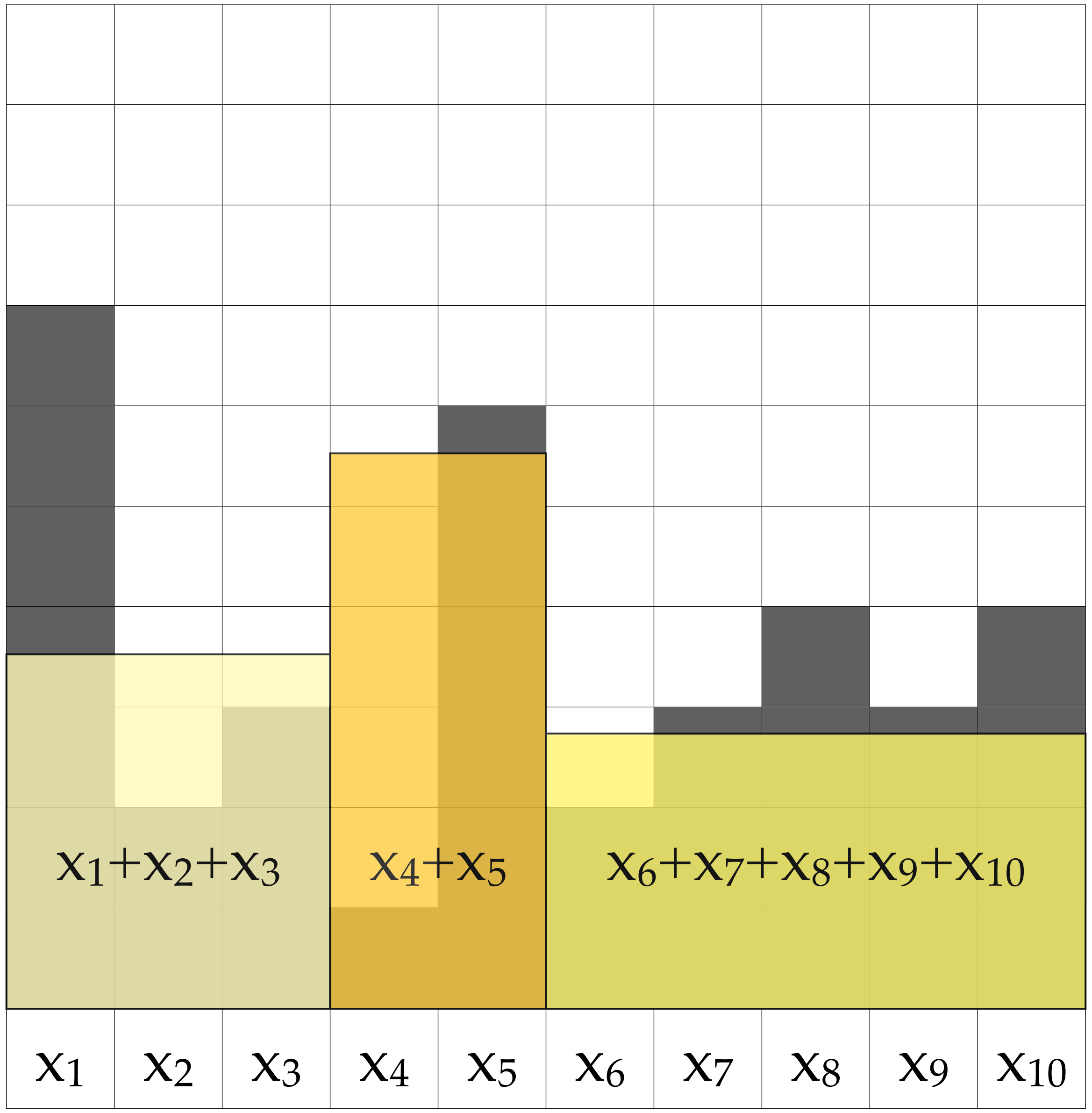
- Detect when additional observations won't help much.

- Challenges:

  - Balance privacy budget between testing data and usable observations.

  - When possible, incorporate test observations into query answers.

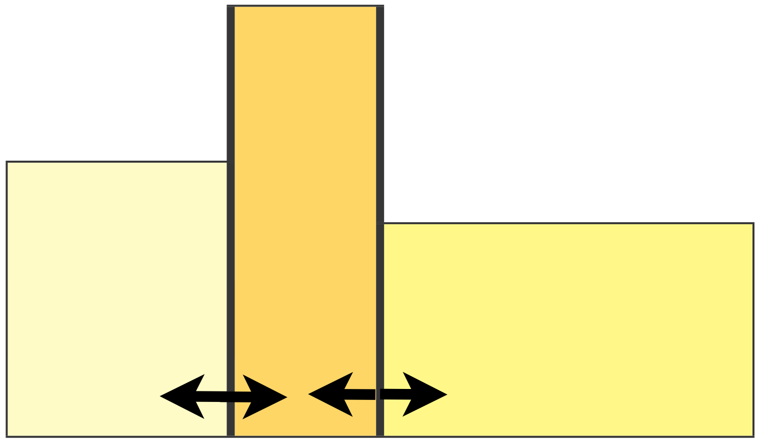
  - Perturbation error vs. approximation error.



Data-aware histogram

[Xu, ICDE ’12]

Workload	1D Range Queries
Parameters	k, ε_1, ε_2 s.t. $\varepsilon_1+\varepsilon_2=\varepsilon$



1. Compute a private estimate of the k-bin, variance-optimal histogram using the exponential mechanism. ε_1

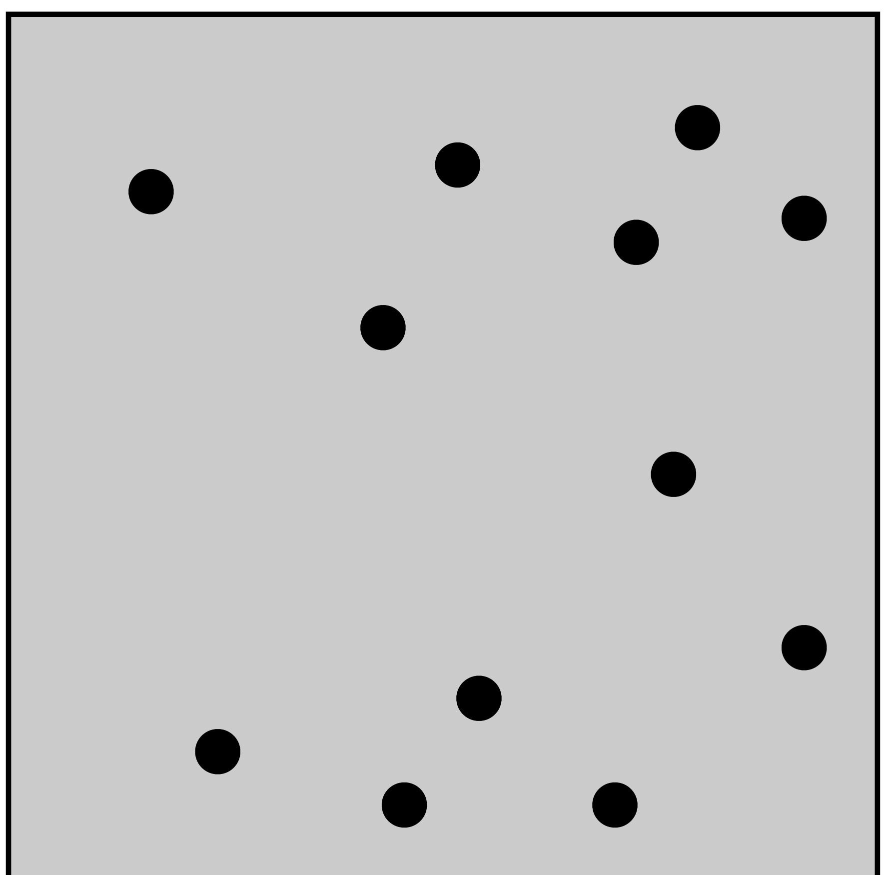
2. Use Laplace mechanism to get bin counts **and** all individual counts. ε_2

3. Derive answers to workload queries using least squares.

Techniques for spatial queries

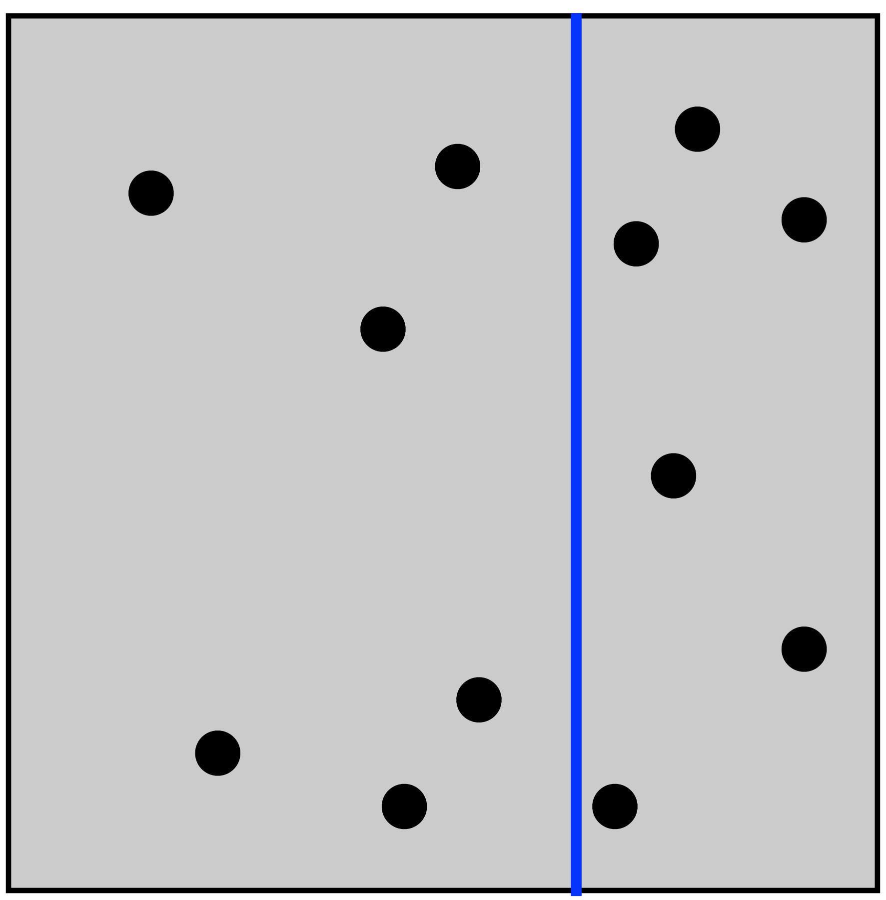
- Spatial queries are 2 dimensional counting queries (typically range queries)

- kd-tree: a data-aware hierarchical space partitioning data structure.



Techniques for spatial queries

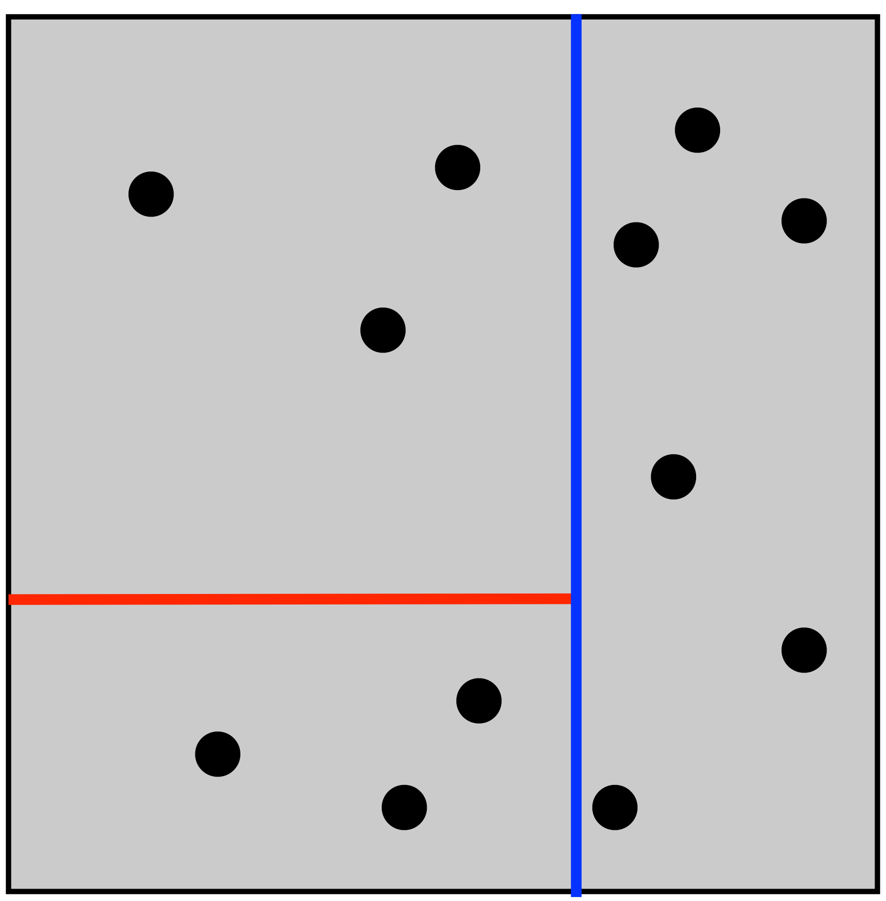
- Spatial queries are 2 dimensional counting queries (typically range queries)
- kd-tree: a data-aware hierarchical space partitioning data structure.



Techniques for spatial queries

- Spatial queries are 2 dimensional counting queries (typically range queries)

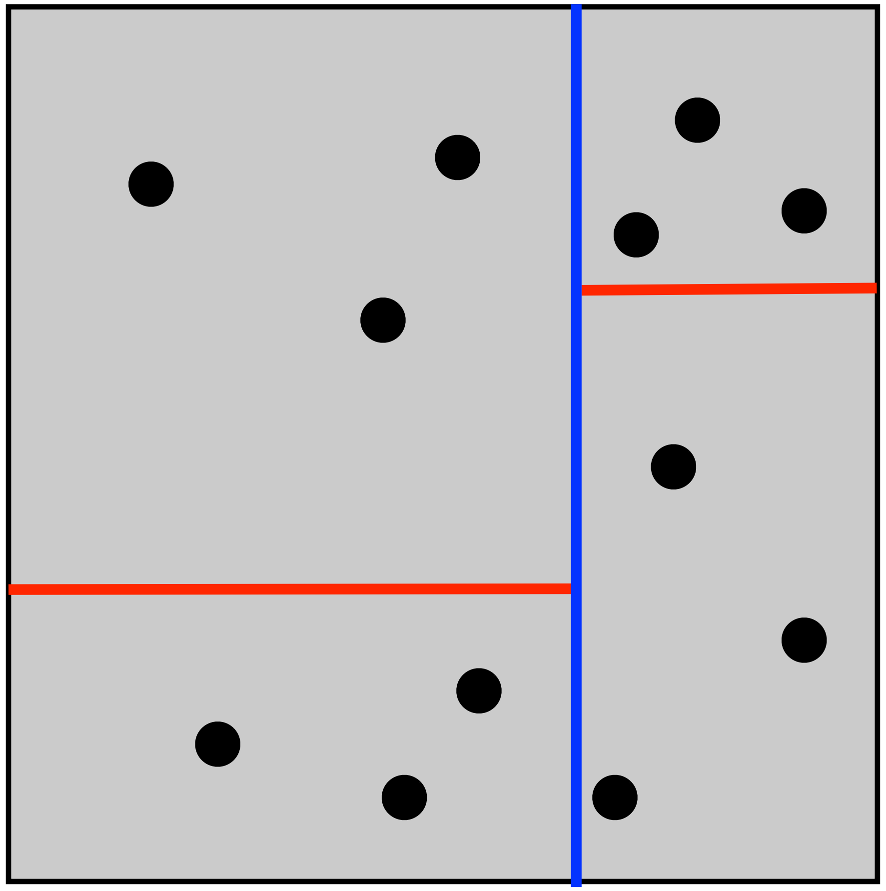
- kd-tree: a data-aware hierarchical space partitioning data structure.



Techniques for spatial queries

- Spatial queries are 2 dimensional counting queries (typically range queries)

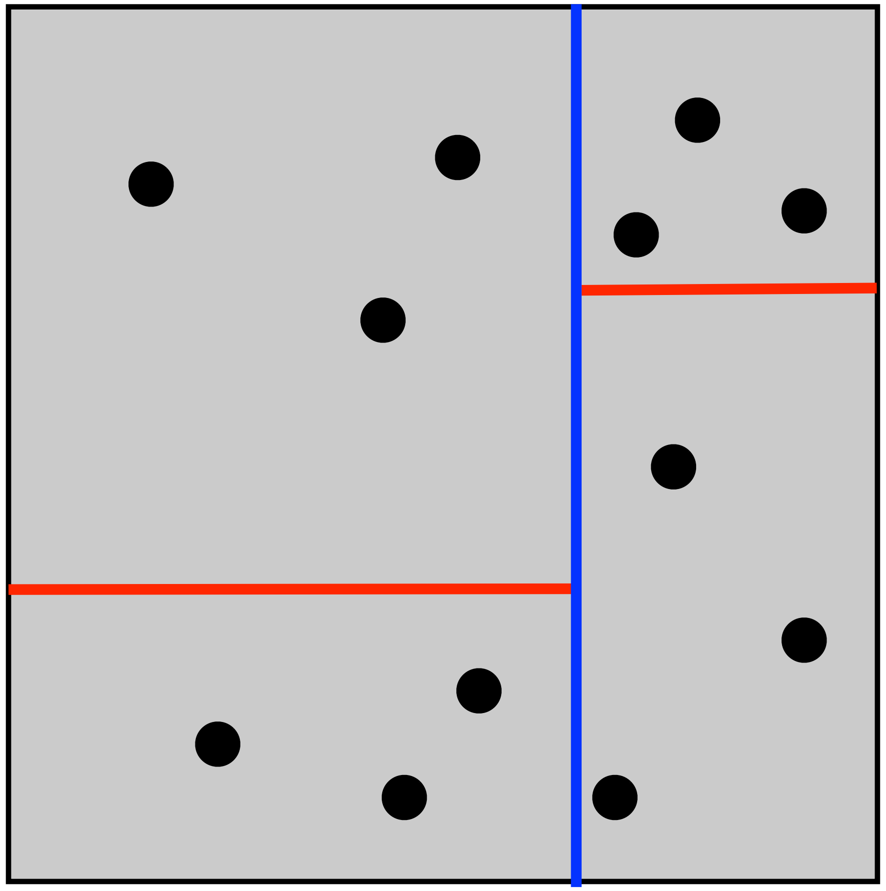
- kd-tree: a data-aware hierarchical space partitioning data structure.

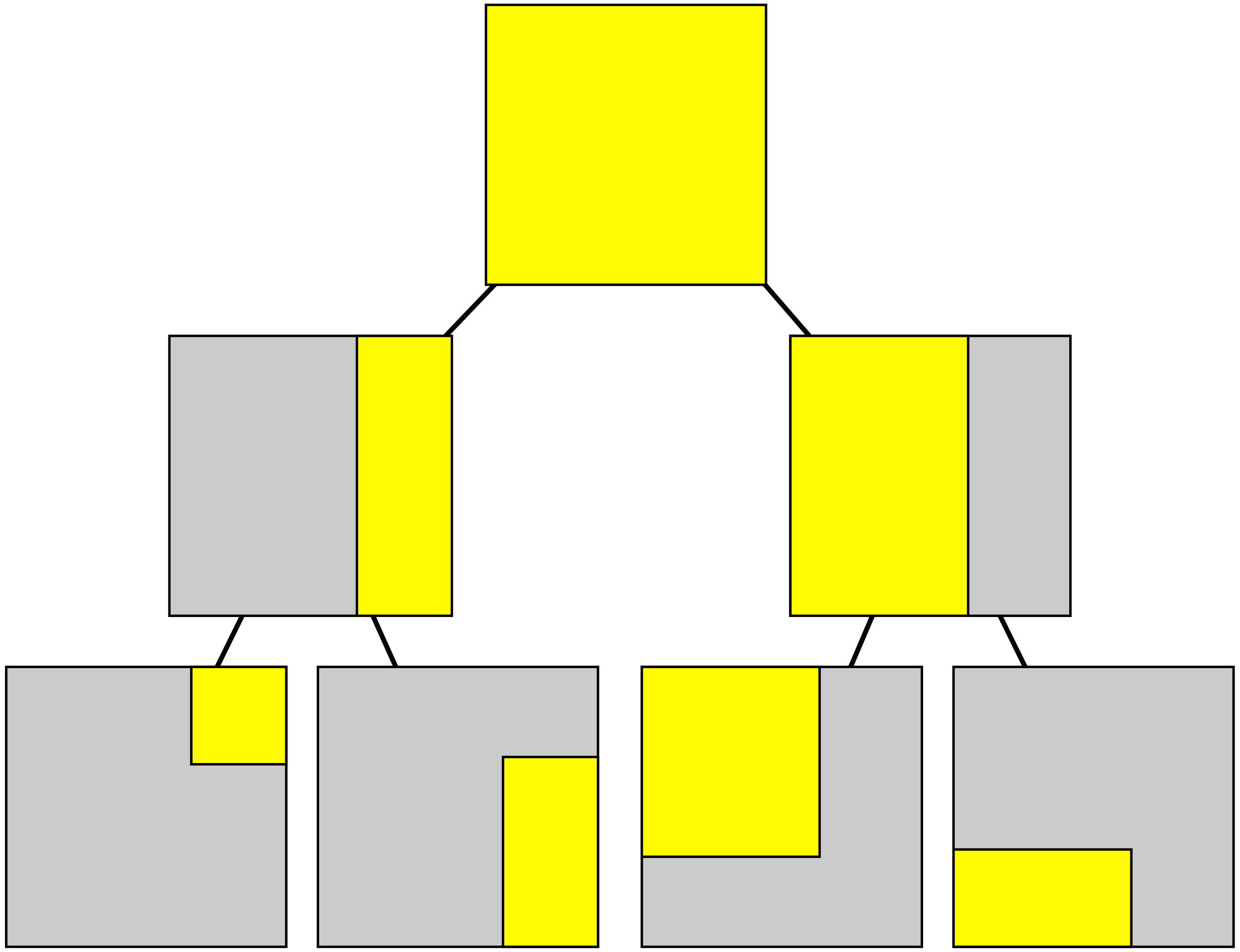


Techniques for spatial queries

- Spatial queries are 2 dimensional counting queries (typically range queries)

- kd-tree: a data-aware hierarchical space partitioning data structure.





Data-aware kd-tree (1)

[Xiao, SDM '10]

Workload	2D Range Queries
Parameters	p_1, p_2, ε_1, ε_2 s.t. $\varepsilon_1+\varepsilon_2=\varepsilon$

1. Use Laplace mechanism to get noisy counts: x'

2. Build kd-tree K from x', but stop splitting if:

   $\varepsilon_1=\varepsilon/2$

   - sum of counts in current region is too small (p_1), or
   - counts in current region are close to uniform (p_2)

3. Use Laplace mechanism to get noisy counts K' for all regions in K.

4. Compute workload answers from K' using least squares.

   $\varepsilon_2=\varepsilon/2$

Data-aware kd-tree (2)

[Cormode, ICDE '12]

Workload	2D Range Queries
Parameters	l, k, ε_1, ε_2 s.t. $\varepsilon_1+\varepsilon_2=\varepsilon$

1. Build hybrid hierarchical structure:
    - l-levels of kd-tree using exponential mechanism to compute median. $\varepsilon_1 = .3\varepsilon$
    - remaining (k-l) levels uniform quad-tree.
2. Use Laplace mechanism to get noisy counts. $\varepsilon_2 = .7\varepsilon$
3. Derive workload query answers using least squares.

Optimizing for relative error

[Xiao, SIGMOD '11]

Workload	marginals
Parameters	T, ε

1. Answer all workload queries using Laplace mechanism with budget ε/T

2. Repeat T-1 times:

   - Refine query answers, by resampling queries with small values.

- Final query answers have same privacy cost as single Laplace random variable with resulting error.

Multiplicative weights

[Hardt, NIPS '12]

Workload	linear queries
Parameters	T, ε_1, ε_2 s.t. $T(\varepsilon_1+\varepsilon_2)=\varepsilon$

- Begin with uniform estimate x_0 of database x
- For i = 1...T :
  - Evaluate all workload queries using current estimate x_{i-1}. Select inaccurate q_i with exponential mechanism. $\varepsilon_1 = \varepsilon / 2T$
  - Laplace mechanism: get noisy estimate m_i of q_i.
  - Update $x_{i-1} \rightarrow x_i$ using m_i: multiplicative weights. $\varepsilon_2 = \varepsilon / 2T$

Multiplicative weights

[Hardt, NIPS '12]

- Provably better dependence on ε than workload-aware techniques: squared error $O(1/\varepsilon^2)$ vs. $O(1/\varepsilon^{2/3})$
- Observations customized to workload.
- Very good accuracy for sparse datasets.
- Output satisfies non-negativity constraints.
- Must compute all workload queries T times.

Representative experimental findings

- Building a data-aware histogram reduces error on range queries by 20-40% compared with fixed workload-aware methods like wavelet or tree-based. **[Xu, ICDE ’12]**
- Neither of the data-aware kd-trees consistently outperform workload-aware quad-tree (on random sets of 2D range queries). **[Cormode, ICDE ’12]**
- For reasonable privacy parameters, small workloads of random range queries on sparse data, multiplicative weights can reduce error by a factor of 10 over matrix mechanism. **[Hardt, NIPS ’12]**
  - (But for other datasets, it can be outperformed by a factor of 10 by a fixed workload-aware method like wavelet.)

Note: ratios based on **root** mean squared error.

Data-aware mechanisms

- Observations selected to match properties of the database; generally efficient, but spending privacy budget on testing doesn't always pay off.

Workload	Observations	Citation
1D range queries	approx. v-optimal histogram	[Xu, ICDE '12]
2D range queries	kd-tree queries	[Xiao, SDM '10]
2D range queries	hybrid kd-tree queries	[Cormode, ICDE '12]
Marginals	scaled workload queries	[Xiao, SIGMOD '11]
Linear queries	subset of workload	[Hardt, NIPS '12]

Summary: data-aware mechanisms

- **Benefits**:
    - Lower error than Approach 1 in some cases.
- **Limitations**:
    - Parameters for algorithms must be selected carefully.
    - Public error rates not available to analyst.
    - Techniques are data-aware, but are they workload-aware?
- **Open questions**:
    - Evaluation highly dependent on workload, dataset, epsilon. What are “real” data and workloads like? What properties of data determine error?

Outline

1. Preliminaries
2. Approach 1: workload-aware
    - Fixed Observations
    - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

Outline

1. Preliminaries
2. Approach 1: workload-aware
   - Fixed Observations
   - Optimized Observations
3. Approach 2: data-aware
4. Conclusions

Summary and conclusions

- Two approaches to batch query answering, each of which provide significant error improvements by building on standard Laplace/ Gaussian mechanisms, but using alternative observations.
  - Workload-aware methods ignore the input data, and choose observations solely by analyzing the workload.
  - Data-aware methods carefully (i.e. privately) exploit properties of the input data.
- Both approaches are efficient for modestly sized domains.

Workload-aware

- **Benefits**
  - Independence of data makes error analysis easy, error rates publishable to analyst, and improves efficiency in some cases.
- **Limitations**
  - Computational dependence on domain size, n.
  - Error dependence on epsilon: $1/\varepsilon^2$
  - For some workloads, there is no set of observations that can help much.
- **Open questions**
  - Alternative derivation methods: e.g. non-negative least squares
  - Relationship with "universal" error lower bounds for DP.

Data-aware

- **Benefits**:
  - Lower error than Approach 1 in some cases.
- **Limitations**:
  - Parameters for algorithms must be selected carefully.
  - Public error rates not available to analyst.
  - Techniques are data-aware, but are they workload-aware?
- **Open questions**:
  - Evaluation highly dependent on workload, dataset, epsilon. What are "real" data and workloads like? What properties of data determine error?

Open issues

- What makes one workload “harder” to answer than another?
- What makes one database “harder” to support accurately?
- Can we avoid the computational dependence on the domain size n?
- How do we analyze the error resulting from non-negative least squares if applied in derivation of matrix mechanism?
- Methods for more expressive queries.

References

[Barak, PODS '07]	B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry, and K. Talwar. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In Principles of Database Systems (PODS) 2007.
[McSherry, FOCS '07]	F. McSherry and K. Talwar. Mechanism design via differential privacy. In FOCS '07
[McSherry, SIGMOD '09]	F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. SIGMOD 2009.
[Hay, PVLDB '10]	M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially-private queries through consistency. PVLDB, 2010.
[Xiao, ICDE '10]	X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. In International Conference on Data Engineering (ICDE), 2010.
[Li, PODS '10]	C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor. Optimizing Linear Counting Queries Under Differential Privacy. Principles of Database Systems (PODS) 2010.
[Xiao, SDM '10]	Y. Xiao, L. Xiong, and C. Yuan. Differentially private data release through multidimensional partitioning. Secure Data Management (SDM) 2010.
[Xiao, SIGMOD '11]	Xiaokui Xiao, Gabriel Bender, Michael Hay, and Johannes Gehrke. iReduct: Differential privacy with reduced relative errors. SIGMOD, 2011.
[Ding, SIGMOD '11]	B. Ding, M. Winslett, J. Han, and Z. Li. Differentially private data cubes: optimizing noise sources and consistency. In SIGMOD 2011.
[Xiao, SIGMOD '11]	X. Xiao, G. Bender, M. Hay, and J. Gehrke. iReduct: Differential privacy with reduced relative errors. In SIGMOD, 2011.
[Cormode, ICDE '12]	G. Cormode, M. Procopiuc, D. Srivastava, E. Shen, and T. Yu. Differentially private spatial decompositions. International Conference on Data Engineering (ICDE), 2012.

References (con't)

[Xu, ICDE '12]	J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu. Differentially private histogram publication. In ICDE, 2012.
[Li, PVLDB '12]	C. Li and G. Miklau. An adaptive mechanism for accurate query answering under differential privacy. Proceedings of the VLDB Endowment (PVLDB) 2012.
[Yuan, VLDB '12]	G. Yuan, Z. Zhang, M. Winslett, X. Xiao, Y. Yang, and Z. Hao. Low-rank mechanism: Optimizing batch queries under differential privacy. VLDB, 2012.
[Hardt, NIPS '12]	M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. In NIPS, 2012.