Cost-Intelligent Data Analytics in the Cloud

Huanchen Zhang

Tsinghua University

Database As a Type of Goods

User Profit		Utility		$ Cost
Π	$=$	$U(p)$	$-$	C

Database As a Type of Goods

User Profit **Utility** **$ Cost**

$$\Pi = \underbrace{U(p)}_{} - C$$

p: Performance

Optimization Logic in Traditional Databases

Traditional

```
0110
1001
1010
```

Optimization Logic in Traditional Databases

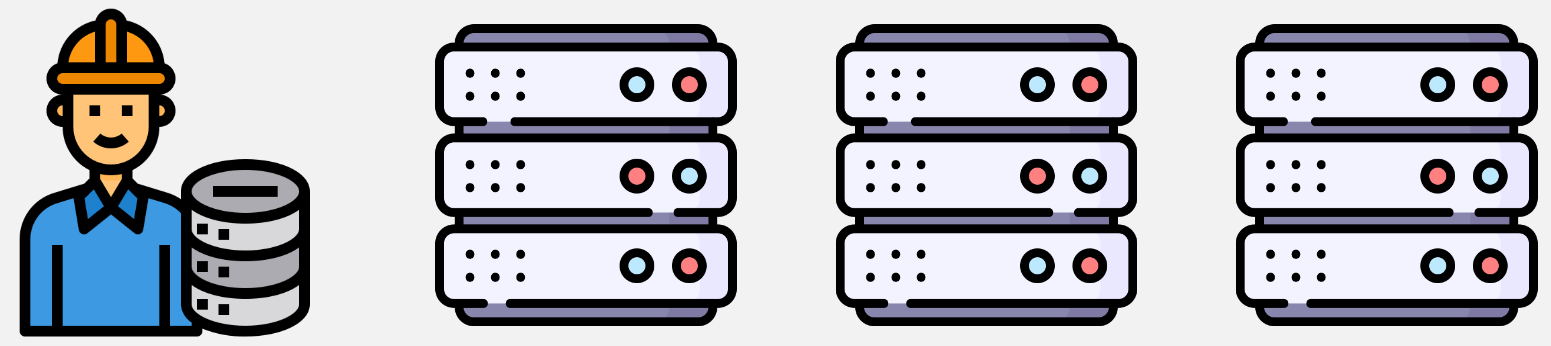
Traditional

```
0110
1001
1010
```

Optimization Logic in Traditional Databases

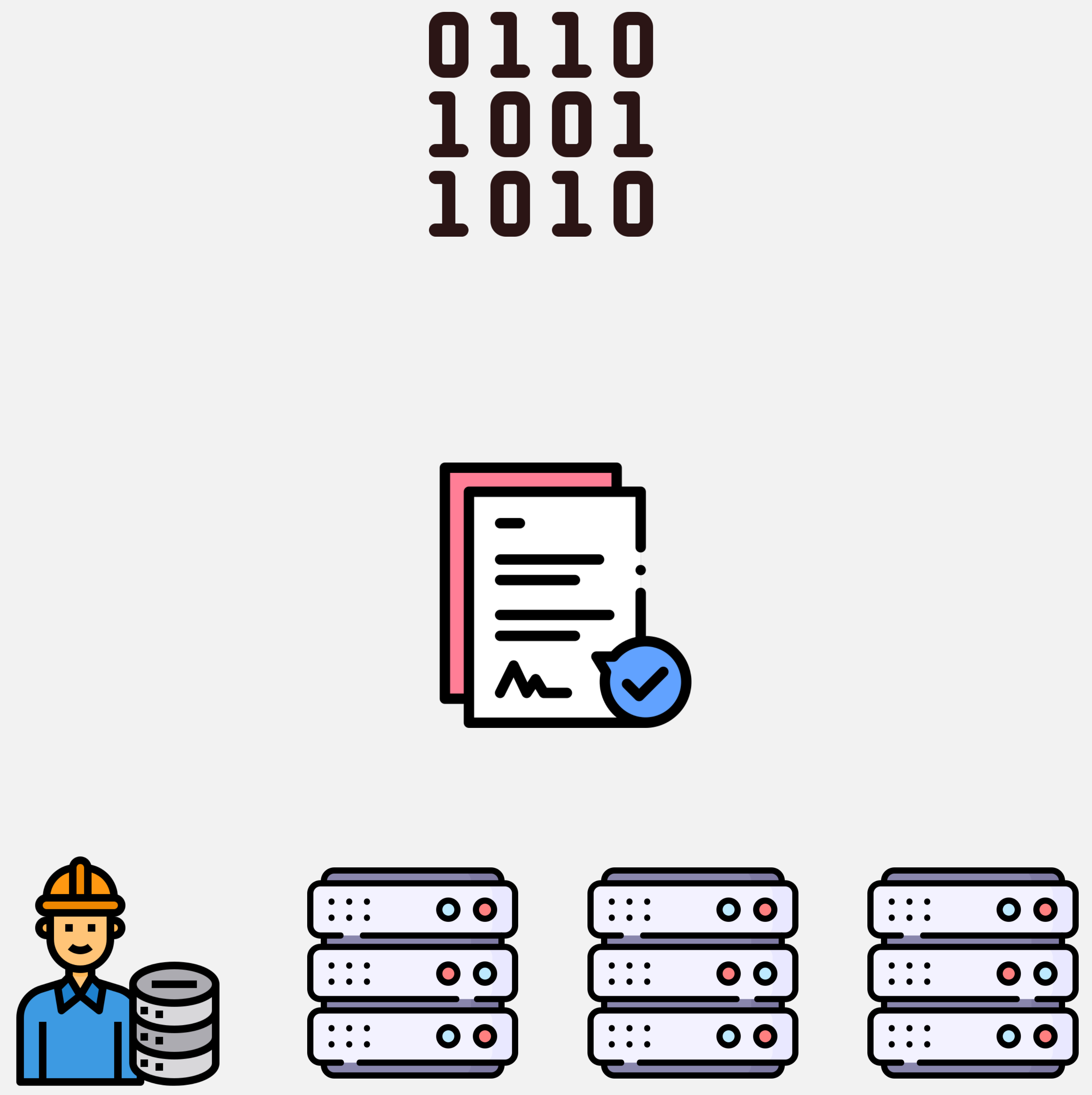
Traditional

```
0110
1001
1010
```



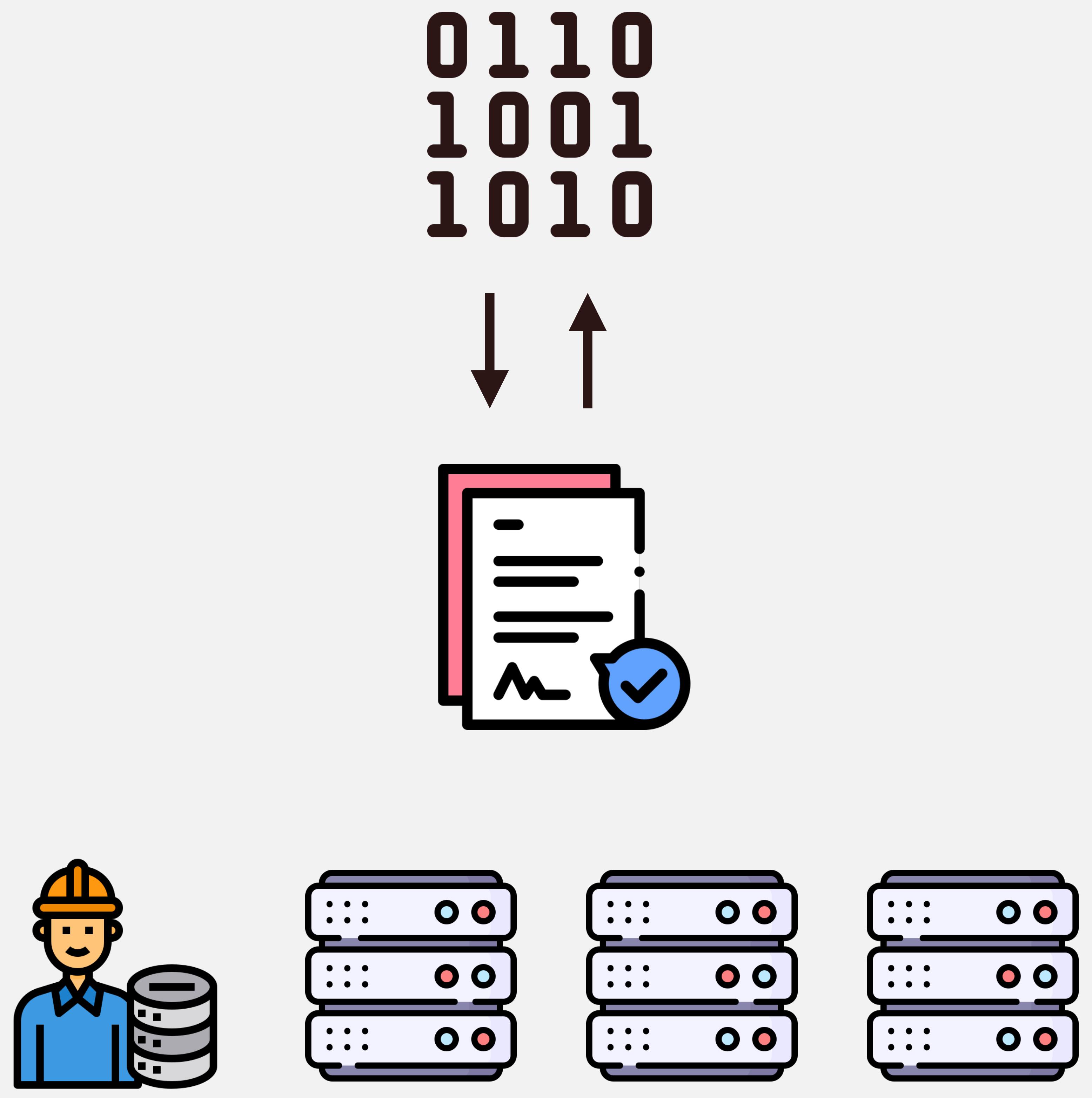
Optimization Logic in Traditional Databases

Traditional



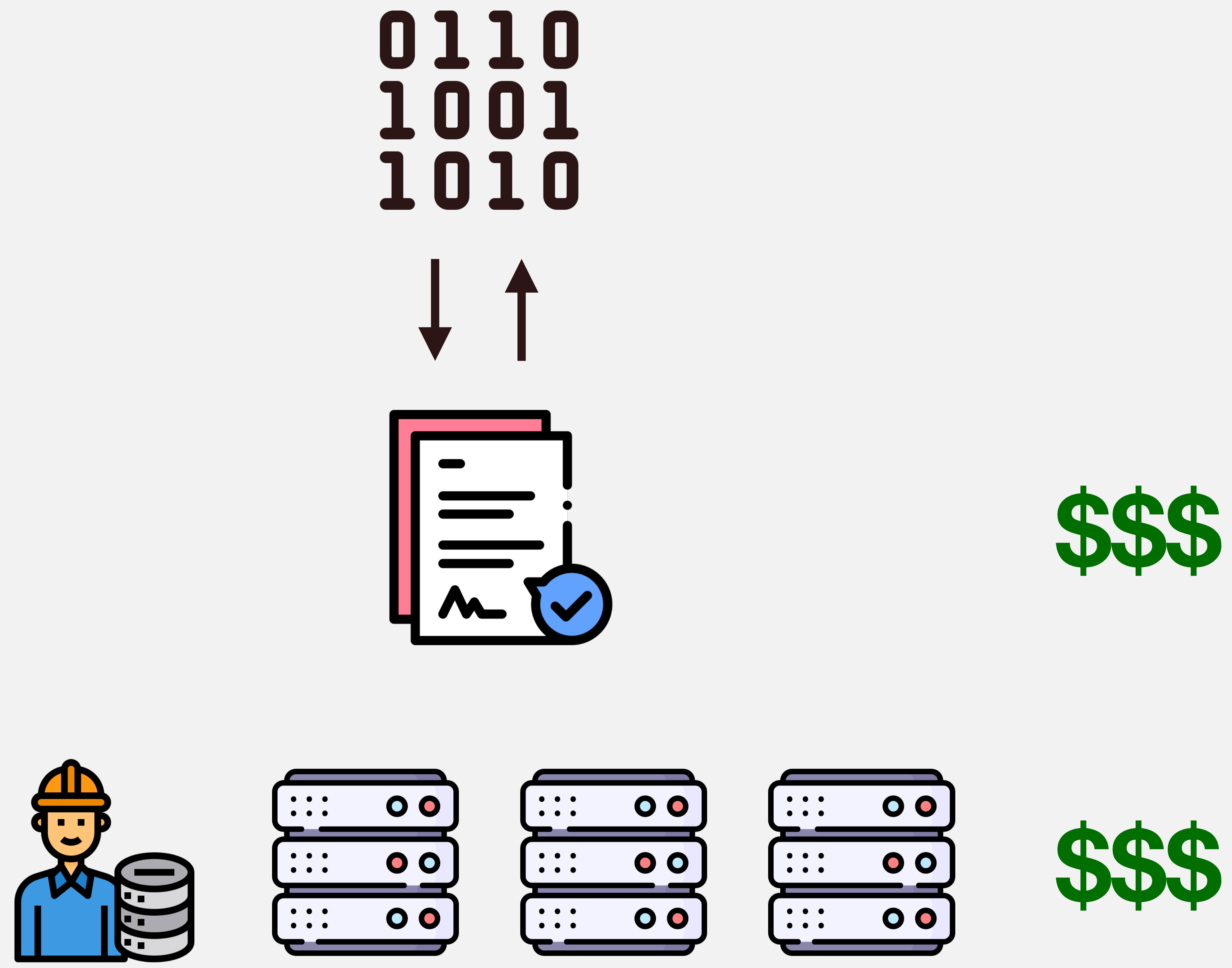
Optimization Logic in Traditional Databases

Traditional

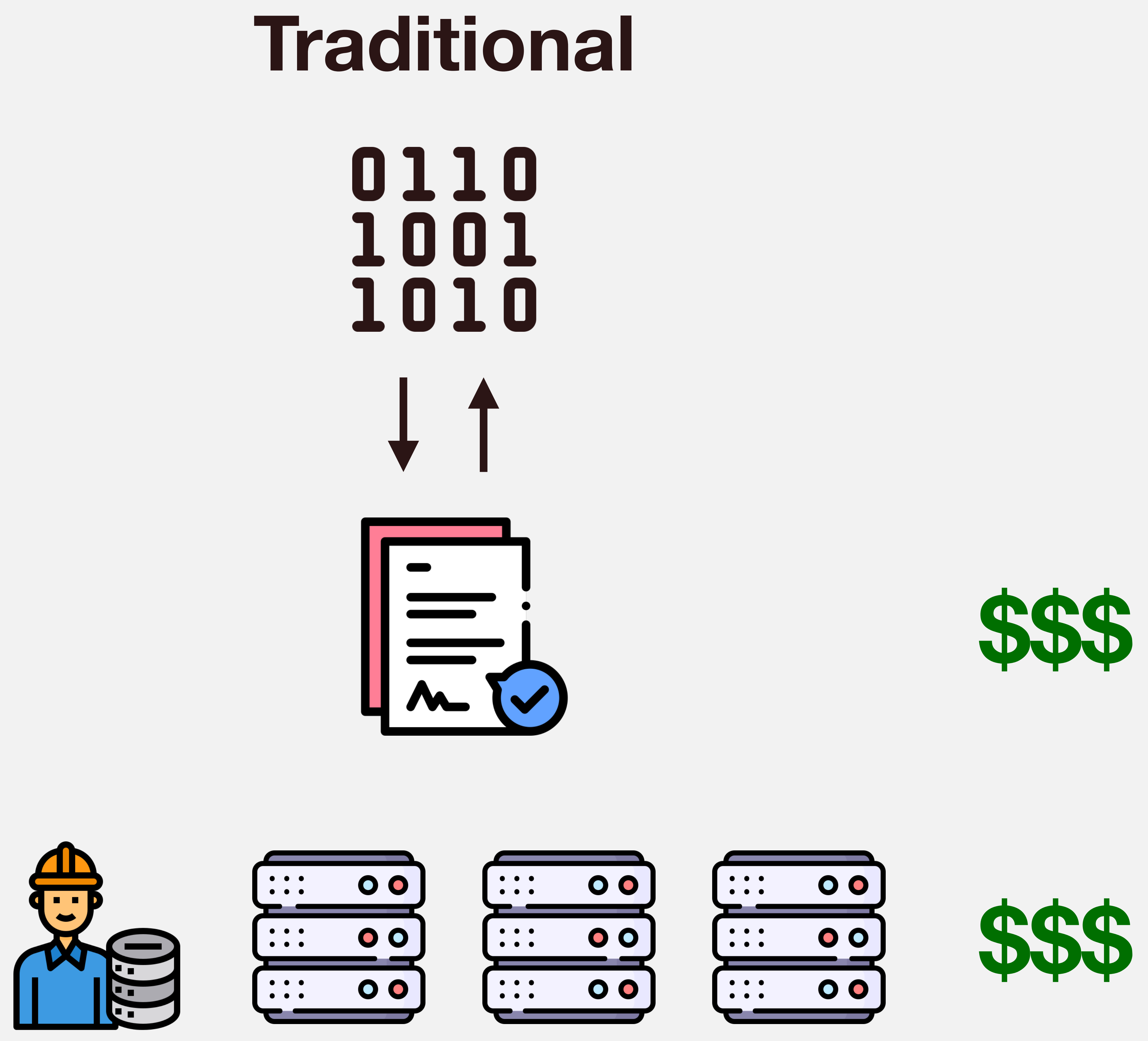


Optimization Logic in Traditional Databases

Traditional



Optimization Logic in Traditional Databases

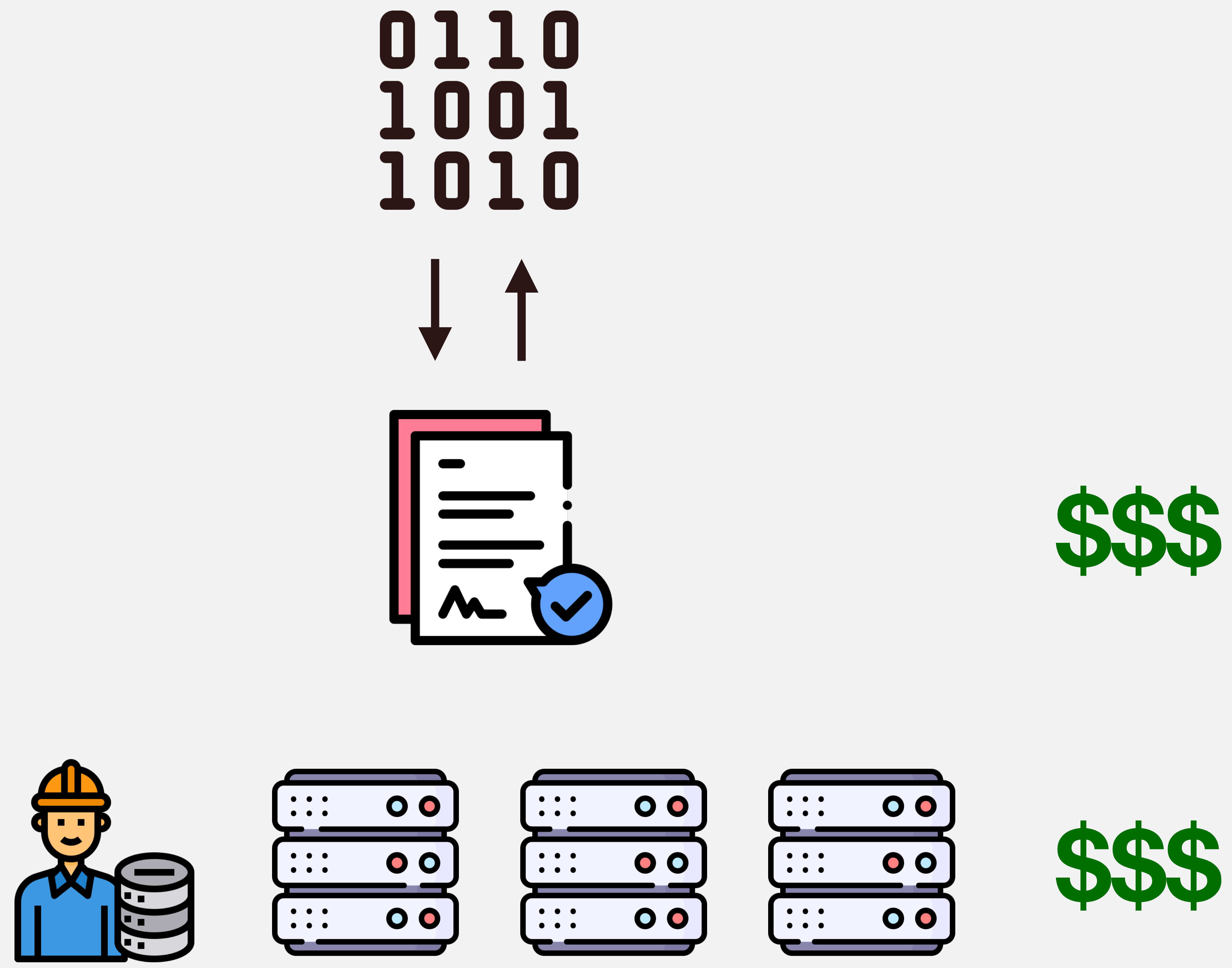


User Profit Utility $ Cost

$$\Pi = U(p) - C$$

Optimization Logic in Traditional Databases

Traditional



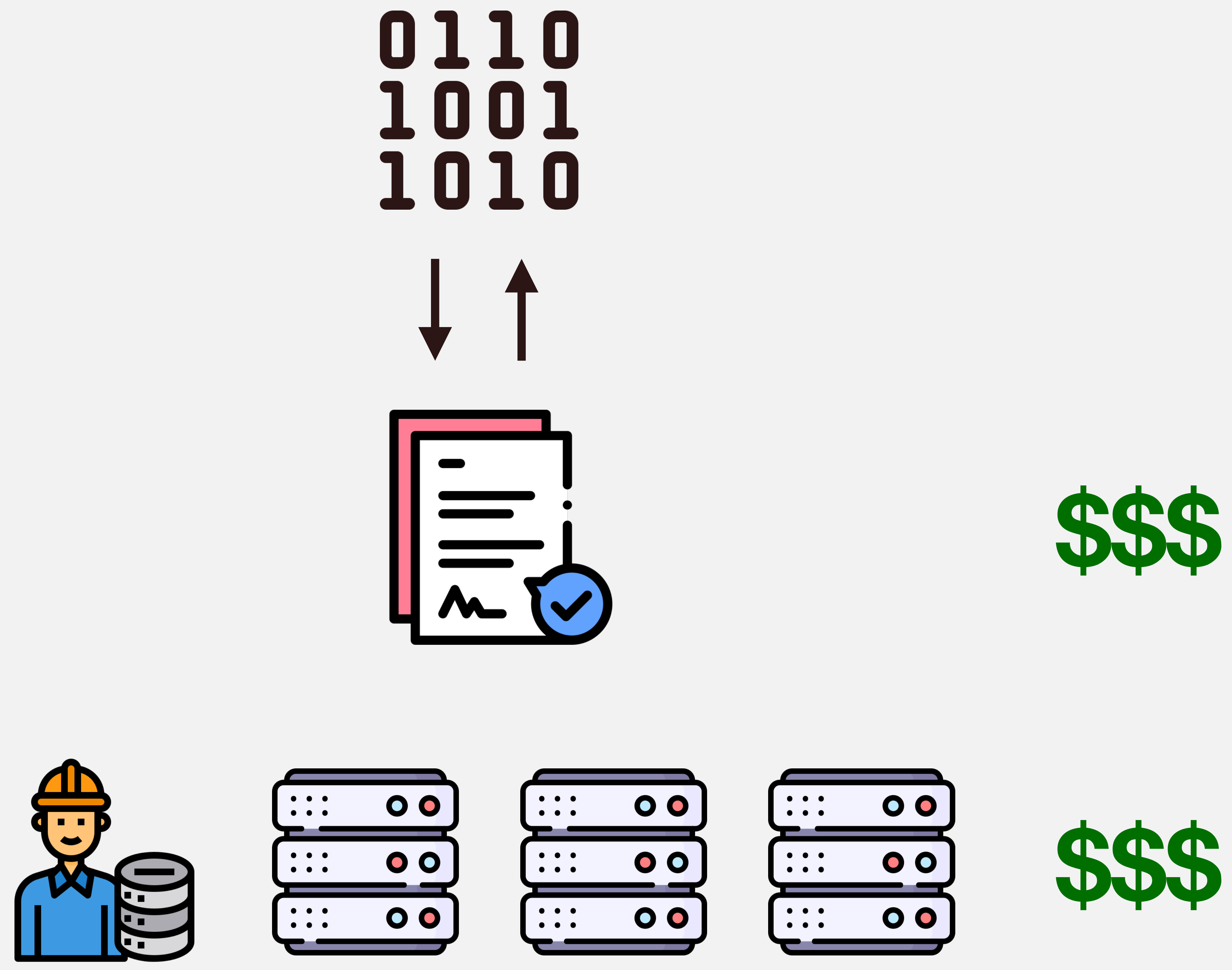
User Profit   Utility $ Cost

$$\Pi = U(p) - C$$

$$C_{sunk} + \Delta C$$

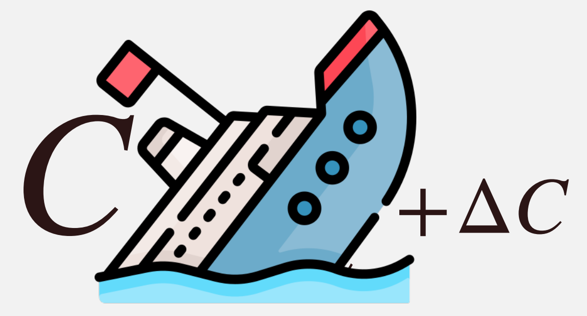
Optimization Logic in Traditional Databases

Traditional



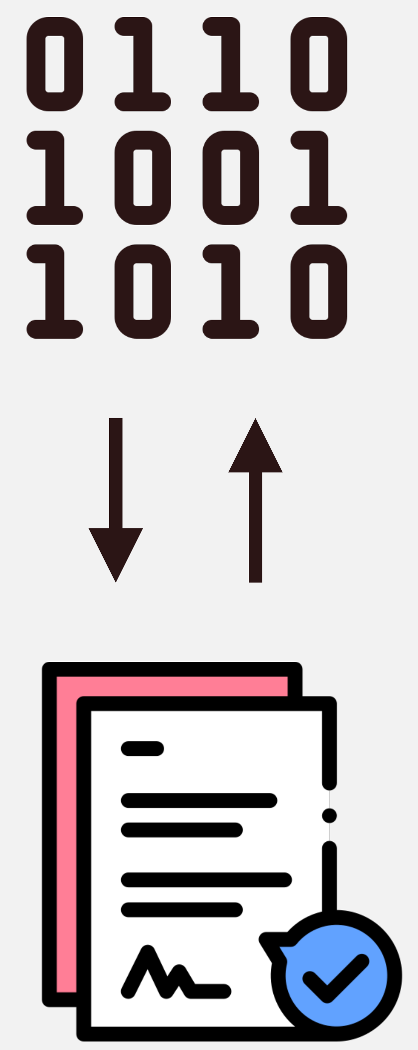
User Profit **Utility** **$ Cost**

$$\Pi = U(p) - C$$

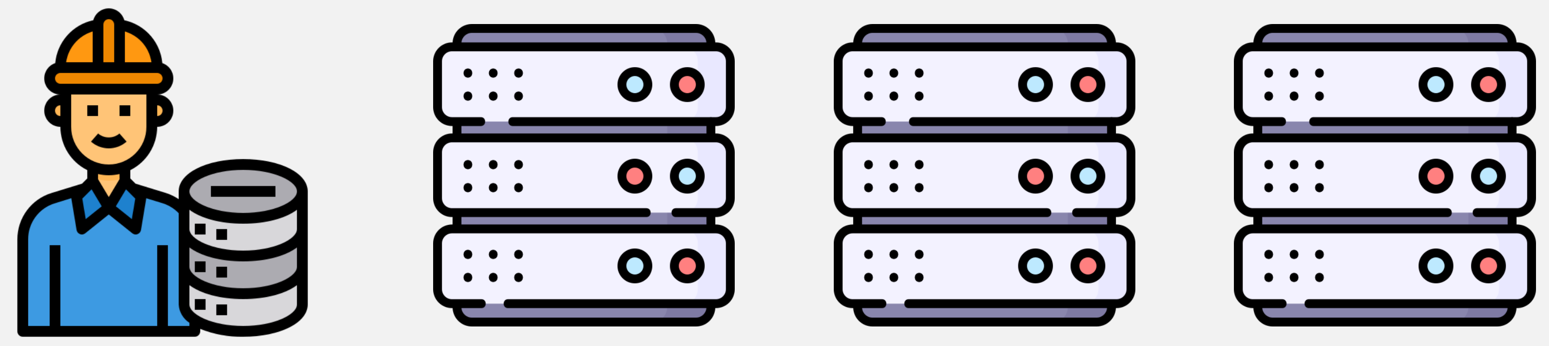


Optimization Logic in Traditional Databases

Traditional



$$$

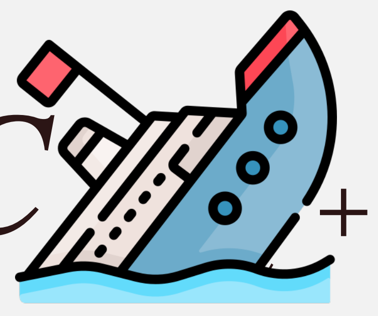


$$$

User Profit | Utility | $ Cost

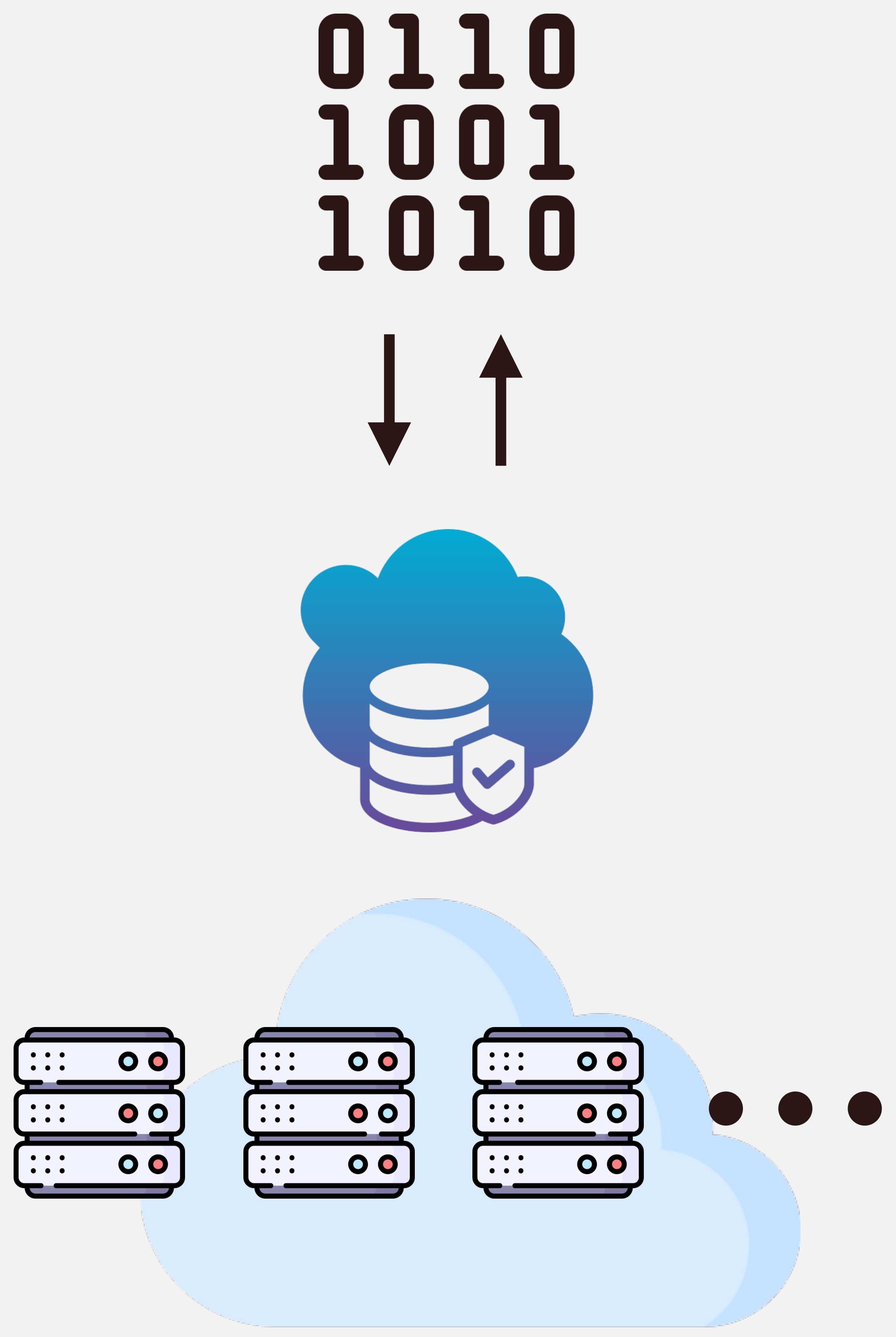
$$\boxed{\Pi} = U(p) - C$$

$$\boxed{p \uparrow} \quad C_{+\Delta C}$$



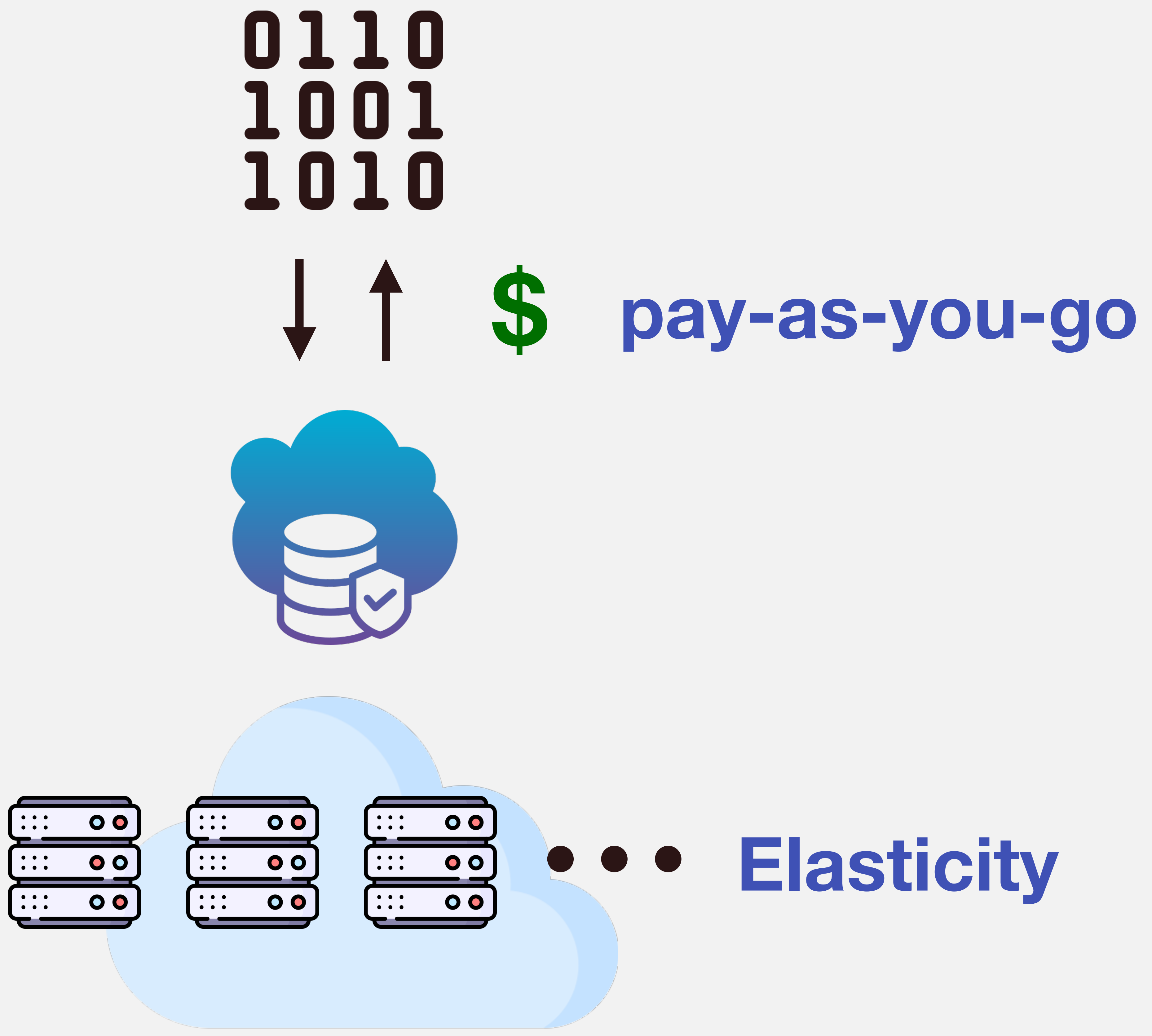
Database Optimization in the Cloud Era

Cloud-Native



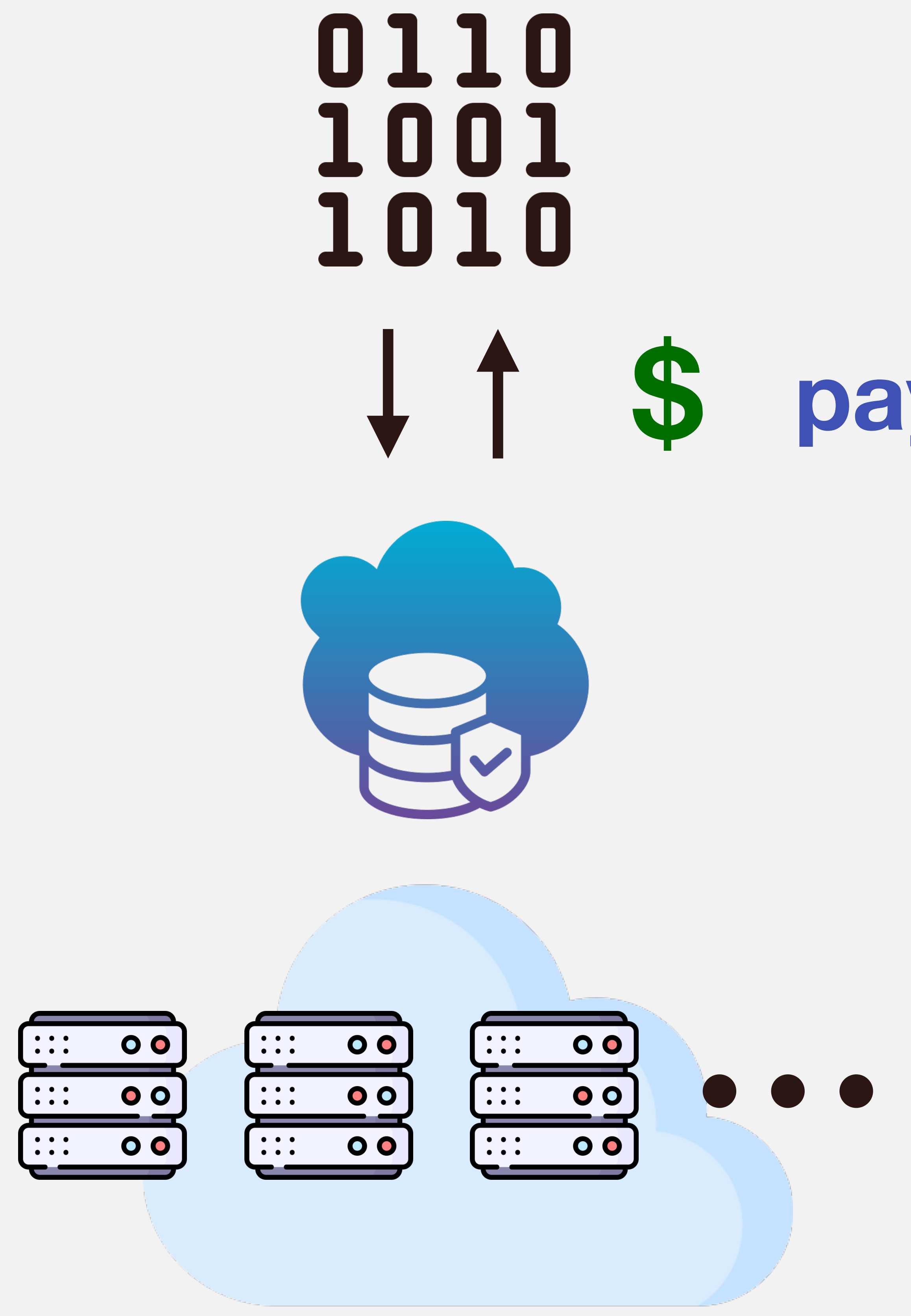
Database Optimization in the Cloud Era

Cloud-Native



Database Optimization in the Cloud Era

Cloud-Native



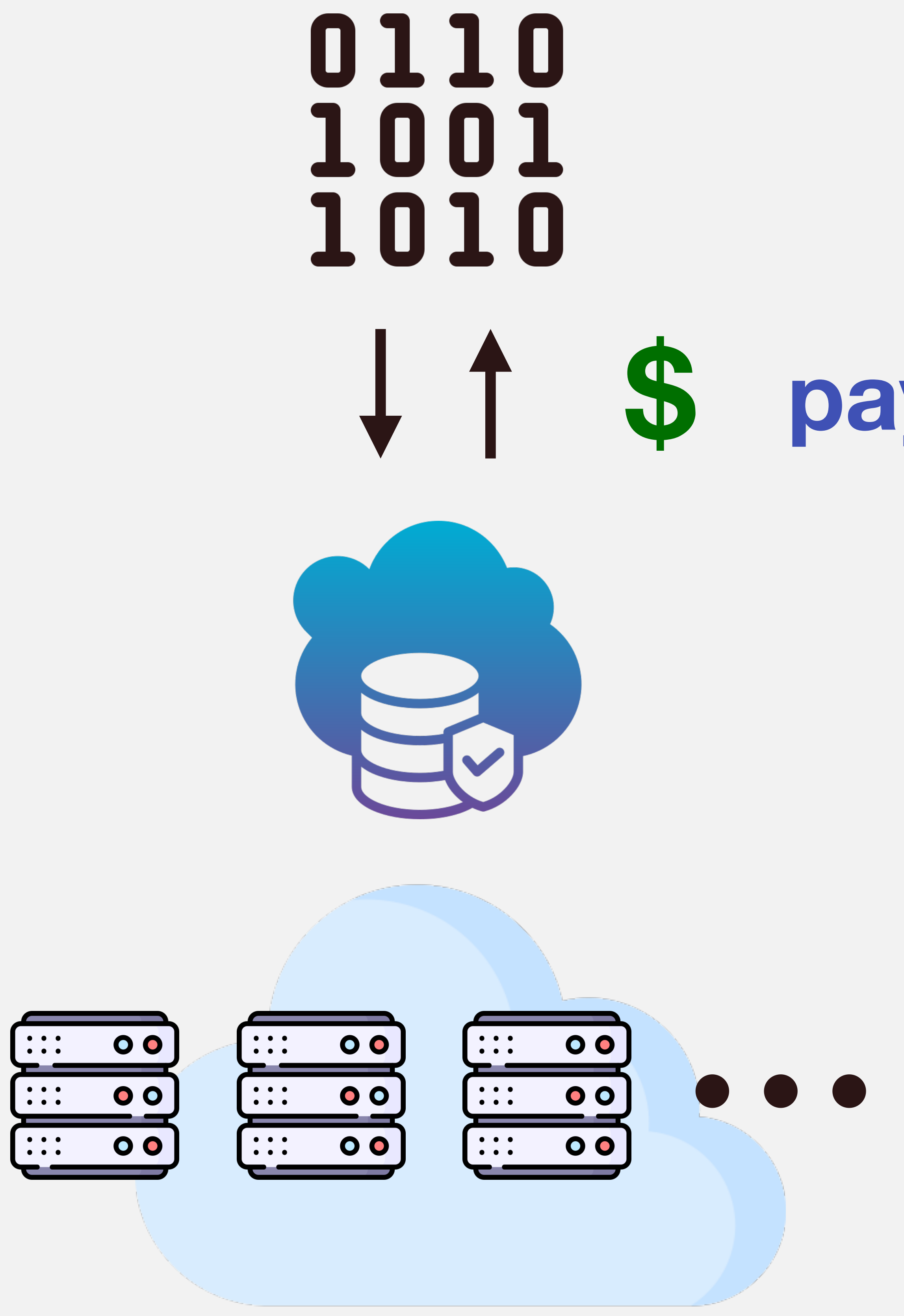
pay-as-you-go

Elasticity

User Profit		Utility		$ Cost
Π	$=$	$U(p)$	$-$	C
		p		ΔC

Database Optimization in the Cloud Era

Cloud-Native



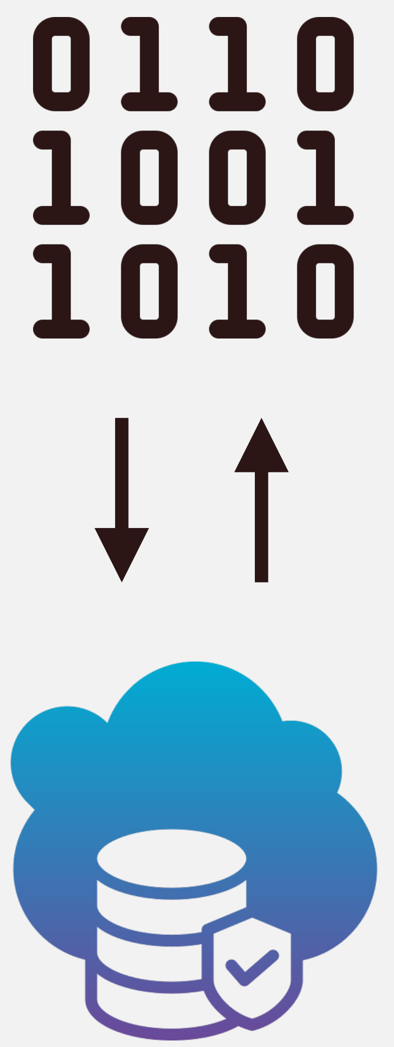
pay-as-you-go

Elasticity

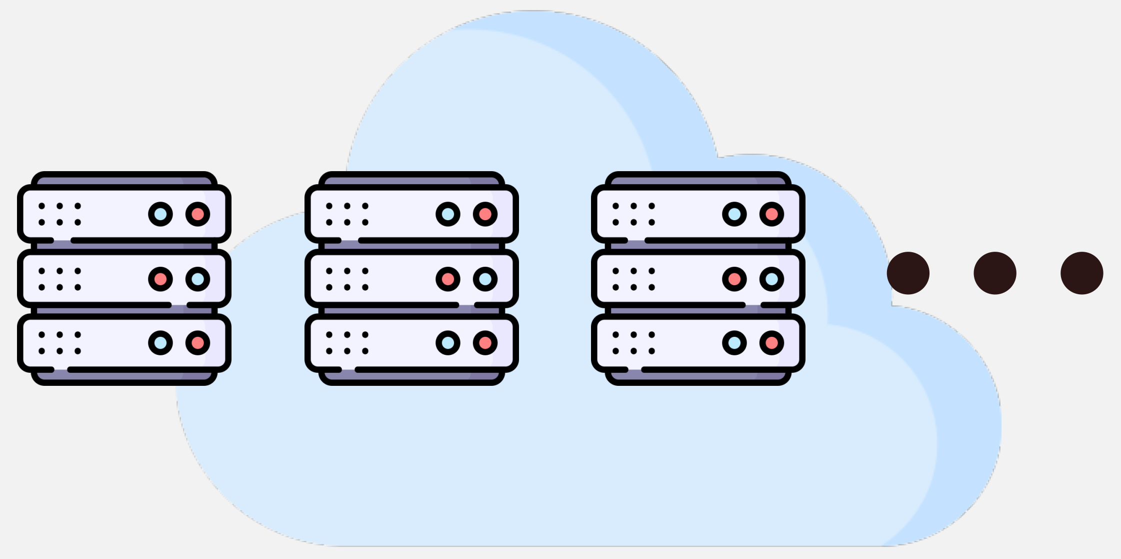
User Profit		Utility		$ Cost
Π	$=$	$U(p)$	$-$	C
		p		$\underline{\Delta C}$

Database Optimization in the Cloud Era

Cloud-Native



$ pay-as-you-go



Elasticity

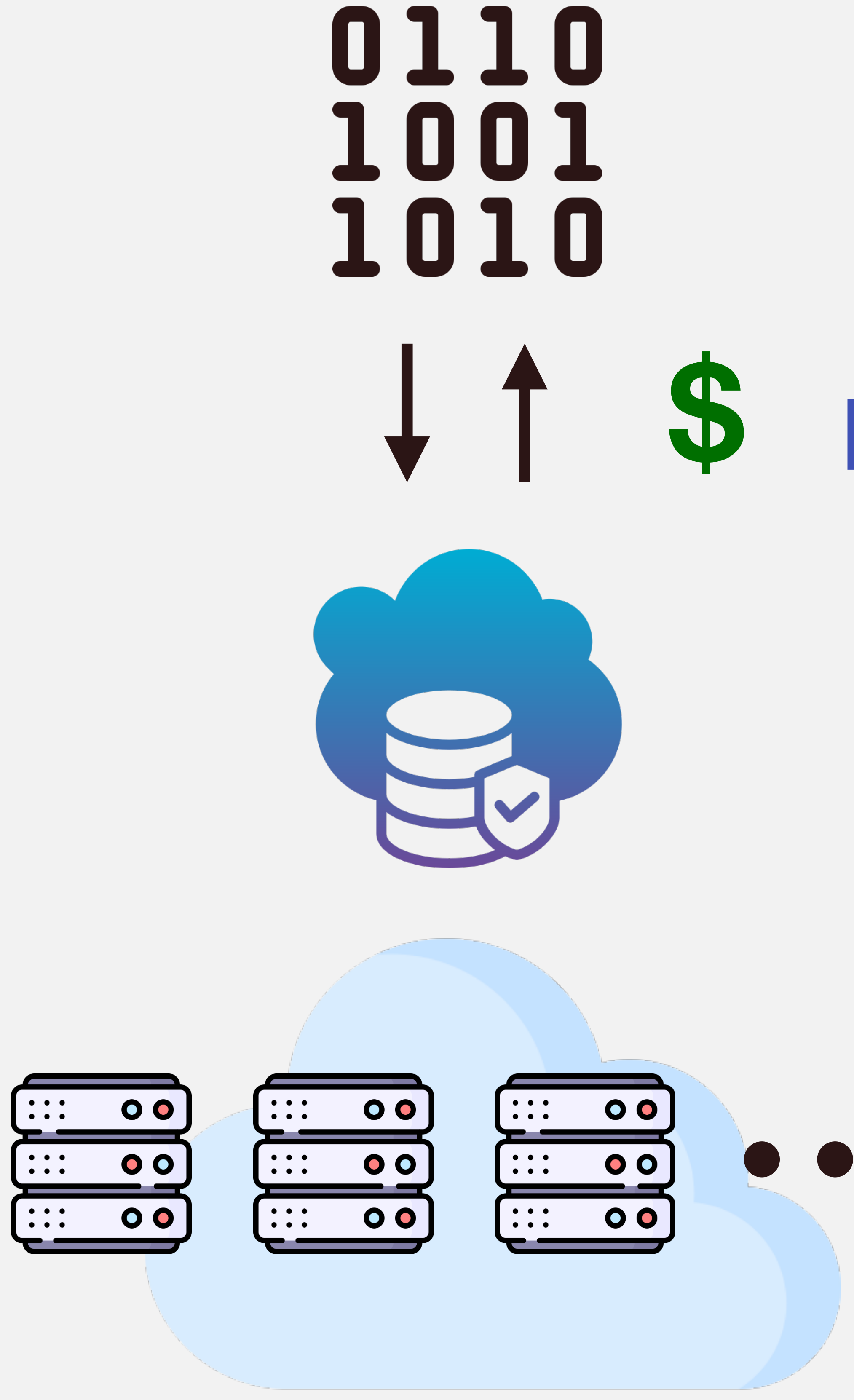
User Profit **Utility** **$ Cost**

$$\Pi = U(p) - C$$

$$\underbrace{p \quad \Delta C}_{\text{Bi-Objective Optimization}}$$

Database Optimization in the Cloud Era

Cloud-Native

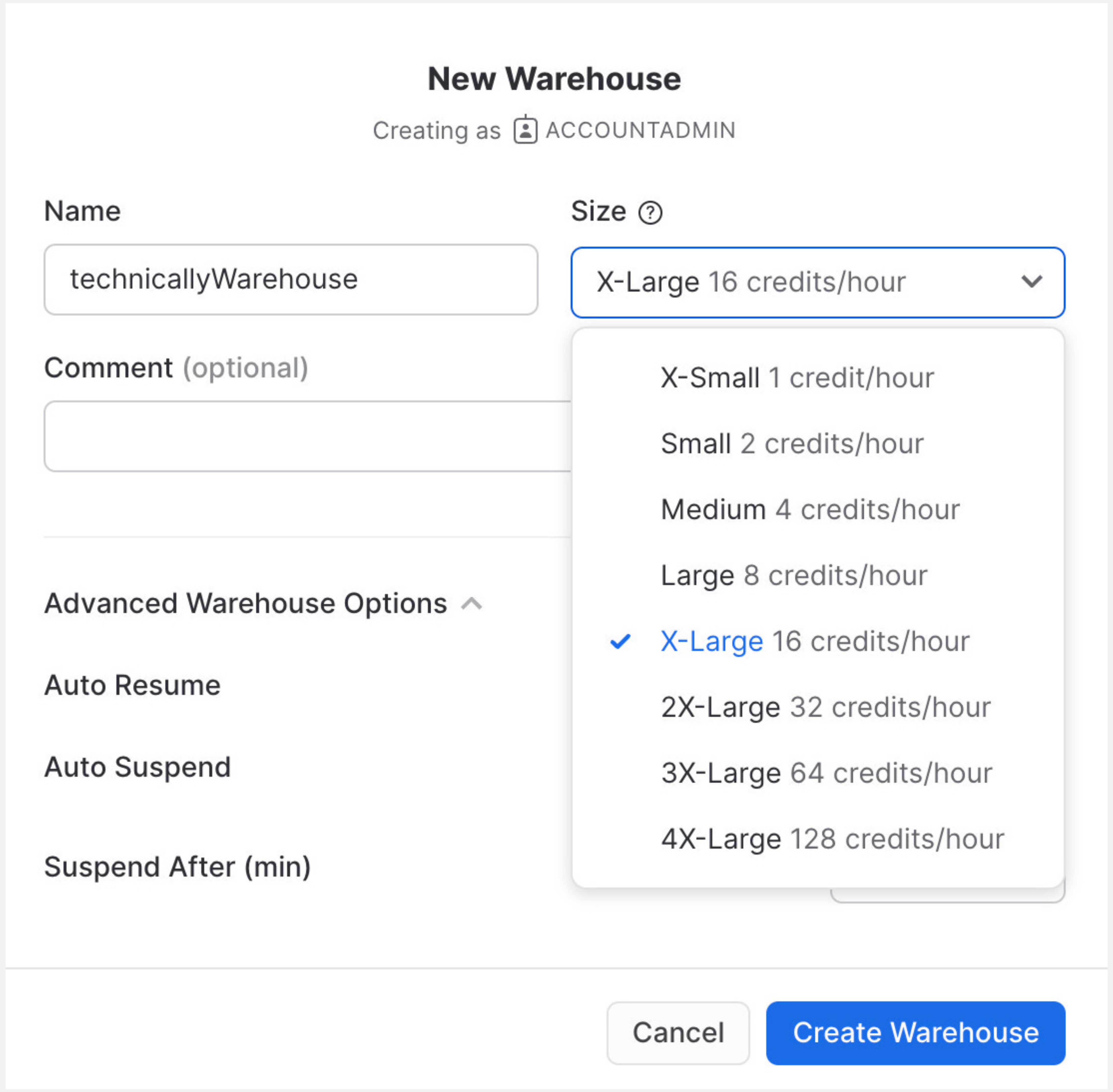


User Profit		Utility		$ Cost
Π	$=$	$U(p)$	$-$	C

$$\underbrace{p \qquad \Delta C}_{\text{Bi-Objective Optimization}}$$

Make cost a first class citizen

Cost Control Is Still Difficult



New Warehouse

Creating as ACCOUNTADMIN

Name: technicallyWarehouse

Size: X-Large 16 credits/hour

- X-Small 1 credit/hour
- Small 2 credits/hour
- Medium 4 credits/hour
- Large 8 credits/hour
- ✓ X-Large 16 credits/hour
- 2X-Large 32 credits/hour
- 3X-Large 64 credits/hour
- 4X-Large 128 credits/hour

Comment (optional)

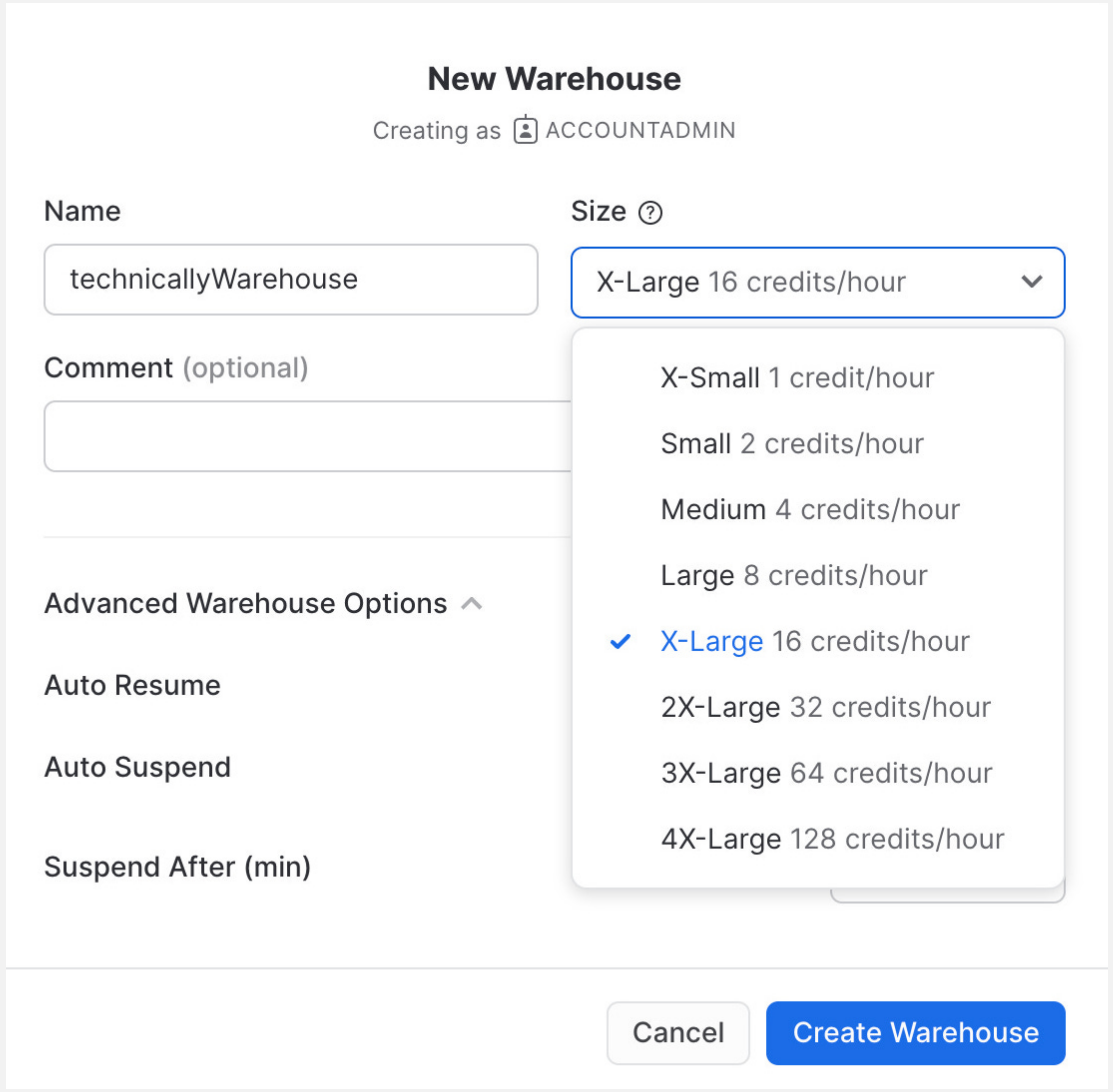
Advanced Warehouse Options

Auto Resume

Auto Suspend

Suspend After (min)

Cost Control Is Still Difficult



New Warehouse

Creating as ACCOUNTADMIN

Name: technicallyWarehouse

Size: X-Large 16 credits/hour

- X-Small 1 credit/hour
- Small 2 credits/hour
- Medium 4 credits/hour
- Large 8 credits/hour
- ✓ X-Large 16 credits/hour
- 2X-Large 32 credits/hour
- 3X-Large 64 credits/hour
- 4X-Large 128 credits/hour

Comment (optional)

Advanced Warehouse Options

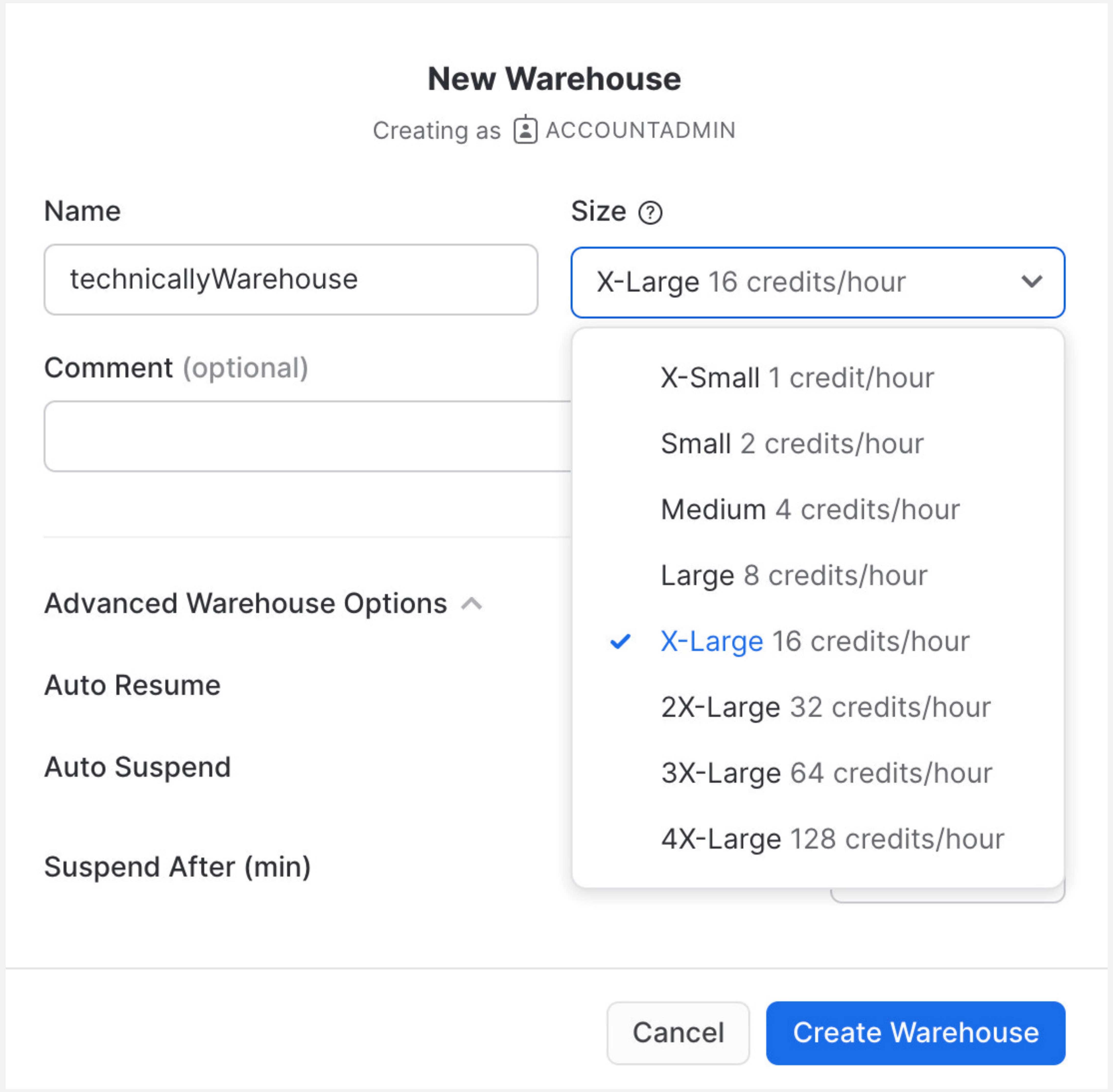
Auto Resume

Auto Suspend

Suspend After (min)

→ **Users tend to over-provision**

Cost Control Is Still Difficult



New Warehouse

Creating as ACCOUNTADMIN

Name: technicallyWarehouse

Size: X-Large 16 credits/hour

Comment (optional)

Advanced Warehouse Options

Auto Resume

Auto Suspend

Suspend After (min)

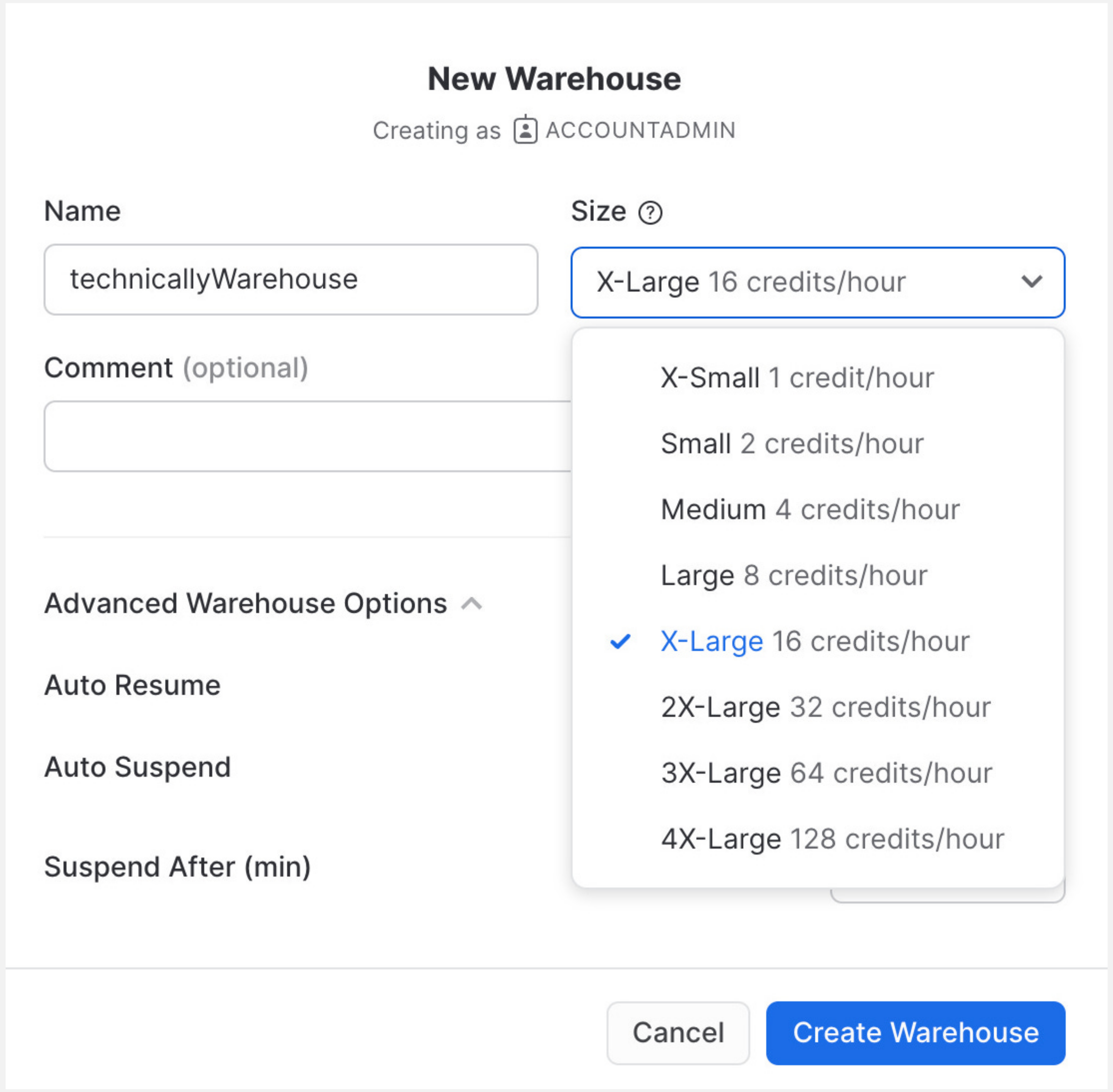
- X-Small 1 credit/hour
- Small 2 credits/hour
- Medium 4 credits/hour
- Large 8 credits/hour
- ✓ X-Large 16 credits/hour
- 2X-Large 32 credits/hour
- 3X-Large 64 credits/hour
- 4X-Large 128 credits/hour

Cancel | Create Warehouse

→ **Users tend to over-provision**

→ **Fixed cluster size over the entire workload**

Cost Control Is Still Difficult


New Warehouse

Creating as ACCOUNTADMIN

Name: technicallyWarehouse

Size: X-Large 16 credits/hour

Comment (optional)

- X-Small 1 credit/hour
- Small 2 credits/hour
- Medium 4 credits/hour
- Large 8 credits/hour
- ✓ X-Large 16 credits/hour
- 2X-Large 32 credits/hour
- 3X-Large 64 credits/hour
- 4X-Large 128 credits/hour

Advanced Warehouse Options

Auto Resume

Auto Suspend

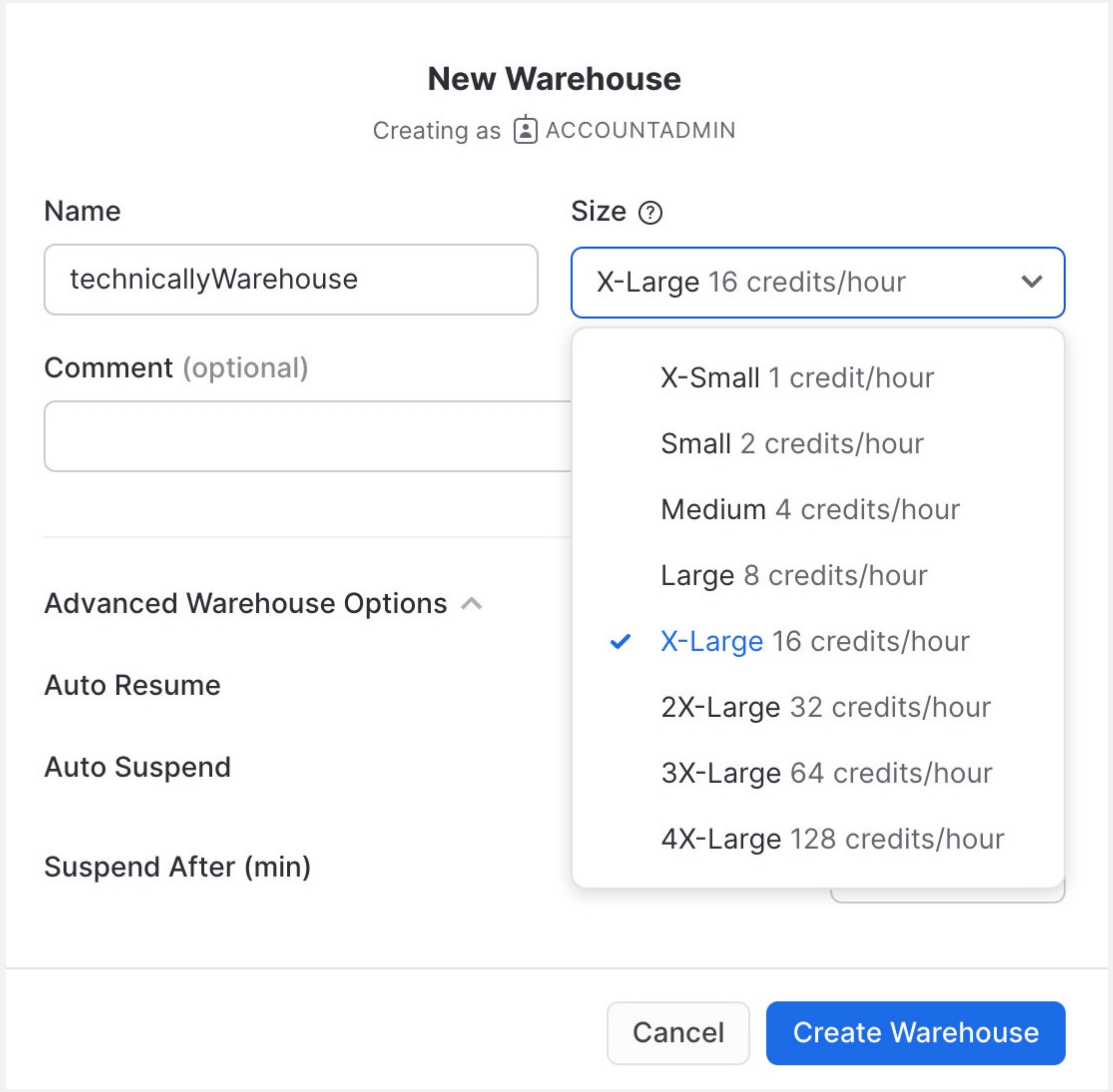
Suspend After (min)

→ **Users tend to over-provision**

→ **Fixed cluster size over the entire workload**

Resource Waste!

Cost Control Is Still Difficult



New Warehouse

Creating as ACCOUNTADMIN

Name: technicallyWarehouse

Size: X-Large 16 credits/hour

Comment (optional)

- X-Small 1 credit/hour
- Small 2 credits/hour
- Medium 4 credits/hour
- Large 8 credits/hour
- ✓ X-Large 16 credits/hour
- 2X-Large 32 credits/hour
- 3X-Large 64 credits/hour
- 4X-Large 128 credits/hour

Advanced Warehouse Options

Auto Resume

Auto Suspend

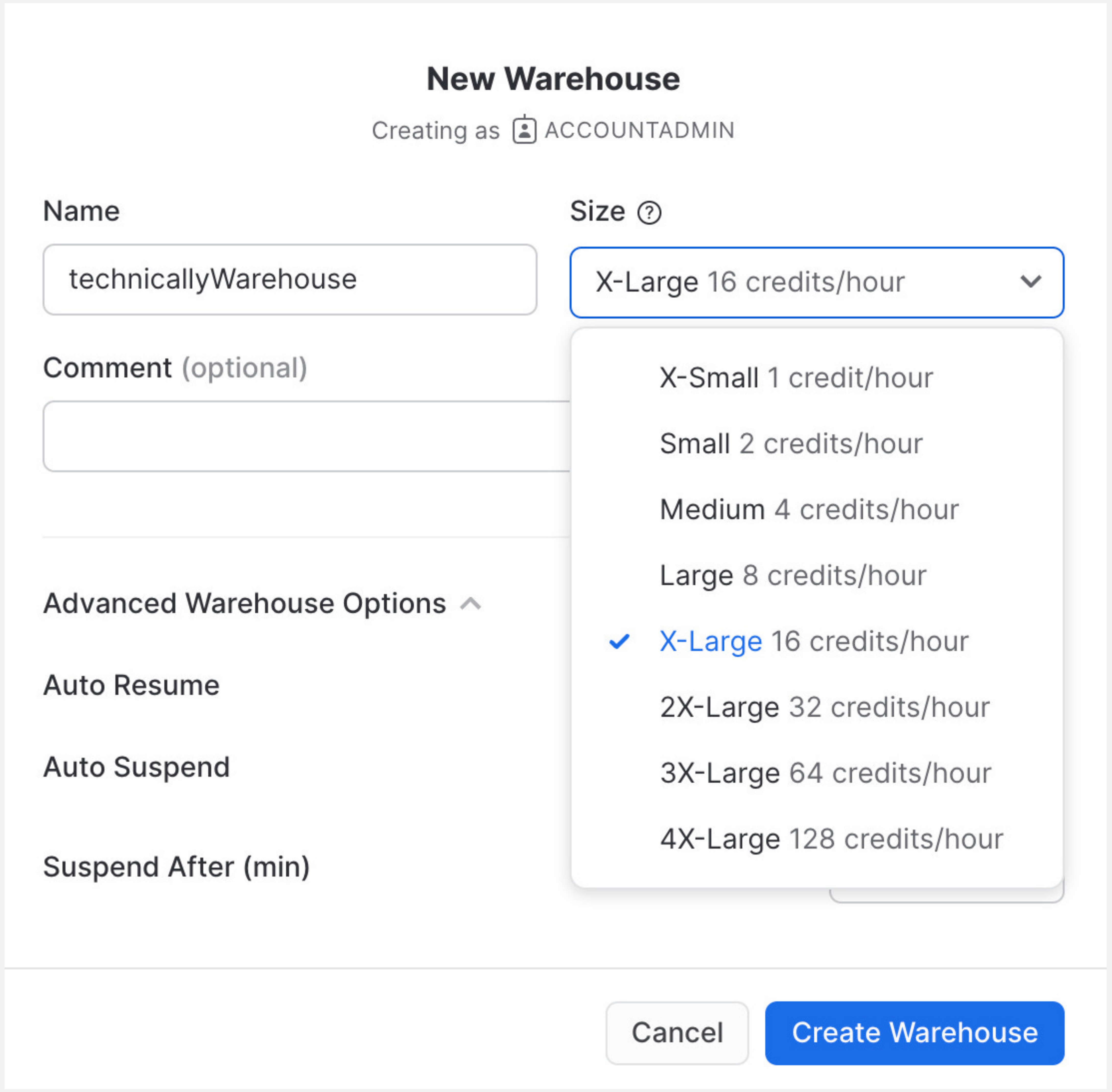
Suspend After (min)

- Build Indexes
- Build Materialized Views
- Re-partition Data
- Re-train a Learned Module

DBA

$$$

Cost Control Is Still Difficult



New Warehouse

Creating as ACCOUNTADMIN

Name: technicallyWarehouse

Size: X-Large 16 credits/hour

Comment (optional)

- X-Small 1 credit/hour
- Small 2 credits/hour
- Medium 4 credits/hour
- Large 8 credits/hour
- ✓ X-Large 16 credits/hour
- 2X-Large 32 credits/hour
- 3X-Large 64 credits/hour
- 4X-Large 128 credits/hour

Advanced Warehouse Options

Auto Resume

Auto Suspend

Suspend After (min)

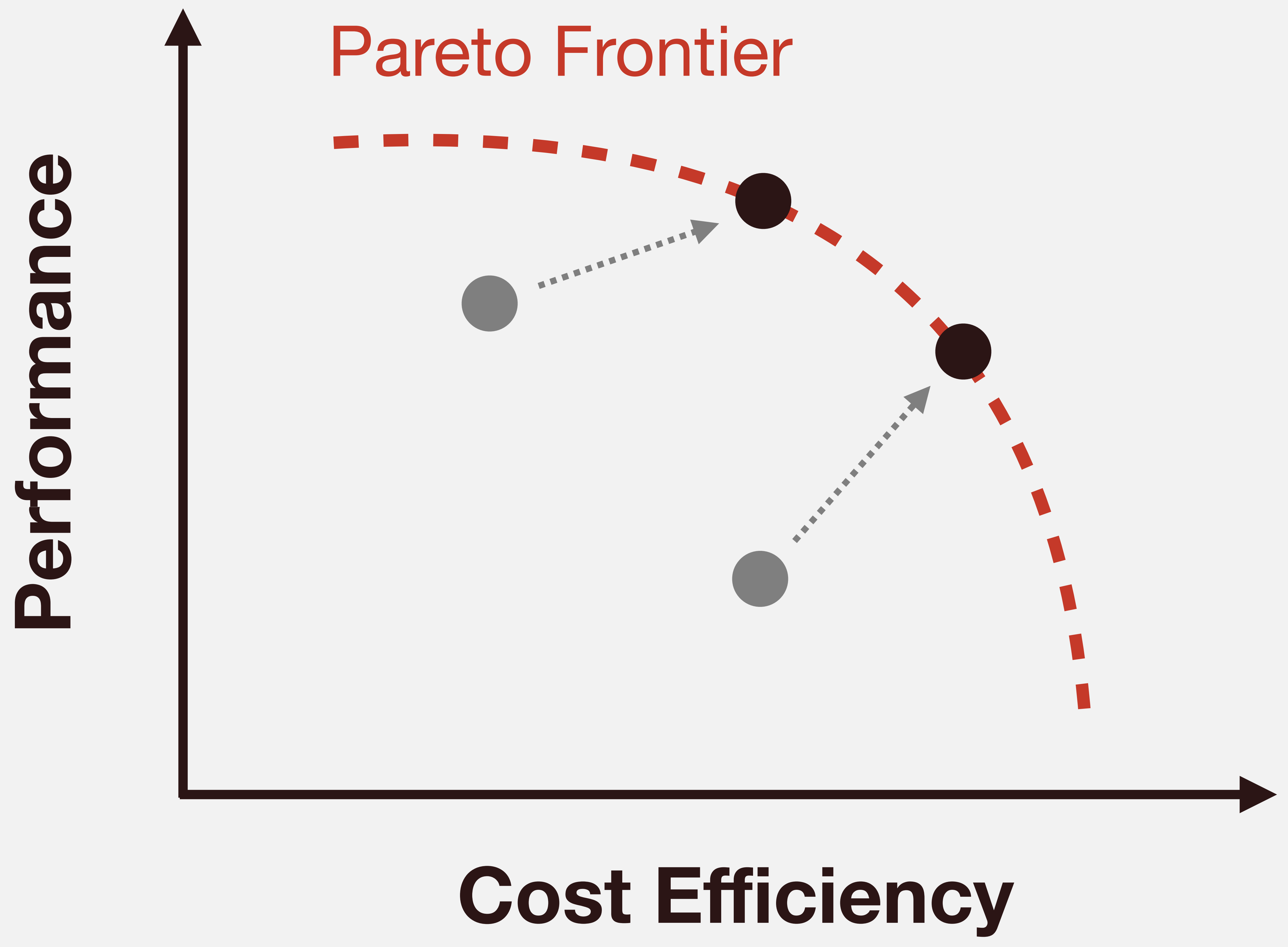
- Build Indexes
- Build Materialized Views
- Re-partition Data
- Re-train a Learned Module

DBA

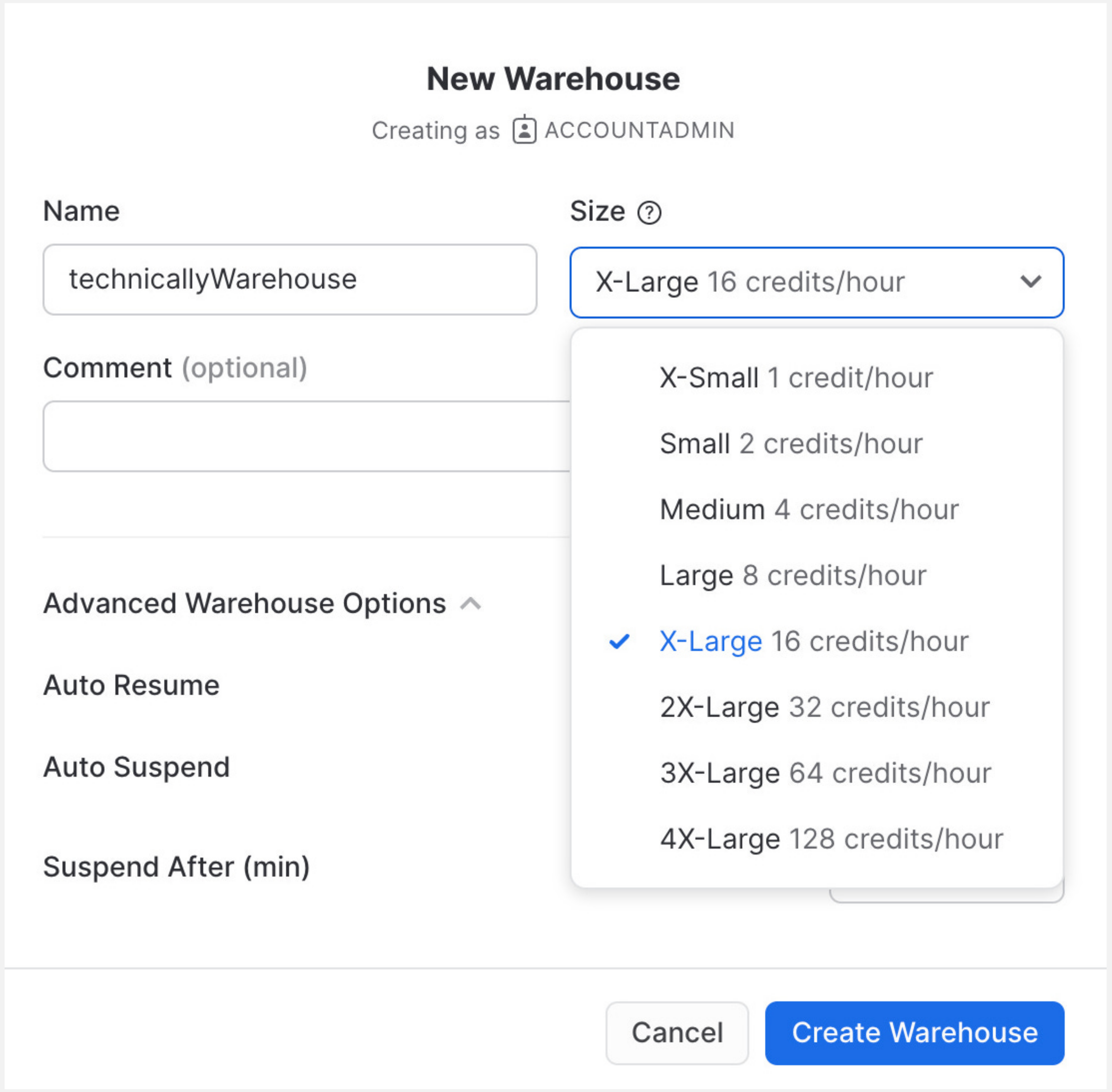
$$$

Cost Intelligence

The system's ability to **self-adapt** to stay **Pareto Optimal** in the performance-cost trade-off under different workloads and user constraints.



An Ideal UI



New Warehouse

Creating as ACCOUNTADMIN

Name: technicallyWarehouse

Size: X-Large 16 credits/hour

- X-Small 1 credit/hour
- Small 2 credits/hour
- Medium 4 credits/hour
- Large 8 credits/hour
- ✓ X-Large 16 credits/hour
- 2X-Large 32 credits/hour
- 3X-Large 64 credits/hour
- 4X-Large 128 credits/hour

Comment (optional)

Advanced Warehouse Options

Auto Resume

Auto Suspend

Suspend After (min)

- Build Indexes
- Build Materialized Views
- Re-partition Data
- Re-train a Learned Module

DBA

$$$

An Ideal UI

Workload SQL

Time: 10s ——— 10min

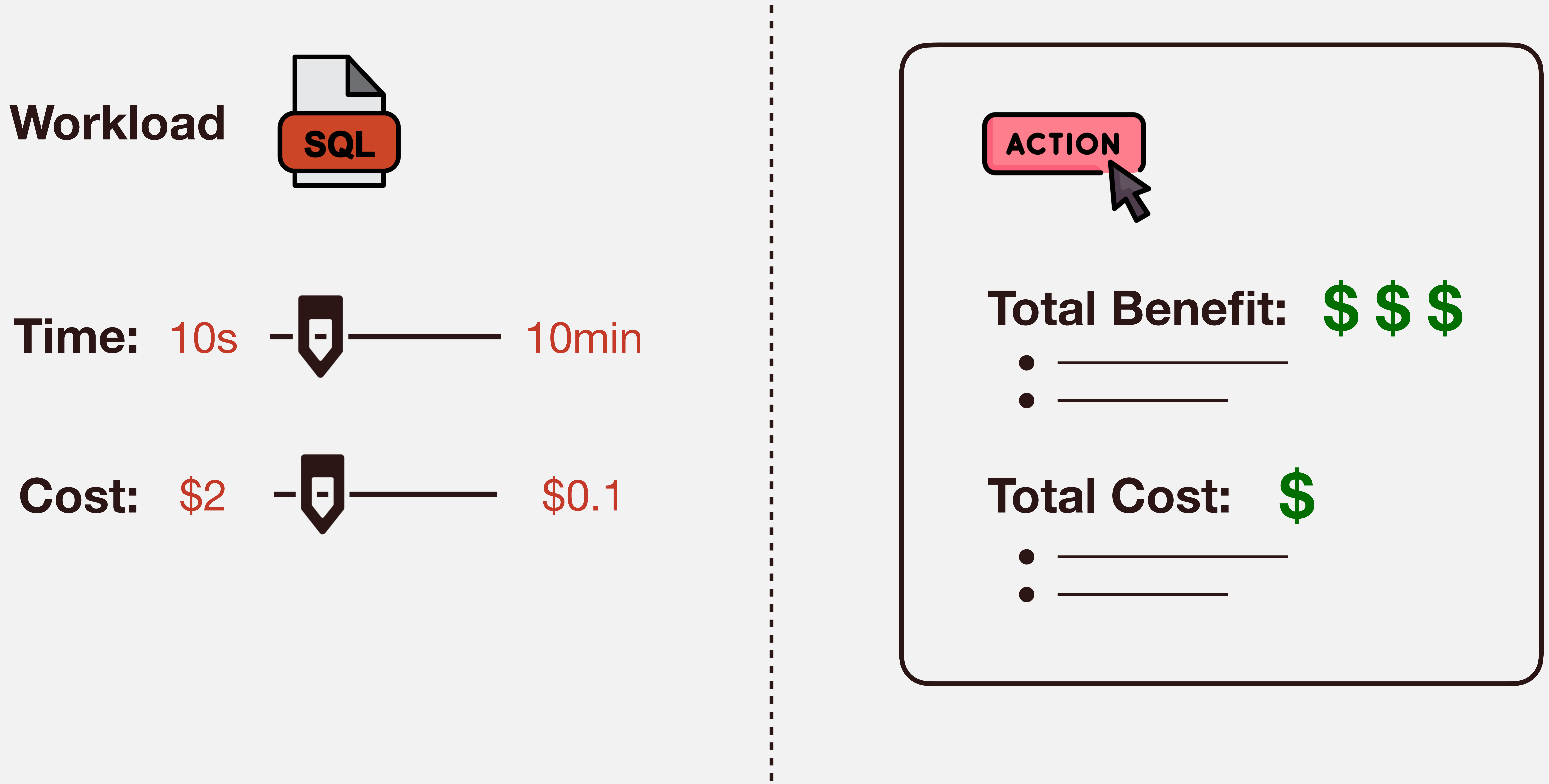
Cost: $2 ——— $0.1

- Build Indexes
- Build Materialized Views
- Re-partition Data
- Re-train a Learned Module

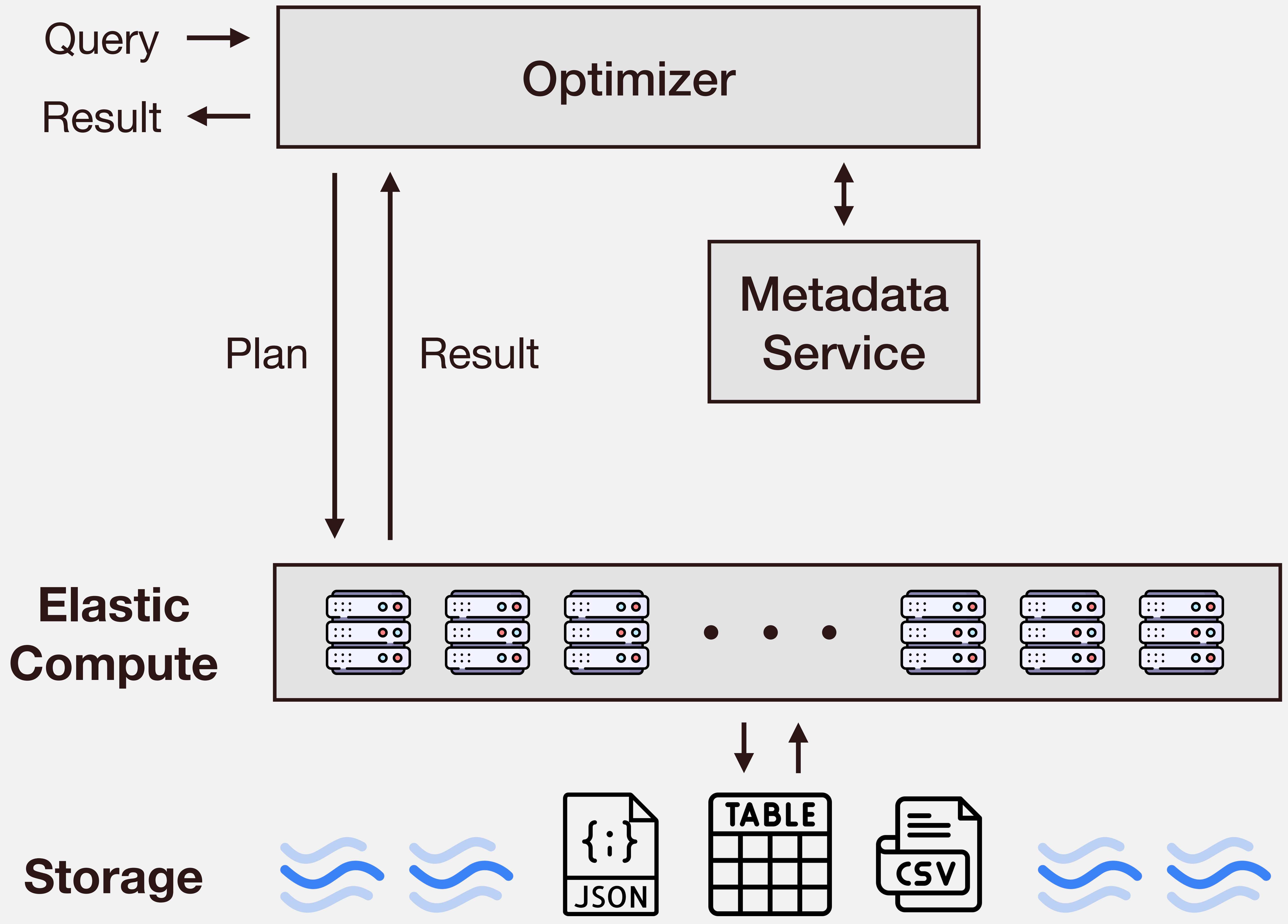
DBA

$$$

An Ideal UI


Workload
SQL
Time: 10s
10min
Cost: $2
$0.1
ACTION
Total Benefit: $ $ $
Total Cost: $

Base System Architecture



Automatic Resource Deployment

Workload

SQL

Automatic Resource Deployment

Workload SQL

Config 1 × **100 min**

Config 2 • • • × **1 min**

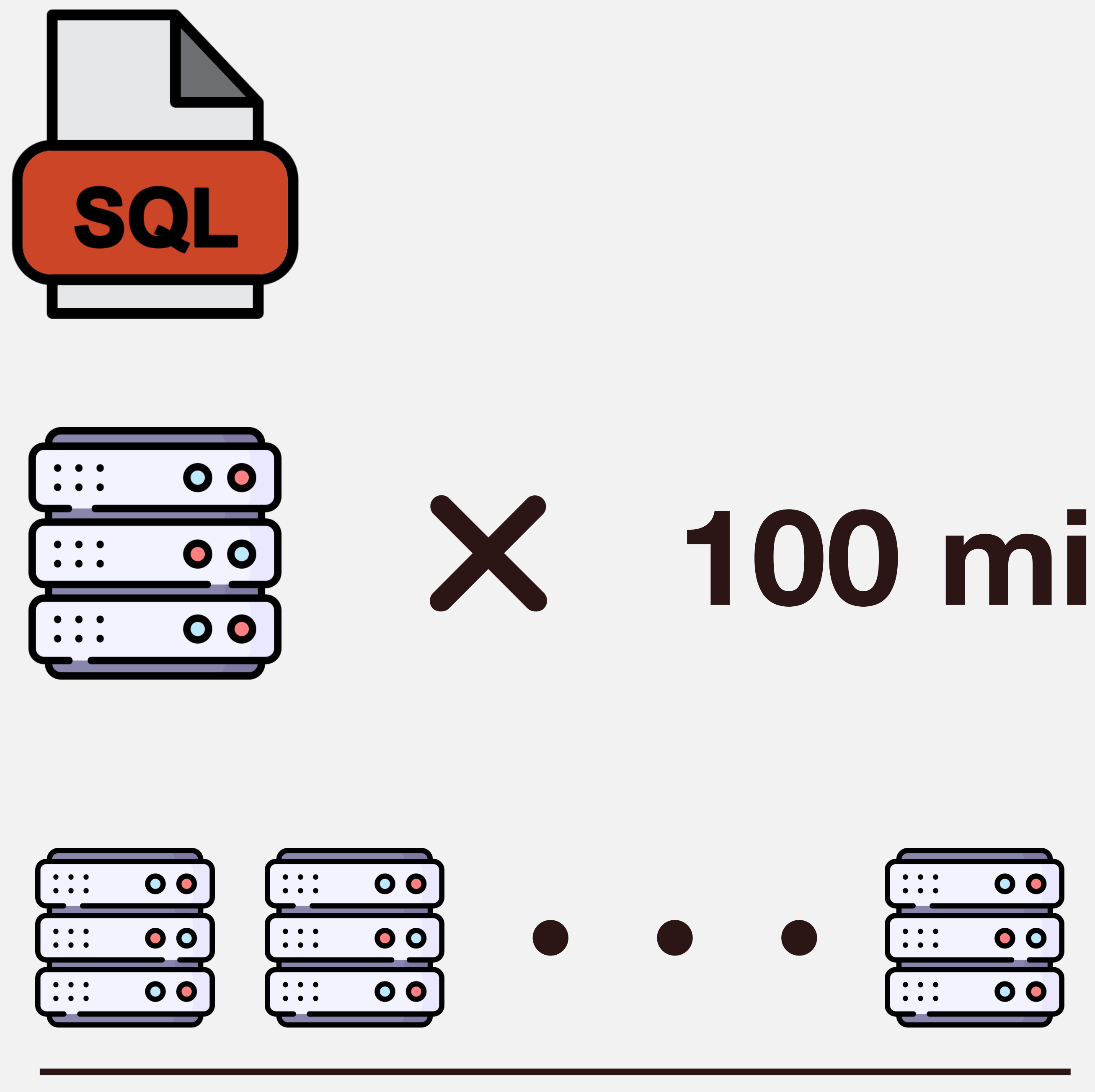
100 servers

Automatic Resource Deployment

Workload

Config 1 × 100 min

Config 2 × 1 min

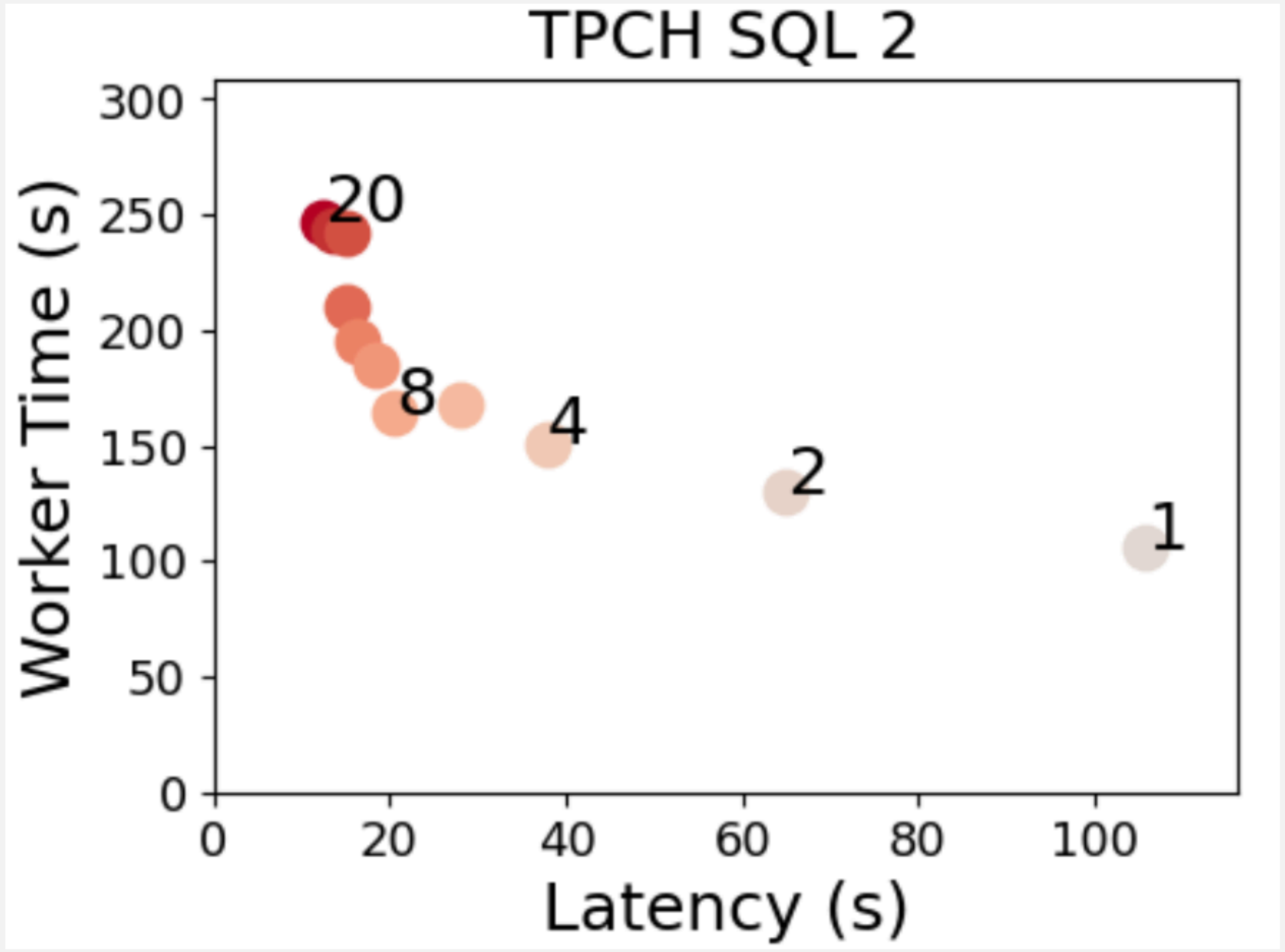


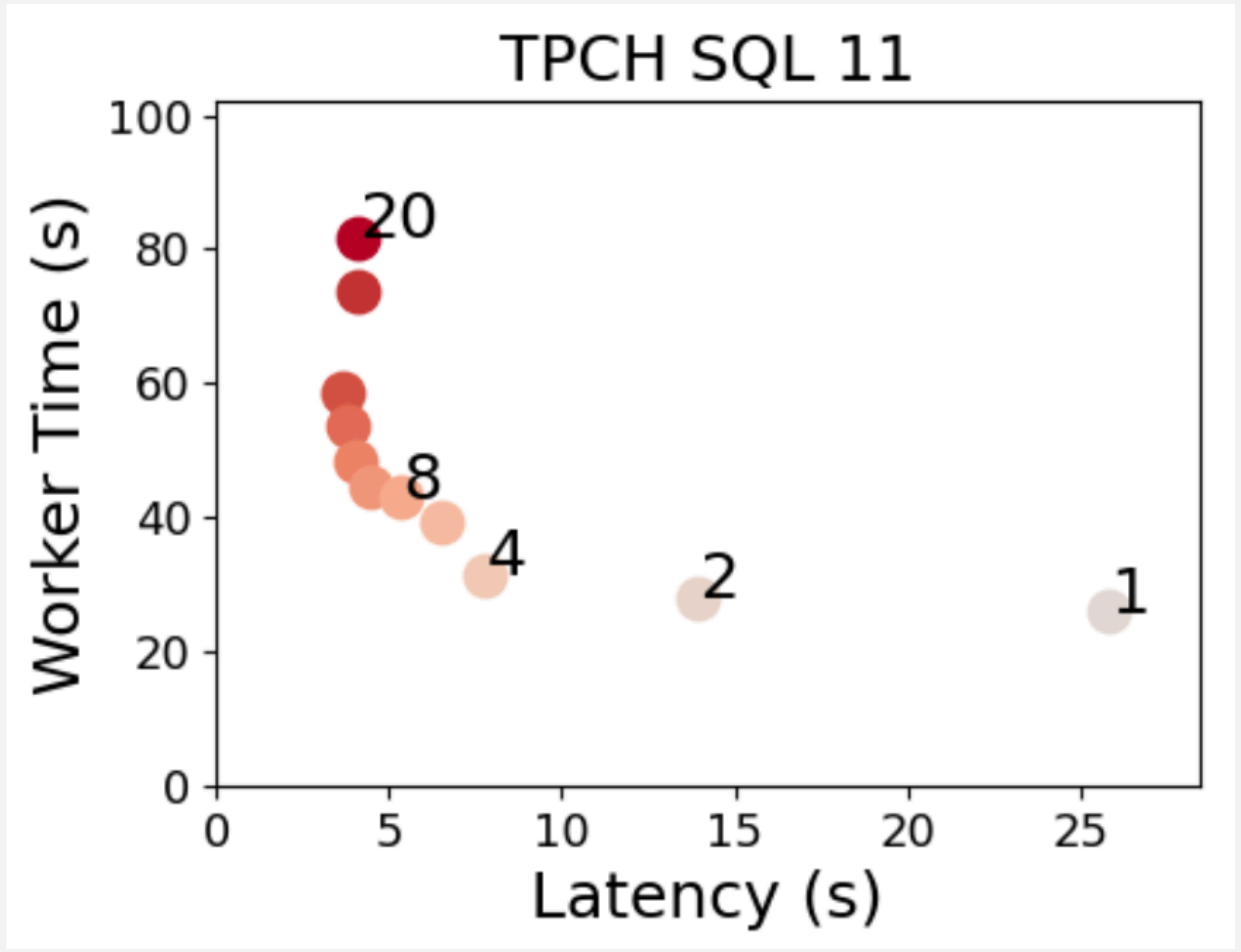
100 servers

Same $ Cost

100x performance boost!

Automatic Resource Deployment





Automatic Resource Deployment

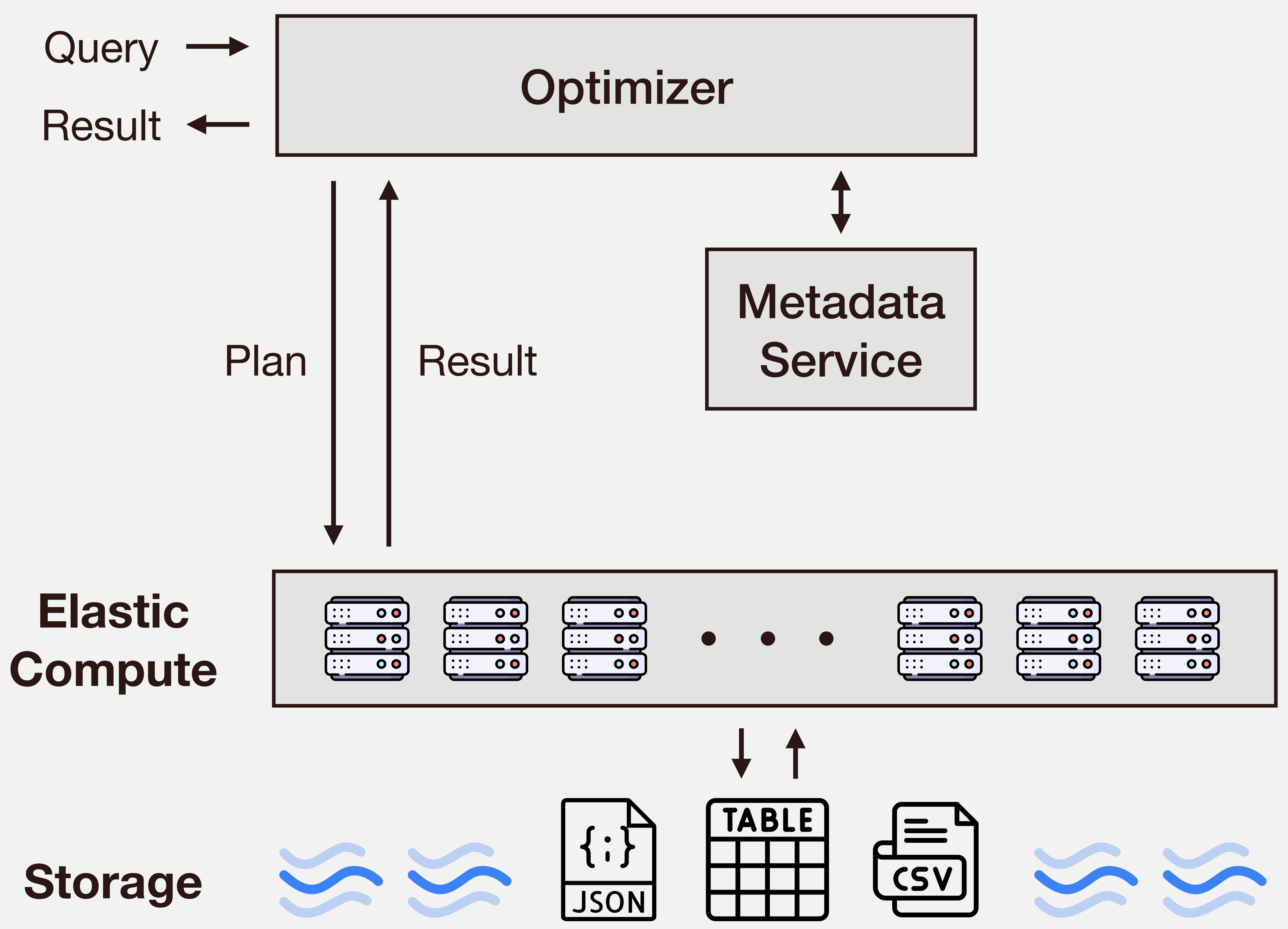
Workload SQL

Static Planning

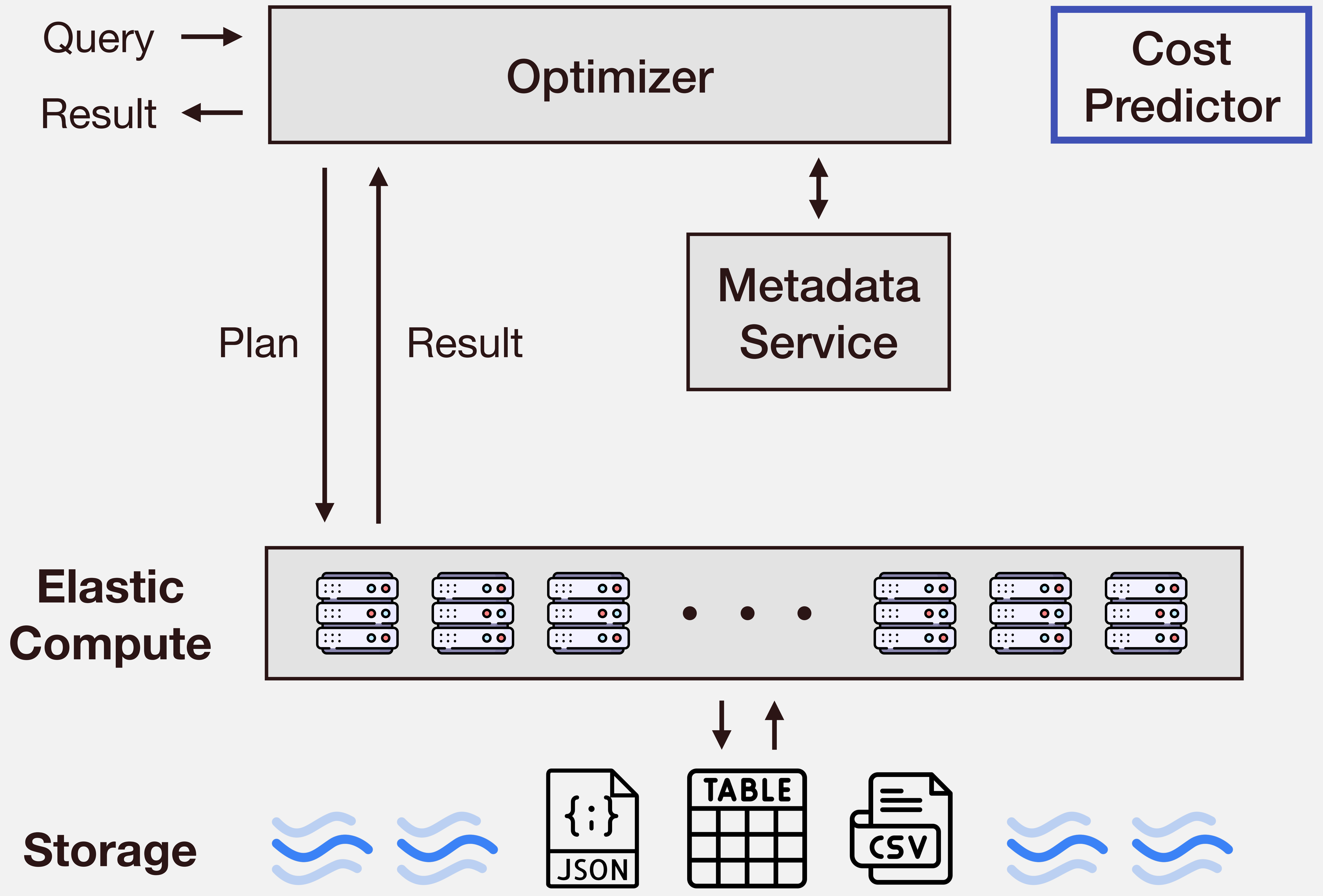
+

Dynamic Adjustment

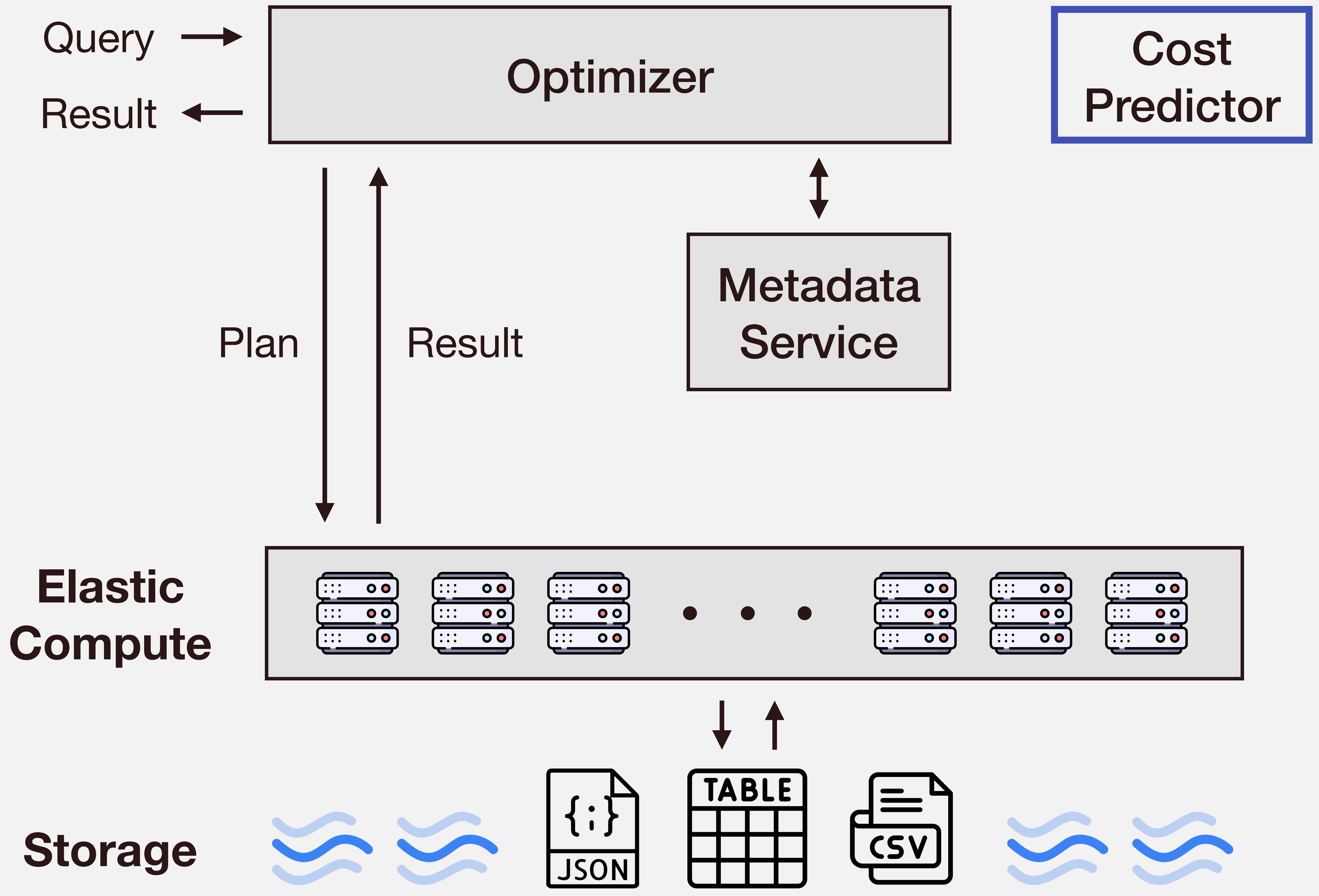
System Architecture



System Architecture

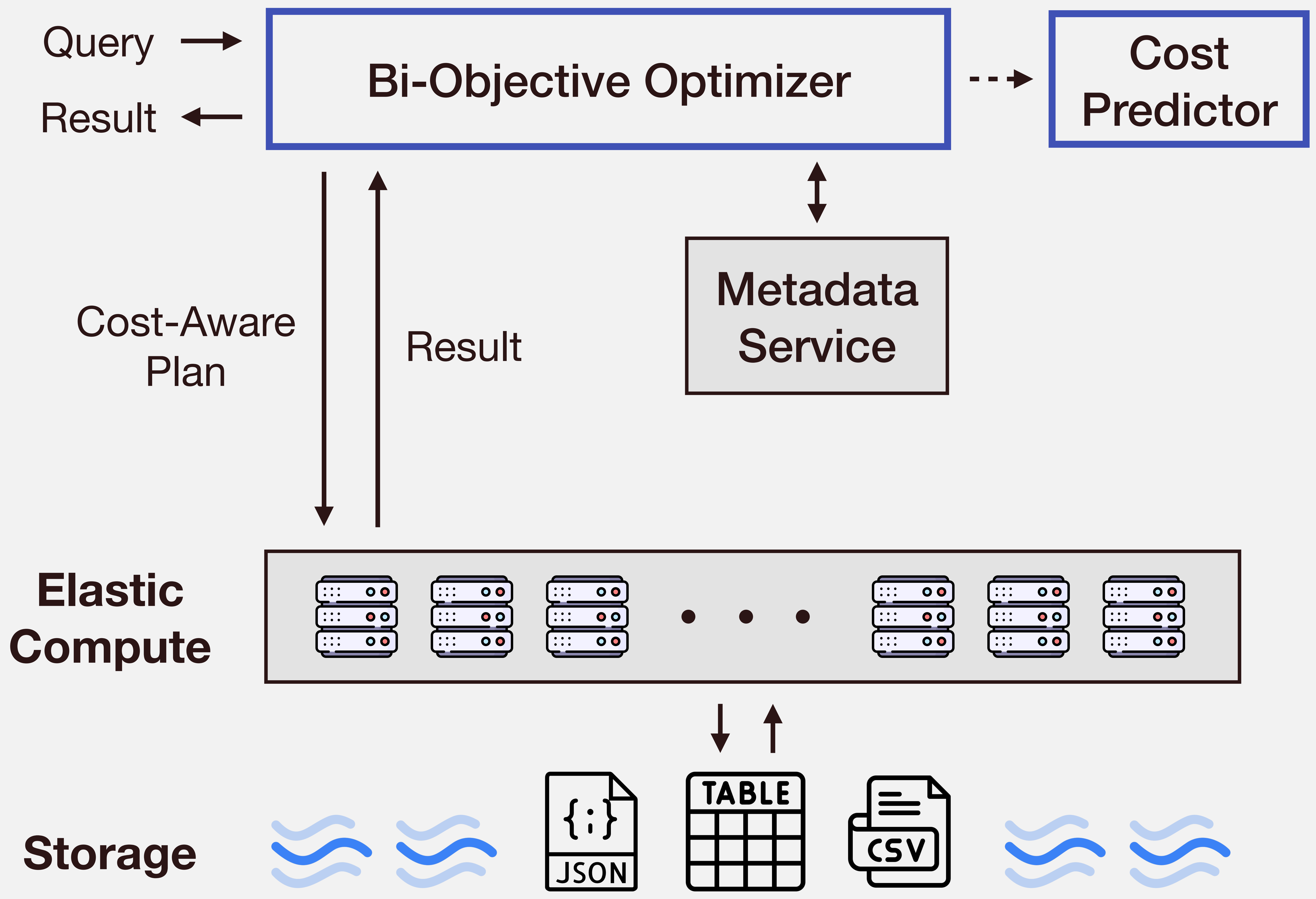


System Architecture

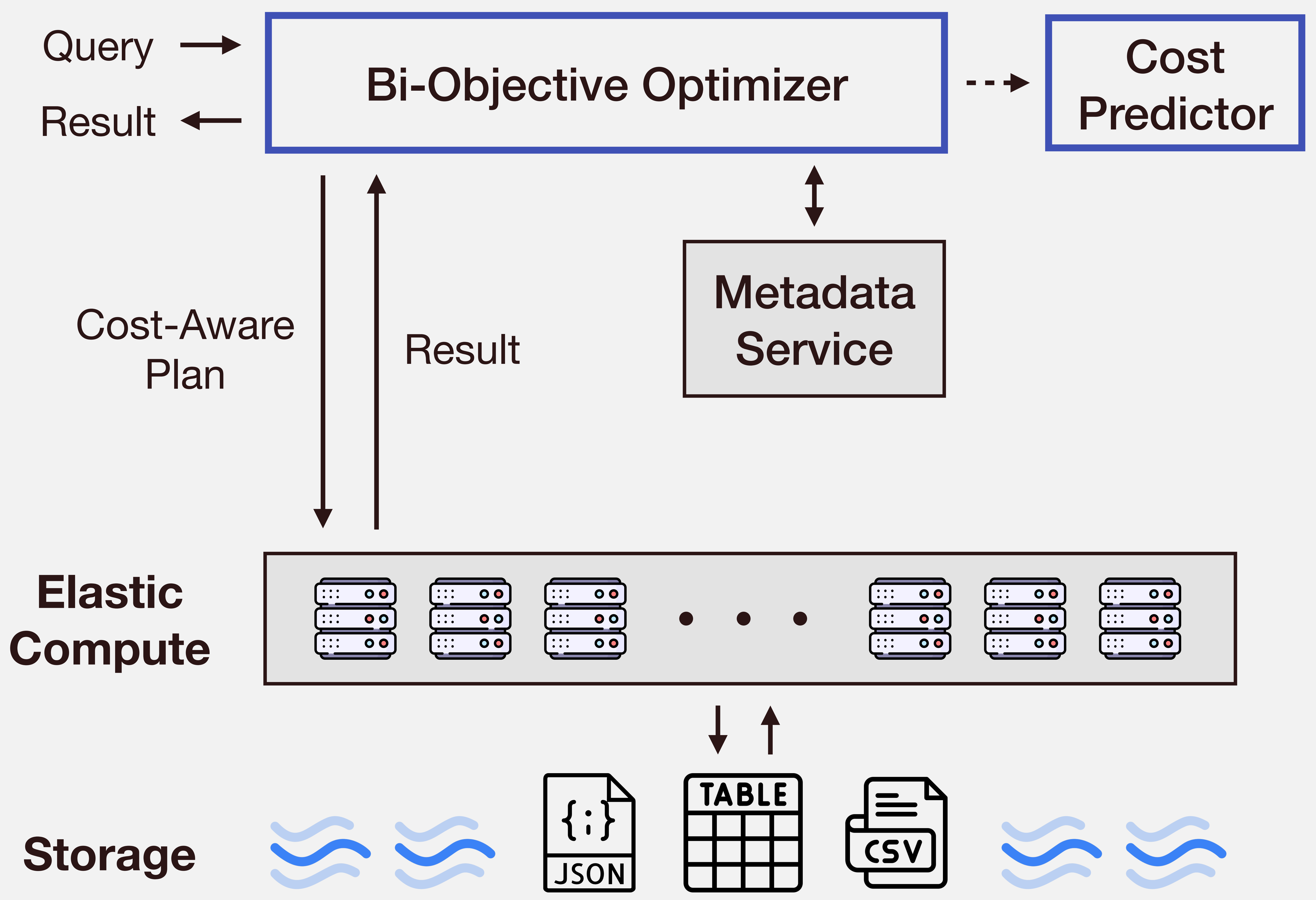


- Accurate
- Lightweight
- Explainable

System Architecture

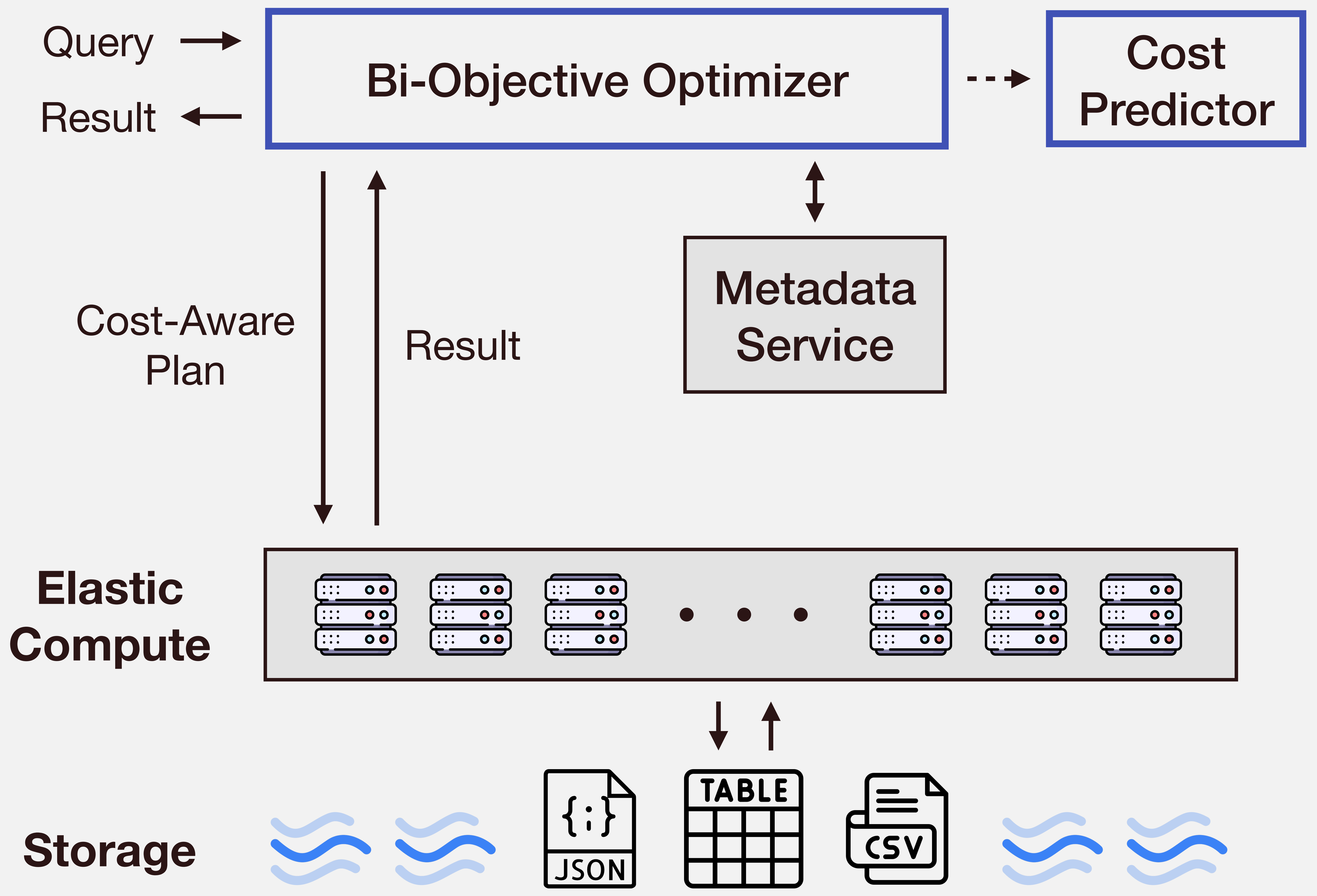


System Architecture



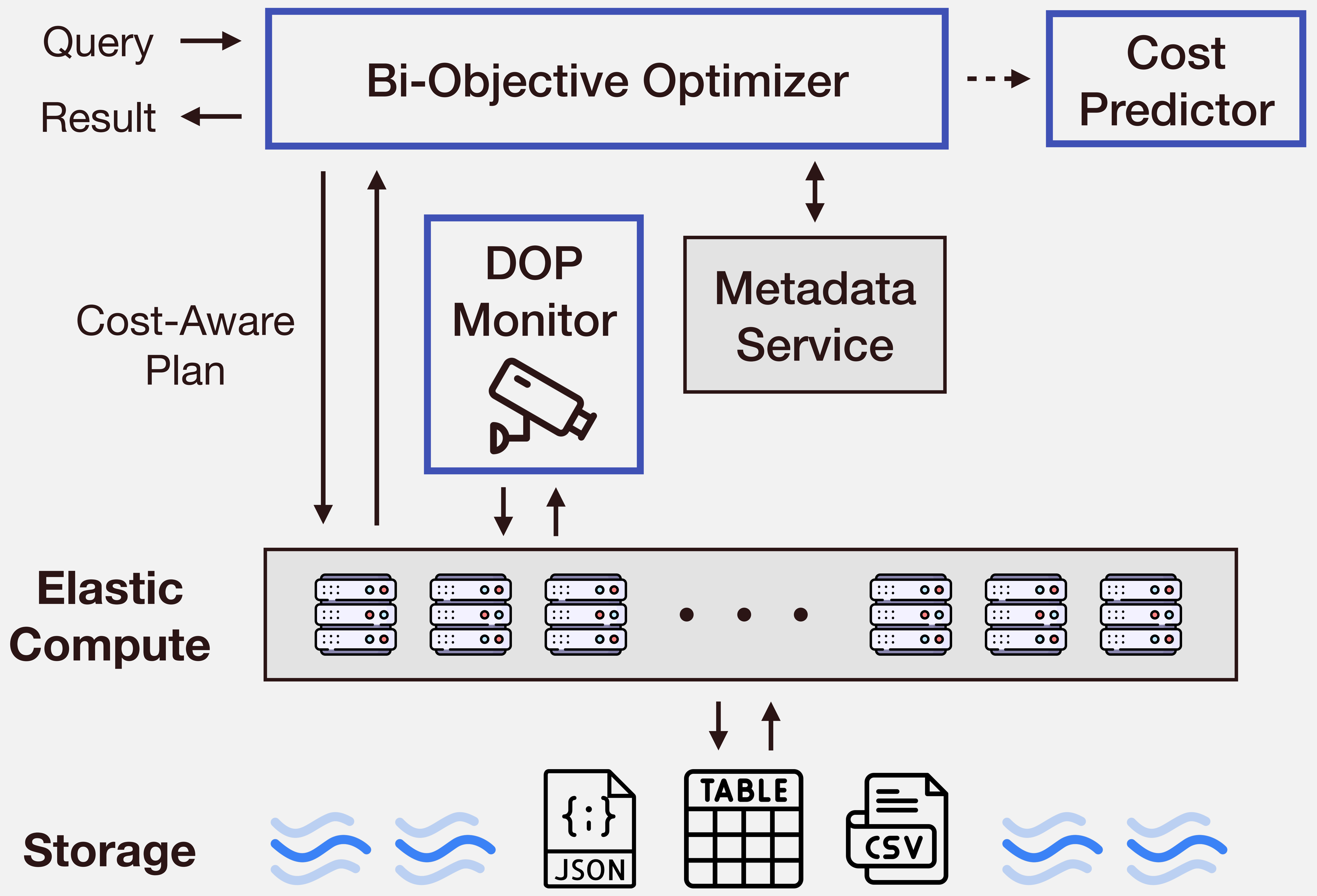
- Downgrade to single-objective with constraints

System Architecture



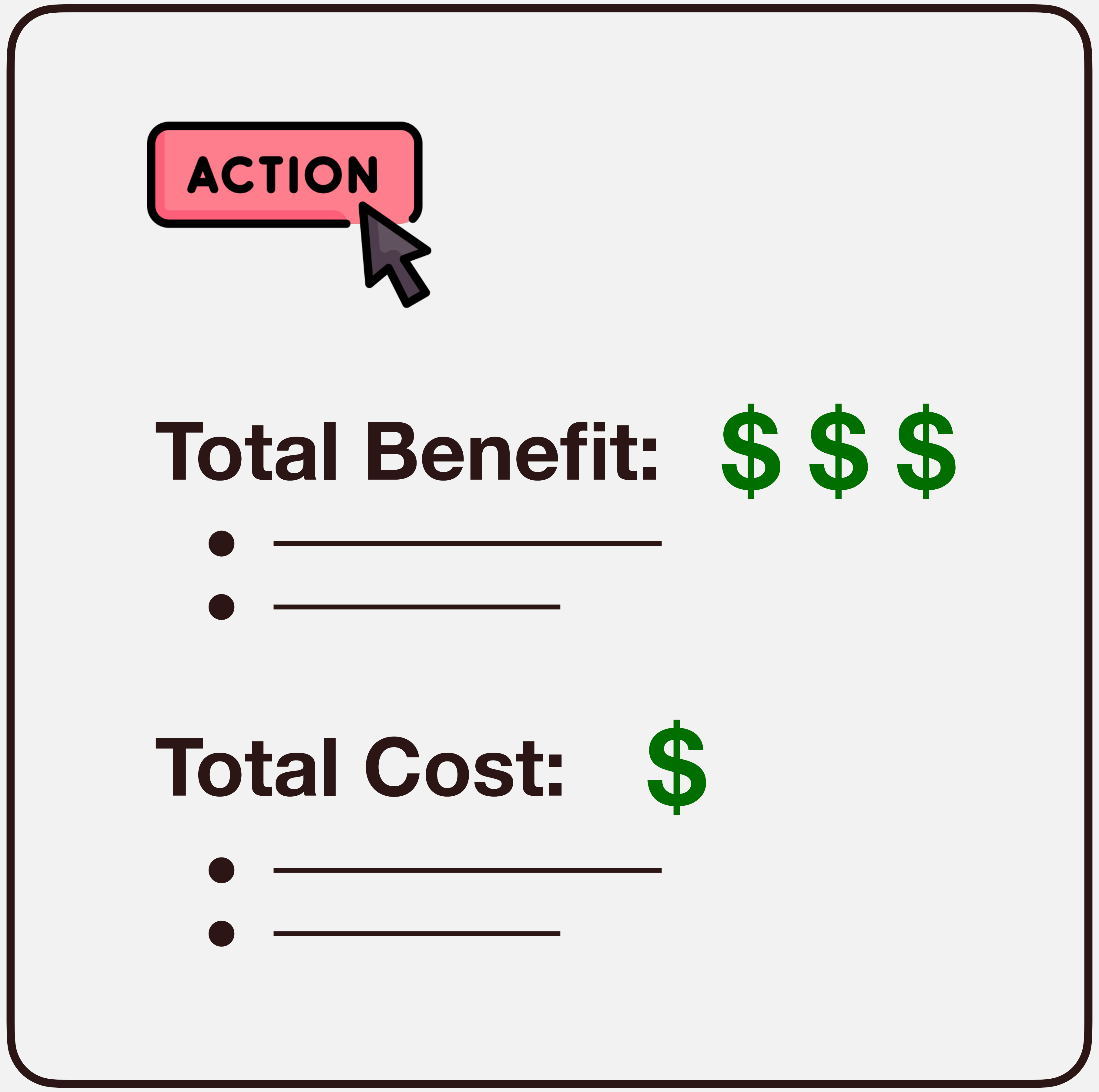
- **Downgrade to single-objective with constraints**
- **Separate DOP planning stage**

System Architecture

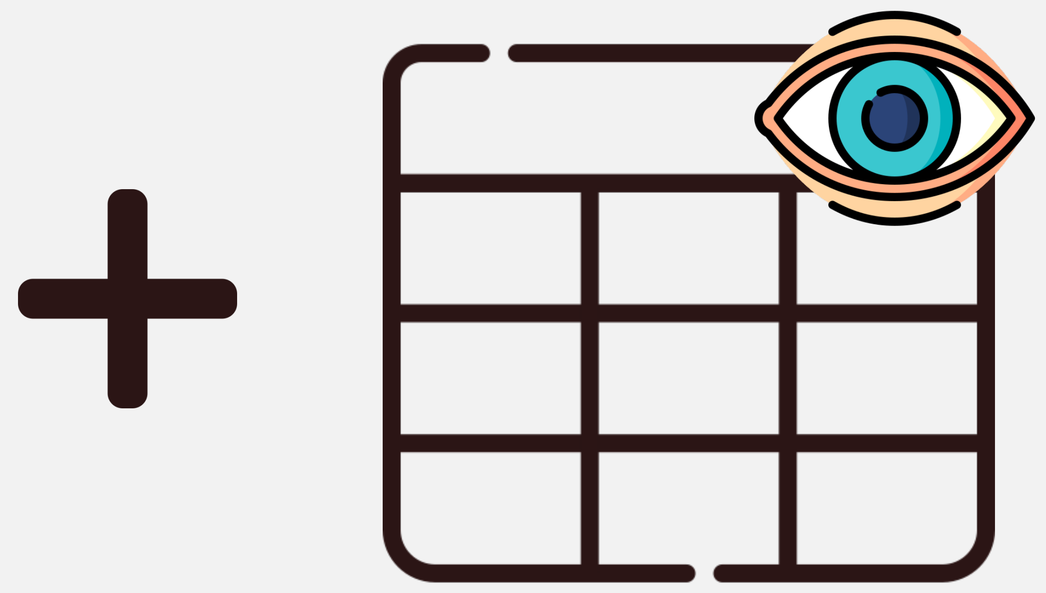


Cost-Oriented Database Auto-Tuning

- Build Indexes
- Build Materialized Views
- Re-partition Data
- Re-train a Learned Module



Database Tuning under Fixed Resources



→ Speeds up a subset of queries

→ MV update slows down writes

Read Perf:

Write Perf:

Database Tuning under Fixed Resources

+

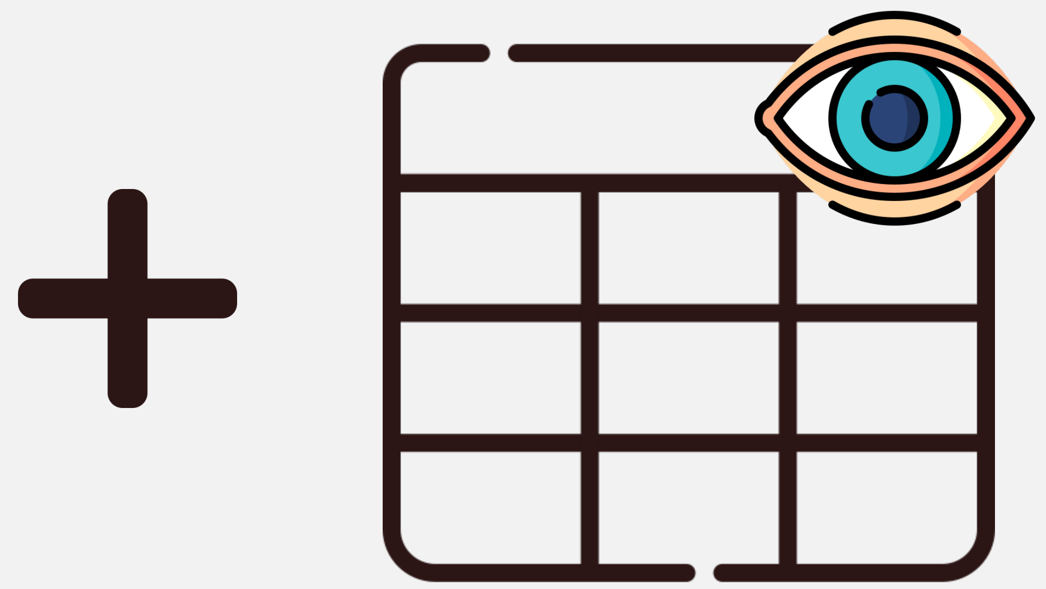
→ Speeds up a subset of queries

→ MV update slows down writes

Read Perf: +

Write Perf:

Database Tuning under Fixed Resources



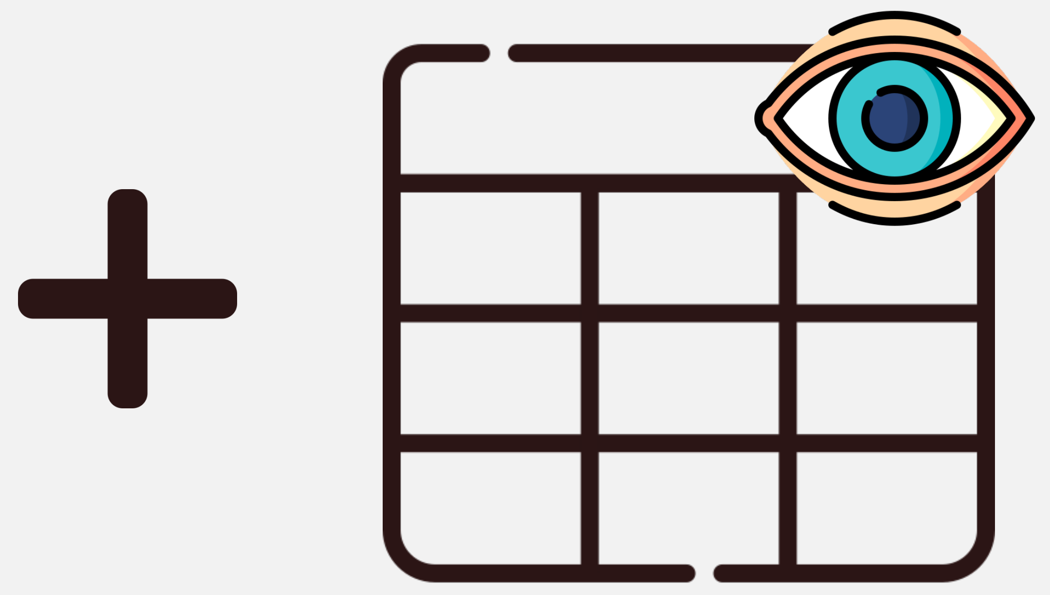
→ Speeds up a subset of queries

→ MV update slows down writes

Read Perf: +

Write Perf: −

Database Tuning under Fixed Resources



→ Speeds up a subset of queries

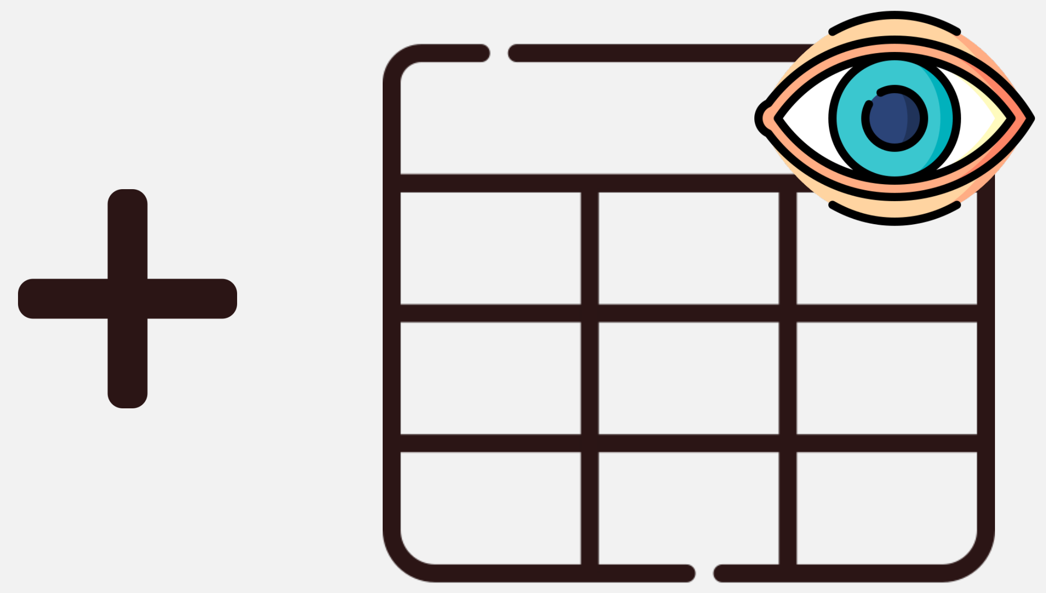
→ MV update slows down writes

Read Perf: + −

Write Perf: −

Resource Contention

Database Tuning under Elastic Resources



→ Speeds up a subset of queries

→ MV update slows down writes

Read Perf:

Write Perf:

Read Cost:

Write Cost:

Database Tuning under Elastic Resources

→ Speeds up a subset of queries

→ MV update slows down writes

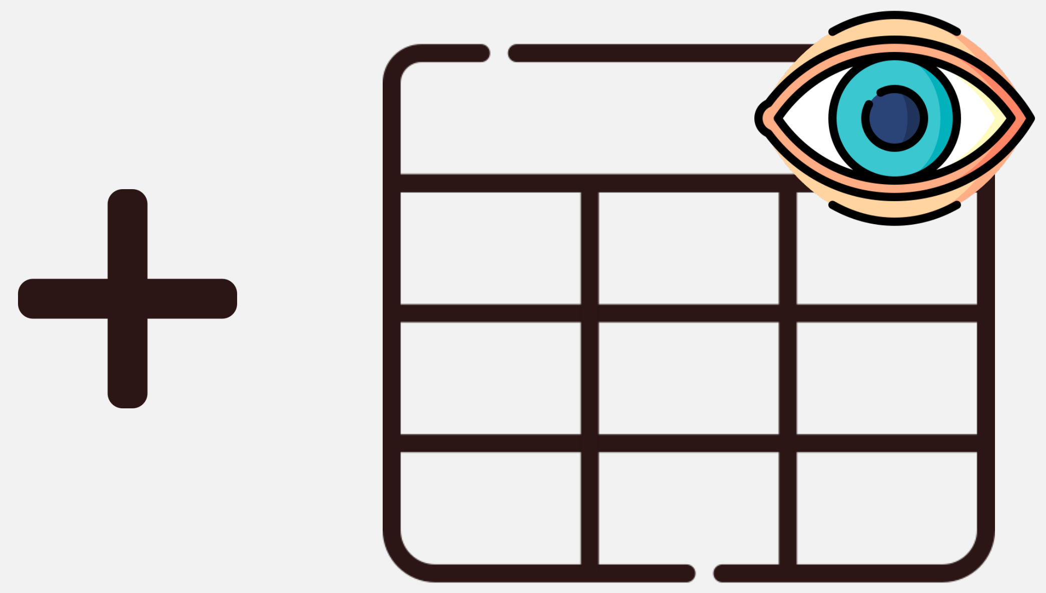
Read Perf: +

Write Perf:

Read Cost: ↓ x

Write Cost:

Database Tuning under Elastic Resources



→ Speeds up a subset of queries

→ MV update slows down writes

Read Perf:	+
Write Perf:	Same
Read Cost:	↓ x
Write Cost:	

Database Tuning under Elastic Resources

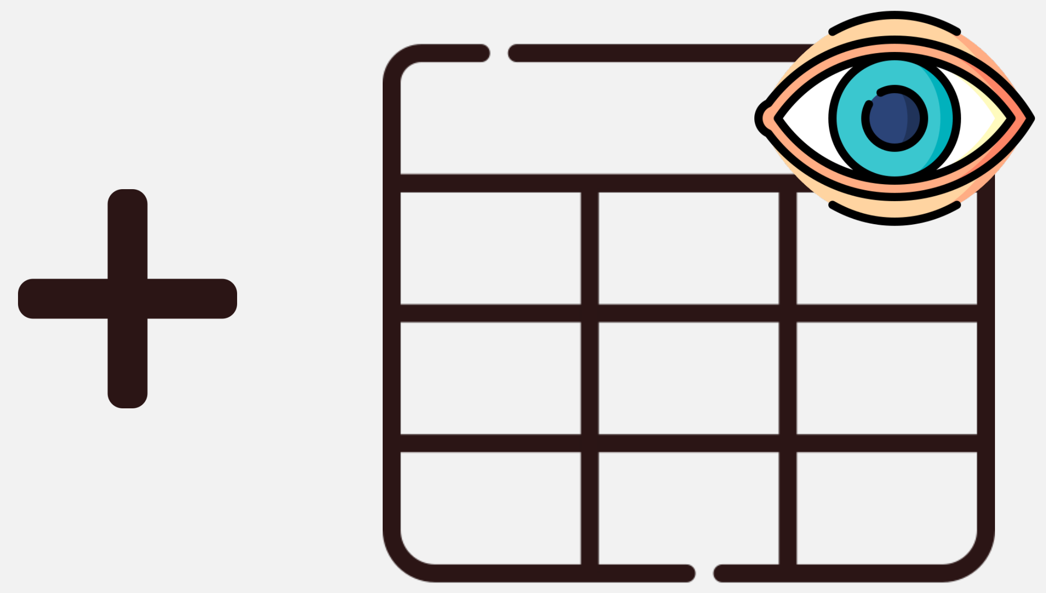
+

→ Speeds up a subset of queries

→ MV update slows down writes

Read Perf:	+
Write Perf:	Same
Read Cost:	↓ x
Write Cost:	↑ y

Database Tuning under Elastic Resources



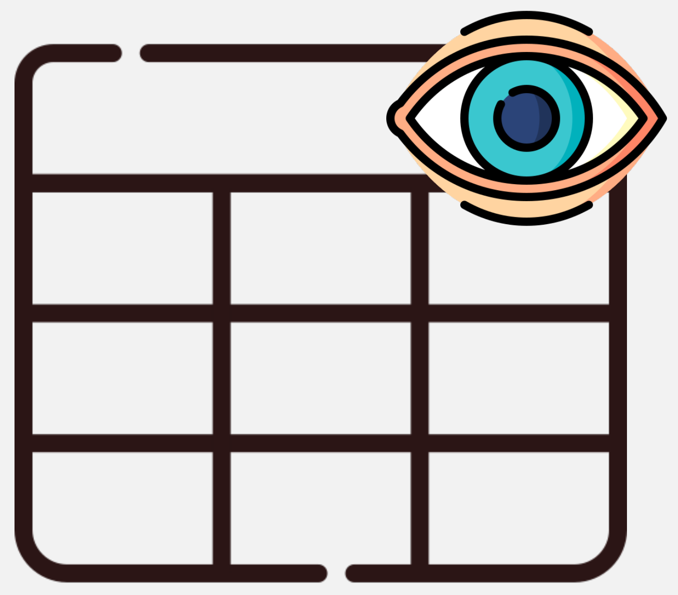
→ Speeds up a subset of queries

→ MV update slows down writes

Read Perf:	+
Write Perf:	Same
Read Cost:	↓ x
Write Cost:	↑ y

$x - y > 0$ ✓

Database Tuning under Elastic Resources

+ 

→ Speeds up a subset of queries

→ MV update slows down writes

Key Challenges:

Accurate Workload Estimation

+

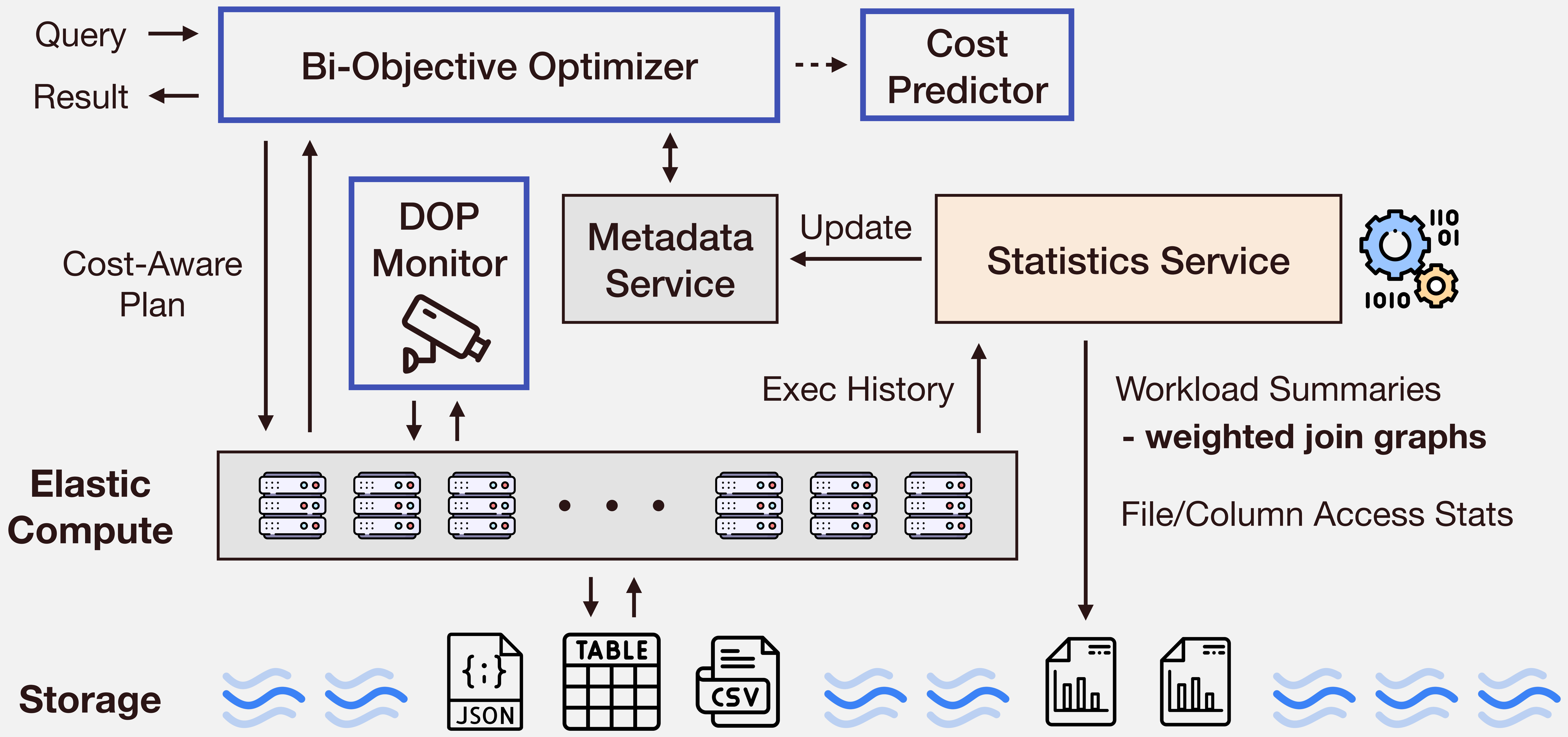
Same

↓ x

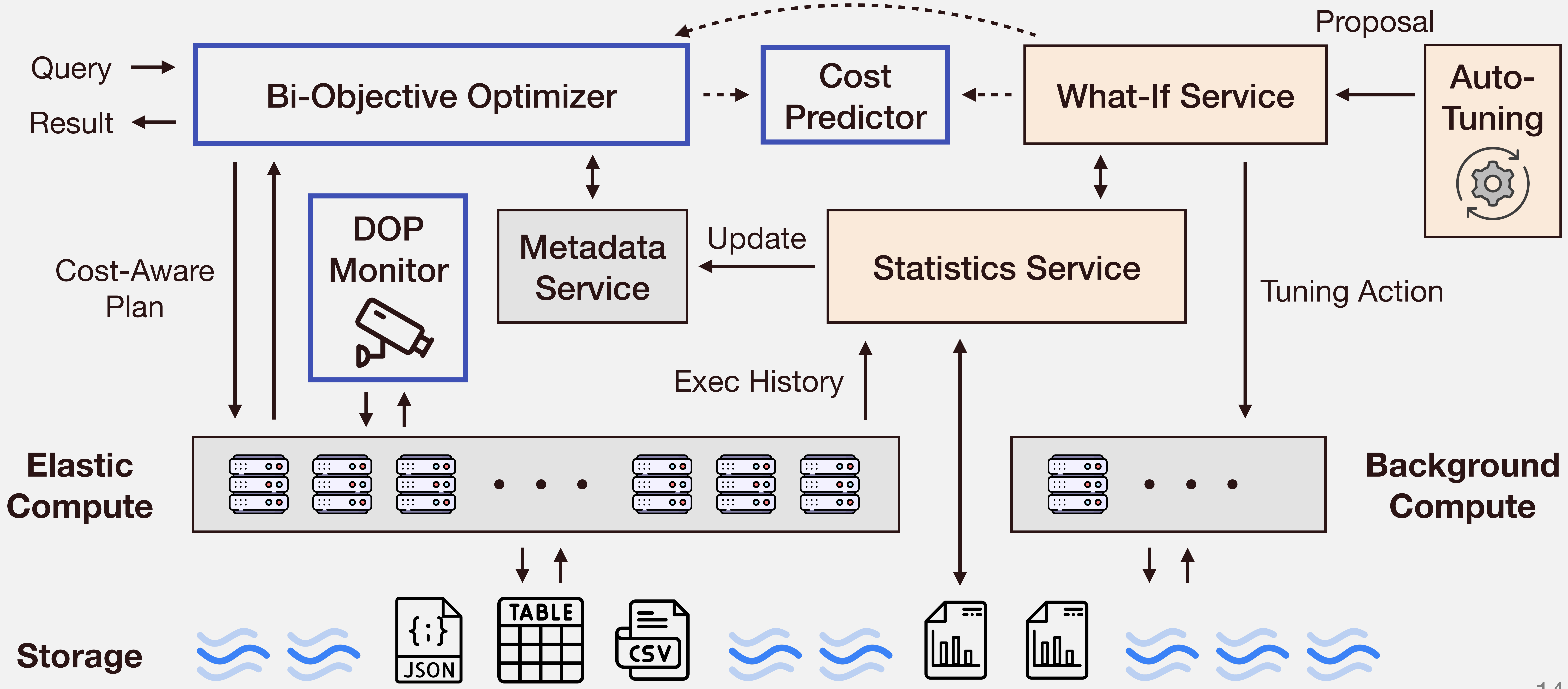
↑ y

$x - y > 0$ ✓

System Architecture



System Architecture



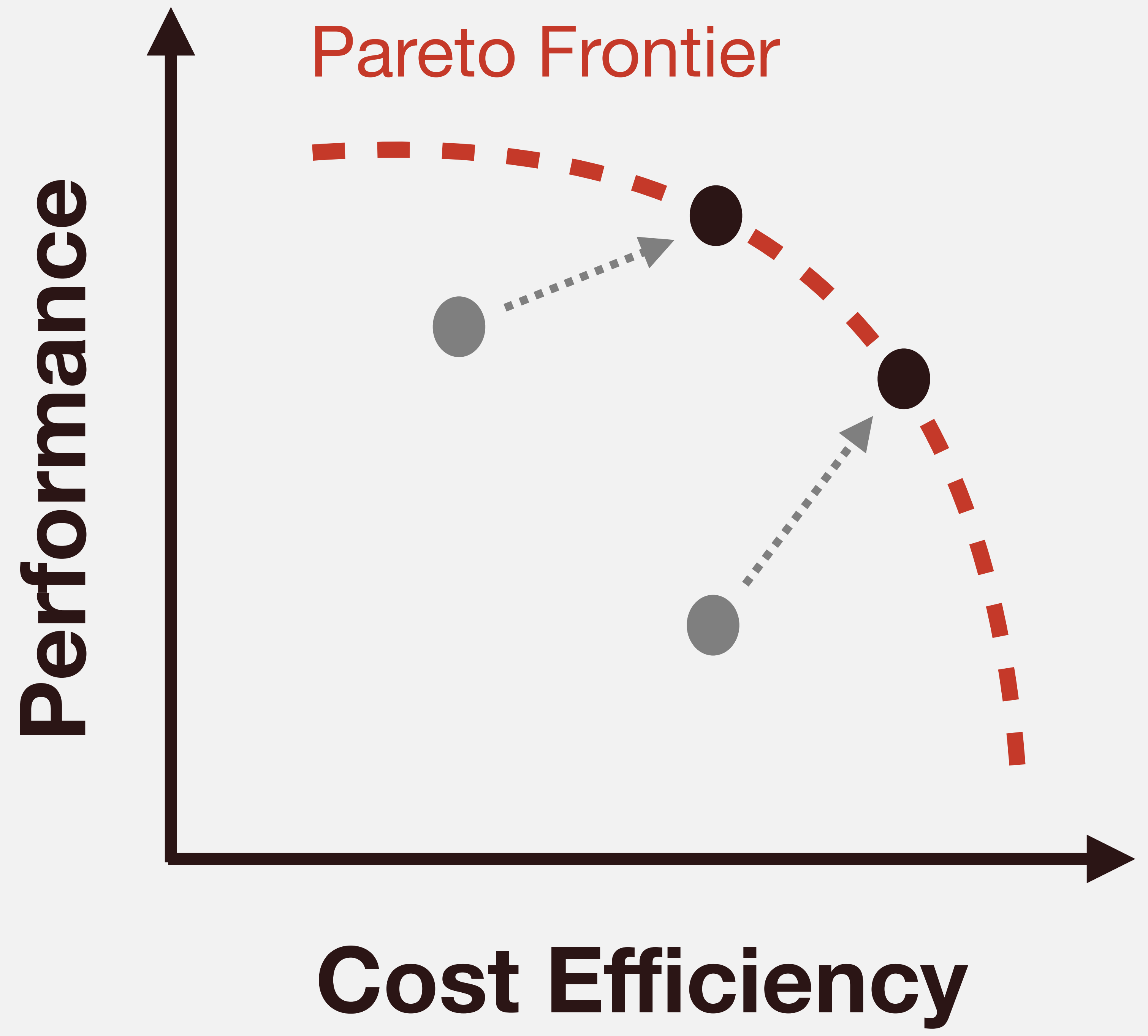
“Economic thinking can help build better systems.”

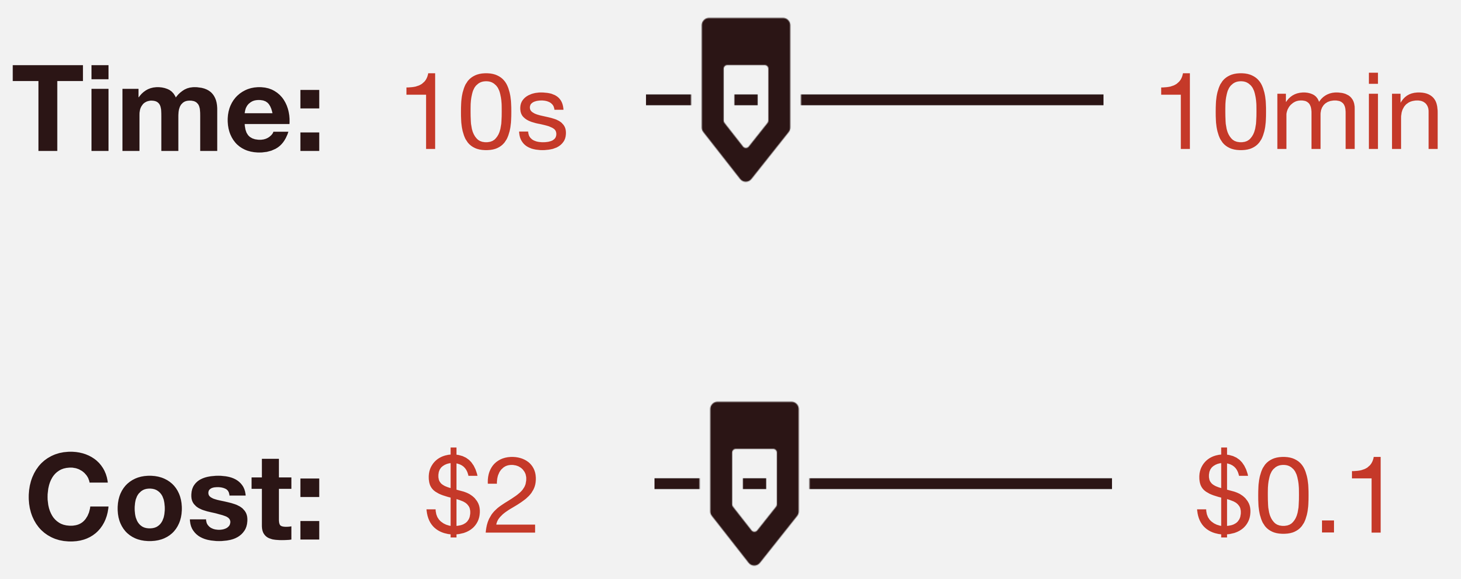
“In the cloud, it’s the only rational way of thinking.”

Towards Cost Intelligence

Cost is as important as performance in cloud-native databases

Vision Paper (CIDR'24): https://arxiv.org/pdf/2308.09569.pdf


Pareto Frontier
Performance
Cost Efficiency


Time: 10s
10min
Cost: $2
$0.1