

The Squirrel, the Witch and the Wardrobe

**How we re-built Synapse SQL
for Microsoft Fabric**

Bogdan Crivat
VP, Synapse Analytics
Microsoft



Who is this guy, again?

- Joined Microsoft in 1999 (left for a while, came back)
- Mostly worked on Analysis Services, on Data Mining and OLAP.
- Co-authored the Vertipaq engine used today by Power BI
- Summer of 2021 – running Power BI Engineering
- Since then – running Synapse Analytics (Data Warehousing, Data Engineering, Data Science)

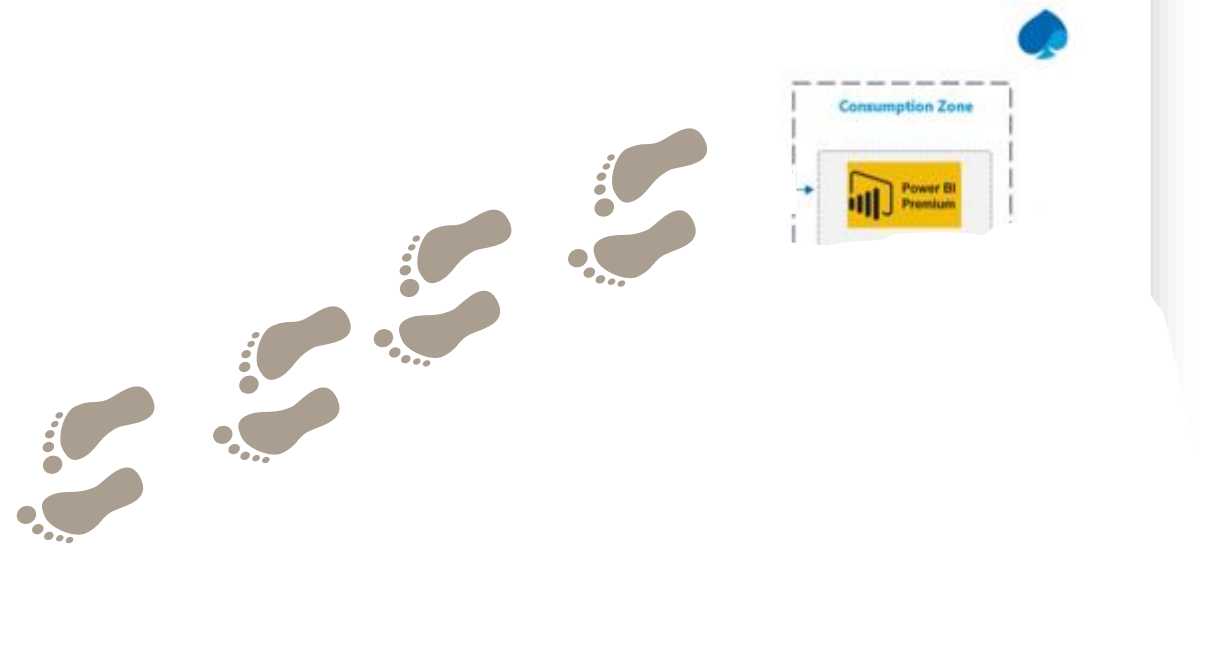


And so, my journey started ...

From a small
BI village

• ...

To a complicated
Big Data
Landscape



1.5 years later ...

You may have
heard ...



Microsoft Fabric

Data analytics for the era of AI



Microsoft Fabric

Data analytics for the era of AI

Complete Analytics Platform

Everything, unified

SaaS-ified

Secured and governed

Lake Centric and Open

OneLake

One copy

Open at every tier

Empower Every Business User

Familiar and intuitive

Built into Microsoft 365

Insight to action

AI Powered

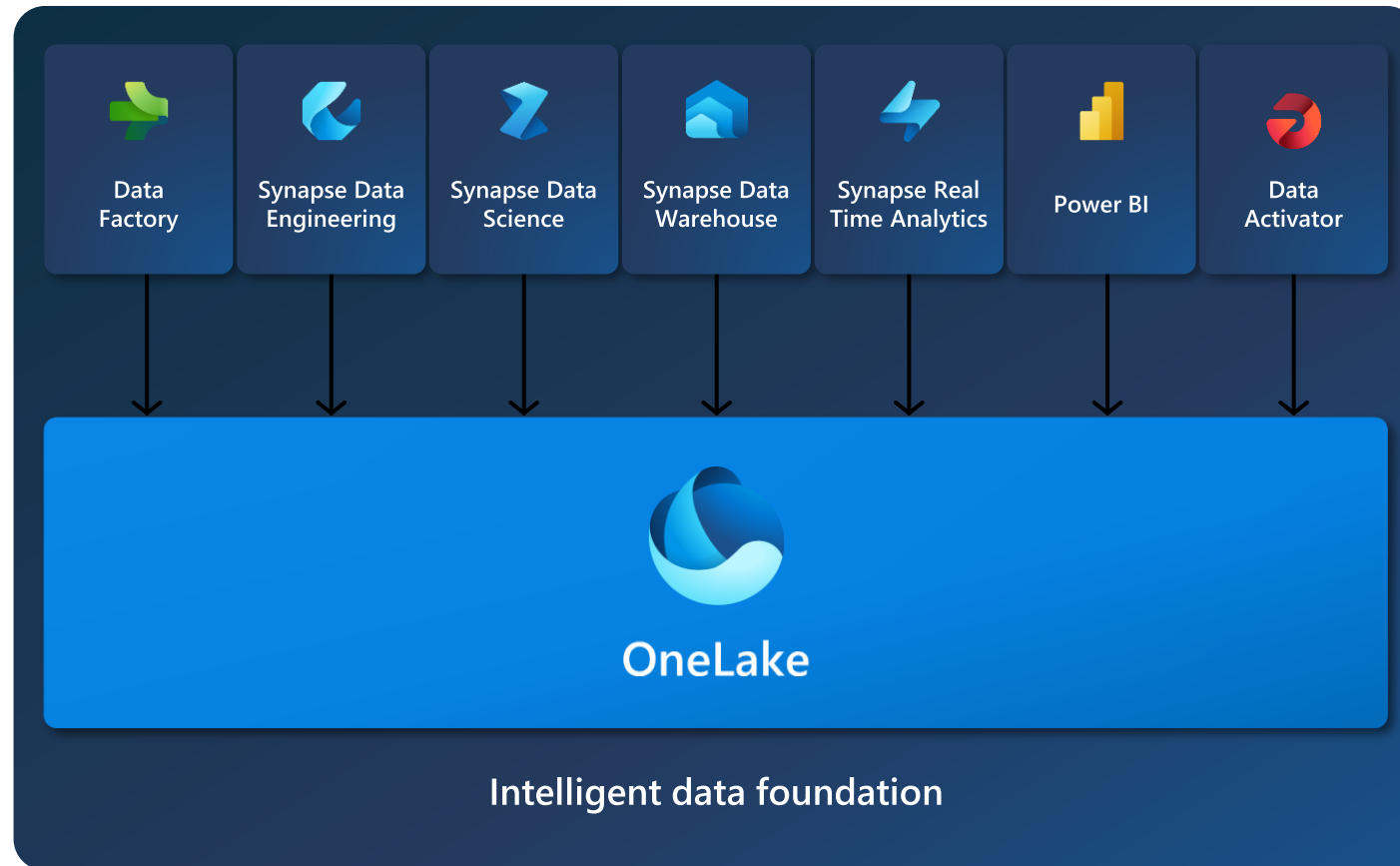
Copilot accelerated

GPT on your data

AI-driven insights

OneLake for all Data

"The OneDrive for Data"



A single SaaS lake for the whole organization

Provisioned automatically with the tenant

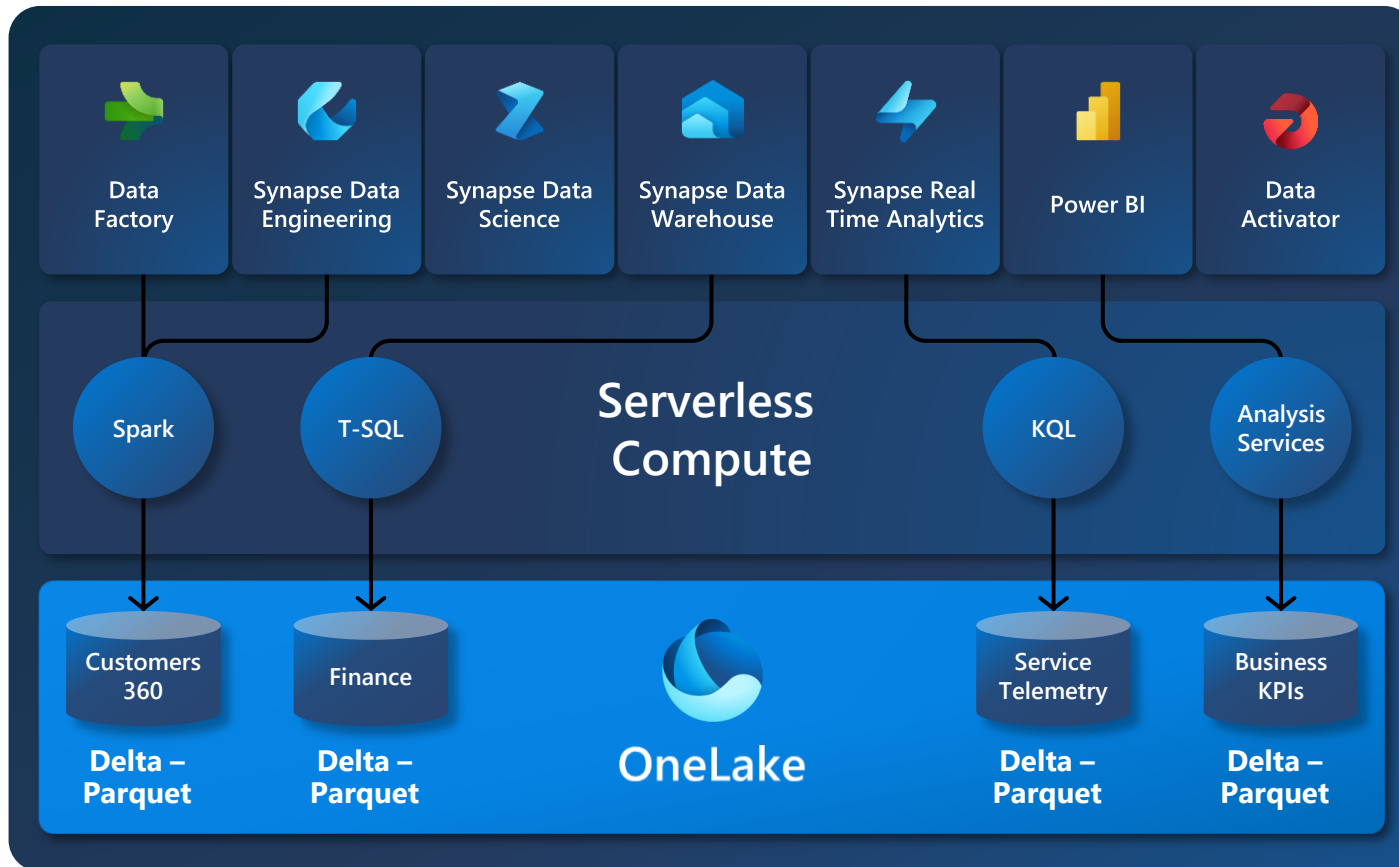
All workloads automatically store their data in the OneLake workspace folders

All the data is organized in an intuitive hierarchical namespace

The data in OneLake is automatically indexed for discovery, MIP labels, lineage, PII scans, sharing, governance and compliance

One Copy for all computes

Real separation of compute and storage



All the compute engines store their data automatically in OneLake

The data is stored in a single common format

Delta – Parquet, an open standards format, is the storage format for all tabular data in Analytics vNext

Once data is stored in the lake, it is directly accessible by all the engines without needing any import/export

All the compute engines have been fully optimized to work with Delta Parquet as their native format

Shared universal security model is enforced across all the engines



Microsoft Fabric

Data analytics for the era of AI



Synapse SQL



OneLake
Delta – Parquet

Note: Big Data SQL is not OLTP SQL

```
SELECT SUM(Sales), City
      FROM Sales
GROUP BY City
```

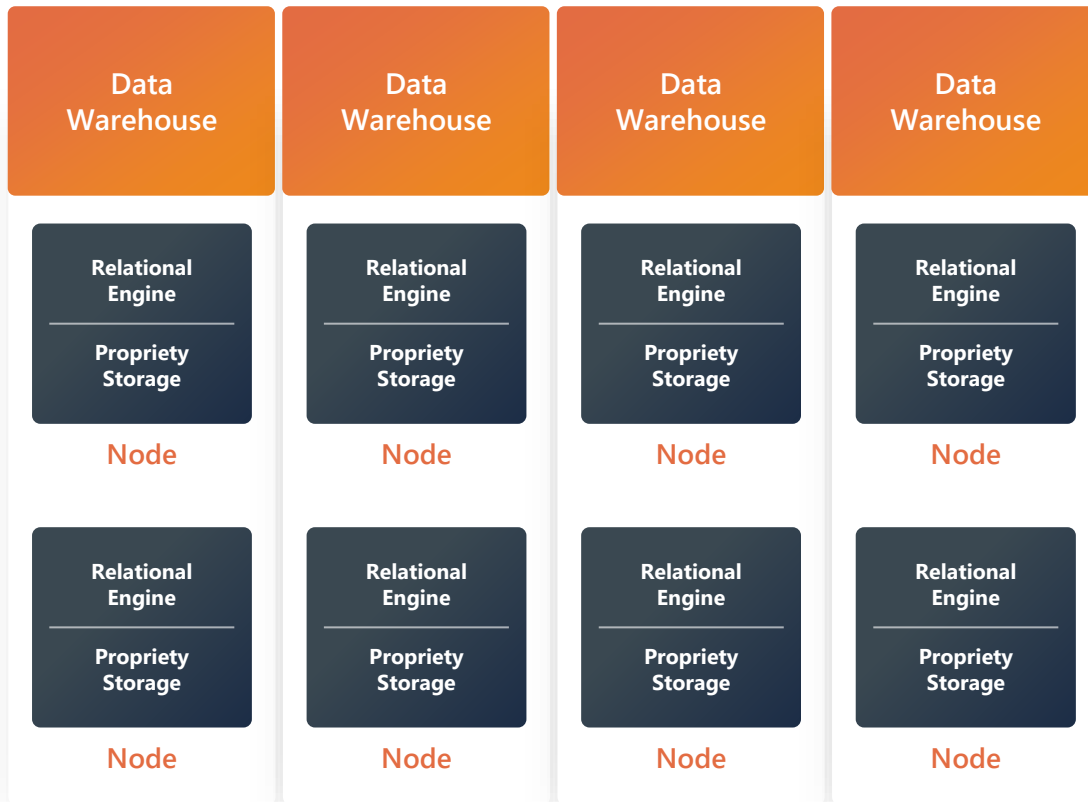
```
INSERT INTO Sales (
      CustomerID,
      City,
      Amount,
      Product.
      ...)
VALUES
      ( ... )
```

Synapse needs a new **Big Data SQL**

Wave 1 – Tightly coupled storage and compute

- The Teradata Wave: Optimized nodes with storage and compute

TERADATA



Dominant in on-premises architecture

Avoids the network at any cost

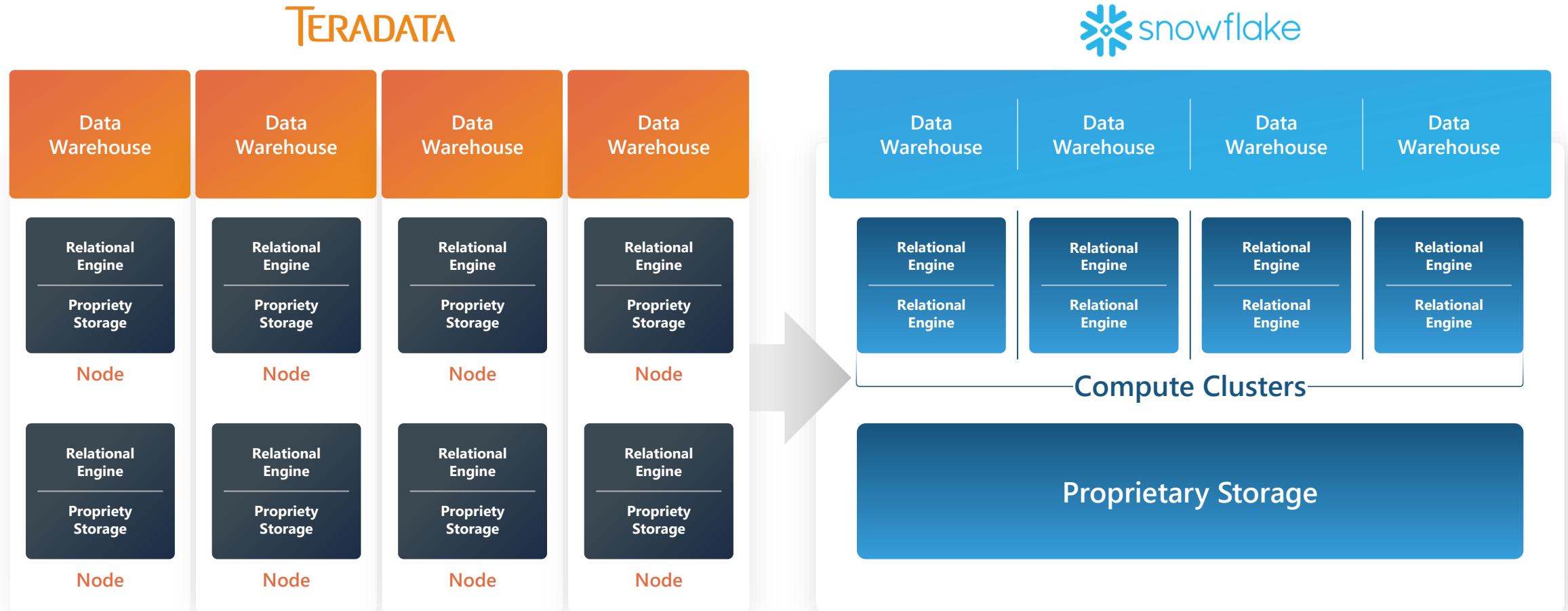
Scales up as much as possible before MPP

Rigid data distribution on nodes makes workload adjustments costly

Each DW is an isolated data island, with limited sharing capabilities

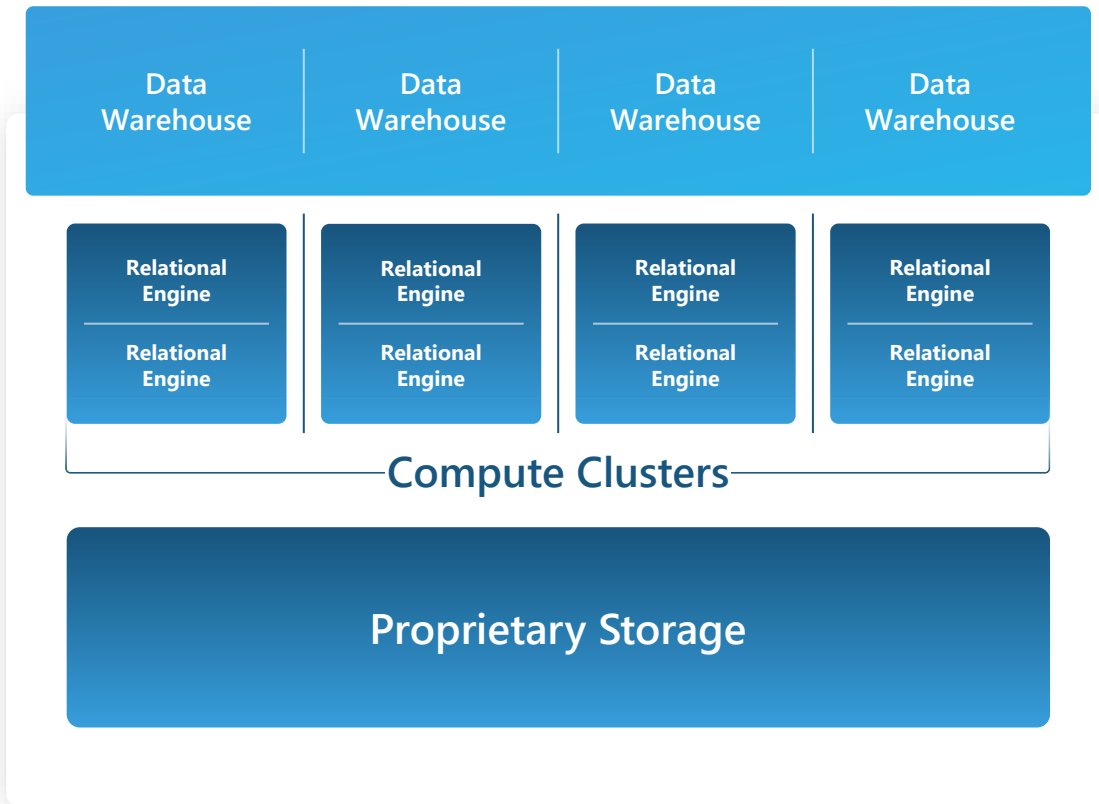
Wave 2 – Separation of compute and storage

The Snowflake Wave: Easy scaling and sharing



Wave 2 – Separation of compute and storage

The Snowflake Wave: Easy scaling and sharing



The first true cloud native systems

Taking advantage of the massive pools of compute clusters available in the cloud

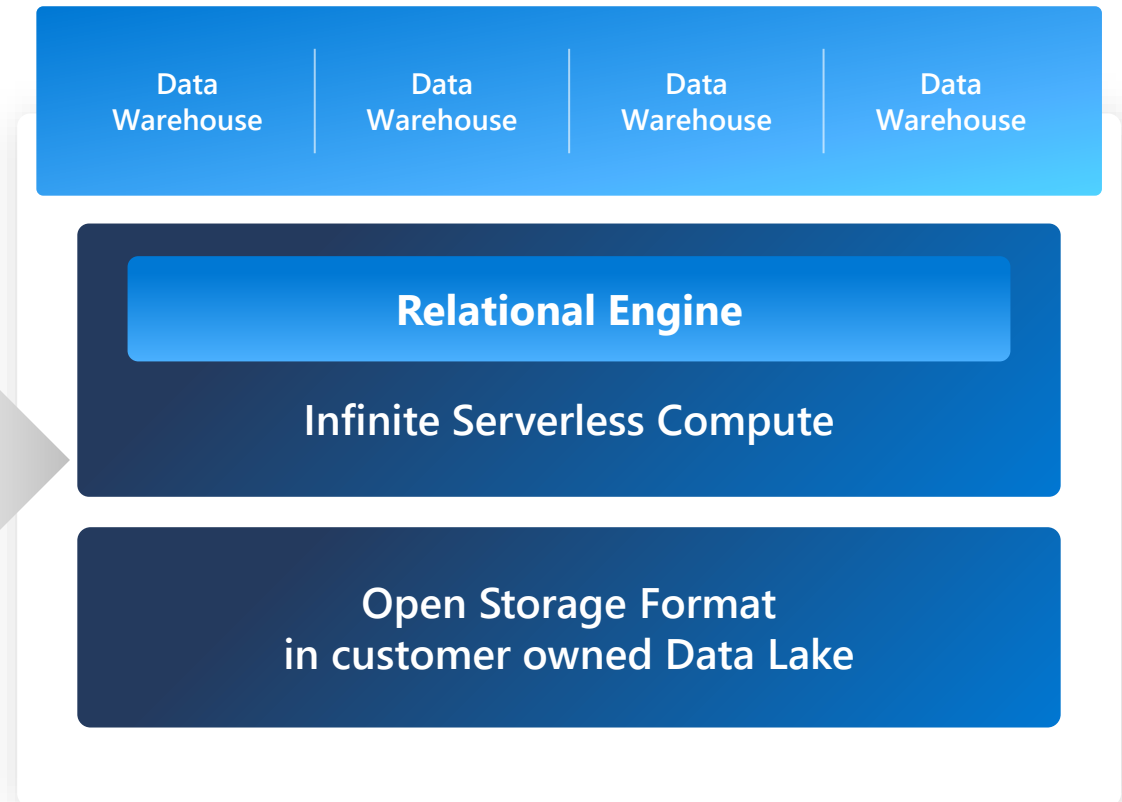
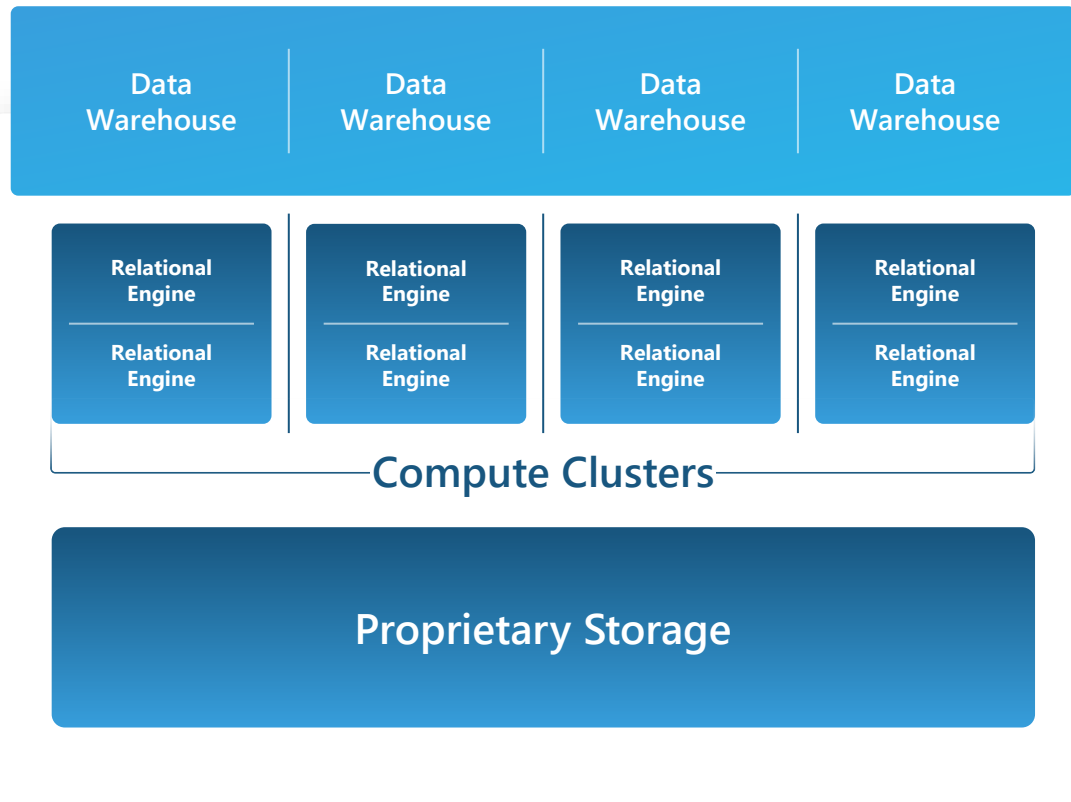
Separating the data storage from the compute allows easy scaling up/down on demand without needing to redistribute data

Data sharing is easy with multiple clusters querying the same data storage

Systems include **Snowflake** and Redshift

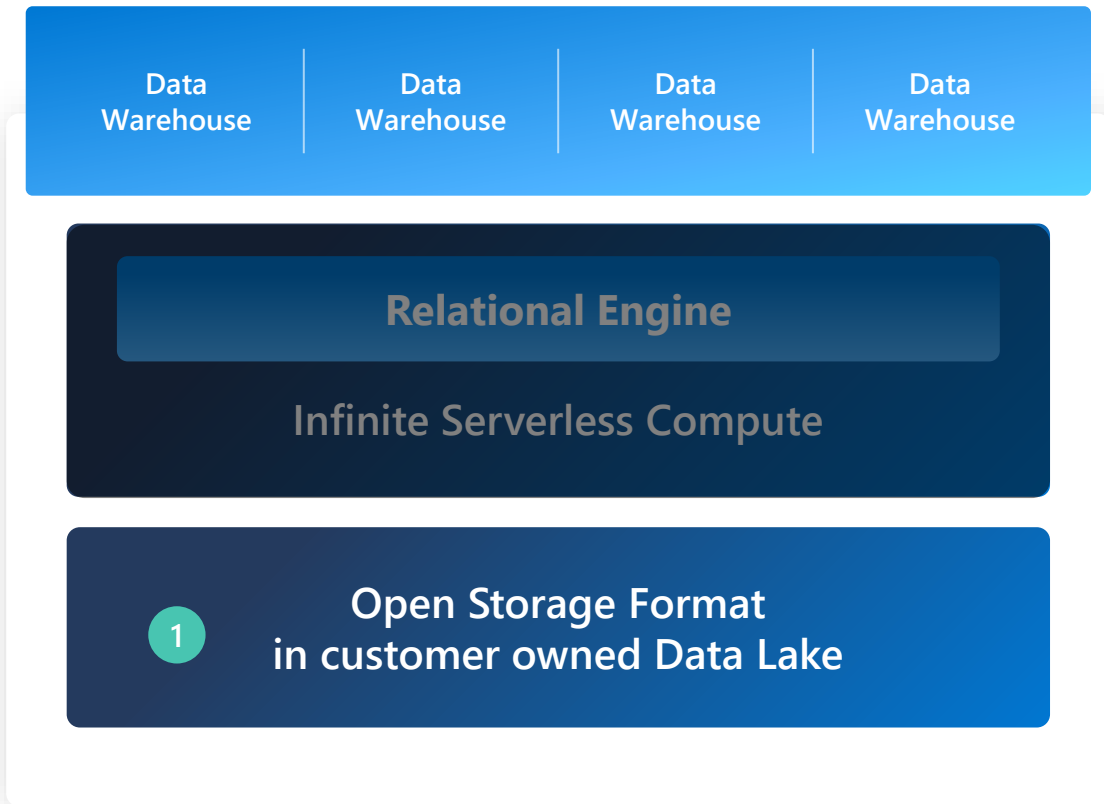
Wave 3 – Unified analytics fabric

The Microsoft Fabric Wave: Infinitely scalable and open



Wave 3 – Unified analytics fabric

Infinitely scalable and open



1 Open standard format in an open datalake replaces proprietary formats as the native storage

Teradata, Snowflake, BigQuery, Redshift and Synapse Gen2 all have their own proprietary storage format

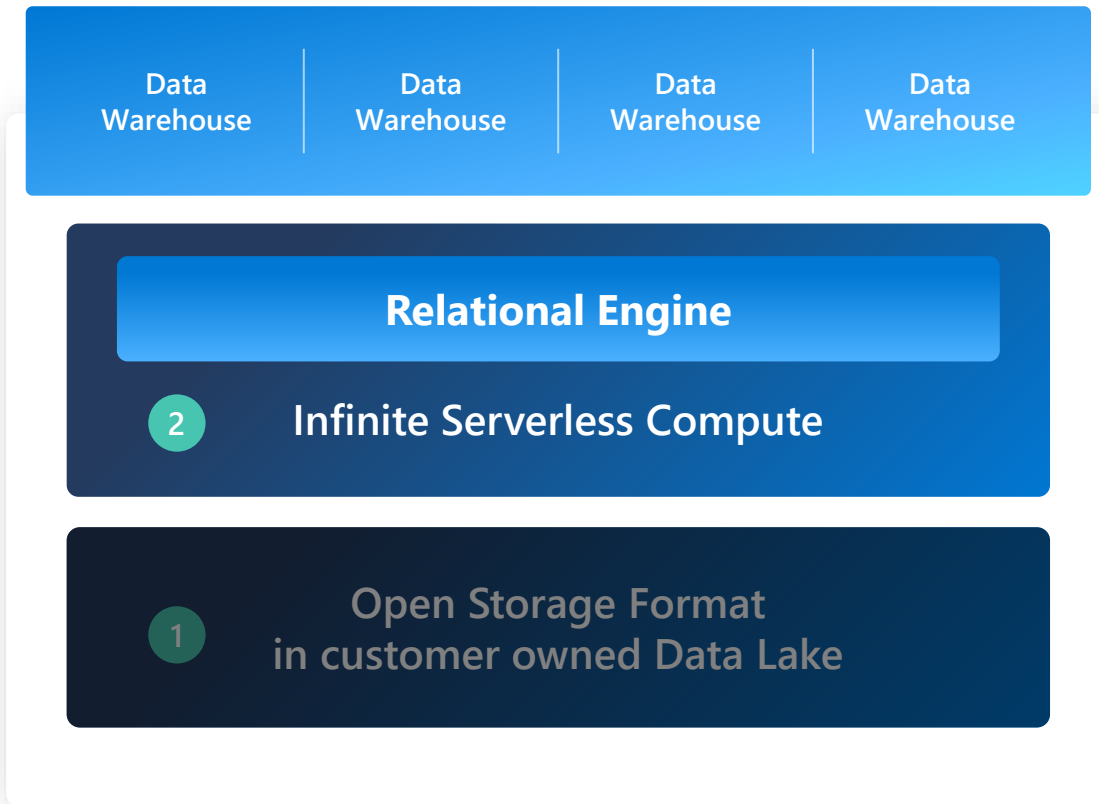
Synapse SQL in Fabric is the first Transactional Data Warehouse natively embracing an open standard format, **fully optimized**

Customers appreciate the openness, lack of lock-in and the ecosystem benefits

This has profound impact on the entire analytics fabric as will be discussed later

Wave 3 – Unified analytics fabric

Infinitely scalable and open



2 Serverless replaces dedicated clusters as the compute engine infrastructure

Serverless is superior with:

- Physical compute resources assigned within milliseconds to jobs
- Infinite scaling with dynamic resource allocation tailored to data volume and query complexity
- Instant scaling up/down with no physical provisioning involved
- Resource pooling providing significant COG efficiencies and pricing power

How to build it

The team

The US team

Synapse Dedicated SQL

The Serbia team

Synapse Serverless SQL

World class
Engineers &
Product Managers



The tech
It's complicated!



The Squirrel

Synapse Serverless SQL

- Serverless, elastic compute!
- Can ship every week
- Works on open formats
- Really good at distributed queries

But

- Cannot carry too much weight yet
- Little or no query optimization/statistics

The Wardrobe

Synapse Dedicated SQL

- It can hold a lot!
- Rich data statistics
- Solid query optimizer

But

- Not exactly flexible or nimble
- Takes months to ship a new one
- Proprietary storage (CCI)



It takes a bit of magic
to get the best of them



The Witch

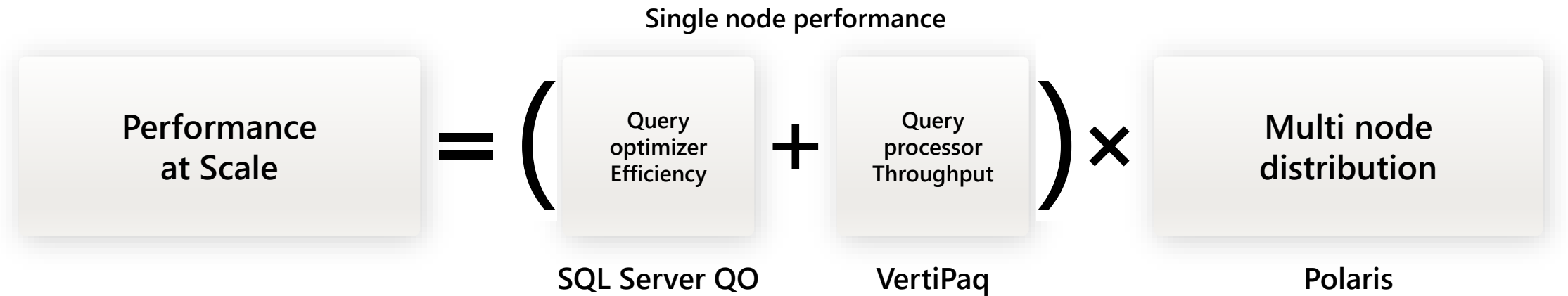
Power BI

- It is really fast!
 - It knows the secret of open lake formats*
-

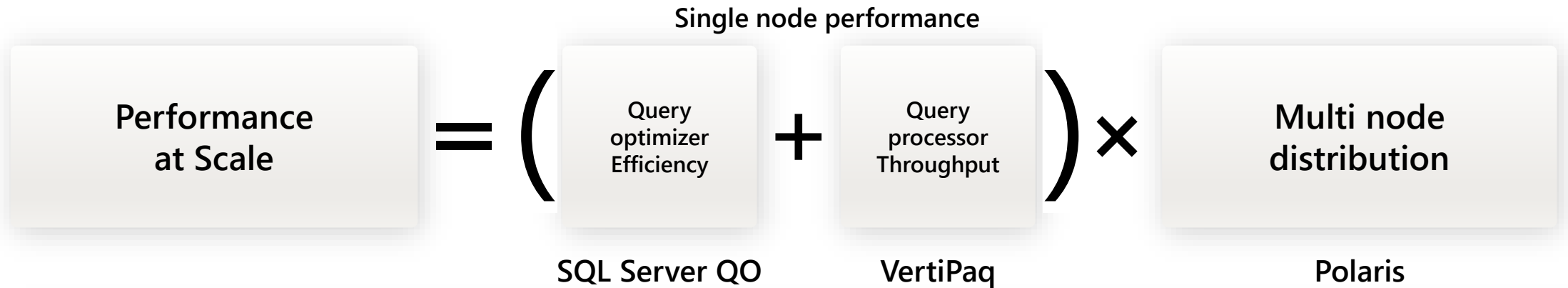
And yes!

It also ships weekly!

Performance at scale formula



Performance at scale formula



SQL Server QO

The best query optimizer in the industry

VertiPaq

The fastest columnar query processor in the world (originated from Power BI)

Polaris


The most scalable distributed query processor in the world with the only published **Petabyte** scale benchmark (VLDB 13, August 2020)

So, we'll talk about...

- ❑ Query processing
- ❑ Query optimization
- ❑ Multi-node distribution

Reading (and writing) data

What is Parquet



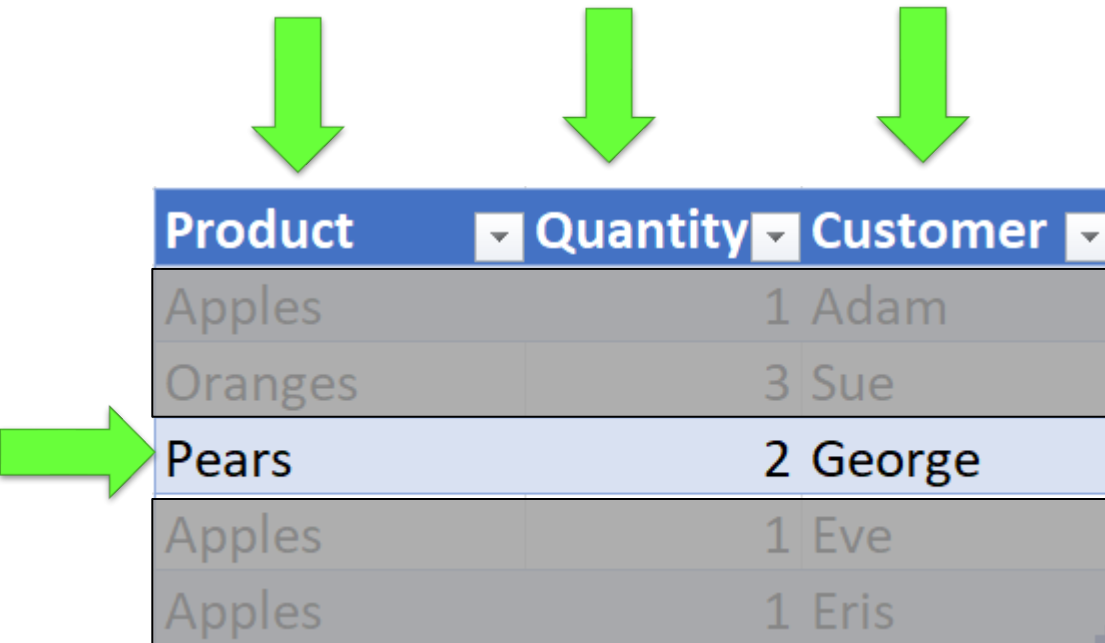
Product	Quantity	Customer
Apples	1	Adam
Oranges	3	Sue
Pears	2	George
Apples	1	Eve
Apples	1	Eris

A columnar format...

Great for this

```
SELECT SUM(Quantity)
GROUP BY Customer
```

What is Parquet



Product	Quantity	Customer
Apples	1	Adam
Oranges	3	Sue
Pears	2	George
Apples	1	Eve
Apples	1	Eris

A columnar format...

Not so great for this:

```
UPDATE (Product, Quantity)
WHERE Customer='George'
```

What is Parquet



Product	Quantity	Customer
Apples	(6)	1 Adam
Oranges	(7)	3 Sue
Pears	(5)	2 George
Apples	(6)	1 Eve
Apples	(6)	1 Eris

Product Size = 30 characters (bytes)

..., dictionary encoded, ...

Let

Apples = 1

Oranges = 2

Pears = 3

What is Parquet



Product	Quantity	Customer
1	1	Adam
2	3	Sue
3	2	George
1	1	Eve
1	1	Eris

Product Size = 5x4 bytes = 20 bytes

..., dictionary encoded, ...

Let

Apples = 1

Oranges = 2

Pears = 3

What is Parquet



Product	Quantity	Customer
0b01	1	Adam
0b10	3	Sue
0b11	2	George
0b01	1	Eve
0b01	1	Eris

Product Size = 5x2 bits = 10 bits ~ 2 bytes

That's 15x smaller!

..., dictionary encoded, ...

2 bits are enough for 3 values!!! - BITPACKING

Let

Apples = 1 = 0b01

Oranges = 2 = 0b10

Pears = 3 = 0b11

What is Parquet

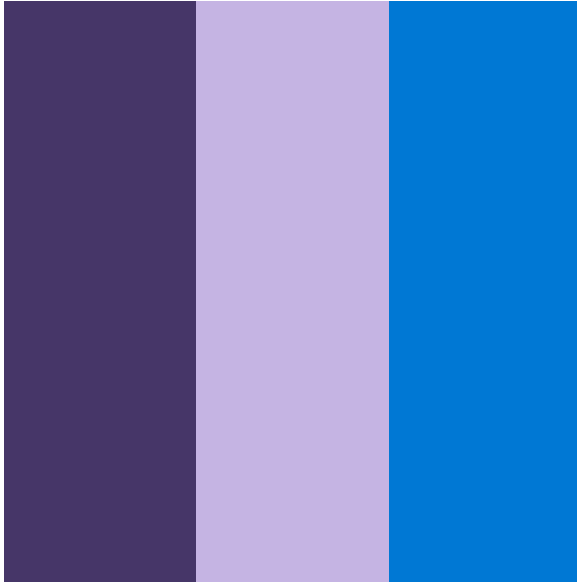
..., with RLE compression

RLE - Run-length encoding

(replace repeated occurrences with the count)

Product	Quantity	Customer
Apples	1	Adam
Oranges	3	Sue
Pears	2	George
Apples	1	Eve
Apples	1	Eris

What is Parquet - conclusion



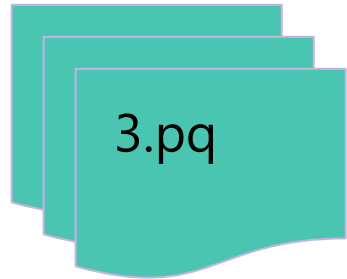
- Columnar
- Compressed/Dictionaryes
- RLE



Dictionaries

Delta | tables are more than files

Table folder: Parquet files



delta_log (snapshots)

- 0.json - Add 1.pq
- 1.json - Add 2.pq
- 2.json - Remove 1.pq/Add 3.pq

Operation

- INSERT INTO ...
- INSERT INTO ...
- UPDATE/DELETE ...
- ...
- ...
- <compaction>

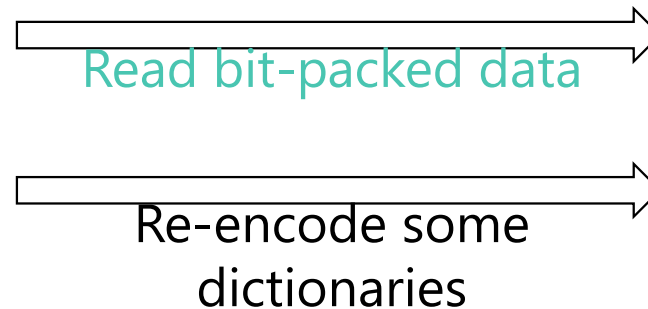
So, what is parquet



- A columnar format
- With dictionary encoding
- bitpacking
- RLE compression

- Just like Power BI's Vertipaq format!

- (well, there are a few differences)



One encoding to another | Transcoding

So, what is parquet



- A columnar format
- With dictionary encoding
- bitpacking
- RLE compression



- Just like Power BI's Vertipaq format!
- (well, there are a few differences)

Vertipaq can do a few more things


- Many operators (aggregations, filters, group by) operate directly on compressed data
- **Code ready to use** in Power BI and SQL (cloned as ColumnStore Index)

What else can we use from Power BI



**Operators over RLE are
REALLY fast**

How about rearranging the rows, to
facilitate RLE?



Product	Quantity	Customer
Apples	1	Adam
Oranges	3	Sue
Pears	2	George
Apples	1	Eve
Apples	1	Eris

What else can we use from Power BI



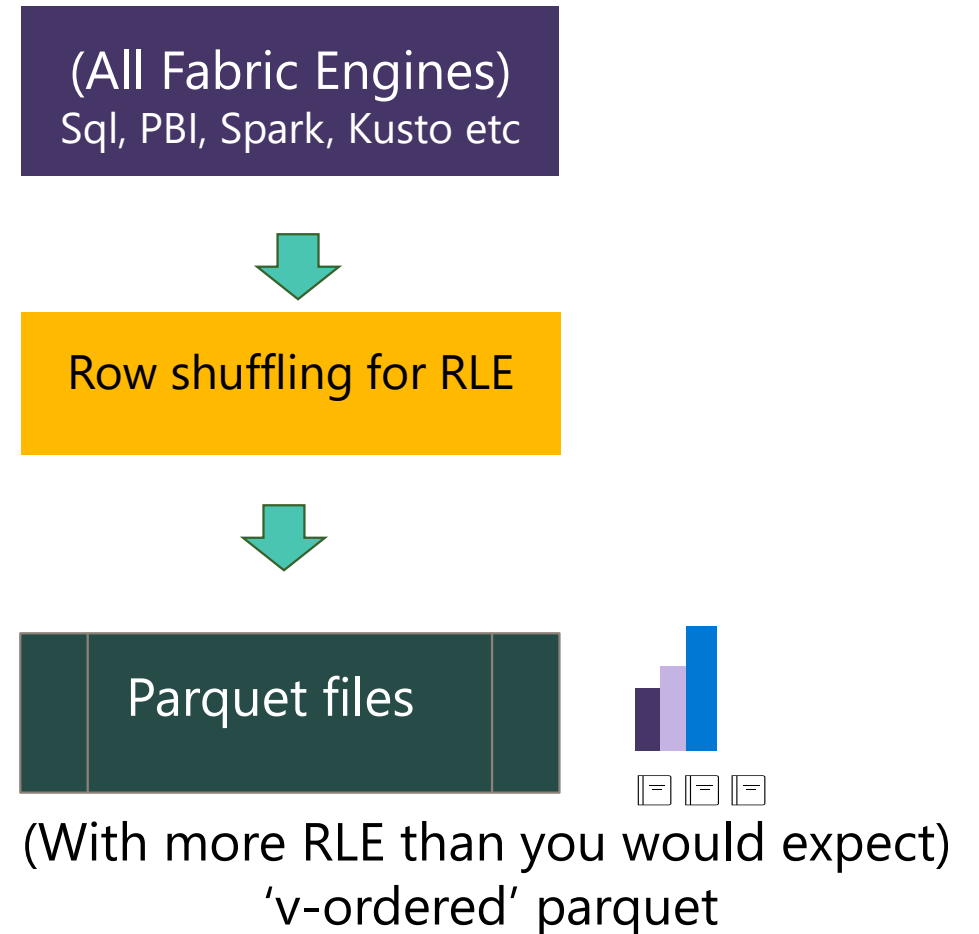
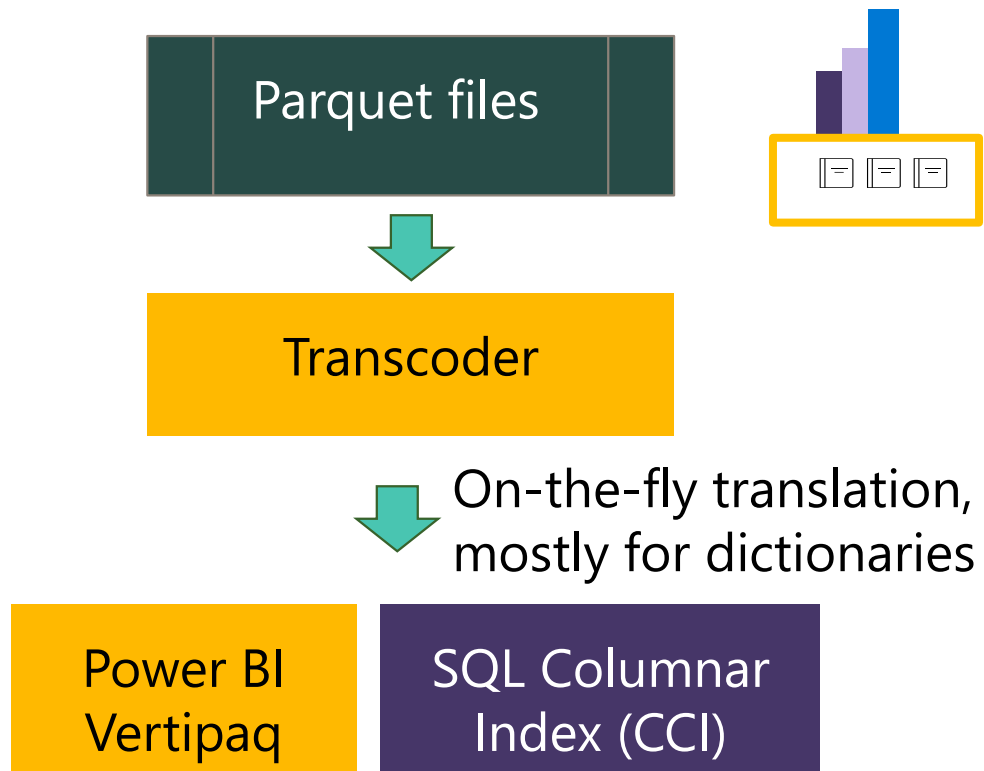
Product	Quantity	Customer
Pears	2	George
Oranges	3	Sue
Apples: <u>3</u>	1: <u>3</u>	Adam
		Eve
		Iris

Operators over RLE are
REALLY fast

How about rearranging the rows, to
facilitate RLE?

SMALLER! FASTER! BETTER!

Processing data | conclusion

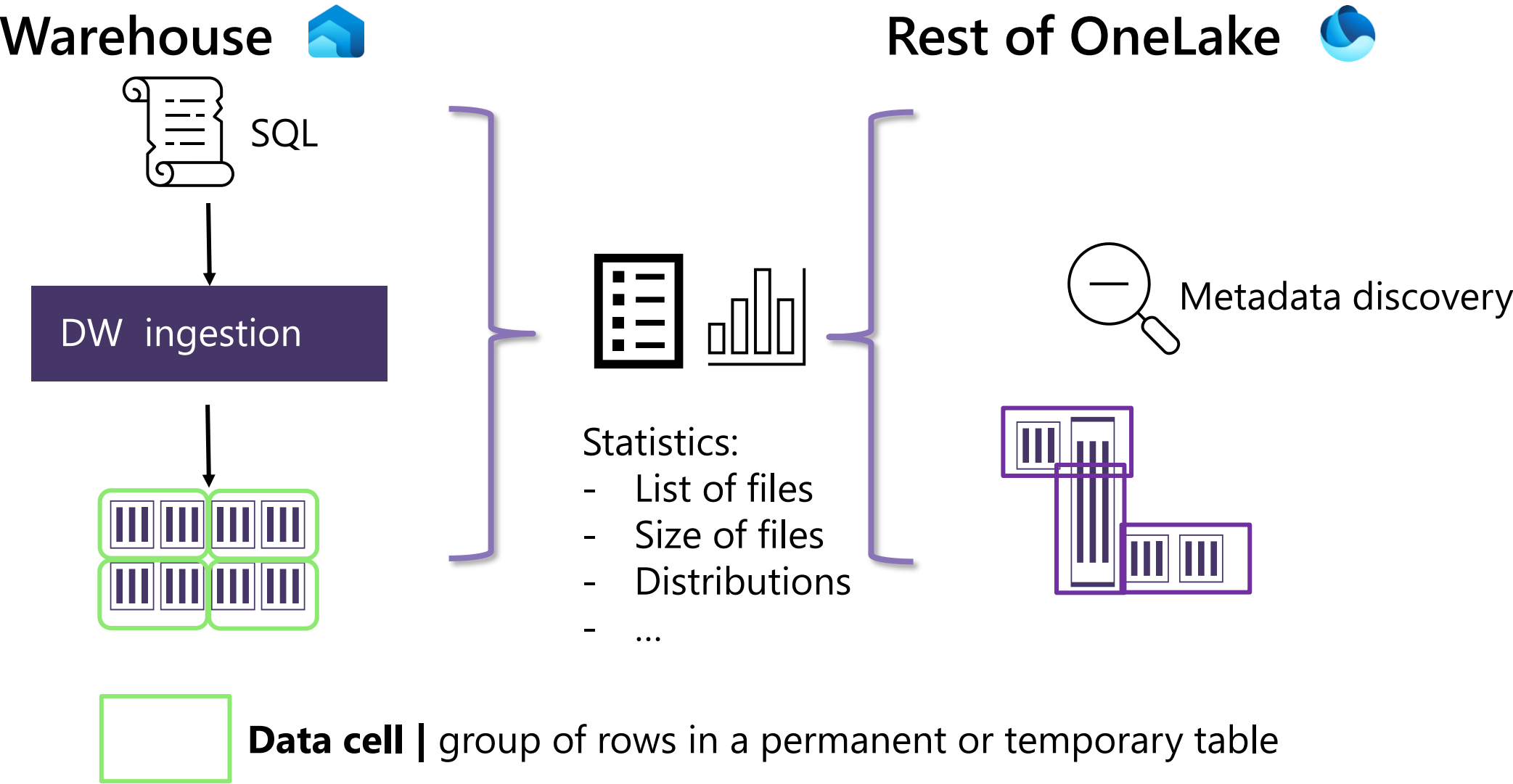


So, we'll talk about...

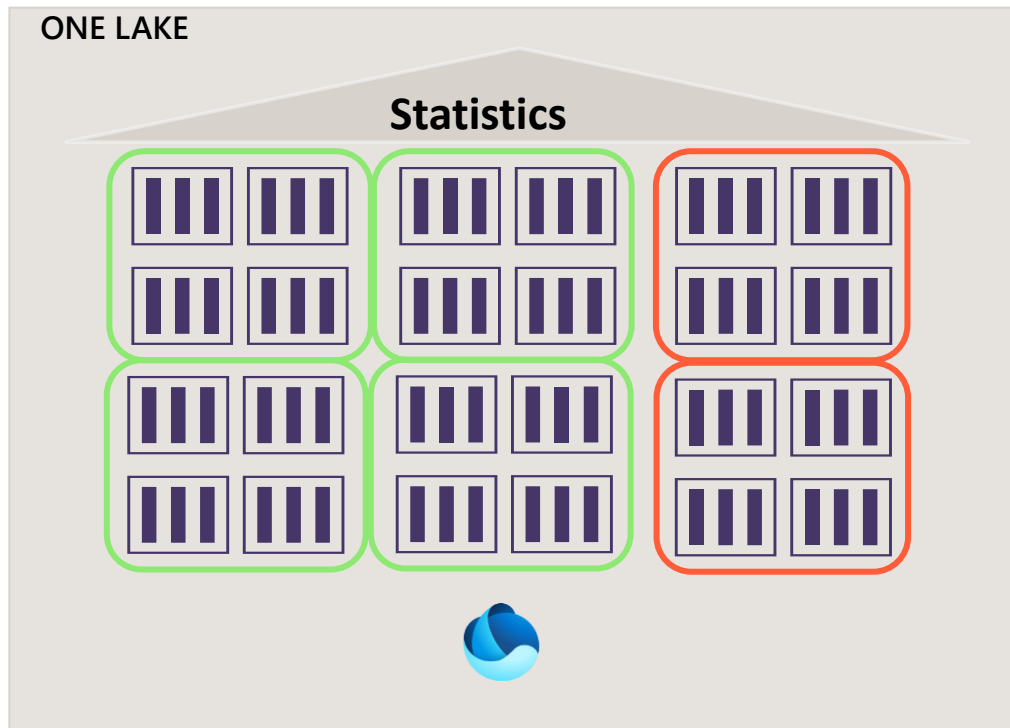
- ✓ ~~Query processing~~ reduced to existing code (CCI)
- ☐ Query optimization
- ☐ Multi-node distribution

Planning and executing the queries

Learn about data in OneLake



Unified view of all data in OneLake



Statistics

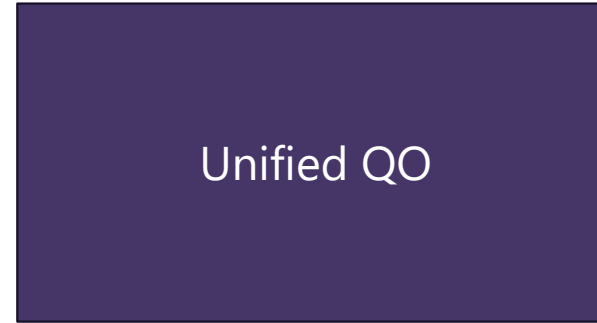
- Distributions in cells
- Estimate operator cost
- Efficient cell skipping

Awareness of distributions

- z-ordering
- Hash distribution
- Hive style partitions

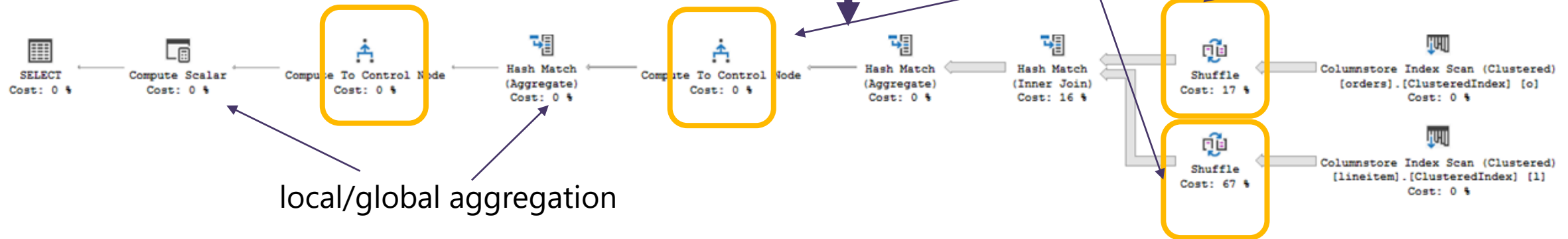
Query Lifecycle – Compilation

```
SELECT COUNT_BIG (*)  
FROM ORDERS AS o, LINEITEM AS l  
ON o.O_ORDERKEY = l.L_ORDERKEY
```



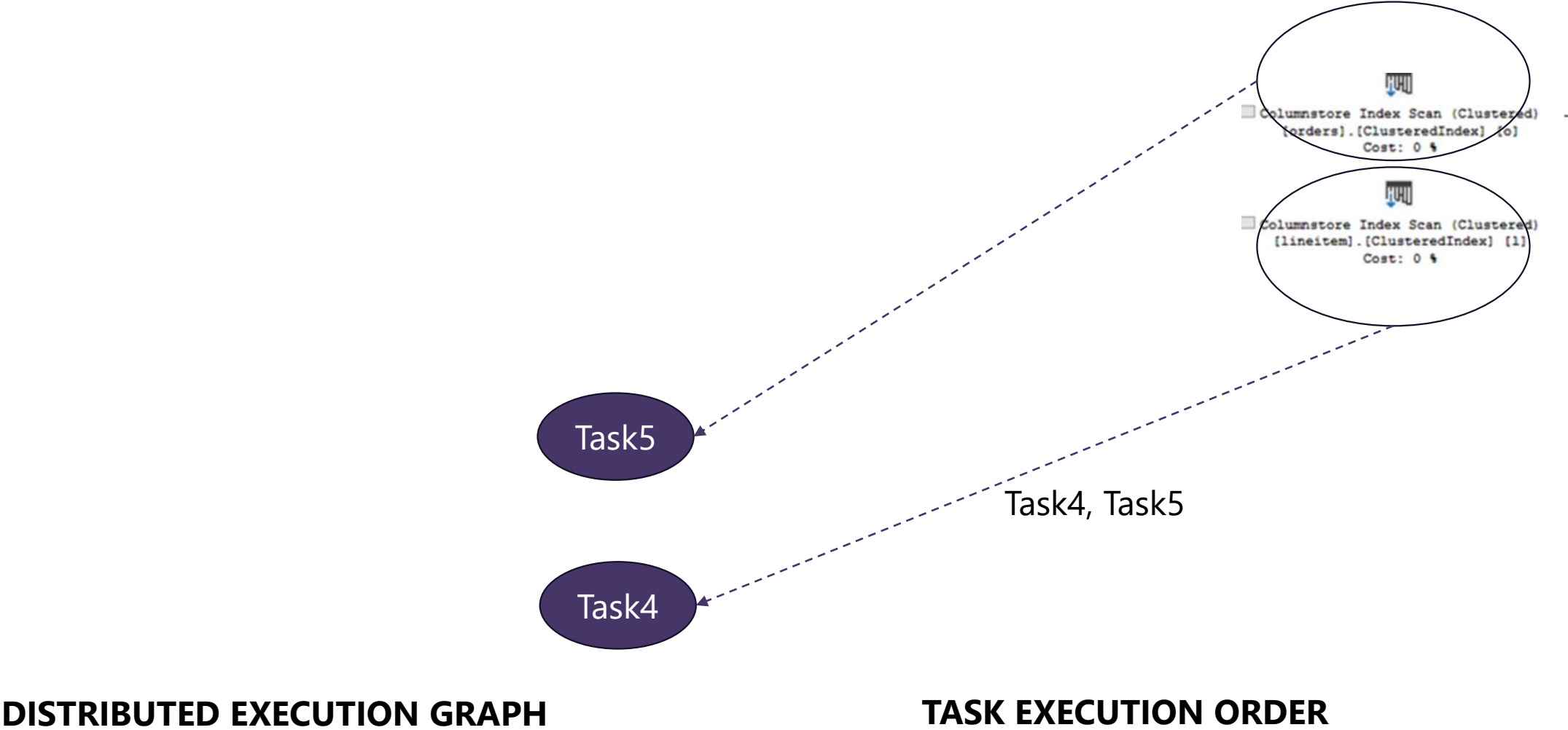
QUERY

data movement operators



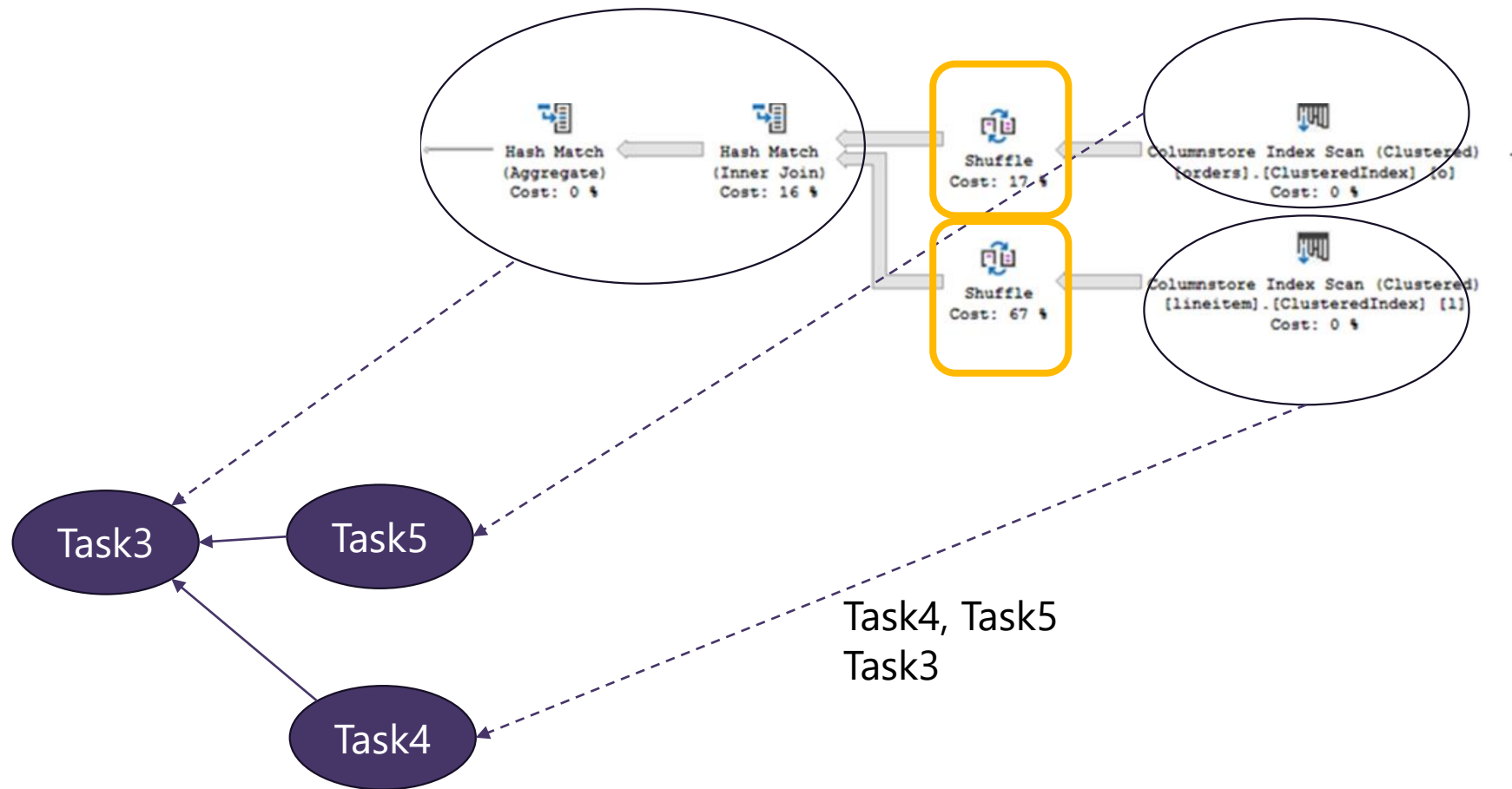
DISTRIBUTED QUERY EXECUTION PLAN

Query Lifecycle – Distributed Execution Plan



Query Lifecycle – Distributed Execution Plan

JERY EXECUTION PLAN

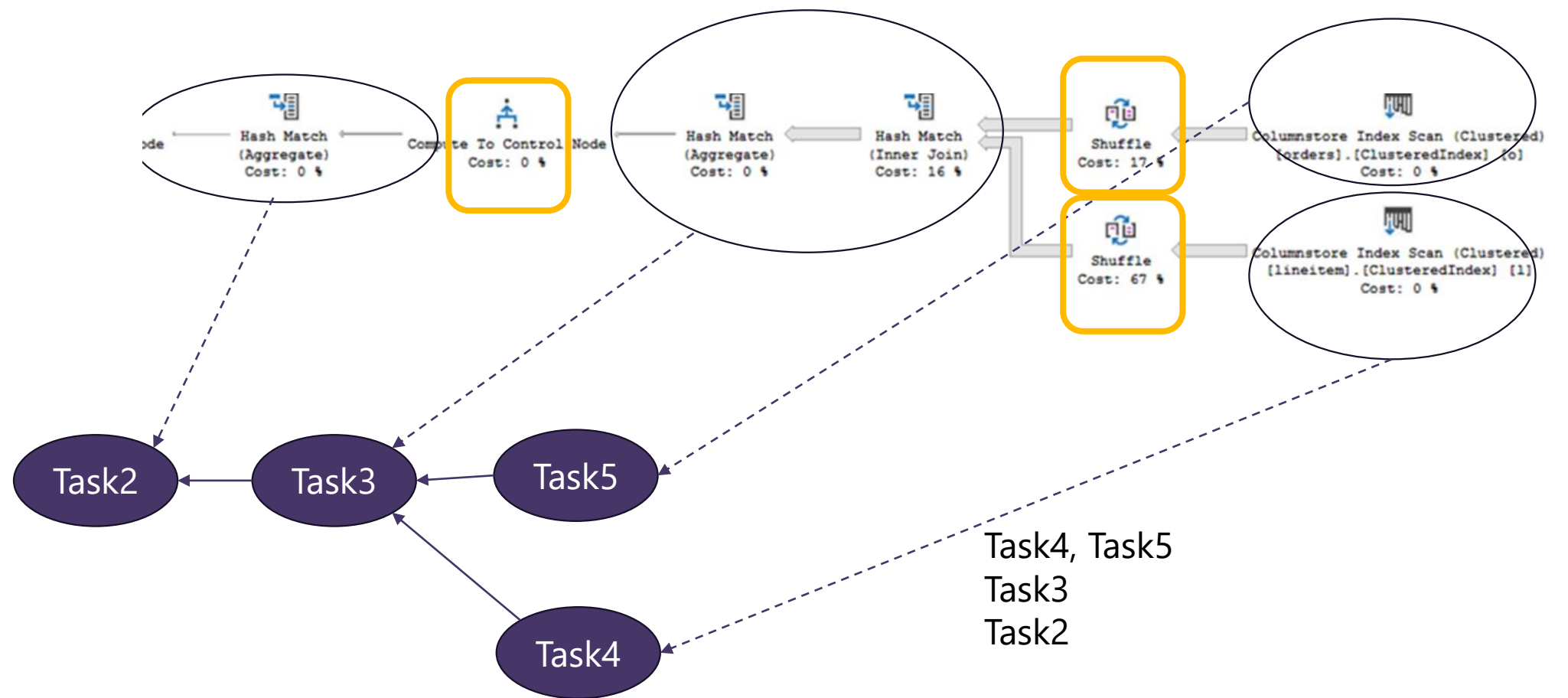


DISTRIBUTED EXECUTION GRAPH

TASK EXECUTION ORDER

Query Lifecycle – Distributed Execution Plan

DISTRIBUTED QUERY EXECUTION PLAN

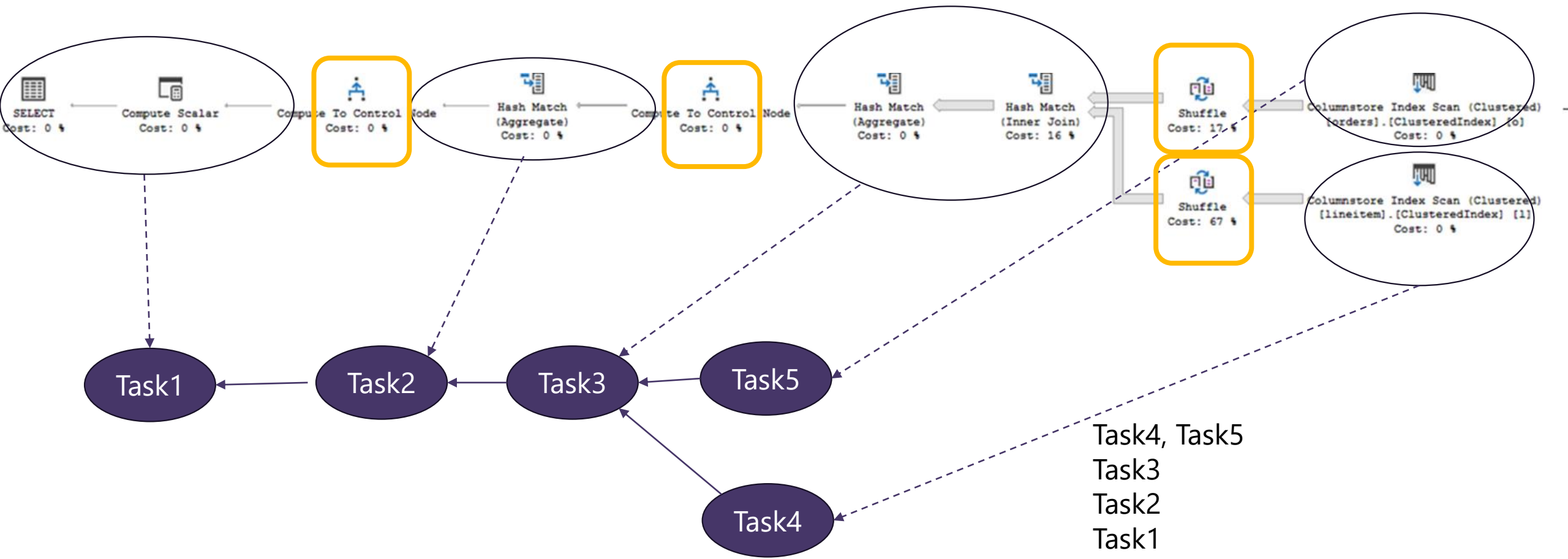


DISTRIBUTED EXECUTION GRAPH

TASK EXECUTION ORDER

Query Lifecycle – Distributed Execution Plan

DISTRIBUTED QUERY EXECUTION PLAN



DISTRIBUTED EXECUTION GRAPH

TASK EXECUTION ORDER

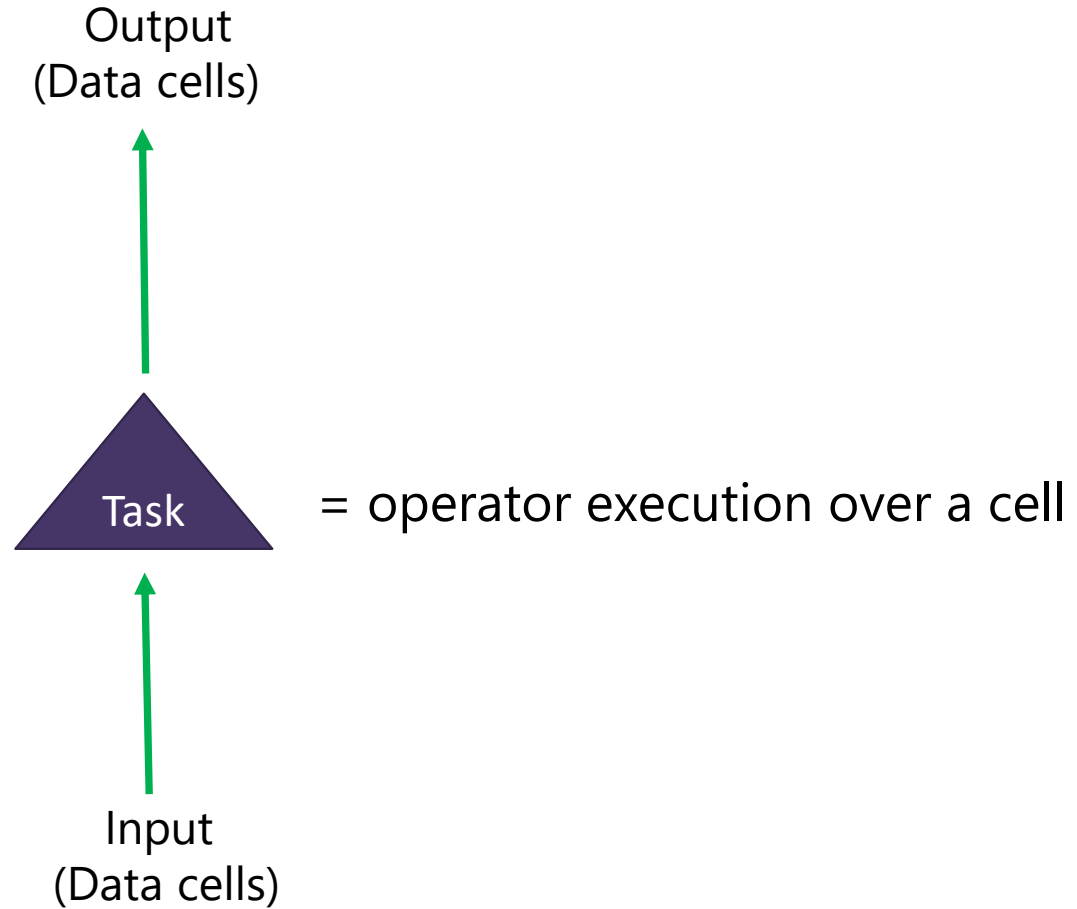
So, we'll talk about...

- ✓ ~~Query processing~~ reduced to existing code (CCI)
- ✓ ~~Query optimization~~ distributed Query Plans
- ❑ Multi-node distribution

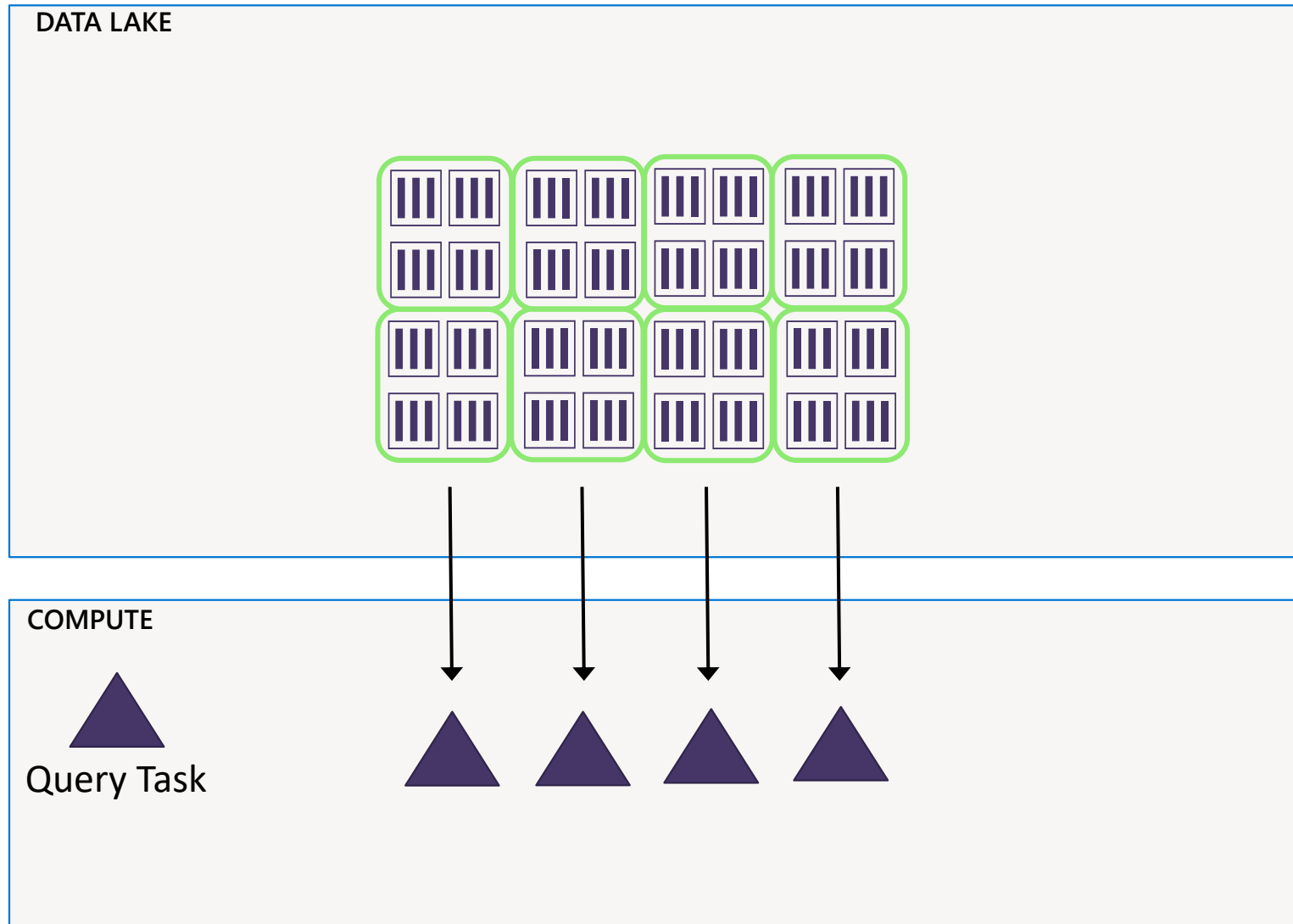
Polaris

Our distributed query processor (DQP)

Query operators | execution building blocks

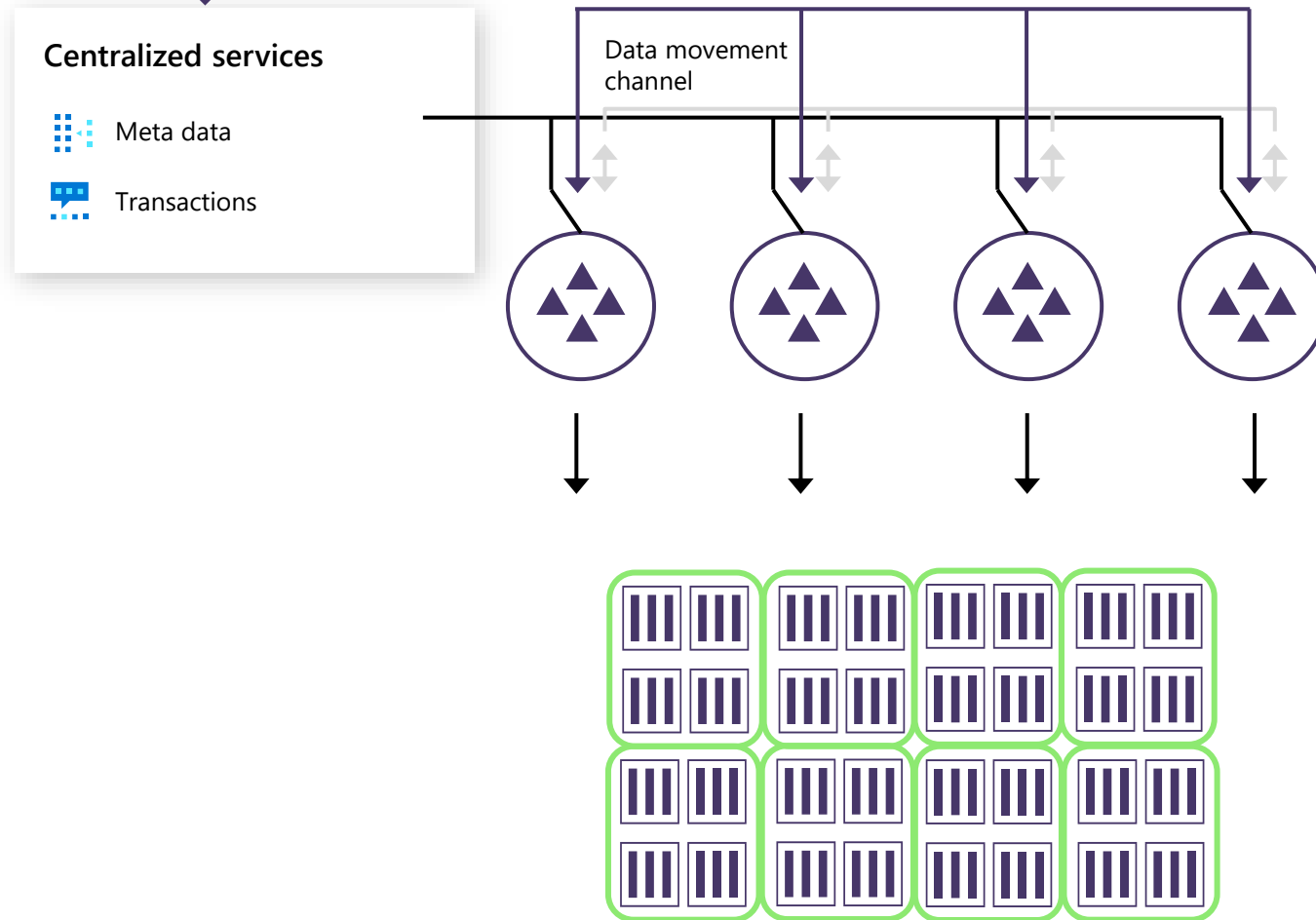


QO task | more data, more parallel tasks



Separation of Storage and Compute

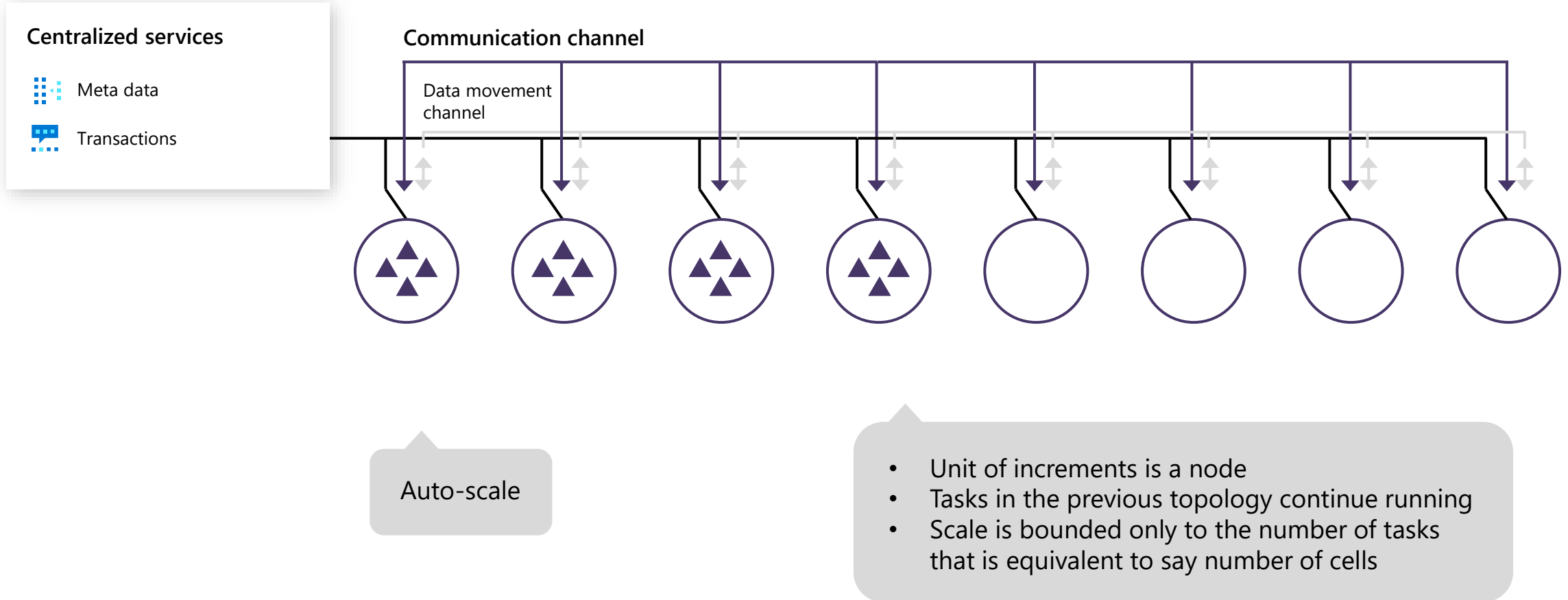
Query



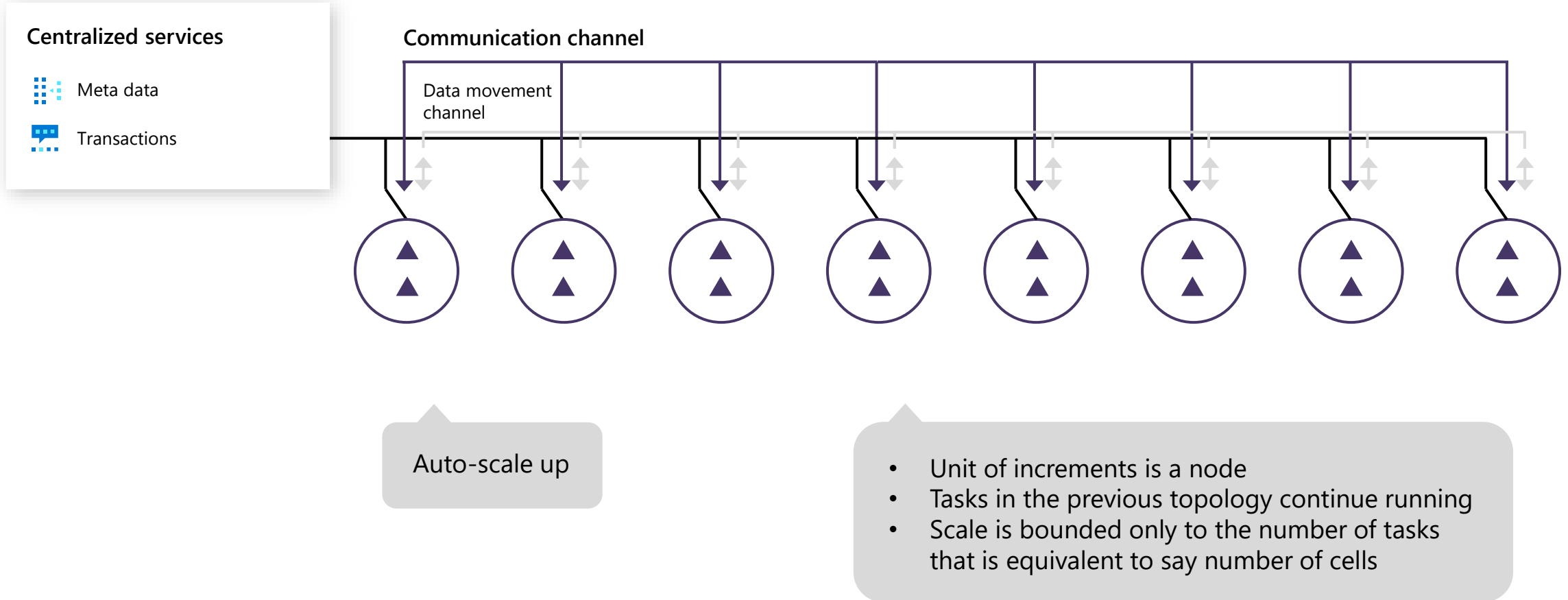
Central services:

- Select current data snapshot
- Compile query
- Use metadata/statistics to:
 - Estimate cost
 - Estimate # of nodes
- Launch execution

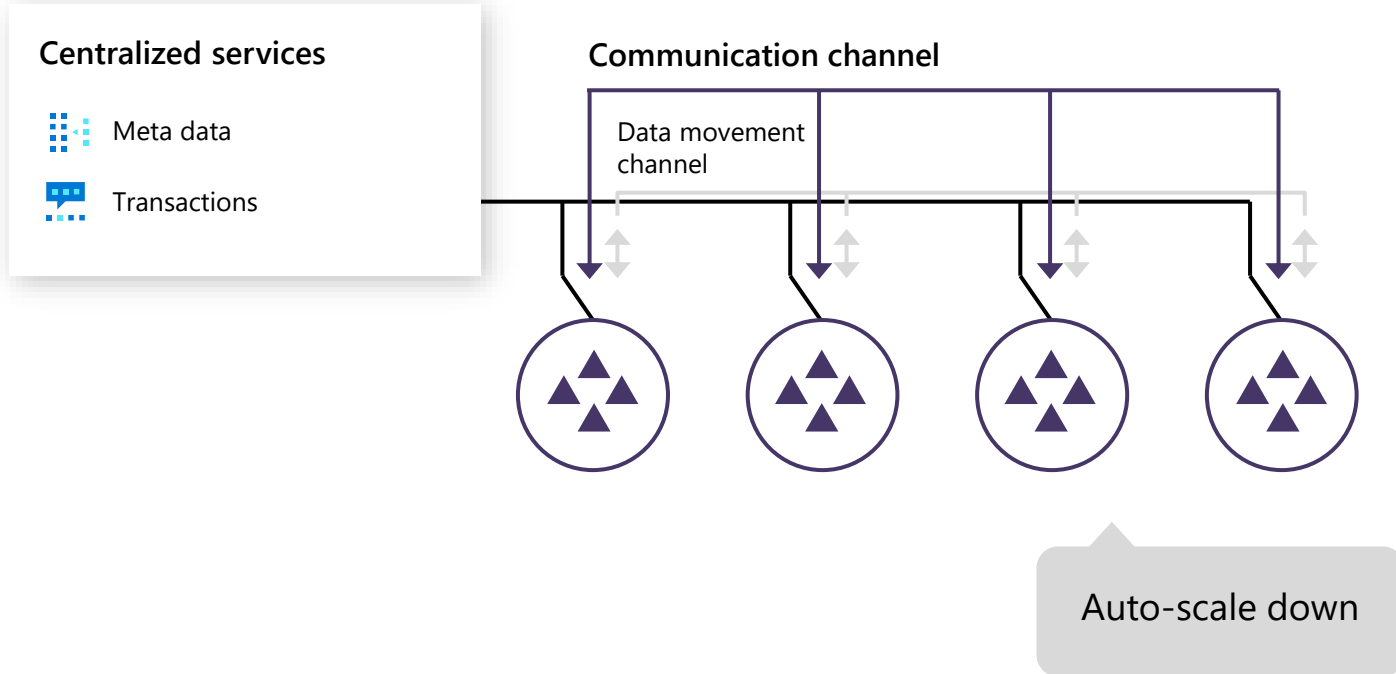
Polaris | Auto-scale



Polaris | Auto-scale



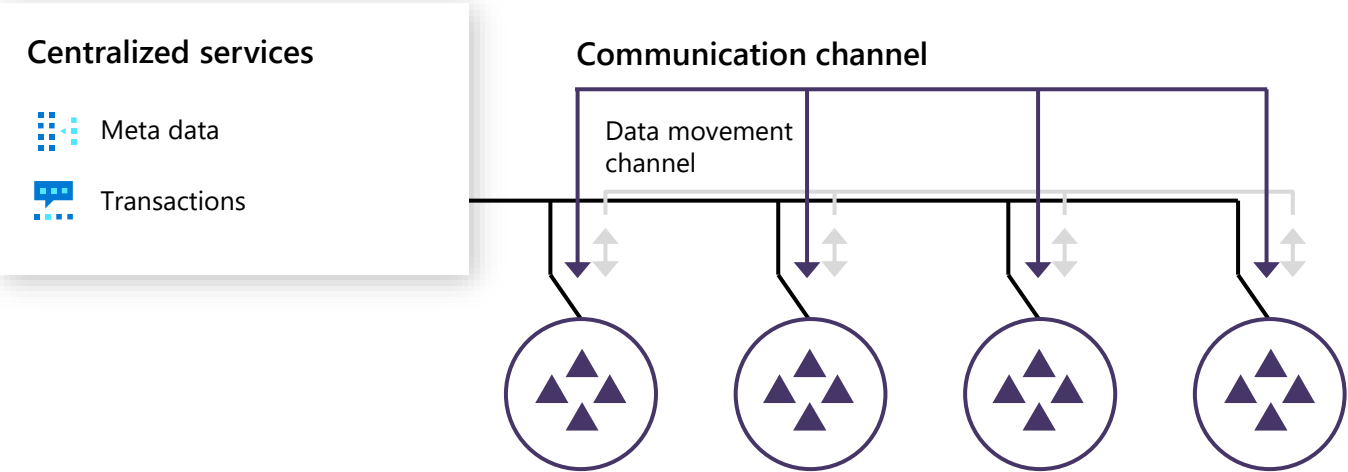
Polaris | Auto-scale



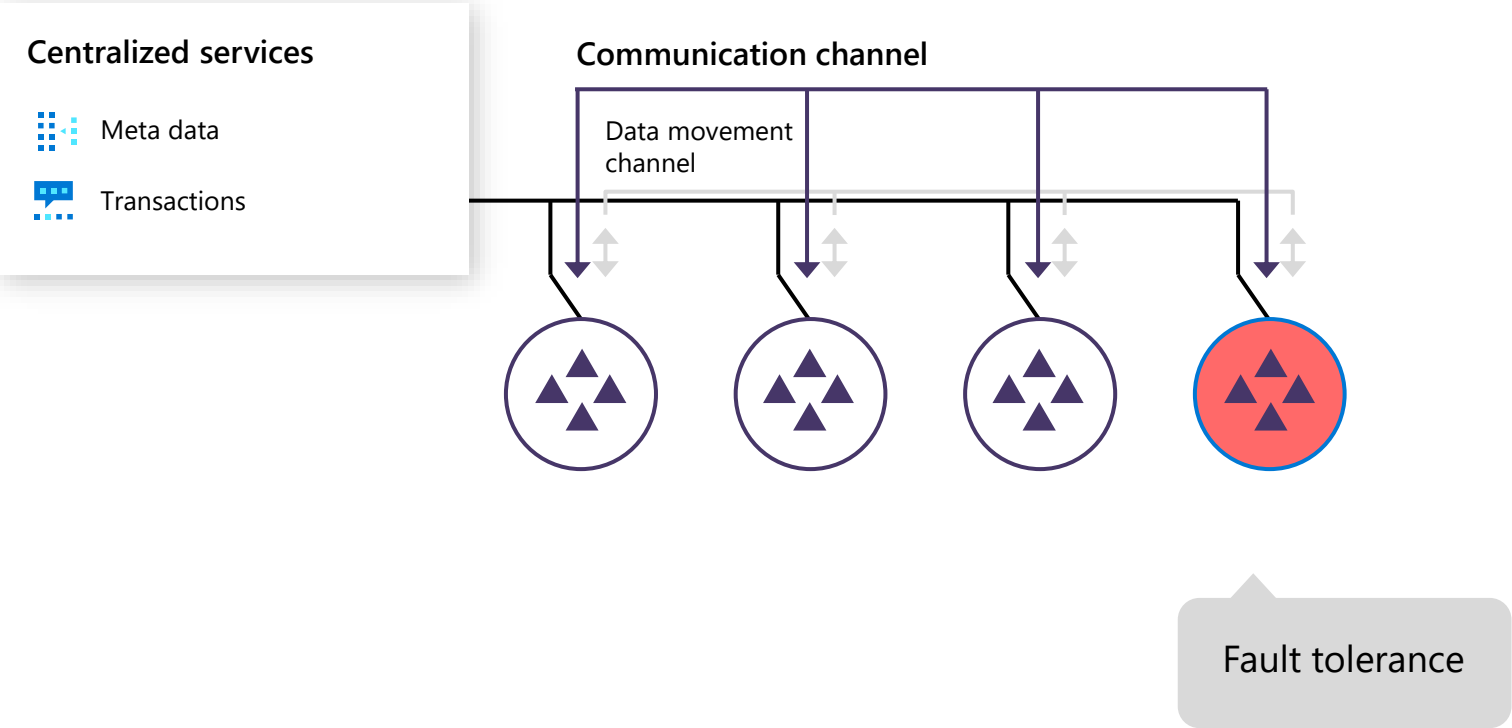
Customer value

Autonomous compute scaling that is transparent to the user

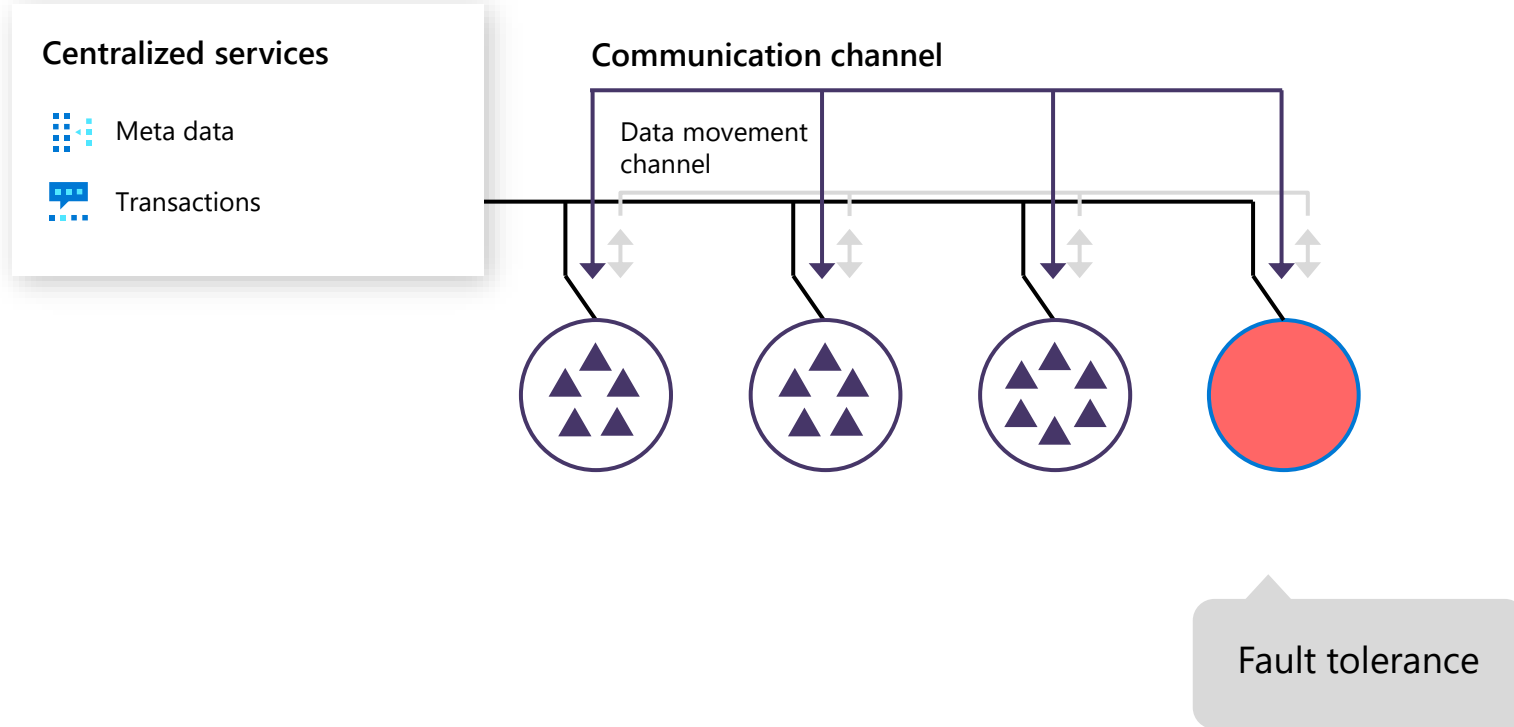
Polaris | Fault Tolerance



Polaris | Fault Tolerance

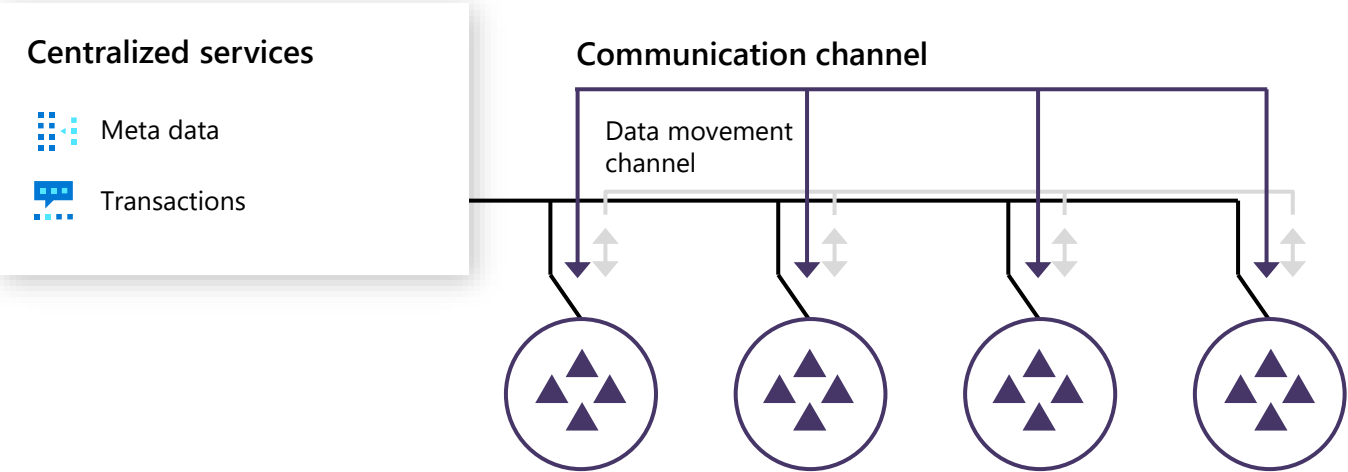


Polaris | Fault Tolerance

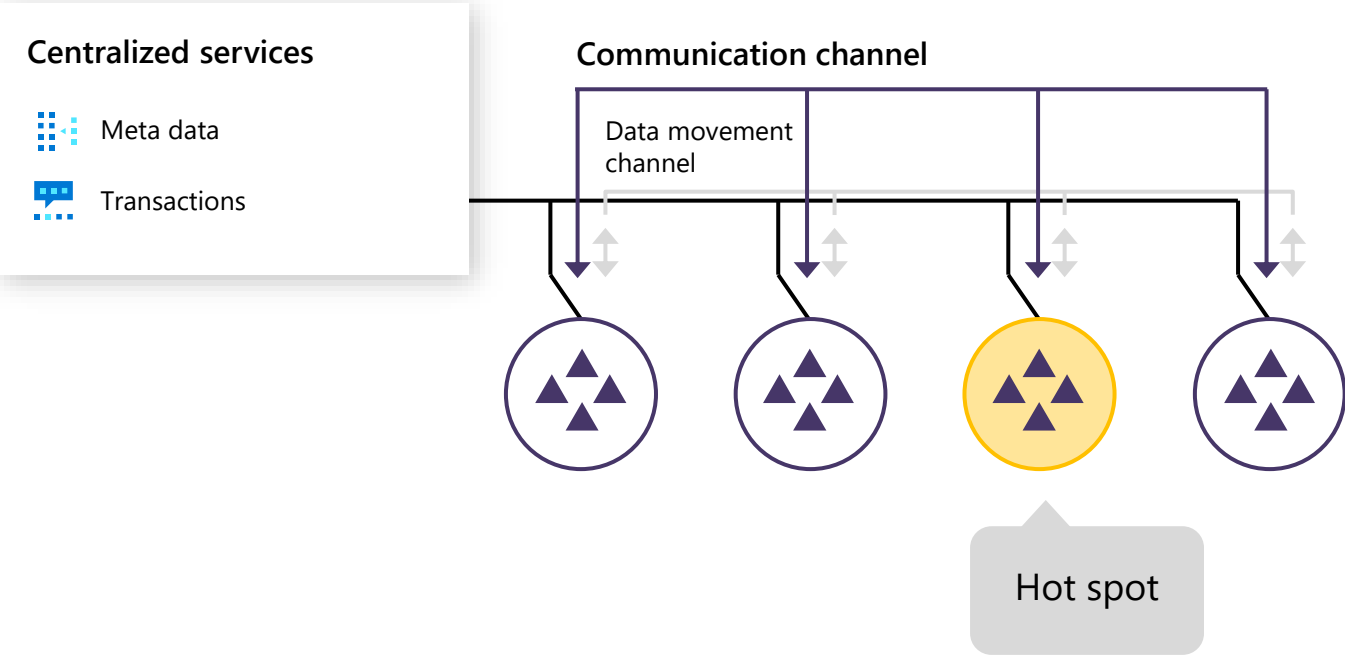


Customer value
Higher reliability

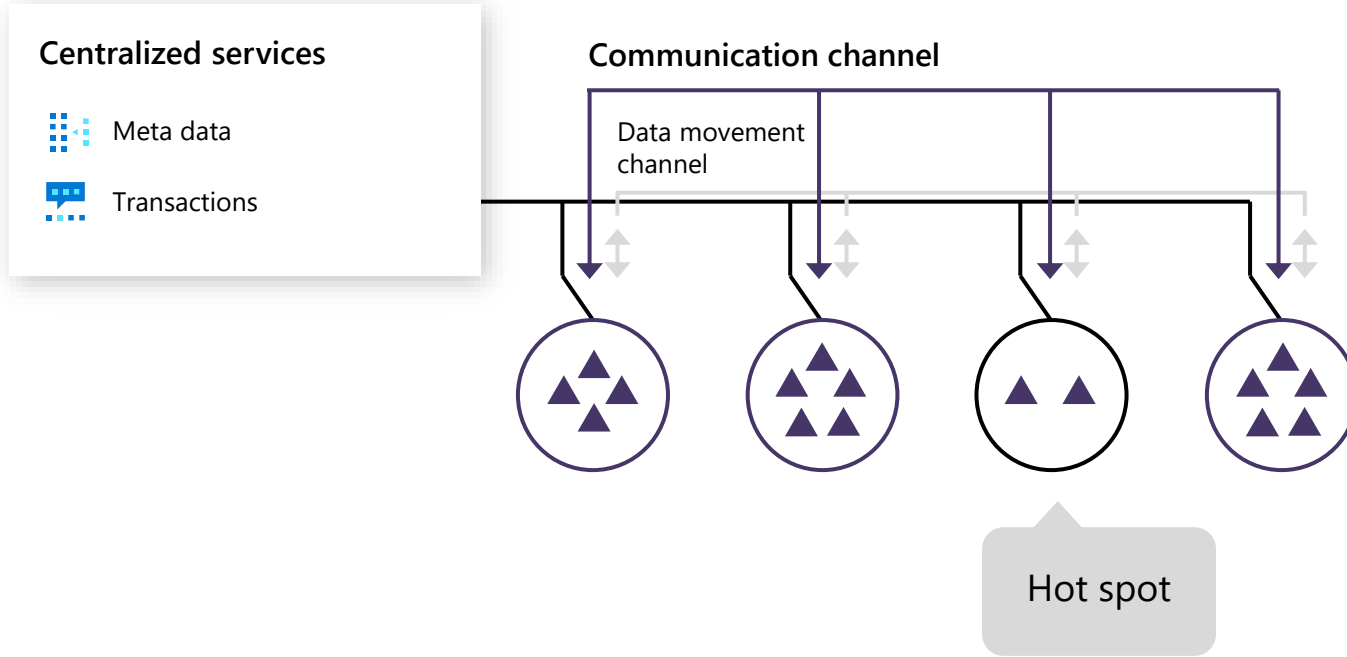
Polaris | Hot Spot



Polaris | Hot Spot



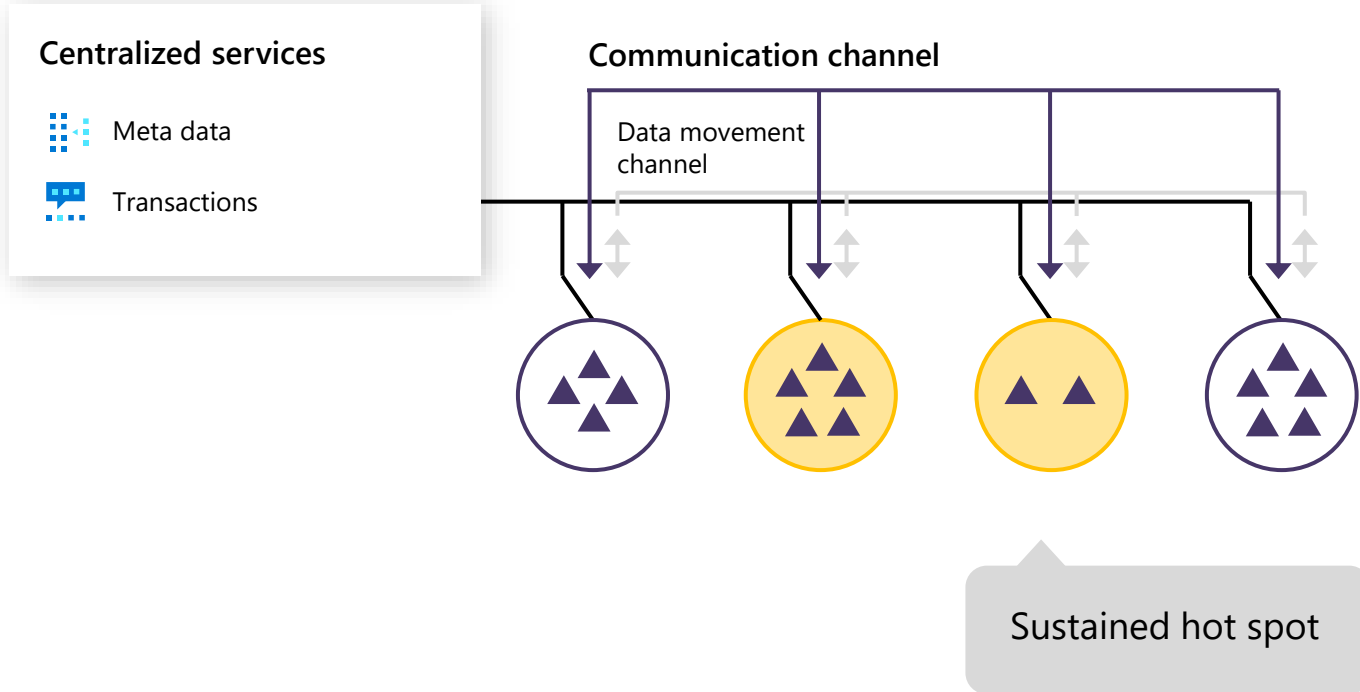
Polaris | Hot Spot



Customer value

Autonomous compute load balancing to maximize resource utilization

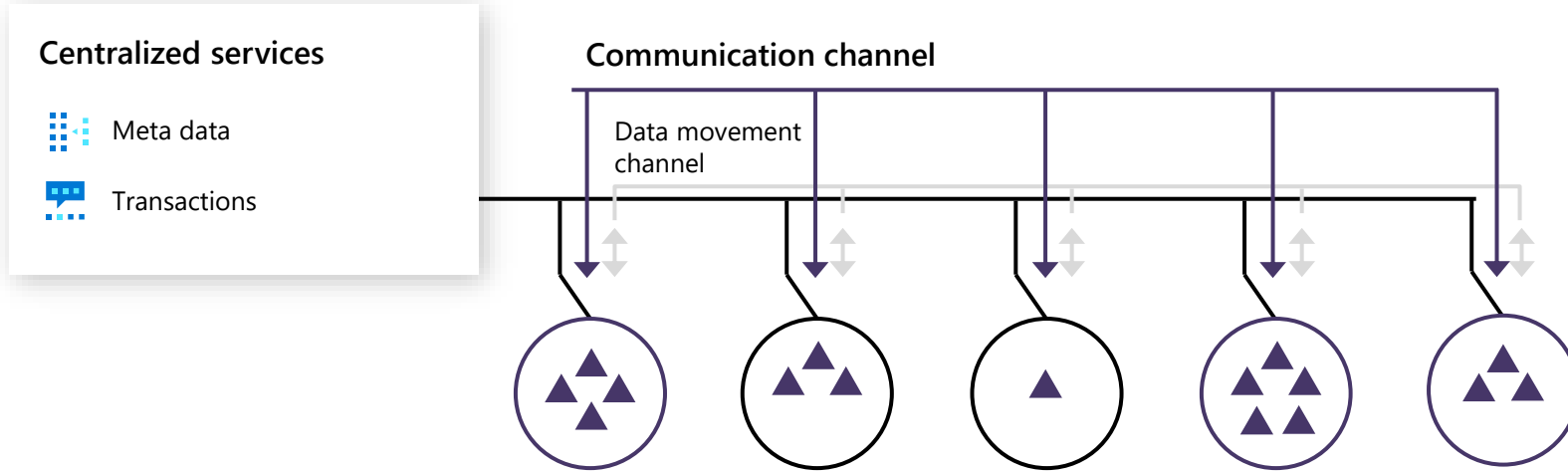
Polaris | Hot Spot



Customer value

Autonomous compute load balancing to maximize resource utilization

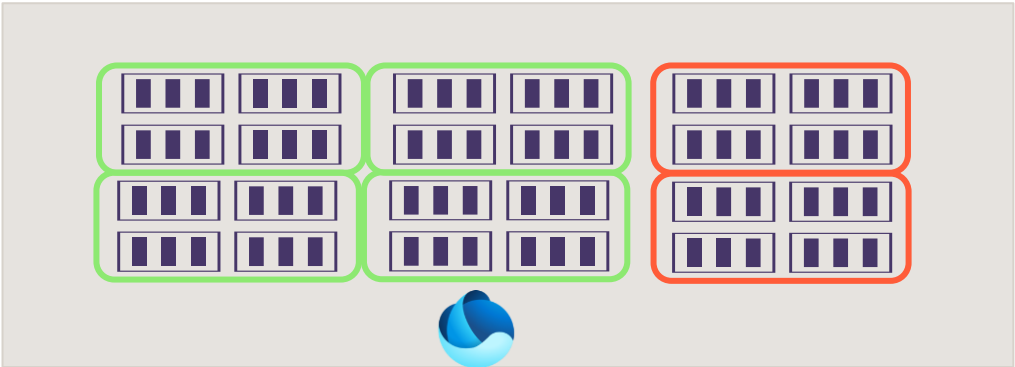
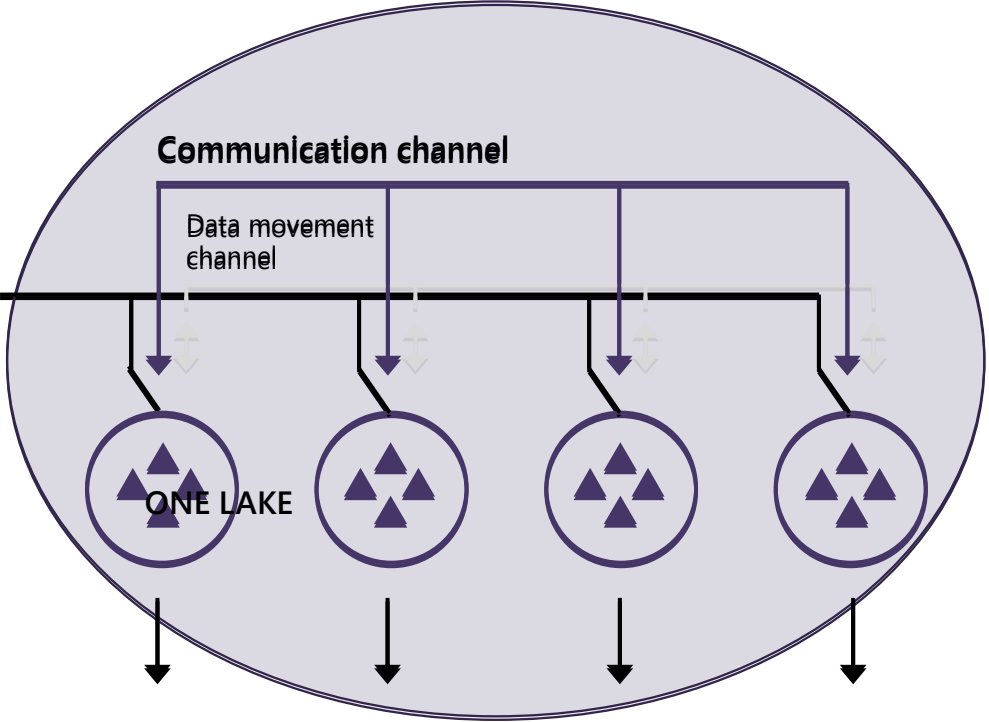
Polaris | Hot Spot



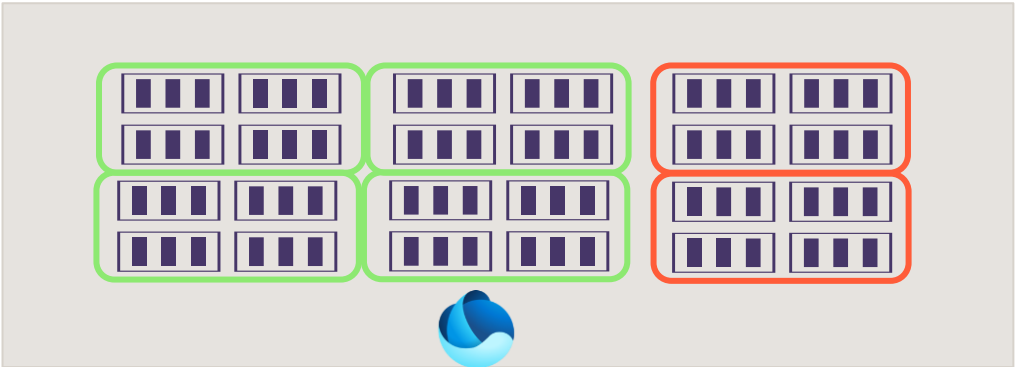
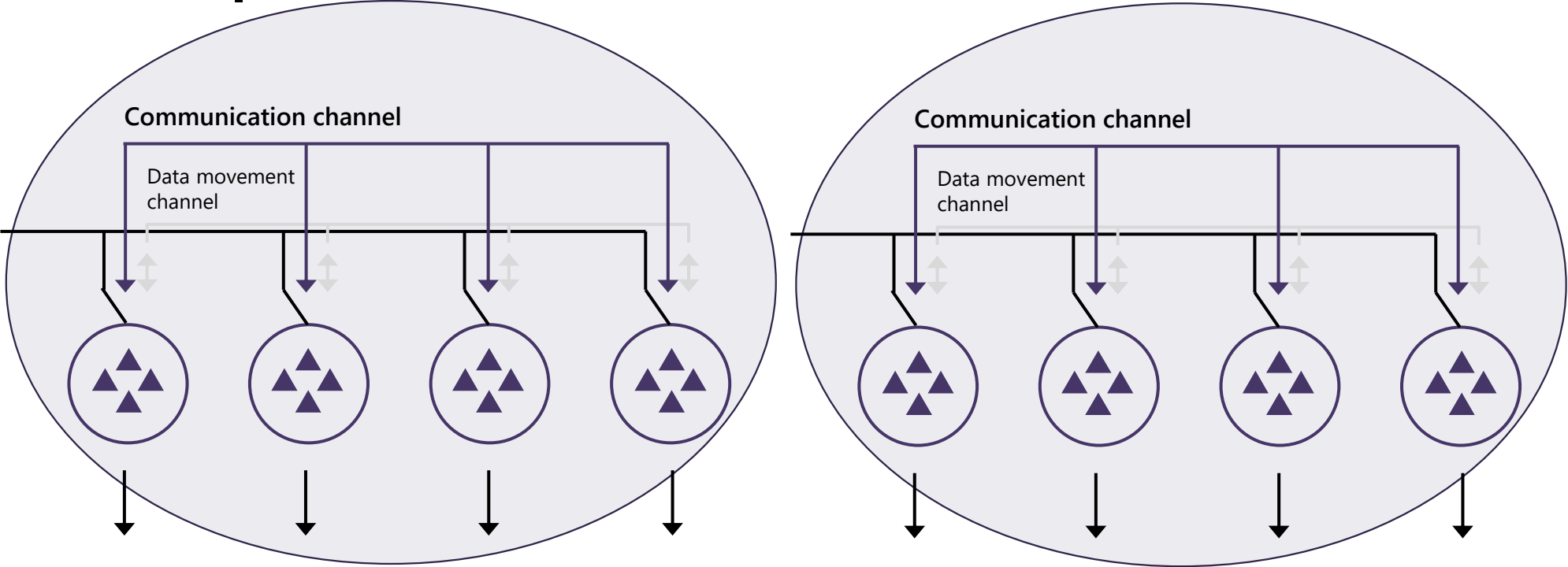
Customer value

Autonomous compute load balancing to maximize resource utilization

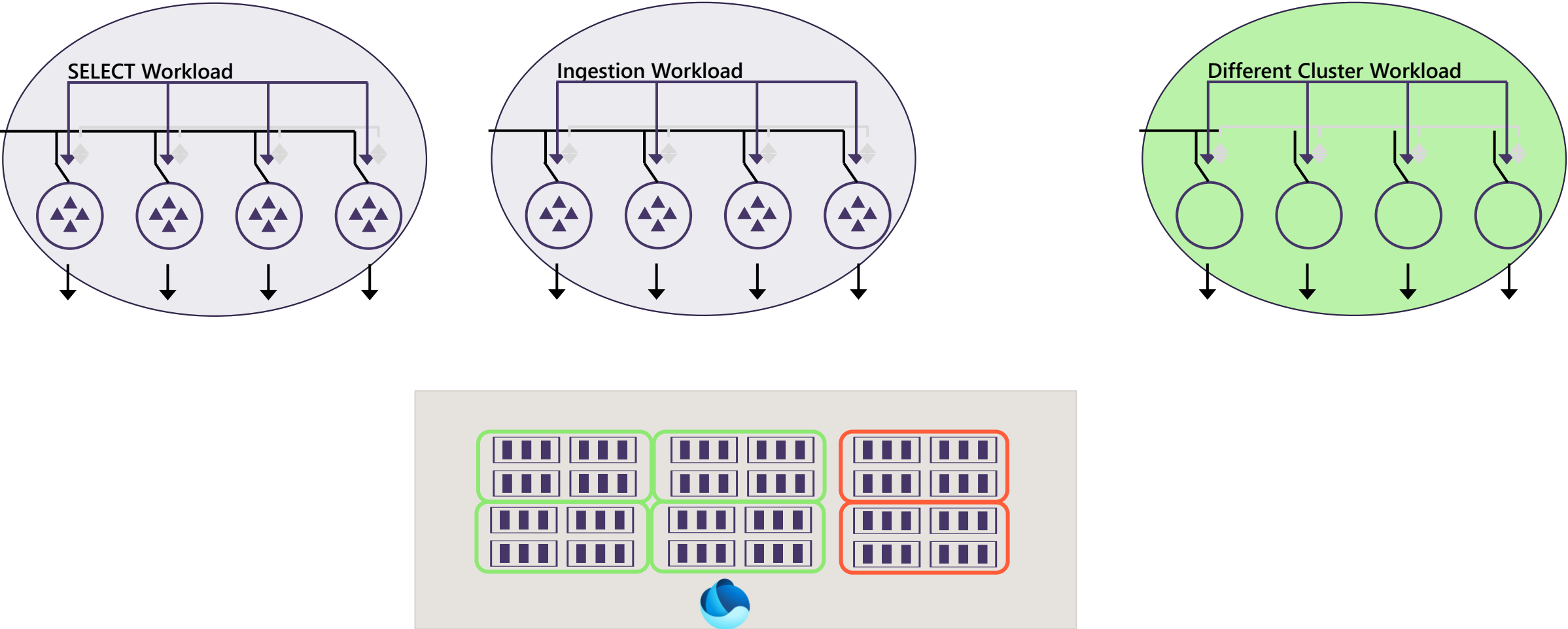
Polaris | Workloads



Polaris | Workloads



Polaris | Automatic Workload Classification

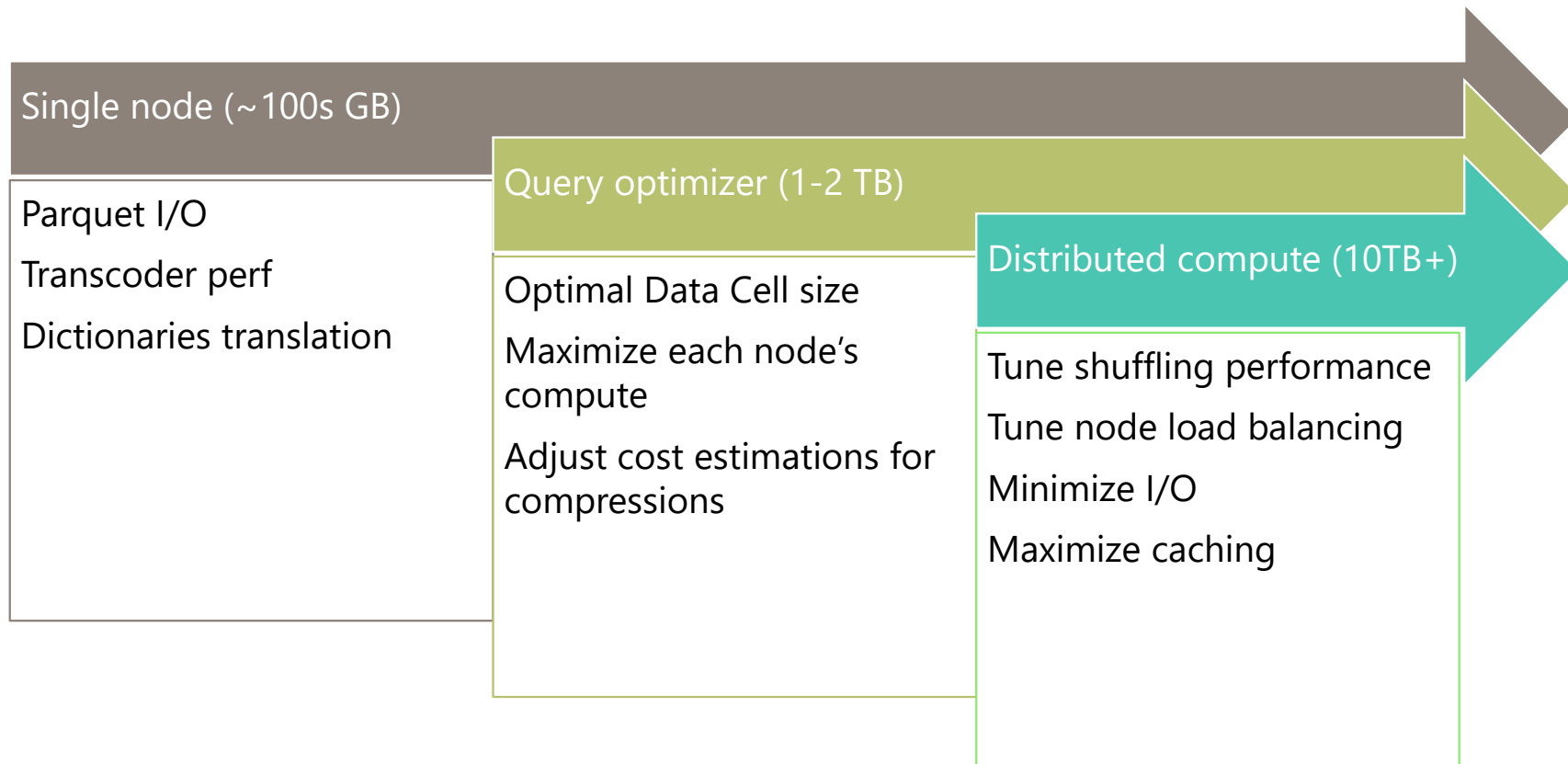


So, we'll talk about...

- ✓ ~~Query processing~~ reduced to existing code (CCI)
- ✓ ~~Query optimization~~ distributed Query Plans
- ✓ ~~Multi-node distribution~~ Polaris

Tuning for performance

Incremental process



Timeline for Synapse SQL in Fabric

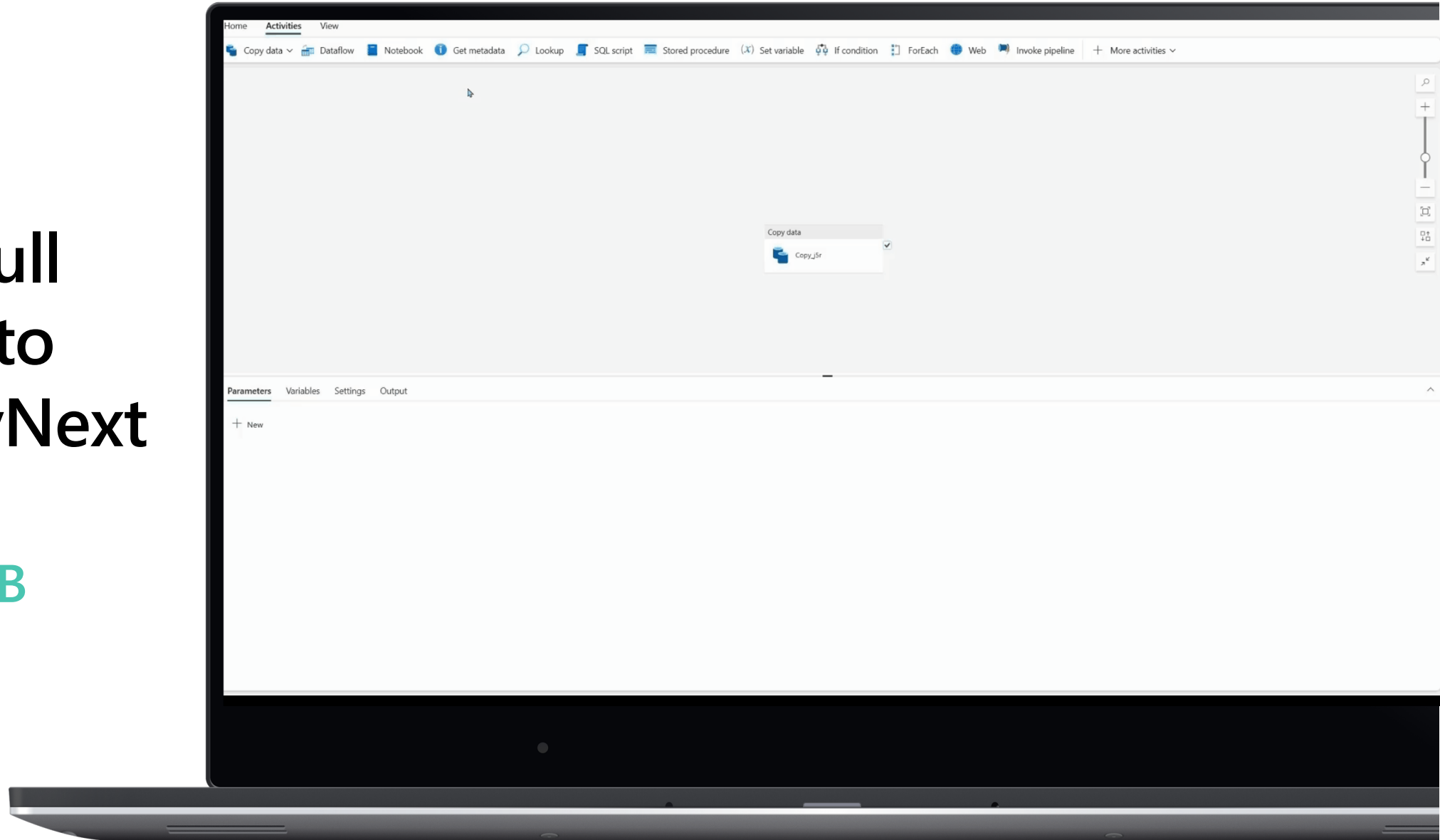
Private Preview (November 2022)	<ul style="list-style-type: none">• Single node• ~300 GB
Public Preview (now)	<ul style="list-style-type: none">• 1-10 TB
+3 Months	<ul style="list-style-type: none">• 10 -30 TB
+6 Months	<ul style="list-style-type: none">• 30 TB+ (likely <100 TB)
Later	<ul style="list-style-type: none">• ...

Demo

Everything in action

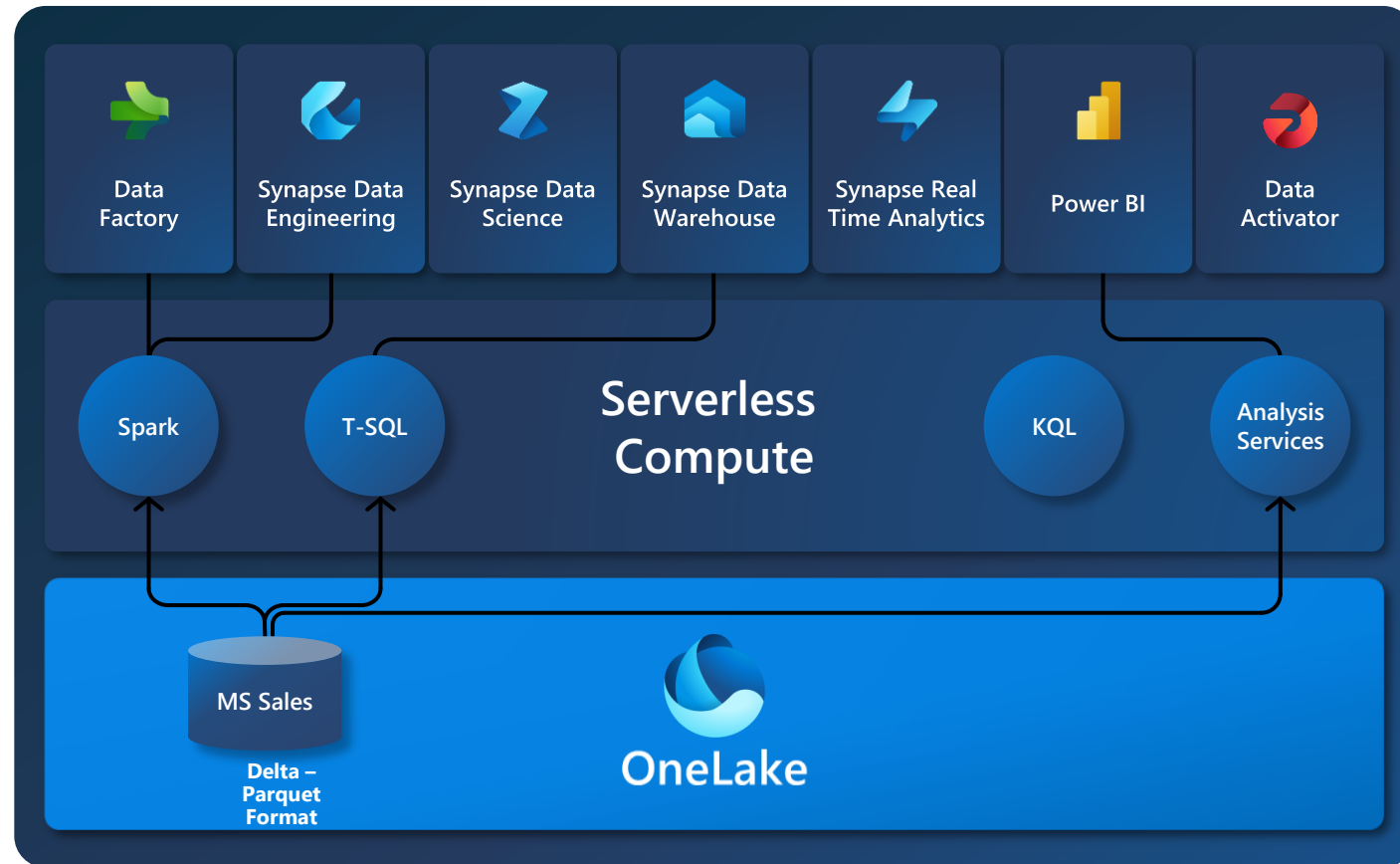
Loading full MS Sales to Synapse vNext

15 minutes
to load 20 TB



One Copy for all computes

Using the Lakehouse



**Demo / recording
or live**



Microsoft Fabric

Data analytics for the era of AI



Synapse SQL



OneLake
Delta – Parquet

Thank you

