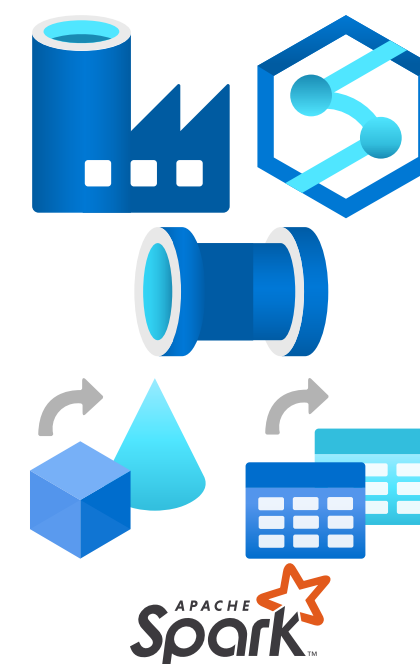


# ETL in Azure Made Easy

With Integration Pipeline Data Flows



Paul Andrew | Group Manager & Analytics Architect



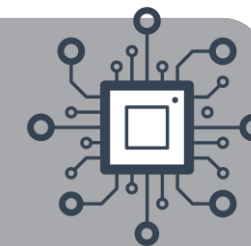
@MrPaulAndrew



In/MrPaulAndrew



MrPaulAndrew.com



<https://github.com/mrpaulandrew>

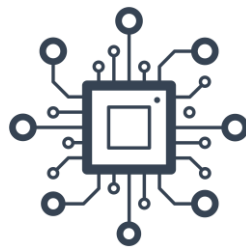
### CommunityEvents



Demo code, content and slides from various community events.

● C++

[{Event/Location}-{Month}-{Year}](#)

# Terminology Clarification










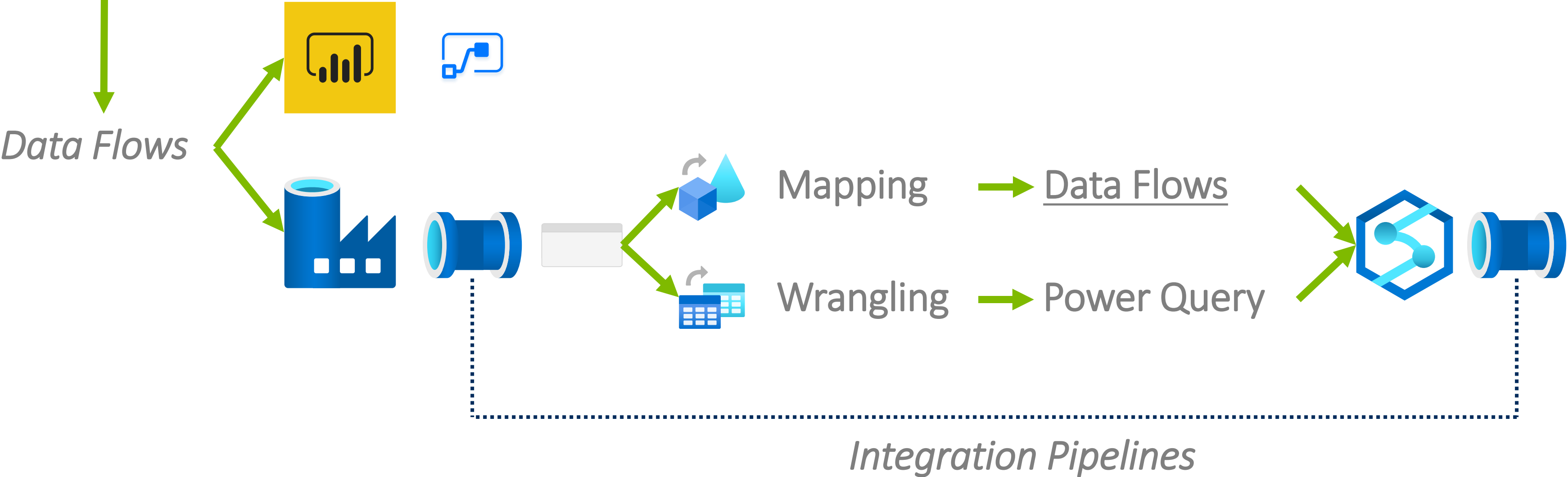
ETL in Azure Made Easy

With Integration Pipeline Data Flows

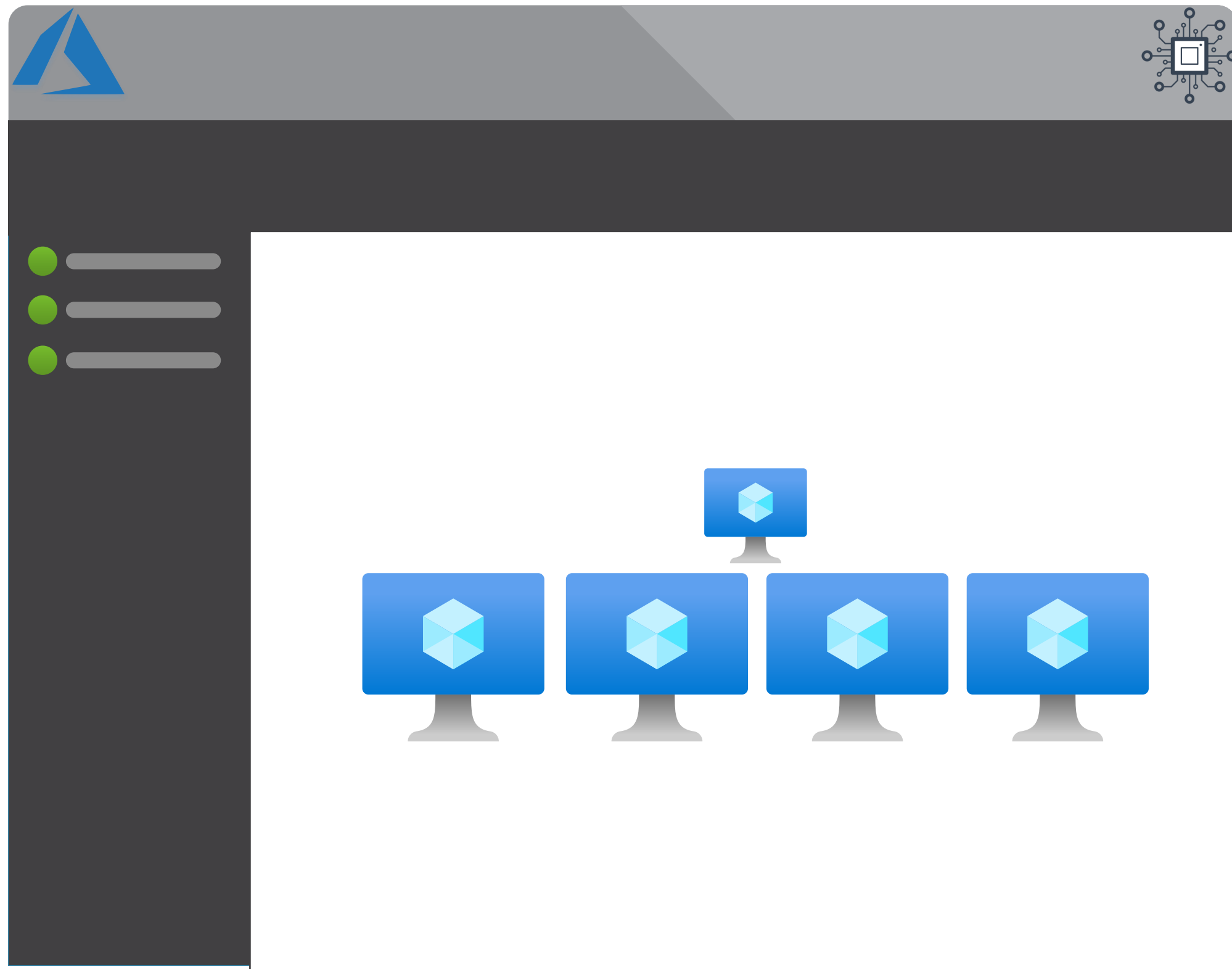
Paul Andrew | Group Manager & Analytics Architect

 @MrPaulAndrew  In/MrPaulAndrew  MrPaulAndrew.com

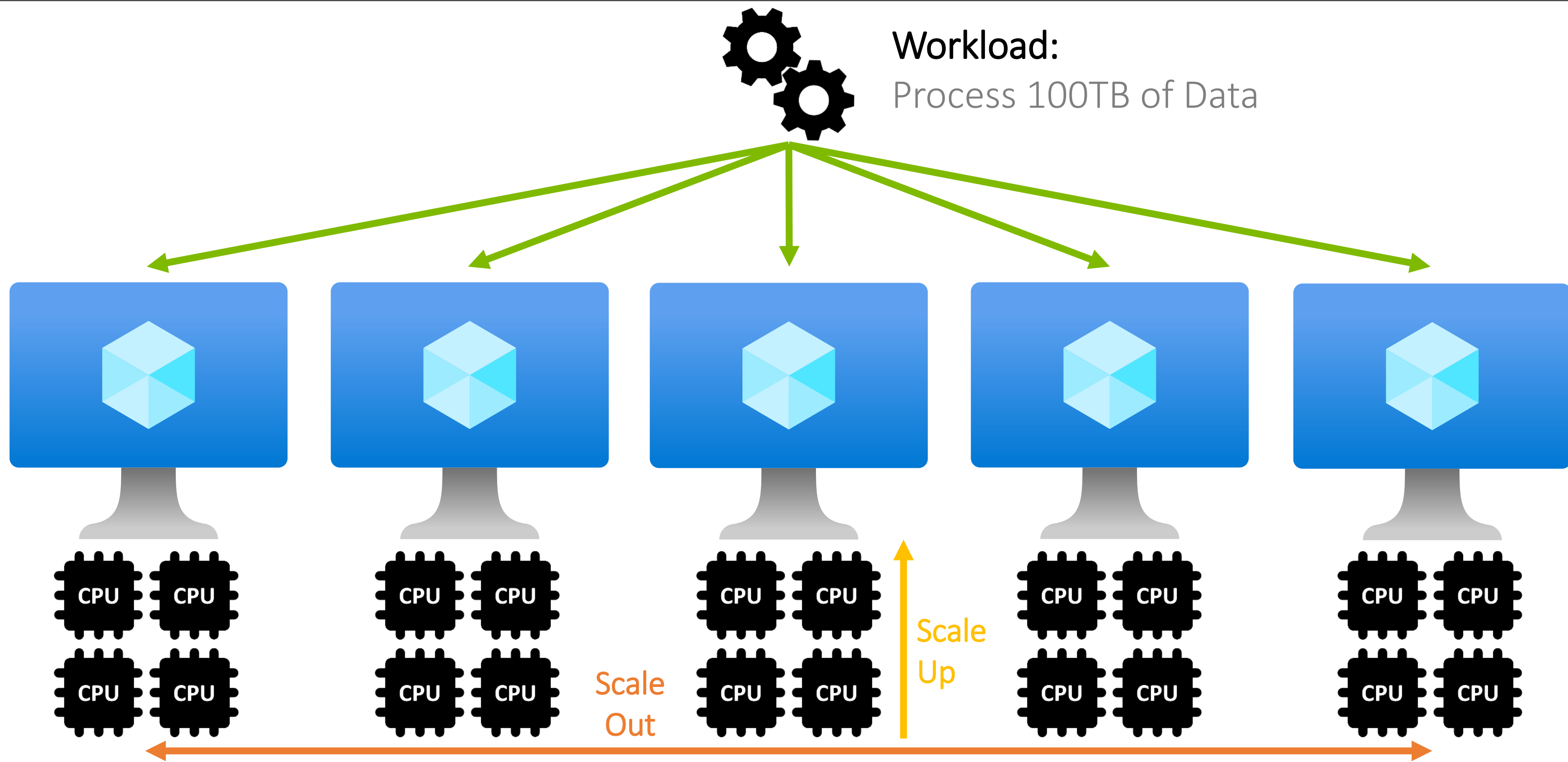
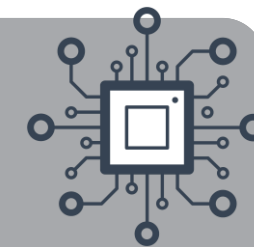


# Scaling Up vs Scaling Out



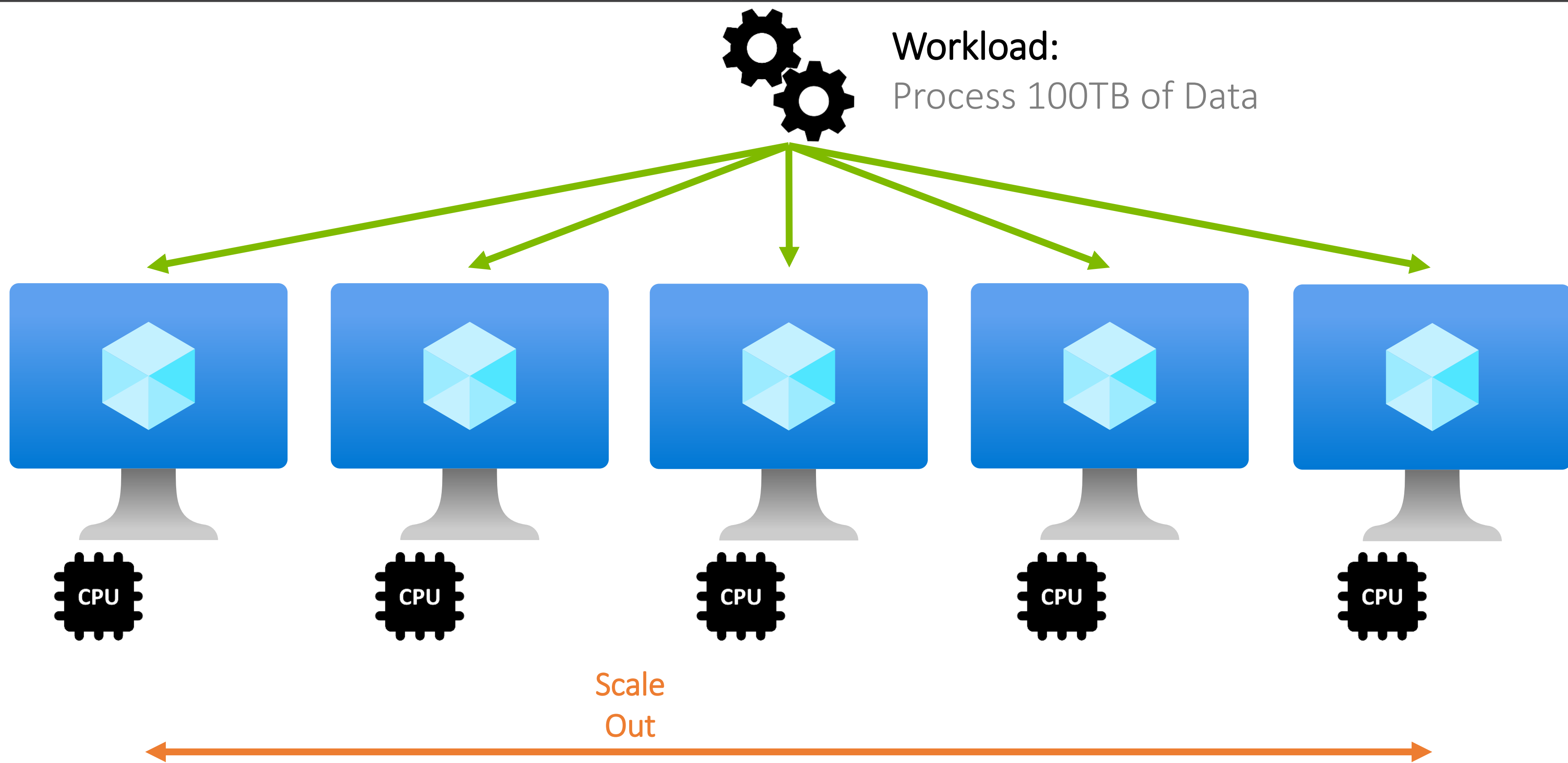
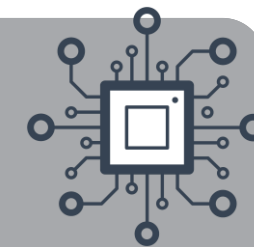


# Scaling Up and/or Scaling Out



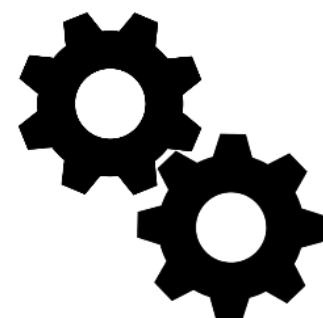
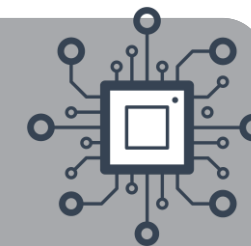


# Scaling Up and/or Scaling Out

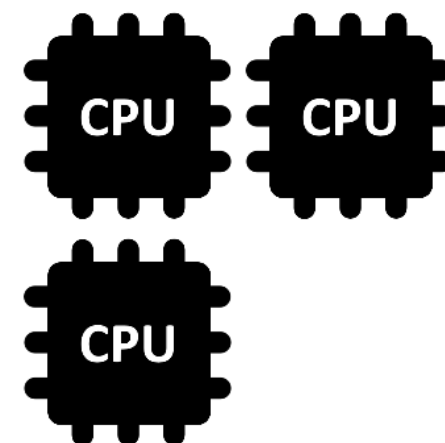
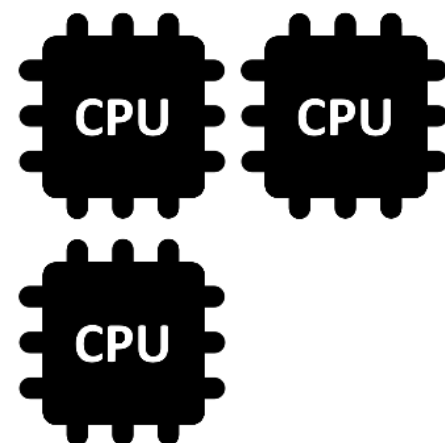
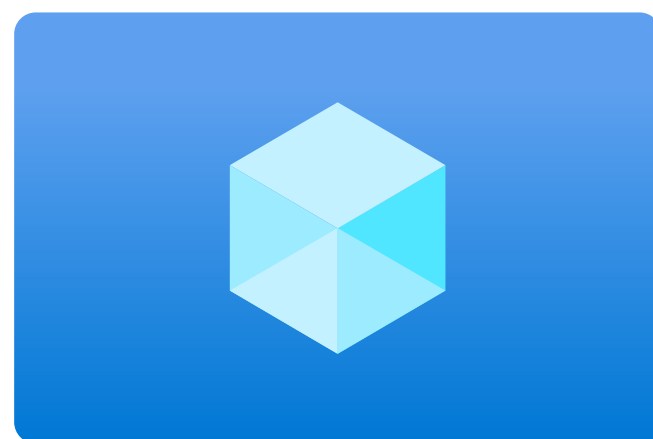
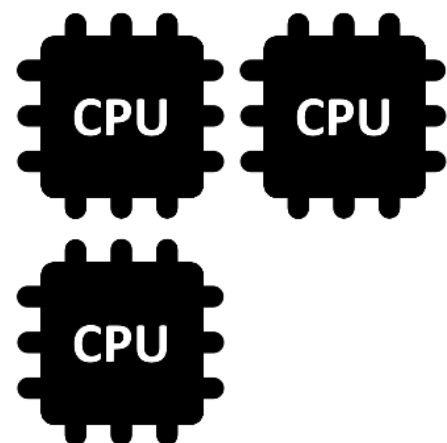
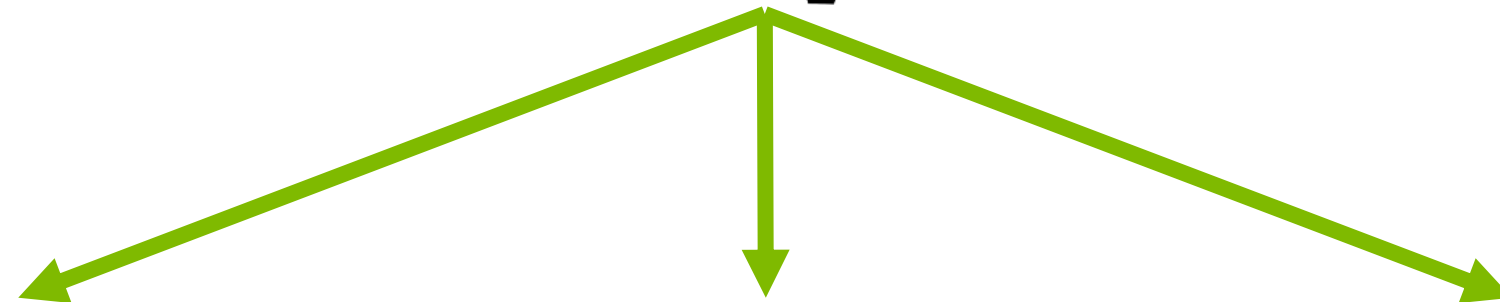




# Scaling Up and/or Scaling Out



Workload:  
Process 100TB of Data



Scale  
Out

Scale  
Up

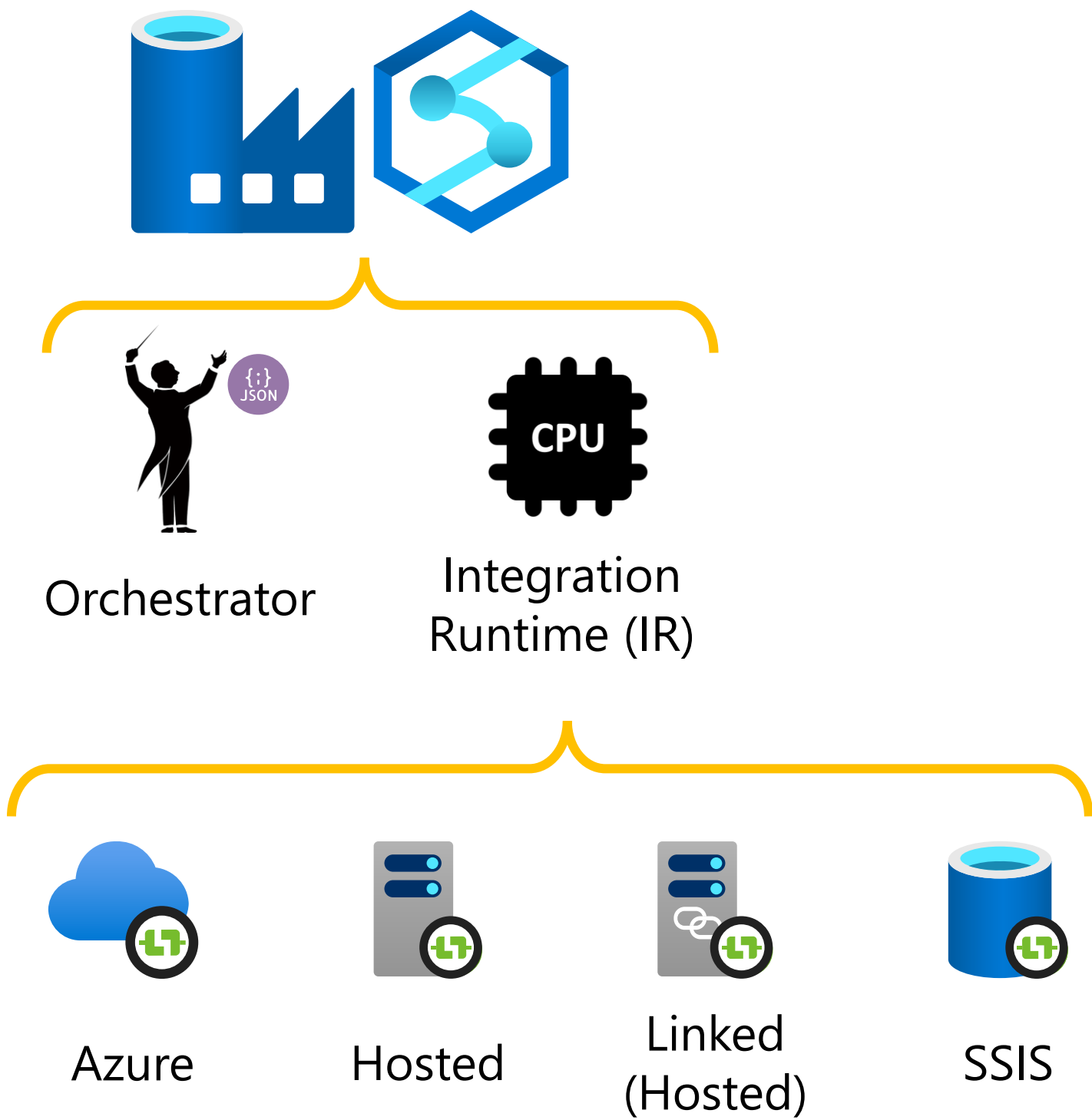


# Integration Components

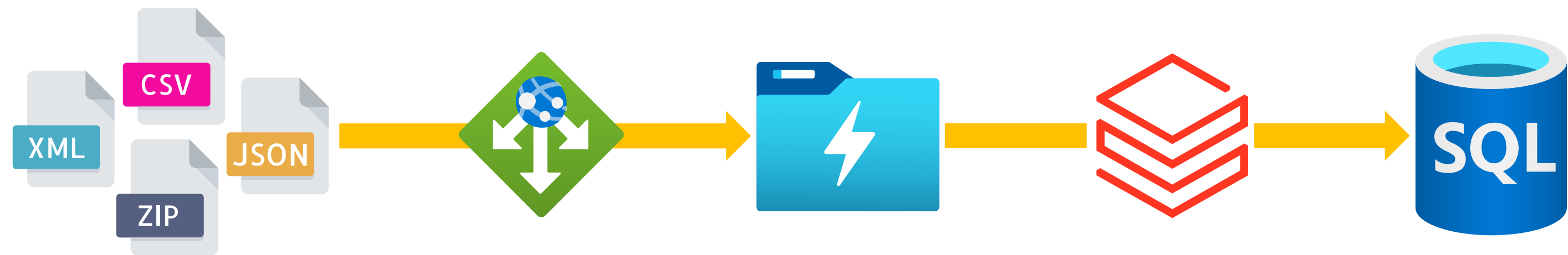




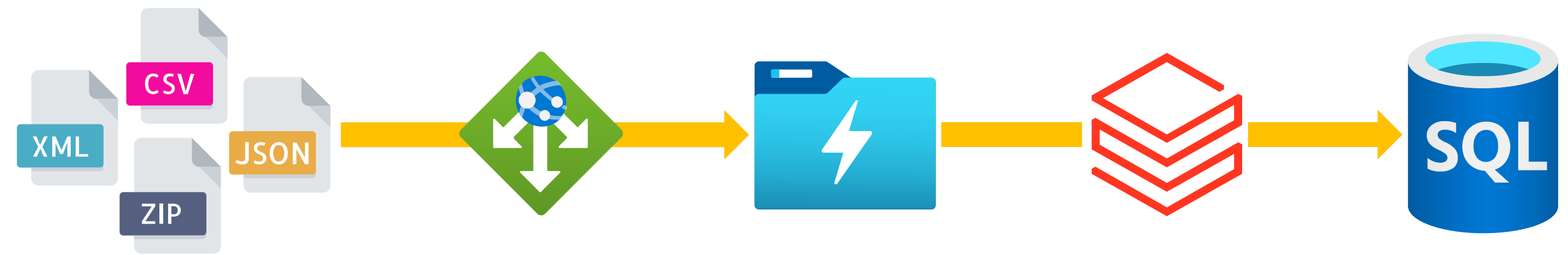
# Integration Runtimes



# Integration Components

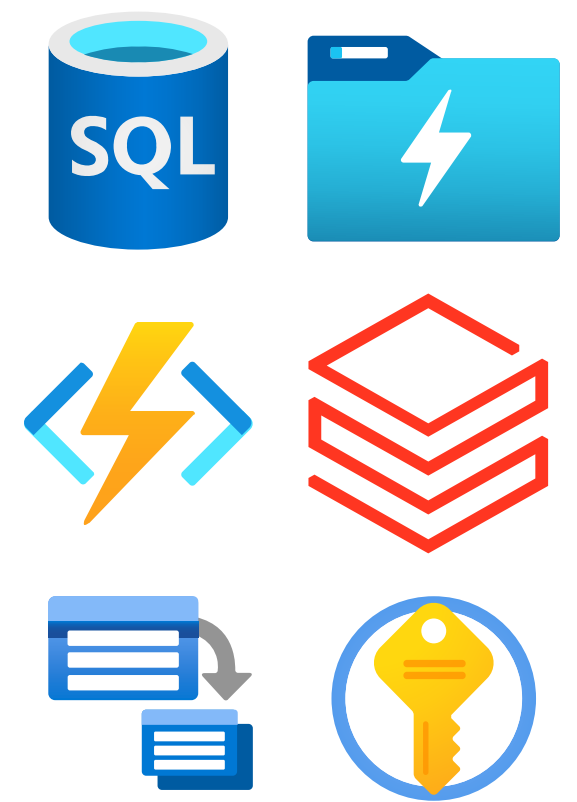


# Integration Components



1

## Linked Services – What to interact with and how?

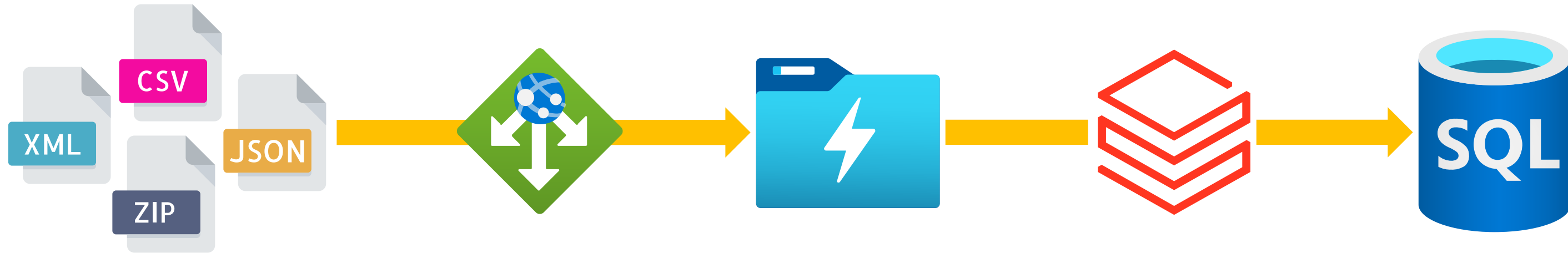




```
SQLDBLinkedService

ConnectionString: Server=MyServer;Database=myDataBase
UserName: "MrPaulAndrew"
Password: *****(10 asterisks)****
```

# Integration Components



1 Linked Services

2 Datasets – Where is my data? What format? What file path/table do I need?

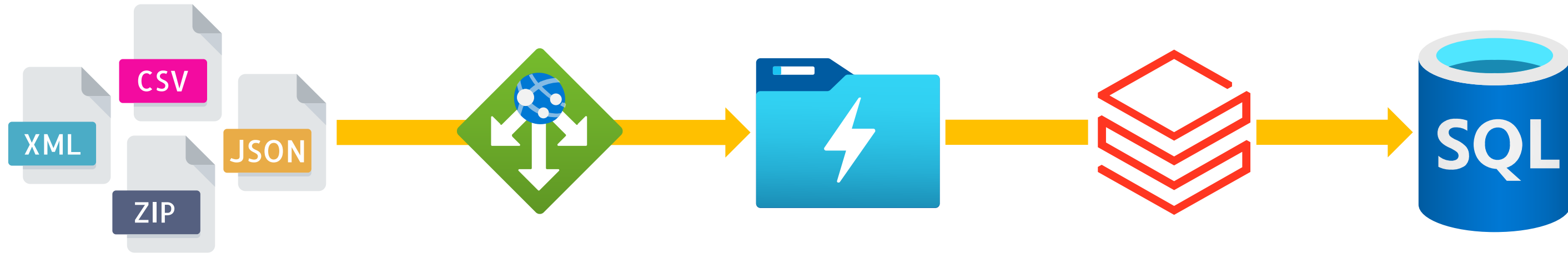


[dbo].[SalesOrders]



/RAW/Orders/2018/01/01/SalesOrders.csv

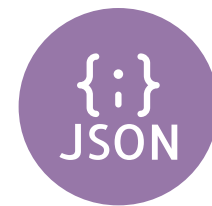
# Integration Components



1 Linked Services

2 Datasets

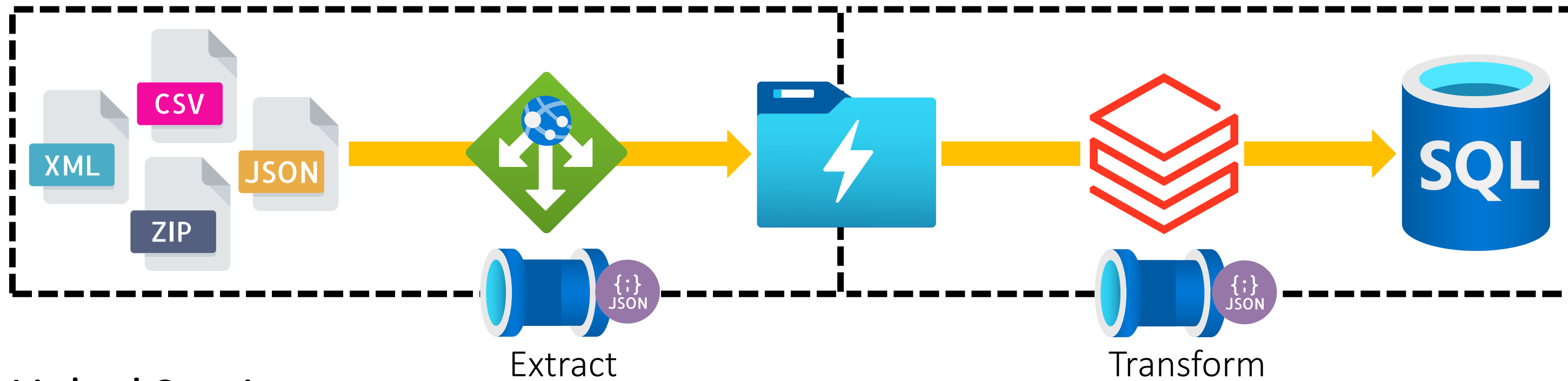
3 **Activities** – What do we want to happen when we invoke a Linked Service?  
With what conditions?



Databricks Notebook Activity

```
notebookPath: /Playground/Playing  
baseParameters: Testing  
libraries[jar]: dbfs:/lib1.jar  
linkedServiceName: BricksOfData01
```

# Integration Components

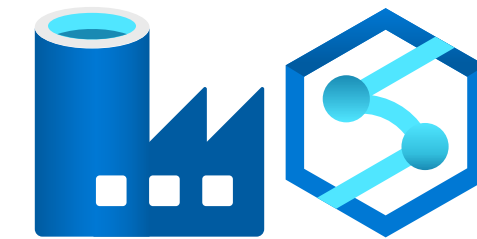
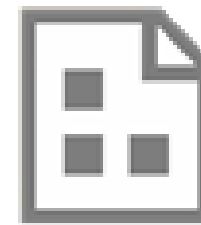


1 Linked Services

2 Datasets

3 Activities

4 **Pipelines** – Logical groups of work that can be executed.



Sequence Container

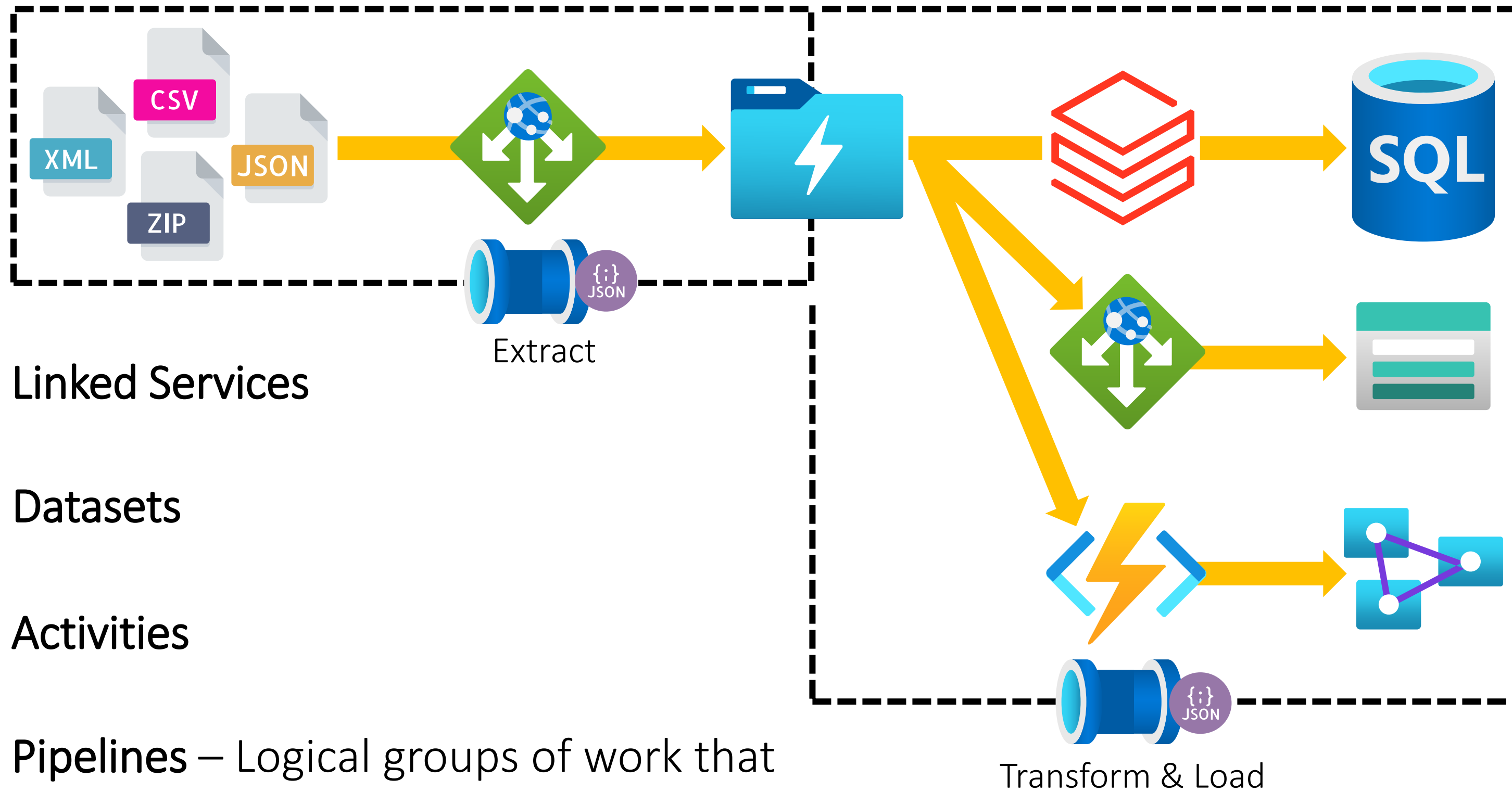


Execute Package Task



Execute Pipeline Activity

# Integration Components



1

Linked Services

2

Datasets

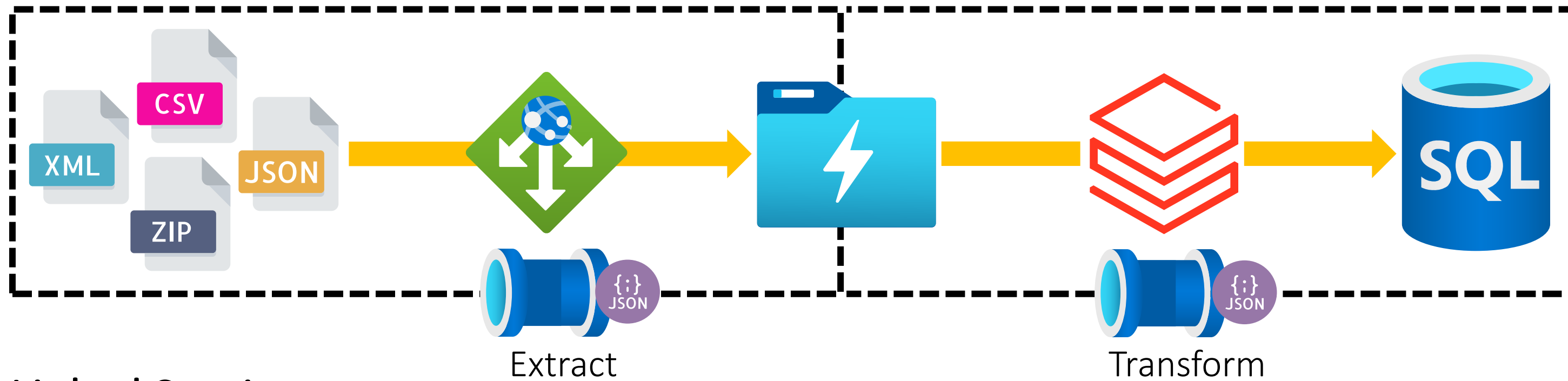
3

Activities

4

**Pipelines** – Logical groups of work that can be executed.

# Integration Components



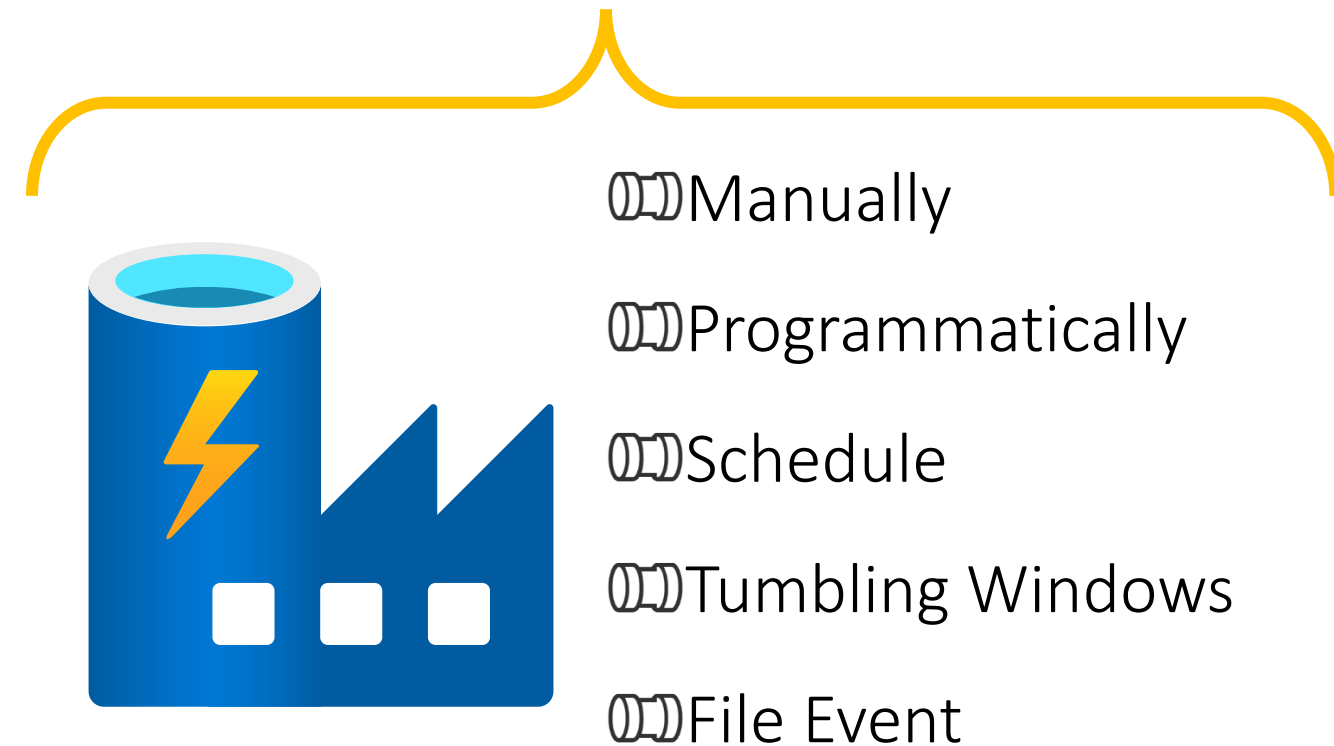
1 Linked Services

2 Datasets

3 Activities

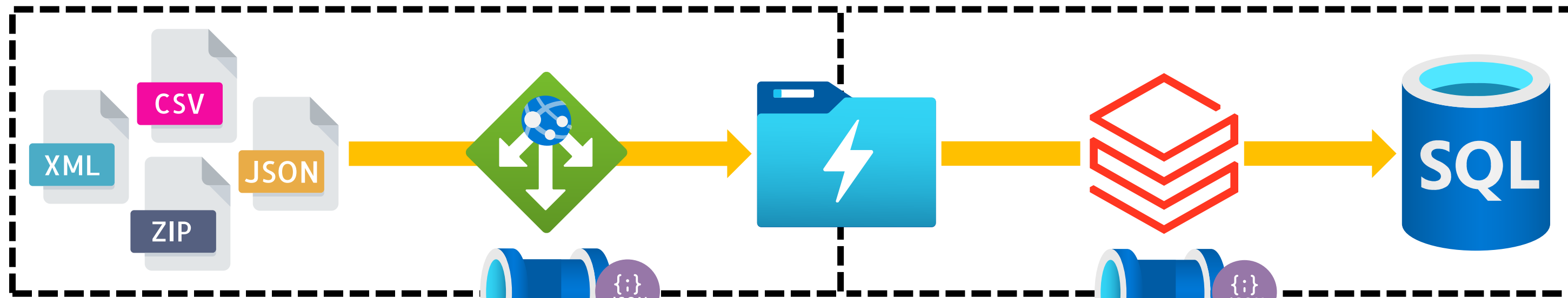
4 Pipelines

5 Triggers – Tell our pipelines when to run.





# Integration Components



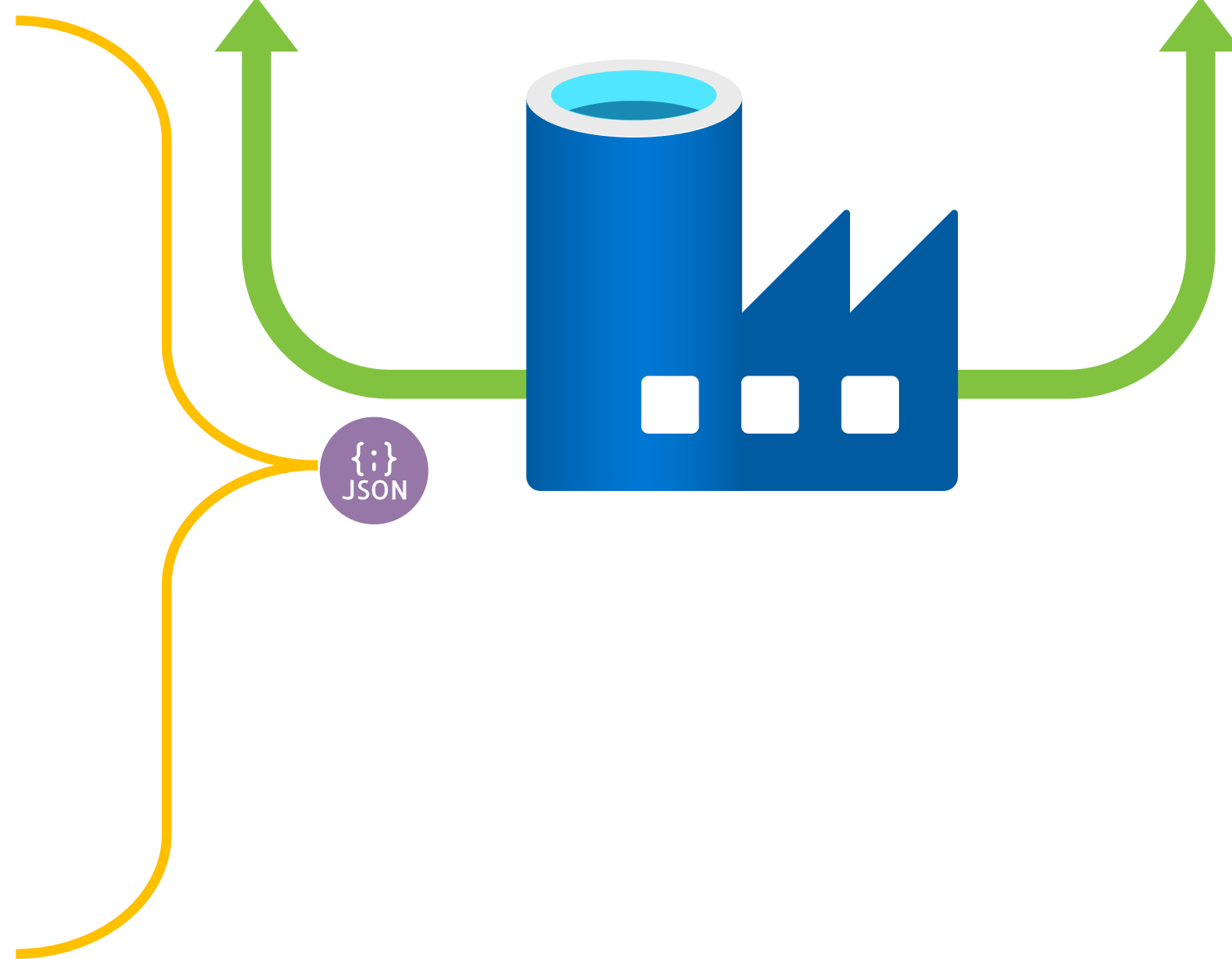
1 Linked Services

2 Datasets

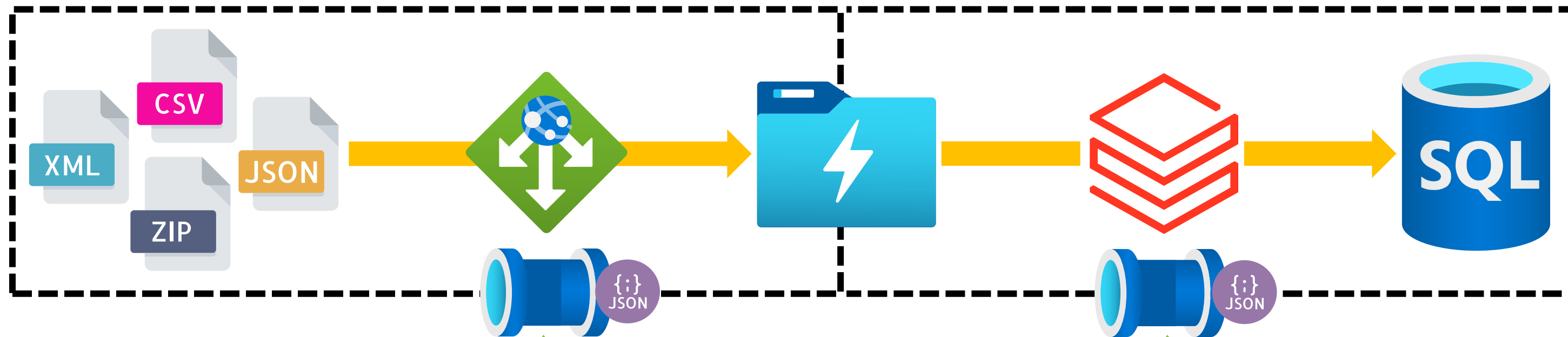
3 Activities

4 Pipelines

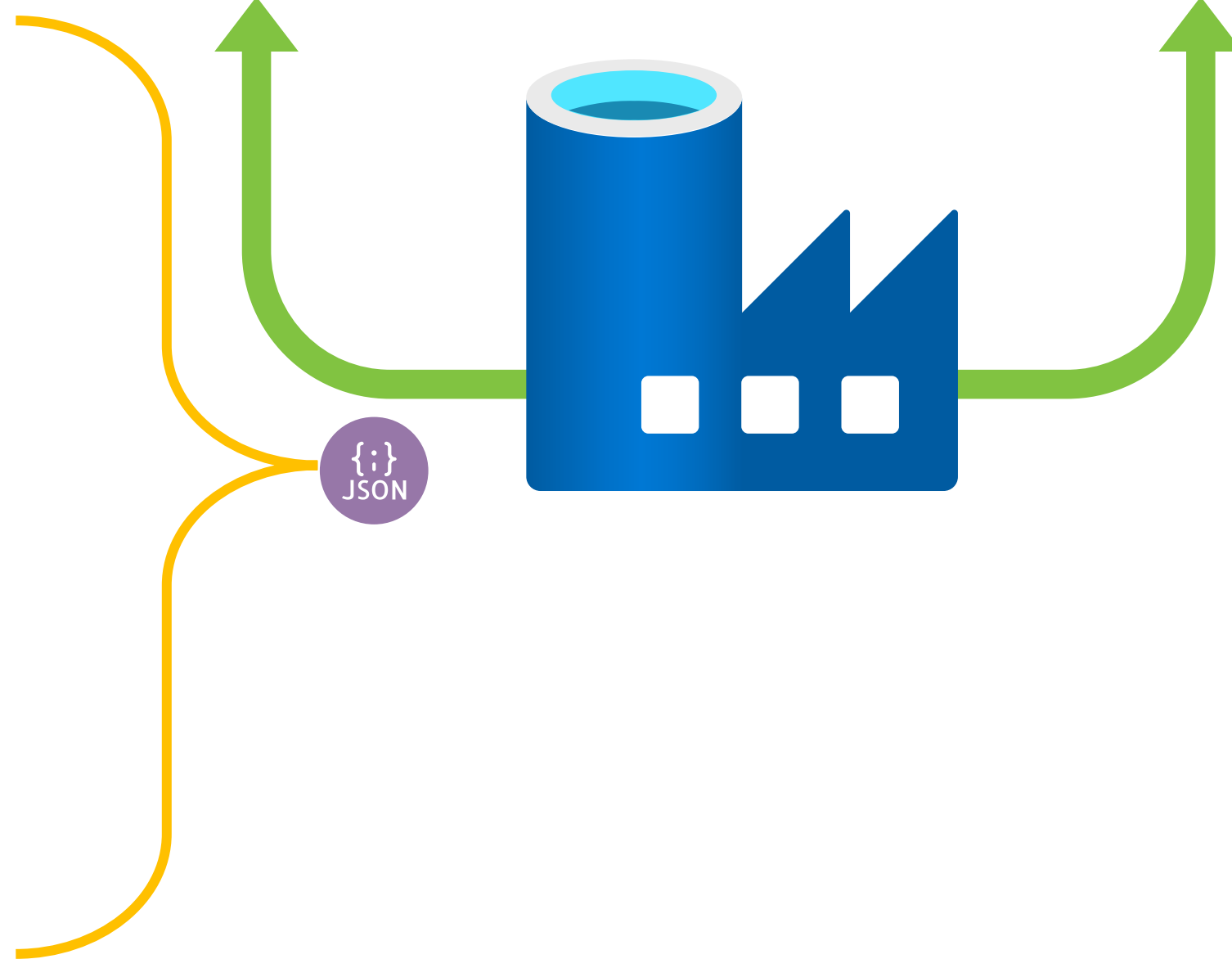
5 Triggers

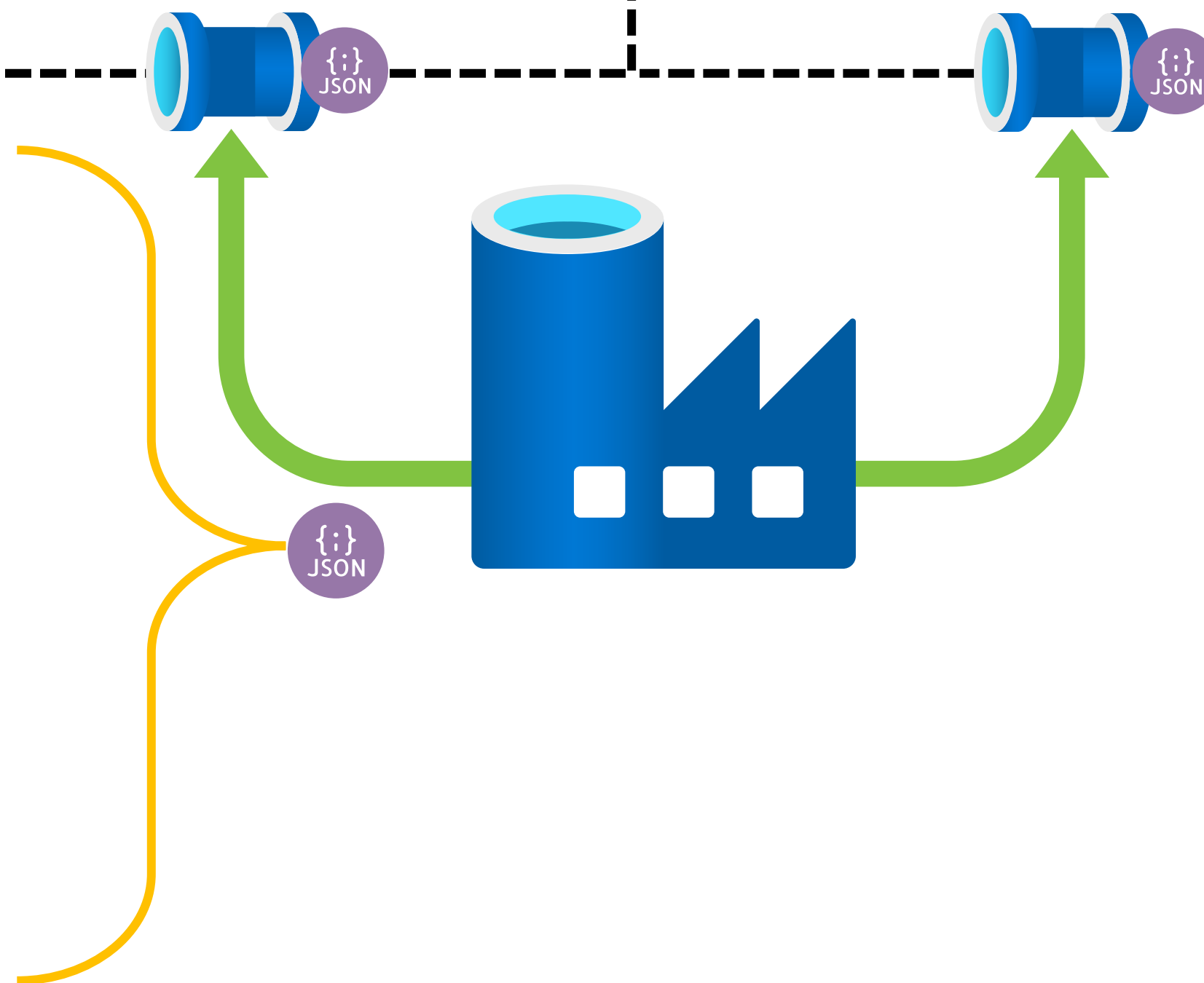
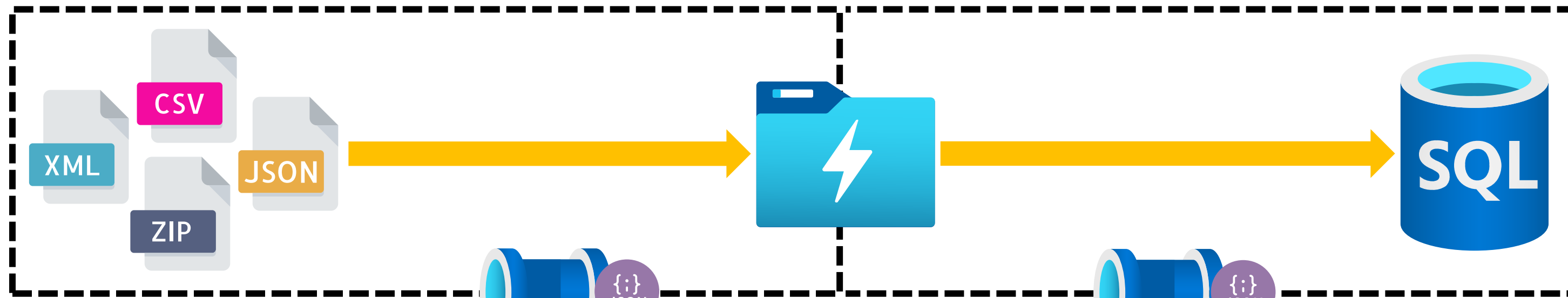


# Integration Control Flow Components



- 1 Linked Services
- 2 Datasets
- 3 Activities
- 4 Pipelines
- 5 Triggers





1 Linked Services

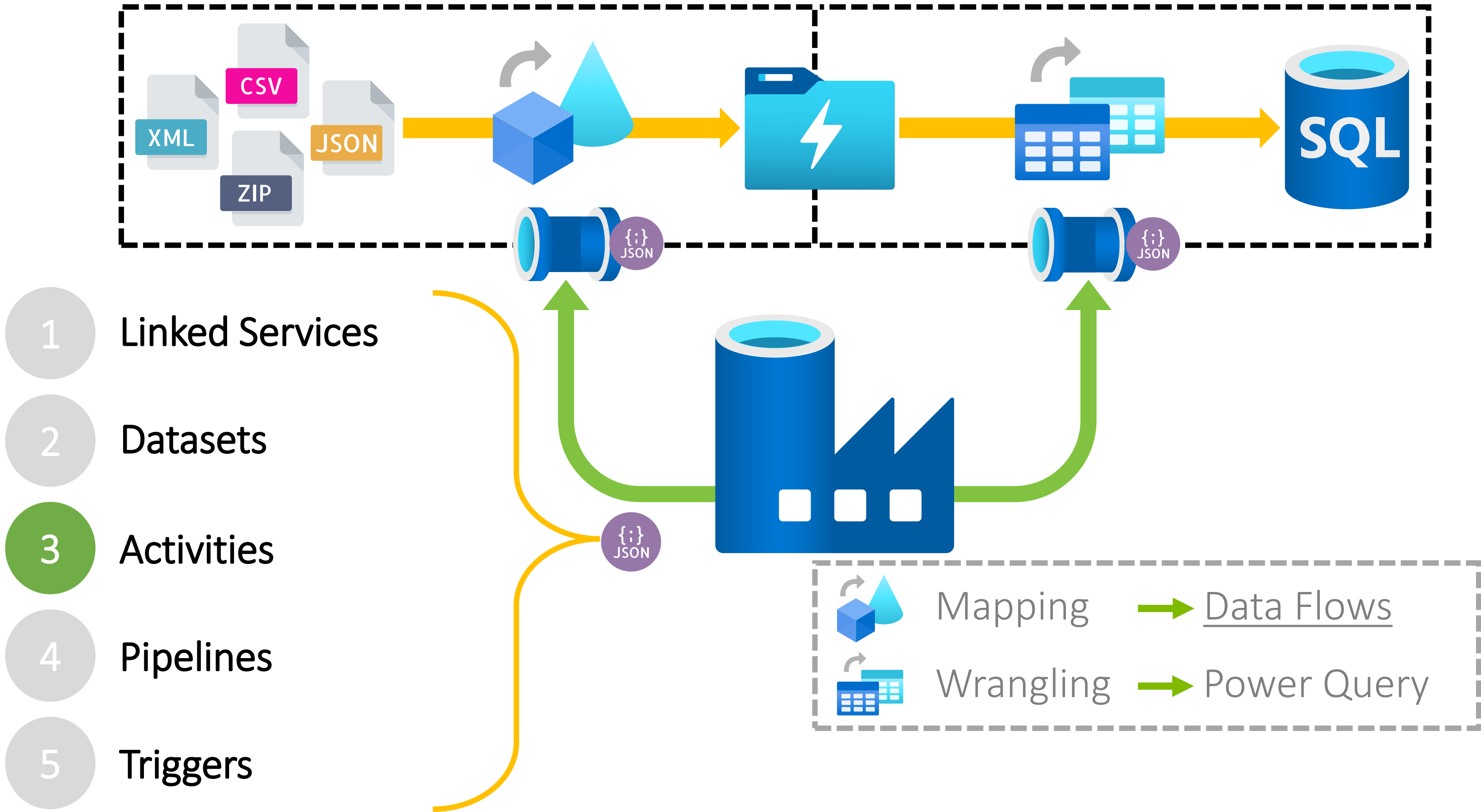
2 Datasets

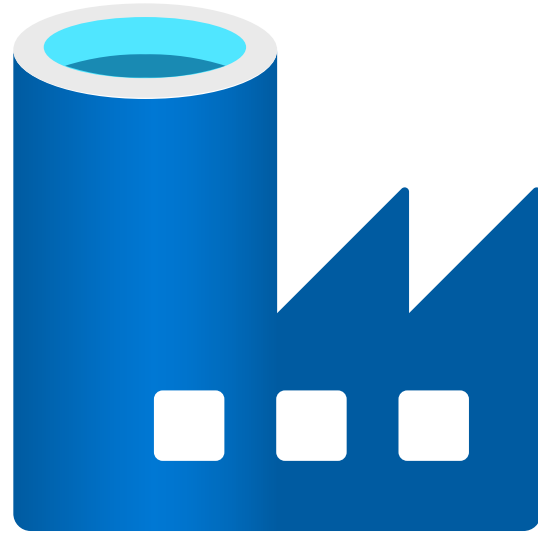
3 Activities

4 Pipelines

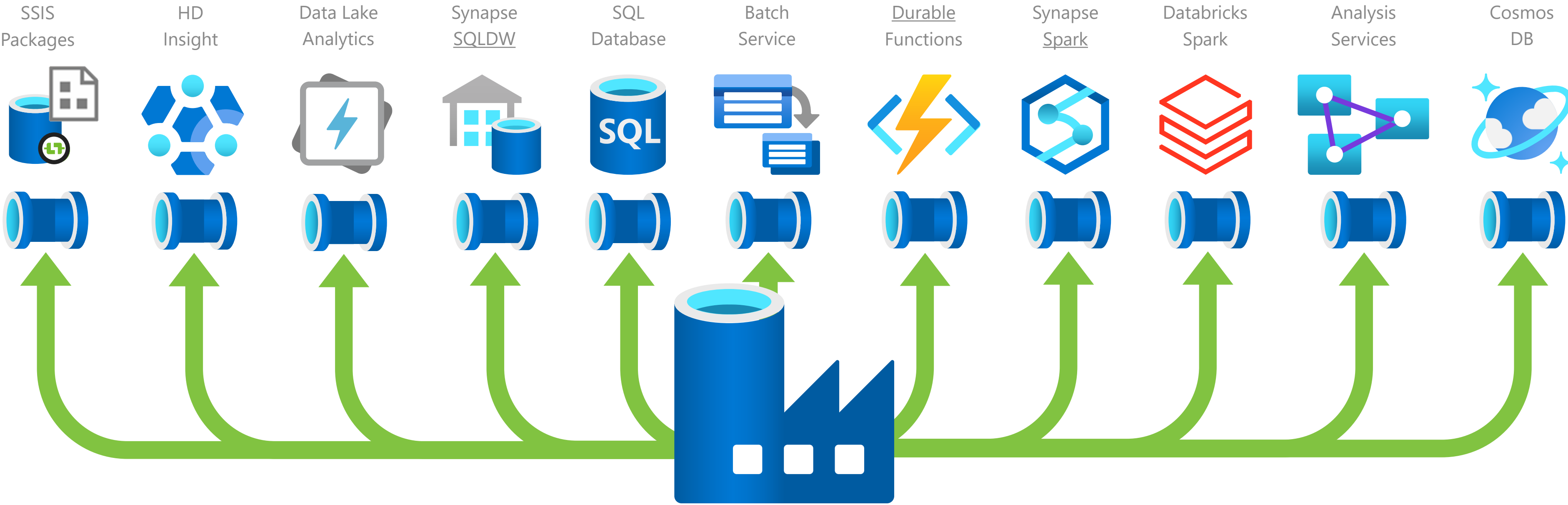
5 Triggers

# Integration Data Flow (Transformation) Activities

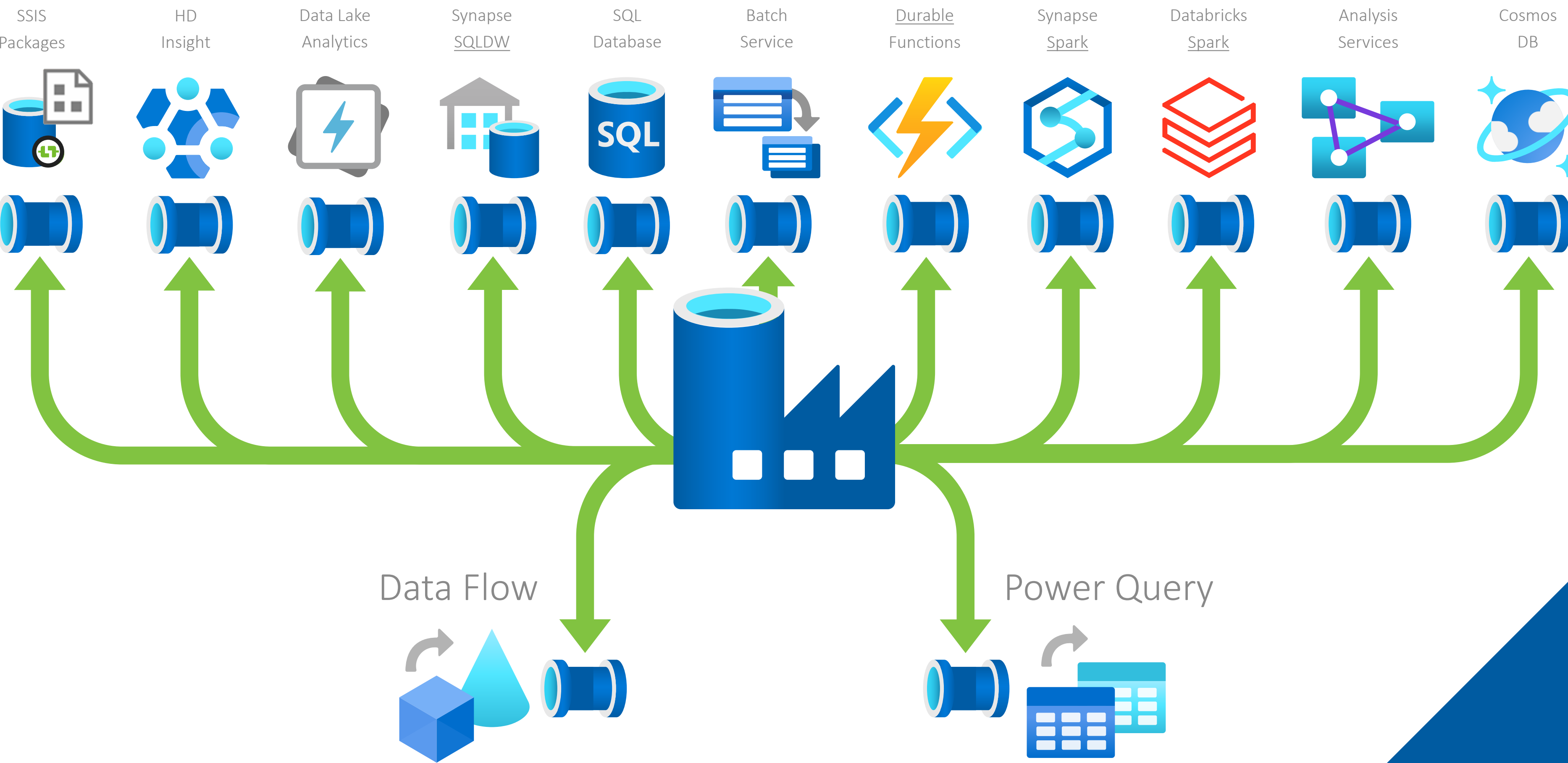




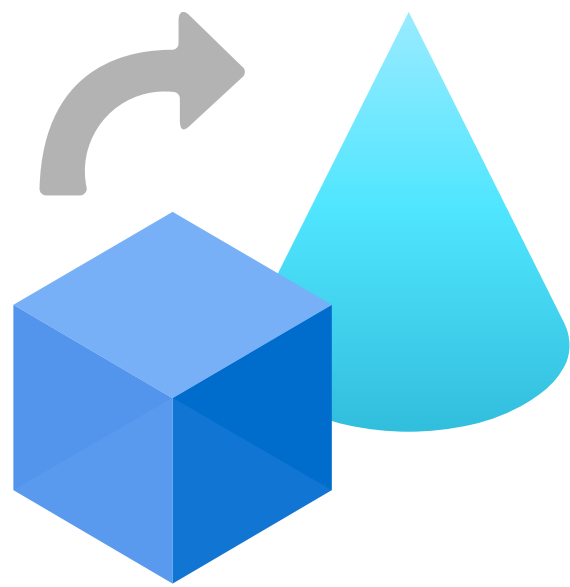
# Other Data Transformation Services in Azure



# When Should We Use These Integration Pipeline Transformation Activities?

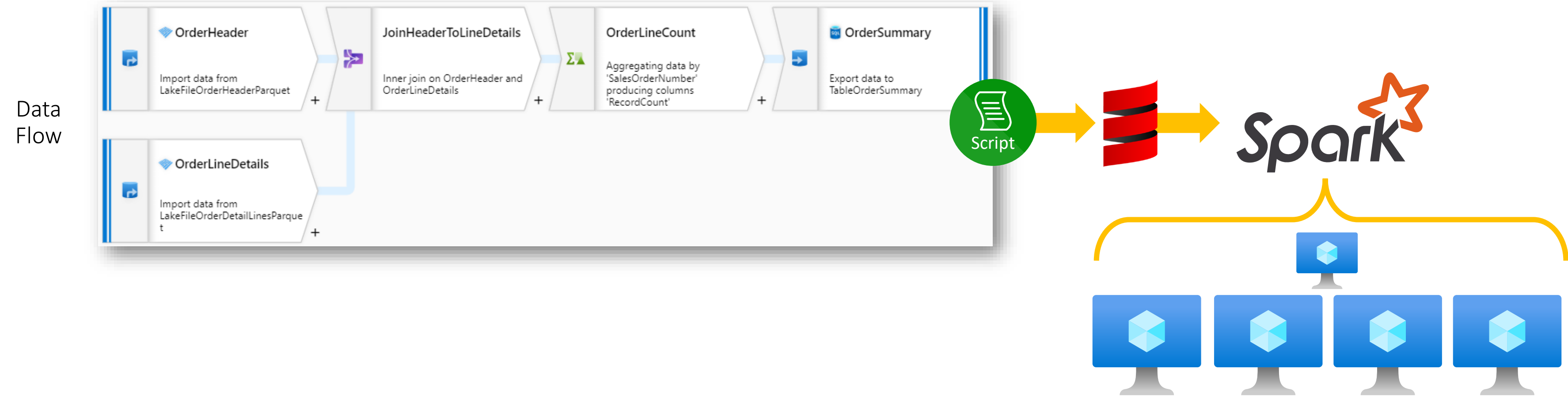


# Data Flows

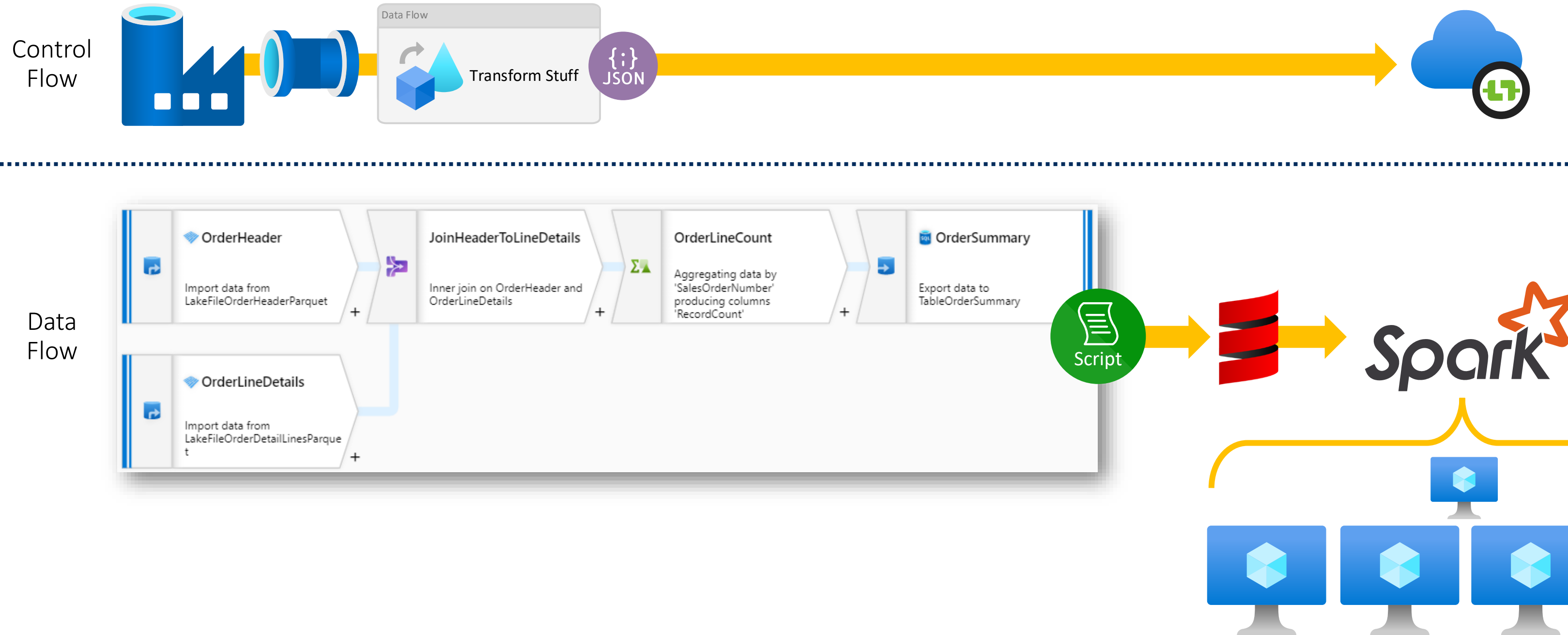




# What is a Mapping Data Flow?



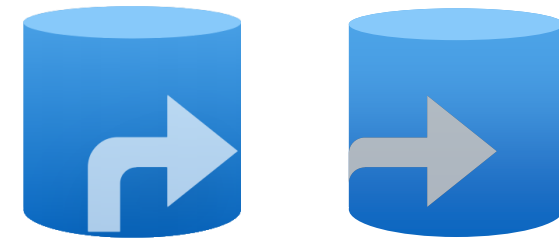
# Q: What is a Mapping Data Flow?



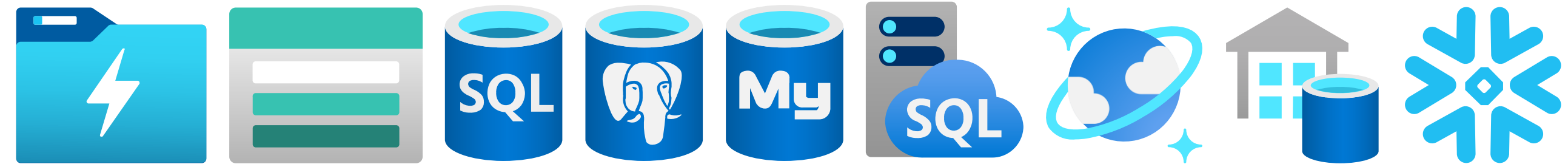
**A:** Graphic no low/low code data transformation tool that sits on top of Apache Spark.

# Data Flows – Inputs & Outputs

Source & Sink



Linked Services



Source  
Types

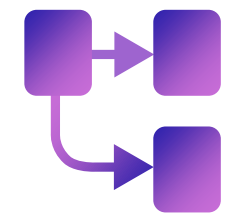
Dataset 



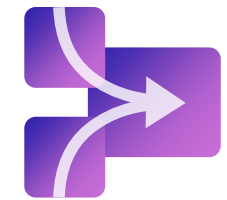
Inline 



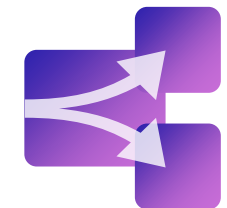
# Data Flows – Transformations



New Branch



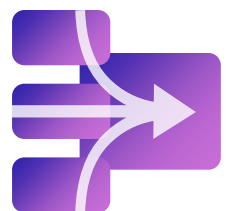
Join



Conditional Split



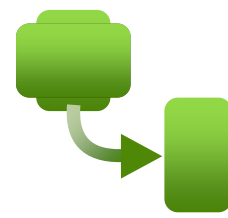
Exists



Union



Lookup



Derived Column



Select



Aggregate



Surrogate Key



Pivot



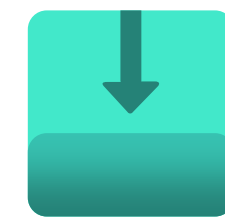
Unpivot



Window



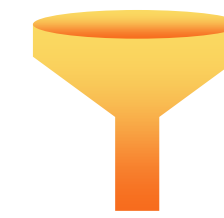
Rank



Flatten



Parse



Filter



Sort



Alter Row

Key

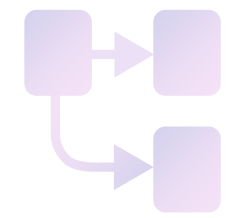
Input & Output Modifiers

Schema Modifiers

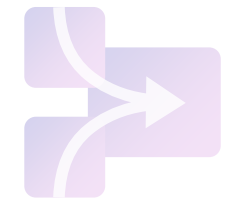
Formatters

Row Modifiers

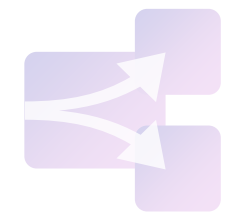
# Data Flows – Transformations



New Branch



Join



Conditional Split



Exists



















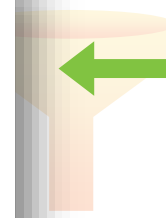
Union



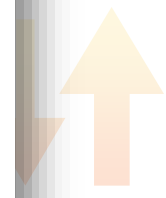
Lookup

## Components

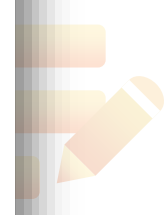
Operation / Activity	Description	SSIS equivalent	SQL Server equivalent
 New branch	Create a new flow branch with the same data	 Multicast (+icon)	<pre>1 SELECT INTO 2 SELECT OUTPUT</pre>
 Join	Join data from two streams based on a condition	 Merge join	<pre>1 INNER/LEFT/RIGHT JOIN, 2 CROSS/FULL OUTER JOIN</pre>
 Conditional Split	Route data into different streams based on conditions	 Conditional Split	<pre>SELECT INTO WHERE condition1 SELECT INTO WHERE condition2 CASE ... WHEN</pre>
 Union	Collect data from multiple streams	 Union All	<pre>SELECT col1a UNION (ALL) SELECT col1b</pre>
 Lookup	Lookup additional data from another stream	 Lookup	<i>Subselect, function,</i> <pre>LEFT/RIGHT JOIN</pre>
 Derived Column	Compute new columns based on the existing ones	 Derived Column	<pre>SELECT Column1 * 1.09 as NewColumn</pre>
 Aggregate	Calculate aggregation on the stream	 Aggregate	<pre>SELECT Year(DateOfBirth) as YearOnly, MIN(), MAX(), AVG() GROUP BY Year(DateOfBirth)</pre>
 Surrogate Key	Add a surrogate key column to output stream from a specific value	 Script Component	<pre>SELECT ROW_NUMBER() OVER(ORDER BY name ASC) AS Row#, name FROM sys.databases</pre>



Filter



Sort



Alter Row

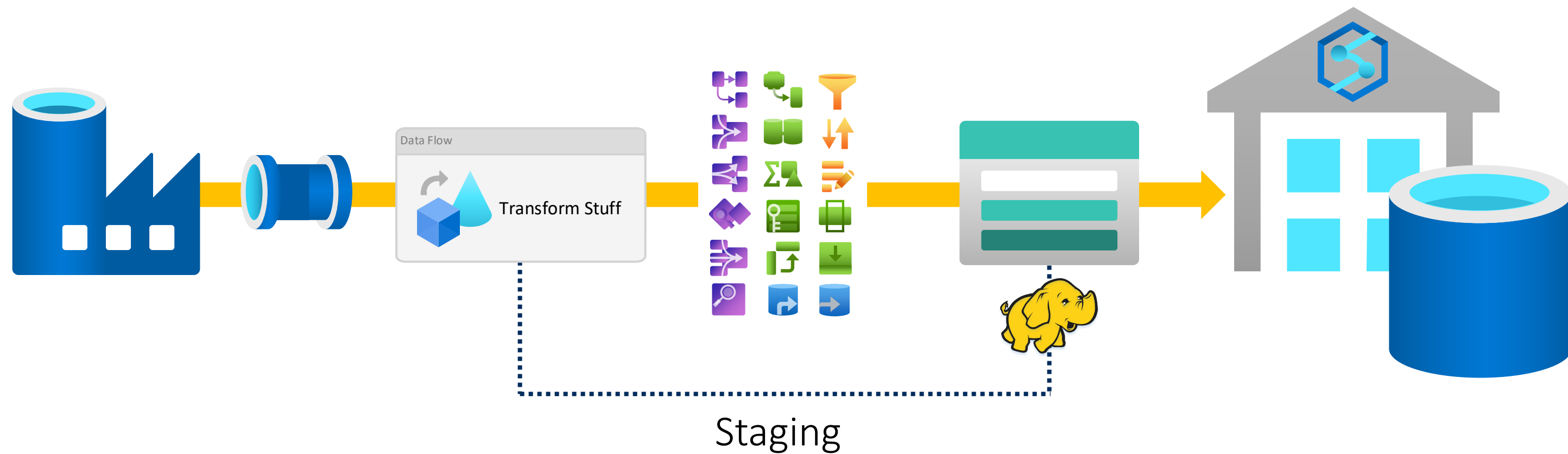
& Output Modifiers

na Modifiers

atters

Modifiers

# Data Flows – Data Warehouse Loading (PolyBase)

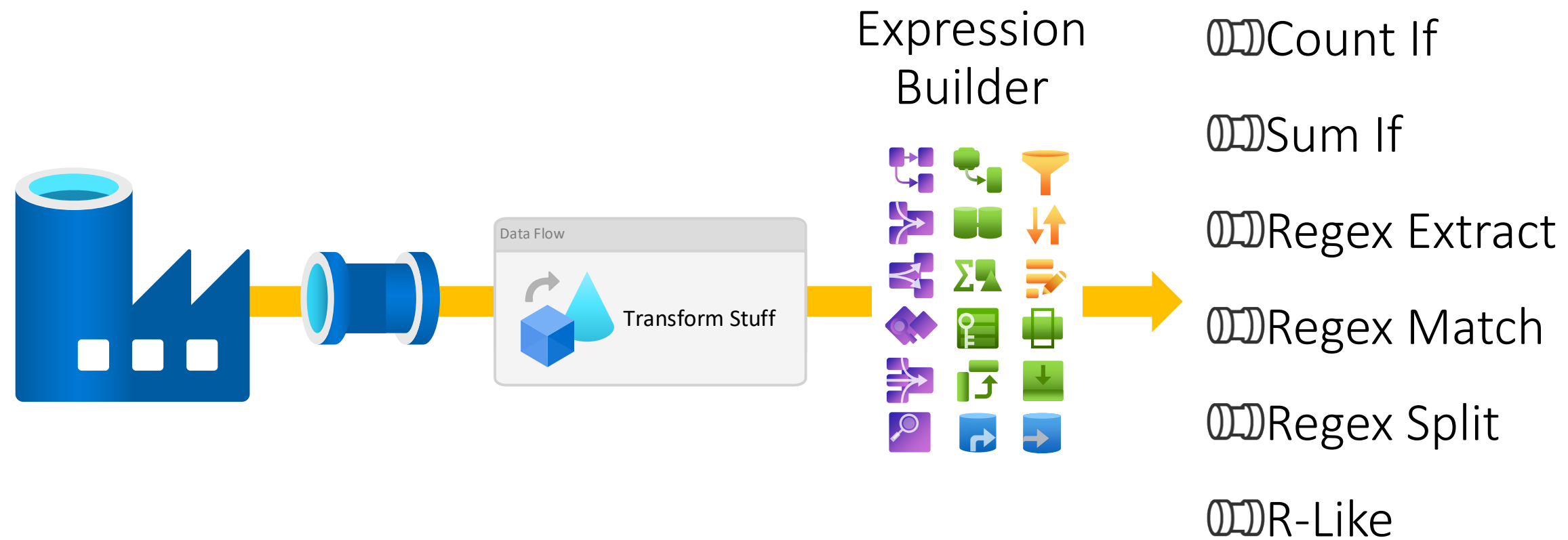


▲ PolyBase ⓘ

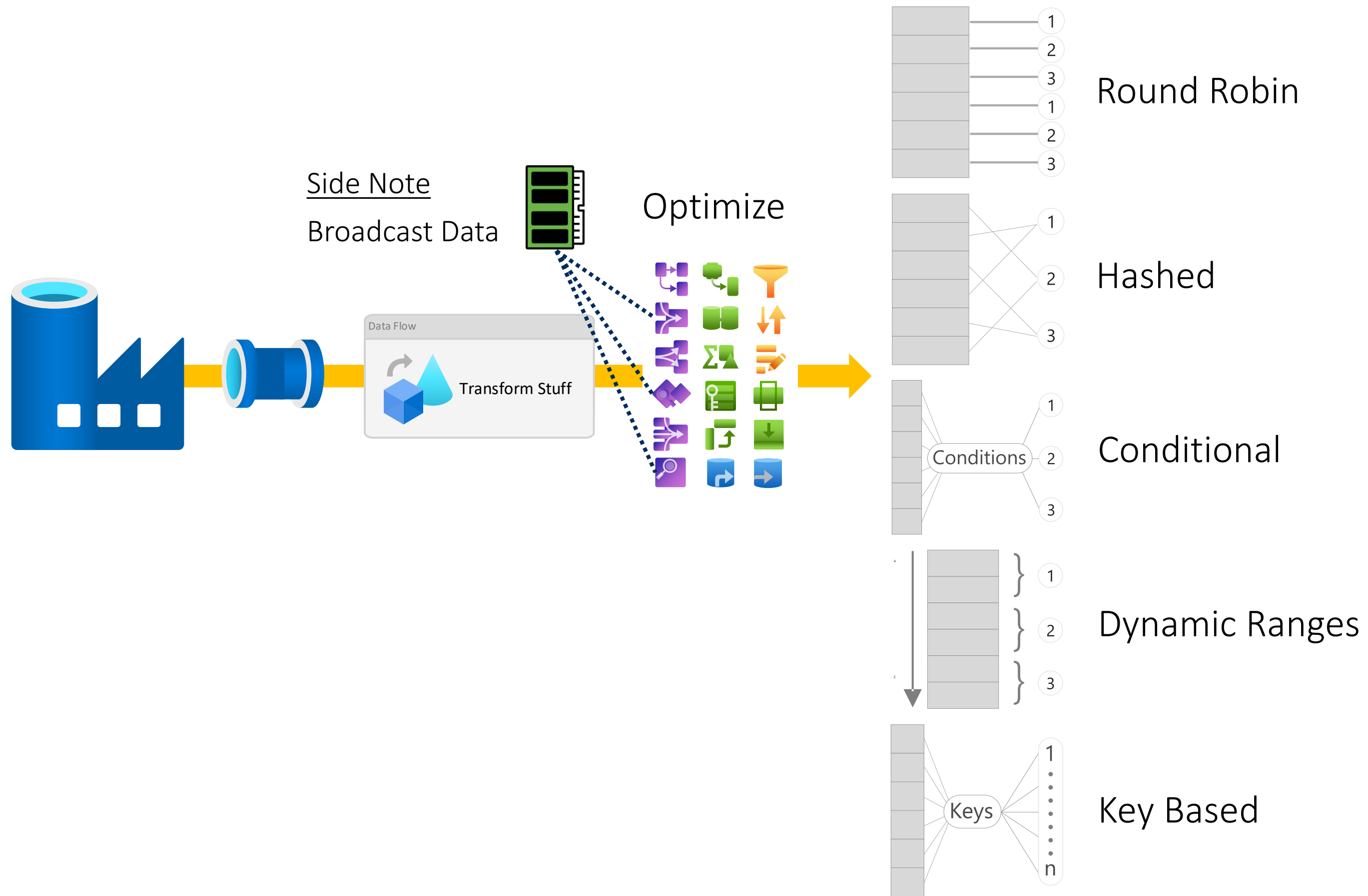
Staging linked service  ⓘ + New

Staging storage folder  /   | ▼

# Data Flows – Expression Builder

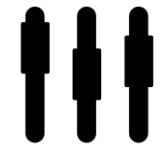


# Data Flows – Partition Handling



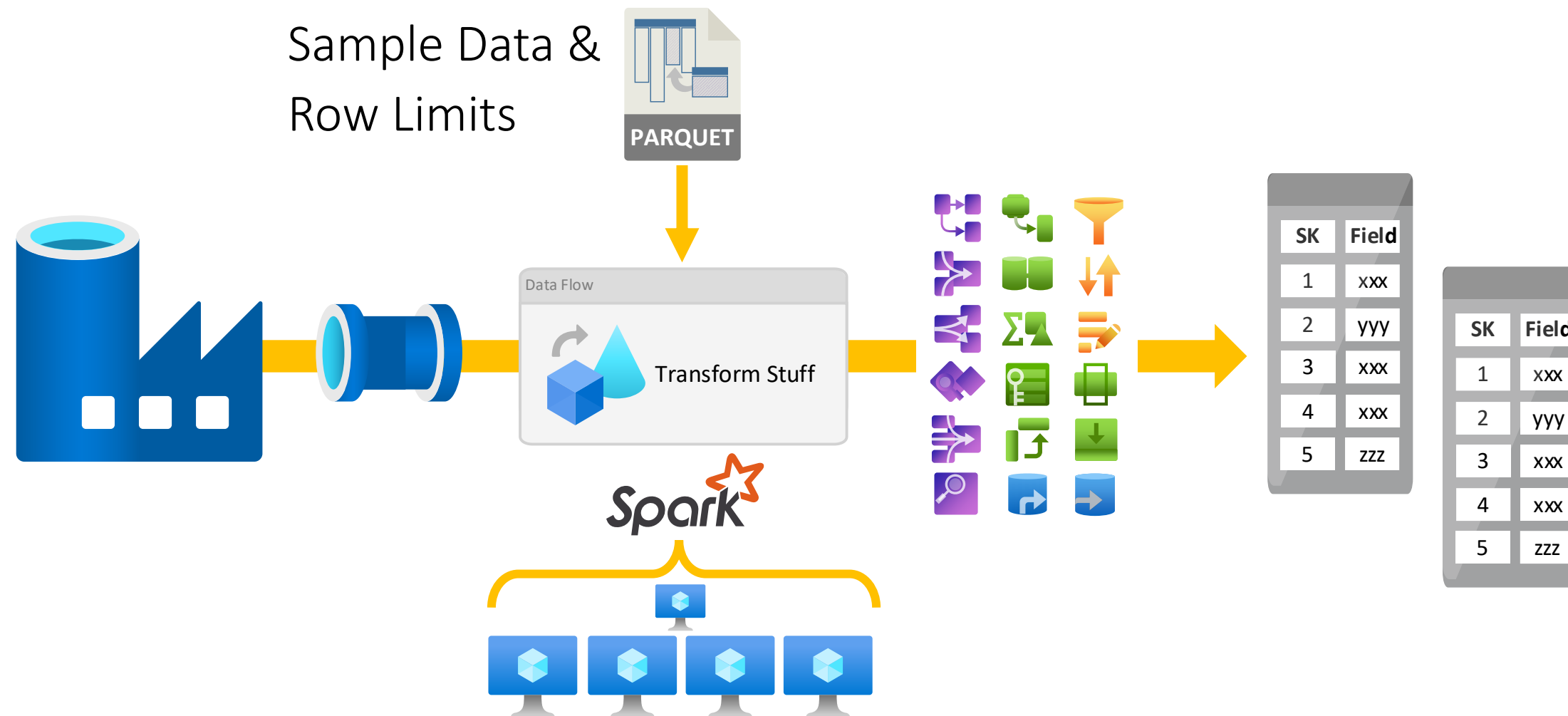


# Data Flows – Debugging

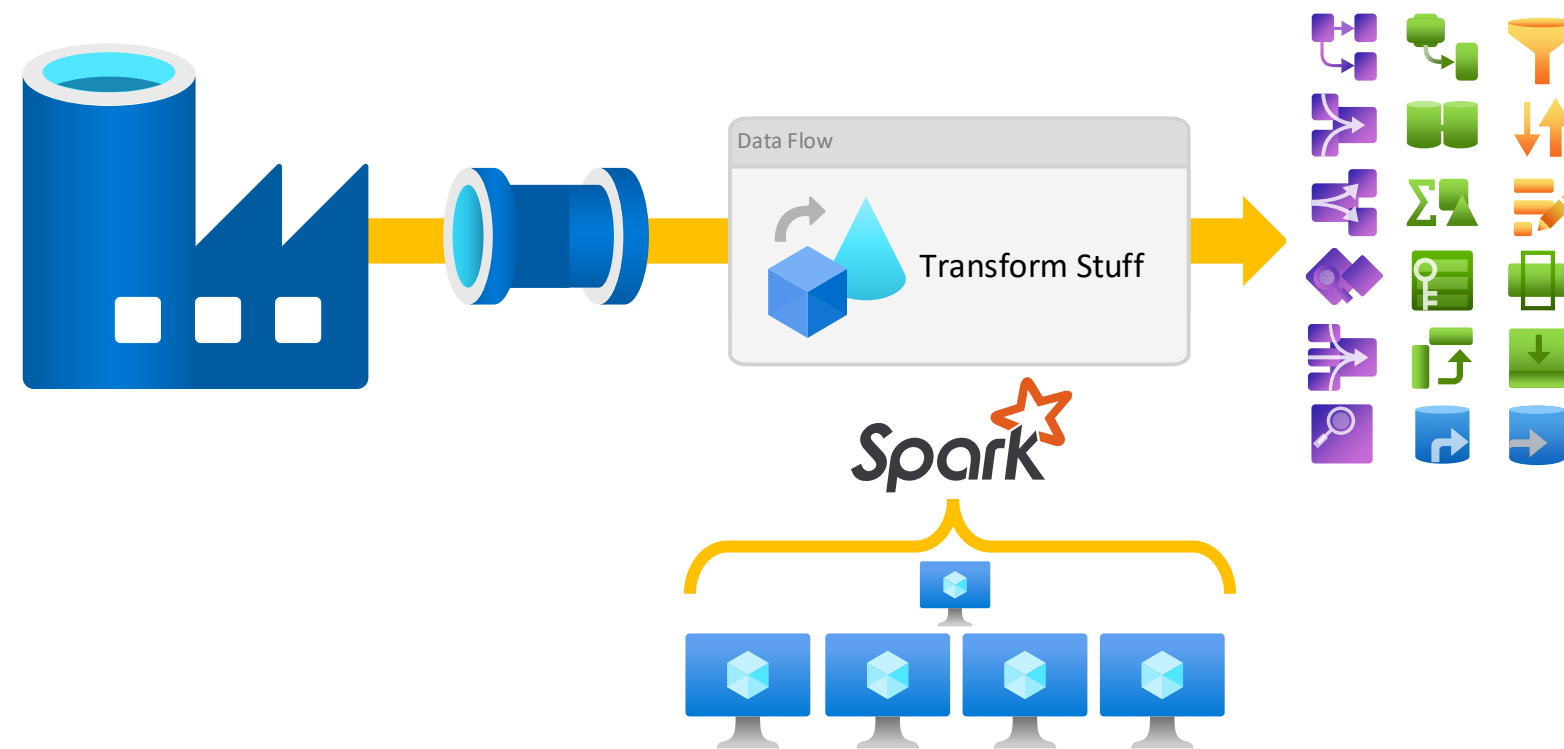


Enable Data Flow Debug Mode

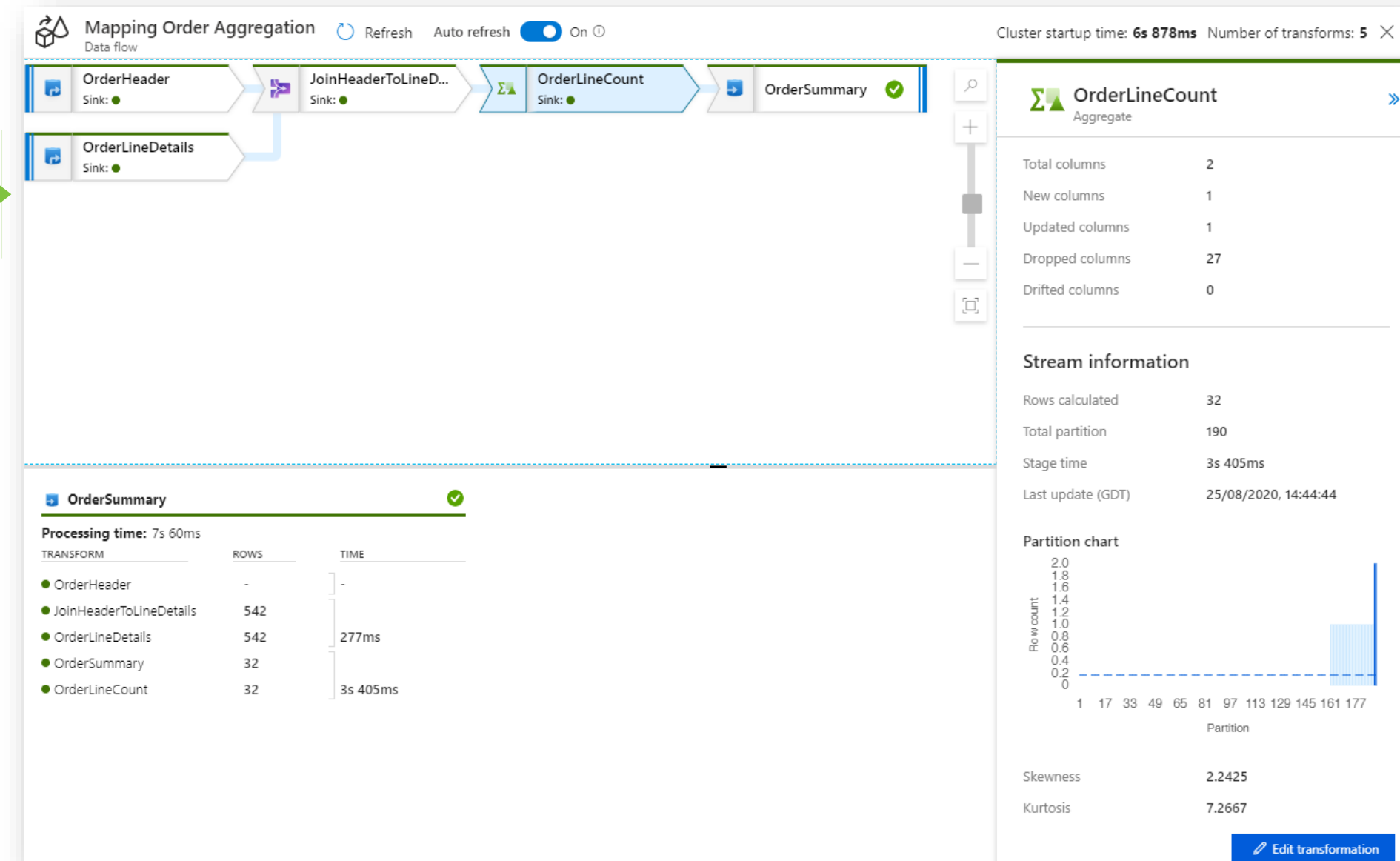
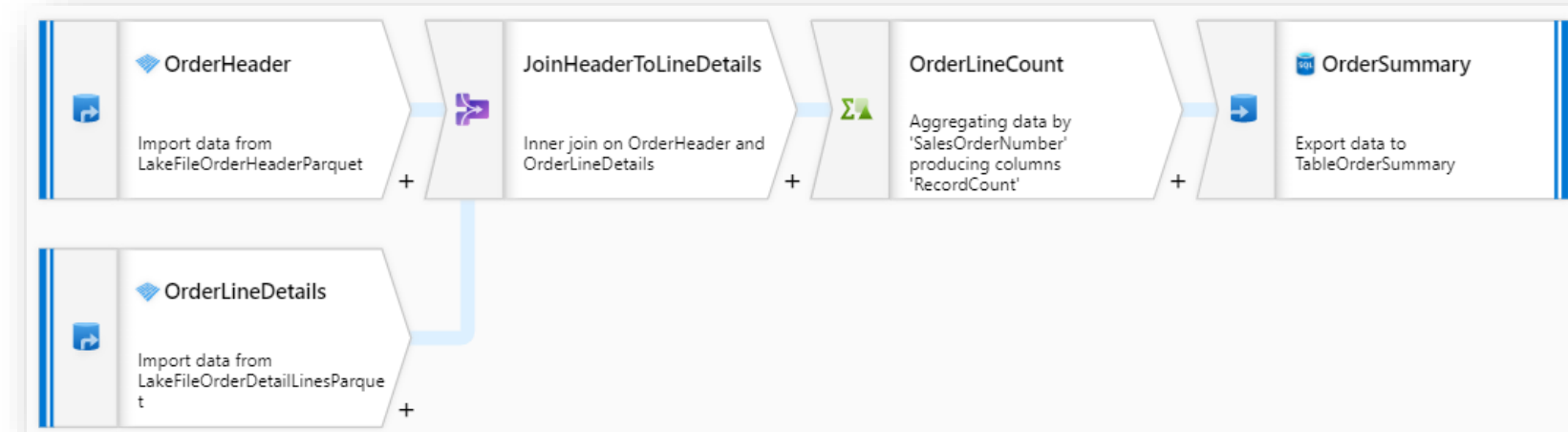
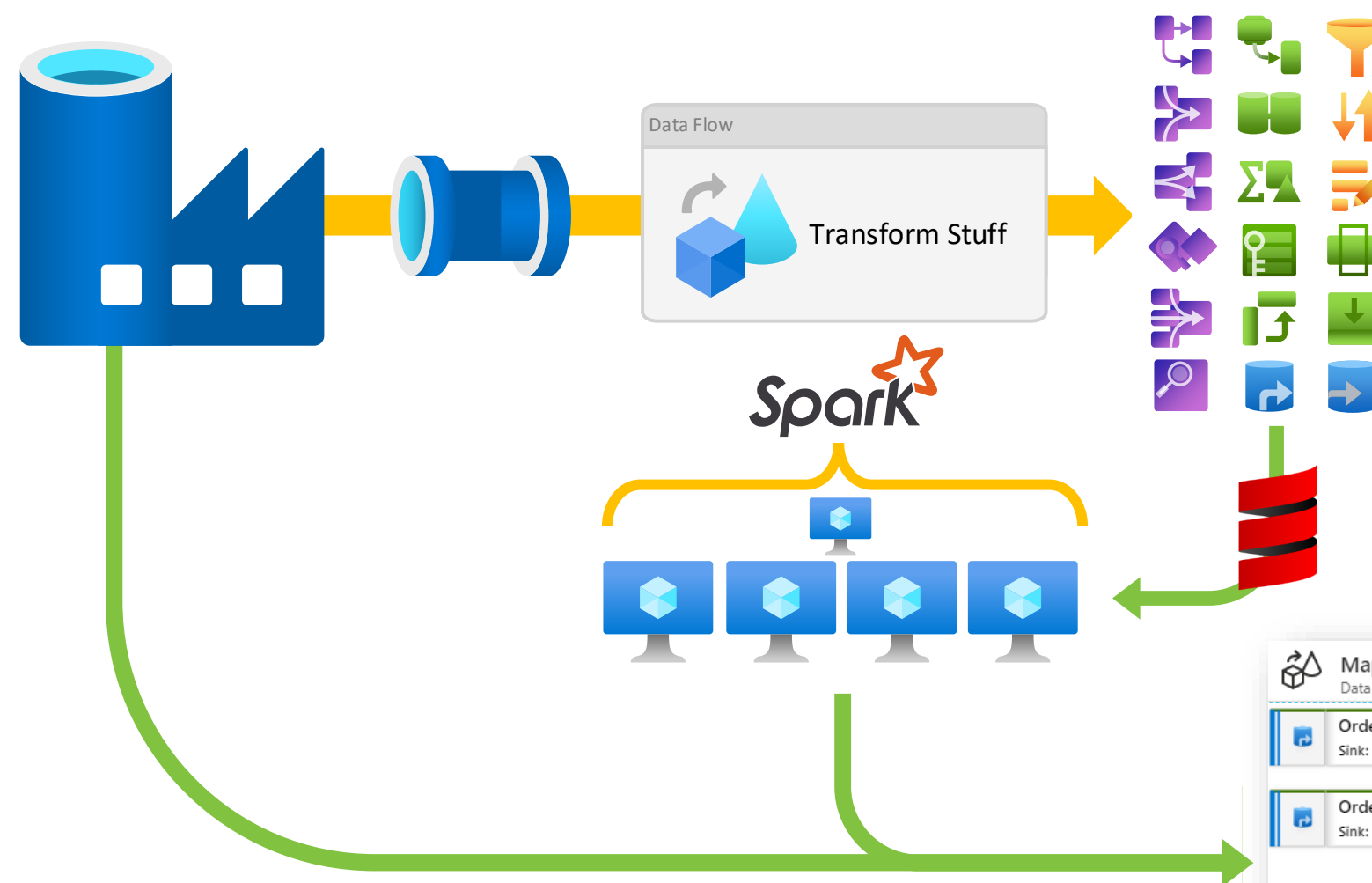
Data Preview

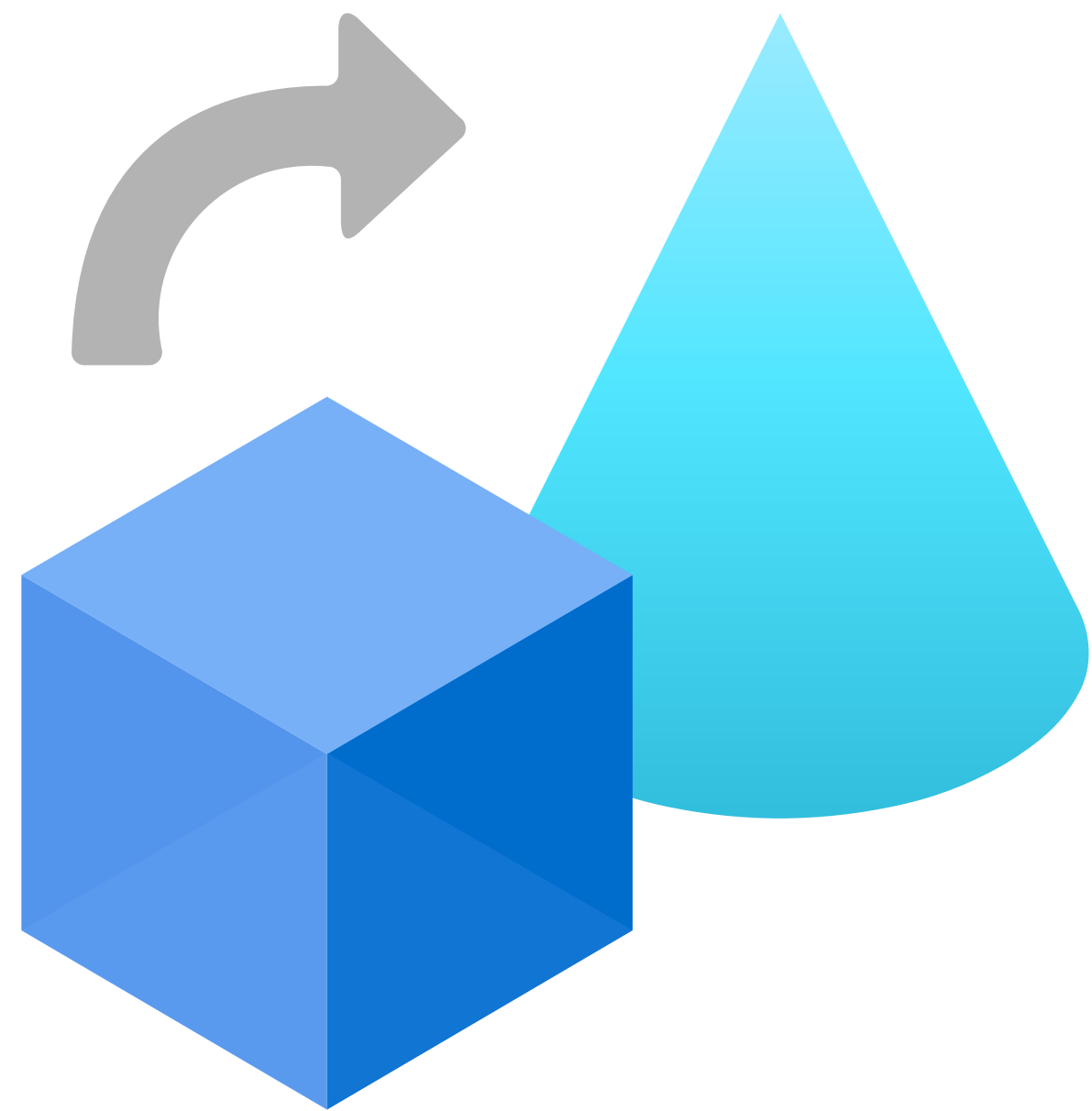


# Data Flows – Monitoring



# Data Flows – Monitoring





Data Flows

# Power Query *(Preview)*

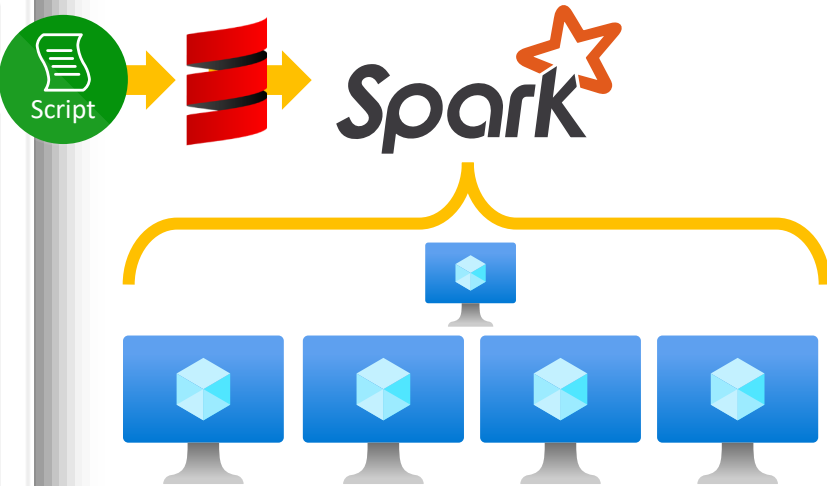
# What is a Power Query Activity?



Data Flow

The screenshot shows the Power Query Editor interface. The ribbon includes tabs for Home, Transform, Add column, and View. The main area displays a table with columns: SalesOrderID, SalesOrderDetailID, OrderQty, ProductID, UnitPrice, UnitPriceDiscount, LineTotal, and rowguid. The table contains 17 rows of data. The right sidebar shows 'Query settings' for 'LakeFileOrderDetailLinesP...' and 'Applied steps' including 'AdfDoc' and 'Parquet'.

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	71774	110562	1	836	356.898	0	356.898	e3a1994c-7a68-4ce8-96a3-77f
2	71774	110563	1	822	356.898	0	356.898	5c77f557-fdb6-43ba-90b9-9a7
3	71776	110567	1	907	63.9	0	63.9	6dbfe398-d15d-425e-aa58-88
4	71780	110616	4	905	218.454	0	873.816	377246c9-4483-48ed-a5b9-e5
5	71780	110617	2	983	461.694	0	923.388	43a54bcd-536d-4a1b-8e69-24
6	71780	110618	6	988	112.998	0.4	406.793	12706fab-f3a2-48c6-b7c7-1cc
7	71780	110619	2	748	818.7	0	1637.4	b12f0d3b-5b4e-4f1f-b2f0-f7cc
8	71780	110620	1	990	323.994	0	323.994	f117a449-039d-44b8-a4b2-b1
9	71780	110621	1	926	149.874	0	149.874	92e5052b-72d0-4c91-9a8c-42
10	71780	110622	1	743	809.76	0	809.76	8bd33bed-c4f6-4d44-84fb-a7c
11	71780	110623	4	782	1376.994	0	5507.976	686999fb-42e6-4d00-9a14-83i
12	71780	110624	2	918	158.43	0	316.86	82940b03-c70b-4183-8660-6b
13	71780	110625	4	780	1391.994	0	5567.976	644b0cd6-b2c3-4e4d-ab43-09
14	71780	110626	1	937	48.594	0	48.594	7f5feb17-8ef4-4236-9f1c-1504
15	71780	110627	6	867	41.994	0	251.964	ac78838d-b503-41a5-9791-48
16	71780	110628	1	985	112.998	0.4	67.799	2c10a282-a13d-442a-8f45-f4d
17	71780	110629	2	989	323.994	0	647.988	654fb79e-70df-4b92-9832-9fa



# What can a Power Query Activity do?

Control Flow



Data Flow

The screenshot shows the Power Query Editor interface. The ribbon at the top includes tabs for Home, Transform, Add column, and View. The Home tab is active, showing various options like Enter data, Options, Manage parameters, Refresh, Properties, Advanced editor, Manage, Choose columns, Remove columns, Keep rows, Remove rows, Sort, Split column, Group by, Data type, Use first row as headers, Replace values, Merge queries, Append queries, and Combine files. The main area displays a data table with the following columns: SalesOrderID, SalesOrderDetailID, OrderQty, ProductID, UnitPrice, UnitPriceDiscount, LineTotal, and rowguid. The table contains 17 rows of data. The right sidebar shows the 'Query settings' pane with the 'Name' field set to 'LakeFileOrderDetailLinesP...' and the 'Applied steps' list containing 'AdfDoc' and 'Parquet'.

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	71774	110562	1	836	356.898	0	356.898	e3a1994c-7a68-4ce8-96a3-77f
2	71774	110563	1	822	356.898	0	356.898	5c77f557-fdb6-43ba-90b9-9a7
3	71776	110567	1	907	63.9	0	63.9	6dbfe398-d15d-425e-aa58-88
4	71780	110616	4	905	218.454	0	873.816	377246c9-4483-48ed-a5b9-e5
5	71780	110617	2	983	461.694	0	923.388	43a54bcd-536d-4a1b-8e69-24
6	71780	110618	6	988	112.998	0.4	406.793	12706fab-f3a2-48c6-b7c7-1cc
7	71780	110619	2	748	818.7	0	1637.4	b12f0d3b-5b4e-4f1f-b2f0-f7cc
8	71780	110620	1	990	323.994	0	323.994	f117a449-039d-44b8-a4b2-b1.
9	71780	110621	1	926	149.874	0	149.874	92e5052b-72d0-4c91-9a8c-42
10	71780	110622	1	743	809.76	0	809.76	8bd33bed-c4f6-4d44-84fb-a7c
11	71780	110623	4	782	1376.994	0	5507.976	686999fb-42e6-4d00-9a14-83i
12	71780	110624	2	918	158.43	0	316.86	82940b03-c70b-4183-8660-6b
13	71780	110625	4	780	1391.994	0	5567.976	644b0cd6-b2c3-4e4d-ab43-09
14	71780	110626	1	937	48.594	0	48.594	7f5feb17-8ef4-4236-9f1c-1504
15	71780	110627	6	867	41.994	0	251.964	ac78838d-b503-41a5-9791-48
16	71780	110628	1	985	112.998	0.4	67.799	2c10a282-a13d-442a-8f45-f4d
17	71780	110629	2	989	323.994	0	647.988	654fb79e-70df-4b92-9832-9fa

# Power Query – Home

Control Flow



Data Flow

Power Query Editor interface showing data transformation steps and results.

**Queries:**

- ADFRResource [1]
- LakeFileOrderDetail...
- UserQuery

**Table: OrderDetailLines**

SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPrice
71774	110562	1	836	356.898	
71774	110563	1	822	356.898	
71776	110567	1	907	63.9	
71780	110616	4	905	218.454	
71780	110617	2	983	461.694	
71780	110618	6	988	112.998	
71780	110619	2	748	818.7	
71780	110620	1	990	323.994	
71780	110621	1	926	149.874	
71780	110622	1	743	809.76	
71780	110623	4	782	1376.994	
71780	110624	2	918	158.43	
71780	110625	4	780	1391.994	
71780	110626	1	937	48.594	
71780	110627	6	867	41.994	
71780	110628	1	985	112.998	
71780	110629	2	989	323.994	

**Query Settings:**

- NAME: OrderDetailLines
- APPLIED STEPS:
  - Source
  - Promoted Headers
  - Changed Type



# Power Query – Transform

Control Flow



Data Flow

Power Query Editor interface showing data transformation steps and results.

**Queries:** ADFResource [1], LakeFileOrderDetail..., UserQuery

**Table:** OrderDetailLines

SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice
71774	110562	1	836	356.898
71774	110563	1	822	356.898
71776	110567	1	907	63.9
71780	110616	4	905	218.454
71780	110617	2	983	461.694
71780	110618	6	988	112.998
71780	110619	2	748	818.7
71780	110620	1	990	323.994
71780	110621	1	926	149.874
71780	110622	1	743	809.76
71780	110623	4	782	1376.994
71780	110624	2	918	158.43
71780	110625	4	780	1391.994
71780	110626	1	937	48.594
71780	110627	6	867	41.994
71780	110628	1	985	112.998
71780	110629	2	989	323.994

**Query Settings:** Name: OrderDetailLines

**APPLIED STEPS:** Source, Promoted Headers, Changed Type

# Power Query – Add Column

Control Flow



Data Flow

Power Query Editor interface showing the 'Add Column' tab and a data table.

**Queries:**

- ADFRResource [1]
- LakeFileOrderDetail...
- UserQuery

**Table Data:**

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPrice
1	71774	110562	1	836	356.898	
2	71774	110563	1	822	356.898	
3	71776	110567	1	907	63.9	
4	71780	110616	4	905	218.454	
5	71780	110617	2	983	461.694	
6	71780	110618	6	988	112.998	
7	71780	110619	2	748	818.7	
8	71780	110620	1	990	323.994	
9	71780	110621	1	926	149.874	
10	71780	110622	1	743	809.76	
11	71780	110623	4	782	1376.994	
12	71780	110624	2	918	158.43	
13	71780	110625	4	780	1391.994	
14	71780	110626	1	937	48.594	
15	71780	110627	6	867	41.994	
16	71780	110628	1	985	112.998	
17	71780	110629	2	989	323.994	

**Query Settings:**

- NAME: OrderDetailLines
- APPLIED STEPS:
  - Source
  - Promoted Headers
  - Changed Type

# Power Query – View

Control Flow



Data Flow

Power Query Editor interface showing the View tab and data preview.

**Queries:**

- ADFResource [1]
- LakeFileOrderDetail...
- UserQuery

**Formula Bar:** = Parquet.Document(AdfDoc)

**View Tab Options:**

- ☒ Formula Bar
- ☐ Monospaced
- ☐ Column distribution
- ☒ Show whitespace
- ☐ Column profile
- ☐ Column quality
- ☐ Always allow
- ☐ Parameters
- ☐ Advanced Editor
- ☐ Query Dependencies

**Queries [1]:**

- OrderDetailLines

**Data Preview:**

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	UnitPrice
1	71774	110562	1	836	356.898	
2	71774	110563	1	822	356.898	
3	71776	110567	1	907	63.9	
4	71780	110616	4	905	218.454	
5	71780	110617	2	983	461.694	
6	71780	110618	6	988	112.998	
7	71780	110619	2	748	818.7	
8	71780	110620	1	990	323.994	
9	71780	110621	1	926	149.874	
10	71780	110622	1	743	809.76	
11	71780	110623	4	782	1376.994	
12	71780	110624	2	918	158.43	
13	71780	110625	4	780	1391.994	
14	71780	110626	1	937	48.594	
15	71780	110627	6	867	41.994	
16	71780	110628	1	985	112.998	
17	71780	110629	2	989	323.994	

**Query Settings:**

**PROPERTIES**

Name: OrderDetailLines

**APPLIED STEPS**

- Source
- Promoted Headers
- Changed Type

# Power Query – View (Advanced Editor)



Data Flow

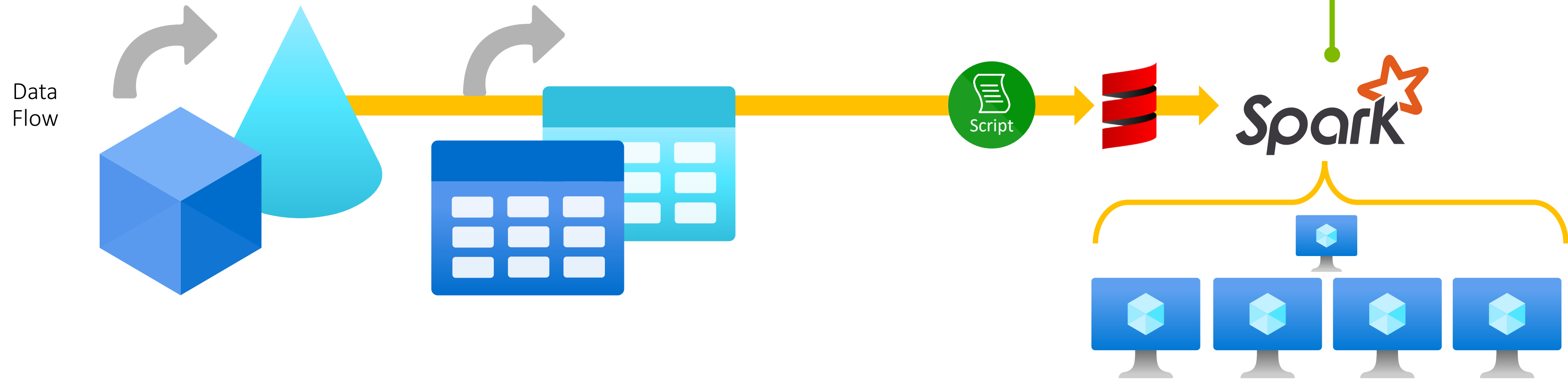
The screenshot shows the Power Query Advanced Editor interface. The top ribbon has tabs for 'Home', 'Transform', 'Add column', and 'View'. The 'View' tab is selected, and the 'Advanced editor' button is highlighted. The main area displays a list of queries on the left, including 'ADFResource', 'LakeFileOrderDetailL...', and 'UserQuery'. The 'Advanced editor' window is open, showing a M query script. The script is as follows:

```
1 let
2   AdfDoc = Web.Contents("https://traininglake01.dfs.core.windows.net/datawarehouse/Raw/OrderDetailLines.parquet"),
3   Parquet = Parquet.Document(AdfDoc),
4   #"Grouped rows" = Table.Group(Parquet, {"SalesOrderID"}, {"Count", each Table.RowCount(_), Int64.Type})
5 in
6   #"Grouped rows"
```

The 'Advanced editor' window has 'OK' and 'Cancel' buttons at the bottom right.

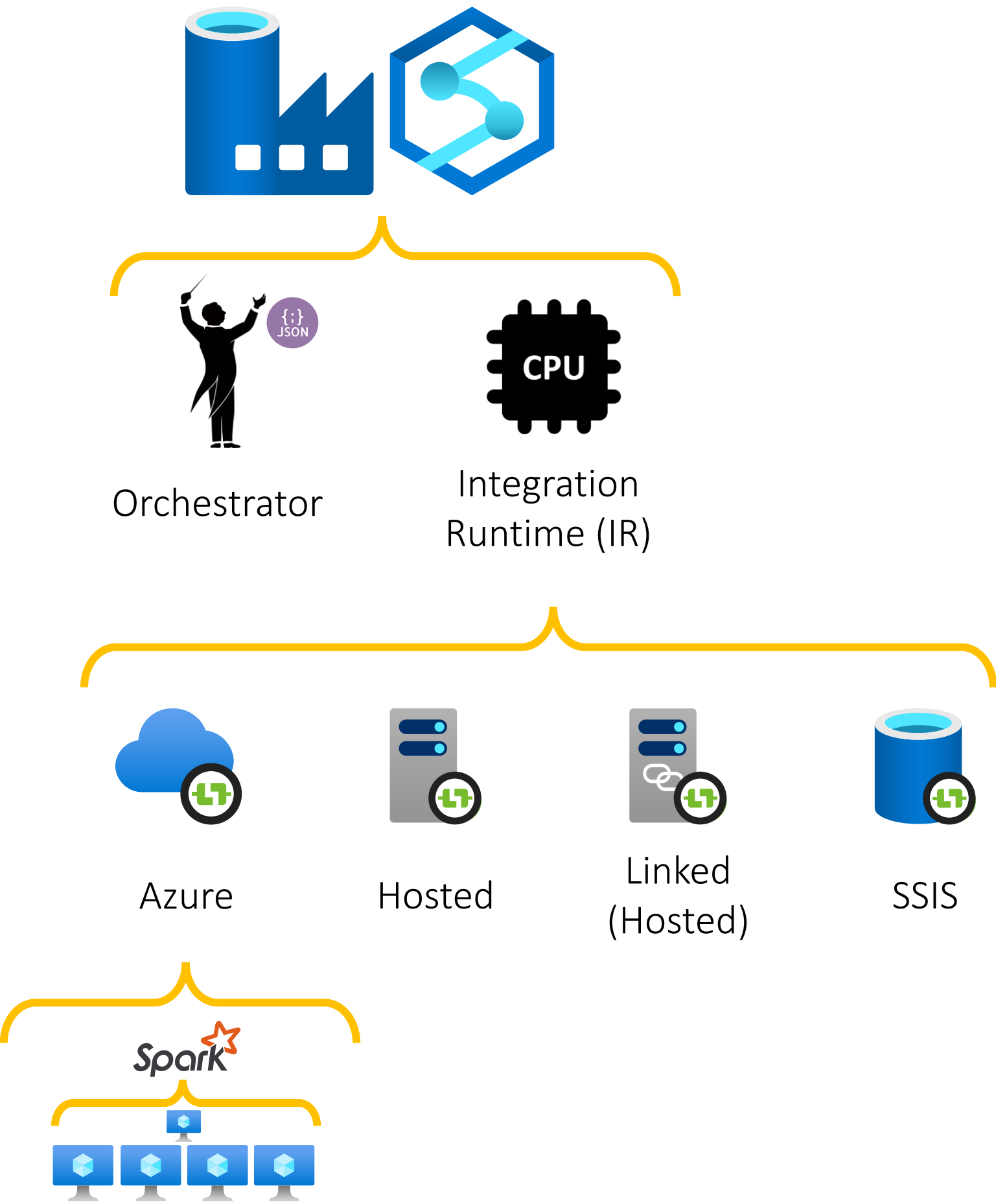
Configuration

# Data Flow Cluster Configuration

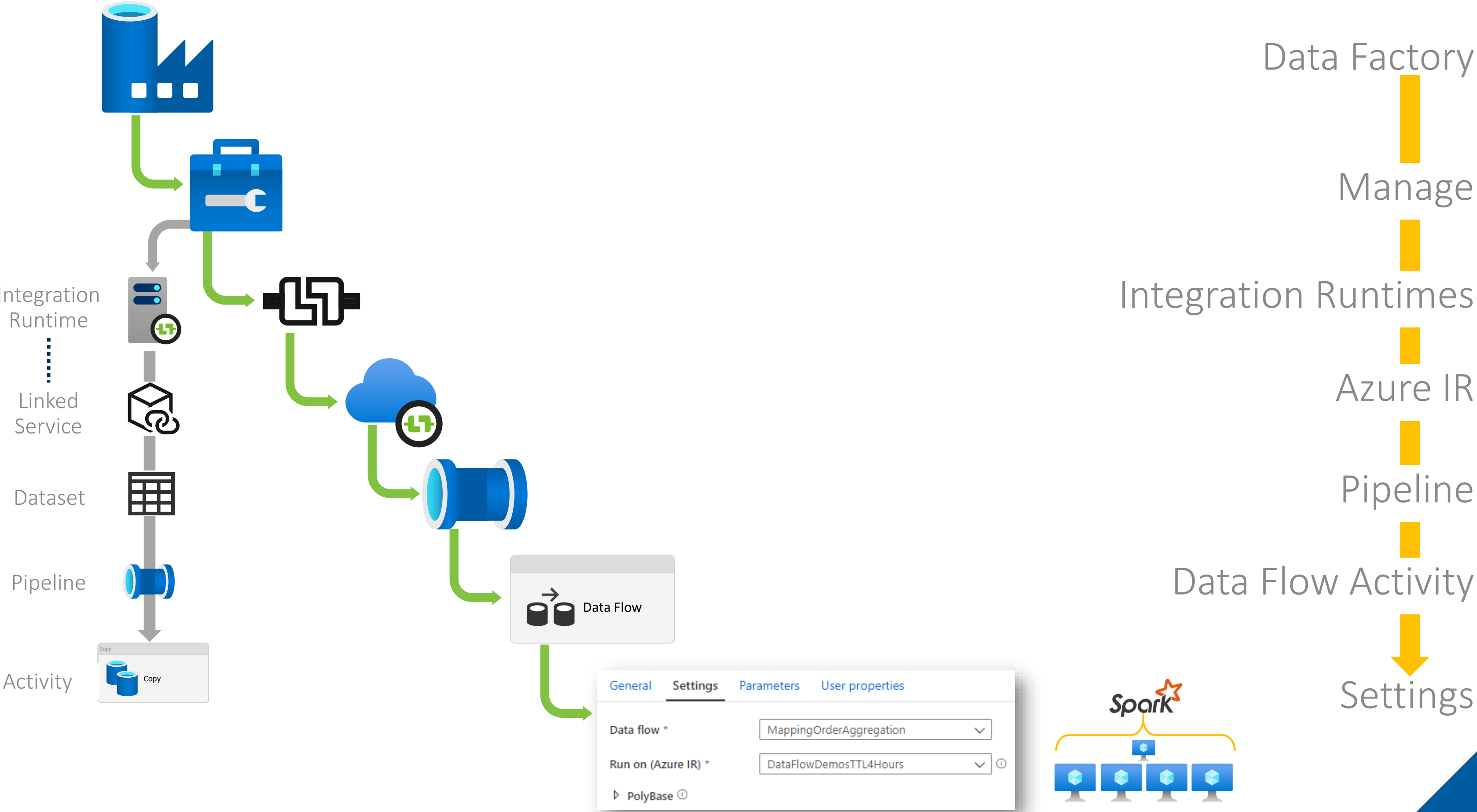


- Compute Type
- Number of Worker Nodes
- Cluster Time to Live

# Integration Runtimes



# Setting the Data Flow Cluster (IR Configuration)



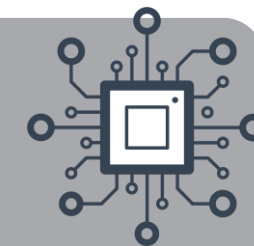



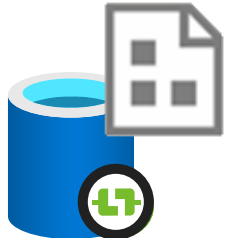

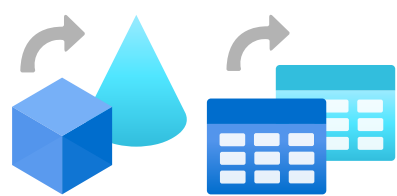
# Use Cases & Conclusions



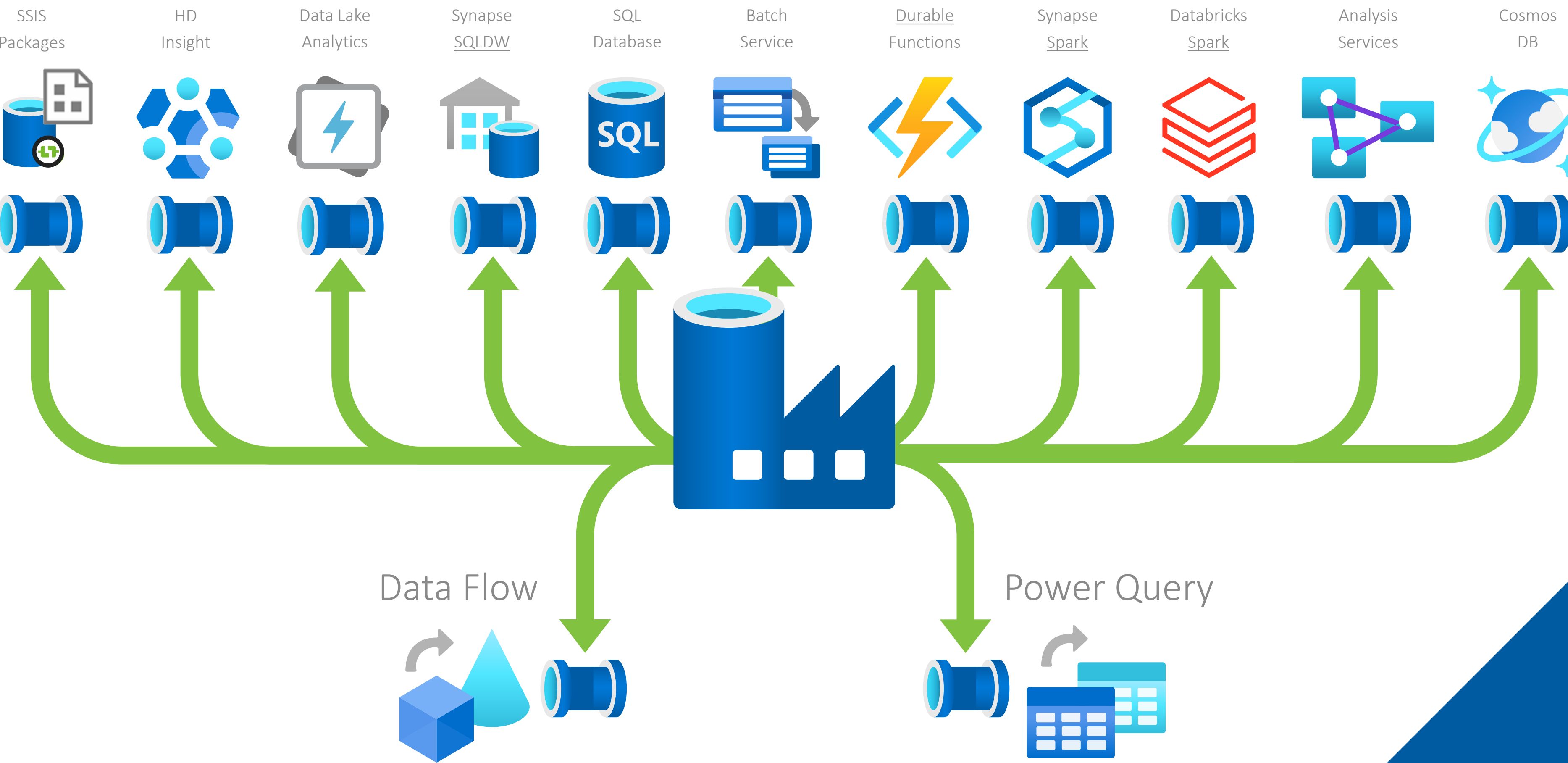


# Data Transformation Services in Azure Comparison



Transformation Tools		Graphical UI (Low/No Code)	Scales Out	Scales Up	Cloud Native Tech
	T-SQL with SQLDB	✗	✗	✓	✗
	SSIS Packages	✓	✗	✓	✗
	Scala/Python/SQL with Databricks	✗	✓	✓	✓
	Data Flows & Power Query	✓	✓	✓	✓

# When Should We Use These Integration Pipeline Transformation Activities?



# Use Cases

SSIS developers who are transferring existing skills to cloud native technologies have a very low barrier to entry and don't need to worry about distributed compute to get started.

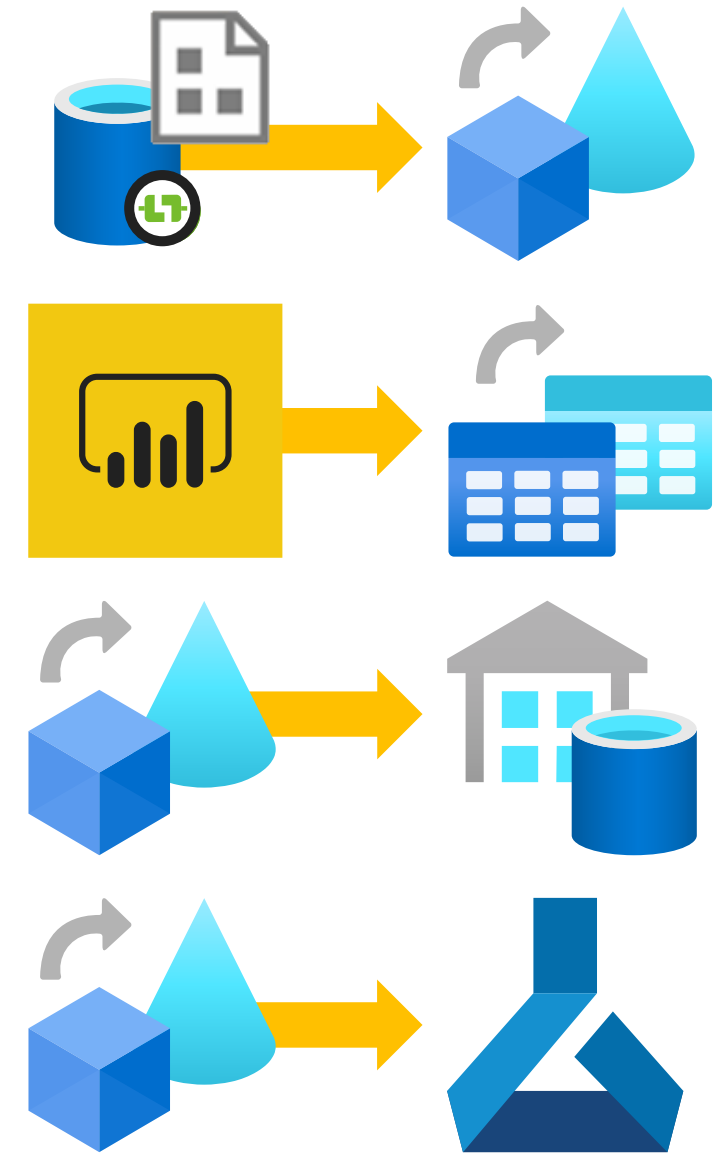
Data engineering made easy for the power users who has grown out of Power BI following a series of Data Lake exploration sessions.

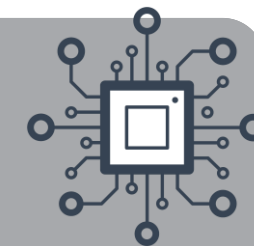
Data insight teams needing to do rapid prototyping and data warehouse loading within a single Azure Resource making deployments simple and release cycles short.

Simpler and quicker data wrangling for data scientists that want to quickly prepare multiple raw datasets ready for model training and testing, also with the ability to use large amounts of compute.

*Data Flows used to deliver all data transformation workloads as part of a end to end cloud based data analytics/warehouse solution.*

*Data Flows script dynamically generated from external metadata and injected into like we once did with BML for SSIS packages.*





# Thank you for listening...

Paul Andrew



**Blog:** [mrpaulandrew.com](http://mrpaulandrew.com)  
**YouTube:** [c/mrpaulandrew](https://www.youtube.com/c/mrpaulandrew)  
**Email:** [paul@mrpaulandrew.com](mailto:paul@mrpaulandrew.com)

**Twitter:** [@mrpaulandrew](https://twitter.com/mrpaulandrew)  
**LinkedIn:** [In/mrpaulandrew](https://www.linkedin.com/in/mrpaulandrew)

**GitHub:** [github.com/mrpaulandrew](https://github.com/mrpaulandrew)