THANK YOU SPONSORS!

Platinum



Gold









Bronze









Empowering Lakehouse Solutions with Python in Fabric

22th of January 2026

Christian Henrik Reich

(Christian) Henrik Reich

- Sr Solution Engineer @ Microsoft
- Redi-School teacher. Teaching Microsoft AI & AL Tech immigrant and refugee women in Denmark
- Ex-Pluralsight assessment test author
- Øredev conference program committee member since 2019 (Malmø)
- Agile Global Summit 2025 program committee member (Tallinn)
- Pass Data Community Summit Program committee member 2023 & 2025 (Seattle)
- Holding 23 active and 3 historical Microsoft Certificates
- Been developing almost everything ☺️

Delta and Parquet: Integer, GUID/UUID or SHA256 as ID?

Christian Henrik Reich 4 min read · May 7, 2024

9 1



Lakehousing: A Schema-change strategy beyond Spark's mergeSchema

Christian Henrik Reich 7 min read · May 12, 2025

44



Spark performance: Let cache() or persist() handle your temporary data when possible

Christian Henrik Reich 6 min read · Jul 19, 2024

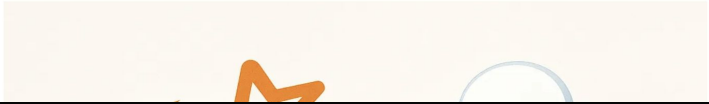
6 1



Microsoft Fabric: Consider making spark.createDataframe obsolete in your Spark Notebook solutions

Christian Henrik Reich 5 min read · May 4, 2025

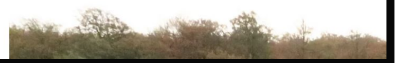
26



Delta and Parquet: GUID/UUID or SHA

Christian Henrik Reich 4 min read · May 7, 2024

9 1



Microsoft Fabric: Introduction to 3D programming

Christian Henrik Reich 5 min read · Jun 5, 2025

```
136  axes = fig.add_subplot(111, projection='3d')
137
138  clear_mattplot_axes(axes)
139
140  precalc_animation = animation.FuncAnimation(fig, partial(ani
141  precalc_animation.save("textured_cube_with_edges.gif", write
142
143  plt.close(fig)
144
145  Image(filename="textured_cube_with_edges.gif")
146
```

[11] ✓ 15 sec - Command executed in 15 sec 867 ms by Henrik Reich on 10:07:55 PM, 6/05/25



chema-change Spark's

, 2025

Spark performance persist() handle you data when possible

Christian Henrik Reich 6 min read · Jul 19, 2024

6 1



Consider making frame obsolete in ook solutions

2025

O'REILLY®

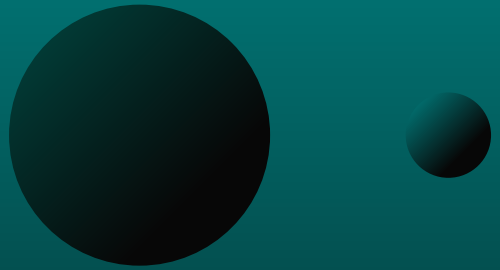
The Data Engineer's Guide to Microsoft Fabric

Building Modern Lakehouse Solutions



Christian Henrik Reich

AGENDA



1 Concepts and internals

2 Procedural Data Processing

3 Introduction to Apache Arrow

4 Avoid overloading the tech stack

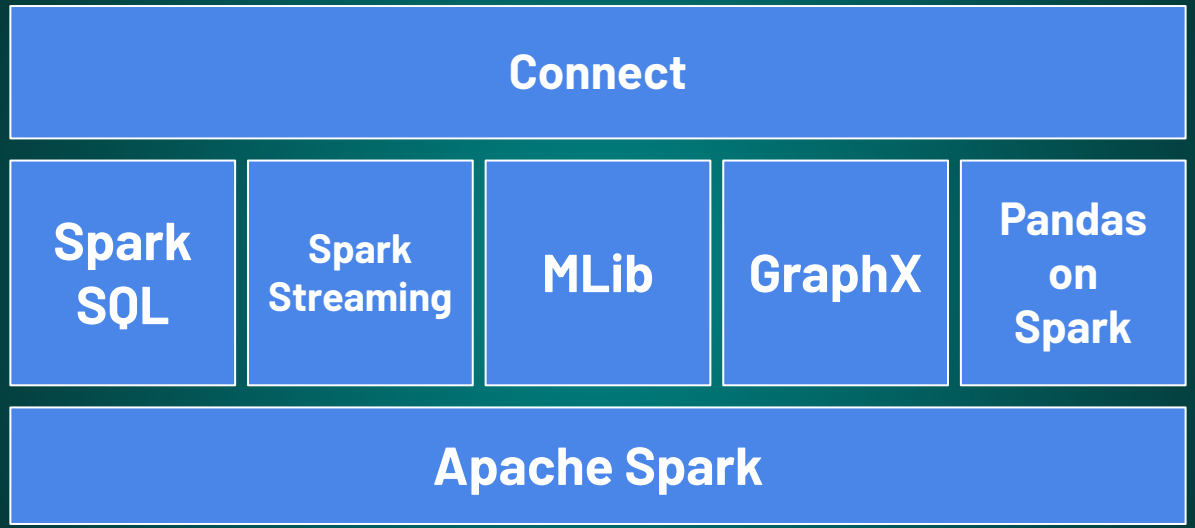
5 Metadata driven lakehousing

6 Q & A

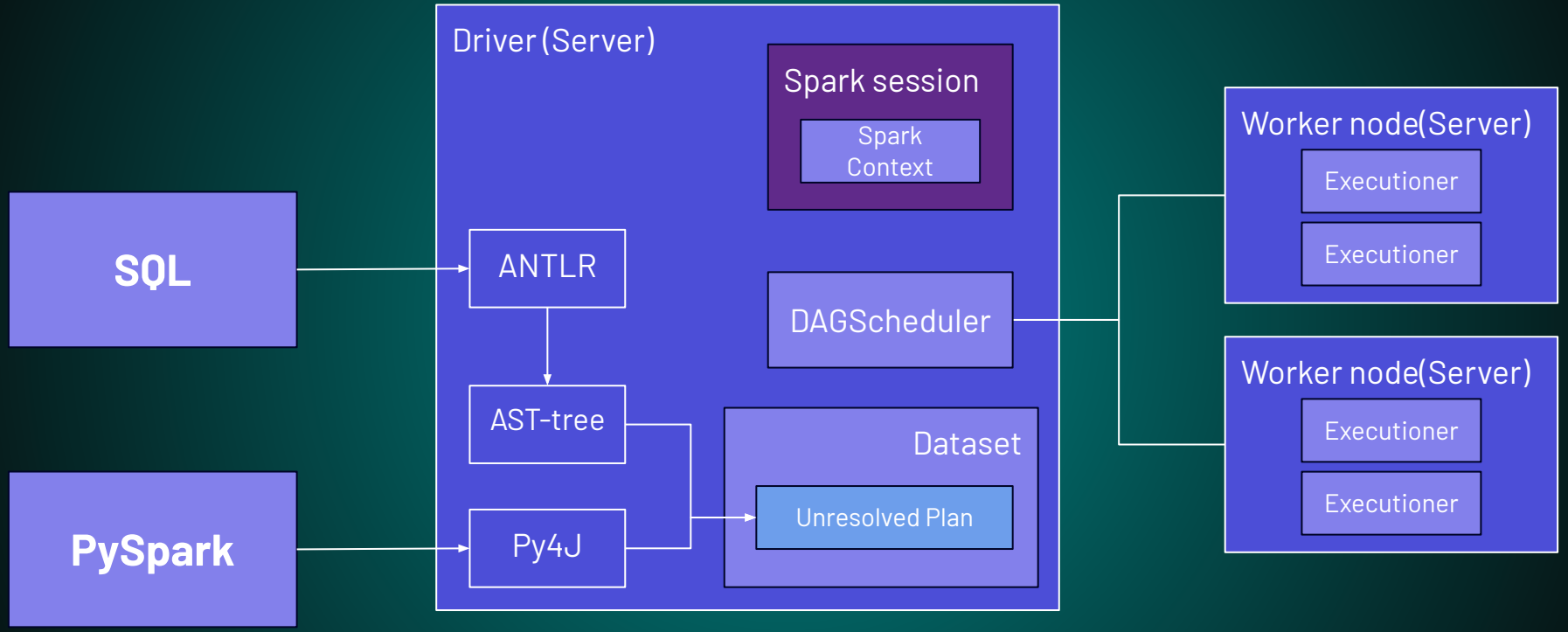
Concepts and principles

Apache Spark

Apache Spark



Apache Spark

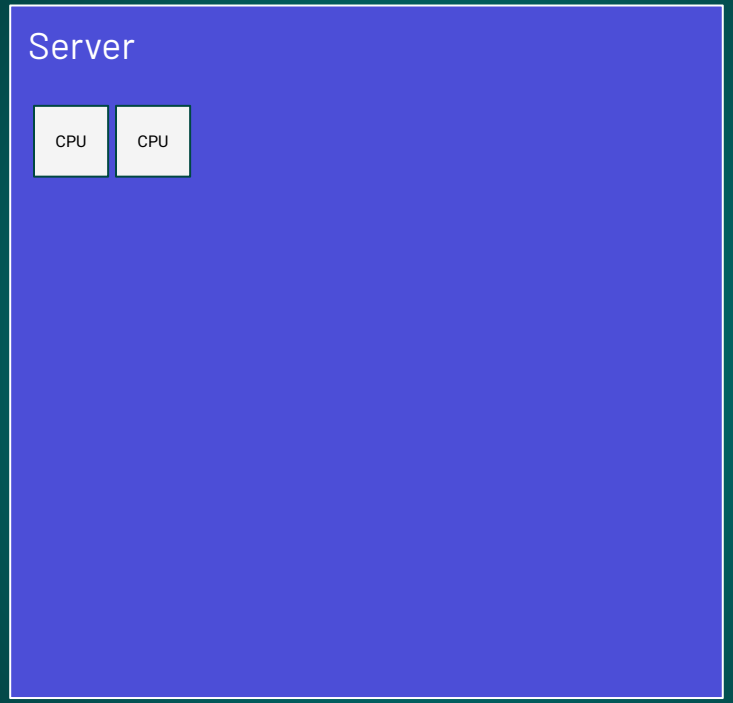


Python

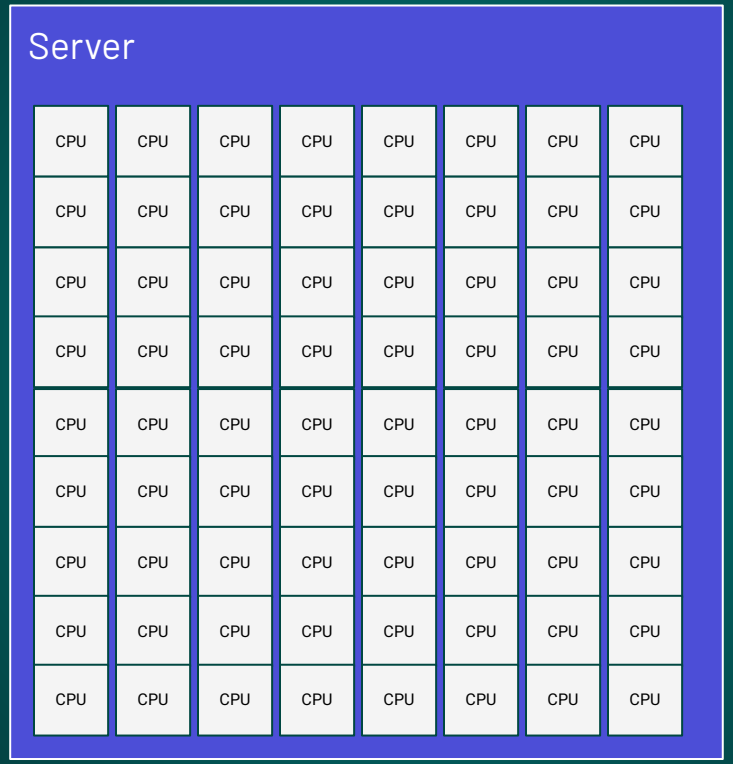
Python



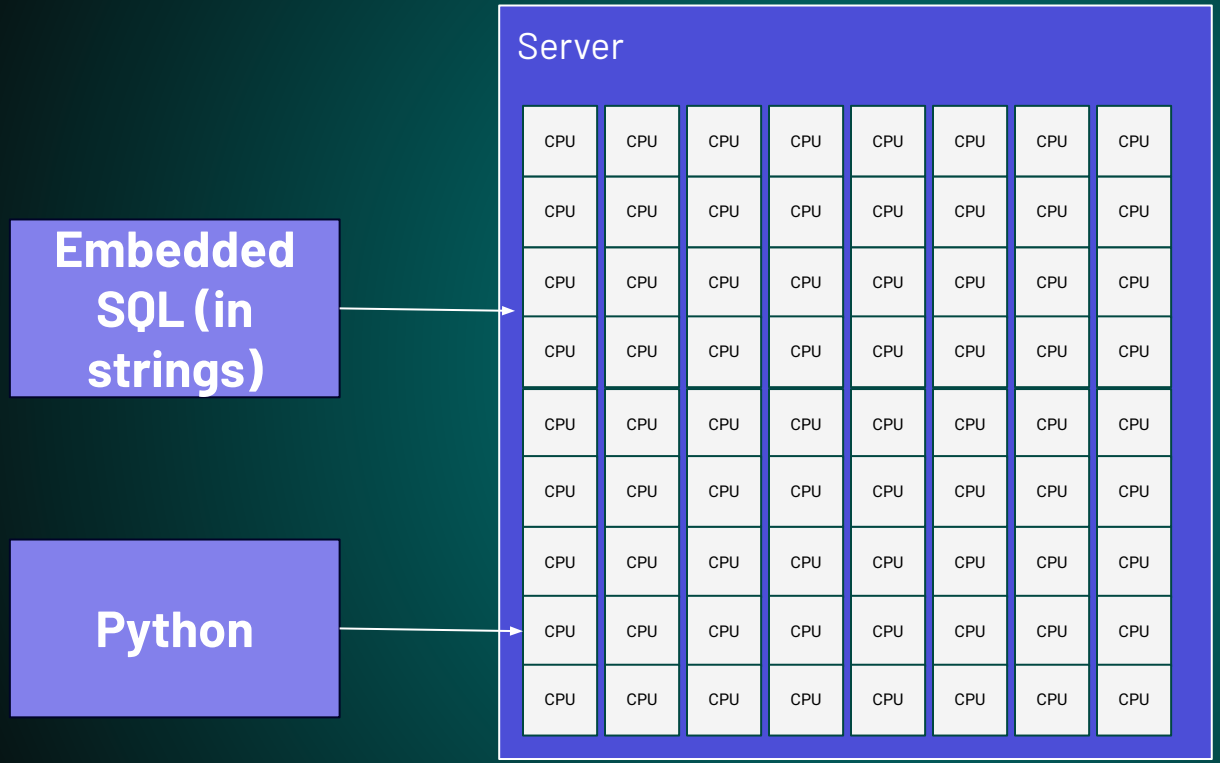
Python



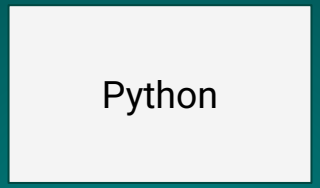
Python



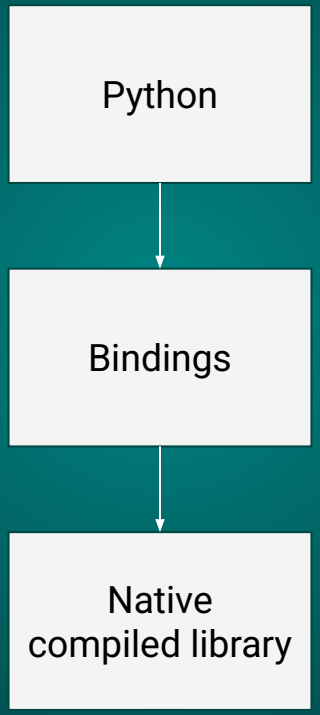
Python



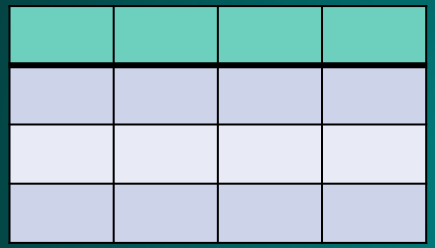
Python



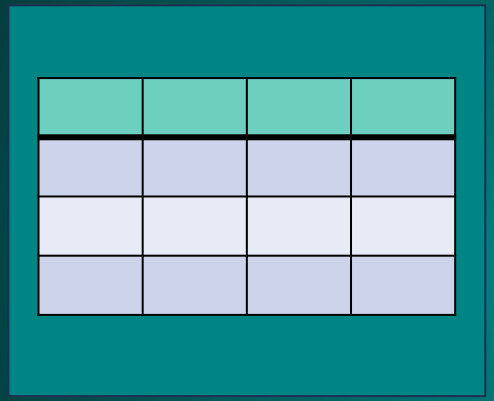
Python



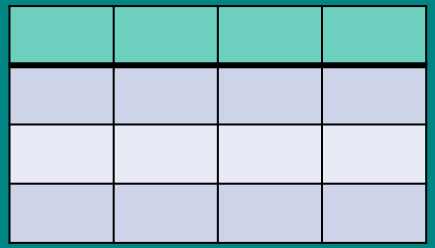
Dataframes and Dataframes



Compute memory



Compute memory

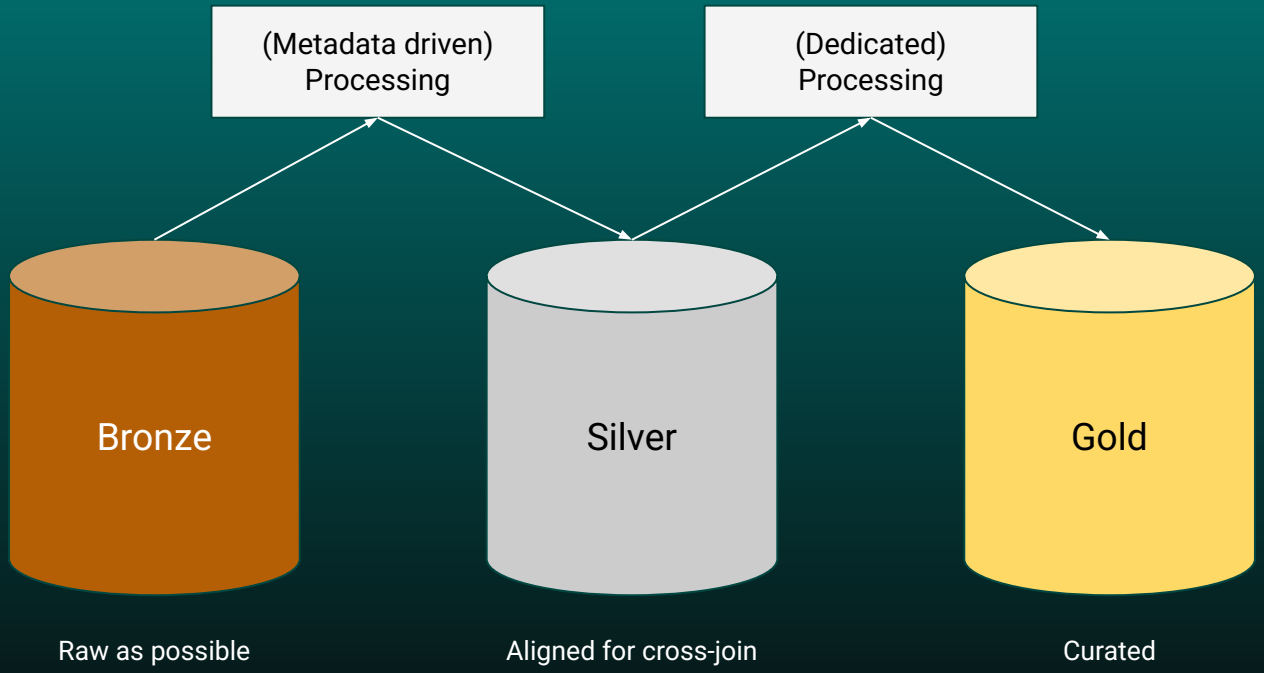


Lazy Dataframe

Read data
Process data
Write data

Procedural Data processing

Copy-ingest to Bronze



Copy-ingest to Bronze



```
/customers
    /year=2025
        /month=05
        /month=06
            /day=01
                /customers.csv
```



id	name	…	year	month	day
1	Alice	…	2025	06	1
2	Bob	…	2025	06	1
3	Carla	…	2025	06	1

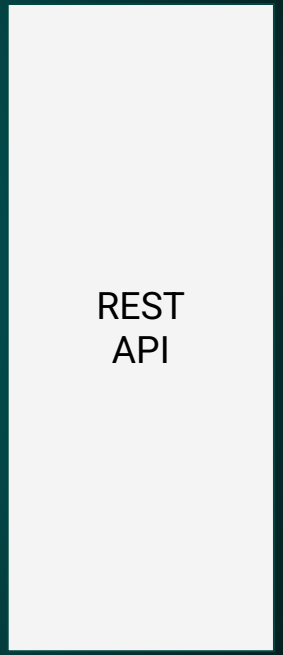
```
/customers_ingest
  /year=2025
    /month=05
    /month=06
      /day=01
        /customers.csv
```



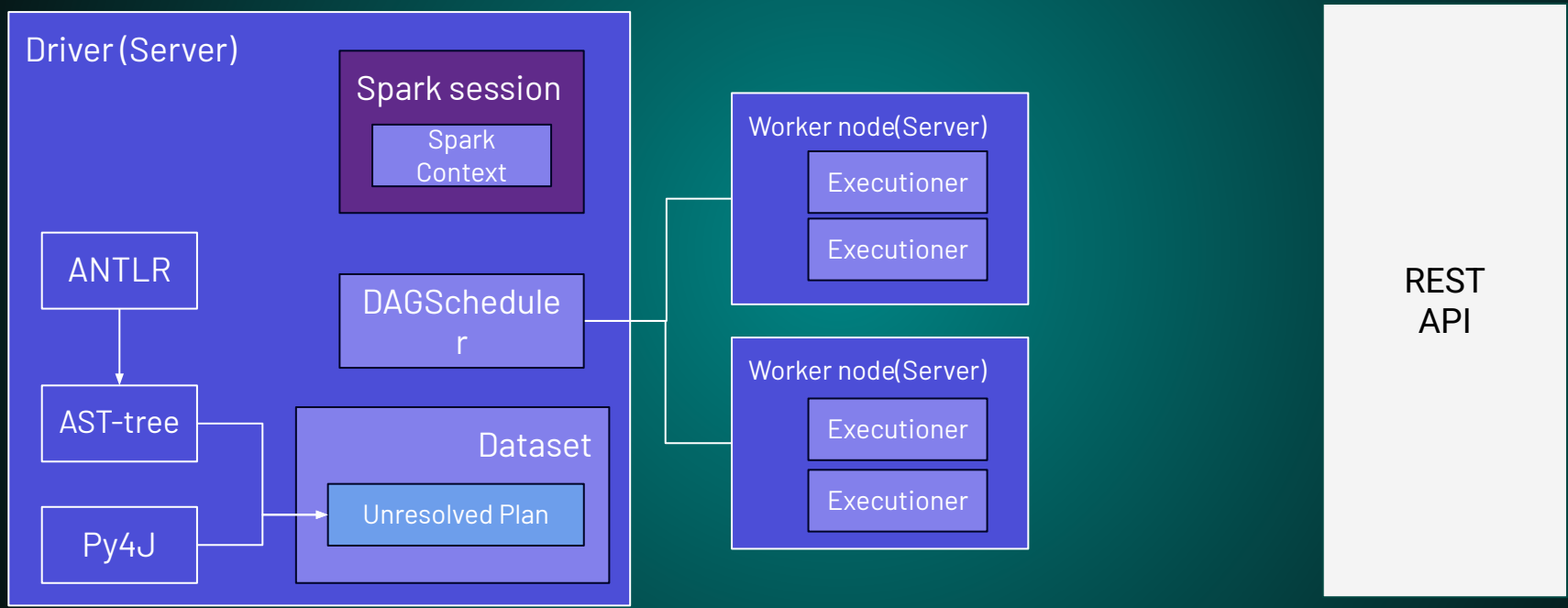
id	name	…	year	month	day
1	Alice	…	2025	06	1
2	Bob	…	2025	06	1
3	Carla	…	2025	06	1

API-ingest to Bronze

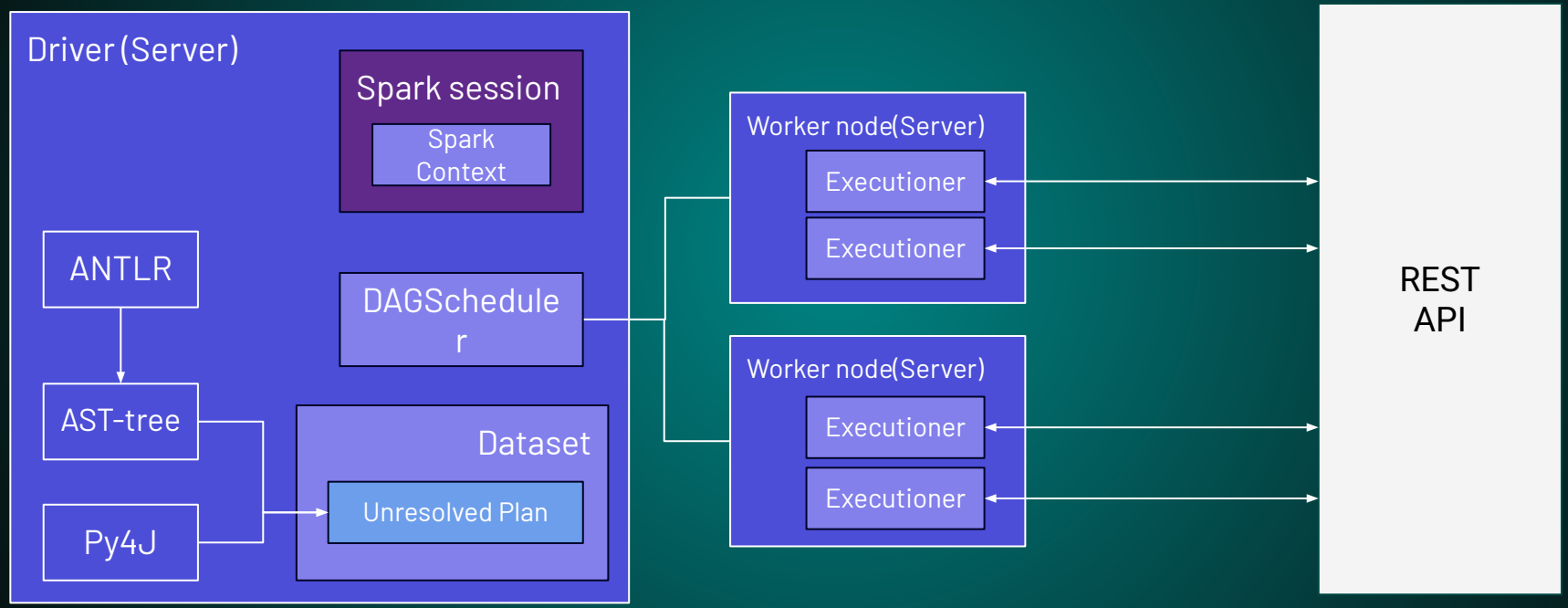
REST API-ingest to Bronze



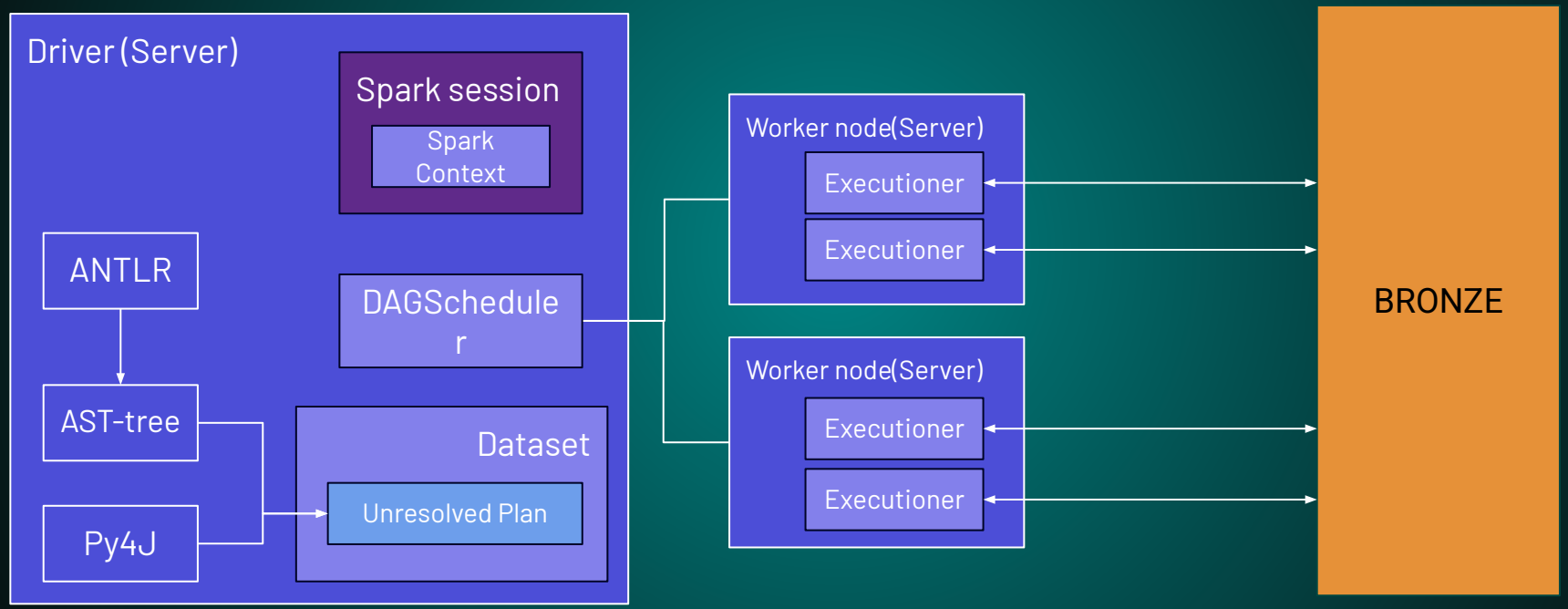
REST API-ingest to Bronze



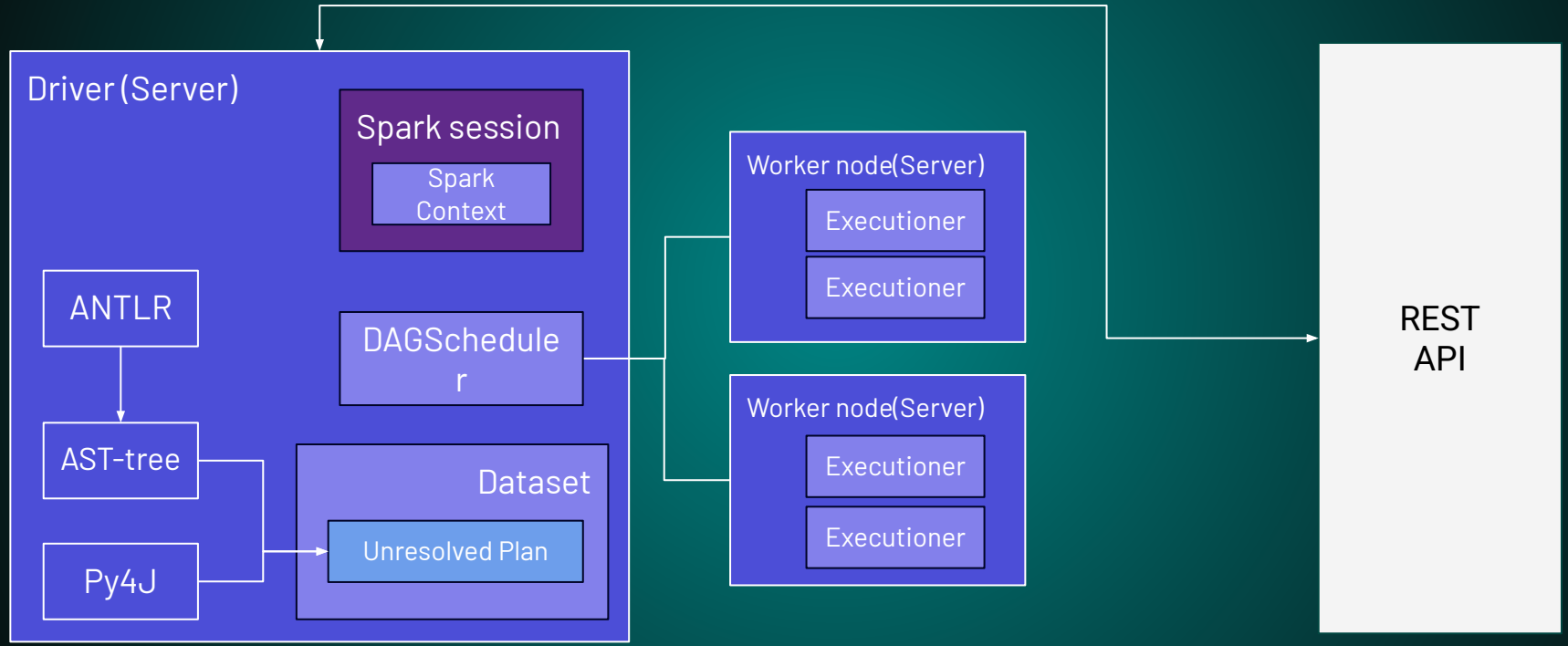
REST API-ingest to Bronze



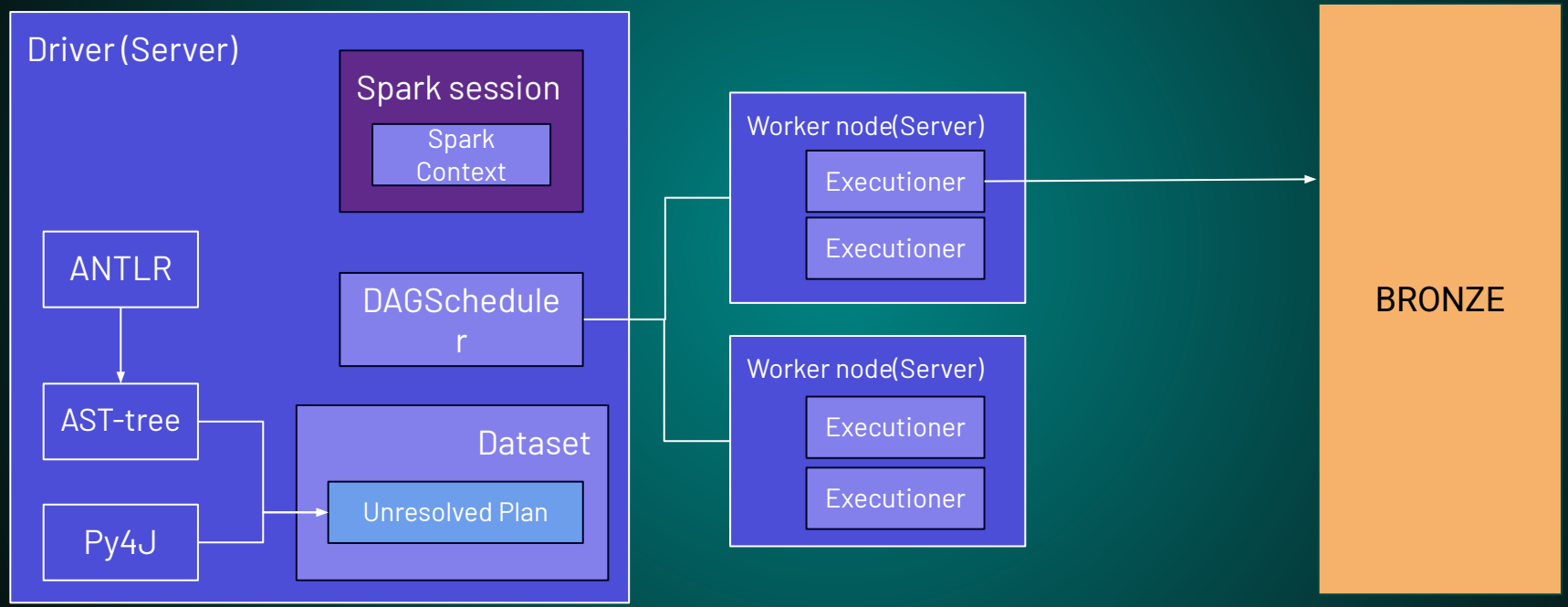
REST API-ingest to Bronze



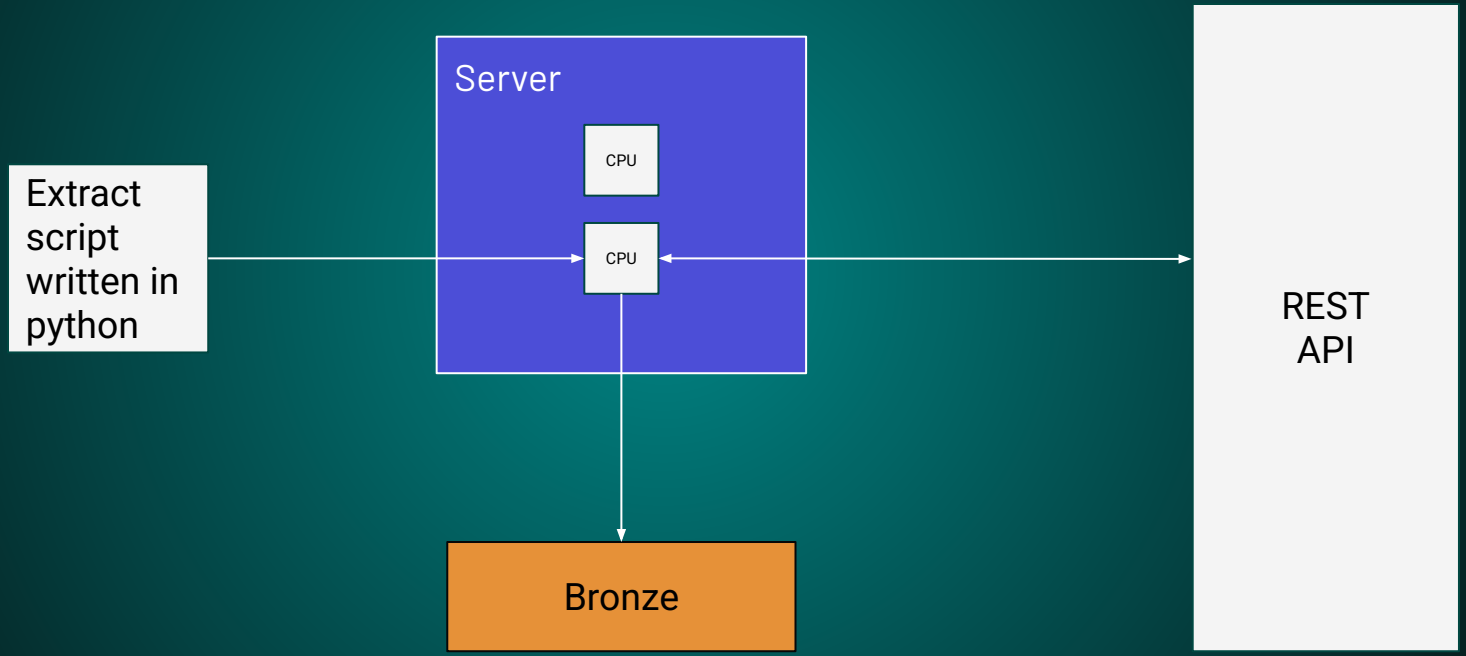
REST API-ingest to Bronze



REST API-ingest to Bronze



REST API-ingest to Bronze





Just Blindbæk: Fabric Spark Notebooks and CU consumption

https://justb.dk/blog/2025/02/fabric-spark-notebooks-and-cu-consumption/

Small files

15503 JSON files

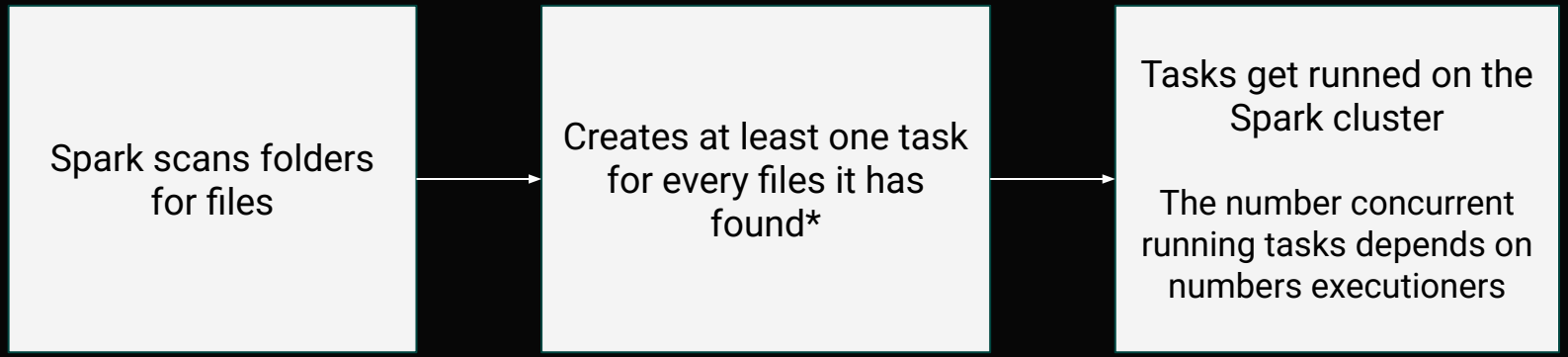
```
1  df = spark.read.option("multiline", "true").json("Files/customer_dataset_small_files/")
2  df.write.saveAsTable('Customers_from_small_files')
```

[*] 24 min 4 sec - Running

PySpa

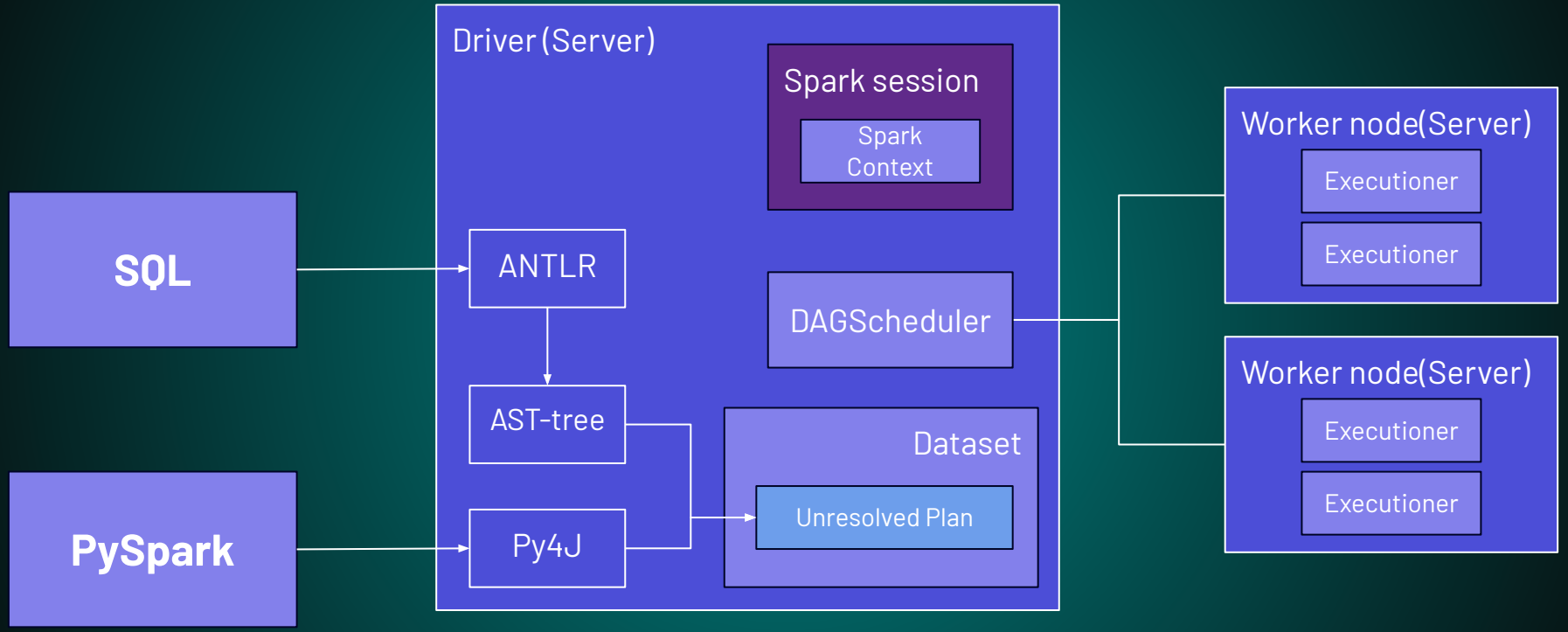
Spark jobs In progress (1) Resources Log

	ID ↑	Description	Status	Stages	Tasks	Duration	Rows	Data read	Data writ
>	Job 9	json at NativeMethodAccessorImpl.java:0	In progress	0/1 (1 active)	0/1 succeeded, 1 running	21 min 15 sec 102 ms	10928	50.91 MB	0 B



* 4Mb thredshold

Apache Spark



Introduction to Apache Arrow

The unsung hero



The unsung hero

Pandas

Apache Spark

Apache Flink

Polars

ClickHouse

DuckDB

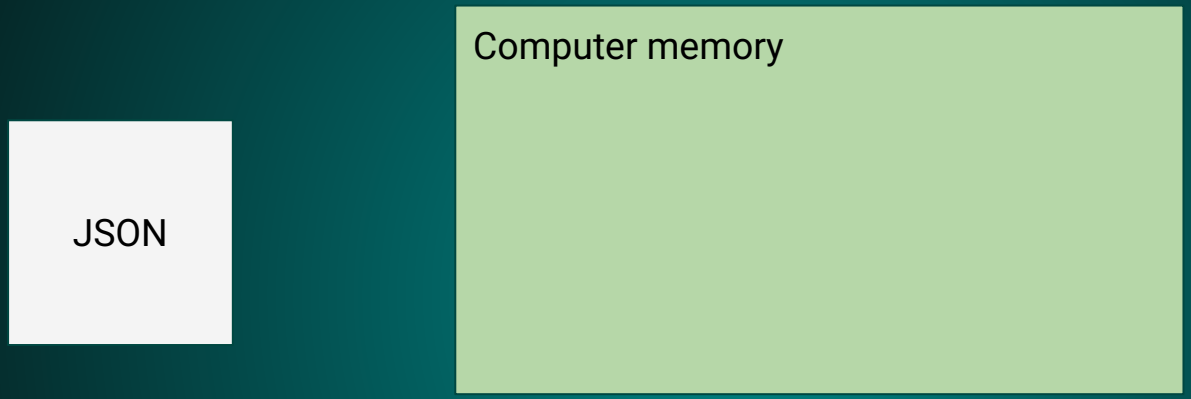
Apache DataFusion

Velox

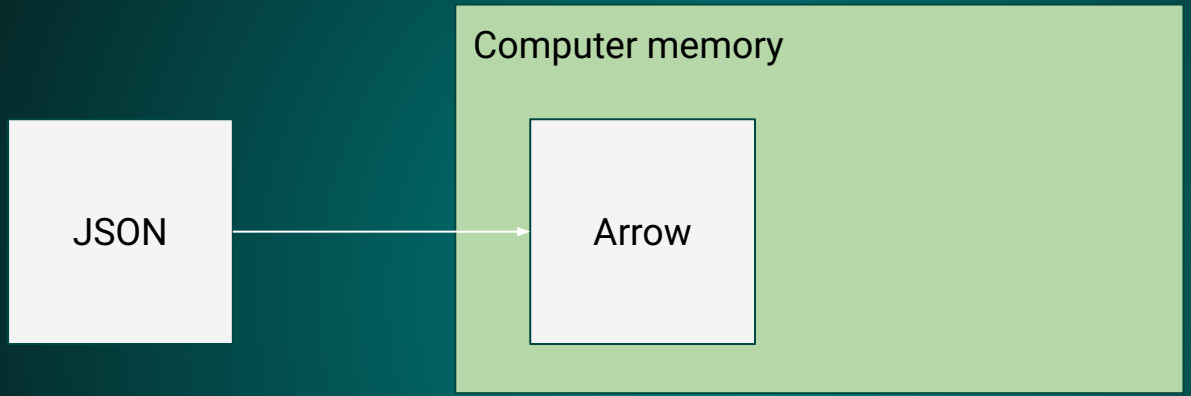
Dask

+More

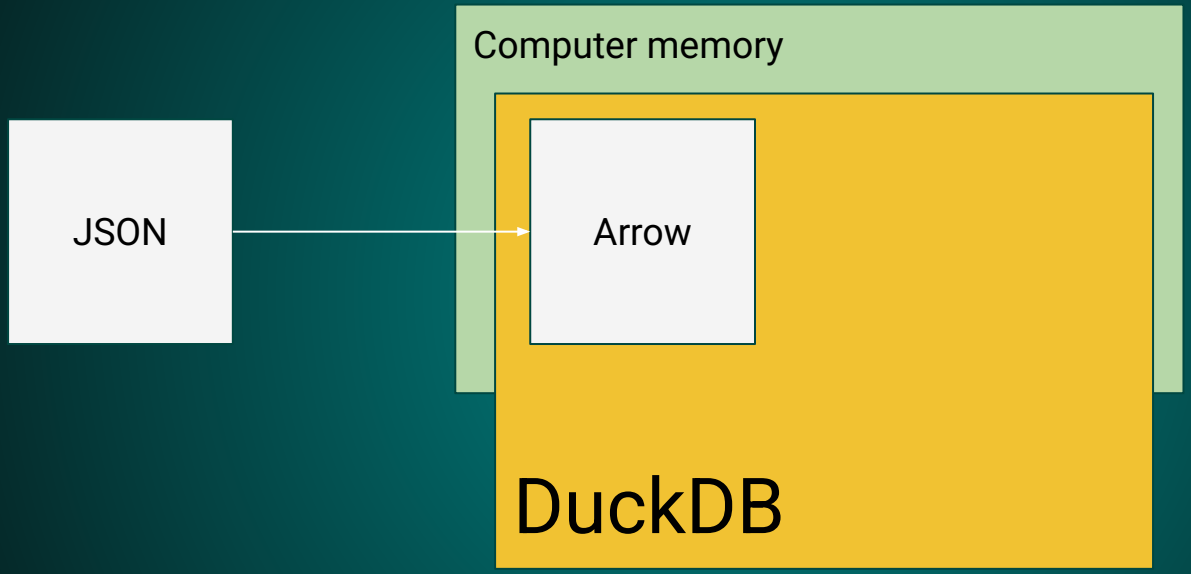
Apache Arrow



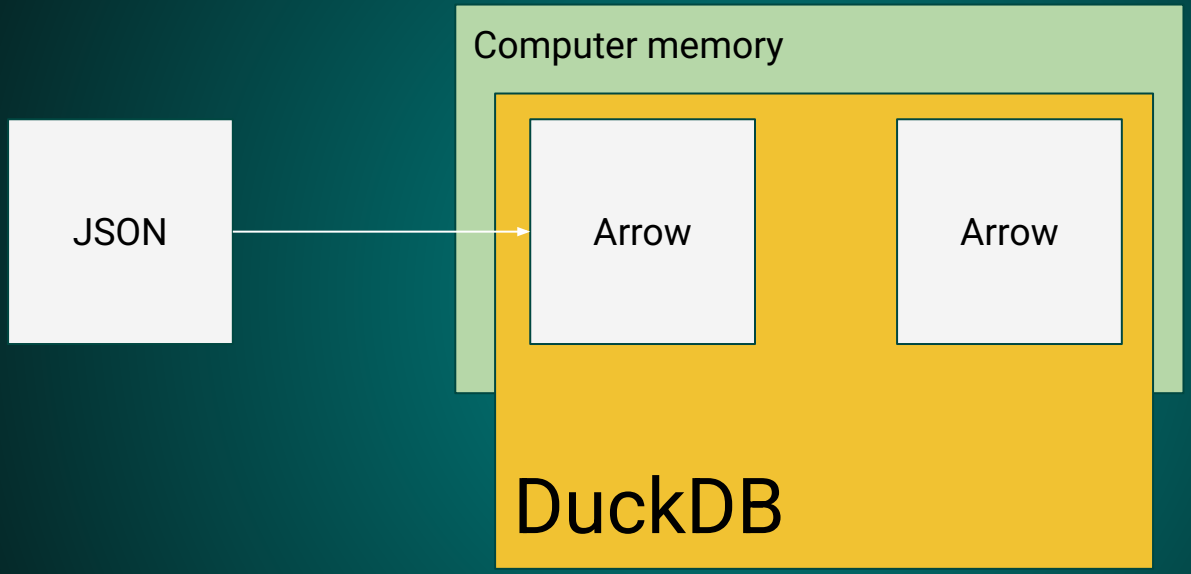
Apache Arrow



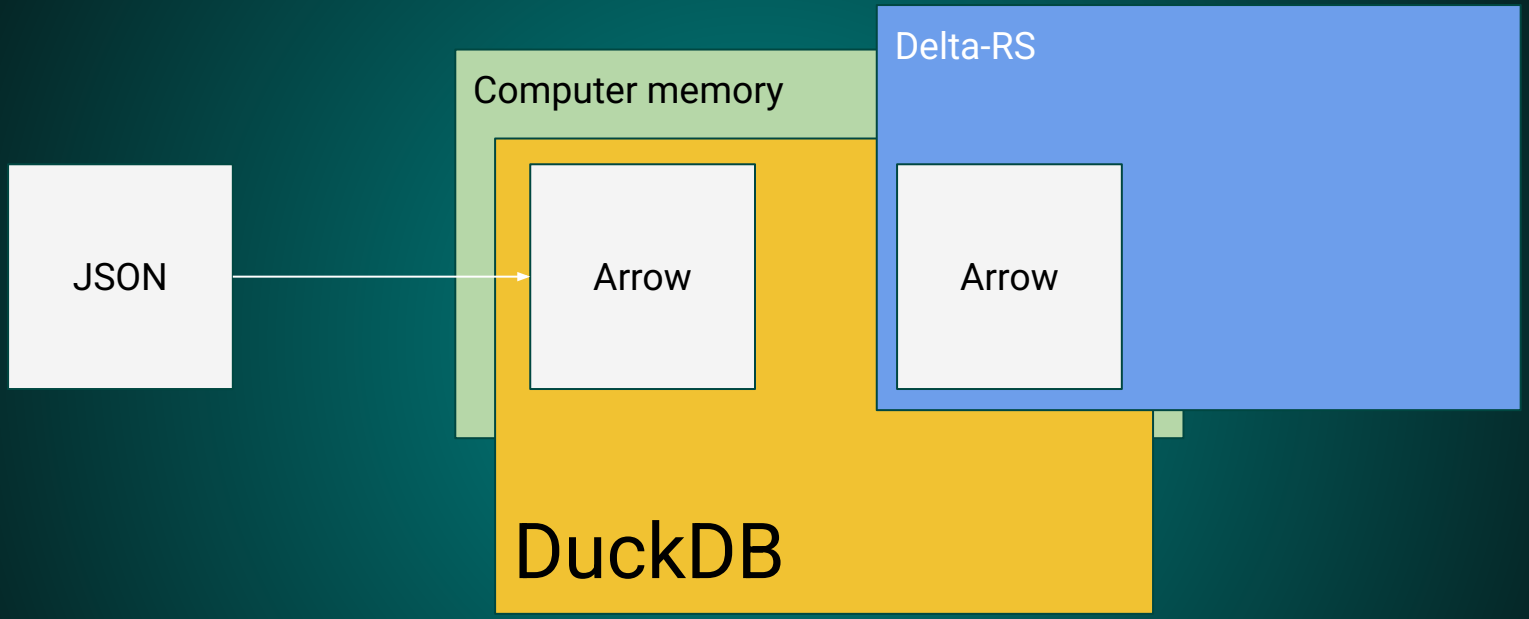
Apache Arrow



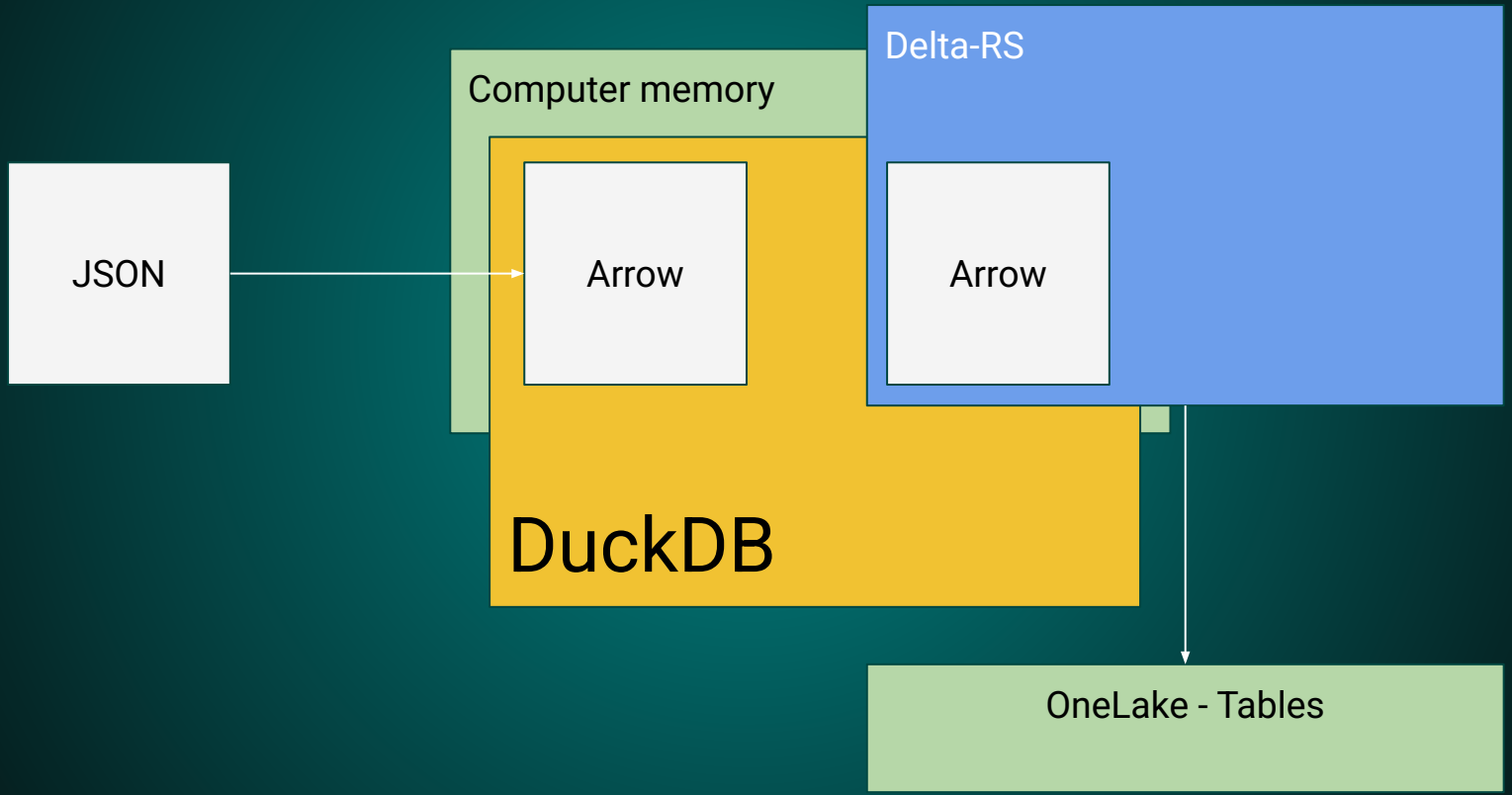
Apache Arrow

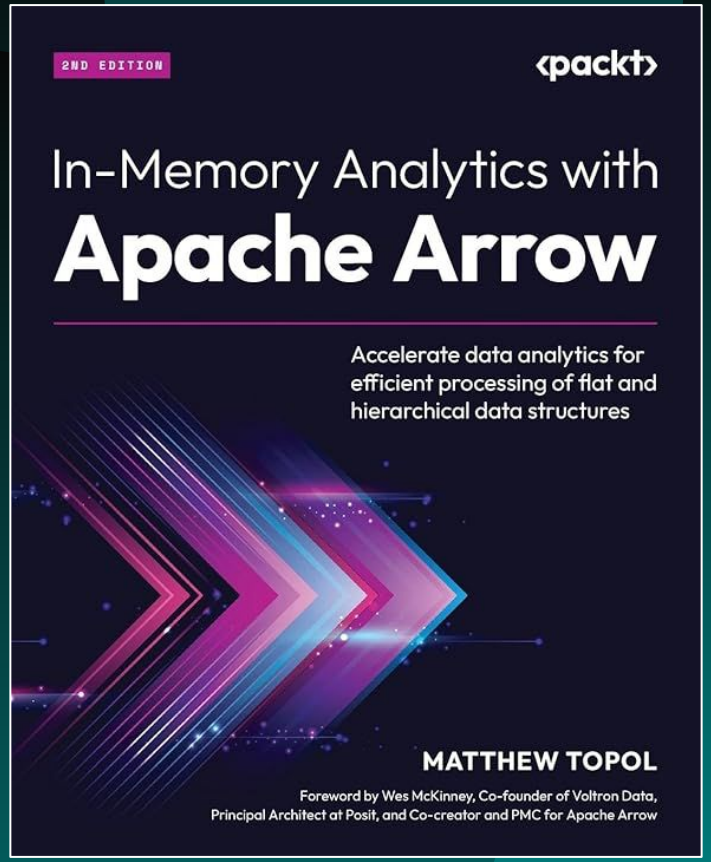


Apache Arrow



Apache Arrow




2ND EDITION
packt
In-Memory Analytics with
Apache Arrow
Accelerate data analytics for efficient processing of flat and hierarchical data structures
MATTHEW TOPOL
Foreword by Wes McKinney, Co-founder of Voltron Data, Principal Architect at Posit, and Co-creator and PMC for Apache Arrow

Avoid overloading the Python techstack

Jobs to be done

Jobs	Framework
Hive partitioning support	Apache Arrow
Delta Lake/Table	Delta-RS*
Dataframes	Polars*
SQL	DuckDB*

* Apache Arrow compatible

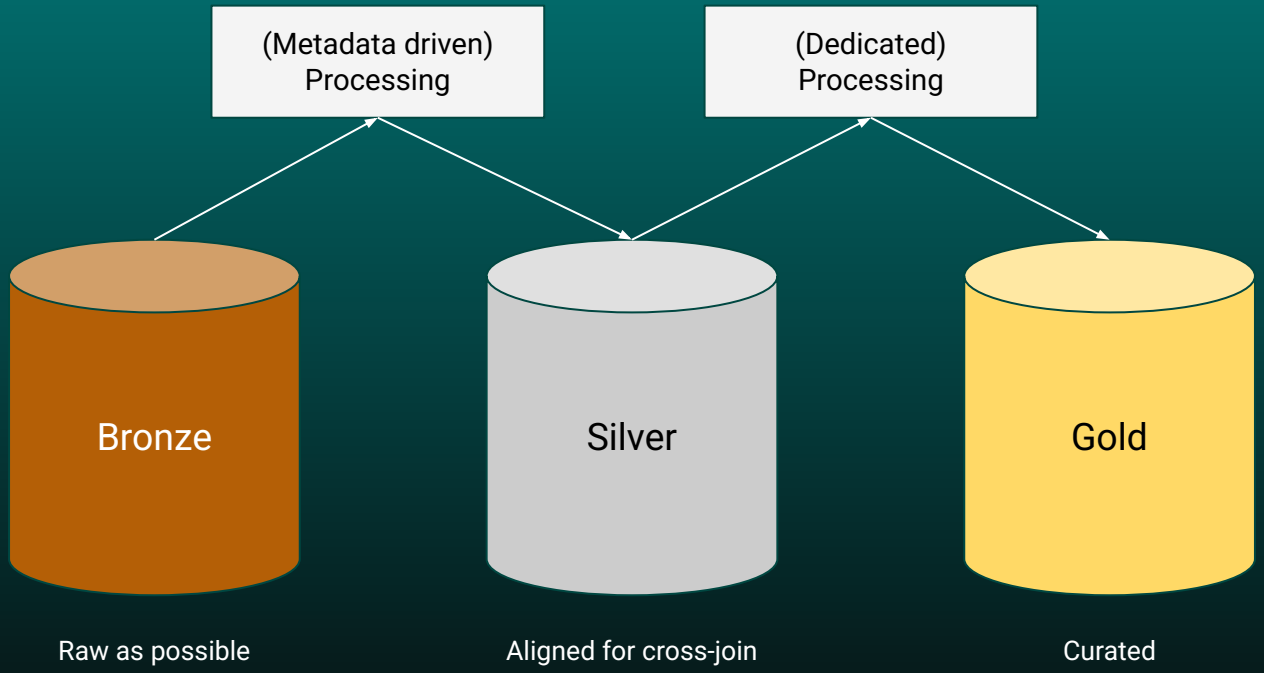
Jobs to be done

Jobs	Framework
Hive partitioning support	Apache Arrow
Delta Lake/Table	Delta-RS*
Dataframes	Apache DataFusion*
SQL	Apache DataFusion*
No data ownership	Apache DataFusion*

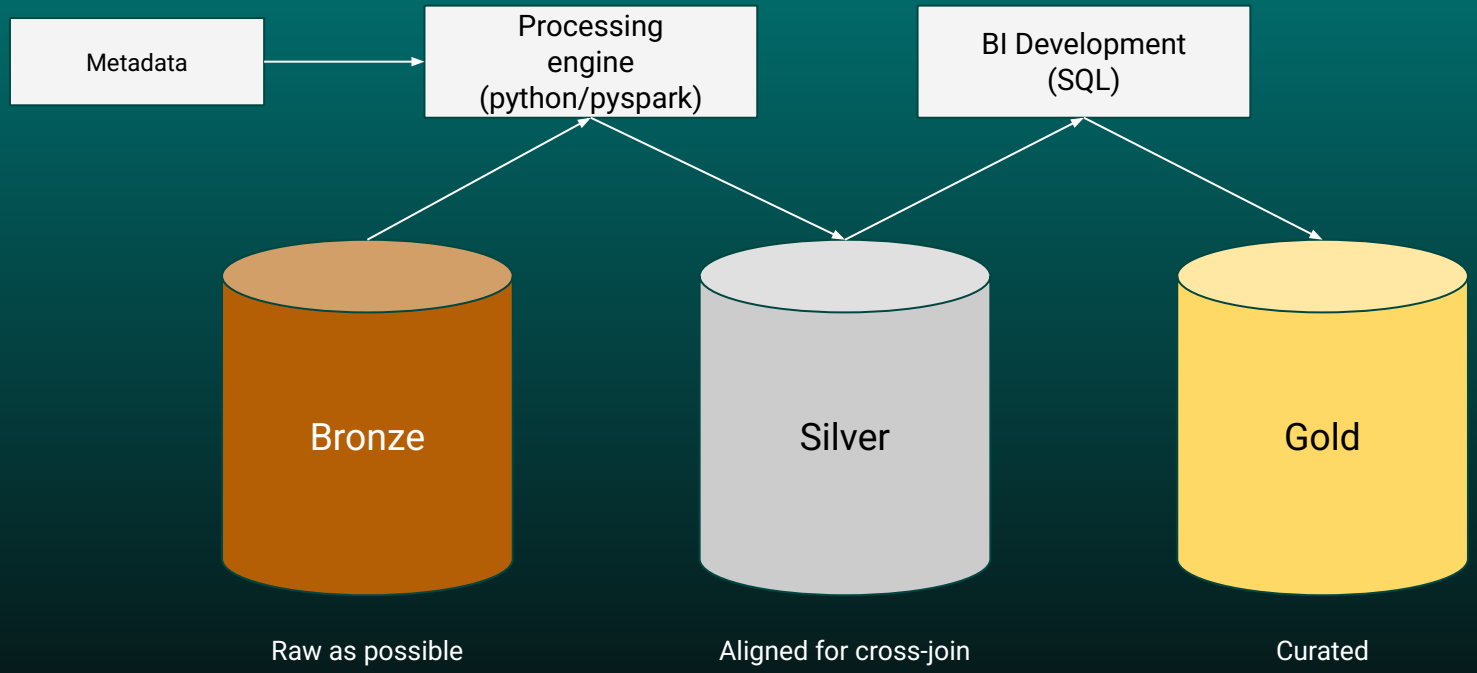
* Apache Arrow compatible

The metadata driven lakehouse

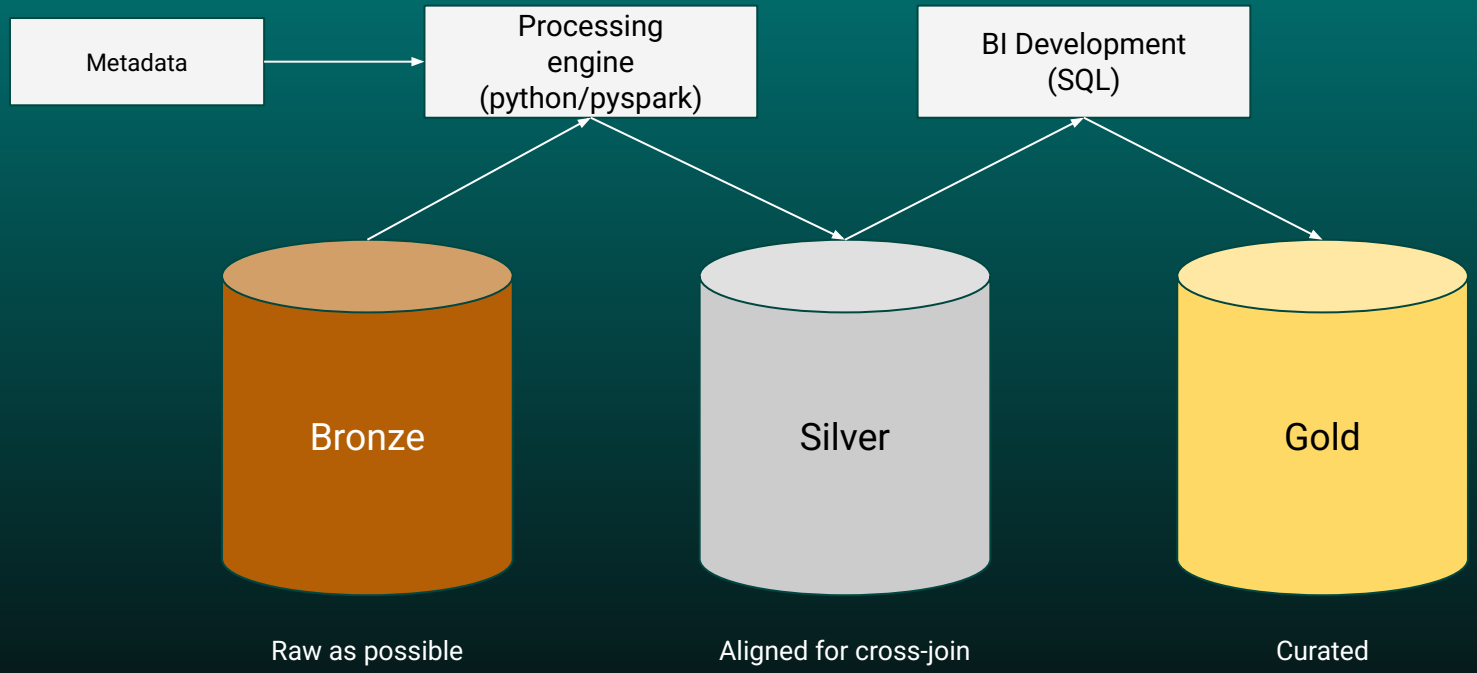
Metadata driven lakehouse



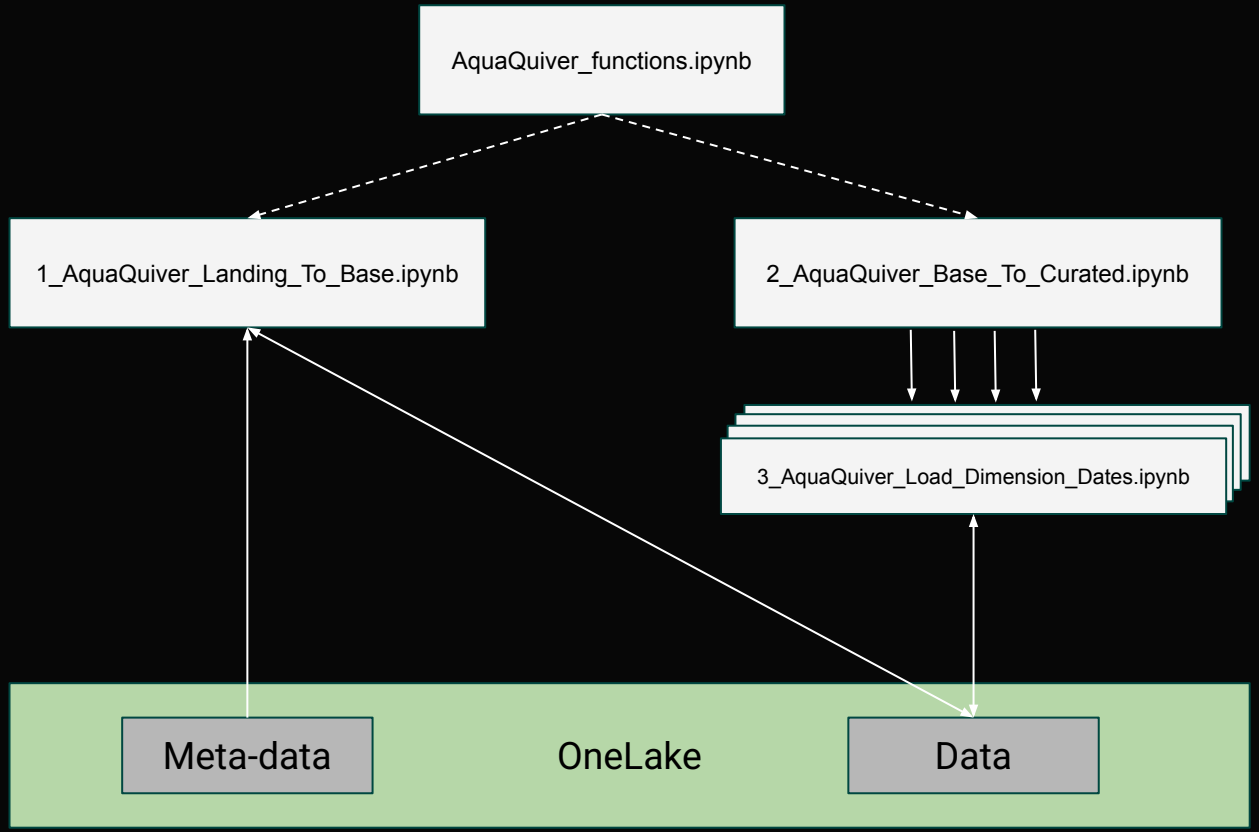
Metadata driven lakehouse



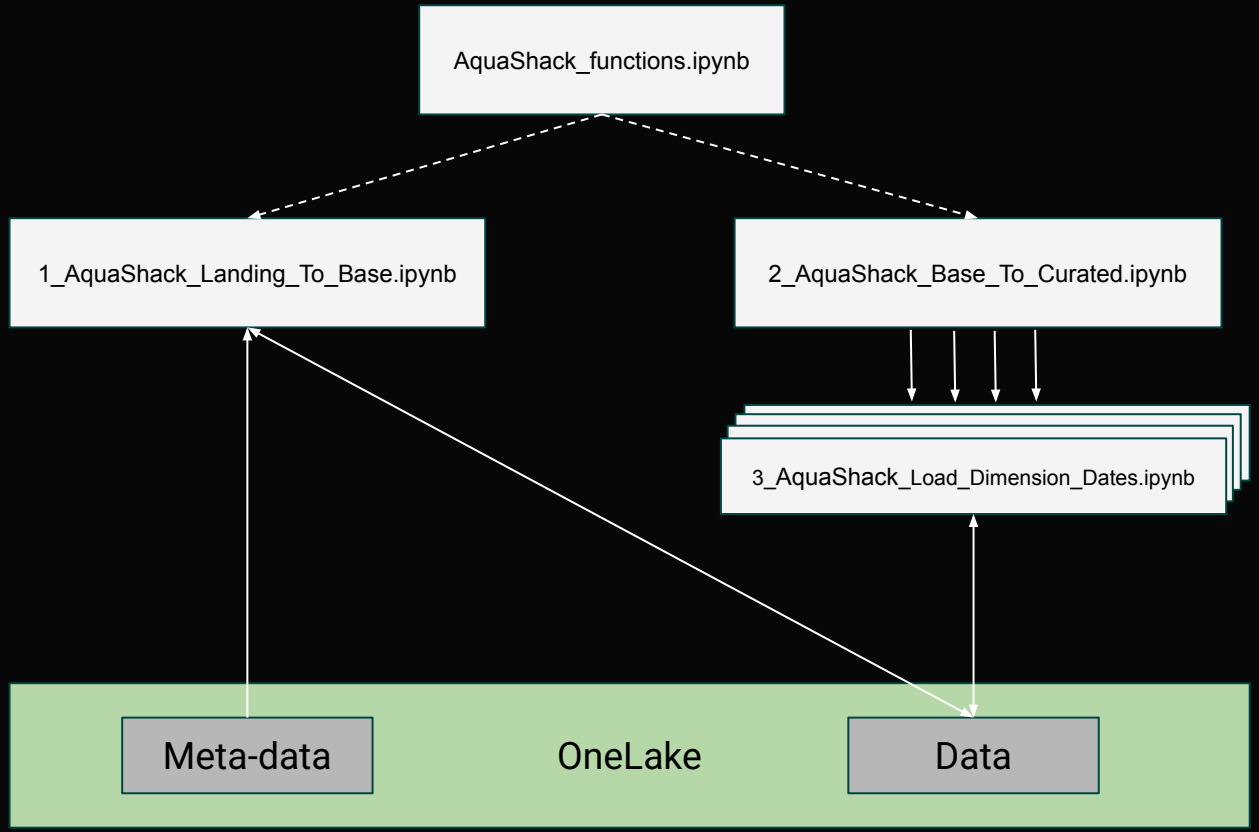
Metadata driven lakehouse



AquaQuiver


AquaQuiver_functions.ipynb
1_AquaQuiver_Landing_To_Base.ipynb
2_AquaQuiver_Base_To_Curated.ipynb
3_AquaQuiver_Load_Dimension_Dates.ipynb
Meta-data
OneLake
Data

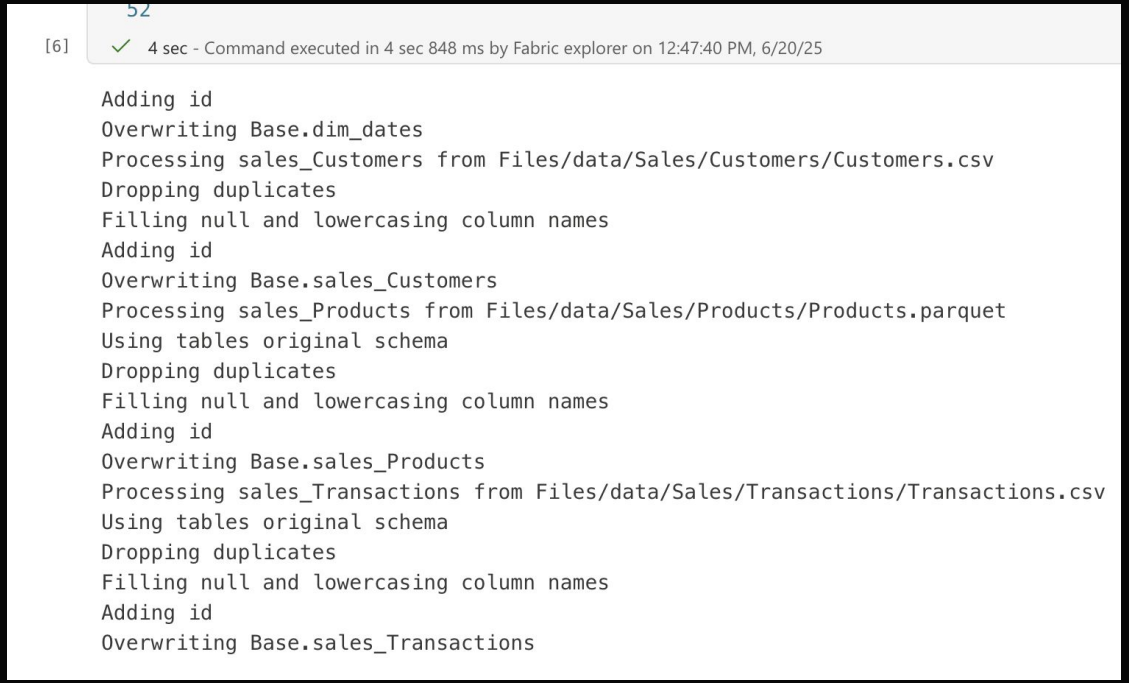
AquaShack


AquaShack_functions.ipynb
1_AquaShack_Landing_To_Base.ipynb
2_AquaShack_Base_To_Curated.ipynb
3_AquaShack_Load_Dimension_Dates.ipynb
Meta-data
OneLake
Data

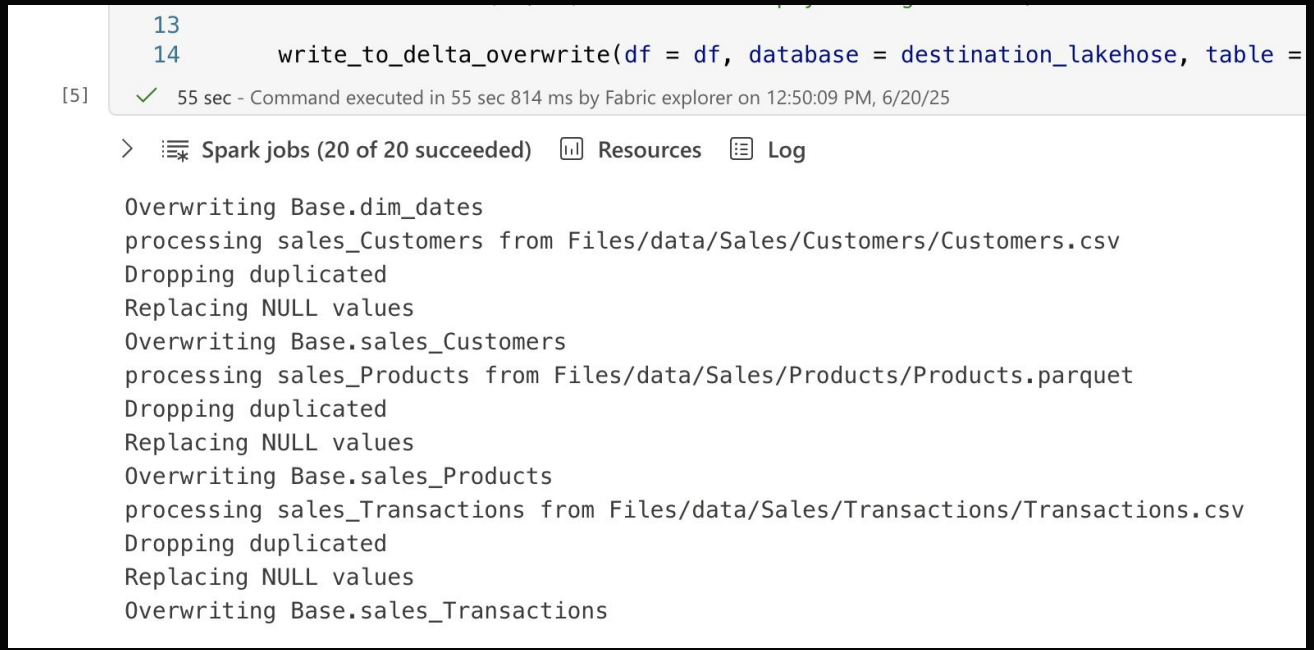


https://github.com/ChristianHenrikReich/AquaQuiver

AquaQuiver - Landing to Base



AquaShack - Landing to Base



AquaQuiver - Base to Curated



```
25  }
26
27  notebookutils.notebook.runMultiple(DAG, {"displayDAGViaGraphviz":True, "DAGLayout":"spectral", "DAGSize":11})
```

[2] ✓ 38 sec - Command executed in 38 sec 525 ms by Fabric explorer on 12:57:17 PM, 6/20/25

Log

Activity name	Snapshot	Status	Progress	Duration
3_AquaQuiver_Load_Dimension_...	3_AquaQuiver_Load_Dimension_Sales_Customers	Succeeded	100%	17 sec 562 ms
3_AquaQuiver_Load_Dimension_...	3_AquaQuiver_Load_Dimension_Sales_Products	Succeeded	100%	17 sec 567 ms
3_AquaQuiver_Load_Dimension_...	3_AquaQuiver_Load_Dimension_Dates	Succeeded	100%	17 sec 560 ms
4_AquaQuiver_Load_Fact_Sales_Tr...	4_AquaQuiver_Load_Fact_Sales_Transactions	Succeeded	100%	10 sec

```
{'3_AquaQuiver_Load_Dimension_Sales_Customers': {'exitVal': '',
 'exception': None},
```

AquaShack - Base to Curated

```
26
27   notebookutils.notebook.runMultiple(DAG, {"displayDAGViaGraphviz":True, "DAGLayout":"spectral", "DAGSize":11})
```

[1] ✓ 1 min 27 sec - Command executed in 1 min 20 sec 802 ms by Fabric explorer on 1:01:47 PM, 6/20/25

Log

Activity name	Snapshot	Status	Progress	Duration
3_AquaShack_Load_Dimension_S...	3_AquaShack_Load_Dimension_Sales_Customers	Succeeded	100%	52 sec 591 ms
3_AquaShack_Load_Dimension_S...	3_AquaShack_Load_Dimension_Sales_Products	Succeeded	100%	53 sec 596 ms
3_AquaShack_Load_Dimension_D...	3_AquaShack_Load_Dimension_Dates	Succeeded	100%	53 sec 599 ms
4_AquaShack_Load_Fact_Sales_Tr...	4_AquaShack_Load_Fact_Sales_Transactions	Succeeded	100%	17 sec 75 ms

{'3_AquaShack_Load_Dimension_Sales_Customers': {'exitVal': '',

AquaShack - Single node

```
13
14      write_to_delta_overwrite(df = df, database = destination_lakehose,
```

[5] ✓ - Command executed in 42 sec 551 ms by Fabric explorer on 8:49:17 AM, 6/21/25

```
Overwriting Base.dim_dates
processing sales_Customers from Files/data/Sales/Customers/Customers.csv
```

```
24      ]
25  }
26
27  notebookutils.notebook.runMultiple(DAG, {"displayDAGViaGraphviz":True, "DAGLayout":"spectral", "DAGSize":11})
```

[1] ✓ 1 min 15 sec - Command executed in 1 min 9 sec 896 ms by Fabric explorer on 10:00:33 AM, 6/21/25

Log

Activity name	Snapshot	Status	Progress	Duration
3_AquaShack_Load_Dimension_S...	3_AquaShack_Load_Dimension_Sales_Customers	Succeeded	100%	41 sec 266 ms
3_AquaShack_Load_Dimension_S...	3_AquaShack_Load_Dimension_Sales_Products	Succeeded	100%	40 sec 255 ms
3_AquaShack_Load_Dimension_D...	3_AquaShack_Load_Dimension_Dates	Succeeded	100%	41 sec 268 ms
4_AquaShack_Load_Fact_Sales_Tr...	4_AquaShack_Load_Fact_Sales_Transactions	Succeeded	100%	15 sec 985 ms

```
{'3_AquaShack_Load_Dimension_Sales_Customers': {'exitVal': '',
  'exception': None},
 '3_AquaShack_Load_Dimension_Sales_Products': {'exitVal': '',
```

Workload	Landing to Base	Base to Curated
AquaShack on clustered Spark	~56 sec	~1 min 20 sec
AquaShack on single node Spark	~42 sec	~1 min 9 sec
AquaQuiver	~4 sec	~38 sec

Should we move away from Apache Spark?


Bronze
Silver
Gold
Raw as possible
Aligned for cross-join
Curated



AquaQuiver



AquaShack



Blogs

Q & A



Empowering Lakehouse Solutions with Apache Arrow and Python Notebooks in Microsoft Fabric