DATA ENGINEERING WITH SNOWPARK

Mike Wies | SE

January 2023

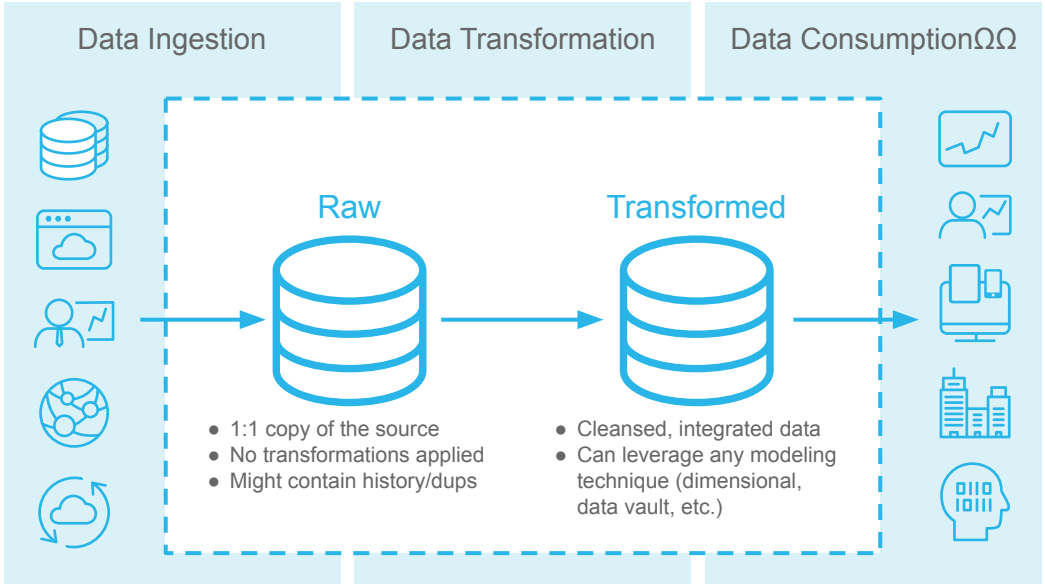
Agenda

- Data Engineering Overview
- Data Engineering with Snowflake
- Snowpark Python

DATA ENGINEERING OVERVIEW

Data Engineering Overview

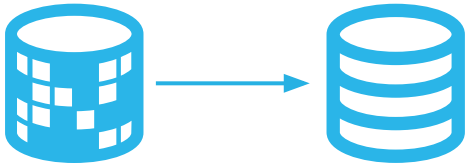
This diagram summarizes the high level phases in data engineering



Data Transformation Stages

All data goes through phases of processing, from its raw form to some final, cleansed, integrated form. What varies are the names and number of phases.

Simplest form



Raw

1:1 copy of source, no transformations

Transformed

Cleansed, integrated, modeled

"Medallion"

Bronze

Raw Ingestion

Silver

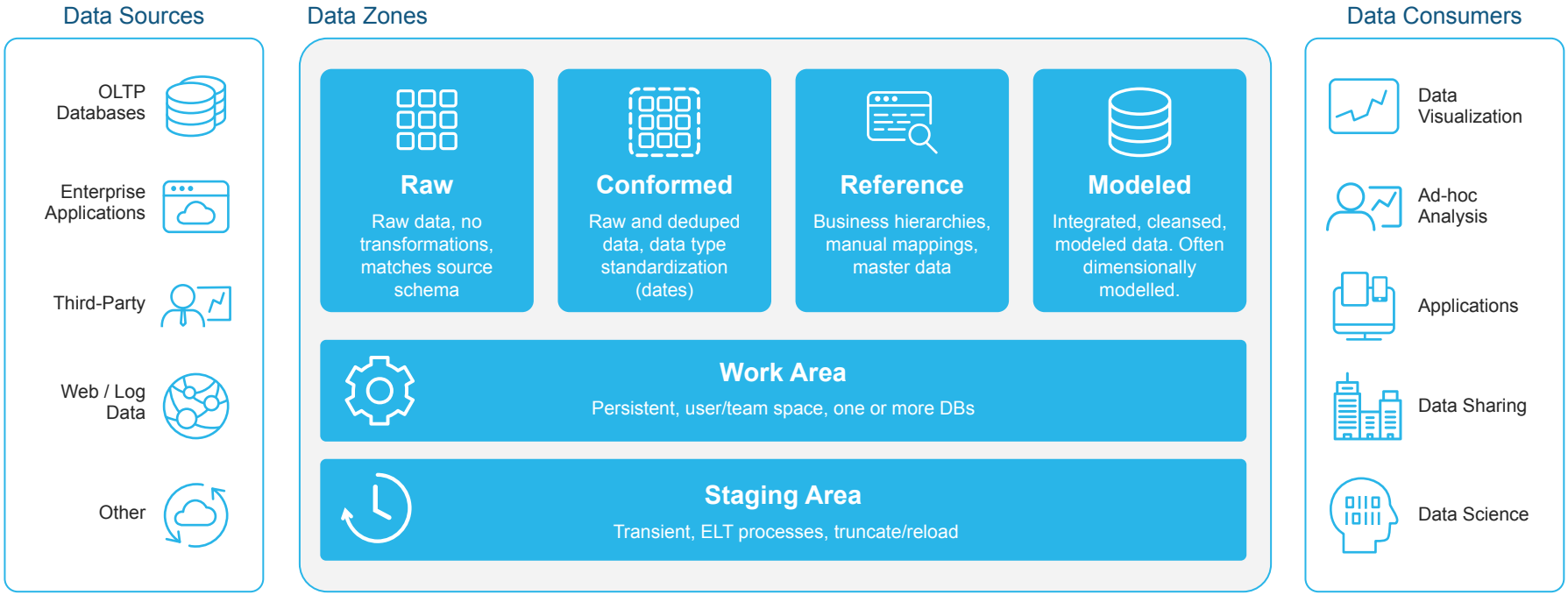
Filtered, Cleaned, Augmented

Gold

Business-level aggregates

Logical Data Zones

At a high level all enterprise data can be grouped into the following logical data zones (or groups)



DATA ENGINEERING WITH SNOWFLAKE

Data Engineering in the Data Cloud

Click on themes to go to deep dive

DATA INGESTION

- Open data formats: Iceberg
- Connectors
- Batch: COPY & Snowpipe
- Streaming: Snowpipe Streaming & Kafka Connector
- Schema Detection / Evolution

DATA TRANSFORMATION

- Snowpark
- SQL
- Streams & Tasks (Serverless Tasks)
- Stored Procedures
- Dynamic Tables (in PrPr)

OBSERVABILITY & PIPELINE EXPERIENCES NEW

- Ingestion & Tasks Dashboard
- Task Viewer
- Alert & Notifications
- Code in Snowsight Directly

GLOBAL DATA PIPELINES WITH REPLICATION NEW

- Ingestion & Tasks Replications

Data Engineering in the Data Cloud

Click on themes to go to deep dive

DATA INGESTION

- Open data formats: Iceberg
- Connectors
- Batch: COPY & Snowpipe
- Streaming: Snowpipe Streaming & Kafka Connector
- Schema Detection / Evolution

DATA TRANSFORMATION

- Snowpark
- SQL
- Streams & Tasks (Serverless Tasks)
- Stored Procedures
- Dynamic Tables

OBSERVABILITY & PIPELINE EXPERIENCES NEW

- Ingestion & Tasks Dashboard
- Task Viewer
- Alert & Notifications
- Code in Snowsight Directly

GLOBAL DATA PIPELINES WITH REPLICATION NEW

- Ingestion & Tasks Replications

Data Engineering in the Data Cloud

Click on themes to go to deep dive

DATA INGESTION

- Open data formats: Iceberg
- Connectors
- Batch: COPY & Snowpipe
- Streaming: Snowpipe Streaming & Kafka Connector
- Schema Detection / Evolution

DATA TRANSFORMATION

- Snowpark
- SQL
- Streams & Tasks (Serverless Tasks)
- Stored Procedures
- Dynamic Tables

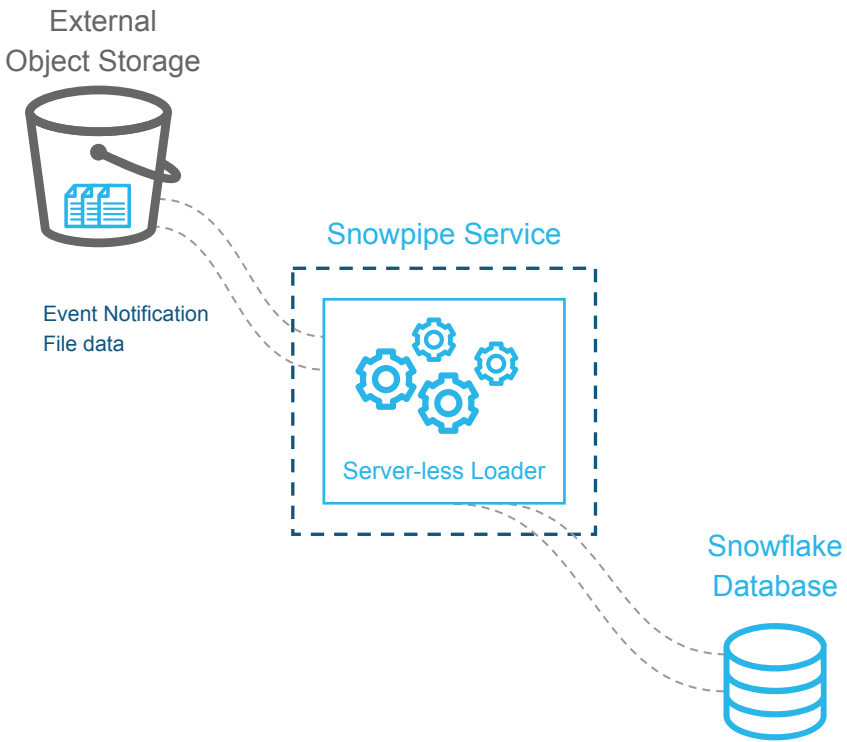
OBSERVABILITY & PIPELINE EXPERIENCES NEW

- Ingestion & Tasks Dashboard
- Task Viewer
- Alert & Notifications
- Code in Snowsight Directly

GLOBAL DATA PIPELINES WITH REPLICATION NEW

- Ingestion & Tasks Replications

Snowpipe



Account

Billing & Usage | Reader Accounts

[redacted] - Billing & Usage

Warehouses	Credits Used	Average Storage Used
8	29.47	47.927 GB

Warehouse Name	Credits Used
LOAD_WH	17.58
PLAYWH	10.20
BI_MEDIUM_WH	1.64
XSMALL	0.03
AUTOMATIC_CLUSTERING	0.01
SNOWPIPE	0.01
CLOUD_SERVICES_ONLY	0.00
DEMO_WH	0.00

Faster Auto-Ingestion With Snowpipe

Continuously generated data is available for analysis in seconds

Avoid repeated manual COPY commands

Full support for semi-structured data on load

50-68%

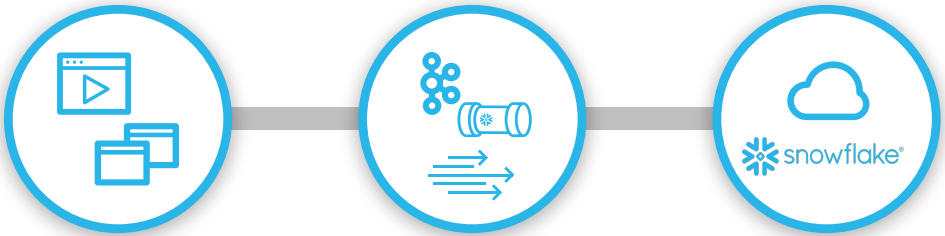
Improvement in Latency

You only pay for the compute time you use to load data

Zero management. No indexing, tuning, partitioning or vacuuming on load

Serverless: No servers to manage or concurrency to worry about.

Snowpipe Streaming



What is it?

Serverless auto-ingestion of streaming data

Value

Simplify architectures by ingesting streaming data directly into Snowflake, without complexity

How it Works

Standard ingestion framework that supports rowset ingestion. Leveraged by Snowpipe, Java client library, Kafka connector, and open to partner ecosystem for further development

In Dev Private Public GA

Dynamic Tables

Description

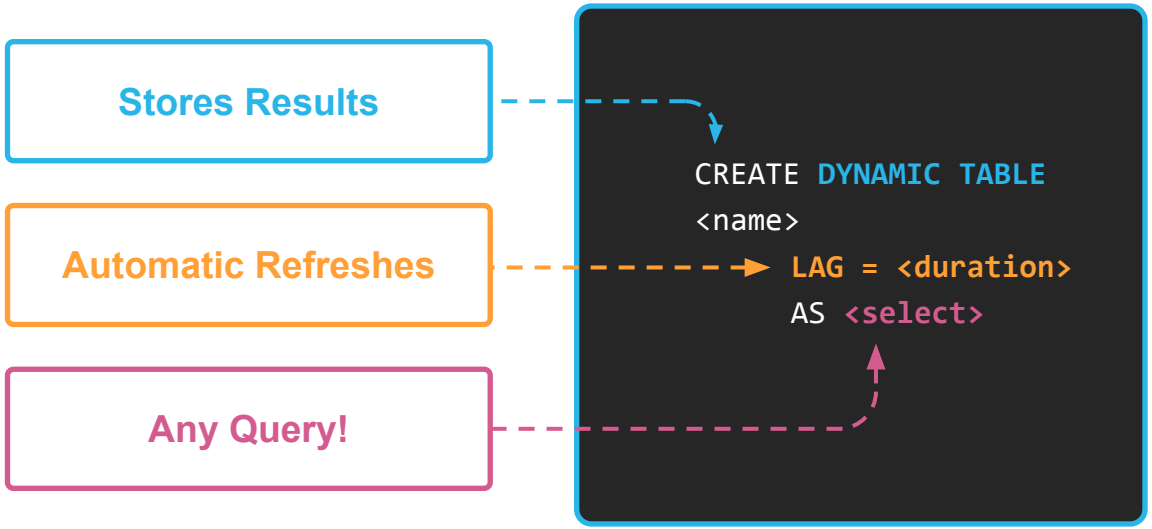
Declarative approach to transformations and simple data pipeline creation

Value

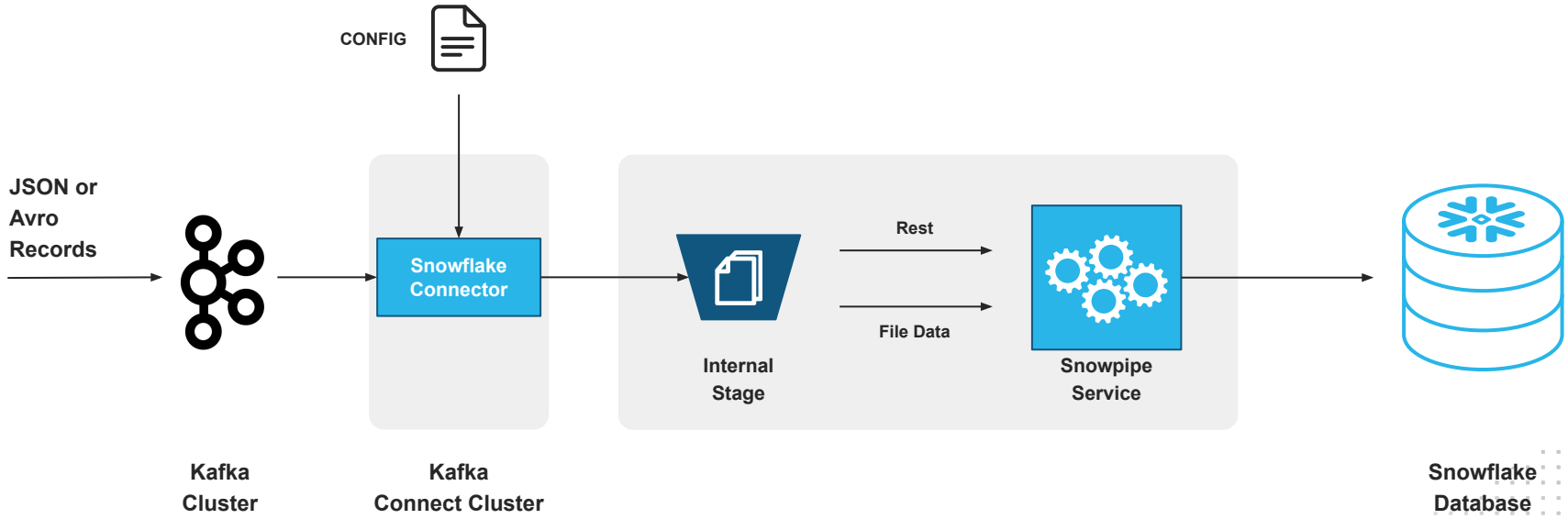
Automate incremental data refresh with low latency using easy-to-use declarative pipelines

Functionality

Join and aggregate across multiple source objects and incrementally update results as sources change



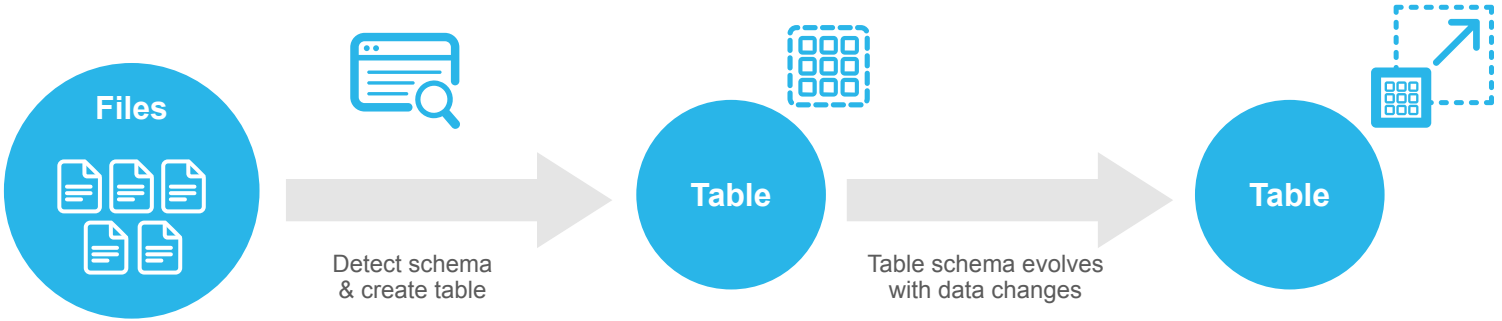
Kafka Connector



Schema Detection and Evolution

In Dev | Private | **Public** | **GA**

Feature	Stage
Schema Detection Read schema of PARQUET, AVRO, ORC files to generate column name and types. Auto creation of table objects from detected schema	GA
Schema Evolution Support for automatic addition of columns in schematized tables when source data changes. Supports nullability of required columns if source data changes	PuPr



Build Efficient Pipelines With Serverless Tasks

Before

```
create task t1
  schedule = '1 minute'
  warehouse = 'transform_wh'
  AS ...
```

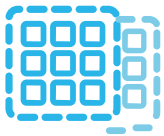
After

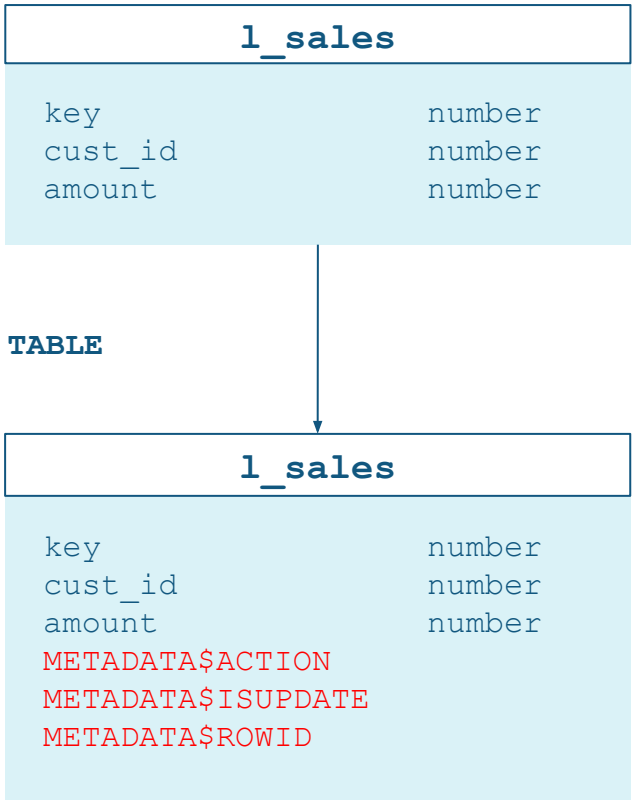
```
create task t1
  schedule = '1 minute'
  warehouse = 'transform_wh'
  AS ... z
```

- No need to specify a warehouse: size selection, packing, resume/suspend
- Simpler and more cost efficient
- Create new task as serverless or alter existing task

Streams

```
CREATE STREAM
s_sales_str ON TABLE
l_sales;
```





Adding a stream to a table appends three metadata columns that can be queried

These columns track the CDC records and their type: appends, deletes, or both (updates = inserts + deletes)

Little additional storage is required, as the stream is a logical pointer to the table's existing Time Travel micro-partitions

SNOWPARK PYTHON

WHAT IS SNOWPARK?

Libraries to securely develop in Python and other languages.
Data Pipelines, ML Models, apps…

Language of Choice on a Single Platform

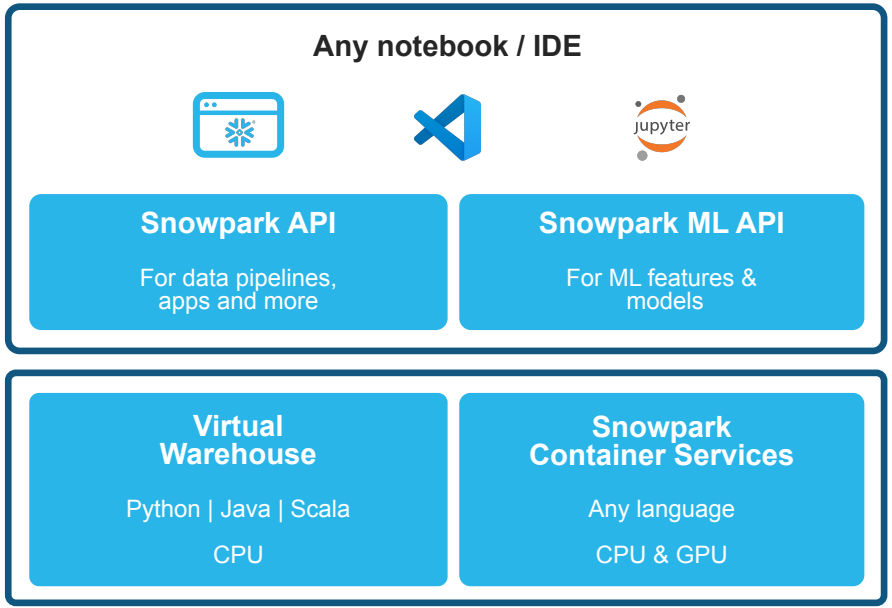
Scalability Without Operational Complexity

Governance and Security of Snowflake SQL

CHOICE OF RUNTIMES AND LIBRARIES WITH SNOWPARK

Code development & deployment
client-side libraries

Code execution
elastic compute runtimes



SNOWPARK CUSTOMER SUCCESS

Median of 3.5x faster performance and 34% cost savings over Spark-based systems*

ICE

80%

Time reduction for ad hoc queries

EDF

3-4x

Increase in product output with new models deployed in days

38%

of capacity customers are using Snowpark**

22M

Snowpark jobs executed daily**

*Based on 30+ customer production use cases and proof-of-concept exercises comparing the speed and cost for Snowpark versus managed Spark services between November 2022 and June 2023.
**Average as of October 31, 2023, figures exclude internal consumption

Snowpark for Python Features

Client API

DataFrame queries / transforms and submit UDFs / Stored Procs for execution.

UDFs

Execute custom Python code, **including OS packages**, in Snowflake secure Python sandbox.

Vectorized UDFs

Pandas dataframe batch processing of vectorized functions (e.g. model inference).

UDTFs: Table Functions

Non 1:1 transformations with custom partitioning guaranteeing contiguous batches.

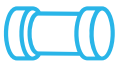
Stored Procs

Host and operationalize Python code and/or Snowpark API calls. Single node bounded.

Why Snowpark

Streamline Architecture

Collaborate on the same data in a single platform by natively supporting different user's programming language of choice



Build Scalable & Optimized Pipelines

Benefit from the Snowflake Data Cloud with superior price/performance and near-zero maintenance

Act With Confidence

Enforce consistent, enterprise-grade governance controls & security across all your workflows

Snowpark for Python

Familiar Programming Constructs

Use familiar syntax with DataFrame abstraction

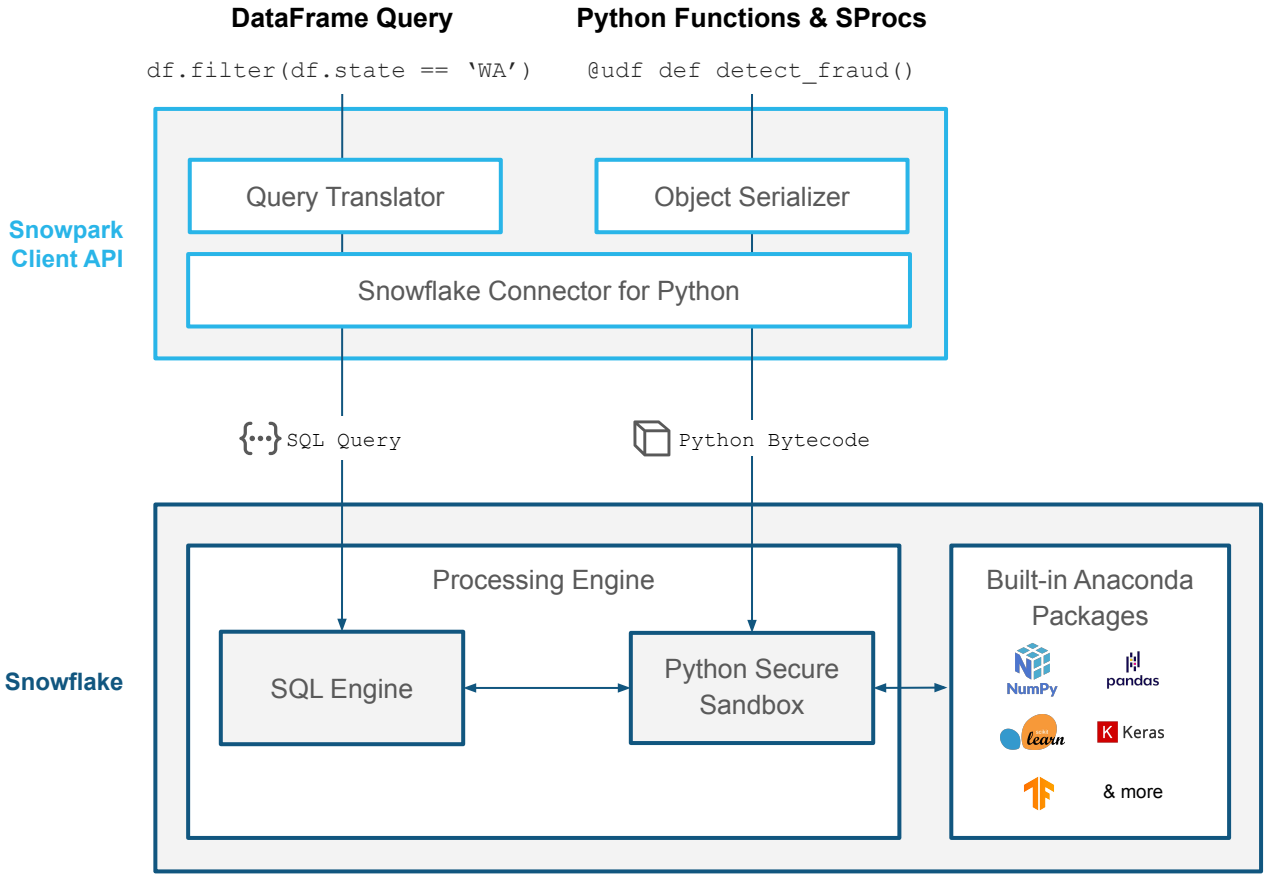
Rich Ecosystem

Easy access to hundreds of packages with automated dependency management

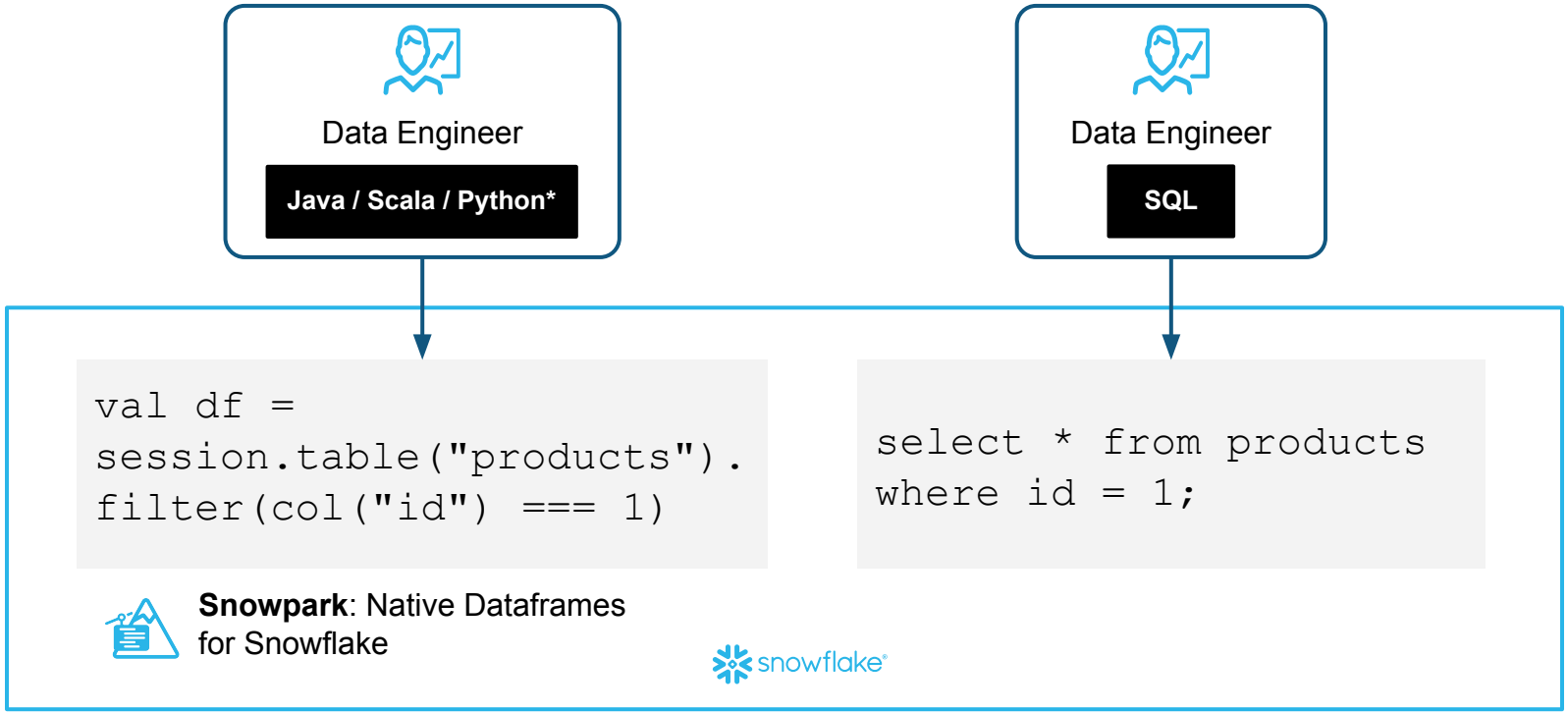
Secure Processing

Build with confidence in a highly secure, sandboxed environment

Snowpark for Python



How it Works



User-Defined Functions

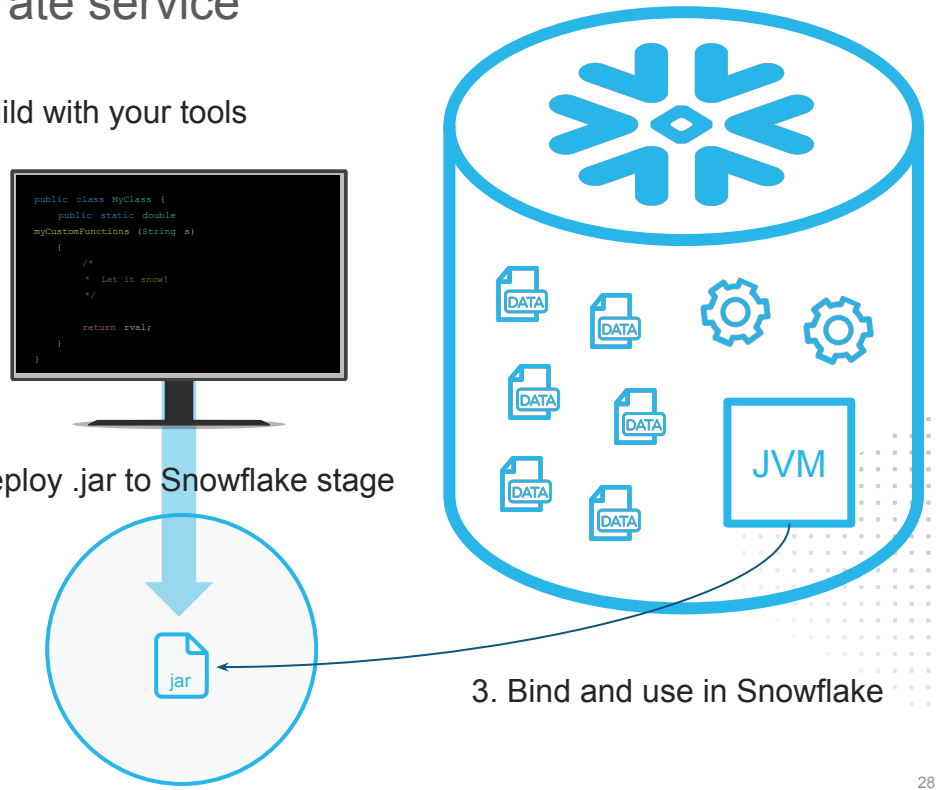
Transform and augment your data using custom logic running right next to your data, with no need to manage a separate service

Example Scenarios:

- ML Scoring
- Apply custom code
- Use third-party libraries

Benefits:

- Developers can build functionality into Snowflake using the popular languages and libraries
- Users can access this functionality as if it were built into Snowflake
- Administrators can rest easy: data never leaves Snowflake



In Dev Private Public GA

SOON

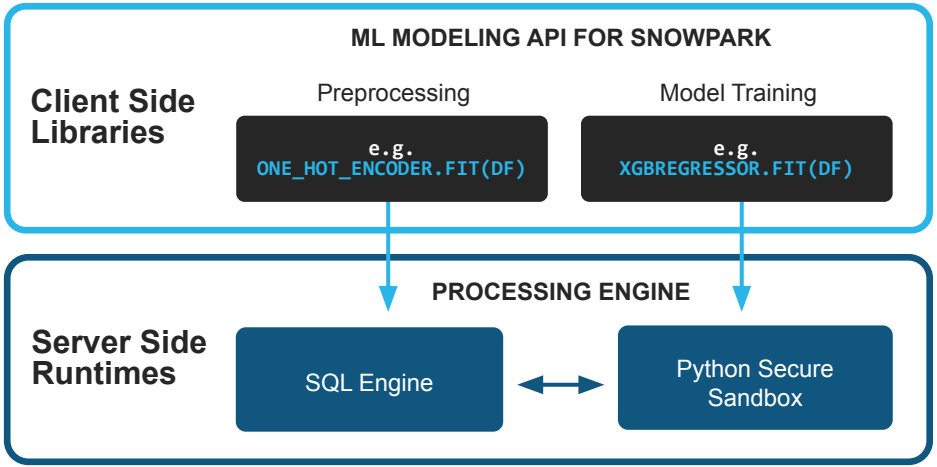
Snowpark ML Modeling API

WHAT IS IT

Popular frameworks for feature engineering and ML training directly in Snowpark.

WHY USE IT

- **Preprocessing:** Improve performance and scalability with distributed execution for common scikit-learn preprocessing functions.
- **Model Training:** Simplify model training for scikit-learn and xgboost models.



HOW TO USE IT

Use in Snowflake Notebooks (PrPr) or work from your tool of choice by installing the Snowpark ML library from the Snowflake Conda Channel or PyPI.

In Dev Private Public GA

SOON

Snowpark Model Registry

WHAT IS IT

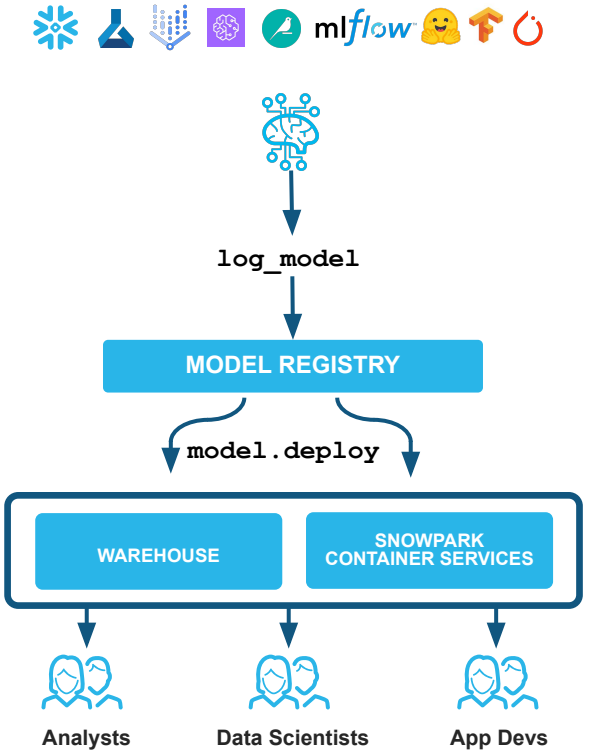
Integrated solution to manage and deploy models and their metadata natively in Snowflake.

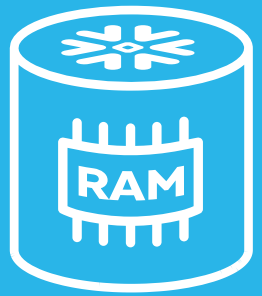
WHY USE IT

Scalable and secure deployment, management, and inference of ML models in Snowflake compute, including warehouses and Snowpark Container Services.

HOW TO USE IT

Use in Snowflake Notebooks (PrPr) or work from your tool of choice by installing the Snowpark ML library from the Snowflake Conda Channel or PyPI.





Snowpark-Optimized Warehouses

16X
memory

10X
cache

Effortless execution of memory-intensive operations

Bring training, in-memory analytics (e.g. correlations) or other memory-intensive operations inside Snowflake's secure Python/Java sandbox.

Accelerate subsequent run execution

Provide speedup when cached artifacts (Python packages, intermediate results, JARs, etc) are reused on subsequent runs

Plus all the benefits of standard virtual warehouses

In Dev | Private | Public* | GA

SOON

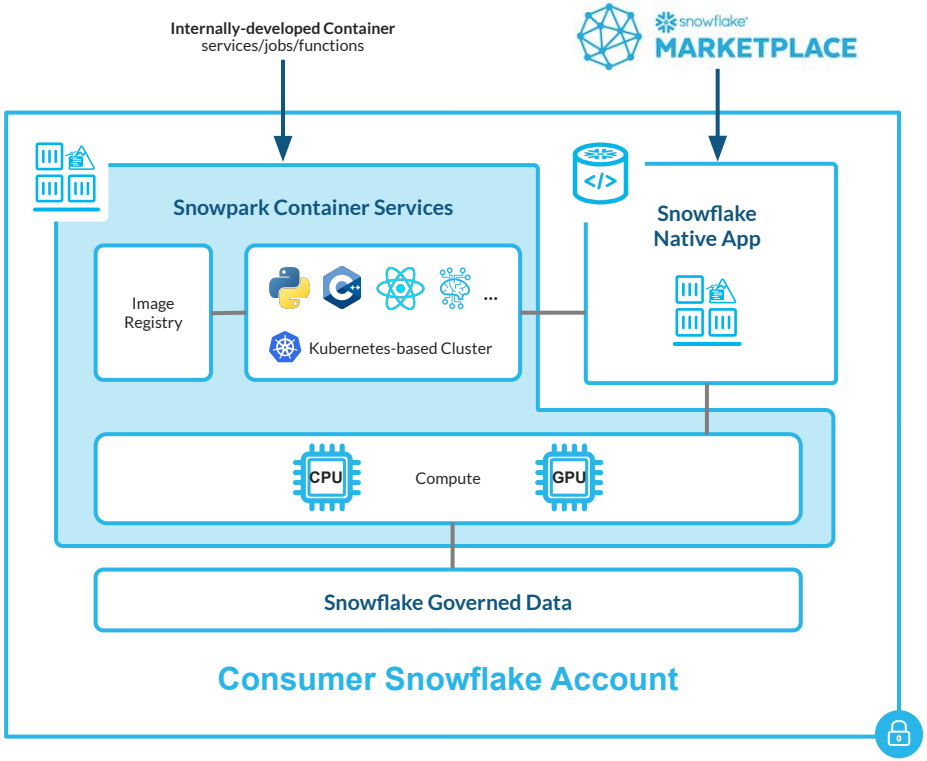
Snowpark Container Services

WHAT IS IT

Additional Snowpark runtime that helps developers register and deploy container images in Snowflake.

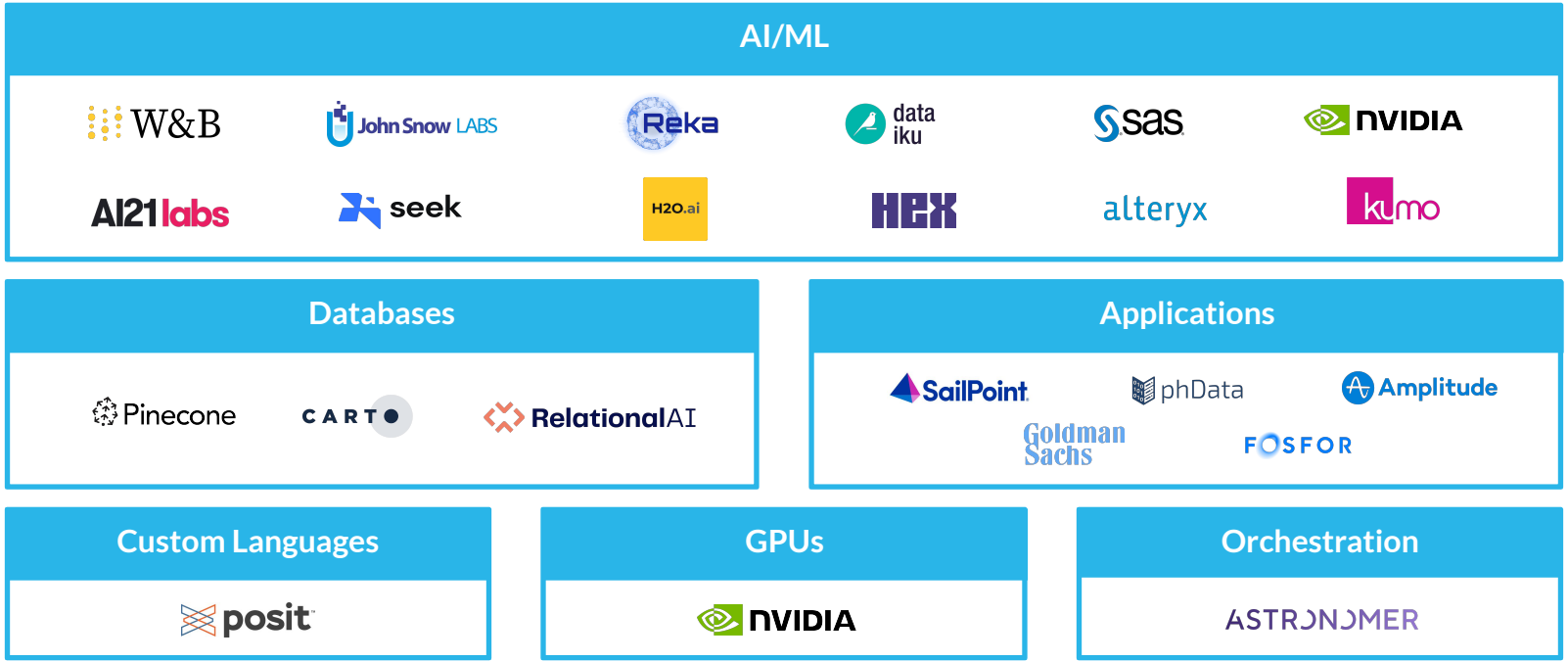
WHY USE IT

- **Language & hardware flexibility:** Build in any programming language, package as a container image and deploy in configurable CPUs & GPUs.
- **Unified services experience:** Effortlessly deploy with integrated image registry, elastic compute infrastructure and managed Kubernetes-based cluster.
- **Bring sophisticated apps to the data:** Run entire containerized applications from third-party developers in your account as Snowflake Native Apps (integration in Private Preview) via Snowflake Marketplace.



* for select AWS regions

Snowpark Container Services Partners



Benefits of Snowflake for Data Engineering

Features that will be highlighted during the demo

- ✓ Snowflake Tables
- ✓ Data ingestion with COPY
- ✓ Schema detection
- ✓ Data sharing/marketplace (instead of ELT)
- ✓ Streams for incremental processing (CDC)
- ✓ Streams on views
- ✓ Python UDFs (with third-party packages)
- ✓ Python Stored Procedures
- ✓ Snowpark DataFrame API
- ✓ Snowpark Python programmability
- ✓ Warehouse elasticity (dynamic scaling)
- ✓ Tasks (with Stream triggers)
- ✓ Task Observability

THANK YOU!