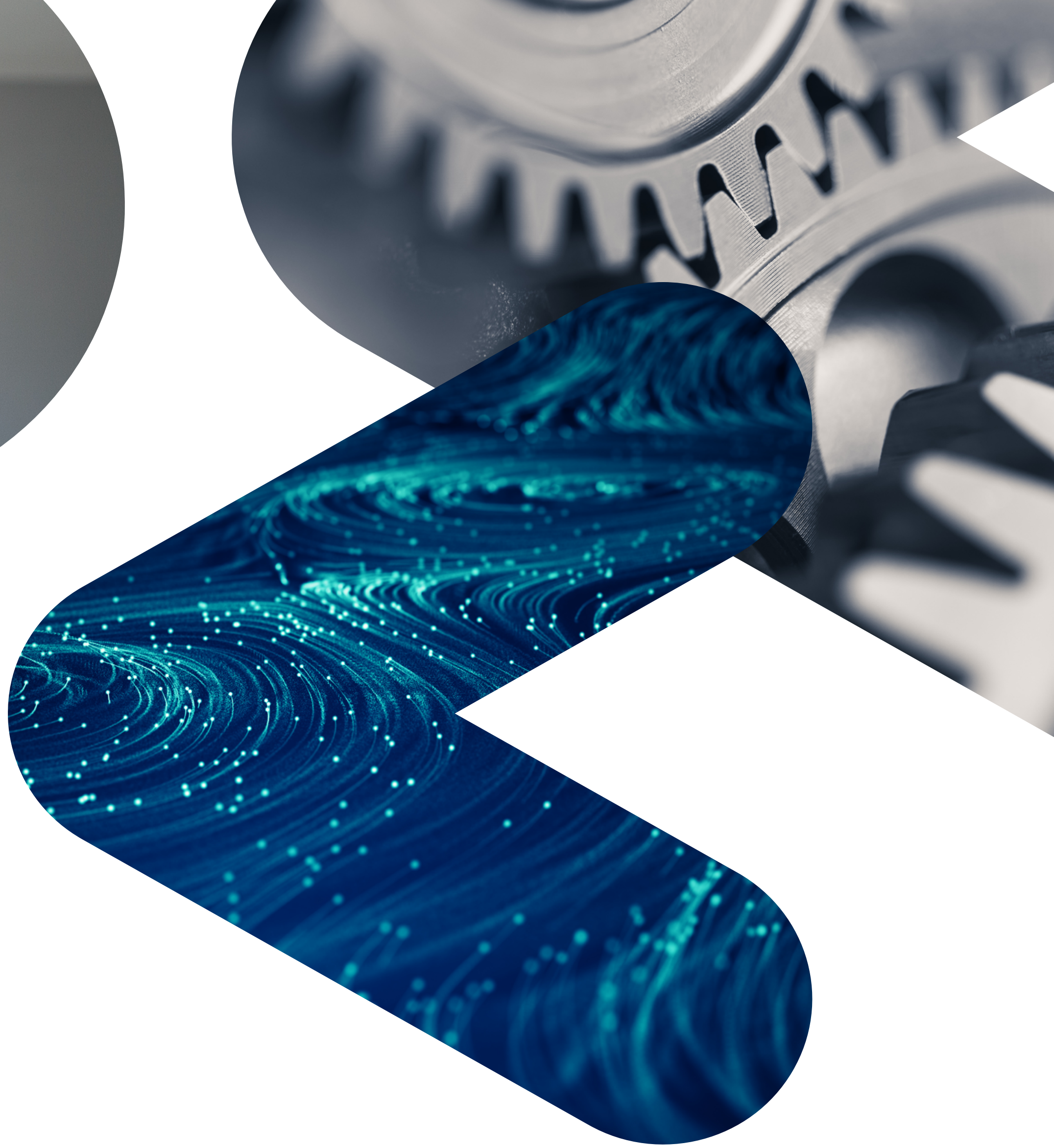


# 7 SNOWFLAKE REFERENCE ARCHITECTURES FOR APPLICATION BUILDERS

For every app type, there is a modern data architecture. Discover yours.

# TABLE OF CONTENTS

# WHY YOUR DATA PLATFORM MATTERS

It's safe to say data application builders will never worry about a lack of data. Data growth is running rampant at the edge, core and cloud, fed by IoT, mobility, transactions, social media and more. Although the ever-increasing amount of data presents immeasurable opportunities for delivering data-driven insights to customers, too many organizations are burdened by costs that arise from traditional data architectures' poor scalability and time-wasting operational overhead.

To assess whether their infrastructure is ready for the demands of today's data-driven business, every organization should ask three crucial questions:

1. Can our underlying architecture scale to meet the needs of our fast-growth business?
2. Can our product ingest and analyze large amounts of structured and semi-structured data together?
3. Can we accomplish these goals while remaining operationally efficient and cost-effective?

These three questions highlight the intrinsic need for a data stack architecture that has scalability, connectivity and support for all data types built into its design. That means selecting cloud-built infrastructure components, the most important of which is your data platform.

As the central hub for all things data, only a modern cloud data platform can deliver the high levels of performance and auto scaling needed to launch and scale applications quickly and cost-effectively. Here's what the Snowflake AI Data Cloud provides:

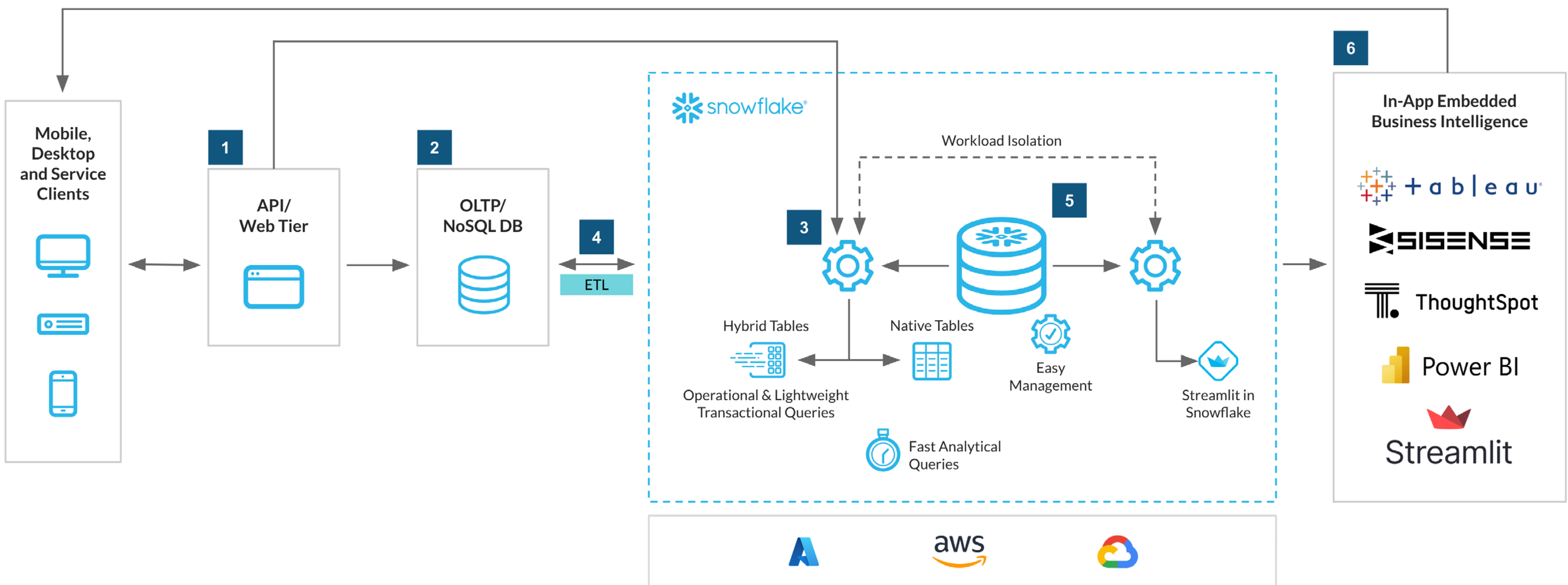
- **High performance and concurrency**
  Through a multi-cluster, shared data architecture, Snowflake spins up dedicated compute clusters that support a nearly unlimited number of concurrent workloads on shared tables. There's never contention for resources.
- **Scalability with true elasticity**
  Snowflake compute resources scale up and down automatically to deliver on-demand high performance that's cost-effective.
- **Connectivity for all programming languages**
  Snowflake has connectors for Python, JDBC, ODBC, Node.js, Go and PHP, which allow engineers to easily build applications in the languages of their choice. You can also use REST APIs to connect clients directly to Snowflake without a connector.
- **No site reliability engineering/DevOps burden**
  As a near-zero management platform, Snowflake automatically handles provisioning, availability, tuning, data protection and other operations, enabling you to focus on your own application development rather than infrastructure maintenance.

Snowflake also enables seamless connections to third-party platforms and APIs, easily fitting into your existing environment.

This ebook provides detailed reference architectures for seven use cases and design patterns. Each demonstrates the importance of a cloud-built data platform that can match scalability and connectivity expectations, both today and in the future.

# EMBEDDED ANALYTICS REFERENCE ARCHITECTURE

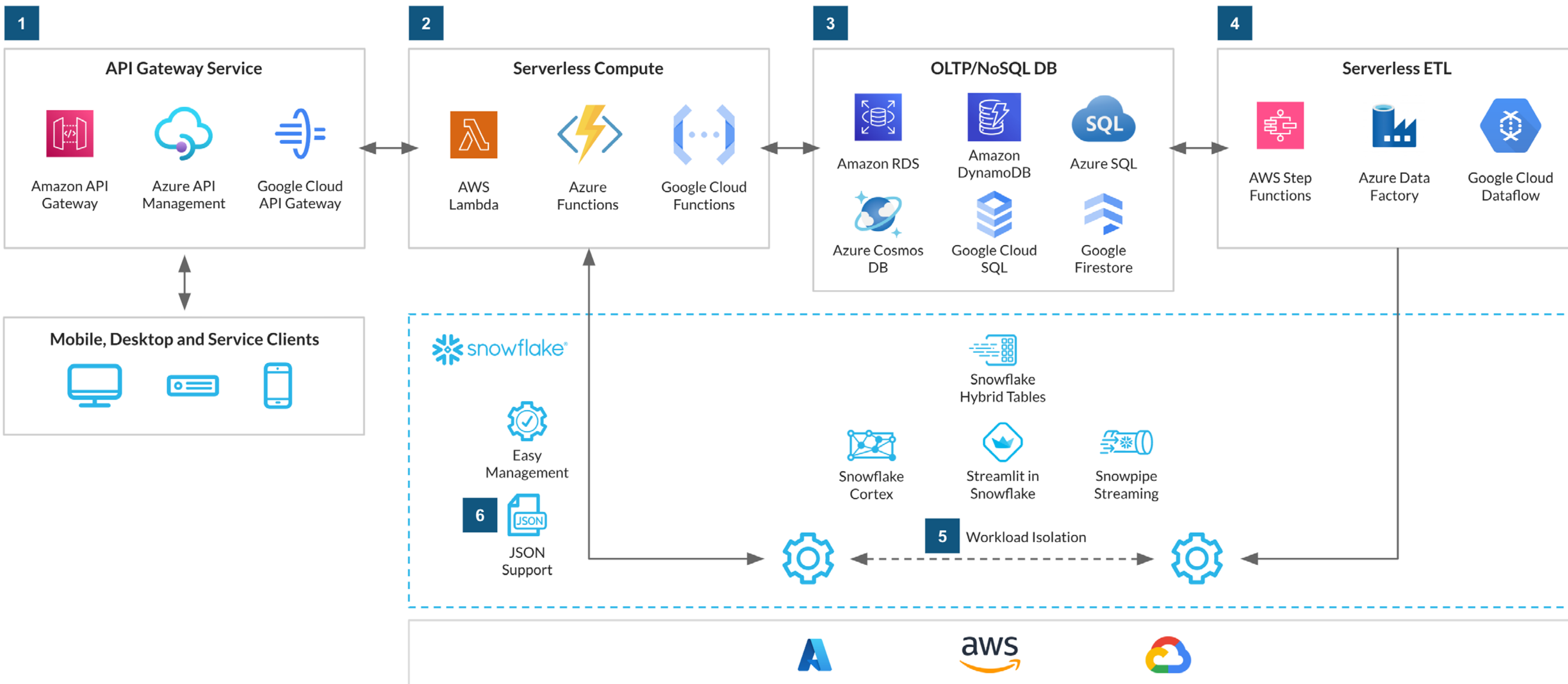
**Objective:** Build analytics-heavy apps that deliver in-app visualizations.




1. The application makes requests via an API or web tier, depending on whether API management is required to enforce an SLA.
2. An OLTP (SQL or NoSQL) database supports the transaction workloads of the app.
3. Snowflake Hybrid Tables natively support operational and lightweight transactional workloads within Snowflake, without the need for another external OLTP database.
4. Snowflake ingests historical transaction data via ETL infrastructure to support analytical workloads. Third-party partners like FiveTran, Matillion, dbt and HVR provide data integration and native connectivity to Snowflake.
5. Snowflake stores all historical data and supports queries by the application and by business intelligence (BI) tools. Virtual warehouses isolate workloads and autoscale compute resources to deliver high performance queries and unlimited concurrency.
6. Embedded BI tools or open-source charting libraries support analytics from within the application. Custom applications can also be built directly on Snowflake with Streamlit in Snowflake or embedded BI tools.

# SERVERLESS DATA STACK REFERENCE ARCHITECTURE

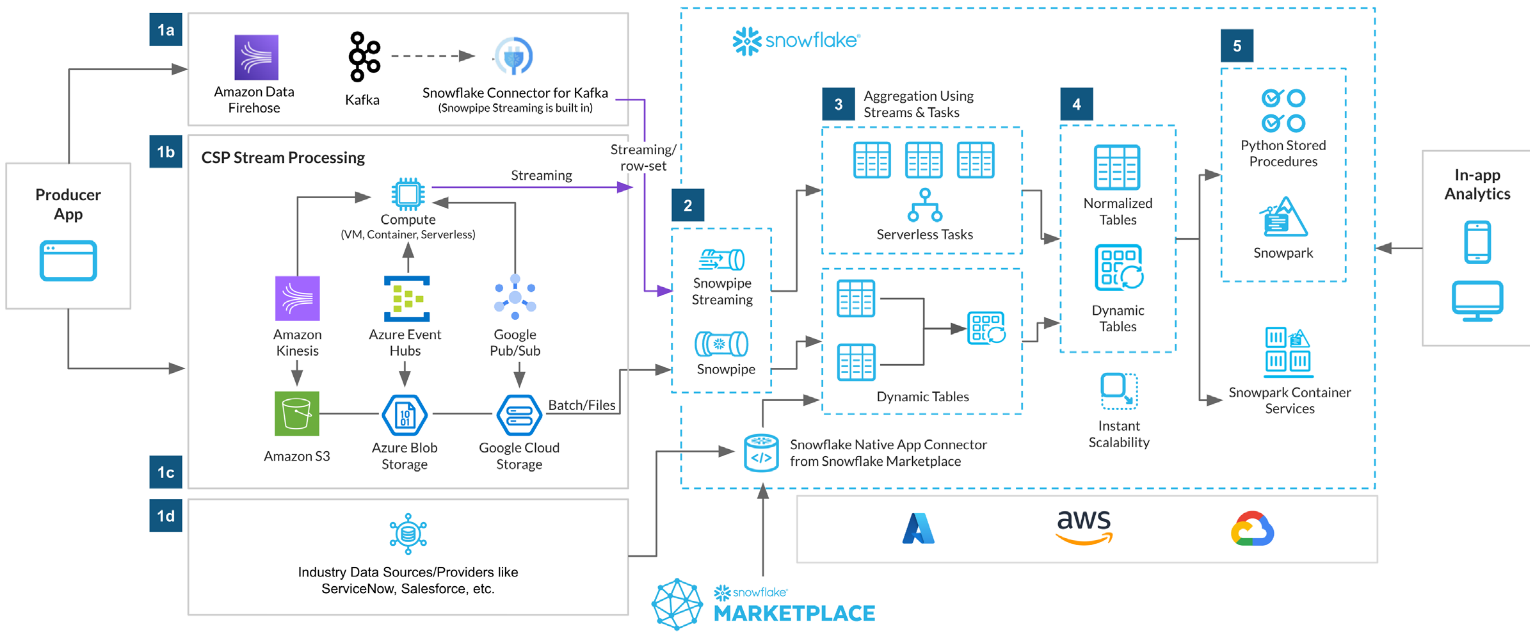
**Objective:** Build data-intensive apps that run on serverless infrastructures.




1. The client-side app, running on mobile or web devices, invokes the application logic on the serverless compute via an API gateway. The gateway protects the infrastructure with a firewall, and authenticates the API calls and proxies to appropriate serverless technology to handle the request.
2. Serverless compute runs the application logic and scales on demand, without the need to provision or manage servers. The application queries data in Snowflake for runtime decisions, such as delivering product recommendations or powering a dashboard for analysis.
3. An OLTP (SQL or NoSQL) database provides the application with high-capacity, simple transaction processing. This OLTP/NoSQL database can also be a serverless service. Snowflake Hybrid Tables is a native OLTP engine that supports operational and light-weight transactional processing within Snowflake itself.
4. A serverless ETL stack orchestrates the workflow and loads data into Snowflake.
5. Snowflake ingests data in batches or in streams and makes it available to the application for analytical queries. Snowflake scales automatically to keep pace with the data pipeline. Workloads are isolated in virtual warehouses where they can run and scale concurrently without resource contention.
6. Native JSON support enables easy ingestion and querying of flexible schema data alongside structured data.

# STREAMING DATA STACK REFERENCE ARCHITECTURE

**Objective:** Build data-intensive apps that run on serverless infrastructures.




1. The producer application generates continuous data that the streaming service ingests and buffers to account for data rate differences between the producer and consumers.

1a. When Kafka is used, the Snowflake Connector for Kafka is deployed to read all data from Kafka and ingest directly to a Snowflake table using either Snowpipe or Snowpipe Streaming. This same design applies to those using Managed Kafka from AWS with Managed Streaming for Apache Kafka and Azure with Event Hubs when using the Kafka compatibility endpoints.

1b. When data latency is critical, you can pair Cloud Service Provider (CSP) streaming technologies with custom code using the Java SDK for Snowpipe Streaming to get data queryable in seconds.

1c. Data streaming can also be batched and written to a blob store using CSP services, which Snowflake can auto ingest via Snowpipe. In this case, the raw data can be retained in blob storage for other processes. Snowpipe with Auto-Ingest automates the data ingestion from object storage.

1d. A Snowflake Native Connector for a third-party data source can be built as a Snowflake Native App and distributed via Snowflake Marketplace. Consumers can deploy the app and get the third-party data directly in their Snowflake account in just a few easy steps.

2. Snowflake ingests data from the streaming service into a table and stores the streamed data for analysis.

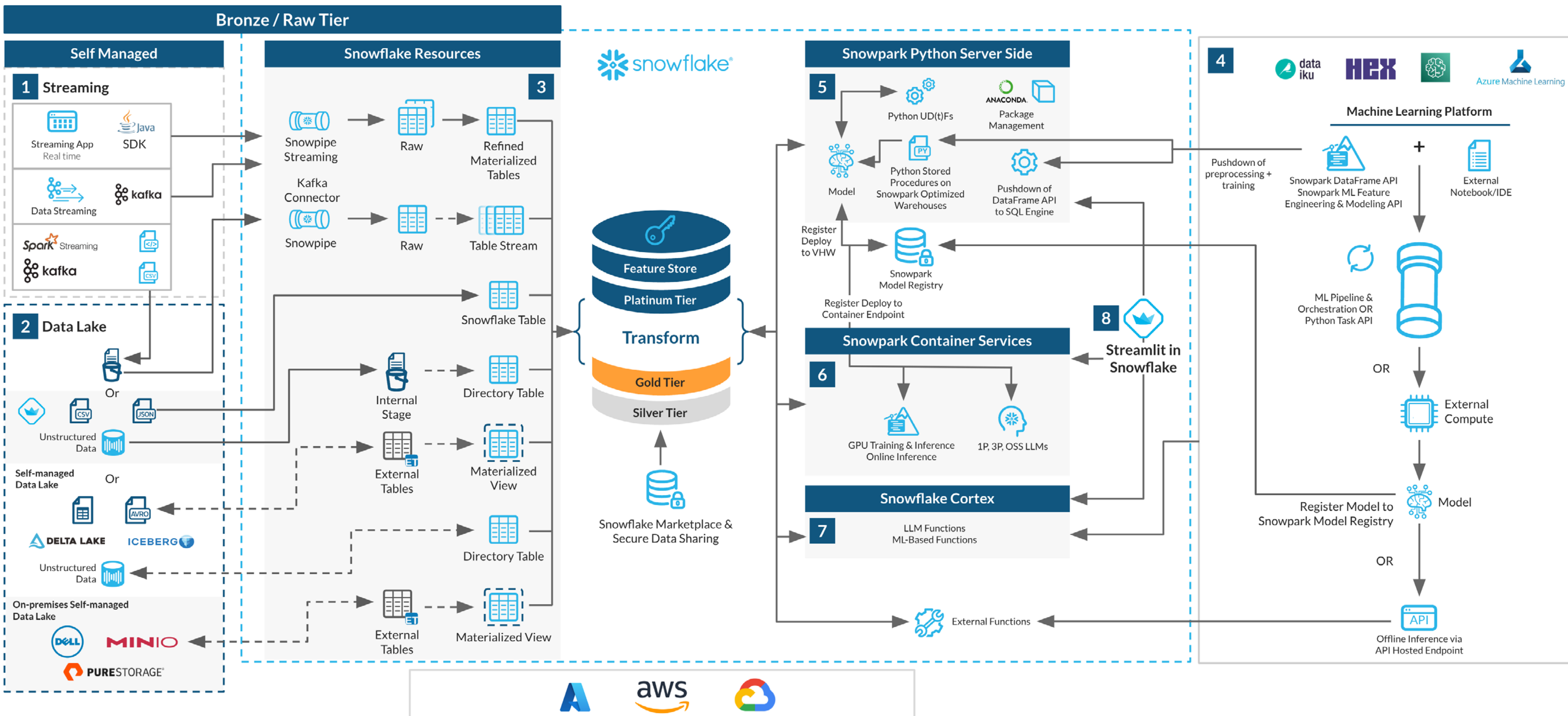
3. Streams and Tasks features detect data changes and schedule tasks to perform any required transformations. Multiple Streams and Tasks can be chained to implement a complex data pipeline.

4. Dynamic Tables can be used to simplify transformation. You can use SQL or Python for declarative stream processing with automated orchestration.

5. Data can be further transformed using Python stored procedures or Snowpark, or used in Snowpark Container Services for downstream consumption.

# MACHINE LEARNING AND DATA SCIENCE REFERENCE ARCHITECTURE

**Objective:** Train machine learning (ML) models for predictive apps.

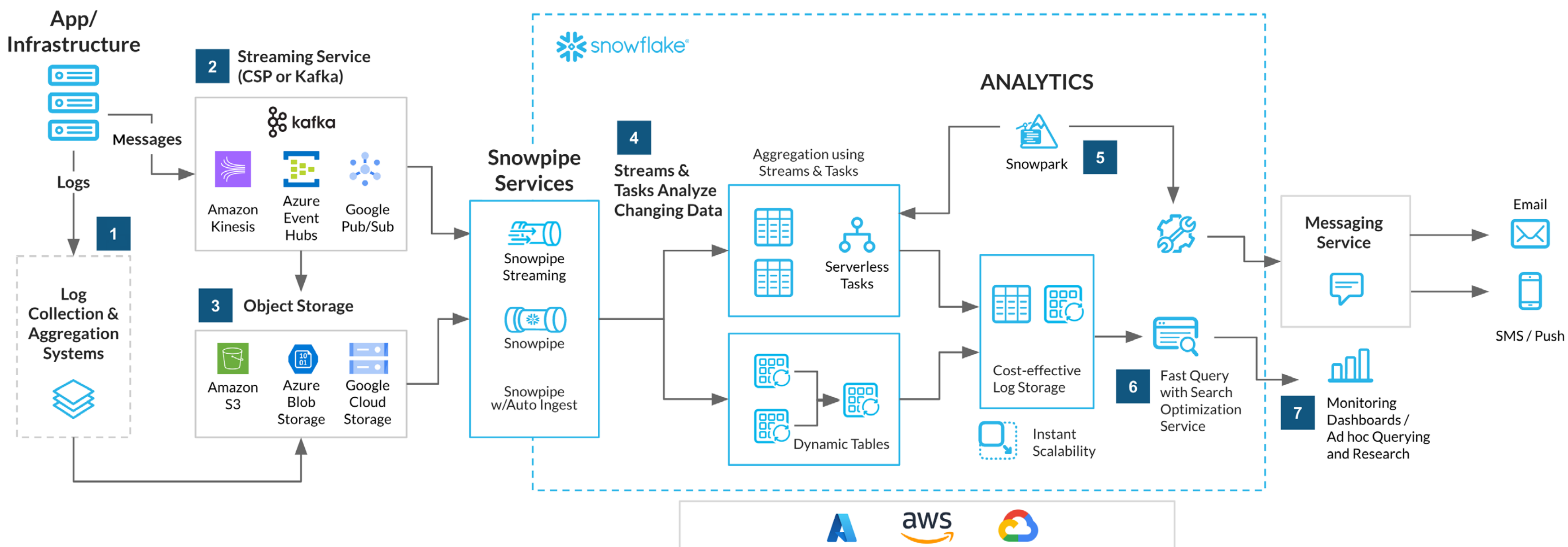


1. The app produces training data, which is sent into streaming (1). Streaming services can then batch that data to be stored in the data lake or send it directly to Snowflake for storage and querying. The streaming service buffers the training data for reliable and continuous ingestion.
2. Depending on the data lake structure and data format, data can be ingested into Snowflake via Snowpipe, queried directly as an external table in Snowflake or queried as a directory table. Snowflake support for Iceberg Tables addresses use cases that require read and write operations while storing Parquet and Iceberg metadata files in customer-supplied storage.
3. Once data is available in Snowflake for query (3), Tasks can schedule Stored Procedures to perform required transformations. Multiple Streams and Tasks can be chained to implement a complex data pipeline. External Tables support querying of data in object storage without ingestion. Data scientists can process training data directly inside Snowflake with flexible, instant scalability to support feature engineering and experimentation using Snowpark. Snowpark provides an API for querying and processing data in a data pipeline, and trained ML models can be run in Snowflake using Python, Scala and Java UDFs (4) or externally with another ML platform leveraging External Functions.
4. Using the data stored in Snowflake, data scientists train models with ML platforms. Once trained, the model artifacts are deployed on the training platforms to support predictions. The Snowpark ML Modeling API is also available and enables the use of popular Python ML frameworks, such as scikit-learn and XGBoost, for feature engineering and model training without the need to move data out of Snowflake.
5. Training can also be done directly in Snowflake leveraging Snowpark Python server side (5) and Snowpark-optimized warehouses. The application performs predictions using these models deployed in Snowpark Container Services in real time, or it schedules batch predictions using the deployed models. For batch predictions, data is read from an input table in Snowflake, and the results are stored in an output table where they are available to the app. Real-time predictions can be run from the client application by calling the UDF or External Functions hosting the model.

   Note: The Snowflake Model Registry allows customers to securely manage models and their metadata in Snowflake, regardless of origin. The Snowflake Model Registry stores ML models as first-class schema-level objects in Snowflake.
6. Developers can utilize Snowpark Container Services to effortlessly deploy, manage and scale containerized workloads (jobs, services, service functions) using secure Snowflake-managed infrastructure with configurable hardware options, such as GPUs.
7. With just a single line of SQL or Python, analysts can instantly access specialized ML and LLM models tuned for specific tasks. They can also leverage more general purpose models for prompt engineering and in-context learning. Since these are fully hosted and managed by Snowflake Cortex, users always have access to them without the need to bring up and manage expensive GPU infrastructure.
8. Streamlit in Snowflake enables data scientists to build applications in Python that analyze and visualize ML data stored in Snowflake.

# APPLICATION HEALTH AND SECURITY ANALYTICS
# REFERENCE ARCHITECTURE

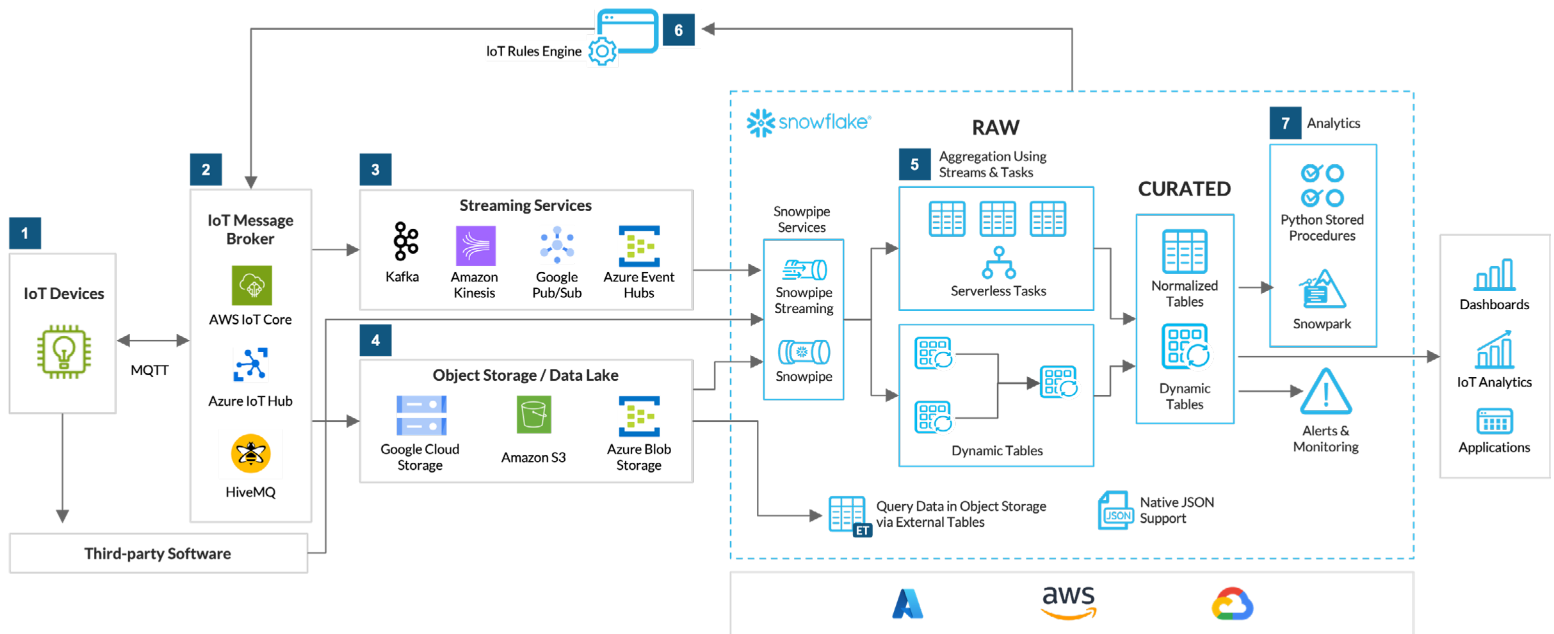
**Objective:** Analyze large volumes of log data to identify security threats and monitor application health.




1. The application and its infrastructure log large volumes of event data that can be used to monitor application health and detect malicious behavior. Log collection and aggregation systems centralize log data from multiple sources and deliver it to a streaming service (2) or to object storage (3).
2. The streaming service buffers log data to ensure reliable and continuous ingestion.
3. Depending on which log collector and aggregation system is used, data can be staged in object storage without the need for a streaming service.
4. Snowflake stores and analyzes the log data, which can be saved for long periods at commodity storage prices. Snowpipe with Auto Ingest automates the ingestion from object storage. Scheduled tasks, written in SQL, Python, Scala, Javascript or Java, detect suspicious behavior or application health concerns from changing data in Streams.
5. Snowpark allows developers to bring in external libraries from internal developers, third parties or open source and helps run code on Snowflake compute without depending on shipping data to an external environment.
6. Using features like Snowflake’s search optimization service, users and applications can perform fast point lookups, which are used in interactive dashboards and apps trying to conduct “needle in a haystack”-style queries.
7. Tasks can detect suspicious activity or health concerns and store and/or alert by using external messaging services via external functions. Operations teams can also monitor the application via dashboards or ad hoc queries.

# IOT
# REFERENCE ARCHITECTURE

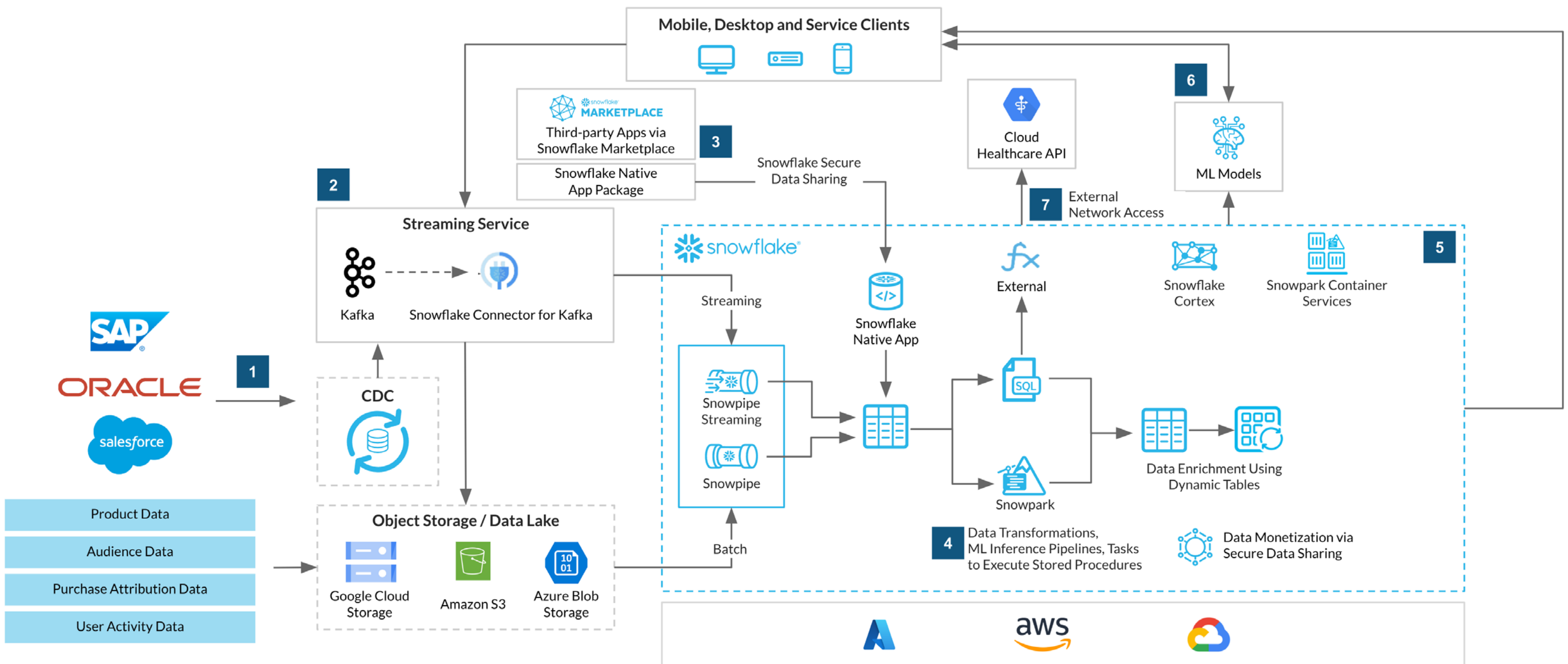
**Objective:** Build apps that analyze large volumes of time-series data from IoT devices and respond in real time.




1. Smart devices, sensors and other IoT devices generate continuous data. Third-party software like HighByte, DXC and so on is deployed at the edge to merge real time, transactional and time-series data into a single payload for consuming apps and to help users collect data from various sources, add context to it, and transform it into a format that other systems can understand.
2. Due to frequently unreliable Internet connectivity, IoT devices communicate using the MQTT protocol and an IoT message broker. The message broker uses a publish and subscribe mechanism to interact with other services, which subscribe to specific topics within the broker to access device data.
3. A streaming service ingests and buffers real-time device data, thus ensuring reliable ingestion and delivery to a staging table in Snowflake (5).
4. In cases where the application requires it, object storage is used to stage batch data prior to ingestion. For example, minute-by-minute data may be stored in object storage, whereas aggregated data over a longer period may be ingested to Snowflake (5).
5. Snowflake offers native support for JSON and other semi-structured data formats for easy ingestion of device data. Snowpipe automatically optimizes Time Series queries by ingesting data chronologically. Snowflake Dynamic Tables automate the workflows required to ingest and aggregate incoming data.
6. An IoT rules engine hosts the business logic required by the application and operates on data available in Snowflake and in the message broker. The rules engine sends messages back to control devices.
7. Snowpark allows developers to bring in external libraries from internal developers, third parties or open source and run code on Snowflake compute without having to move data to an external environment.

# CUSTOMER 360
# REFERENCE ARCHITECTURE

**Objective:** Build sales and marketing apps that use historical and real-time data to accomplish "360-degree view" customer goals.




1. Data lake stages application data—such as data about products, audiences, purchase attributions and user activity—for ingestion.
2. A streaming service ensures reliable and continuous ingestion by buffering event data, such as clickstreams.
3. Snowflake Native Apps and Secure Data Sharing enable data from third-party sources to be used without copying or moving the data.
4. Snowpark helps orchestrate the workflow to transform the loaded data from object storage into Snowflake. Partner tools like dbt help write and execute the data transformation jobs, compile it to SQL and run against your database.
5. Snowflake supports all the analytics workloads within the application. External Tables support queries of data in object storage without ingestion. Streams and Tasks features automate the ingestion and data enrichment process. Native support for JSON and other semi-structured formats simplifies the ingestion of event data. Secure Data Sharing enables monetization of fresh data without copying or moving the data.
6. ML models are trained to optimize offers based on historical data stored in Snowflake. The application makes real-time predictions via an API and uses Snowflake tables to store input data and batch prediction results.
7. Snowflake external access provides flexibility to reach public internet endpoints like Cloud Healthcare API from Snowpark without any additional infrastructure setup. This means Snowpark UDFs/UDTFs or vectorized UDFs and stored procedures can call external network locations based on the egress network rules containing a list of trusted IP or host URLs determined by the customer.

# BUILD YOUR APPLICATIONS ON A MODERN CLOUD DATA PLATFORM

Regardless of the type of applications you build or what architectural design pattern you select, you must meet the core data platform requirements for scalability and connectivity if you want to attract and keep customers. With Snowflake, you can meet customer expectations with a modern foundation for your data stack that delivers a highly performant service, both now and in the future.

Rather than spend valuable development time rearchitecting your data stack over and over again to chase ever-evolving scalability needs, a cloud data platform lets you focus on what you do best: building and improving your application to entice new customers.

## NEXT STEPS

Ready to build? Still have some questions? Check out these resources:

- Register for Powered by Snowflake and get access to resources and go-to-market benefits to help you build, market, and operate applications in the Data Cloud.
- Try our Quickstart: Getting Started with Snowflake ML
- Download The Modern Data Streaming Pipeline: Get top analytical streaming reference architectures and use cases across 7 industries
- Explore Generative AI & LLM School at the AI Data Cloud Academy
- Visit the Snowflake Developers' Solutions Center for demos, tutorials, best practices and more—all designed to boost your development efforts.

# ABOUT SNOWFLAKE

Snowflake makes enterprise AI easy, efficient and trusted. Thousands of companies around the globe, including hundreds of the world's largest, use Snowflake's AI Data Cloud to share data, build applications, and power their business with AI. The era of enterprise AI is here

Learn more at **snowflake.com** (NYSE: SNOW)