



# 7 SNOWFLAKE REFERENCE ARCHITECTURES FOR APPLICATION BUILDERS

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



CHAMPION  
GUIDES

EBOOK

# TABLE OF CONTENTS

- 3** Why your data platform matters
- 5** Embedded analytics reference architecture
- 6** Serverless data stack reference architecture
- 7** Streaming data stack reference architecture
- 8** Machine learning and data science reference architecture
- 9** Application health and security analytics reference architecture
- 10** IoT reference architecture
- 11** Customer 360 reference architecture
- 12** Future-proof your applications
- 13** About Snowflake



# 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, there are three crucial questions every startup and established ISV provider should ask:

- **CAN OUR UNDERLYING ARCHITECTURE SCALE TO MEET THE NEEDS OF OUR FAST-GROWTH BUSINESS?**
- **CAN OUR PRODUCT INGEST AND ANALYZE LARGE AMOUNTS OF STRUCTURED AND SEMI-STRUCTURED DATA TOGETHER?**
- **MOST IMPORTANTLY, CAN WE ACCOMPLISH THESE GOALS WHILE REMAINING OPERATIONALLY EFFICIENT AND COST-EFFECTIVE?**



Today, too many organizations are burdened by infrastructure costs that arise from traditional architectures. When companies can achieve scalability only by throwing more resources at the problem, they face an expensive and never-ending problem. Traditional architectures are also riddled with operational overhead in the form of maintenance and tuning, which wastes valuable engineering time and slows growth.

The questions above 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 cloud data platform can deliver the performance and nearly infinite autoscaling needed to launch and scale applications quickly and cost-effectively. Here's what the Snowflake 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 or an unhappy user.

- **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.

- **Reduced site reliability engineering and DevOps burden**

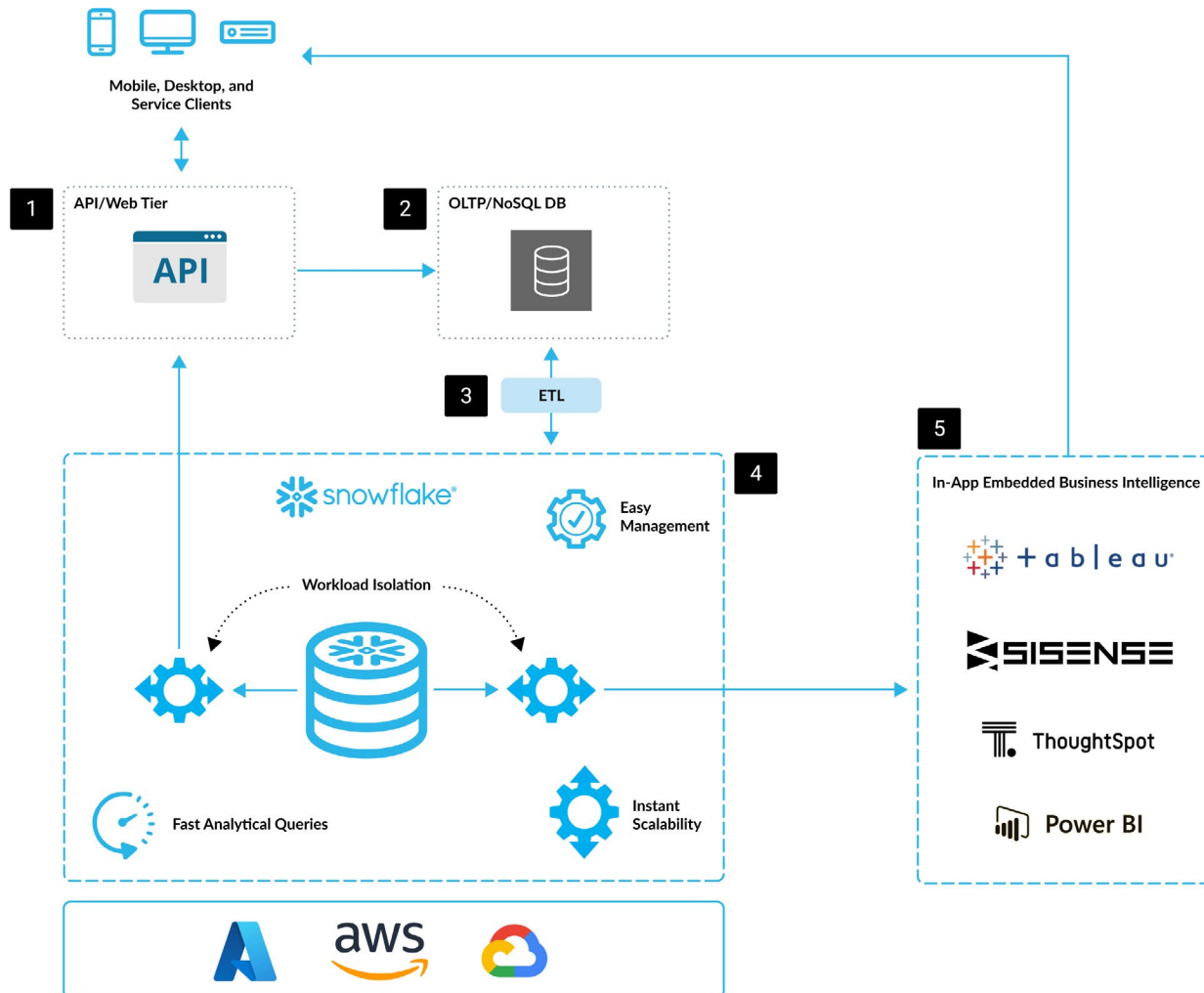
As a near-zero management platform, Snowflake automatically handles provisioning, availability, tuning, data protection, and other operations, enabling you to focus on your own application rather than maintenance.

Snowflake also ensures seamless connections to third-party platforms and APIs, easily fitting in with your existing environment.

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



# EMBEDDED ANALYTICS REFERENCE ARCHITECTURE



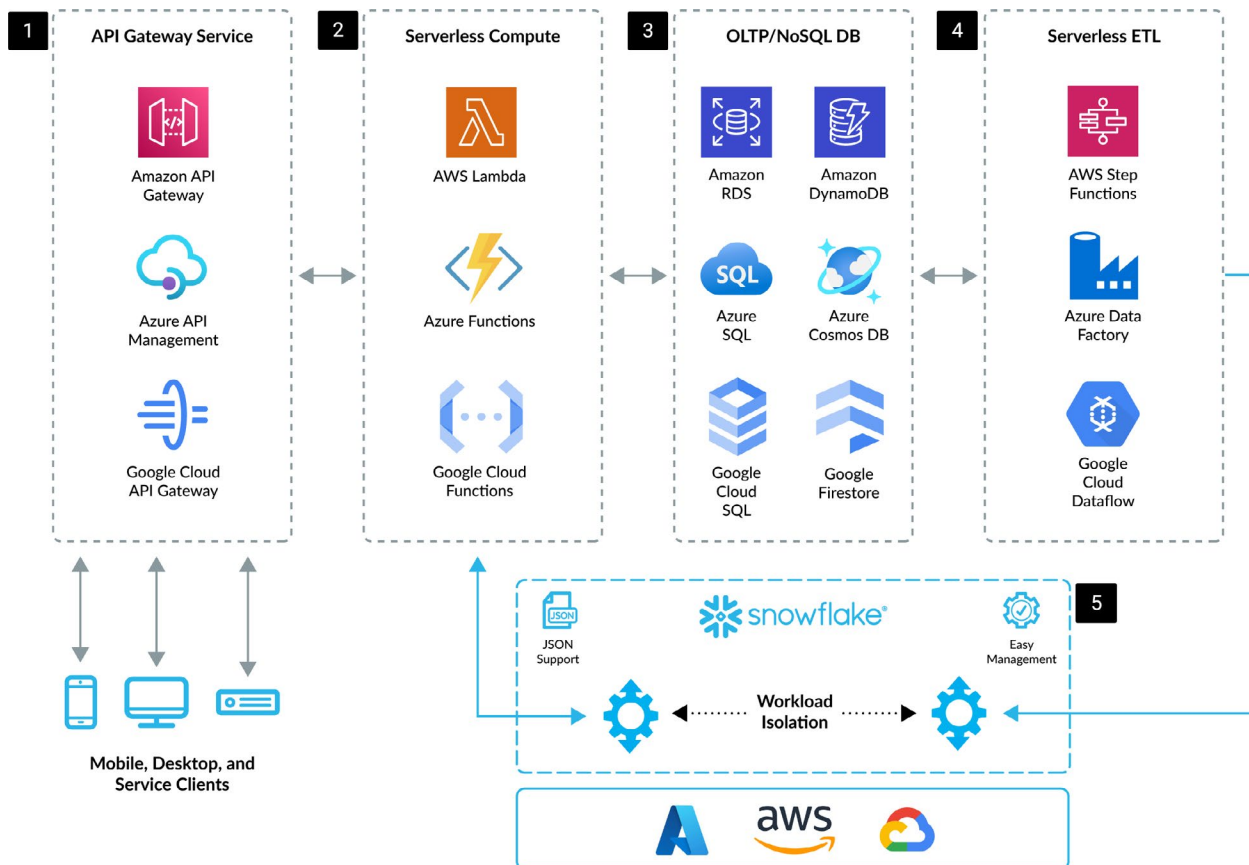
## OBJECTIVE

Build analytics-heavy applications that deliver in-app visualizations.

## DESCRIPTION

- 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 or NoSQL database supports the transaction workloads of the application.
- 3 Snowflake ingests historical transaction data via ETL infrastructure to support analytical workloads. Tools from third-party partners like FiveTran, Matillion, dbt, and HVR provide data integration and native connectivity to Snowflake.
- 4 Snowflake stores all historical data and supports queries by the application and by business intelligence tools (5). Virtual warehouses isolate workloads and autoscale compute resources to deliver high performance queries and unlimited concurrency.
- 5 Embedded business intelligence tools or open-source charting libraries support analytics from within the application. Custom applications can be built directly on Snowflake or embedded business intelligence tools to address analytics needs.

# SERVERLESS DATA STACK REFERENCE ARCHITECTURE



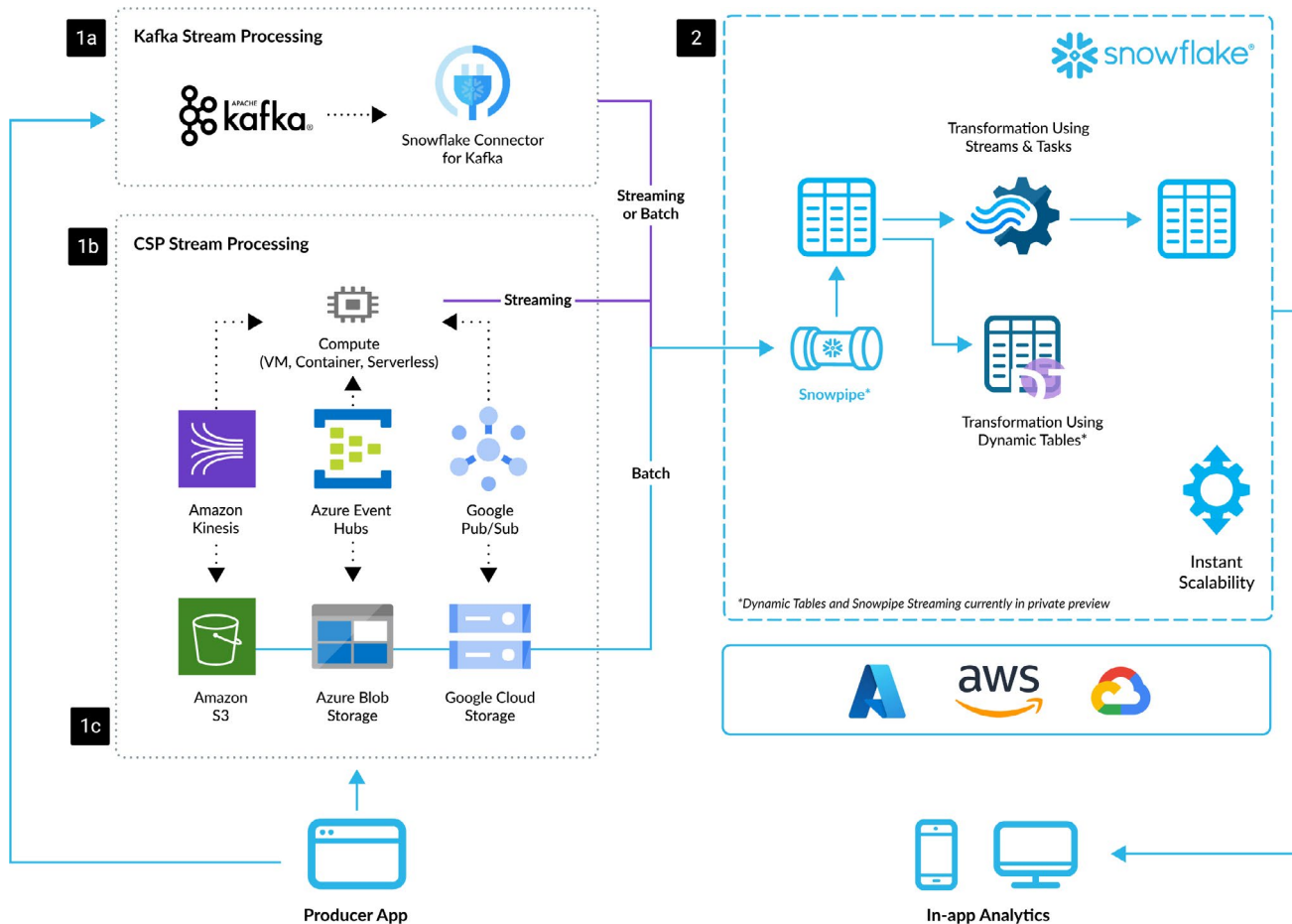
## OBJECTIVE

Build data-intensive applications that run on serverless infrastructures.

## DESCRIPTION

- 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.
- 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.
- An OLTP or NoSQL database provides the application with high-capacity, simple transaction processing. This NoSQL/OLTP database can also be a serverless service.
- An ETL serverless stack orchestrates the workflow and loads data into Snowflake.
- 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. Native JSON support enables easy ingestion and querying of flexible schema data alongside structured data.

# STREAMING DATA STACK REFERENCE ARCHITECTURE



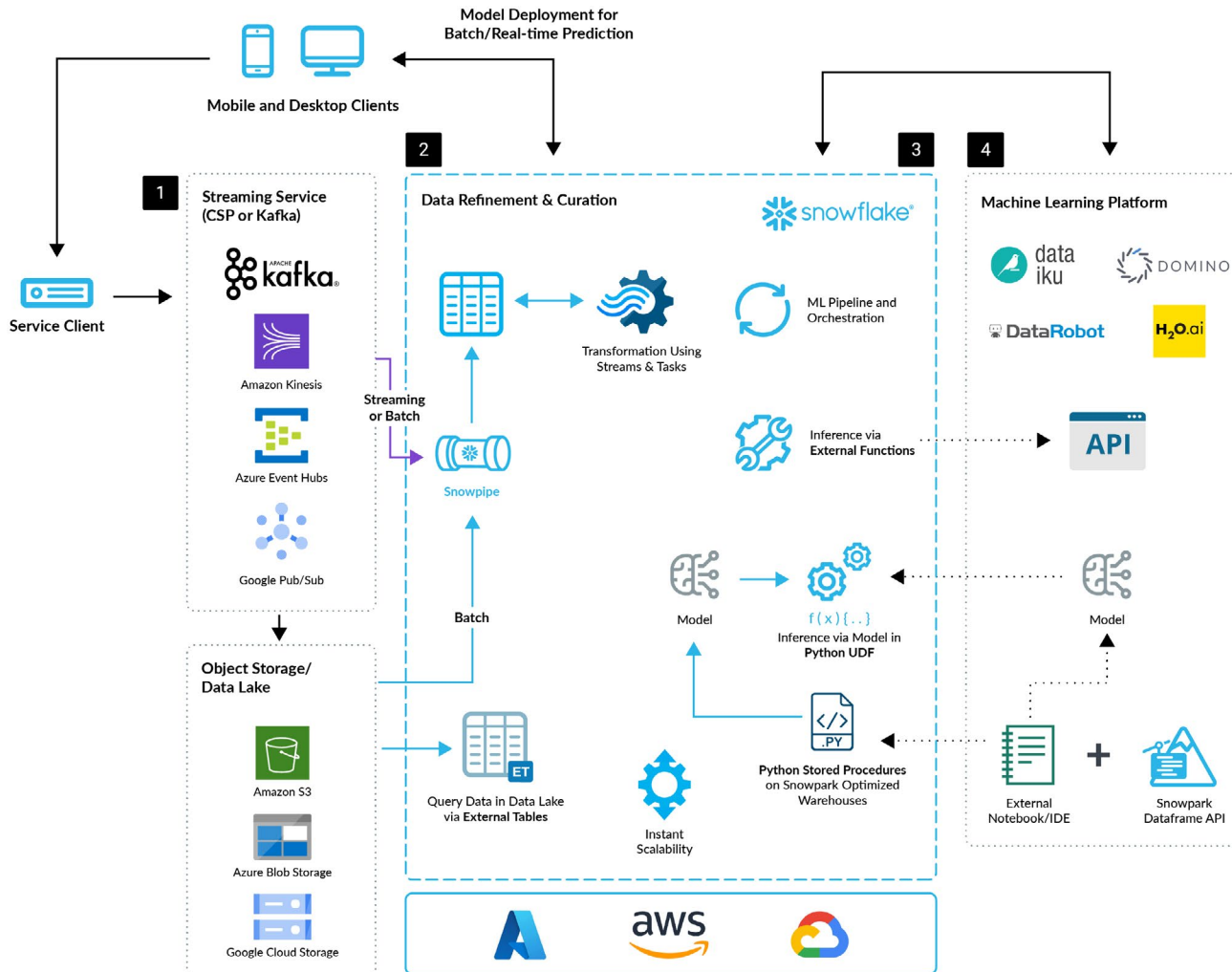
## OBJECTIVE

Build data-intensive applications that rely on streaming data ingestion and analysis.

## DESCRIPTION

- 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 (currently in public preview) based on configuration. 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 of utmost importance, the Cloud Service Providers (CSP) streaming technologies can be paired with custom code using the Java SDK for Snowpipe Streaming (currently in public preview) to get data queryable in seconds.
  - 1c** The 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.
- 2** Snowflake ingests data from the streaming service into a table and stores the streamed data for analysis. 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. Dynamic Tables (currently in private preview) can be used to simplify transformation; it is written using SQL and runs automatically in Snowflake.

# MACHINE LEARNING AND DATA SCIENCE REFERENCE ARCHITECTURE



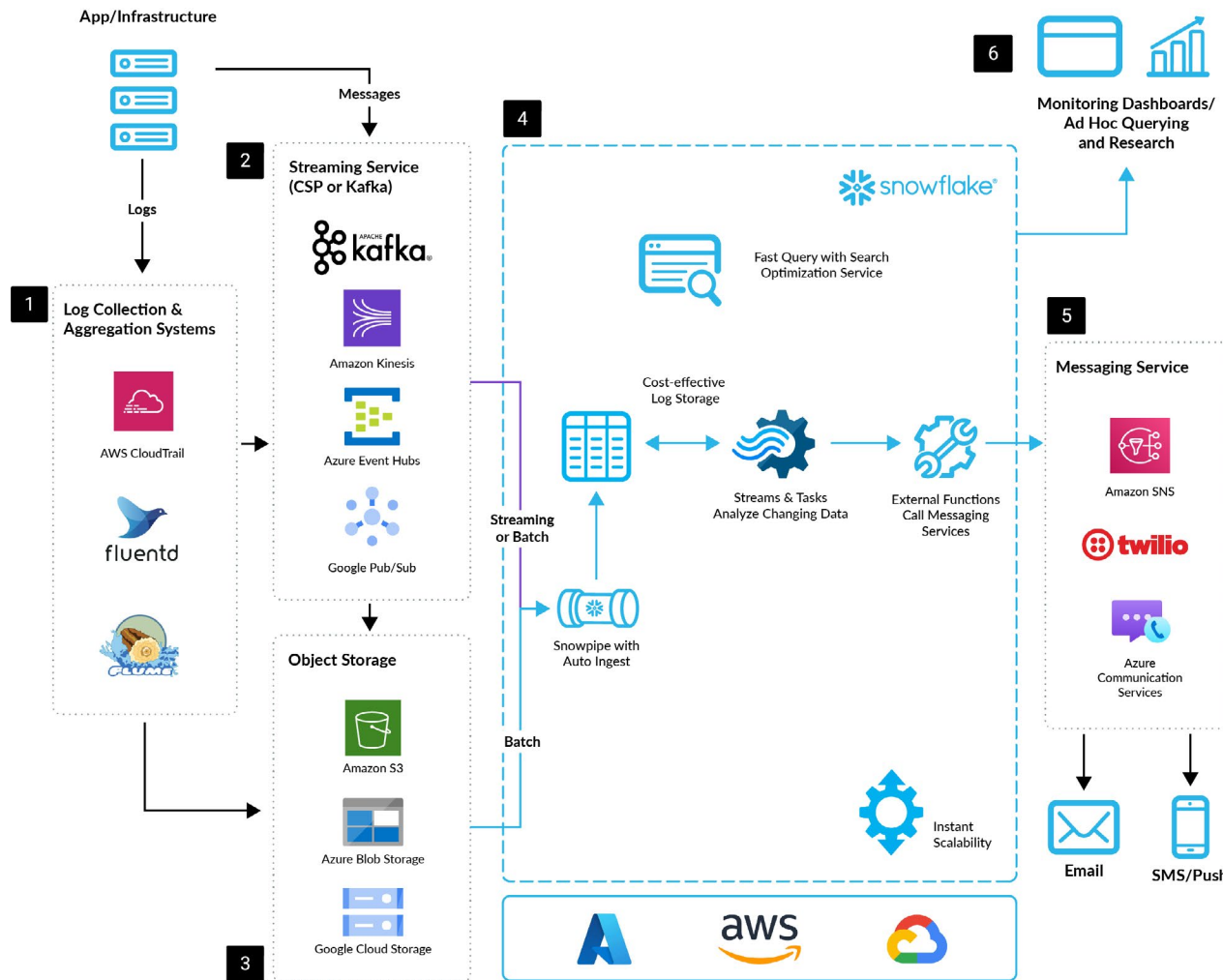
## OBJECTIVE

Train machine learning (ML) models to build predictive applications, such as recommendation engines.

## DESCRIPTION

- 1** The application produces training data, which is sent into the streaming service. Streaming services can then batch that data to be stored in the data lake, which can be queried directly as an external table in Snowflake, or to send it directly to Snowflake for storage and querying. The streaming service buffers the training data to ensure reliable and continuous ingestion.
- 2** Once data is available in Snowflake for query, 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. The Snowpark developer platform 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.
- 3** Using the data stored in Snowflake, data scientists train models with ML platforms. Once trained, the model artifacts are deployed on the training platforms (4) to support predictions.
- 4** Training can also be done directly in Snowflake leveraging Snowpark and Snowpark Optimized Warehouses. The application performs predictions using these models in real-time or 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 application. For real-time predictions, they can be run from the client application by calling the UDF or External Functions hosting the model.

# APPLICATION HEALTH AND SECURITY ANALYTICS REFERENCE ARCHITECTURE



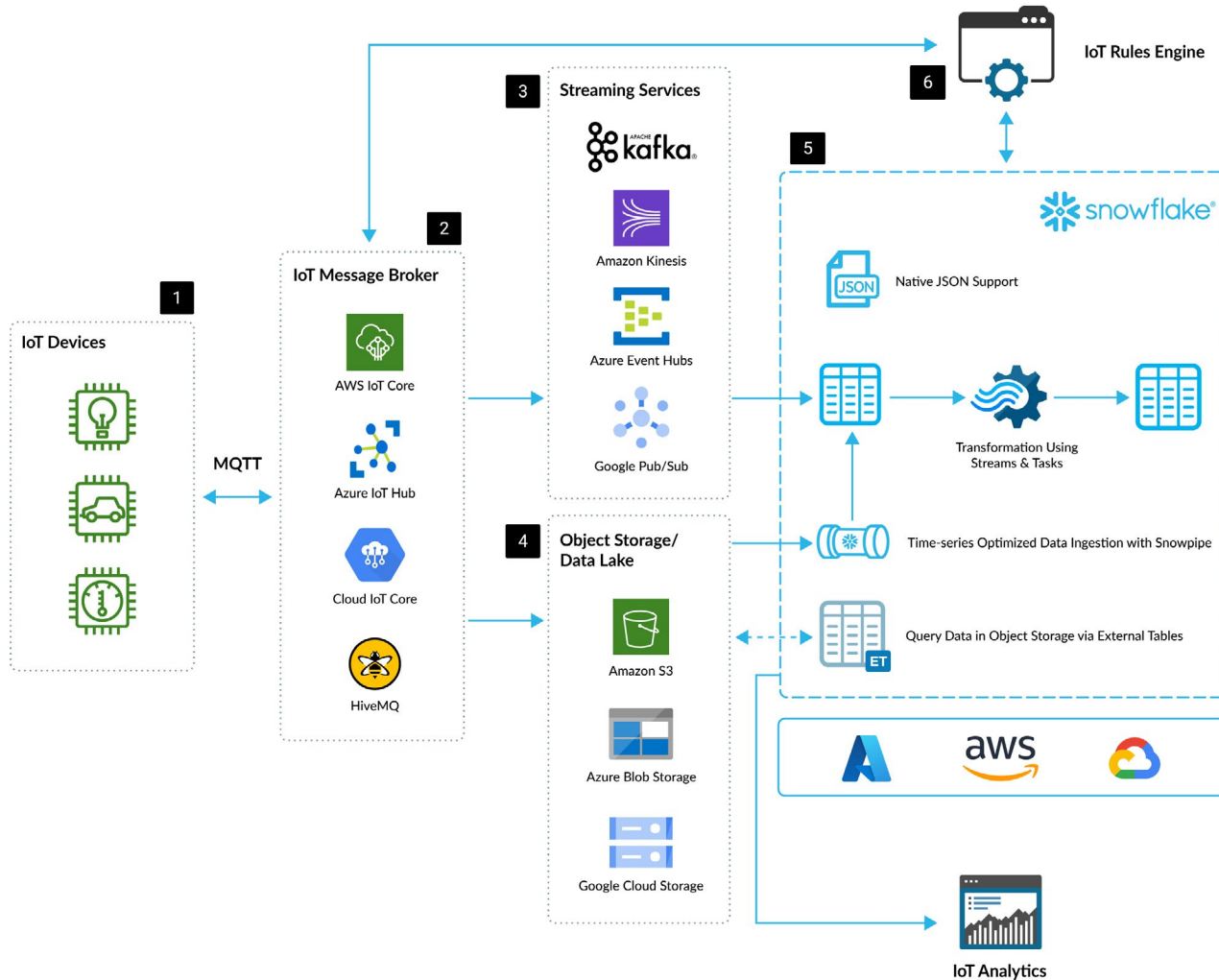
## OBJECTIVE

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

## DESCRIPTION

- 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).
- The streaming service buffers log data to ensure reliable and continuous ingestion.
- Depending on which log collector and aggregation system is used, data can be staged in object storage without the need for a streaming service.
- 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.
- Tasks can detect suspicious activity or health concerns and store and/or alert utilizing external messaging services via external functions. A messaging service uses email, SMS, or push notifications to notify operations teams of events that require attention.
- Operations teams can also monitor the application via dashboards or ad hoc queries. The instant scalability provided by Snowflake enables quick threat detection and response capabilities needed by the organization.

# IOT REFERENCE ARCHITECTURE



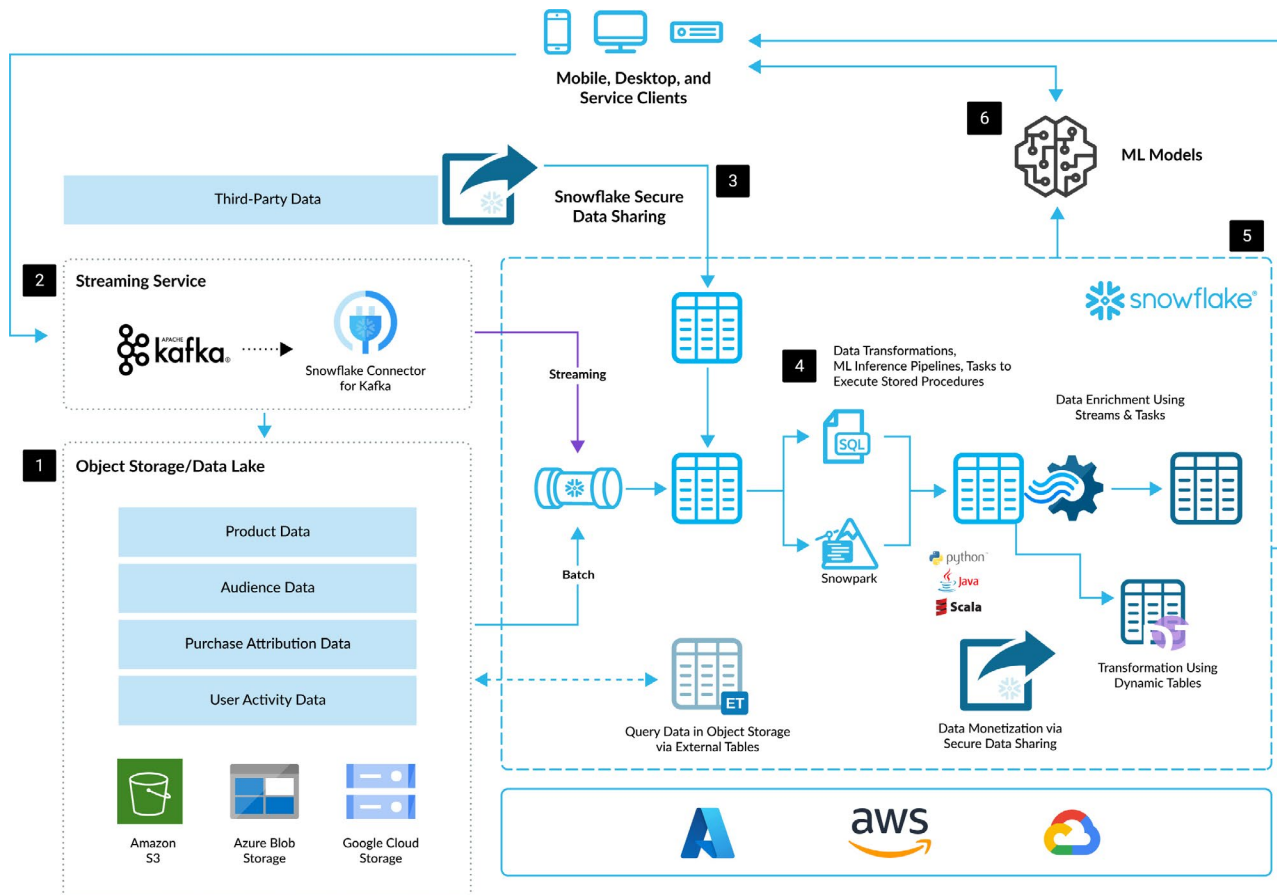
## OBJECTIVE

Build applications that analyze large volumes of time-series data from IoT devices and respond in real time.

## DESCRIPTION

- Smart devices, sensors, and other IoT devices generate continuous data.
- 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.
- A streaming service is used to ingest and buffer real-time device data, thus ensuring reliable ingestion and delivery to a staging table in Snowflake (5).
- 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).
- 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's Streams and Tasks features or Dynamic Tables (currently in private preview) automate the workflows required to ingest and aggregate incoming data.
- 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.

# CUSTOMER 360 REFERENCE ARCHITECTURE



## OBJECTIVE

Build sales and marketing applications that use historical and real-time data to accomplish “360-degree view” customer goals, such as finding new segments and sending personalized offers.

## DESCRIPTION

- 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 Secure Data Sharing enables 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 then 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. The 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.

# FUTURE-PROOF YOUR APPLICATIONS

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.

And that's something you can hang your app on.

## NEXT STEPS

Ready to build? Still have some questions? Take the next step with 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.

**Check out the Snowflake Developers' Center** for demos, tutorials, best practices, and more—all designed to boost your development efforts.



## ABOUT SNOWFLAKE

Snowflake enables every organization to mobilize their data with Snowflake's Data Cloud. Customers use the Data Cloud to unite siloed data, discover and securely share data, and execute diverse analytic workloads. Wherever data or users live, Snowflake delivers a single data experience that spans multiple clouds and geographies. Thousands of customers across many industries, including 573 of the 2022 Forbes Global 2000 (G2K) as of January 31, 2023, use Snowflake Data Cloud to power their businesses.

Learn more at [snowflake.com](https://www.snowflake.com)



© 2023 Snowflake Inc. All rights reserved. Snowflake, the Snowflake logo, and all other Snowflake product, feature and service names mentioned herein are registered trademarks or trademarks of Snowflake Inc. in the United States and other countries. All other brand names or logos mentioned or used herein are for identification purposes only and may be the trademarks of their respective holder(s). Snowflake may not be associated with, or be sponsored or endorsed by, any such holder(s).