snowflake®

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

WHY YOUR DATA PLATFORM MATTERS

It's safe to say data application builders will never worry about a lack of data. Approximately 40 zettabytes (ZB) of new data was generated in 2019, and IDC predicts that with a steady growth trajectory, 175 ZB will be generated in 2025. Although these ever-increasing amounts of data present 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, companies 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 Cloud Data Platform provides:

- **High performance and unlimited 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.

- **SQL for all data**

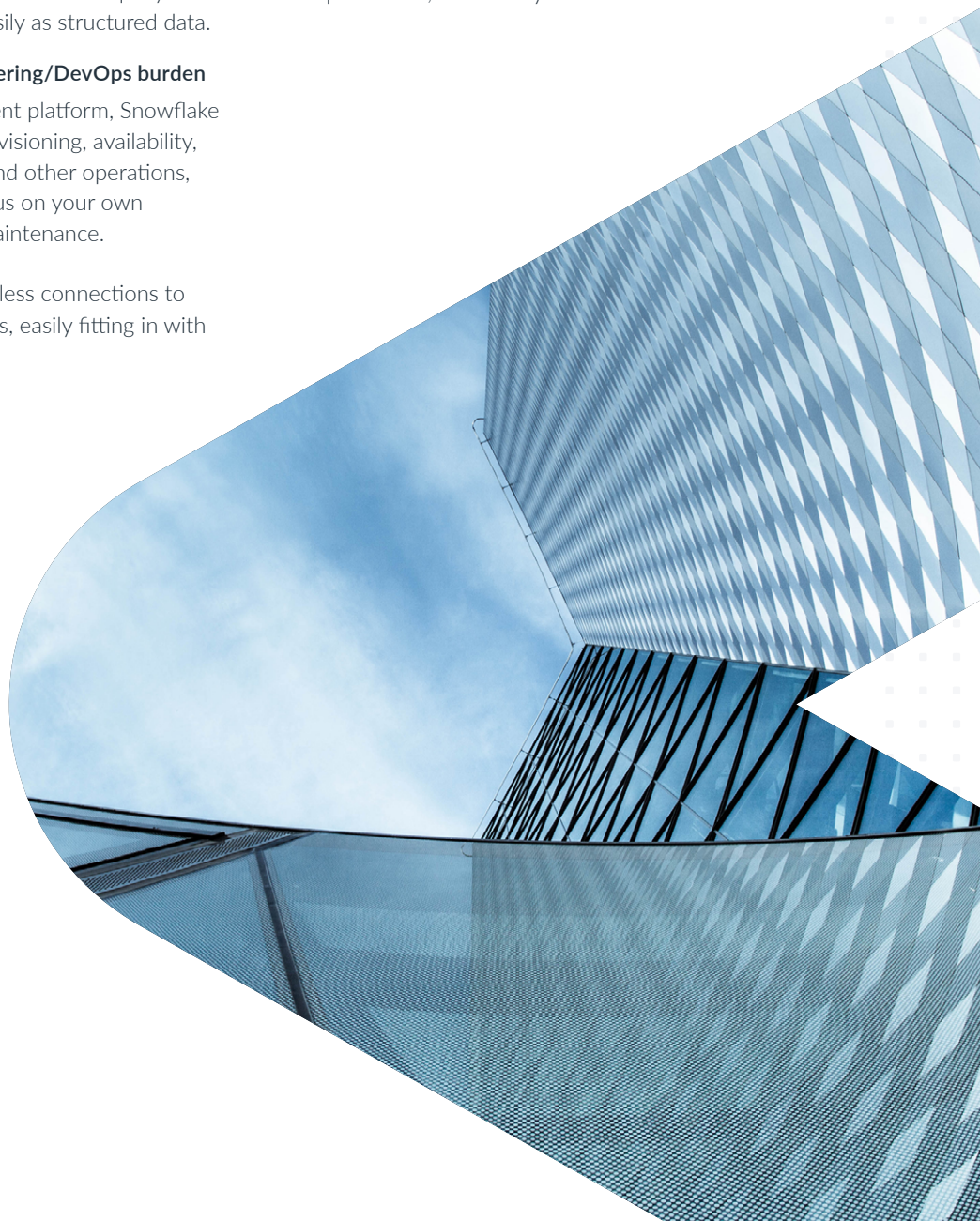
  Snowflake ingests JSON, Avro, Parquet, and other data without transformations or requiring pipeline fixes every time the schema changes. With ANSI SQL, Snowflake enables your teams to query semi-structured data just as easily as structured data.

- **No Site Reliability Engineering/DevOps burden**

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

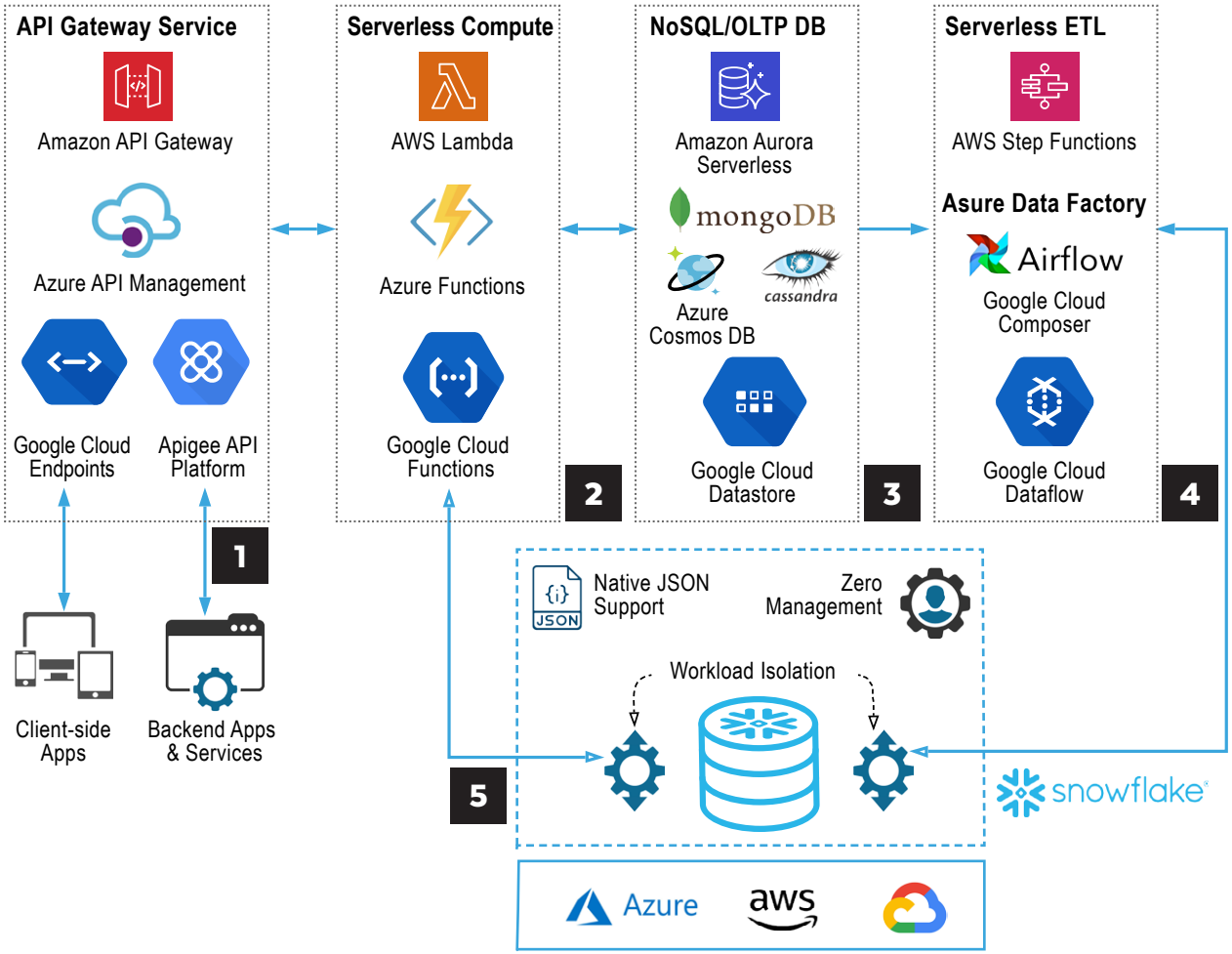
Snowflake also ensures seamless connections to third-party platforms and APIs, easily fitting in with your existing environment.

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.



SERVERLESS DATA STACK REFERENCE ARCHITECTURE

SERVERLESS DATA STACK



OBJECTIVE

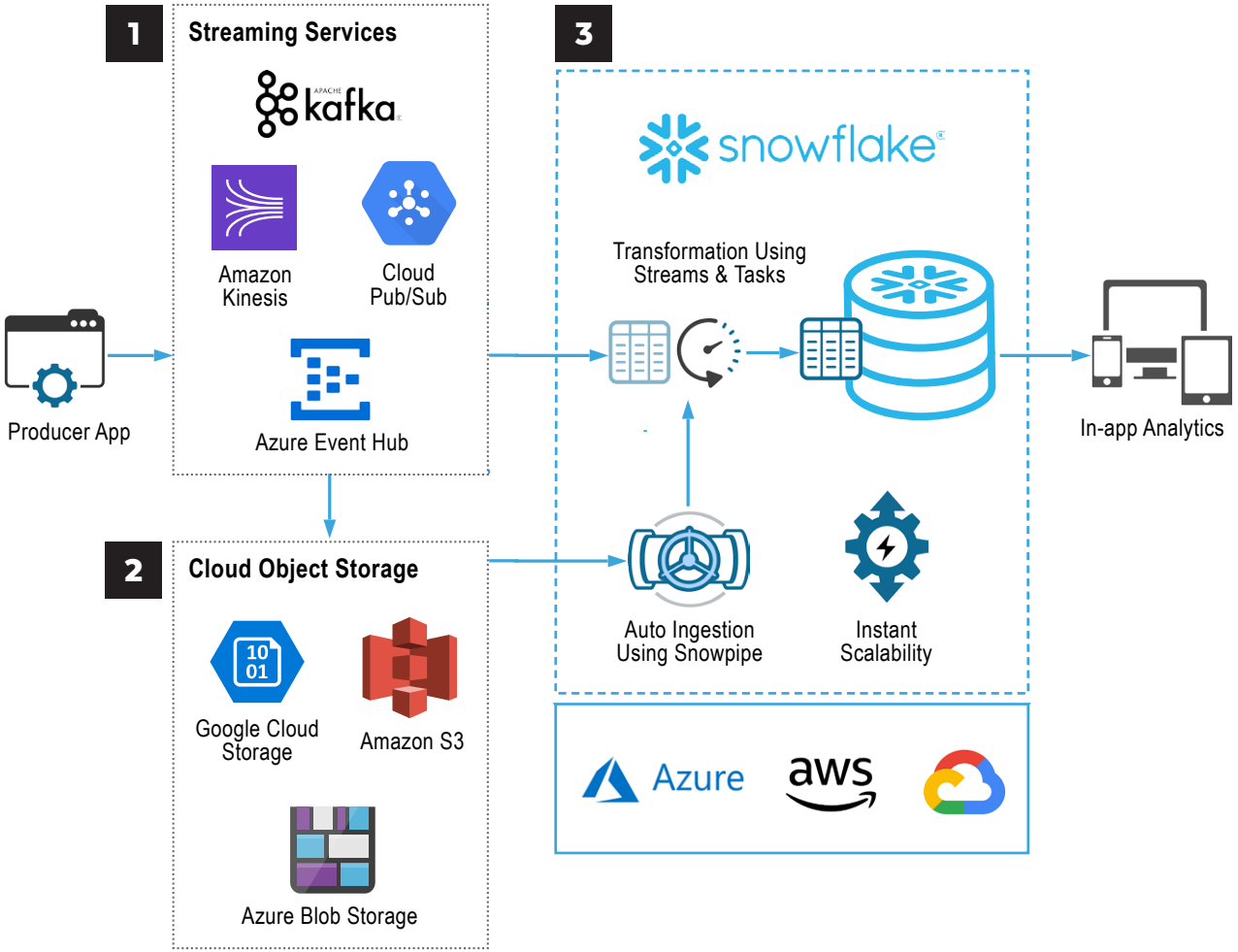
Build data intensive applications that run on serverless infrastructures.

DESCRIPTION

1. The client-side app, running on mobile or web devices, invokes the application logic on the serverless compute via an API gateway service. The gateway authenticates the API calls and throttles them, based on SLAs.
2. Serverless compute runs the application logic and scales on demand, without the need to provision or manage servers. The application queries Snowflake data (5) for runtime decisions, such as delivering product recommendations or powering a dashboard for analysis.
3. An OLTP or NoSQL database provides the application with high-capacity transaction processing. This NoSQL/OLTP database can also be a serverless service.
4. An ETL serverless stack orchestrates the workflow and loads transaction data into Snowflake.
5. Snowflake ingests data in batches or in streams and makes it available to the application for queries. Snowflake scales automatically to keep pace with the data pipeline and ensure data is always fresh. 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

STREAMING DATA STACK



OBJECTIVE

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

DESCRIPTION

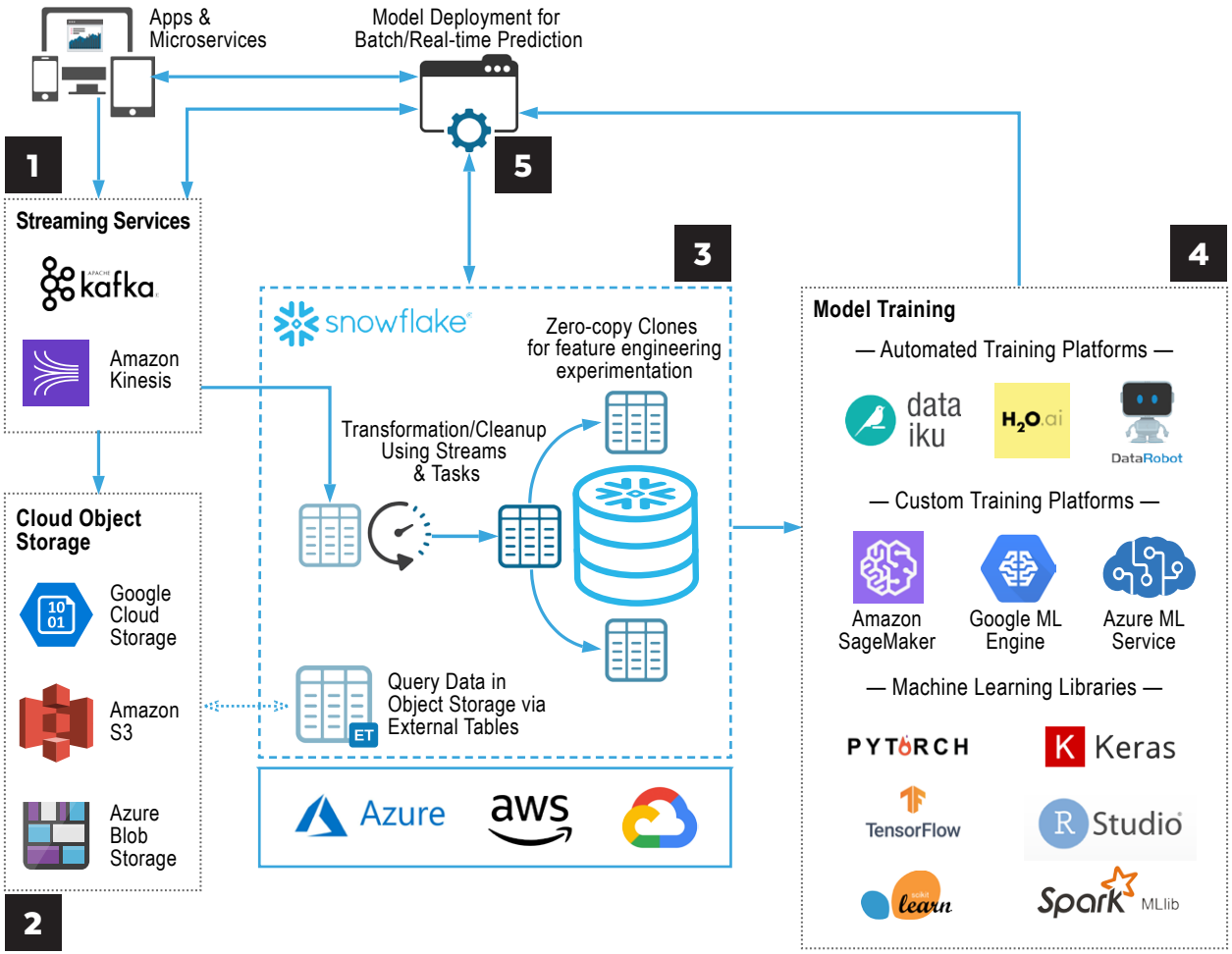
1. The producer application generates continuous data that the streaming service ingests and buffers to account for data rate differences between the producer and consumers. Depending on the application's needs, Snowflake ingests data directly from the streaming service or via cloud object storage (2).

2. In cases where the application requires raw data to persist in cloud object storage, the streaming service processes the raw data and batches it into larger chunks, thus lowering the API storage expenses. When Amazon Kinesis is used as the streaming service, data is staged in cloud object storage before ingestion.

3. Snowflake ingests data from the streaming service into a staging table and stores the streamed data for analysis. Its 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. Snowpipe with Auto-Ingest automates the data ingestion from cloud object storage.

MACHINE LEARNING AND DATA SCIENCE REFERENCE ARCHITECTURE

MACHINE LEARNING AND DATA SCIENCE



OBJECTIVE

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

DESCRIPTION

1. The application produces training data, which Snowflake (3) ingests via the streaming service or via cloud object storage (2). The streaming service buffers the training data to ensure reliable and continuous ingestion.

2. When cloud object storage is used, the streaming service batches training data into larger chunks to lower the API storage expenses.

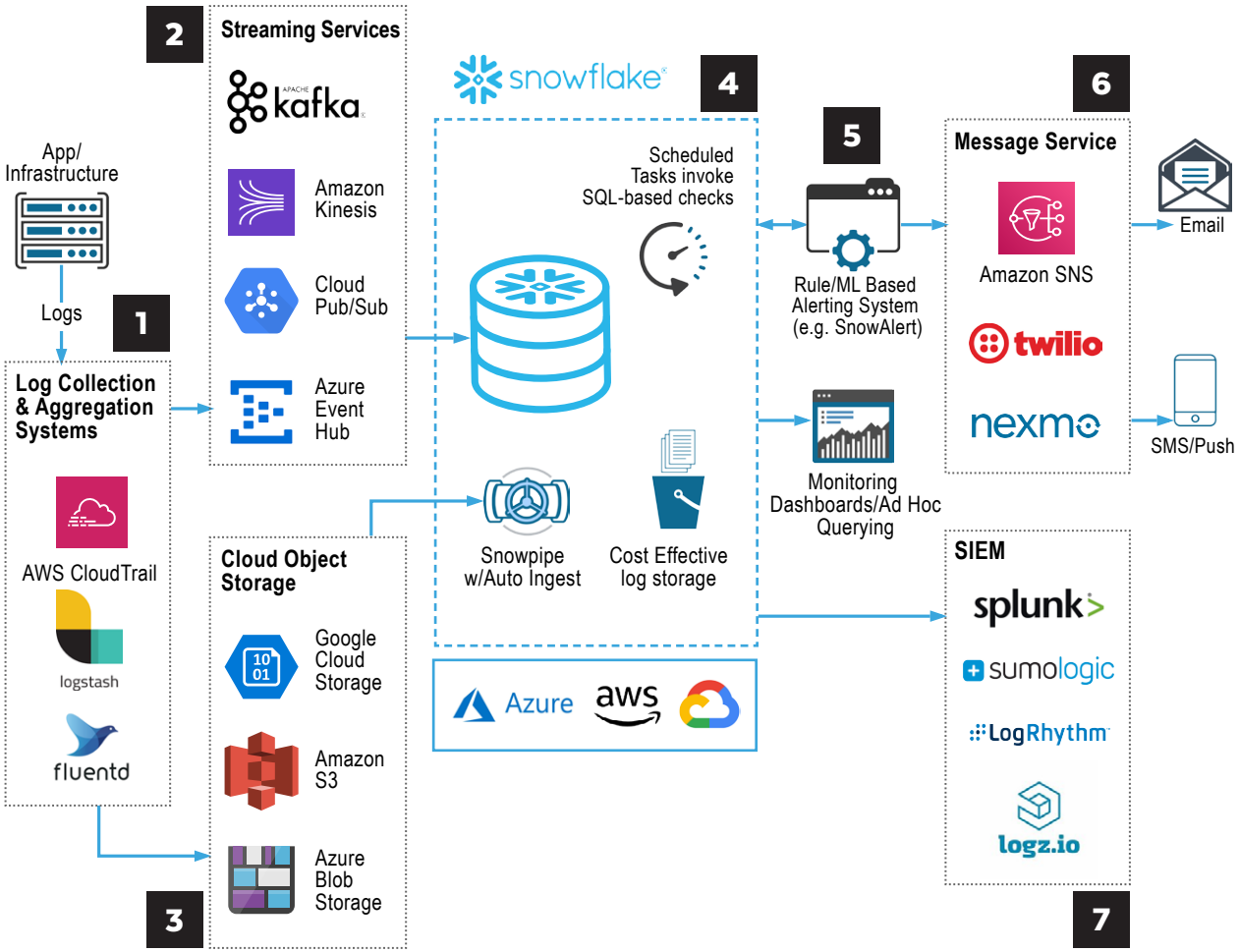
3. Snowflake ingests data into a staging table. When new data is detected, the Streams and Tasks feature schedule required transformations. Multiple streams and tasks can be chained to implement a complex data pipeline. External Tables support queries of data in cloud object storage without ingestion. Data scientists can create zero-copy clones of the training data to support feature engineering and experimentation.

4. Using the data stored in Snowflake, data scientists train models with ML platforms and available libraries. Once the model artifacts are trained, they are deployed on the training platforms or on a separate process (5) to support predictions.

5. The application performs predictions 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. In cases where subsecond response time is required, predictions can also be performed using input data from the streaming service.

APPLICATION HEALTH AND SECURITY ANALYTICS REFERENCE ARCHITECTURE

APPLICATION HEALTH AND SECURITY ANALYTICS



OBJECTIVE

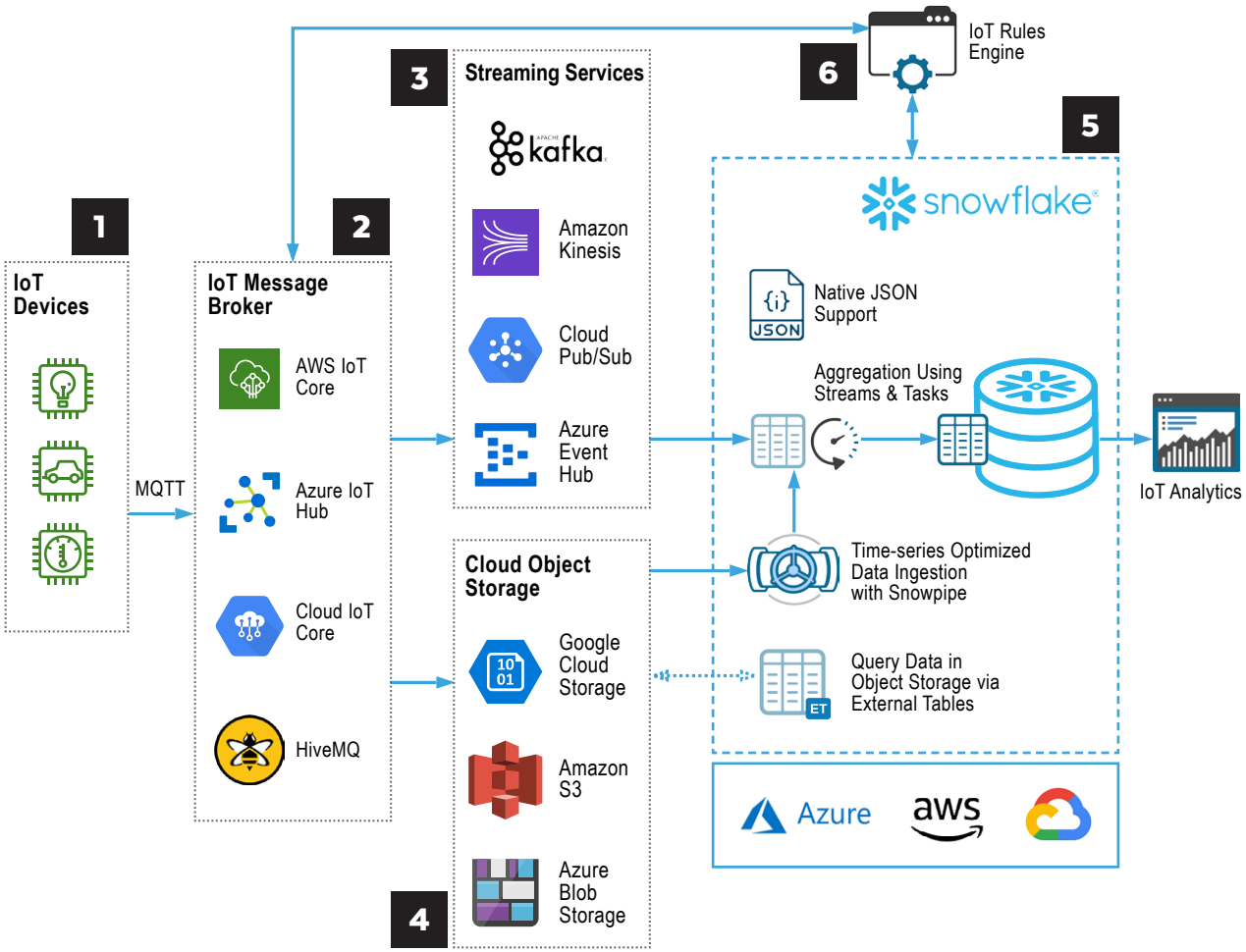
Analyze large volumes of log data to identify security threats and monitor application health.

DESCRIPTION

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 cloud 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 cloud 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 cloud object storage. Scheduled tasks invoke SQL-based queries to detect suspicious behavior or application health concerns.
5. External rule-based alerting systems, such as SnowAlert, can detect suspicious activity or health concerns. Operations teams can monitor the application via dashboards or ad hoc queries.
6. A messaging service uses email, SMS, or push notifications to notify operations teams of events that require attention.
7. SIEM systems can leverage data in Snowflake for advanced searching and alerting capabilities.

IOT REFERENCE ARCHITECTURE

IOT



OBJECTIVE

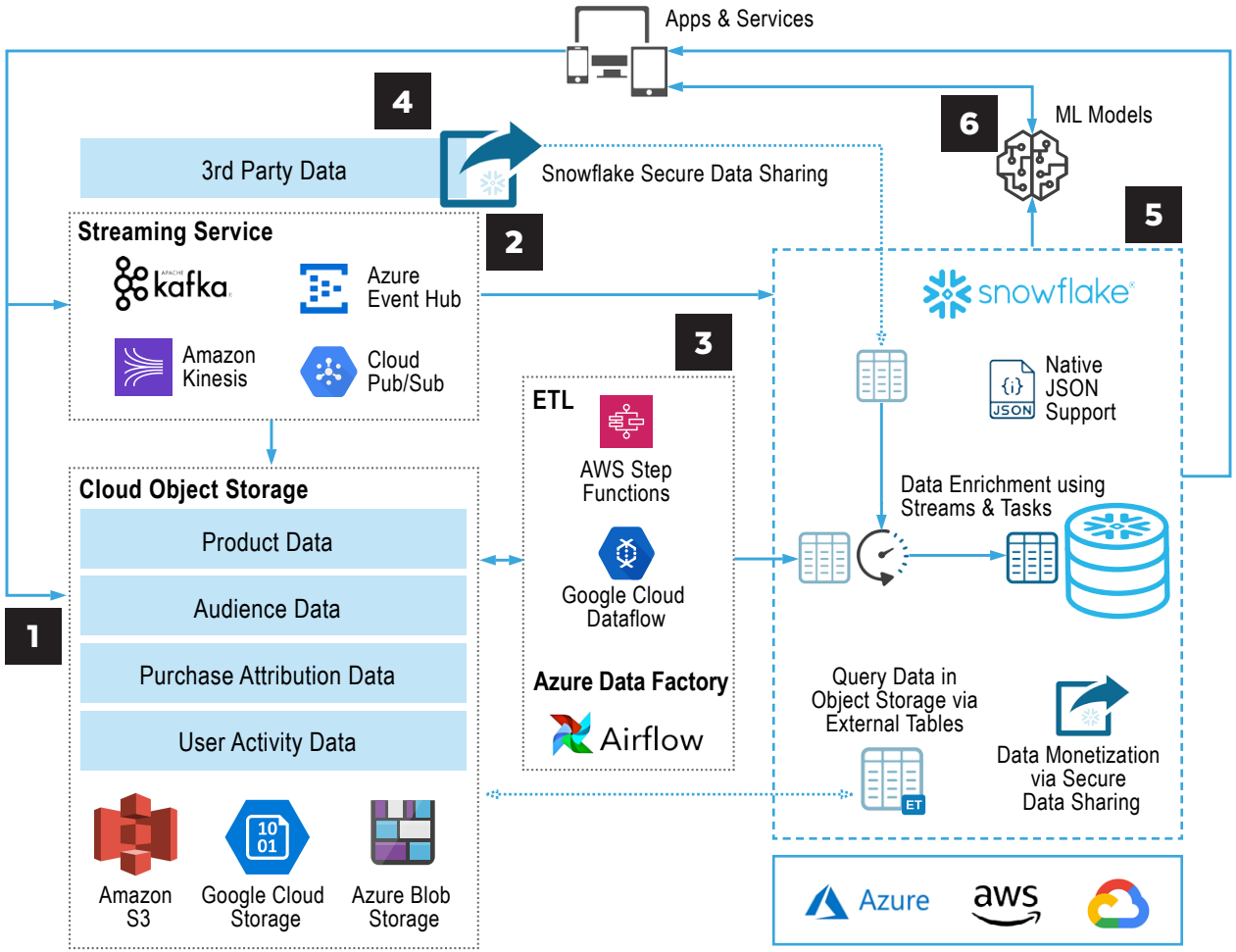
Build applications that analyze large volumes of time-series data from IoT devices and respond in real time.

DESCRIPTION

1. Smart devices, sensors, and other IoT devices generate continuous data.
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 is used to ingest and buffer 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, cloud object storage is used to stage batch data prior to ingestion. For example, minute-by-minute data may be stored in cloud object storage, whereas aggregated data over a longer period may be stored in 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's Streams and Tasks features 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 controls devices.

CUSTOMER 360 REFERENCE ARCHITECTURE

CUSTOMER 360



OBJECTIVE

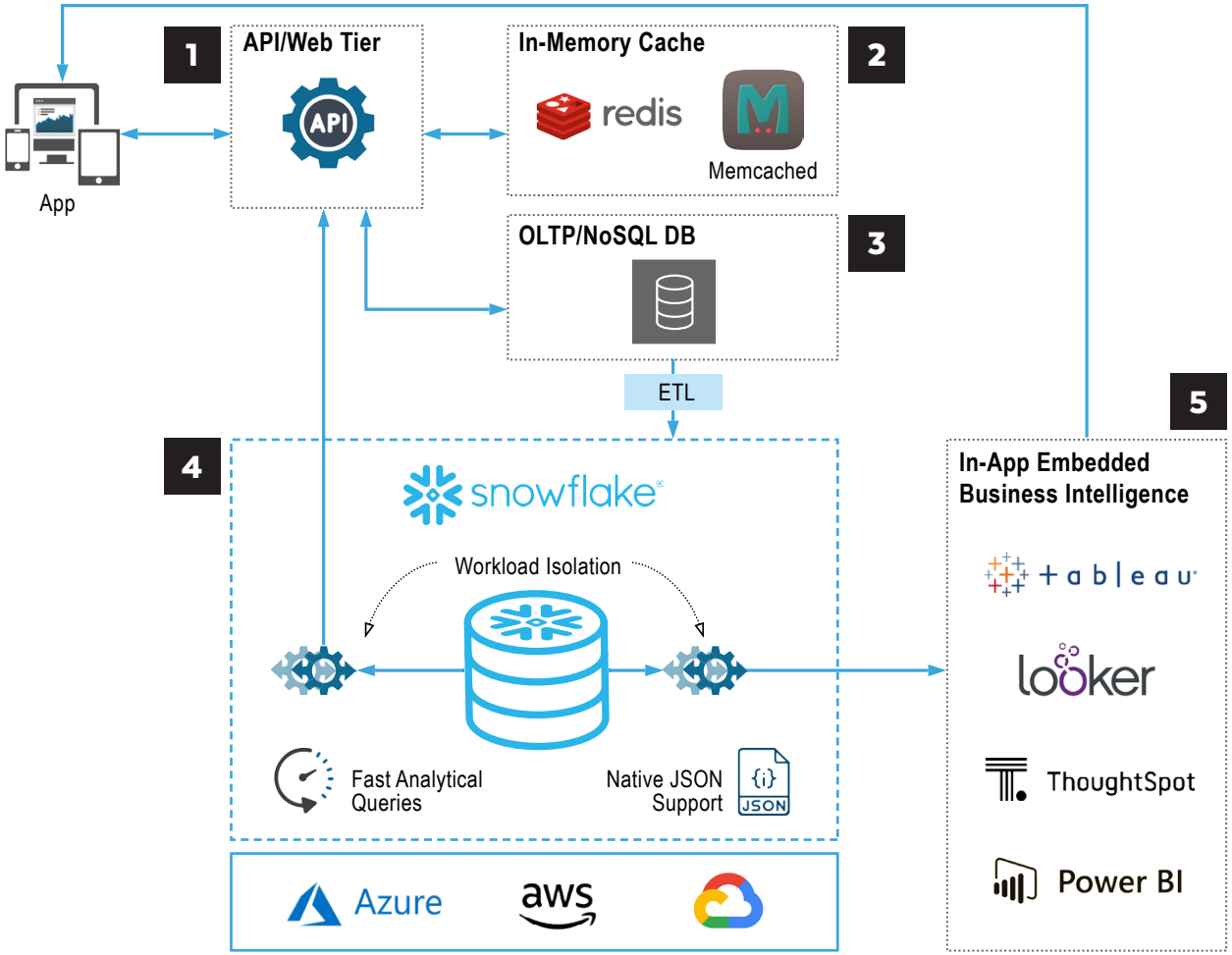
Build sales and marketing applications that use historical and real-time data to accomplish "360-degree view" customer goals, such as finding new segments and sending personalized offers.

DESCRIPTION

1. Cloud object storage stages application data, such as data on 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. ETL services orchestrate the workflow to load data from cloud object storage into Snowflake.
4. Snowflake Secure Data Sharing enables data from third-party sources to be used without copying or moving the data.
5. Snowflake supports all the analytics workloads within the application. External Tables support queries of data in cloud 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.

EMBEDDED ANALYTICS REFERENCE ARCHITECTURE

EMBEDDED ANALYTICS



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. In-memory cache provides in-session read requests to ensure millisecond response time.
3. An OLTP or NoSQL database supports the transaction workloads of the application. Snowflake (4) ingests historical transaction data via ETL infrastructure to support analytical workloads.
4. Snowflake stores all historical data and supports queries by the application and 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.

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 to grow your business. 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.

ABOUT SNOWFLAKE

Snowflake's cloud data platform shatters the barriers that have prevented organizations of all sizes from unleashing the true value from their data. More than 2,000 customers deploy Snowflake to advance their businesses beyond what was once possible by deriving all the insights from all their data by all their business users. Snowflake equips organizations with a single, integrated platform that offers the only data warehouse built for the cloud; instant, secure, and governed access to their entire network of data; and a core architecture to enable many types of data workloads, including a single platform for developing modern data applications. Snowflake: Data without limits. Find out more at **snowflake.com**

CITATIONS

[1] "The Digitization of the World From Edge to Core." IDC. bit.ly/2QuFiKk