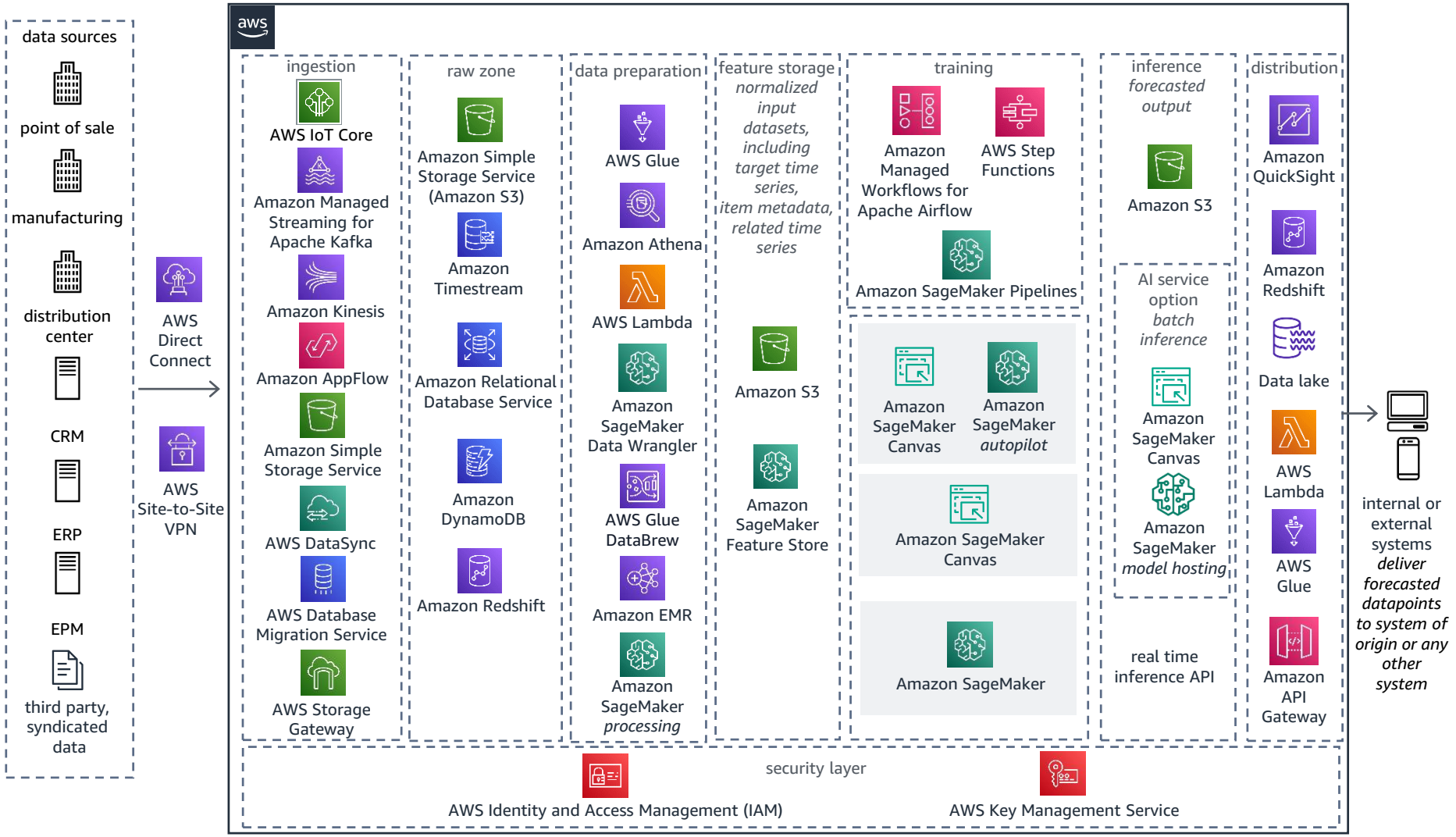
Guidance for Demand Forecasting for Retail on AWS

Follow this logical end-to-end architectural design to perform demand forecasting for retail by using AWS services and highlighting options for low-code, no-code, or extremely advanced approaches.



Demand forecasting in retail is the process of developing an estimate of future customer demand. It includes the analyses of numerous internal and external variables that impact demand, from seasonality to promotions, inventory levels to market trends. Generally, demand forecasting considers historical data and other analytical information to produce the most accurate predictions. More specifically, methods of demand forecasting entail using predictive analytics of historical data to understand and predict customer demand in order to understand key economic conditions and assist in making crucial supply decisions to optimize business profitability.

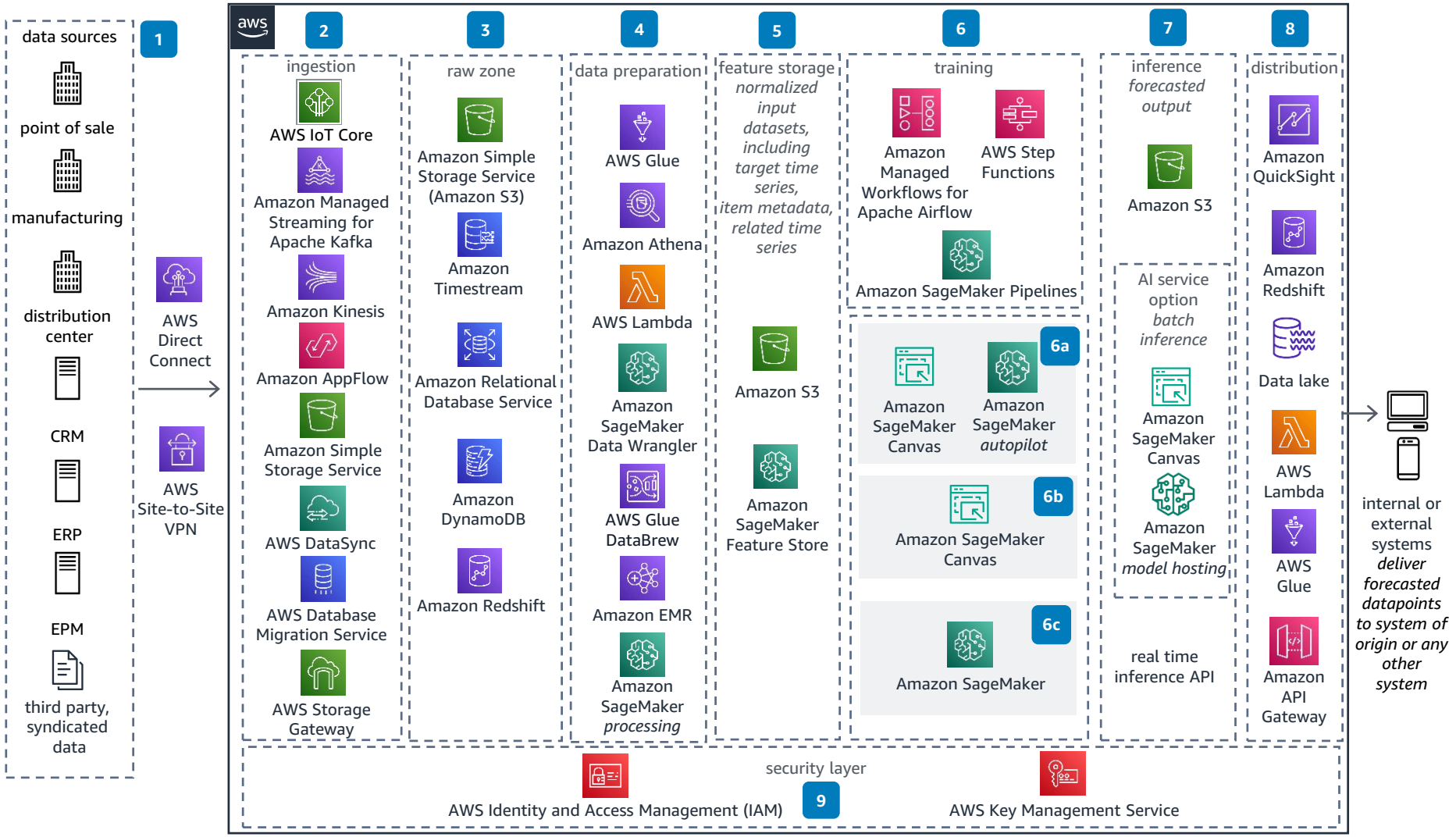
Using an iterative process of machine learning for demand forecasting can help customers:

- Optimize inventory management.
- Optimize replenishment processes (reduce warehouse and logistic costs).
- Enable business growth.
- Improve customer satisfaction.
- Reduce waste and operational cost.

Reviewed for technical accuracy September 26, 2024

AWS Reference Architecture

Data Ingestion



1 Current and historical event data is gathered from retail stores (point of sale), manufacturing and distribution centers, customer relationship management (CRM), enterprise resource planning (ERP), enterprise performance management (EPM), and third-party systems like product vendors and logistic partners. For hybrid architectures, customers can use **AWS Direct Connect** or **AWS Site-to-Site VPN**.

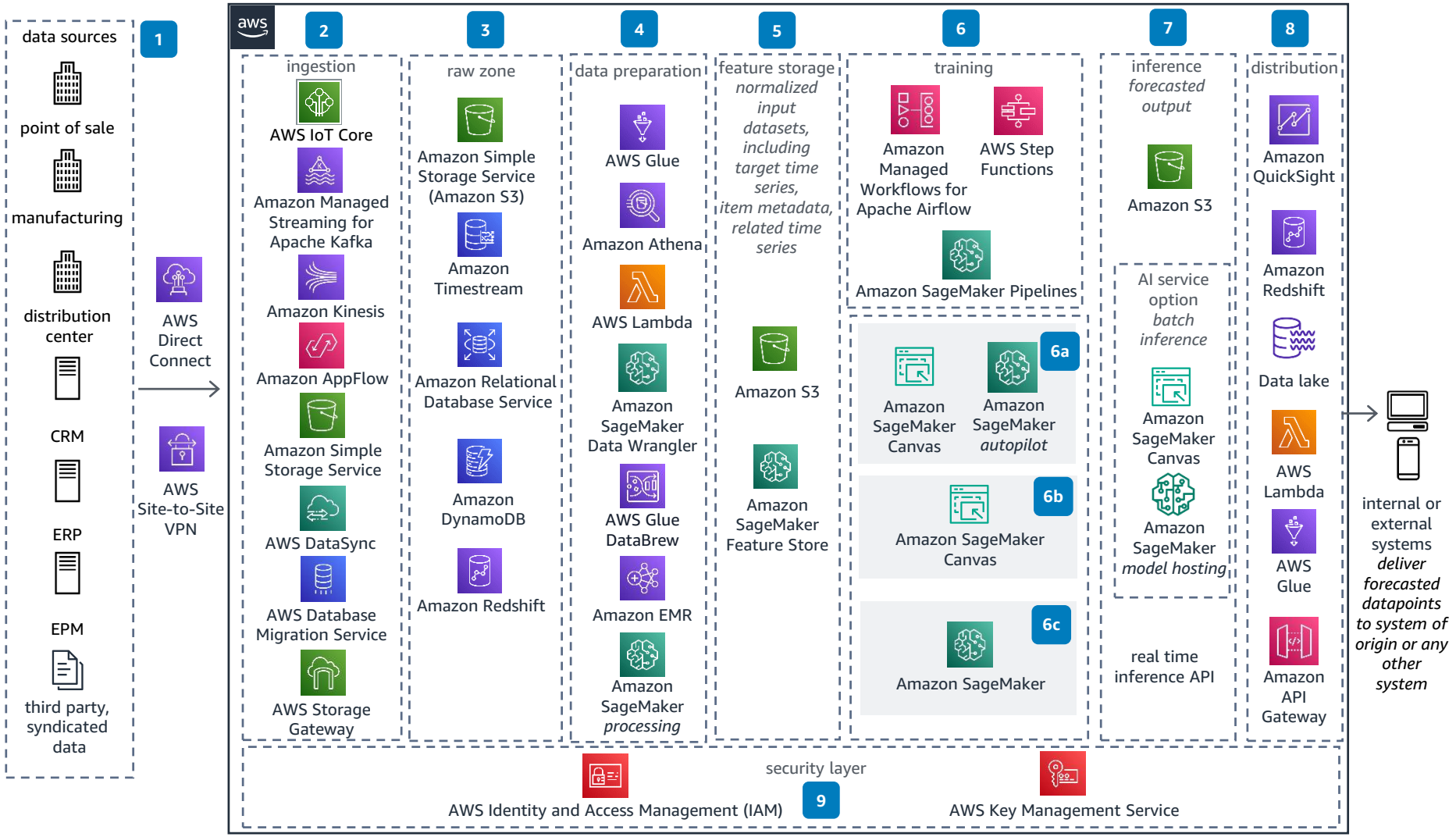
2 Data is ingested in batch or real-time pipelines. AWS provides many tools capable of pulling data from source endpoints, including services such as **Amazon Kinesis** for streaming data ingestion and **Amazon IoT Core** for IoT devices. Customers can use **AWS Database Migration Service** (AWS DMS) to fetch data from existing databases. Customers can also use **AWS DataSync** and **AWS Storage Gateway** for transfers. Optionally, customers can upload data into **Amazon Simple Storage Service** (Amazon S3) using the **Amazon S3** CLI commands, API, or console. Customers can also use **Amazon AppFlow** for SAP and Salesforce.

3 Data stores like **Amazon S3**, **Amazon Timestream**, **Amazon Relational Database Service** (Amazon RDS), **Amazon DynamoDB**, and **Amazon Redshift** provide an intermediate landing zone (also called a *raw zone*) for historical and real-time time series data, item metadata, and related time series data.

Reviewed for technical accuracy September 26, 2024

AWS Reference Architecture

Data Preparation and Feature Engineering

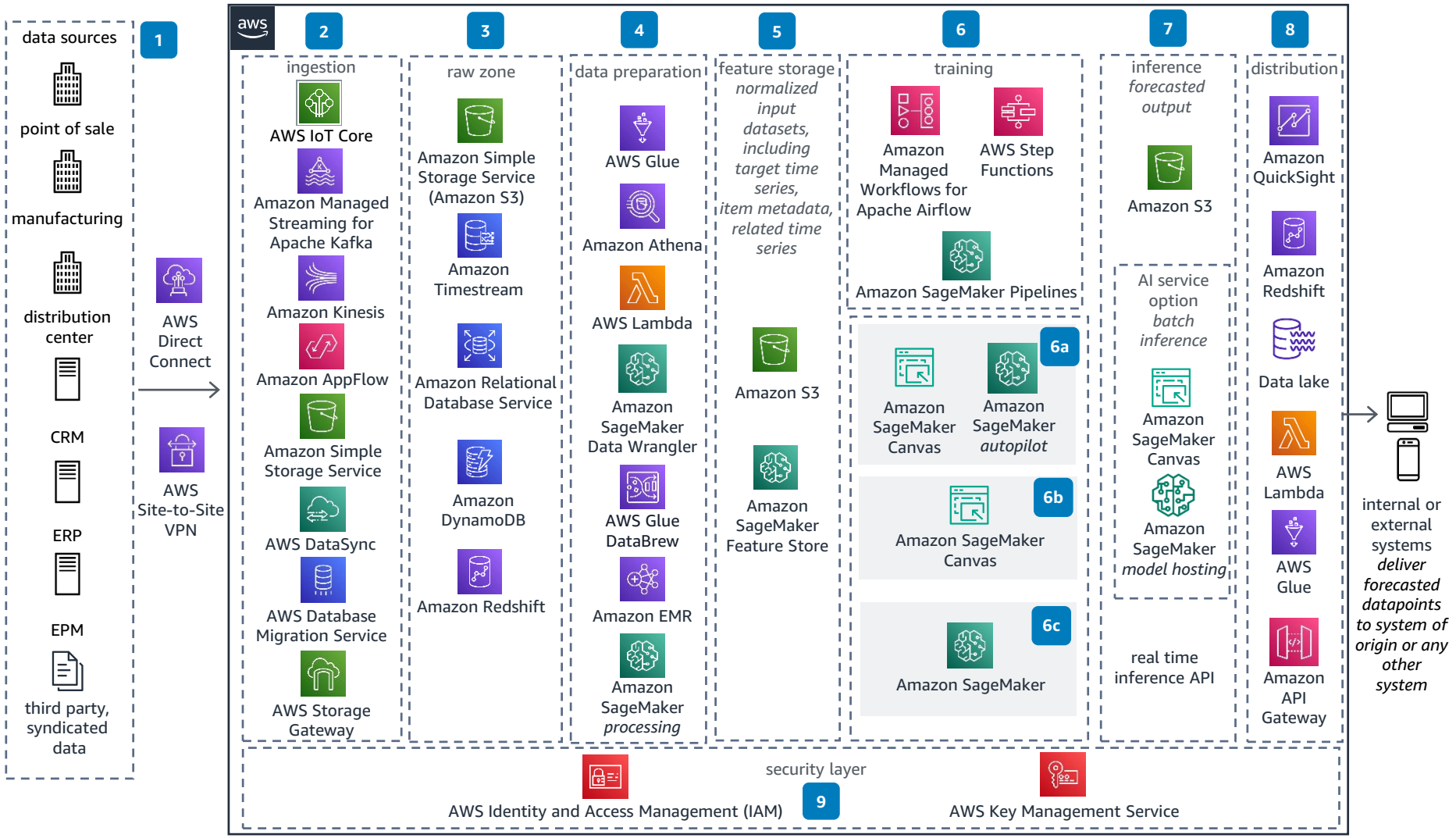


4 Data transformation and feature engineering of raw data produce accurate machine learning (ML) models. During this step customers can select and query the data from various sources, cleanse and explore the data, check for outliers and statistical bias by visualizing the data, analyze feature importance, perform feature engineering and finally export the prepared data. In this tier, **Amazon SageMaker Data Wrangler** is a low-code/no-code option. With the **Amazon SageMaker Data Wrangler** data selection tool, you can quickly select data from multiple data sources such as **Amazon Simple Storage Service** (Amazon S3), **Amazon Athena**, **Amazon Redshift**, **AWS Lake Formation**, Snowflake, and Databricks Delta Lake; transform by using over 300 pre-configured data transformations; and exploration and visualize data. Another low-code option is **AWS Glue DataBrew**, which provides a visual data preparation tool that can help clean and normalize data to prepare for machine learning. Customers can also use **Amazon EMR** and **Amazon SageMaker** processing jobs, which are advanced options for data transformation and feature engineering. **AWS Glue** can also filter, aggregate, and reshape data. **Amazon Athena** can be used to perform federated queries on multiple relational, non-relational, object, and custom data sources. **AWS Lambda** can be used for data transformation.

5 Prepared data is placed into special datasets on **Amazon S3** and **Amazon SageMaker Feature Store**, ready to be ingested for training.

AWS Reference Architecture

Training and Tuning



6 In this step, training and hyperparameter tuning take place. Customers can use orchestrators like **Amazon Step Functions**, **Amazon Managed Workflow for Apache Airflow**, or **Amazon SageMaker Pipeline** to help orchestrate the end-to-end ML process.

6a Training can be performed in **Amazon SageMaker Canvas** or **Amazon SageMaker Autopilot** as low-code/no-code options. Customers can simply browse the data and select the target, then **Amazon SageMaker** generates a model. Additionally, **Amazon SageMaker** performs hyperparameter tuning according to the defined objective metrics. This option is perfect for business analysts and data analysts.

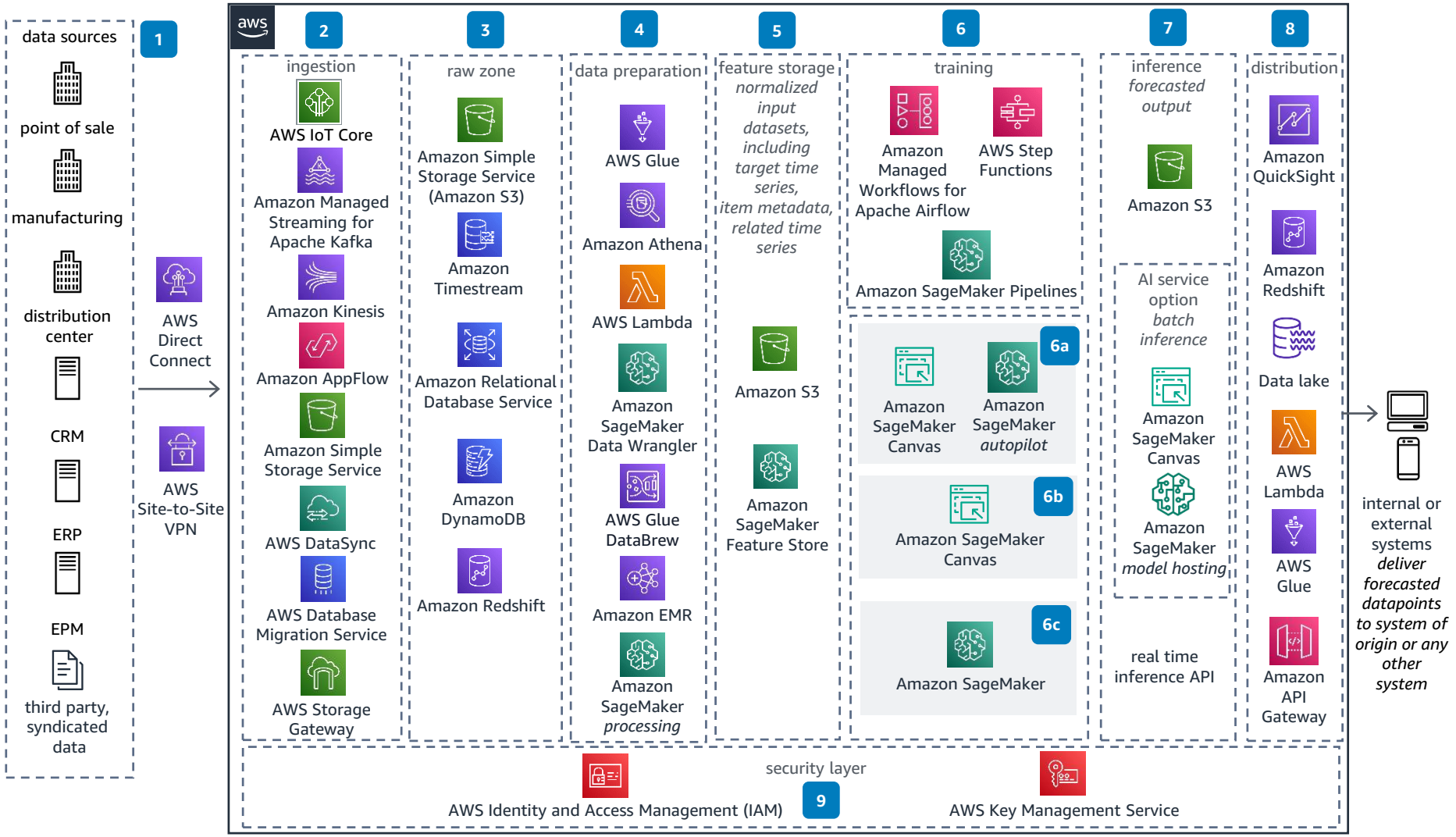
6b Optionally, customers can also use **Amazon SageMaker Canvas**, a fully managed service. Customers can use neural network algorithms like CNN-QR or DeepAR+ when an item's metadata dataset is available, or prophet, ARIMA, or ETS. Customers can use RETAIL data domain options available in **SageMaker Canvas**. This option is great for data engineers and developers.

6c **Amazon SageMaker** is an advanced option for developers and data scientists who can use build-in algorithms like DeepAR or AutoML libraries such as AutoGluon, or import existing models and perform transfer learning. Customers can use advanced options like data parallelism and model parallelism, then use built-in features like **Amazon SageMaker Clarify** for bias and **SageMaker Model Monitor** for monitoring.

Reviewed for technical accuracy September 26, 2024

AWS Reference Architecture

Prediction and Consumption



7 The output of the prior phase is a model that can be deployed for an ML inference to predict the forecast. There are two options, depending on the use case: Batch inference for asynchronous forecasting or real-time inference that leverages real-time API. For batch inference, the forecast is placed in **Amazon S3** as a CSV file. For real-time inference, the model is hosted in **SageMaker** or **SageMaker Canvas** and served as an API for business applications to invoke and get predictions.

8 The forecasted output can be consumed and distributed through various channels like **Amazon QuickSight** dashboards for business intelligence and analysis, exposed as an API using **Amazon API Gateway** for consumption, or pushed to a data lake or data warehouse like **Amazon Redshift**. Once the distribution channel is set up, the data can either be consumed by the source systems like ERP, EPM, or CRM, or by other internal or external applications. The entire process is iterative, where the latest business and customer data is fed in continuously and the model is retrained to ensure accuracy is maintained.

9 Security: During the integration of various AWS services, **AWS Identity and Access Management** (IAM) roles are used to ensure least privileged access and **AWS Key Management Service** (AWS KMS) is used to ensure data encryption.

Reviewed for technical accuracy September 26, 2024

AWS Reference Architecture