AWS Architecting and SysOps

Applications Integration and Analytics

June-July 2019

Contents

Application Integration
Analytics with Kinesis

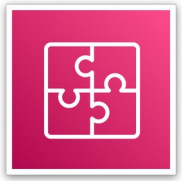
Application Integration

Pub-Sub, Queue and Workflow Services in the cloud

Application Integration

Amazon Simple Notification Service

Amazon Simple Queue Service



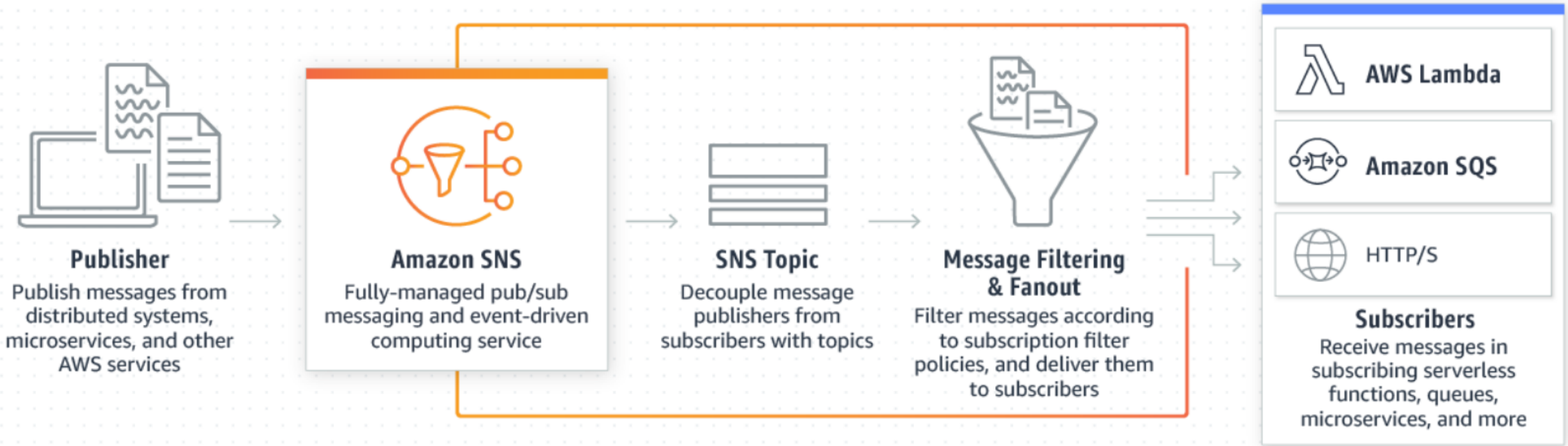
Amazon Simple Workflow Service

Simple Notification Service (SNS)

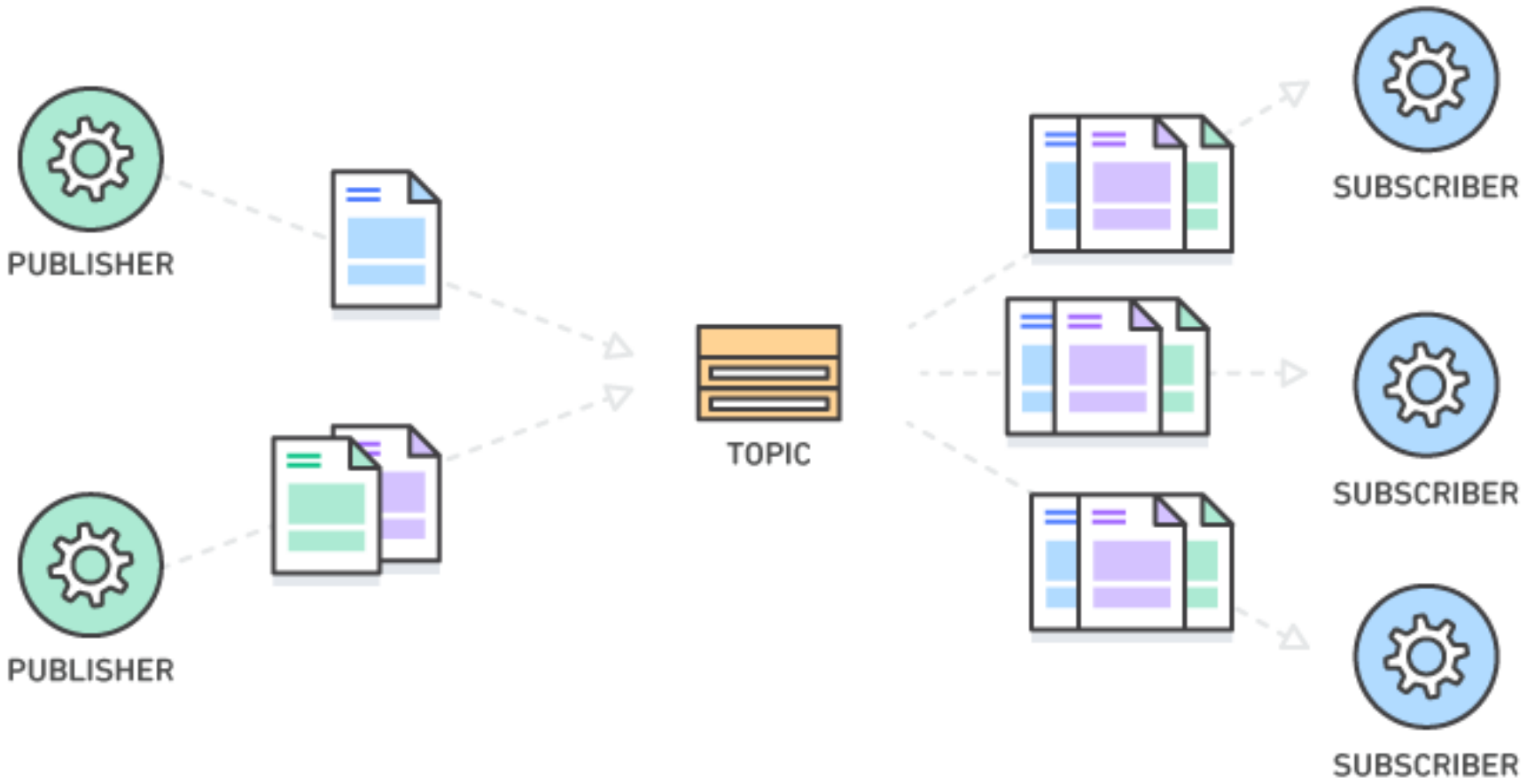
- An event-driven subscriber messaging web service that coordinates and manages the delivery and sending of messages to subscribing endpoints or clients
- Highly available, durable, secure, and fully managed
- Easy to setup, operate and send reliable communications
- Provides topics for high-throughput, push-based, many-to-many messaging
- An SNS topic is a logical access point which acts as a communication channel
- A topic lets you group multiple endpoints (such as Lambda, SQS, HTTP/S, or an email address).
- Native integration with a wide variety of AWS event sources (including EC2, S3, and RDS) and AWS event destinations (including SQS, and Lambda)

SNS topic

- SNS topic: logical access point which acts as a communication channel
- Message filtering: to receive only a subset of the message
- Publishers and subscribers have an asynchronous communication



SNS topic



SNS

- In SNS, there are two types of clients—publishers and subscribers—also referred to as producers and consumers
- Publishers communicate asynchronously with subscribers by producing and sending a message to a topic
- Subscribers (web servers, email addresses, SQS queues, Lambda functions) consume or receive the message or notification over one of the supported protocols (like SQS, HTTP/S, email, SMS, Lambda) when they are subscribed to the topic
- Using SNS topics, your publisher systems can distribute messages to a large number of subscriber endpoints for parallel processing, including SQS queues, Lambda functions, and HTTP/S webhooks
- Additionally, SNS can be used to distribute notifications to end users using mobile push, SMS, and email

SNS features

- Message filtering
  - The subscriber can create a filter policy, so that it only gets the notifications it is interested in, as opposed to receiving every single message posted to the topic
- Message fanout
  - Occurs when a message is sent to a topic and then replicated and pushed to multiple endpoints
  - This allows parallel asynchronous processing
- Message encryption
  - SNS provides encrypted topics to protect your messages from unauthorized and anonymous access
  - When you publish messages to encrypted topics, Amazon SNS immediately encrypts your messages. The encryption takes place on the server using KMS service.
  - Messages are stored in encrypted form, and decrypted as they are delivered to subscribing endpoints
- Mobile notifications
  - Fanout mobile push notifications to mobile devices, including SMS and email messages

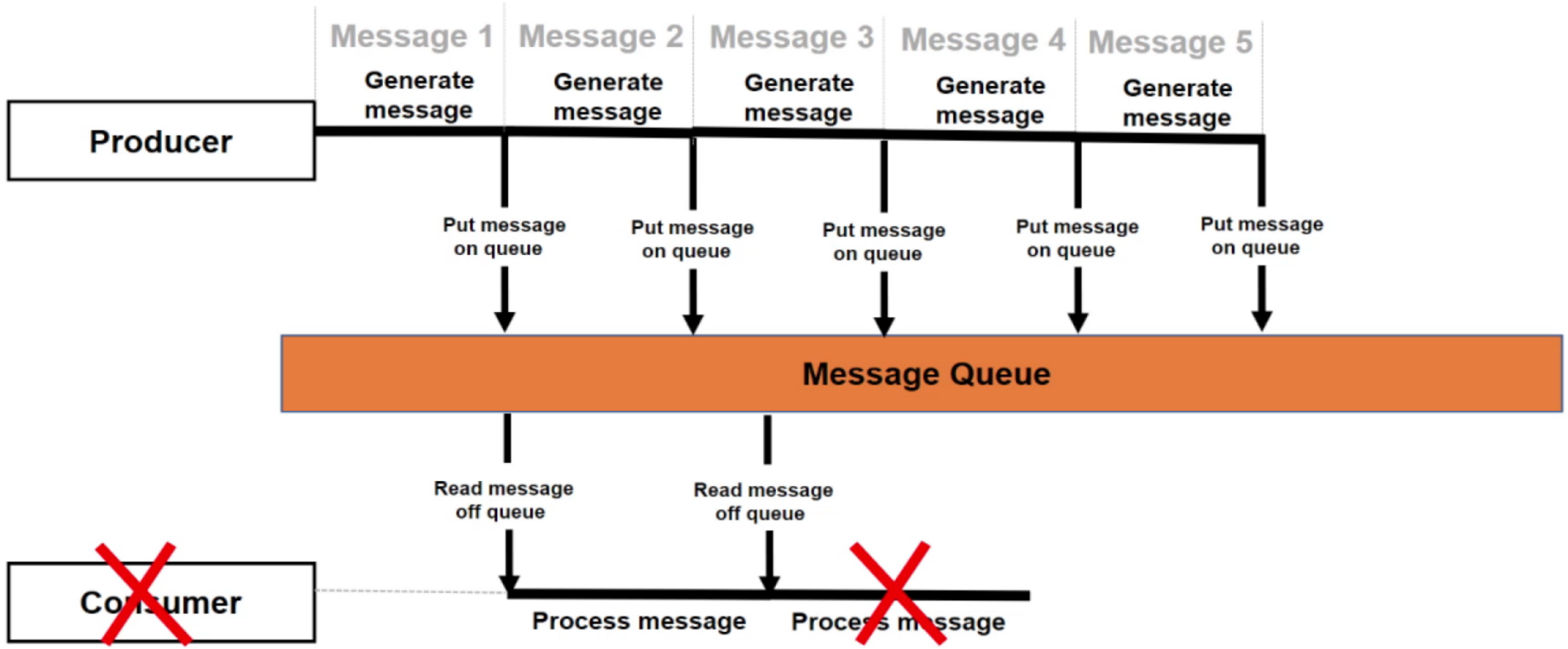
Simple Queue Service (SQS)

- A fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications
- SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work
- Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available
- SQS offers two types of message queues:
  - Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery
  - FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent

Message queue

- A named destination to which a message can be sent
  - A message can be up to 256KB
  - We put and take messages in / out the queue
- Enables you to create an asynchronous process

- An asynchronous process is a loosely coupled system
- ✓ More fault tolerant
- ✓ More scalable
- ➢ You can add and remove producers and consumers as your application requires



SQS queue types: Standard Queue

- ✓Unlimited Throughput
  - Standard queues support a nearly unlimited number of transactions per second (TPS) per API action
- ✓At-Least-Once Delivery
  - A message is delivered at least once, but occasionally more than one copy of a message is delivered
- ✓Best-Effort Ordering
  - Occasionally, messages might be delivered in an order different from which they were sent

- ✓Send data between applications when throughput is important:
  - Decouple live user requests from intensive background work: let users upload media while resizing or encoding it
  - Allocate tasks to multiple worker nodes: process a high number of credit card validation requests
  - Batch messages for future processing: schedule multiple entries to be added to a database

SQS queue types: FIFO Queue

- ✓High Throughput
  - FIFO queues support up to 300 messages per second. When you batch 10 messages per operation (maximum), FIFO queues can support up to 3,000 messages per second
- ✓Exactly-Once Processing
  - A message is delivered once and remains available until a consumer processes and deletes it. Duplicates aren't introduced into the queue
- ✓First-In-First-Out Delivery
  - The order in which messages are sent and received is strictly preserved

- ✓Send data between applications when the order of events is important:
  - Ensure that user-entered commands are executed in the right order
  - Display the correct product price by sending price modifications in the right order
  - Prevent a student from enrolling in a course before registering for an account

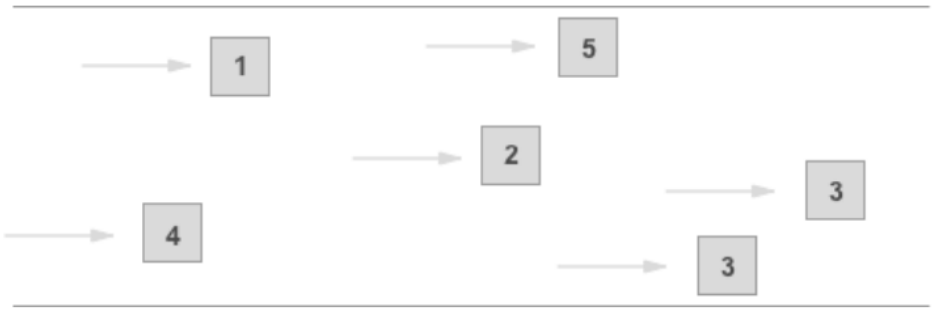
SQS queue types

Standard Queue

Unlimited Throughput: Standard queues support a nearly unlimited number of transactions per second (TPS) per API action.

At-Least-Once Delivery: A message is delivered at least once, but occasionally more than one copy of a message is delivered.

Best-Effort Ordering: Occasionally, messages might be delivered in an order different from which they were sent.



Send data between applications when the throughput is important, for example:

- Decouple live user requests from intensive background work: let users upload media while resizing or encoding it.
- Allocate tasks to multiple worker nodes: process a high number of credit card validation requests.
- Batch messages for future processing: schedule multiple entries to be added to a database.

FIFO Queue

High Throughput: FIFO queues support up to 300 messages per second (300 send, receive, or delete operations per second). When you batch 10 messages per operation (maximum), FIFO queues can support up to 3,000 messages per second. To request a limit increase, file a support request.

First-In-First-out Delivery: The order in which messages are sent and received is strictly preserved.

Exactly-Once Processing: A message is delivered once and remains available until a consumer processes and deletes it. Duplicates are not introduced into the queue.

Send data between applications when the order of events is important, for example:

- Ensure that user-entered commands are executed in the right order.
- Display the correct product price by sending price modifications in the right order.
- Prevent a student from enrolling in a course before registering for an account.

Simple Workflow Service (SWF)

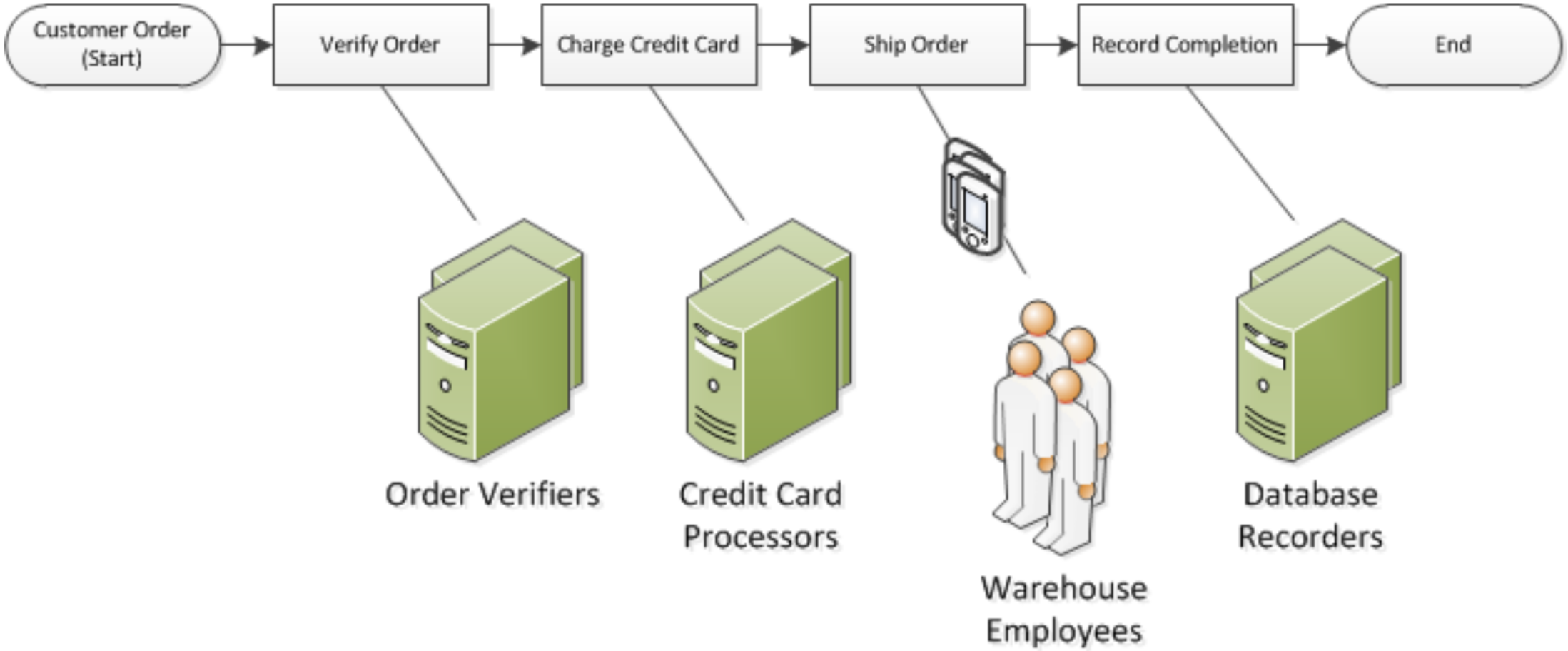
- Fully managed workflow service that makes it easy to coordinate work across distributed application components
- Enables applications for a range of use cases, including media processing, web application backends, business process workflows, to be designed as a coordination of tasks
- A task represents a logical unit of work that is performed by a component of your application
- Coordination of tasks involves managing execution dependencies, scheduling, and concurrency in accordance with the logical flow of the application
- Provides simple API calls that can be executed from your code and run on your EC2 instances, or on-prem machines
- Gives you full control over implementing and coordinating tasks without worrying about tracking their progress and maintaining their state

SWF Workflows and Activities

- Coordinates and manages the execution of activities that can be run asynchronously across multiple computing devices and that can feature both sequential and parallel processing
- When designing a workflow, you analyze your application to identify its component tasks that are normally represented by activities in SWF. The order in which activities are performed is determined by the workflow's coordination logic
- After the coordination logic and the activities have been designed, you register these components as workflow and activity types with SWF
- After registering your workflow type, you can run it as often you like
- A workflow execution is a running instance of a workflow and it can be started by any process or application, even another workflow execution
- Tasks in a workflow can be either sequential or parallel

A simple workflow example: E-commerce app

- Tasks in sequential order involving both people and automated processes



SWF Actors

- Workflow Starters
  - Any application that can initiate workflow executions
- Workflow Deciders
  - An implementation of a workflow's coordination logic (only one decider per workflow)
  - Whenever a change occurs during a workflow execution, such as the completion of a task, a decision task including the entire workflow history will be passed to a decider who then analyzes it to determine the next appropriate steps in the workflow execution
- Activity Worker
  - A single process (or thread) that performs the activities that are part of your workflow
  - Each activity worker polls for new activities until it receives one that it is processed to completion and then reports to SWF that the task was completed and provides the result. Then it polls for a new one.
  - All activity workers associated with a workflow execution continue in this way, processing tasks until the workflow execution itself is complete

SWF Tasks

- Activity task or activity
  - Contains all the information that the activity worker needs to perform its function
- Lambda task
  - Executes a Lambda function instead of a traditional SWF activity
- Decision task
  - Tells a decider that the state of the workflow execution has changed so that the decider can determine the next activity that needs to be performed
  - SWF schedules a decision task when the workflow starts and whenever the state of the workflow changes, such as when an activity task completes
  - Each decision task contains a paginated view of the entire workflow execution history
  - The decider analyzes the workflow execution history and responds back to SWF with a set of decisions that specify what should occur next in the workflow execution
  - Essentially, every decision task gives the decider an opportunity to assess the workflow and provide direction back to SWF

Application Integration documentation

- Amazon Simple Notification Service
  - https://docs.aws.amazon.com/sns/
- Amazon Simple Queue Service
  - https://docs.aws.amazon.com/sqs/
- AWS Simple Workflow Service
  - https://docs.aws.amazon.com/swf/


Analytics with Kinesis
Real-time and Streaming Data Analysis on AWS

Analytics

Amazon Kinesis

Kinesis

- Kinesis is a service that makes easy to collect, process, and analyze real-time, streaming data so you can get timely insights, in seconds or minutes, and react quickly to new information
- Offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application
- With Kinesis, you can ingest, buffer, and process real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications
- Kinesis enables you to process and analyze data as it arrives and respond instantly instead of have to wait until all your data is collected before the processing can begin
-

Kinesis Services

- Kinesis can handle any amount of streaming data and process data from hundreds of thousands of sources with very low latencies
- Services part of Kinesis shared some common core functionality but they are offered as separate services as they cover different use cases
  - Kinesis Data Streams
  - Kinesis Video Streams
  - Kinesis Data Firehose
  - Kinesis Data Analytics

Kinesis Data Streams (KDS)

- A fully managed, massively scalable and durable real-time streaming data service
  - Massive = up to terabytes per hour
- KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events
- The data collected is available in milliseconds to enable real-time analytics
- It manages the infrastructure, storage, networking, and configuration needed to stream your data at the level of your data throughput
- You do not have to worry about provisioning, deployment, ongoing-maintenance of hardware, software, or other services for your data streams
- In addition, Kinesis data streams synchronously replicates data across three facilities in an region, providing high availability and data durability.

Kinesis Data Streams (KDS)

Input

Capture and send data to Amazon Kinesis Data Streams

Amazon Kinesis Data Streams

Ingests and stores data streams for processing

- Amazon Kinesis Data Analytics
- Spark on EMR
- Amazon EC2
- AWS Lambda

Build custom, real-time applications using Amazon Kinesis Data Analytics, stream processing frameworks like Apache Spark, or your code running Amazon EC2 or AWS Lambda

Output

Analyze streaming data using your favorite BI tools

Kinesis Video Streams

- Makes it easy to securely stream video from connected devices to AWS for analytics, machine learning, and other processing
- Automatically provisions and elastically scales all the infrastructure needed to ingest streaming video data from millions of devices
- It also durably stores, encrypts, and indexes video data in your streams, and allows you to access your data through easy-to-use APIs
- Easily stream live and recorded video from your Kinesis video streams to your browser or mobile application using Kinesis Video Streams' HTTP Live Streaming (HLS) capability.
- Allows you to quickly build applications that take advantage of computer vision and video analytics through integration with Amazon Recognition Video, and libraries for ML frameworks such as Apache MxNet, TensorFlow, and OpenCV

Kinesis Video Streams

Input

Camera devices securely stream video to AWS using the Kinesis Video Streams SDK

Kinesis Video Streams

Ingests, durably stores, encrypts, and indexes video streams for real-time and batch analytics

Amazon Rekognition Video

Amazon SageMaker

MxNet / TensorFlow

HLS-based video playback

Custom video processing

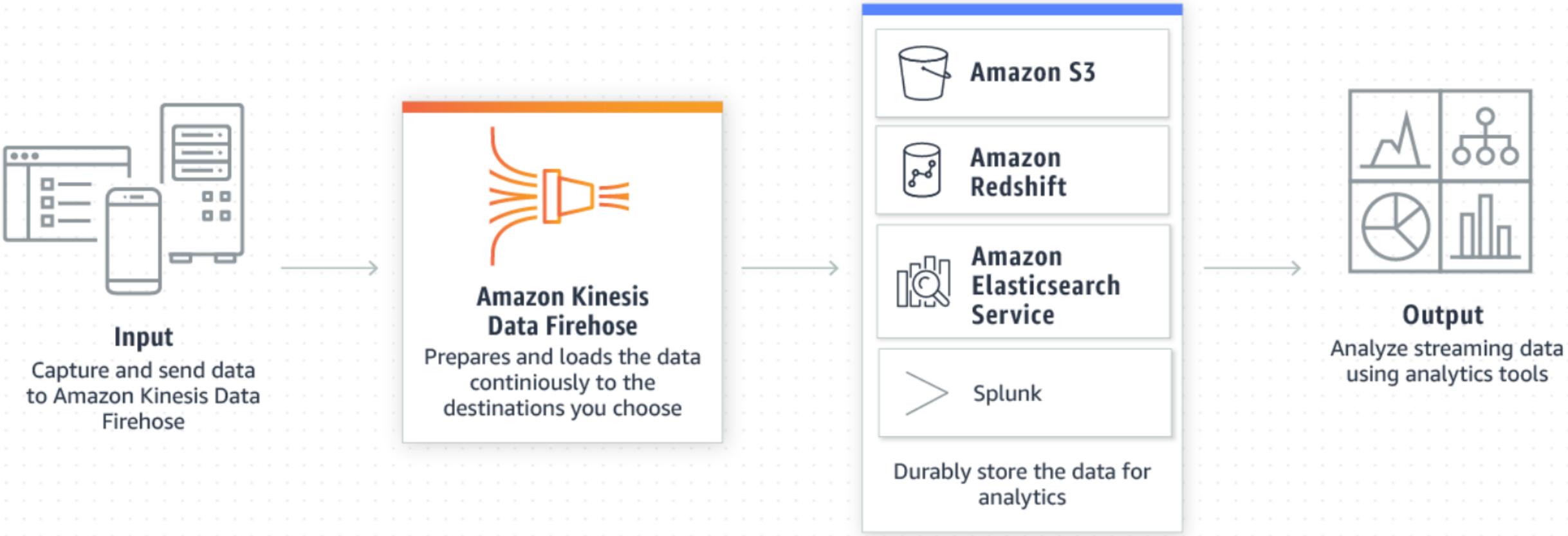
Output

Real-time and batch-driven machine learning applications, video processing and playback services use Kinesis Video Streams API to access and retrieve indexed video

Kinesis Data Firehose

- It is the easiest way to reliably load streaming data into data stores and analytics tools
- It can capture, transform, and load streaming data into S3, Redshift, and Elasticsearch Service, enabling near real-time analytics with existing business intelligence tools and dashboards you're already using today
- It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration
- It can also batch, compress, transform, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security

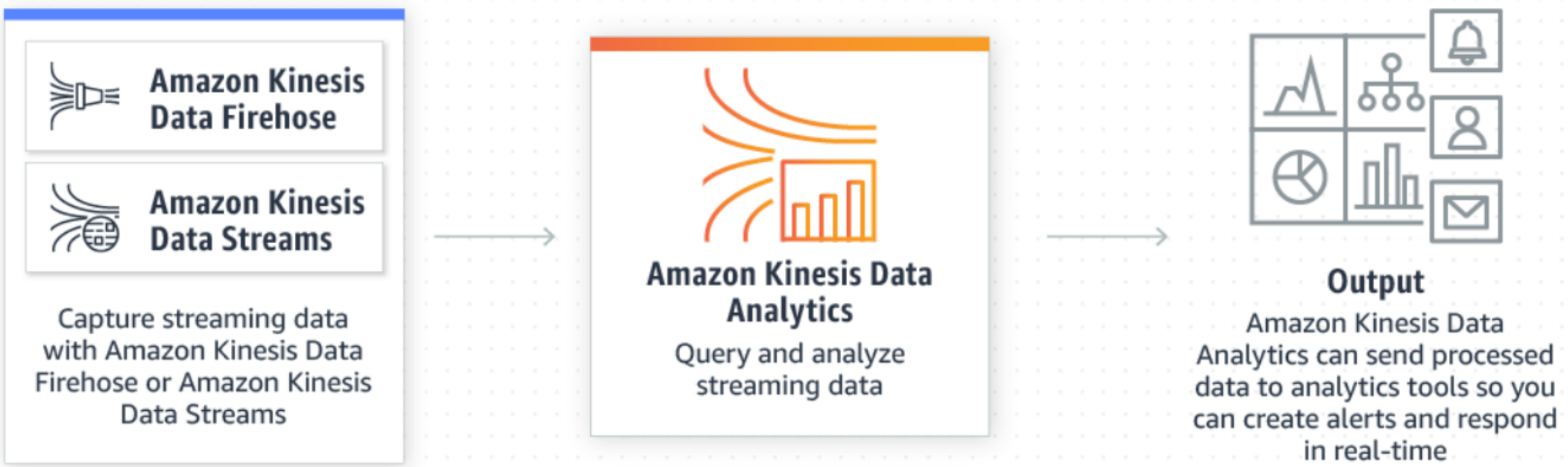
Kinesis Data Firehose



Kinesis Data Analytics

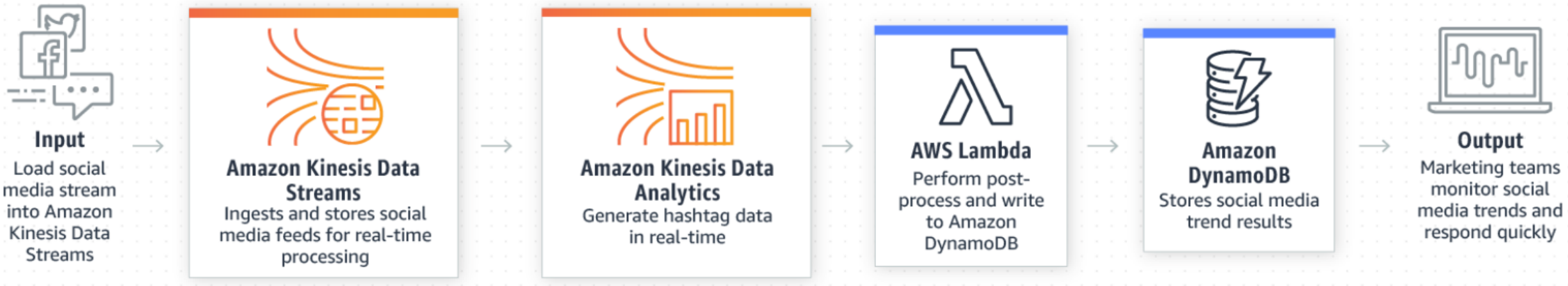
- It is the easiest way to analyze streaming data, gain actionable insights, and respond to your business and customer needs in real time
- It reduces the complexity of building, managing, and integrating streaming applications with other AWS services
- SQL users can easily query streaming data or build entire streaming applications using templates and an interactive SQL editor
- Java developers can quickly build sophisticated streaming applications using open source Java libraries and AWS integrations to transform and analyze data in real-time
- It takes care of everything required to run your real-time applications continuously and scales automatically to match the volume and throughput of your incoming data

Kinesis Data Analytics



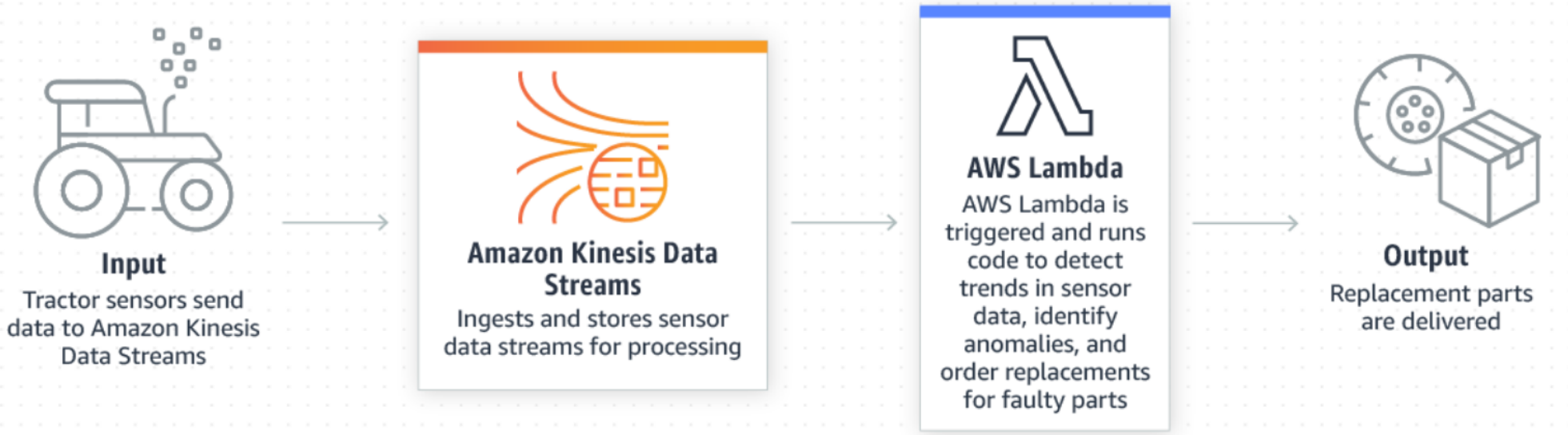
Kinesis use cases

- Build Real-time applications
  - Application monitoring, fraud detection, and live leader-boards
  - You can ingest streaming data using Kinesis Data Streams, process it using Kinesis Data Analytics, and emit the results to any data store or application using Kinesis Data Streams with millisecond end-to-end latency
  - This can help you learn about what your customers, applications, and products are doing right now and react promptly.



Kinesis use cases

- Analyze IoT device data
  - You can use Amazon Kinesis to process streaming data from IoT devices such as consumer appliances, embedded sensors, and TV set-top boxes
  - You can then use the data to send real-time alerts or take other actions programmatically when a sensor exceeds certain operating thresholds
  - Use our sample IoT analytics code to build your application. No need to start from scratch



Kinesis documentation

- Amazon Kinesis
  - https://docs.aws.amazon.com/kinesis/