Module 16:
Azure Service Bus

Azure Service Bus

Azure Service Bus

- Microsoft Azure Service Bus is a fully managed enterprise message broker with message queues and publish-subscribe topics.

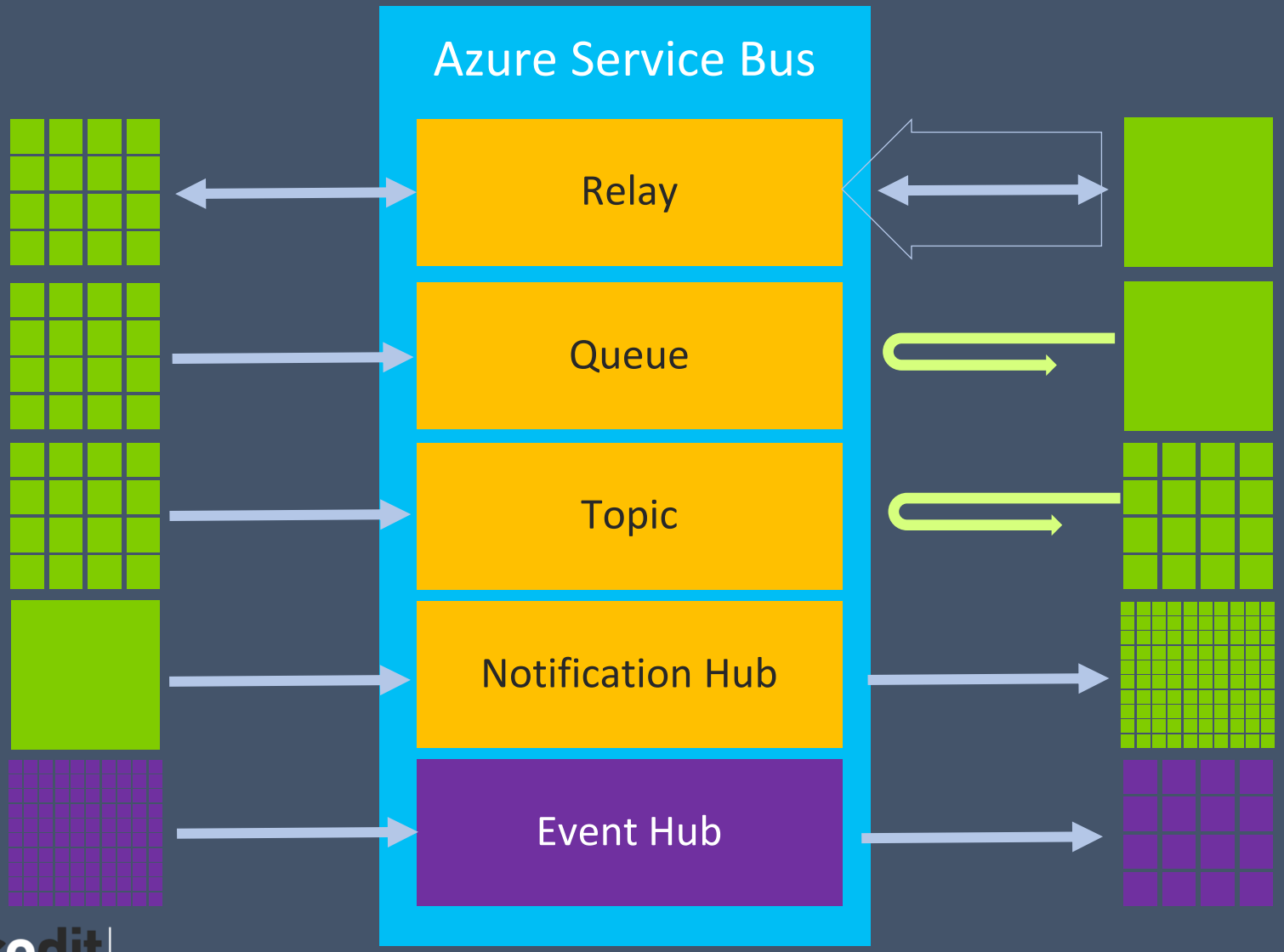
- Reliable cloud messaging as a service (MaaS)

- Benefits:
  - Load-balancing work across competing workers
  - Safely routing and transferring data and control across service and application boundaries
  - Coordinating transactional work that requires a high-degree of reliability

Azure Service Bus

- Enables your applications to interact in several different ways
- Uses a namespace as a scoping container for all messaging components
- The three communication mechanisms are:
  - Queues
  - Topics
  - Relays

Azure Service Bus



Request – Response services

High-Reliability Async Messaging
Sessions, Scheduled Delivery, etc.

Queues with rules-based pub/sub

Mobile push notification services
Millions of notification targets

partitioned event ingestion

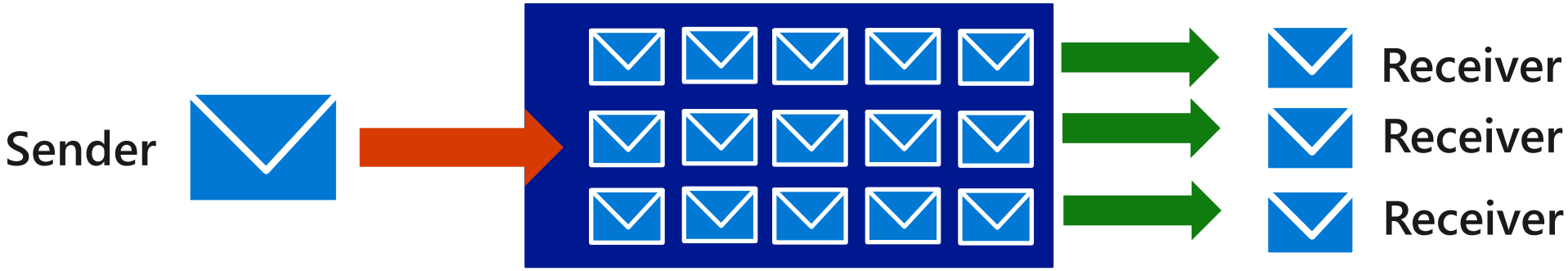
Queues

- Messages are sent to and received from queues
- Enables you to store messages until the receiving application is available to receive and process them
- Supports a brokered messaging communication model
- A general-purpose technology that can be used for a wide variety of scenarios

Message queue with messages

Topics and subscriptions

- Implements publish/subscribe (pub-sub) model
  - Receivers subscribe to a topic, and they can even filter down by interest
  - A sender publishes messages to the topic
  - Asynchronously, receivers get their own copy of the message
- Subscriptions are independent, which allows for many independent "taps" into a message stream



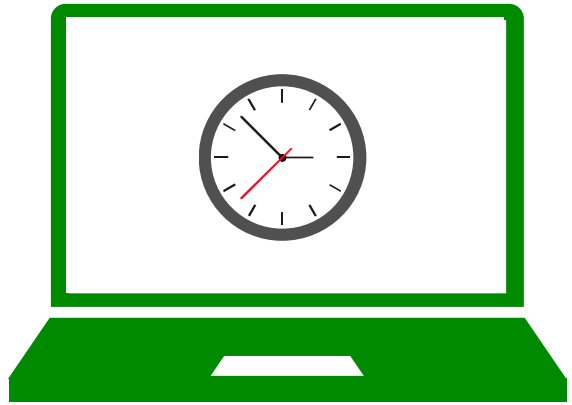
Topic with three subscriptions with messages

Messages, payloads, and serialization

- Messages carry multiple things
  - Metadata about the message itself (in key-value pairs)
  - Predefined Broker properties
  - The message binary payload
- Message payload is not visible to Service Bus at any point
  - Serializes as opaque, binary content
  - Can be deserialized by using client SDK libraries
  - Gives you the flexibility to explicitly define how you want to serialize content

Demo:
Azure Service Bus

- Topic
- Queue

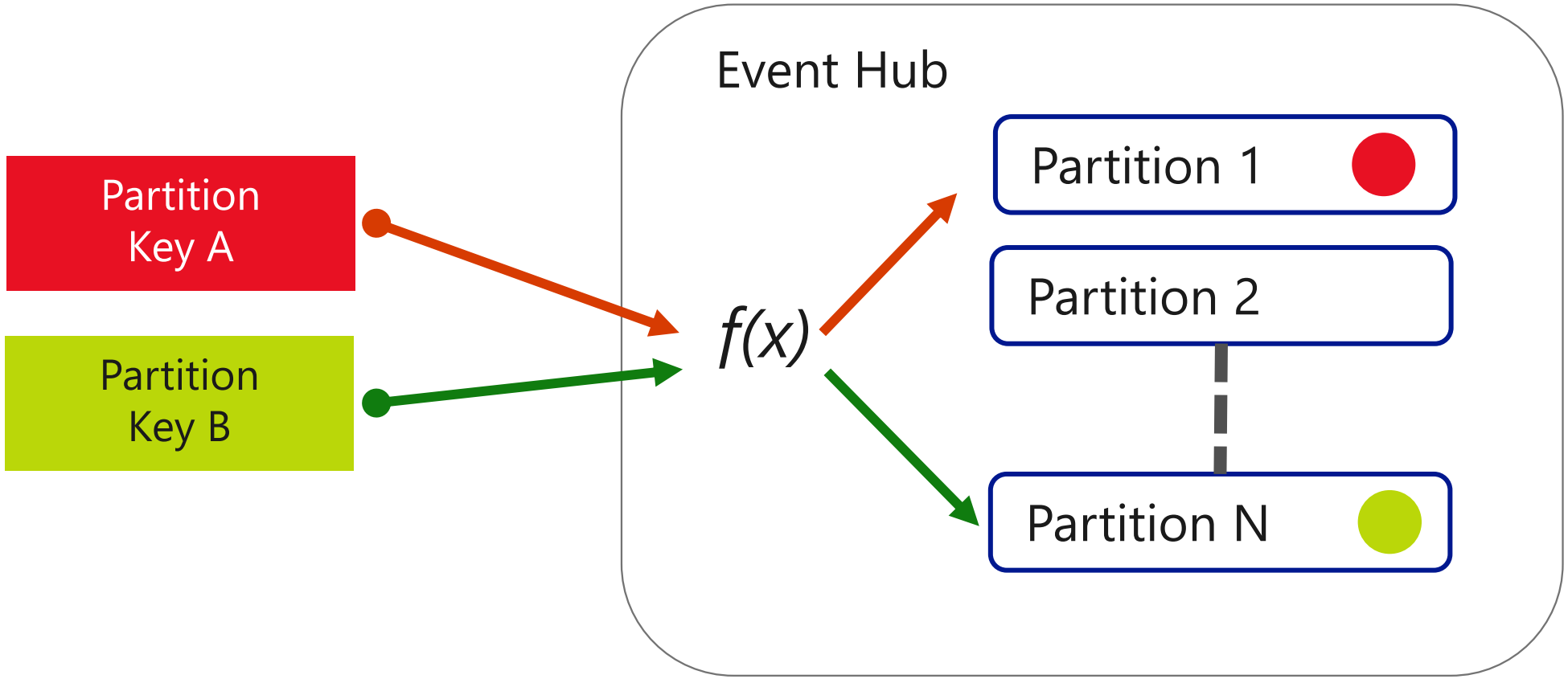


Azure Event Hub

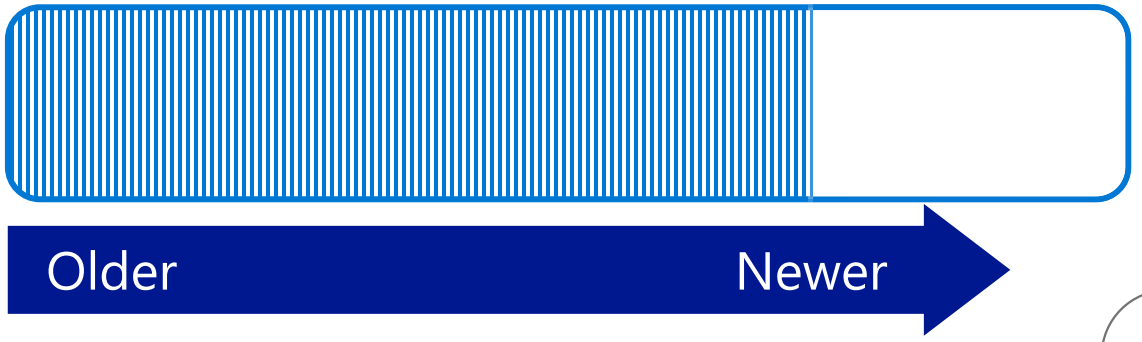
Azure Event Hubs

- Can process and store events, data, or telemetry produced by distributed software and devices
- Provide a distributed stream processing platform with low latency, and seamless integration with data and analytics services inside and outside of Azure
- Contain the following key components:
  - Event producers
  - Partitions
  - Consumer groups
  - Throughput units
  - Event receivers

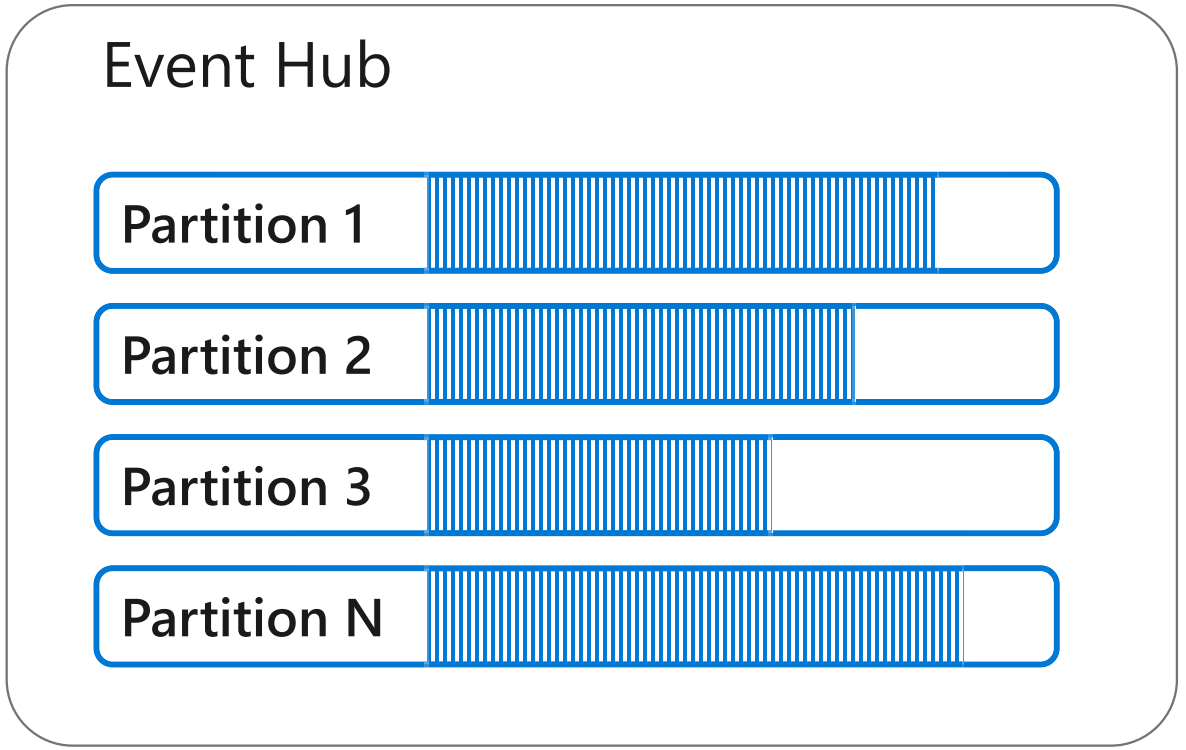
Event publishers



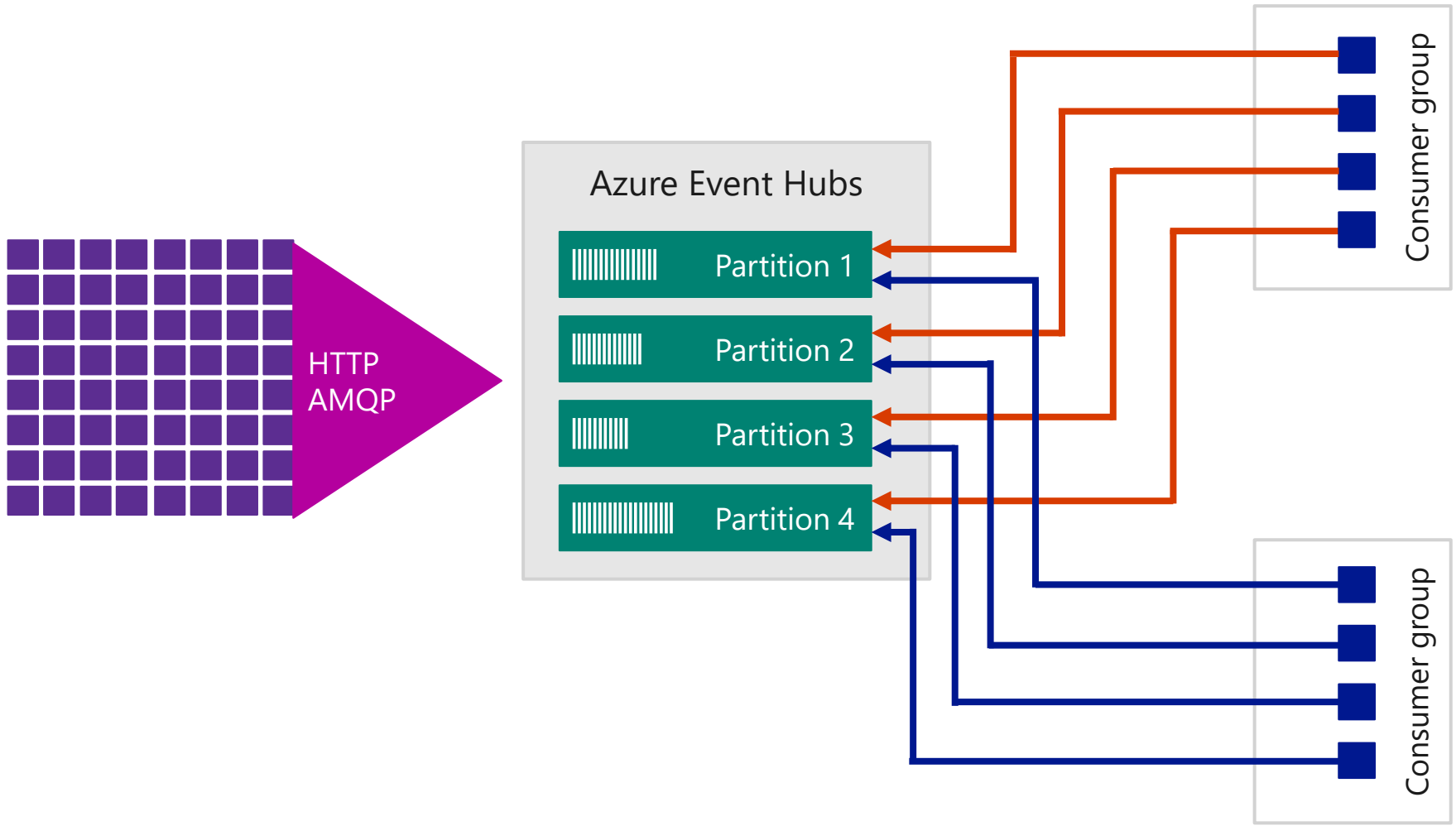
Partitions



A partition is an ordered sequence of events that is held in an event hub. As newer events arrive, they are added to the end of this sequence.



Consumer groups



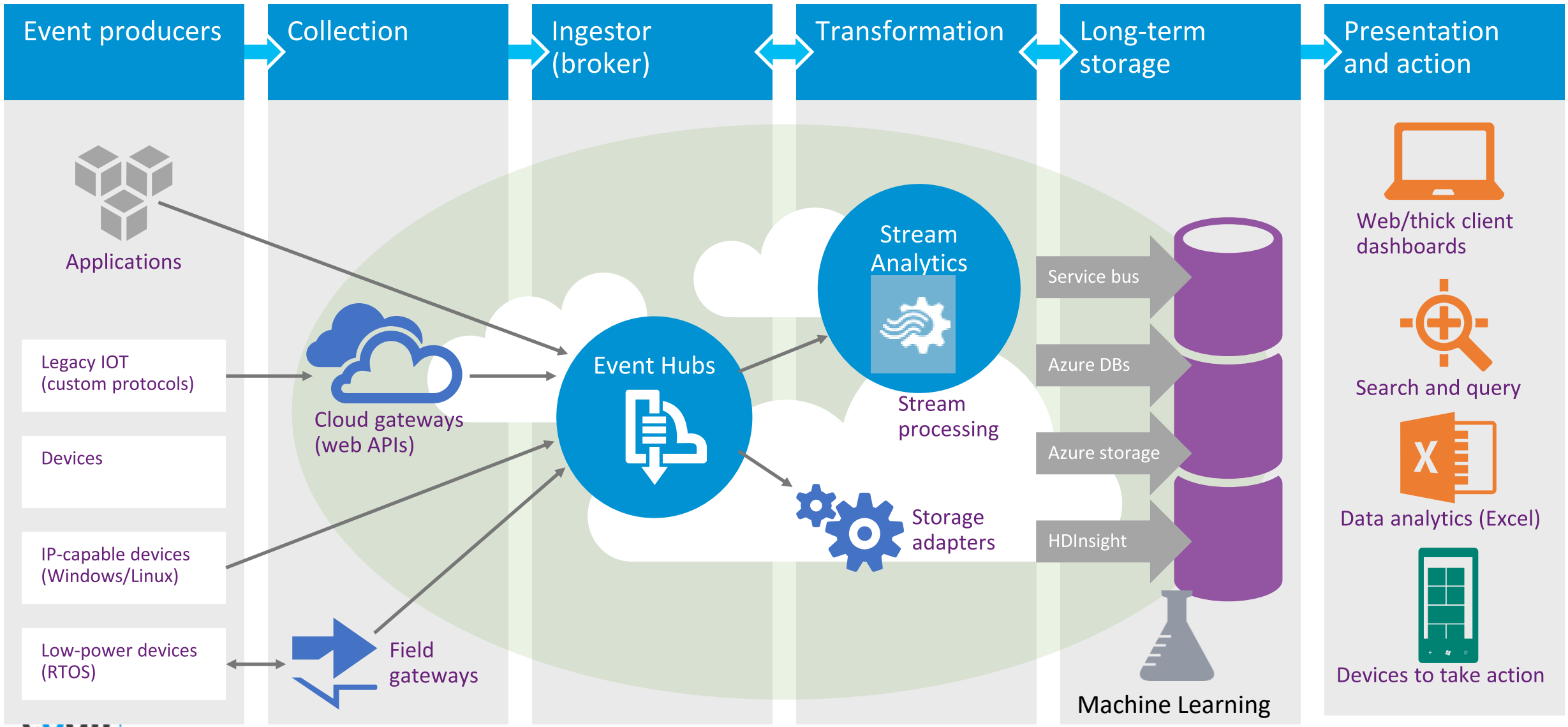
Capture

- Data can be automatically captured
  - Stored in Azure Blob storage or Azure Data Lake Storage
  - Capture-time or size intervals can be specified
- You can specify a window to control capturing
  - Must specify a minimum size and time configuration
  - First trigger encountered causes a capture operation
- Data is stored by using a naming convention:

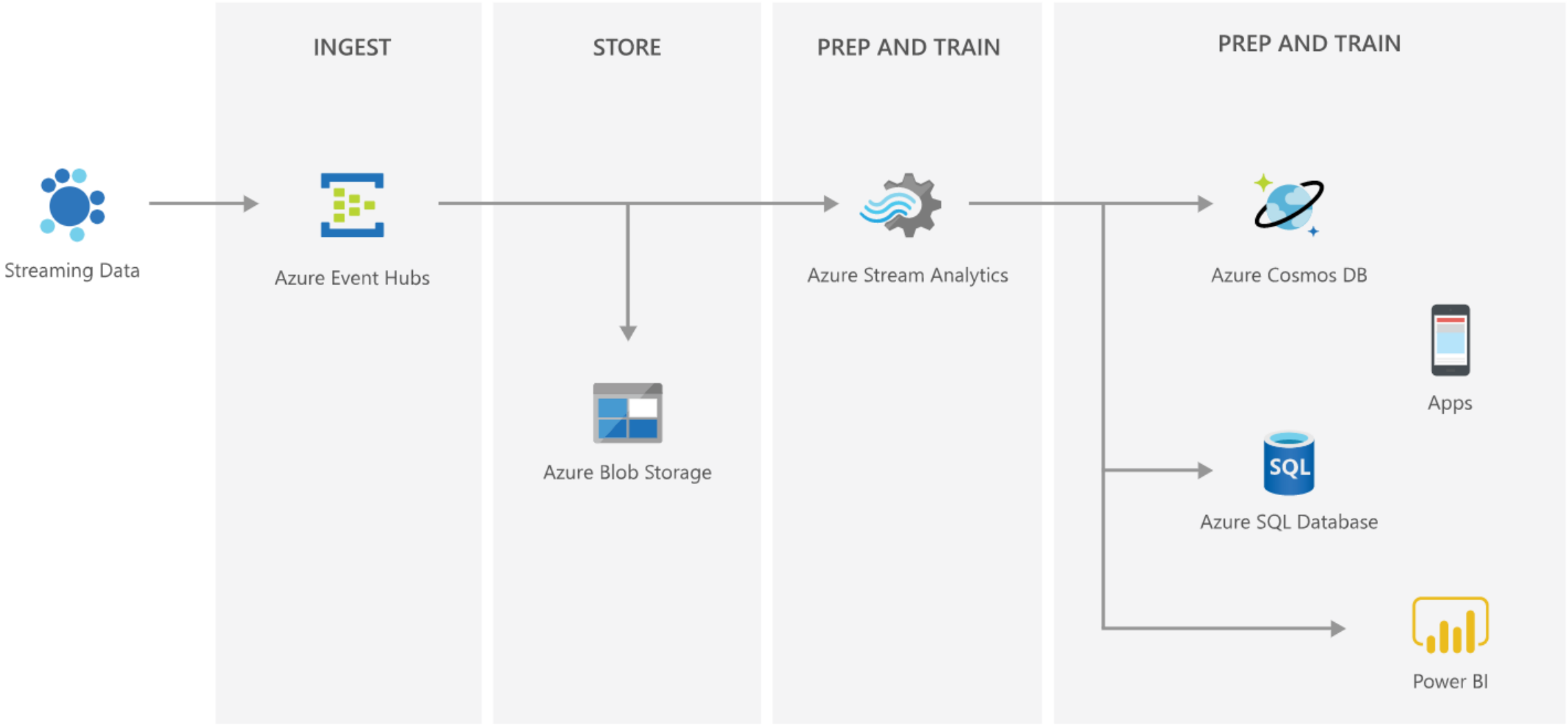
```
{Namespace}/{EventHub}/{PartitionId}/{Year}/{Month}/{Day}/{Hour}/{Minute}/{Second}
```

```
https://mystorageaccount.blob.core.windows.net/mycontainer/mynamespace/myeventhub/0/2017/12/08/03/03/17.avro
```

Event Hub


Event producers
Collection
Ingestor (broker)
Transformation
Long-term storage
Presentation and action
Applications
Legacy IOT (custom protocols)
Devices
IP-capable devices (Windows/Linux)
Low-power devices (RTOS)
Cloud gateways (web APIs)
Field gateways
Event Hubs
Stream Analytics
Stream processing
Storage adapters
Service bus
Azure DBs
Azure storage
HDInsight
Machine Learning
Web/thick client dashboards
Search and query
Data analytics (Excel)
Devices to take action

Use Case


INGEST
STORE
PREP AND TRAIN
PREP AND TRAIN
Streaming Data
Azure Event Hubs
Azure Blob Storage
Azure Stream Analytics
Azure Cosmos DB
Apps
SQL
Azure SQL Database
Power BI

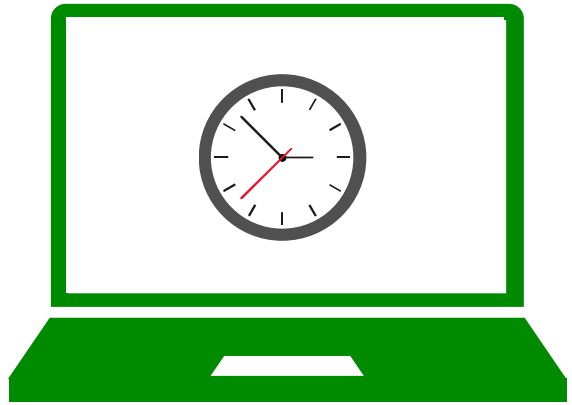
Events vs. messaging services

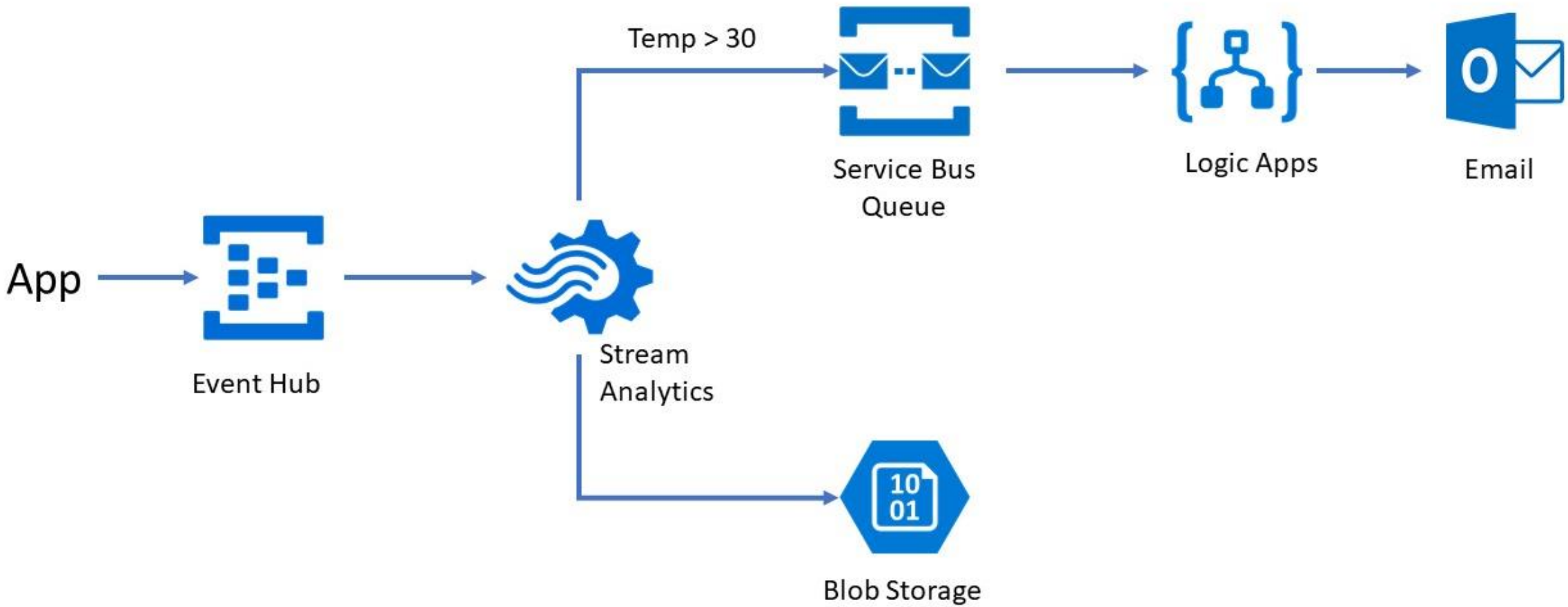
Service	Purpose	Type	When to use
Event Hubs	Big data pipeline	Event streaming (series)	Telemetry and distributed data streaming
Service Bus	High-value enterprise messaging	Message	Order processing and financial transactions

Comparing cloud messaging options

Requirement	Simple queuing	Big data streaming	Enterprise messaging
Product	Queue storage	Event Hubs	Service Bus
Supported advantages	• Communication within an app • Individual message • Simple and easy to use • Pay as you go	• Many messages in a Stream (think in MBs) • Ease of use and operation • Low cost • Fan in • Strict ordering • Works with other tools	• Instantaneous consistency • Strict ordering • Java Messaging Service • Non-repudiation and security • Geo-replication and availability • Rich features (such as deduplication and scheduling)
Weaknesses	• Ordering of messaging • Instantaneous consistency	• Server-side cursor • Only once	• Cost • Simplicity
Type	Serverless	Big data	Enterprise

Demo:
Azure Event Hub




App
Event Hub
Stream Analytics
Temp > 30
Service Bus Queue
Logic Apps
Email
10
01
Blob Storage