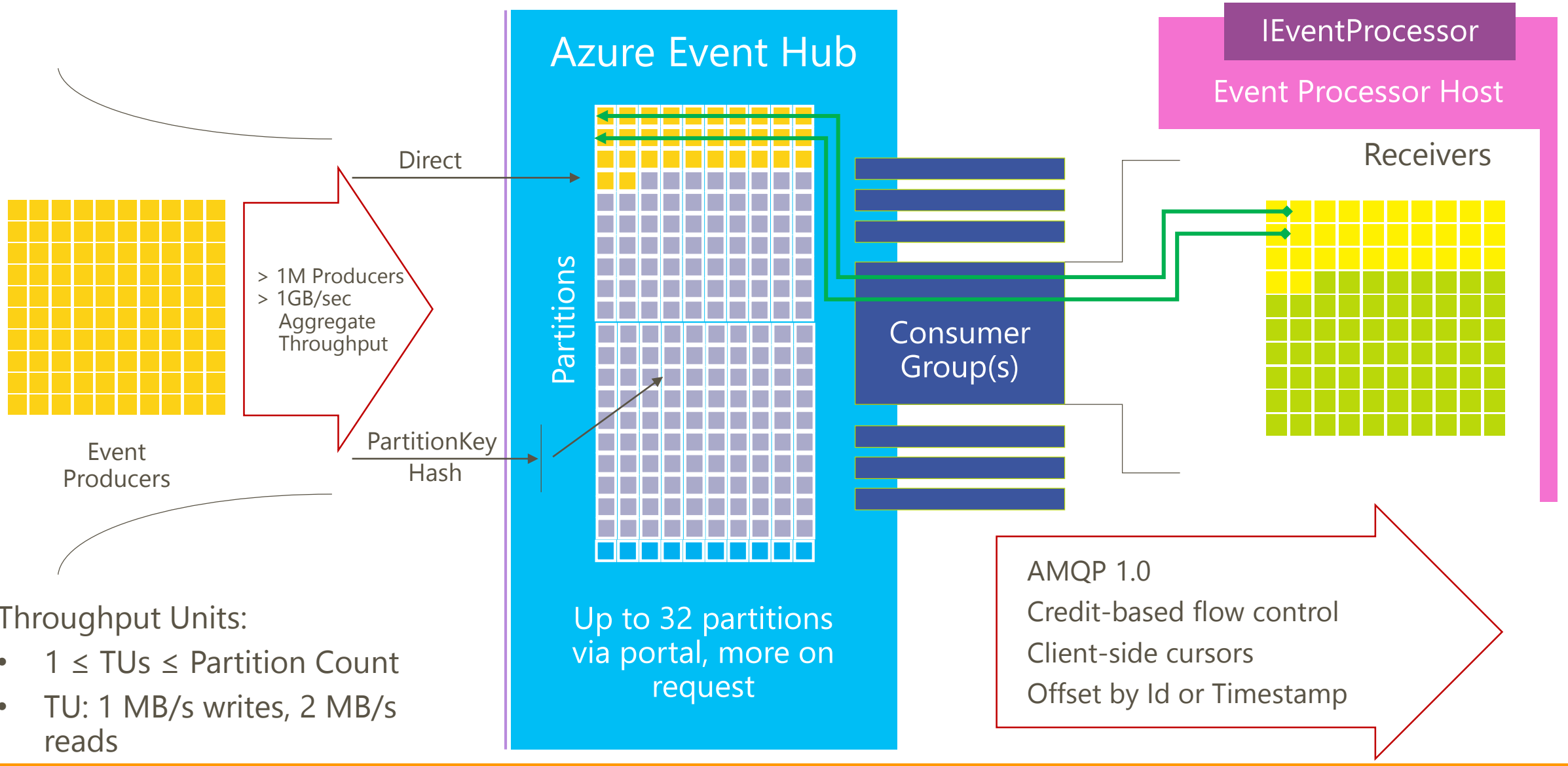
# Introducing Azure Event Hubs




Throughput Units:

- 1 ≤ TUs ≤ Partition Count
- TU: 1 MB/s writes, 2 MB/s reads

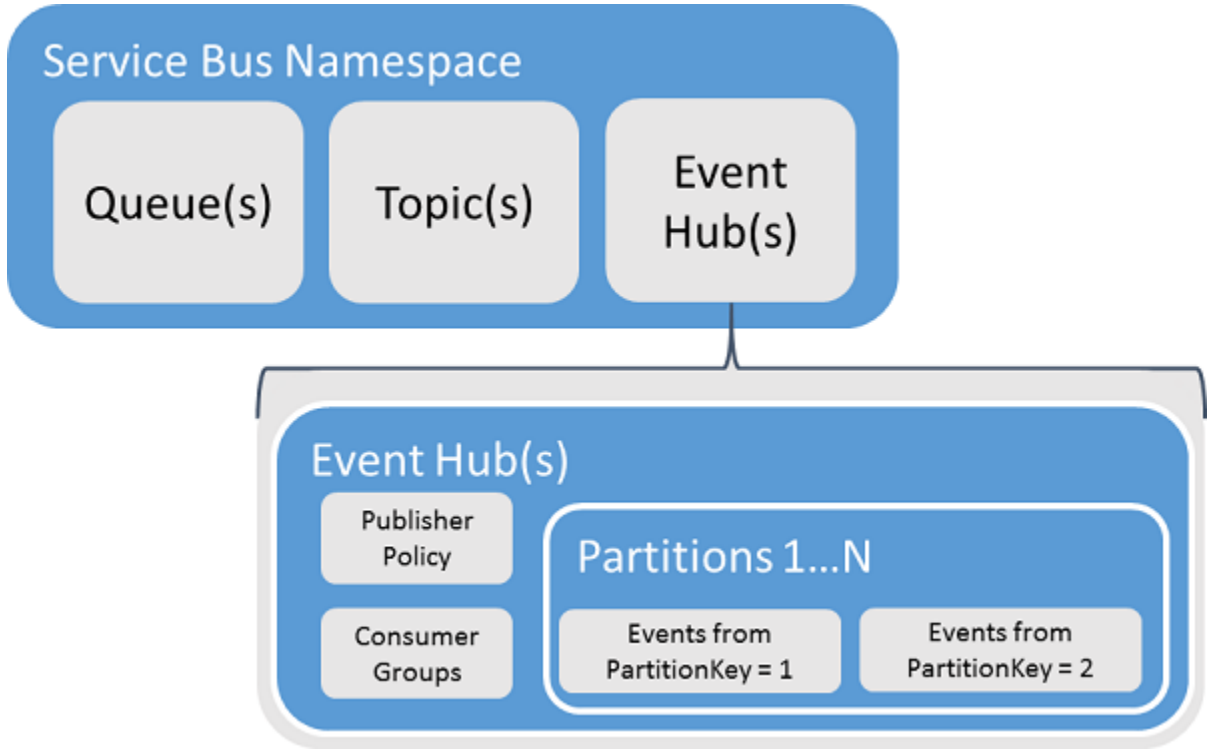
# Event Hubs - Overview

Event Hubs is a highly scalable ingestion system that can process millions of events per second, enabling your application to process and analyze the massive amounts of data produced by your connected devices and applications.


The Azure Event Hubs capability differs from Service Bus topics in that it is strongly biased towards high throughput and event processing scenarios.

# Event Hubs - Overview

An Event Hub is created at the namespace level in Service Bus, similar to queues and topics. Event Hubs uses AMQP and HTTP as its primary API interfaces.



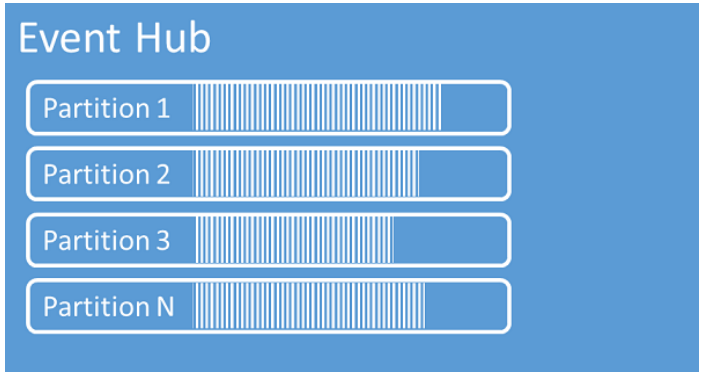
# Event Hubs - Partitions

Event Hubs provides message streaming through a partitioned consumer pattern.

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. A partition can be thought of as a "commit log."

The number of partitions is specified at the Event Hub creation time and must be between 8 and 32.

Segmentation of event stream for scale-out.



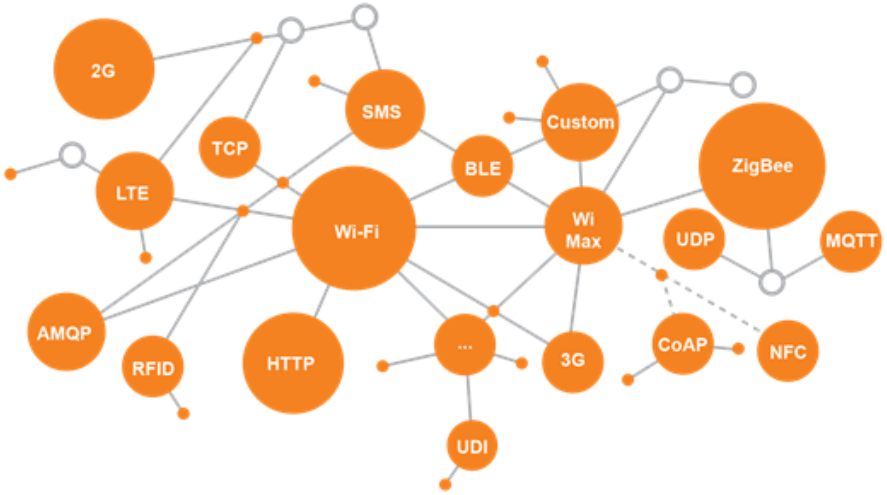
# Event Hubs - Event Data

In the Event Hubs context, messages are referred to as *event data*. Event data contains the body of the event, a user defined property bag, and various metadata about the event such as its offset in the partition and its number in the stream sequence. Partitions are filled with a sequence of event data (max 256kb).

You can publish an event via AMQP 1.0 or HTTP. Service Bus provides an EventHubClient class that you can use for publishing events to an Event Hub from .NET clients. For other runtimes and platforms, you can use any AMQP 1.0 client such as Apache Qpid.

# Event Hubs - What is AMQP 1.0?

Advanced Message Queuing Protocol (AMQP) is an efficient, reliable, wire-level messaging protocol (OASIS standard) that you can use to build robust, cross-platform, messaging applications. The protocol has a simple goal: to define the mechanics of the secure, reliable, and efficient transfer of messages between two parties.



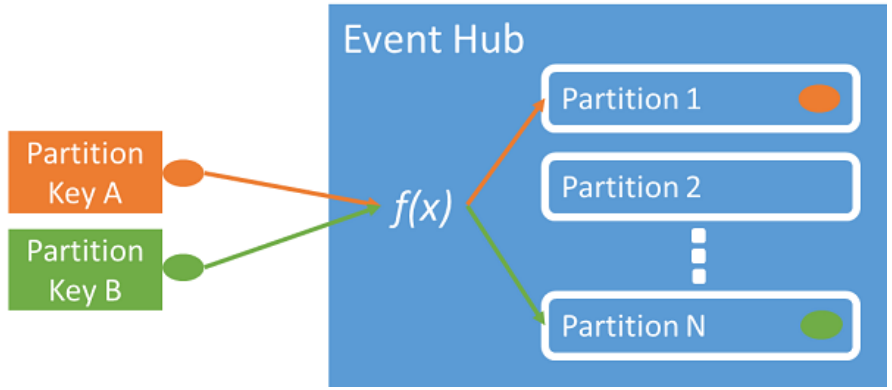
# Event Hubs - AMQP or HTTP ?

The choice to use AMQP or HTTP is specific to the usage scenario. AMQP requires the establishment of a persistent bidirectional socket in addition to transport level security (TLS) or SSL/TLS. This can be a costly operation in terms of network traffic, but only happens at the beginning of an AMQP session. HTTP has a lower initial overhead, but requires additional SSL overhead for every request. For publishers who frequently publish events, AMQP offers significant performance, latency, and throughput savings

# Event Hubs - Partition key

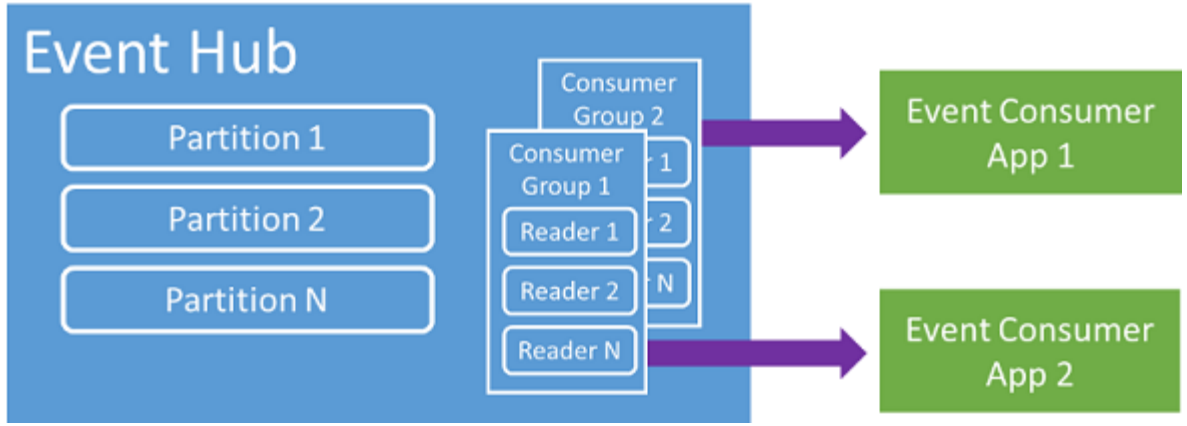
A partition key is a value that is used to map incoming event data into specific partitions for the purposes of data organization. The partition key is a sender-supplied value passed into an Event Hub. It is processed through a static hashing function, the result of which creates the partition assignment.

Event Hubs ensures that any and all events sharing the same partition key value are delivered in order, and to the same partition.



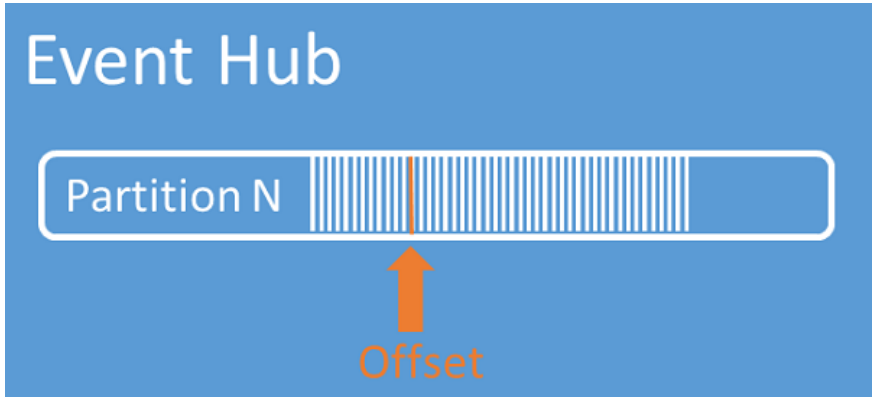
# Event Hubs – Consumer Groups

The publish/subscribe mechanism of Event Hubs is enabled through *consumer groups*. A consumer group is a view (state, position, or offset) of an entire Event Hub. Consumer groups enable multiple consuming applications to each have a separate view of the event stream, and to read the stream independently at its own pace and with its own offsets.



# Event Hubs - Stream Offset

An offset is the position of an event within a partition. You can think of an offset as a client-side cursor. The offset is a byte numbering of the event. This enables an event consumer (reader) to specify a point in the event stream from which they want to begin reading events. You can specify the offset as a timestamp or as an offset value.



# Event Hubs - Reading Events

In order to consume events from an Event Hub, a consumer must connect to a partition. Only a single reader should be active on a partition at any one time within a consumer group. Each event data instance contains important metadata such as the offset and sequence number that are used to facilitate checkpointing on the event sequence.

**EventData**

- Offset
- Sequence Number
- Body
- User Properties
- System Properties

# Event Hubs - Throughput Units

The throughput capacity of Event Hubs is controlled by throughput units. Throughput units are pre-purchased units of capacity. A single throughput unit includes the following:

Write: Up to 1MB per second or 1000 events per second.

Read: Up to 2MB per second.

Included retention: 84GByte/day (24h at full ingress rate)

Sending data above this amount results in a quota exceeded exception !

Throughput units are billed per hour and are purchased ahead of time

Up to 20 throughput units can be purchased for a Service Bus namespace

# Event Hubs vs Queues & Topics

## Patterns

Q&T : useful for Command Message and Request/Replay Message (response queue)

EH : useful for Event Messages

## Cursor

Q&T : on server side. Message consumed and deleted from queue, cursor to next available message

EH : on client side. Client can rewind on the stream and re-read same events (during their retention). Access partition by offset or timestamp

## Retention

Q&T : TTL at queue/topic level or message level

EH : max 7 days

# Event Hubs vs Queues & Topics

## Security & Authentication

### Q&T and EH

- SSL/TLS via HTTP(S) or AMQP(S)
- SAS (Secure Access Signature) for sending/receiving

### EH

- Publisher policy (SAS Token)
- Fine grained per device
- Revoke/Restore publisher

## Other

EH doesn't have dead lettering, transaction, ... to have higher throughput