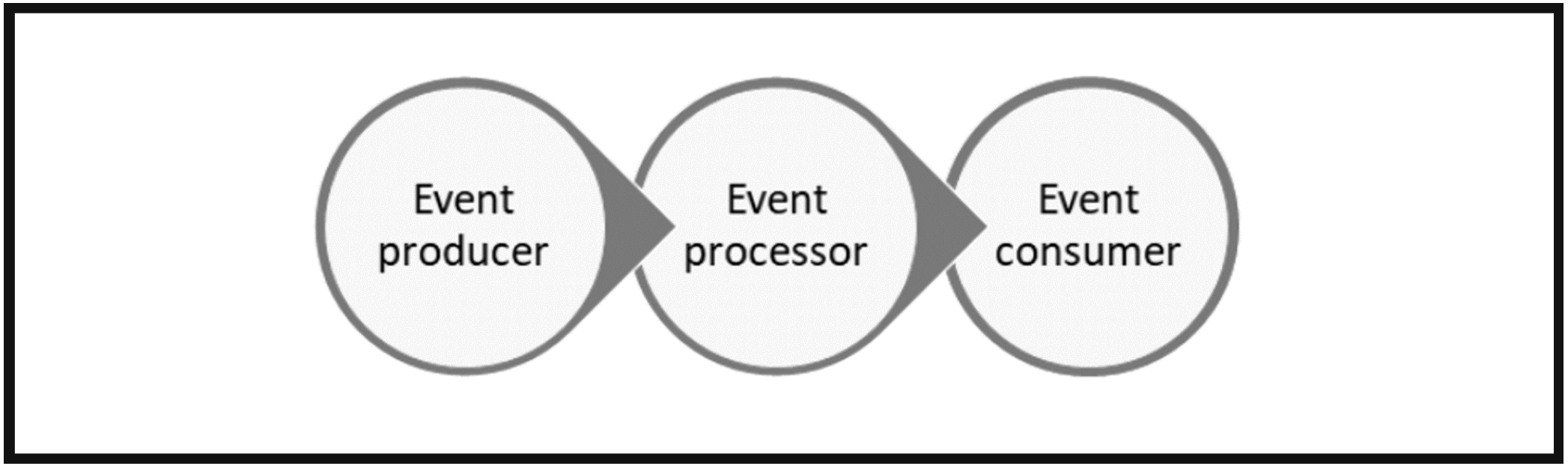
Azure Streaming Analytics

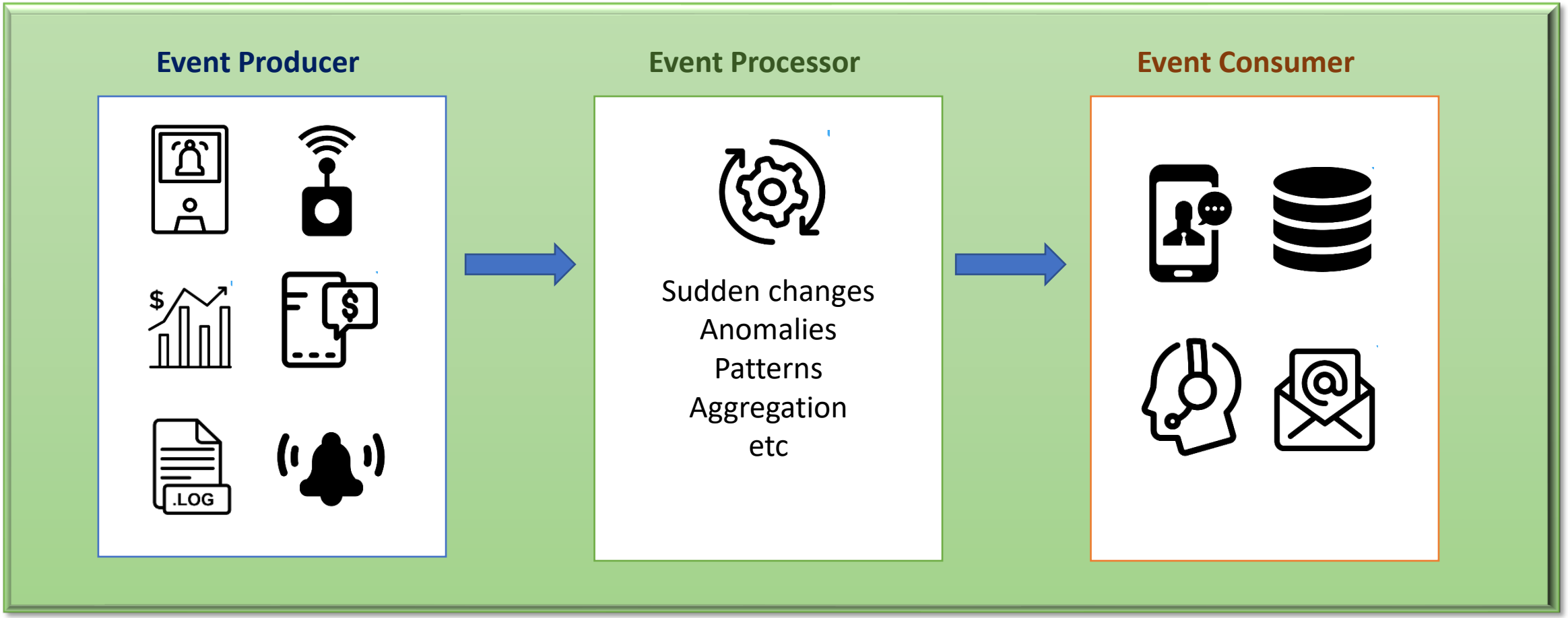
LearnCloud.Info

Event Processing



- **Event Producer** – Process that generate data continuously
- **Event Processor -** An engine to consume event data streams and derive insights from them. -
- **Event Consumer-** An application that consumes the data and takes specific action based on the insights.

Live Event Processing



Challenges

Live Data Processing Challenges

- Data ingestion, processing and output should happen in real-time
- Support high volume of data
- Enough processing power
- Output storage should have high bandwidth
- Quick act on Output processing

Azure options for Live Data Processing

HDInsight with Spark Streaming

HDInsight with Storm

Apache Spark in Azure Databricks

Azure Functions

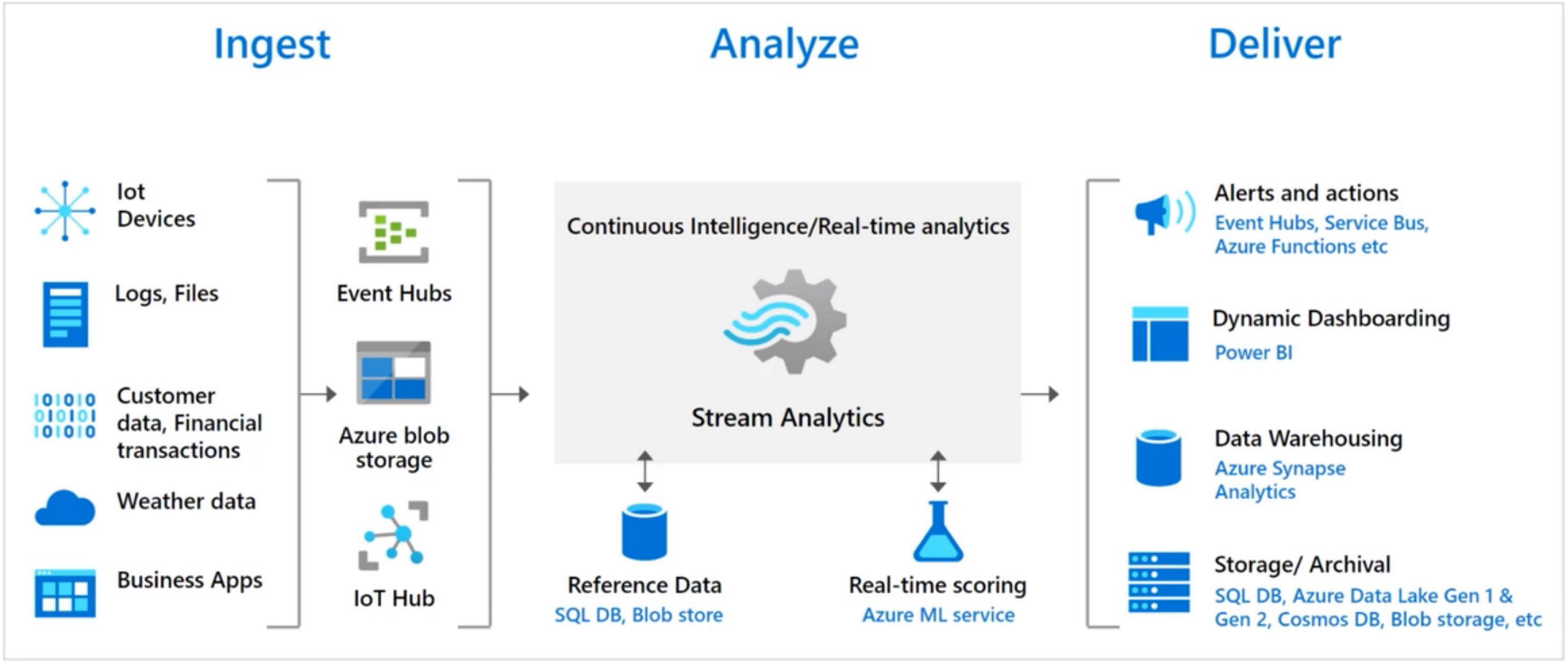
WebJobs

Azure Stream Analytics

Azure Stream Analytics

"A fully managed, real-time analytics service designed to process fast moving streams of data."

Azure Stream Analytics Data Flow

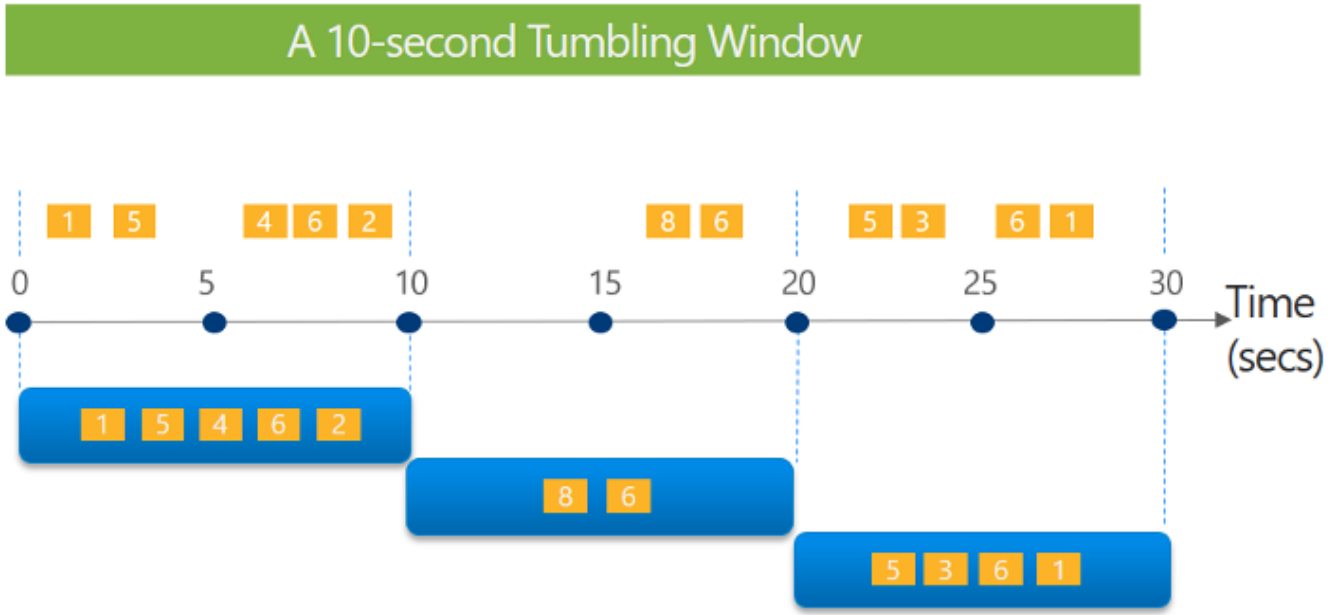


Azure Stream Analytics Windowing

- Each data event has a timestamp
- There is an need to perform an operation (e.g. Count) on events falling in the same time window.
- Azure Stream Analytics achieve this through windows
- Four types of window functions
  - Tumbling window
  - Hopping window
  - Sliding window
  - Session window

TUMBLING WINDOW

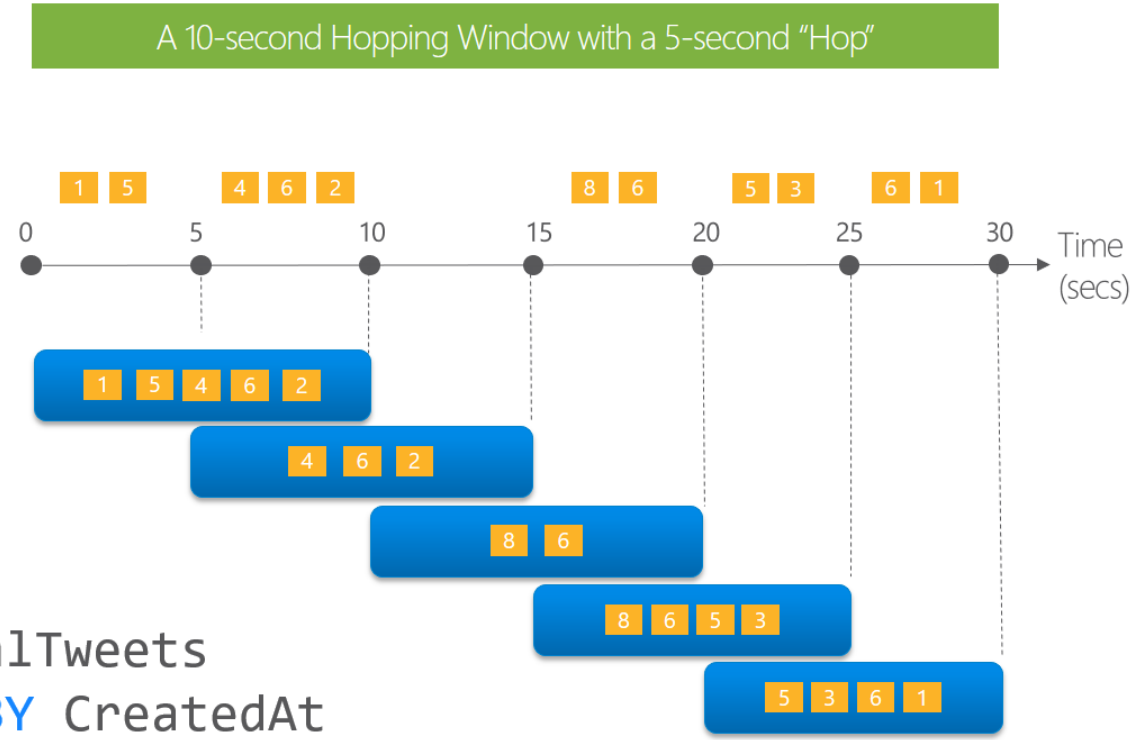
Tell me the count of tweets per time zone every 10 seconds




```
SELECT TimeZone, COUNT(*) AS Count
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY TimeZone, TumblingWindow(second,10)
```

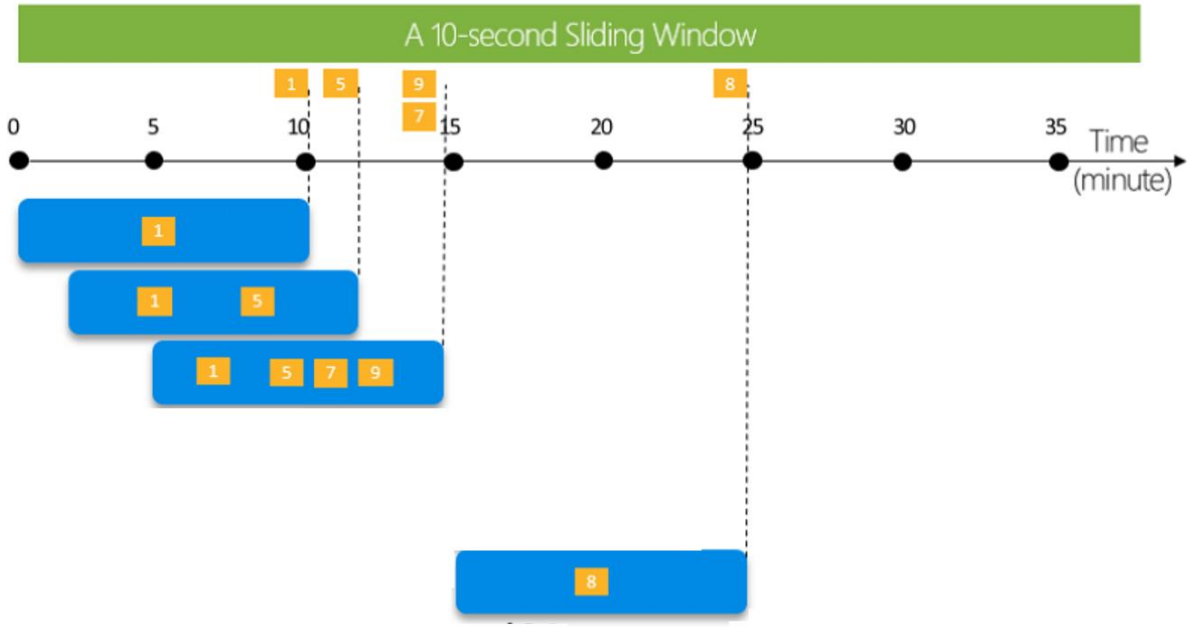
HOPPING WINDOW

Every 5 seconds give me the count of tweets over the last 10 seconds




```
SELECT Topic, COUNT(*) AS TotalTweets
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY Topic, HoppingWindow(second, 10 , 5)
```

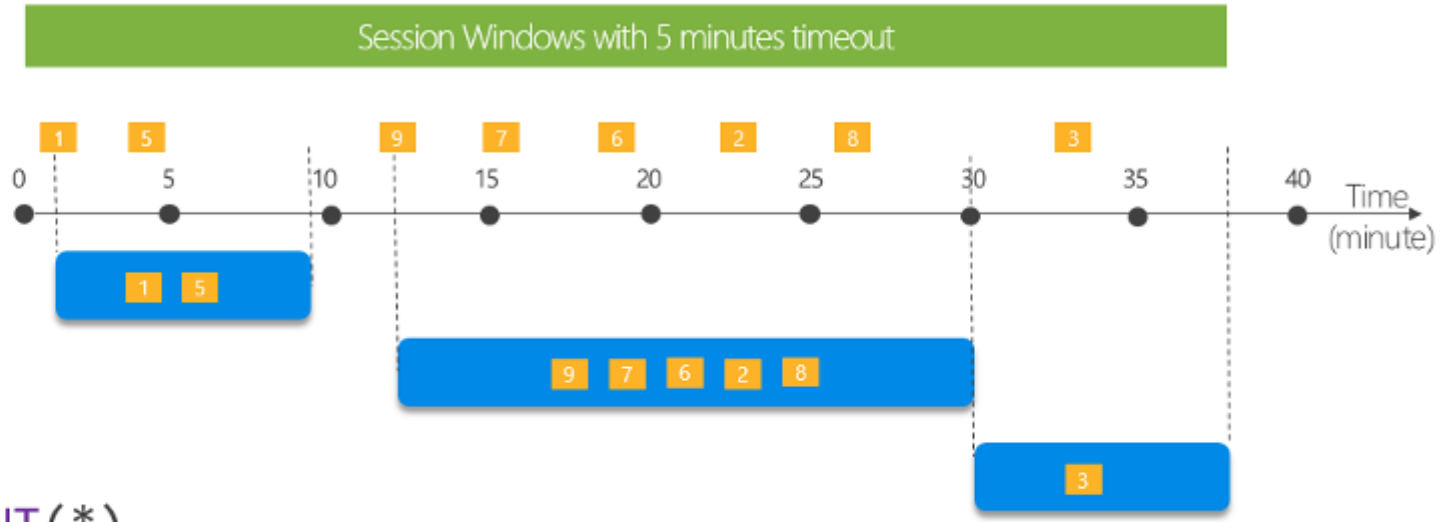
SLIDING WINDOW




```
SELECT COUNT(*)
FROM Input
GROUP BY SlidingWindow(second, 10)
```

SESSION WINDOW

Tell me the count of tweets that occur within 5 minutes to each other.




```
SELECT Topic, COUNT(*)
FROM TwitterStream TIMESTAMP BY CreatedAt
GROUP BY Topic, SessionWindow(minute, 5, 10)
```

Demo Overview

Azure Blob Storage

INPUT

OUTPUT

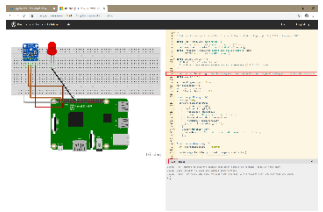
INPUT

OUTPUT

Streaming analytics job

Query (Processing Logic)

Demo Overview



Azure IoT Hub

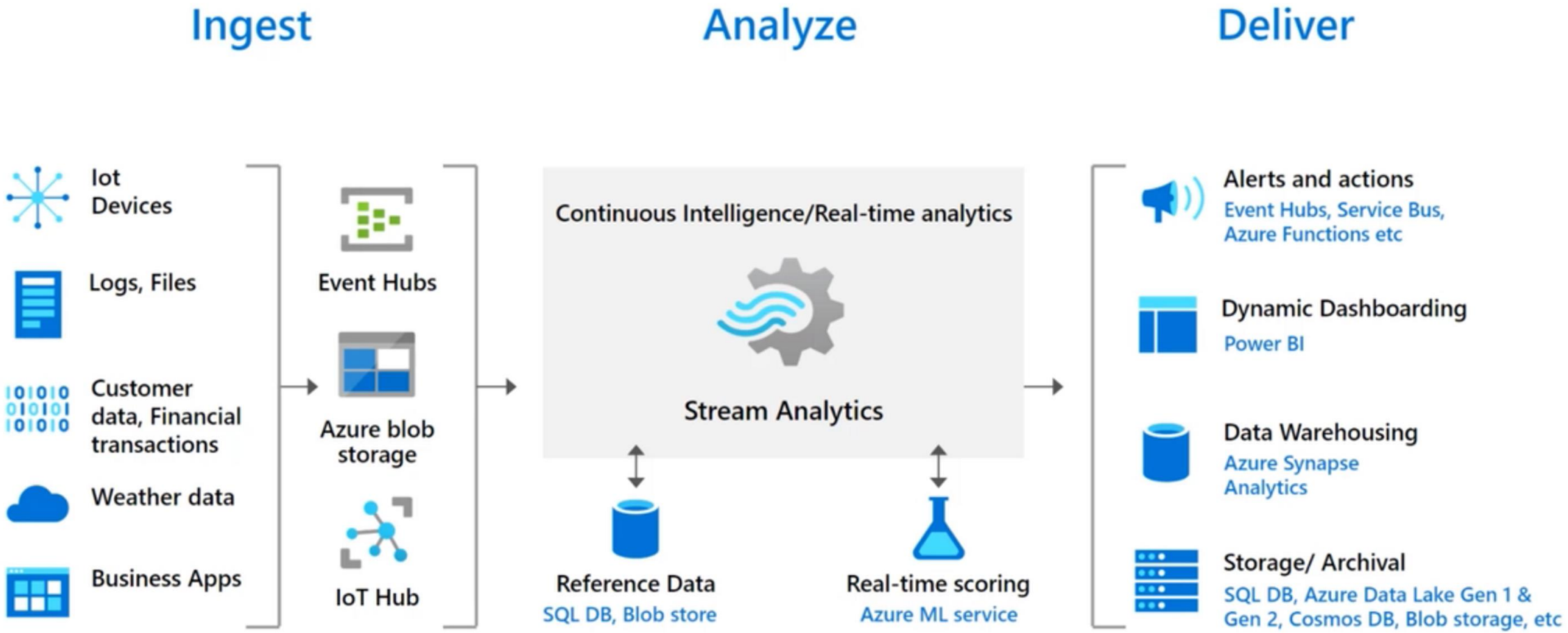
INPUT

Streaming analytics job

OUTPUT

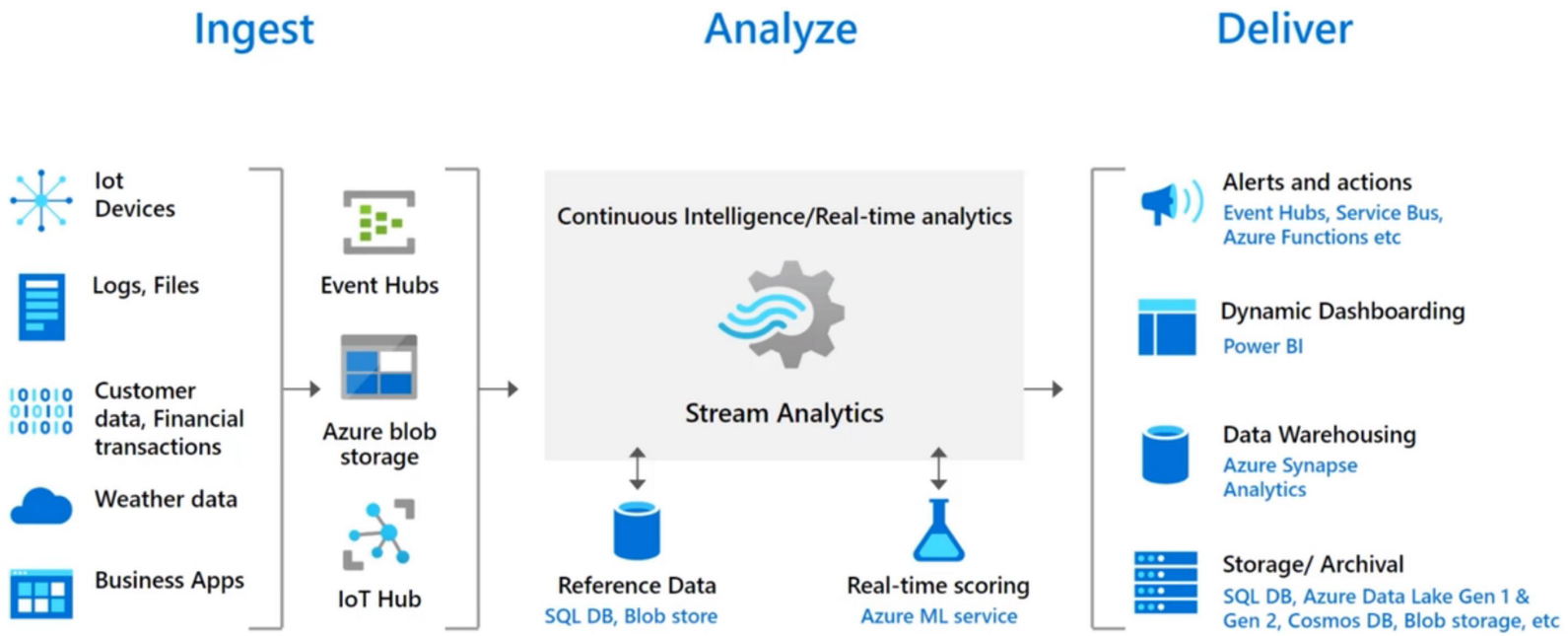
Azure Blob Storage

Azure Stream Analytics Data Inputs

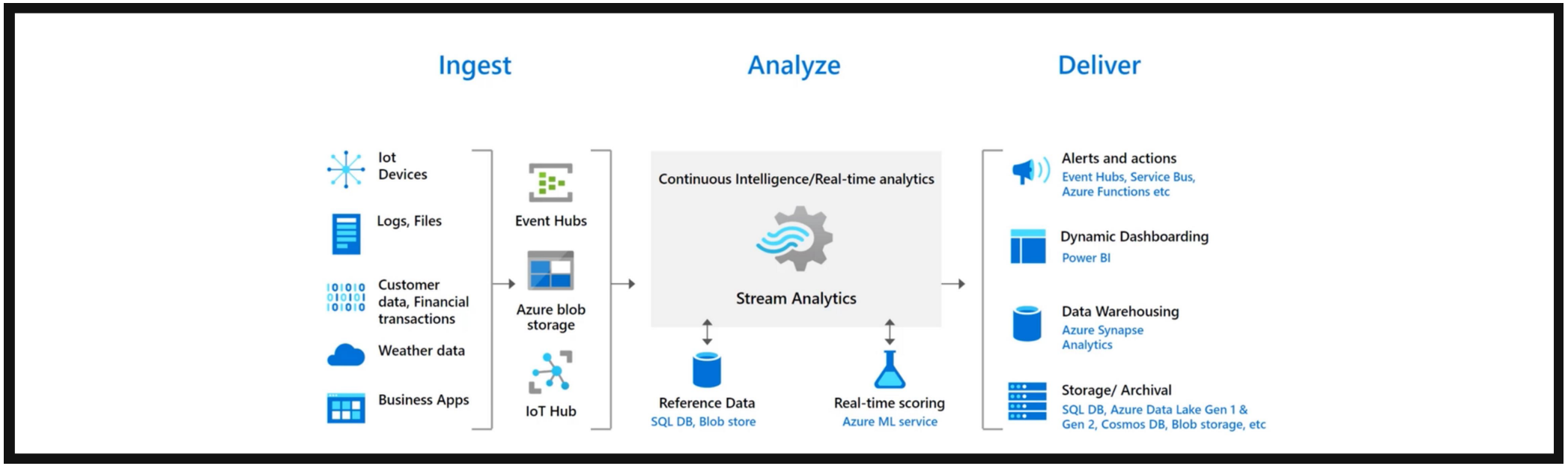


Azure Stream Analytics Data Inputs

- Reference Data Inputs
  - Metadata Lookups (Device name, etc.)



Reference Data Inputs



Metadata Lookup

Device capacity, name, etc.

Acceptable thresholds

Allowed temperatures, etc.

Trusted entities

Registered devices

Any lookup or slow

Changing data

Azure Stream Analytics Stream Data Output

Azure SQL Database

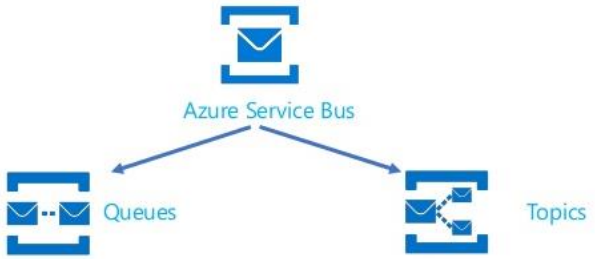
Azure Blob Storage

Azure Data Lake Storage Gen2

Azure Event Hubs

Power BI

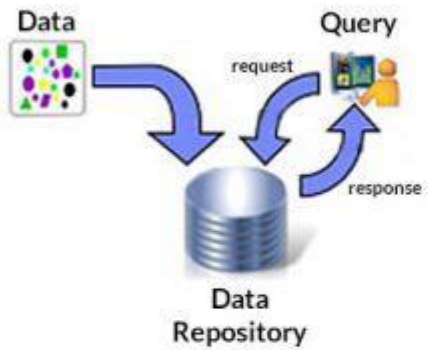
Table Storage



Azure Cosmos DB

Azure Functions

Traditional Processing



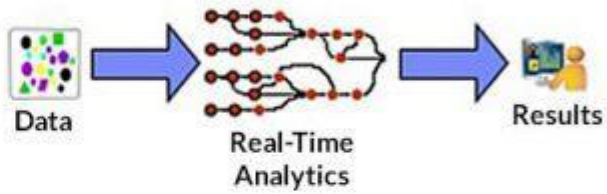
Historical fact finding

Find and analyze information stored on disk

Batch paradigm, pull model

Query-driven: submits queries to static data

Stream Processing



Current fact finding

Analyze data in motion – before it is stored

Low latency paradigm, push model

Data driven: bring data to the analytics

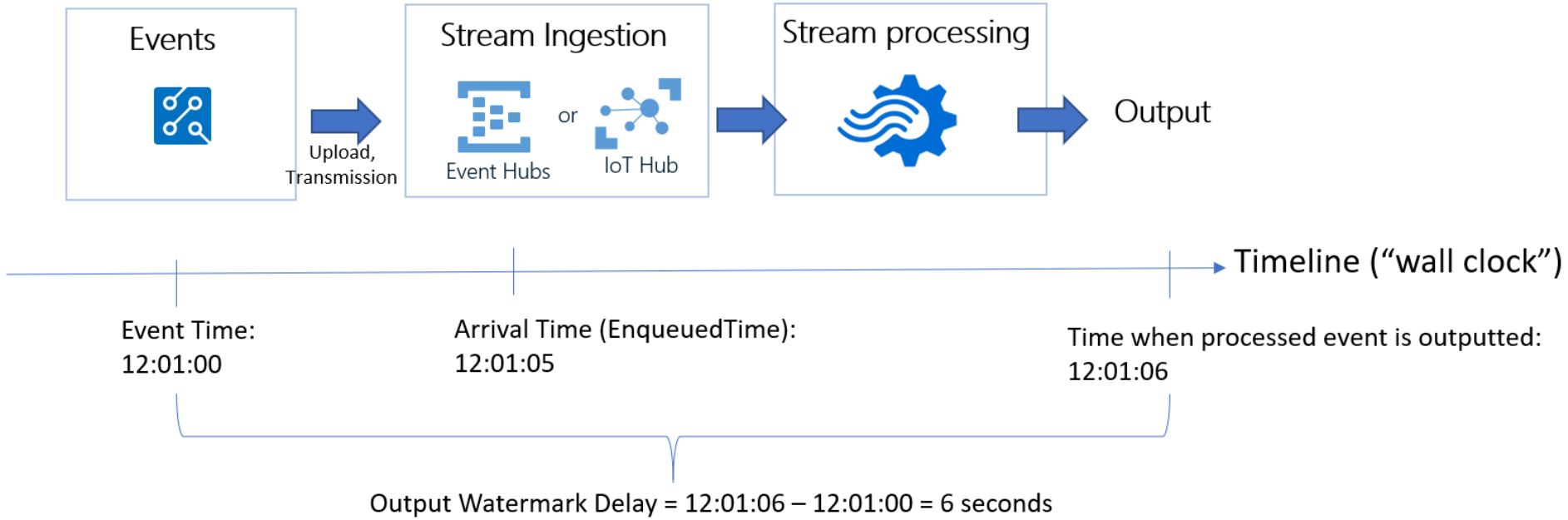
Stream Analytics Service

- **Jobs can be monitored**
  - Azure Portal
  - PowerShell
  - .NET SDK
  - Visual Studio
- **Important metrics**
  - SU% Utilization
  - Runtime Error
  - Watermark delay
  - Input deserialization error
  - Backlogged Input events
  - Data Conversion Errors

Watermark delay matrics

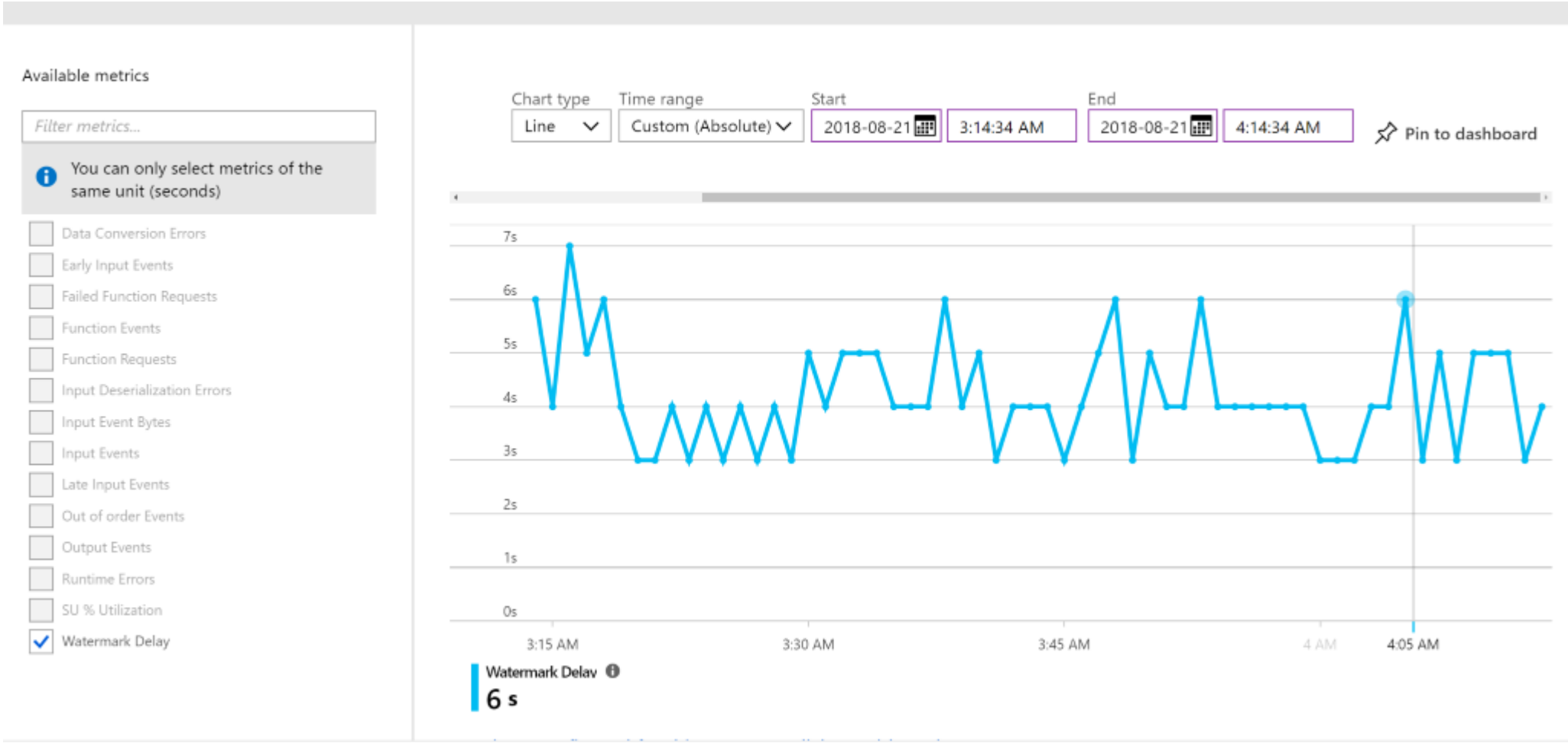
Simple case: no time window, late arrival and out-of-order policy set to 10 seconds

```
SELECT *
FROM input TIMESTAMP BY eventTime
```



Source: Microsoft

Watermark delay matrics



Source: Microsoft

Stream Analytics – Optimization

Stream Analytics – Optimization

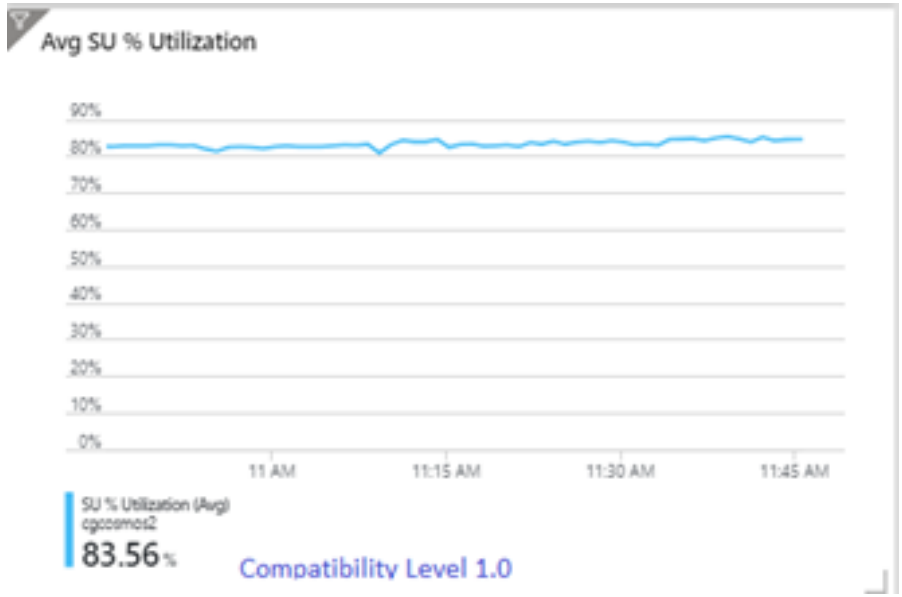
Three main component

- Input
- output
- Data processing Query

Stream Analytics – Optimization

Streaming Units (SUs)

- Processing power (CPU and Memory) allocated to your stream analytics job.
- Azure Stream Analytics jobs perform all processing in memory
- If SU% utilization is low and input events get backlogged
- Microsoft recommends setting an alert on 80% SU Utilization metric to prevent resource exhaustion
- The best practice is to start with 6 SUs for queries that don't use PARTITION BY
- Complex query logic could have high SU% utilization even when it is not continuously receiving input events.



Stream Analytics – Optimization

Parallelization

- Partitioning helps to divide data in subsets.
- This would be based on partition key.
- If the data in the Event Hub has a partition key defined, then it is highly recommended to define the partition key in the input of Stream Analytics Job.
- Input are already partitioned, output needs to be partitioned
- Embarrassingly parallel jobs
  - An embarrassingly parallel job is the most scalable scenario in Azure Stream Analytics.
  - It connects one partition of the input to one instance of the query to one partition of the output.
  - The number of input partitions must equal the number of output partitions.

```sql
SELECT *
INTO output
FROM input
PARTITION BY DeviceID
INTO 10
```

Stream Analytics – Optimization

Steps in Query

- You can have multiple step in a query.
- You can start with 6 SUs for queries that don't use PARTITION BY
- You can also add 6 streaming units for each partition in a partitioned step.
- Example:
  - Let’s say your input stream is partitioned by value of 10, and you only have one step in query

SQL

```
SELECT *
INTO output
FROM input
PARTITION BY DeviceID
INTO 10
```

Stream Analytics – Optimization

LearnCloud.Info