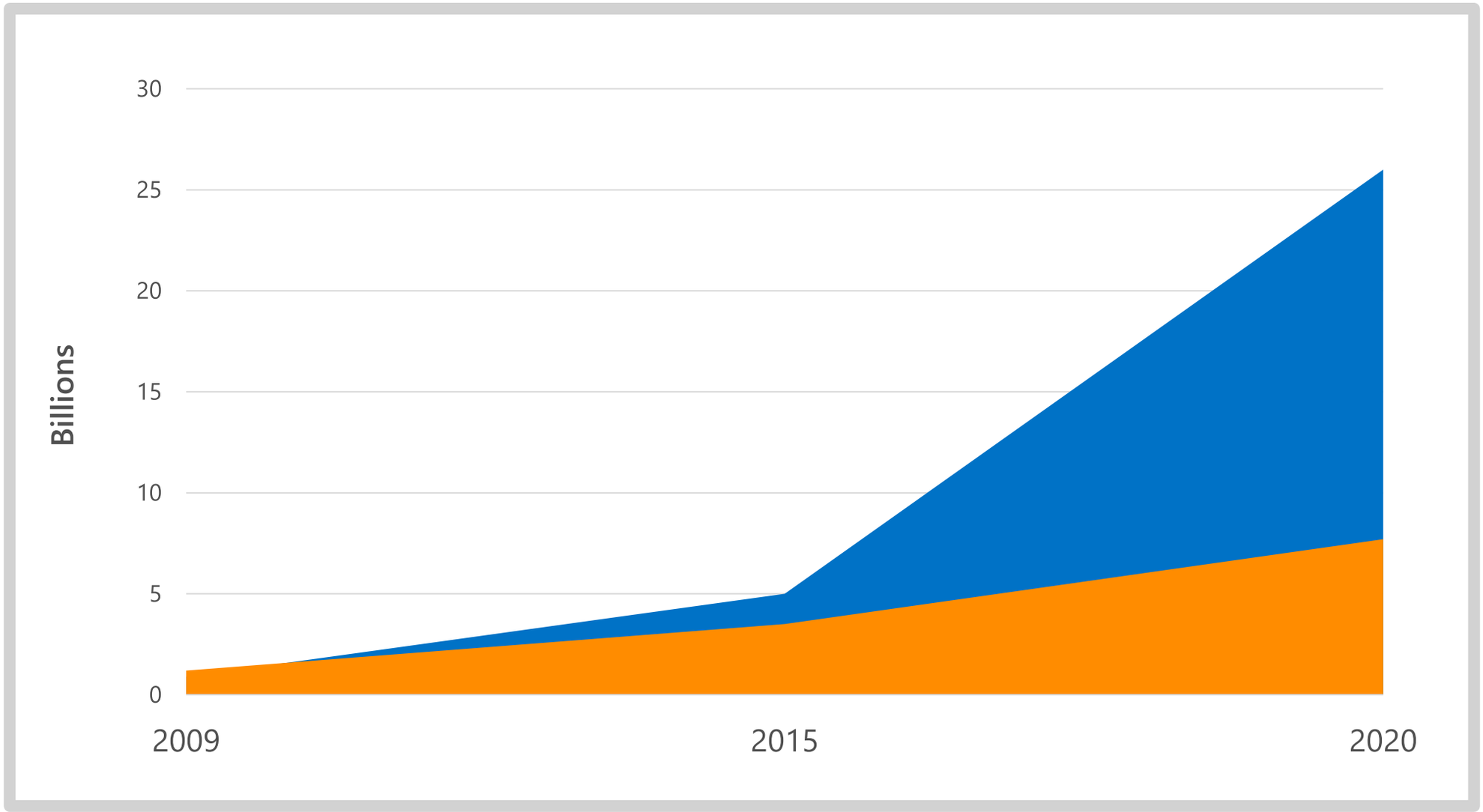
Microsoft

Azure Stream Analytics

Growth of internet connected devices vs. connected PCs, smartphones, tablets

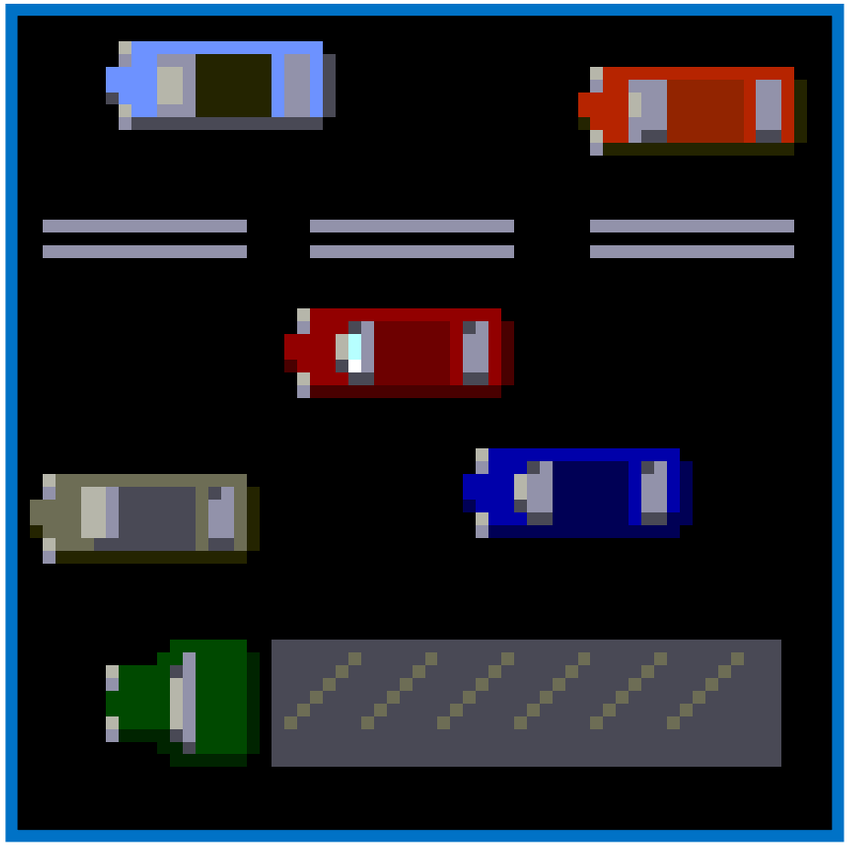


http://www.gartner.com/newsroom/id/2905717
Forecast: The Internet of Things, Worldwide, 2013

What is streaming?

Data at rest

Data in motion



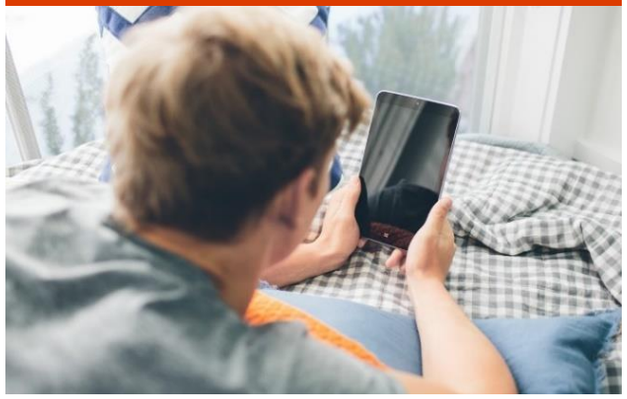
The need for real-time processing

Real-time fraud detection

Connected car

Click-stream analysis

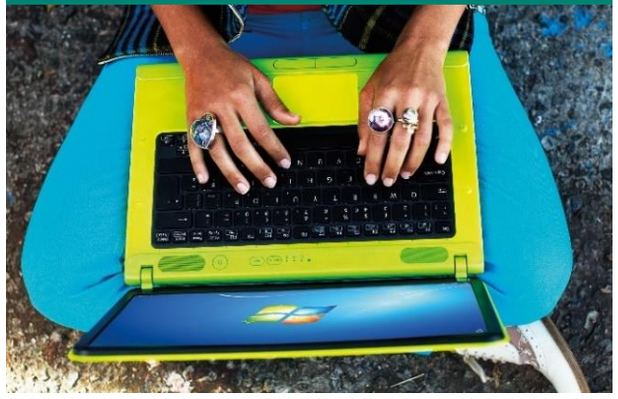
Remote device monitoring



Smart grid

CRM alerting sales

Predictive maintenance



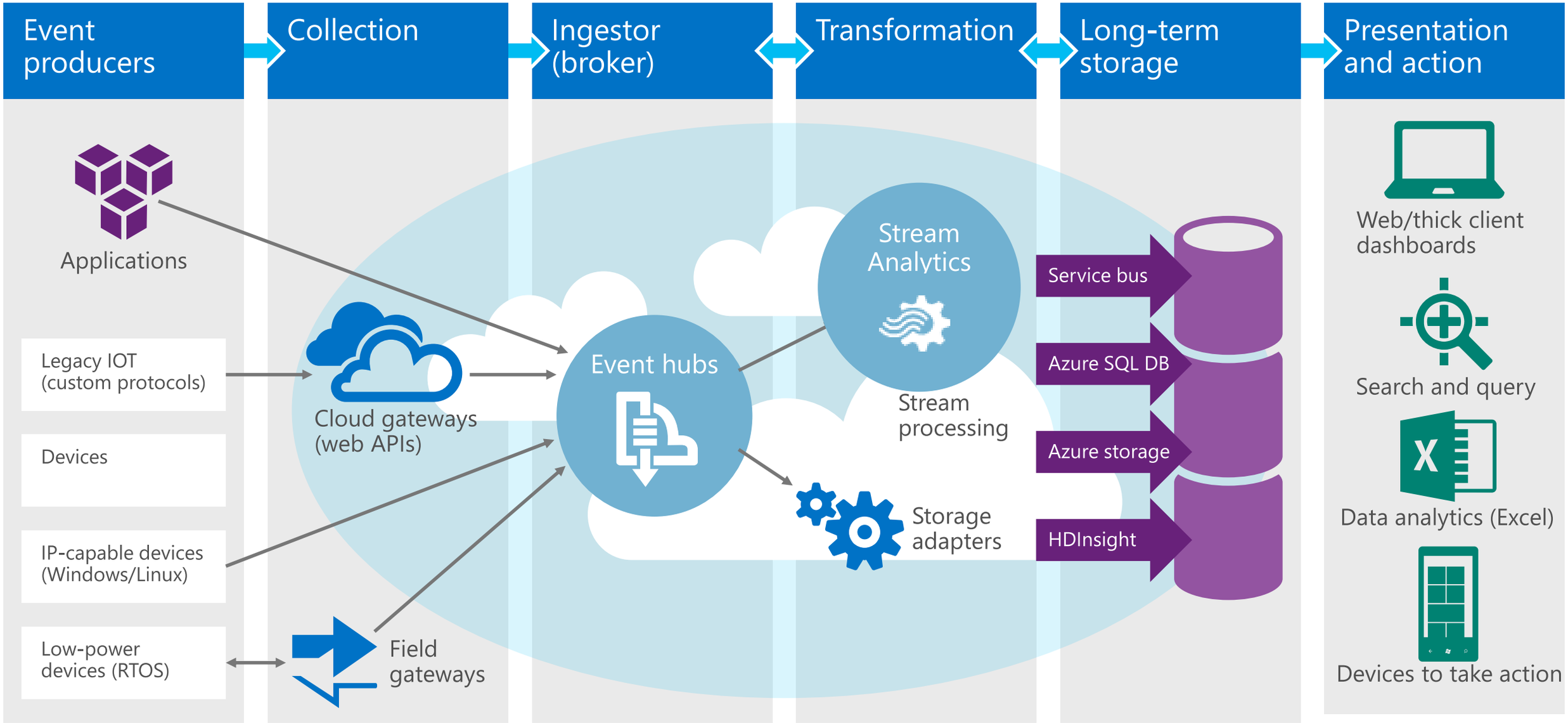
Real-time financial sales tracking

How do customers create a real-time streaming solution?

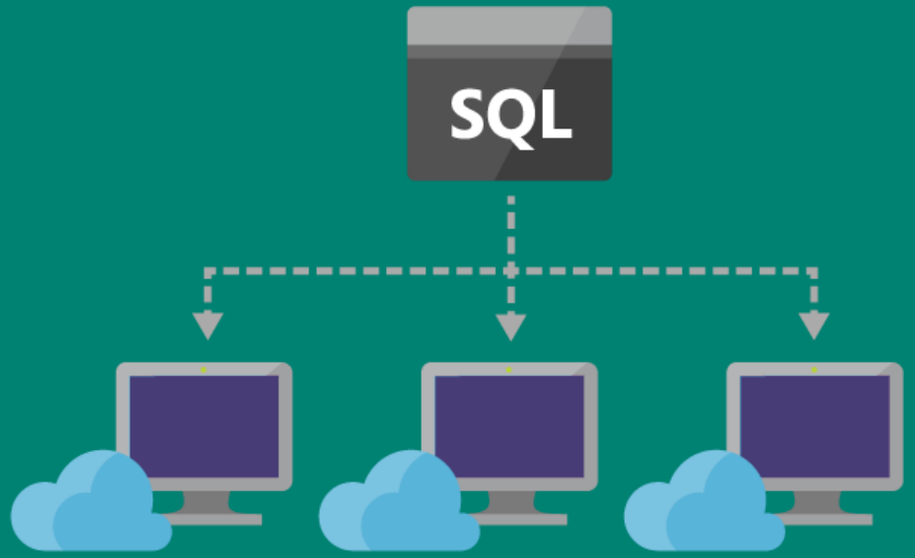
Customers using Azure Stream Analytics

From event or data streams to real time insights in **less time with less people resources**

End-to-end stream processing on Microsoft Azure



Rapid development

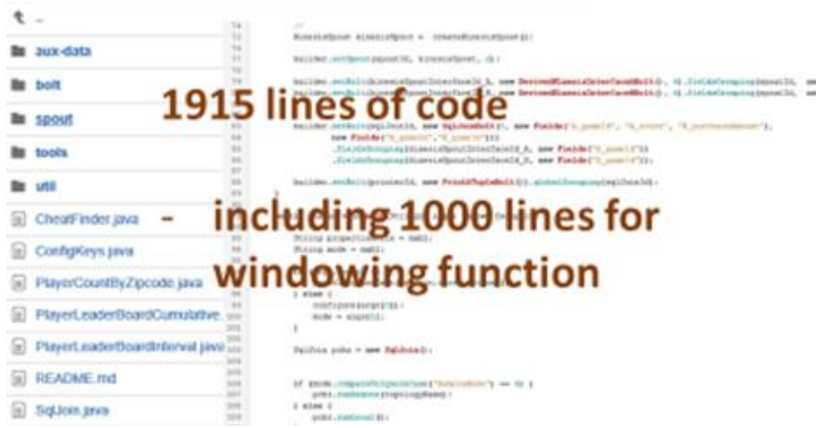


Only SQL queries needed

Developers uses declarative SQL commands

Some functions take several lines of code versus thousands from other solutions

Thousand lines of code in other solutions, such as Apache Storm

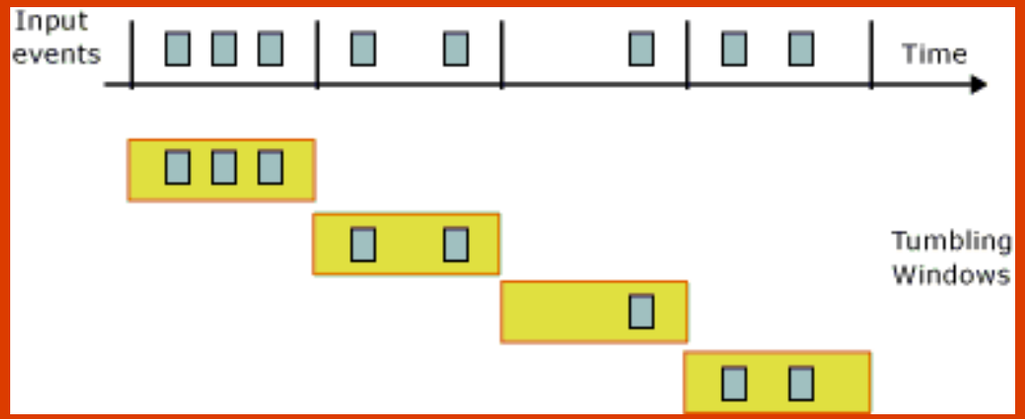


Versus

3 lines of code in Stream Analytics

```
SELECT Avg(Purchase), Score
FROM GameDataStream
GROUP BY TumblingWindow(5, Minute), Score
```

Built in temporal semantics



Implement temporal functions

Tumbling Windows
Hopping Windows
Sliding Windows

Manage out-of-order events

With configuration instead of code

Manage actions on late events

Using policy settings instead of code

Functions New
Use Azure ML web services

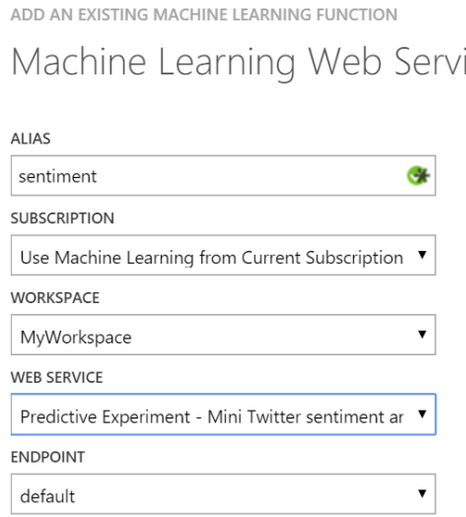
Implement Azure ML web services in Real-Time

Includes Real-time R/Python models

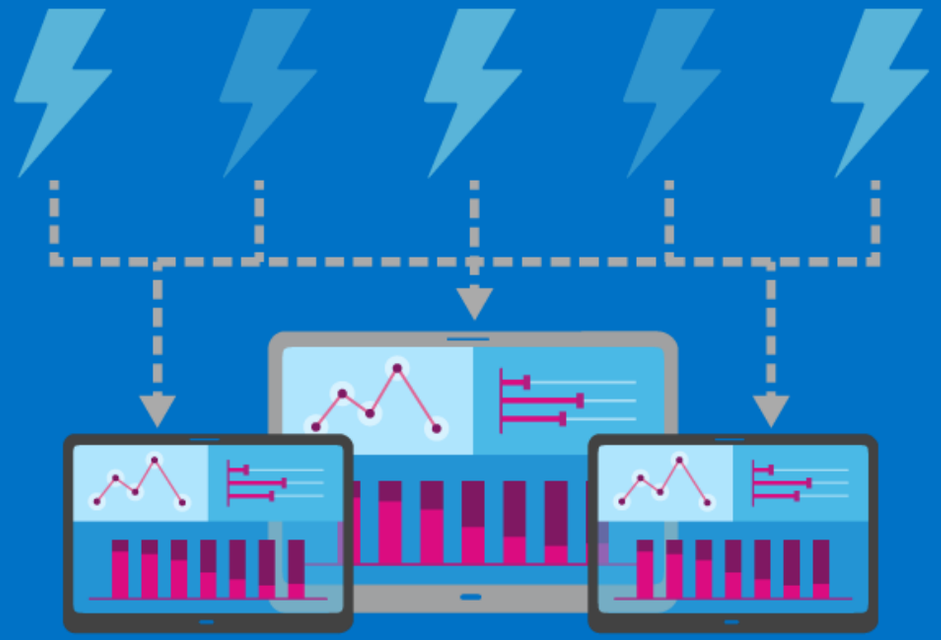
Example Query

```
WITH subquery AS(SELECT text, sentiment(text) AS result
FROM myinput)

SELECT text, result.[Score]
INTO myoutput
FROM subquery
```



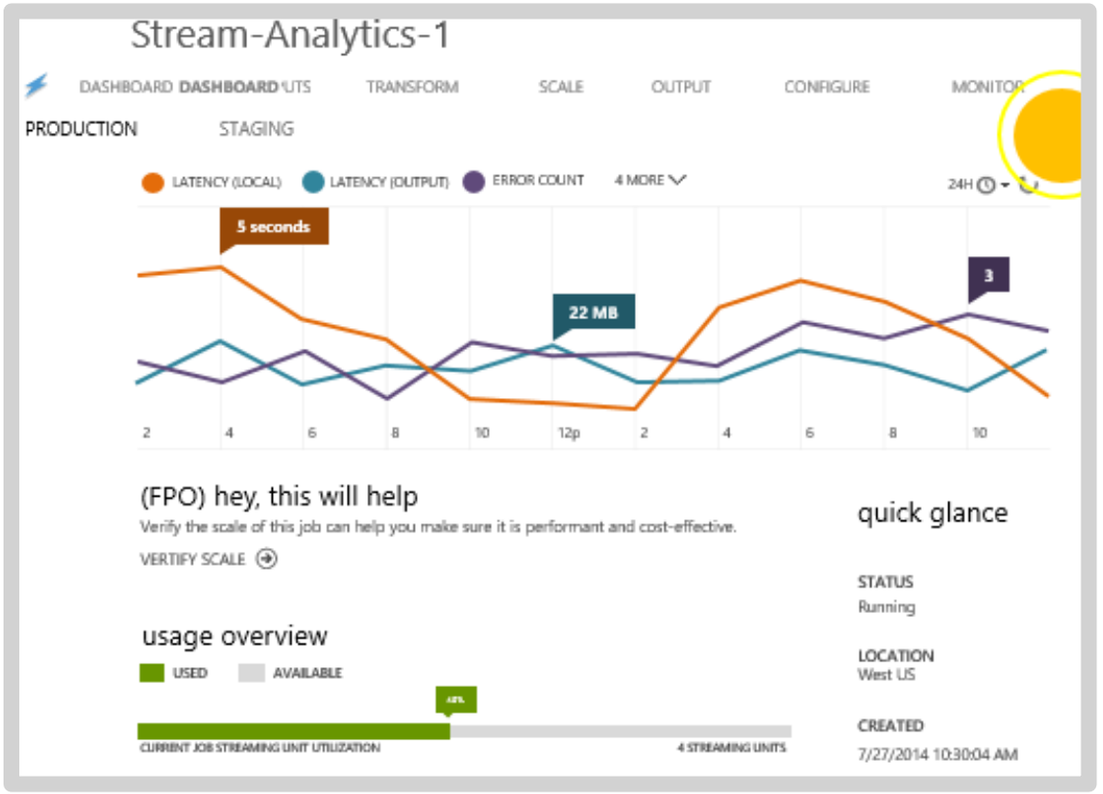
Scheduling and monitoring built in



Built-in monitoring

View your system's performance at a glance

Help you find the cost-optimal way of deployment



Azure Stream Analytics Query Language

2014

Azure Stream Analytics – Query Basics

- ASA has a set of logical operators
  - These should (mostly) look familiar if you know databases
- Temporal Semantics are built-in
  - All operators respect, and some use, the temporal qualities of events
- You write declarative queries against ASA in SQL

Query Language - Overview

DML Statements

- SELECT
- FROM
- WHERE
- GROUP BY
- HAVING
- CASE
- JOIN
- UNION

Statistical Functions

- VAR
- VARP
- STDEV
- STDEVP

Date and Time Functions

- DATENAME
- DATEPART
- DAY
- MONTH
- YEAR
- DATETIMEFROMPARTS
- DATEDIFF
- DATADD

Windowing Extensions

- Tumbling Window
- Hopping Window
- Sliding Window
- Duration

Aggregate Functions

- SUM
- COUNT
- AVG
- MIN
- MAX

Scaling Functions

- WITH
- PARTITION BY

String Functions

- LEN
- CONCAT
- CHARINDEX
- SUBSTRING
- PATINDEX

Supported types

Type	Description
bigint	Integers in the range -2^63 (-9,223,372,036,854,775,808) to 2^63-1 (9,223,372,036,854,775,807).
float	Floating point numbers in the range - 1.79E+308 to -2.23E-308, 0, and 2.23E-308 to 1.79E+308.
nvarchar(max)	Text values, comprised of Unicode characters. Note: A value other than max is not supported.
datetime	Defines a date that is combined with a time of day with fractional seconds that is based on a 24-hour clock and relative to UTC (time zone offset 0).

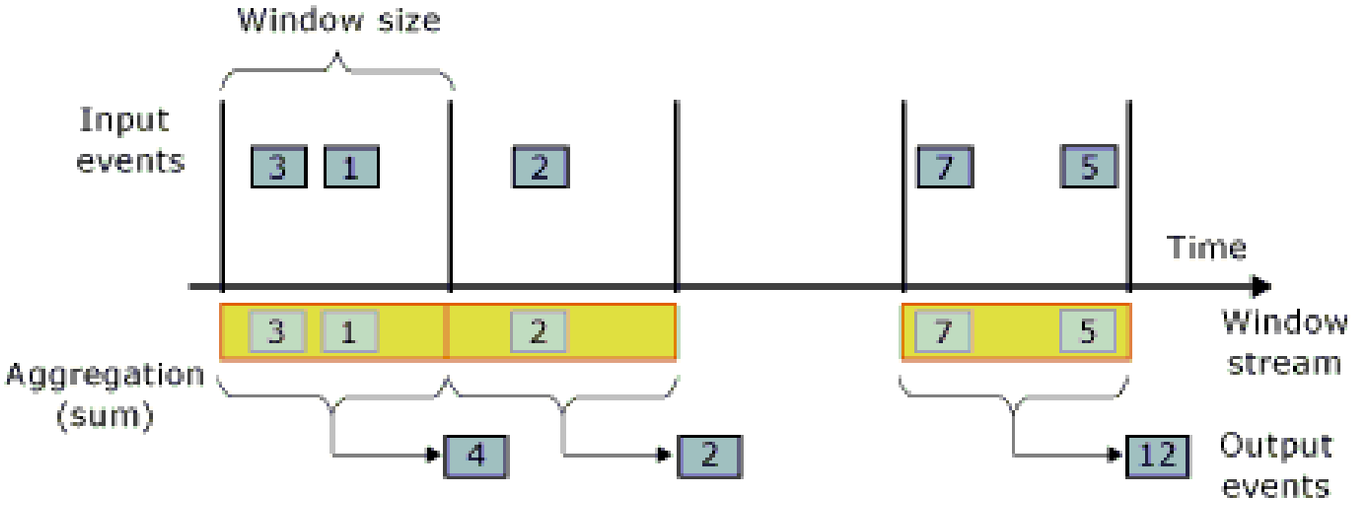
Select and Filter


```
SELECT (value * value) AS Square
FROM MySource
```

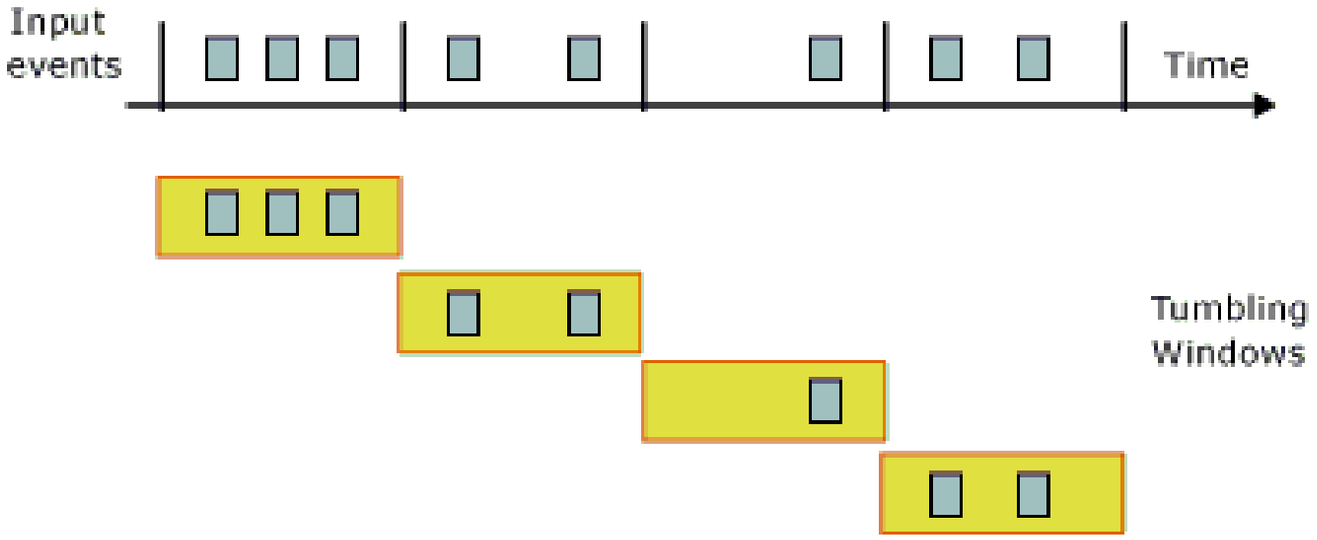

```
SELECT * FROM MySource
WHERE name = "foo"
```

Temporal Windows

- Tumbling Windows
  - Repeating, non-overlapping, fixed interval windows
- Hopping Windows
  - Generic window, overlapping, fixed size
- Sliding Windows
  - Slides by an epsilon and produces output at the occurrence of an event

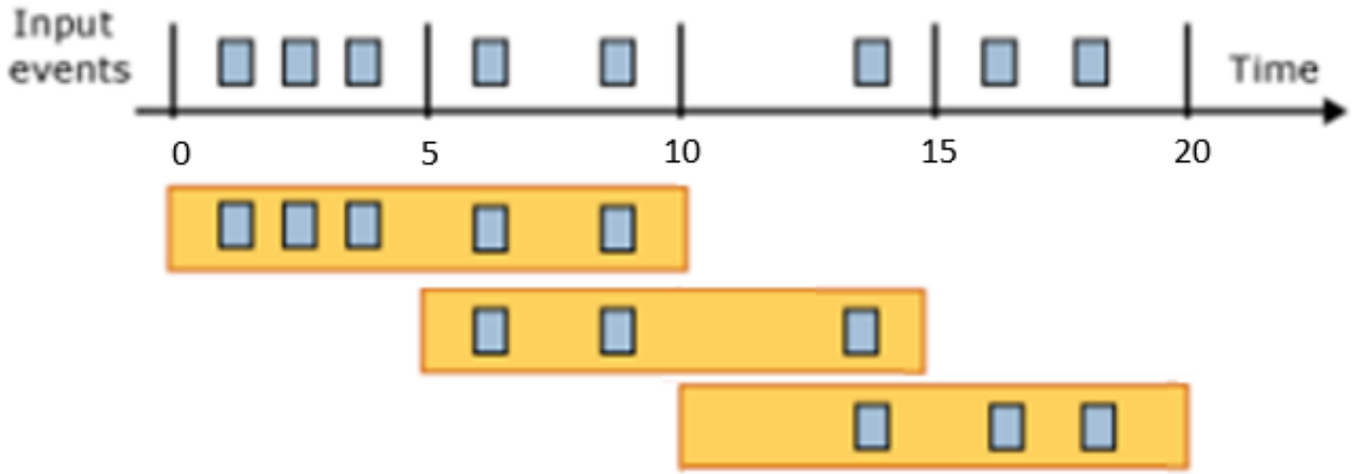


Tumbling Window




```
SELECT System.TimeStamp AS OutTime, TollId,
COUNT (*)
FROM Input TIMESTAMP BY EntryTime
GROUP BY TollId, TumblingWindow(minute,5)
```

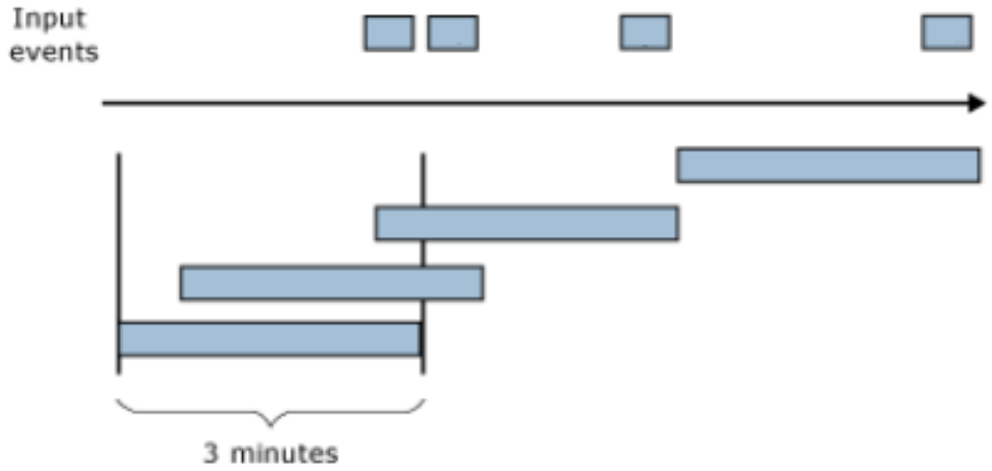
Hopping Windows




```
SELECT System.TimeStamp AS OutTime, TollId,
COUNT (*)
FROM Input TIMESTAMP BY EntryTime
GROUP BY TollId, HoppingWindow(minute, 10 , 5)
```

Sliding Windows

Finds all toll booths which have served more than 3 vehicle in the last 3 minutes




```
SELECT System.TimeStamp AS OutTime, TollId, COUNT (*)
FROM Input TIMESTAMP BY EntryTime
GROUP BY TollId, SlidingWindow(minute, 3)
HAVING Count(*) > 3
```