



IoT & Azure

Mirco Vanini

info@proxsoft.it

@MircoVanini

<http://proxsoft.it>

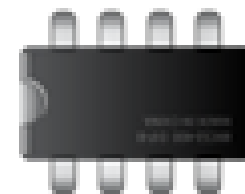


Microsoft® MVP Windows Embedded

Sponsor



TinyCLR.it



β EPS

Best Embedded Products & Solutions

Leonardo
RICERCHE

innovactive
(software engineering)

h0va
community

proxima software



embedded & mobile solutions
www.proxsoft.it | info@proxsoft.it



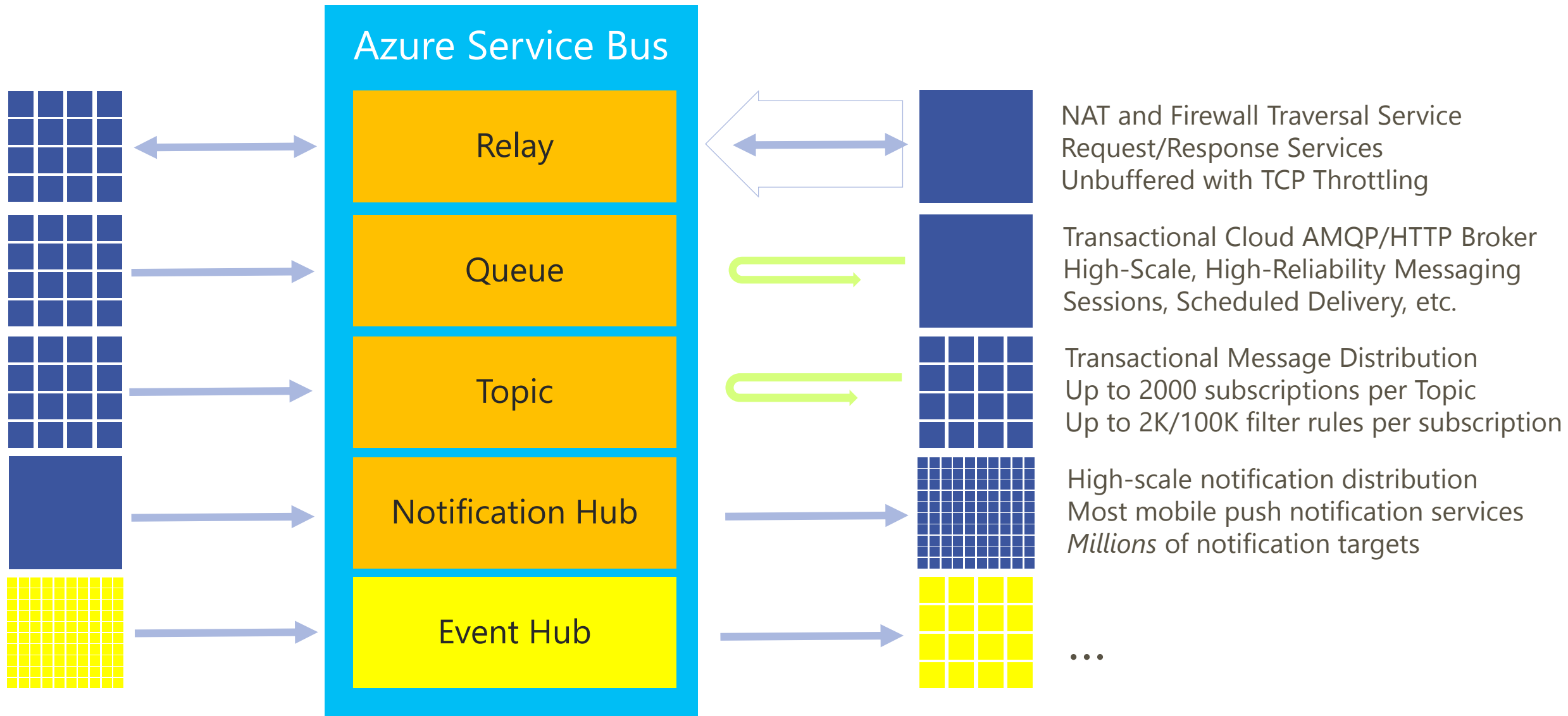
- Microsoft Azure IoT Services
- Azure Service Bus
- Azure Event Hubs
- Azure Stream Analytics
- Demo
- Recap

Microsoft Azure IoT services

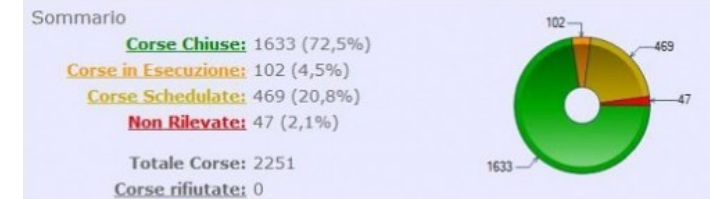
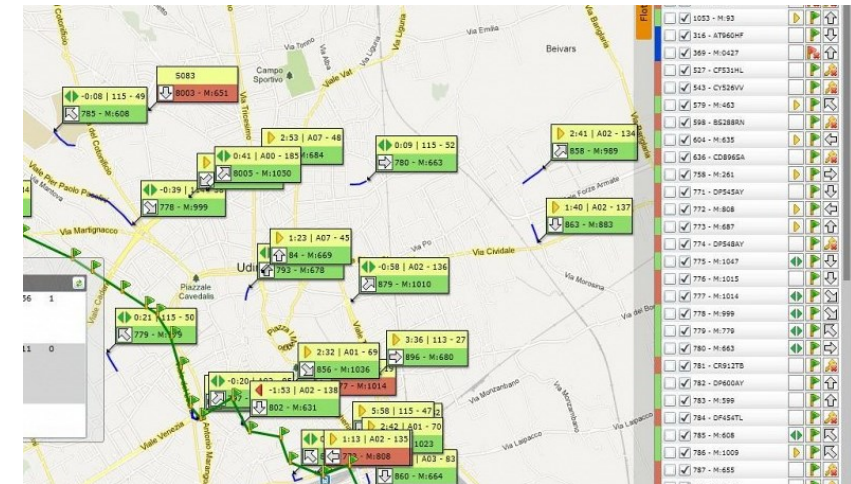


Producers	Connect Devices	Storage	Analytics	Take Action
	Event Hubs	SQL Database	Machine Learning	Azure Websites
	Service Bus	Table/Blob Storage	Stream Analytics	Power BI
	External Data Sources	DocumentDB	HDInsight	Notification Hubs
		External Data Sources	Data Factory	Mobile Services
				BizTalk Services

Azure Service Bus



Intelligent Mobility and Connected Car



Telemetry ... the problem



Information flowing from a device to other systems for conveying status of device and environment

Unidirectional

Data frequency can be different based on applications/conditions



Service Bus - Queues & Topics



Message are durably stored but with TTL

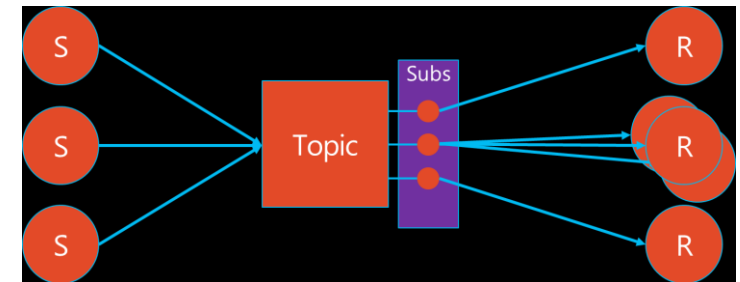
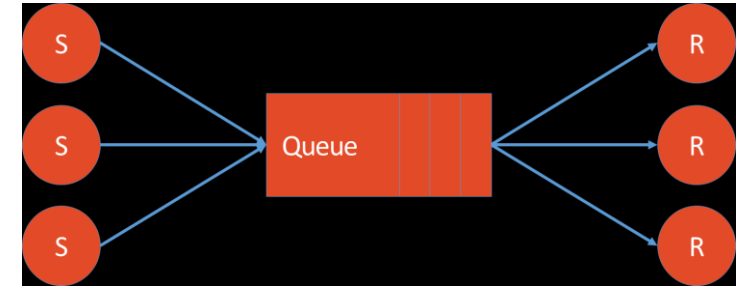
Receive & Delete or Peek Lock

Sessions (for FIFO feature)

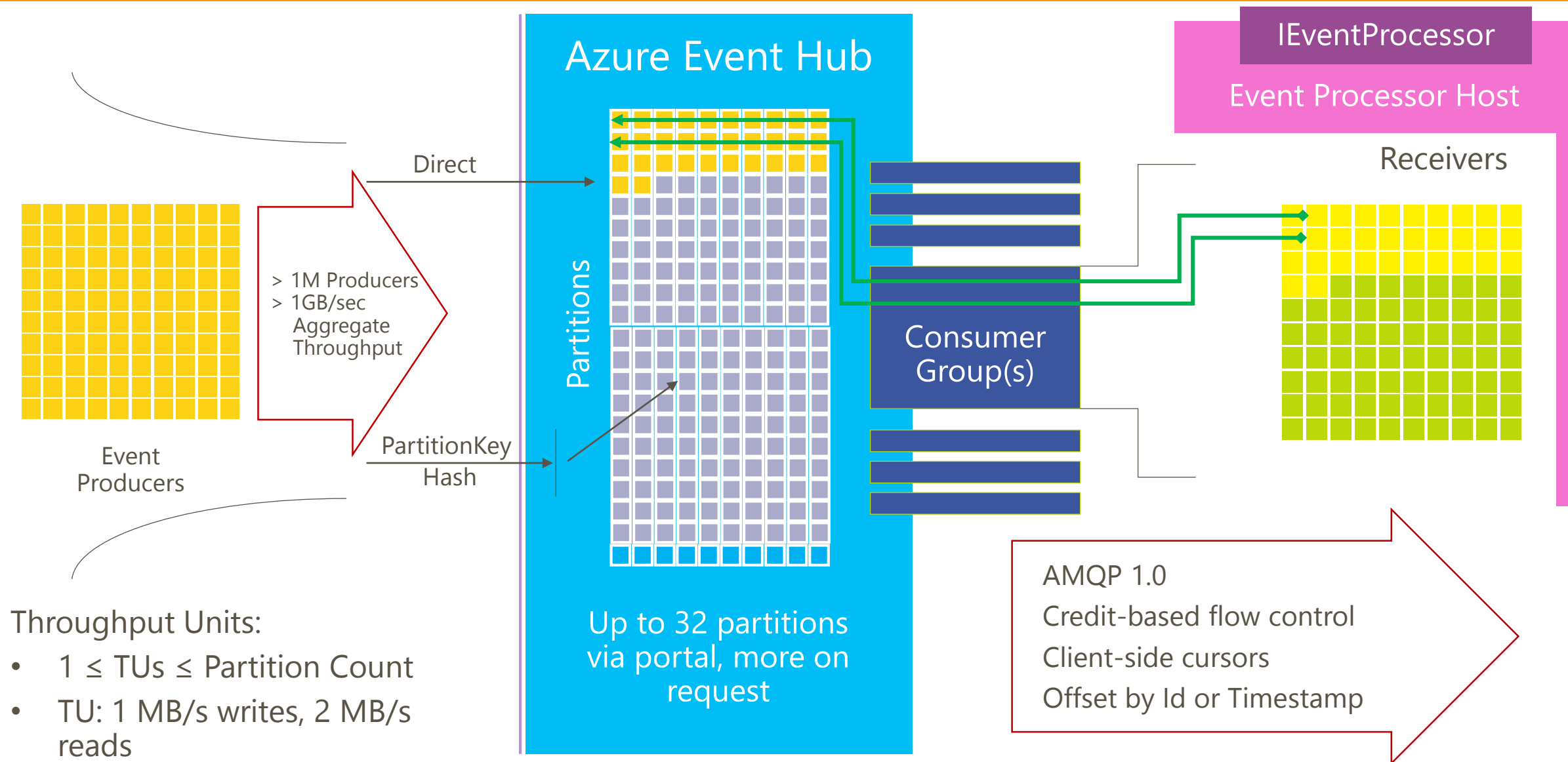
Request/Reply pattern (based on correlation)

Transaction for batch send/receive

Dead-letter queue (TTL or "poisoned" messages)



Introducing Azure Event Hubs



Throughput Units:

- $1 \leq \text{TUs} \leq \text{Partition Count}$
- TU: 1 MB/s writes, 2 MB/s reads



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.

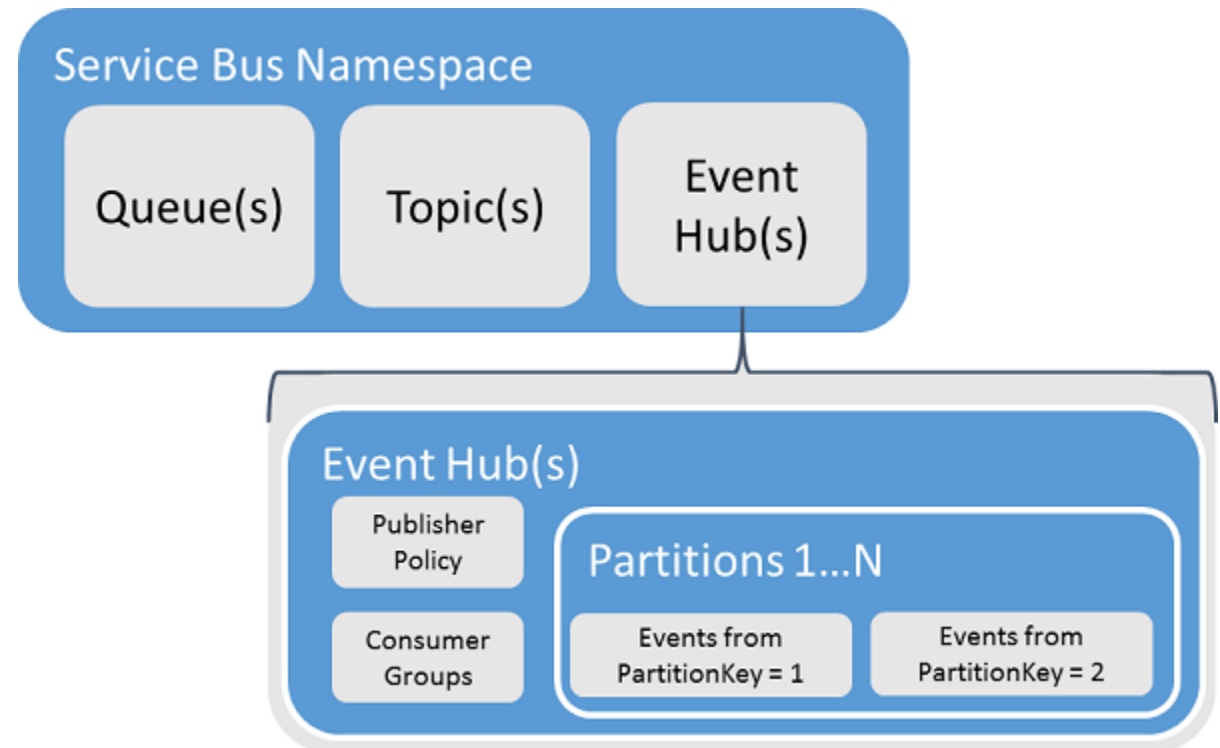


The Azure Event Hubs capability differs from Service Bus topics in that it is strongly biased towards high throughput and event processing scenarios.

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

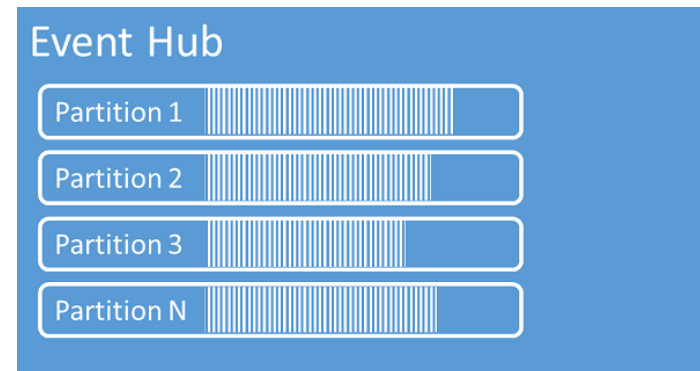


Event Hubs provides message streaming through a partitioned consumer pattern.

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.





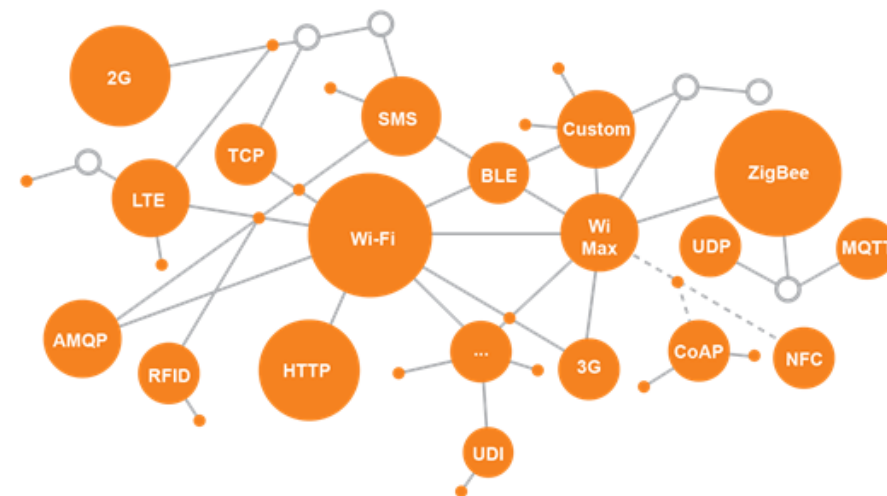
In the Event Hubs context, messages are referred to as *event data*. Event data contains the body of the event, a user defined property bag, and various metadata about the event such as its offset in the partition and its number in the stream sequence. Partitions are filled with a sequence of event data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



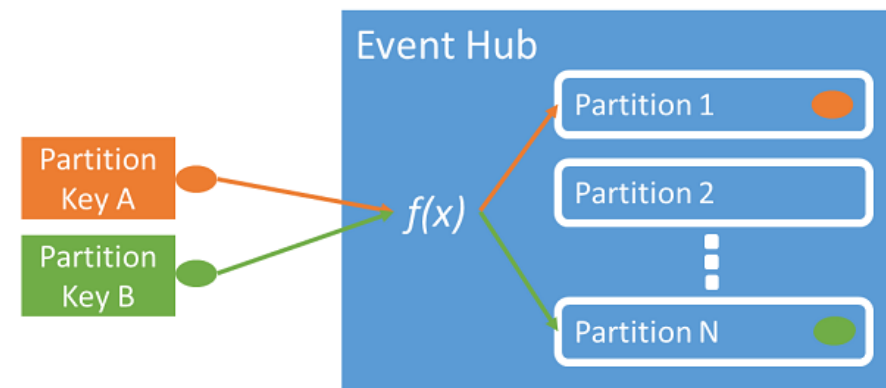
The choice to use AMQP or HTTP is specific to the usage scenario. AMQP requires the establishment of a persistent bidirectional socket in addition to transport level security (TLS) or SSL/TLS. This can be a costly operation in terms of network traffic, but only happens at the beginning of an AMQP session. 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.

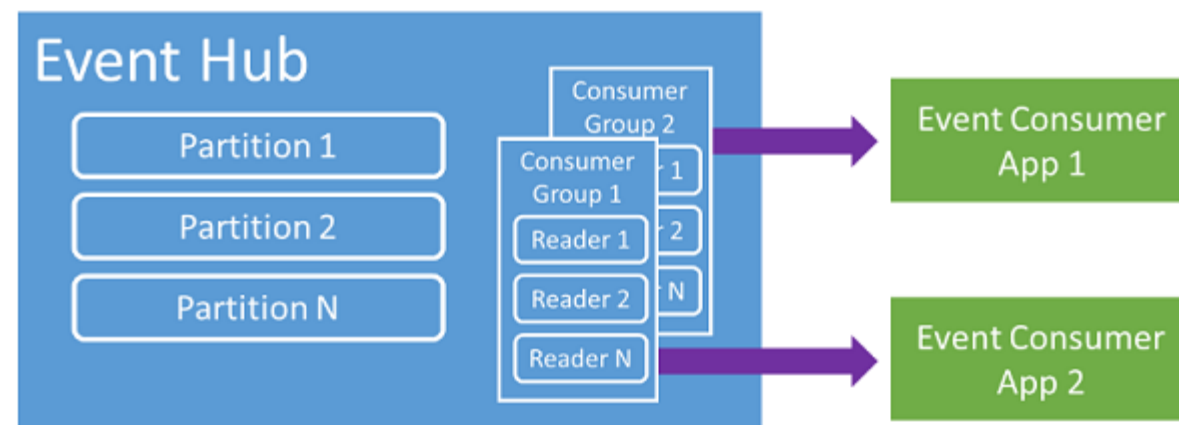
Event Hubs ensures that any and all events sharing the same partition key value are delivered in order, and to the same partition.



Event Hubs – Consumer Groups



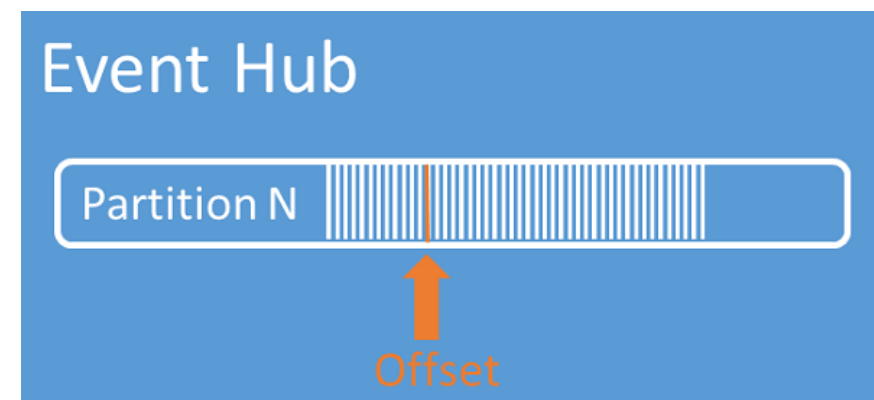
The publish/subscribe mechanism of Event Hubs is enabled through *consumer groups*. A consumer group is a view (state, position, or offset) of an entire Event Hub. Consumer groups enable multiple consuming applications to each have a separate view of the event stream, and to read the stream independently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





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

Stream Analytics - Overview



Real-time business insights

Stream Analytics is an event processing engine that helps uncover real-time insights from devices, sensors, infrastructure, applications and data.

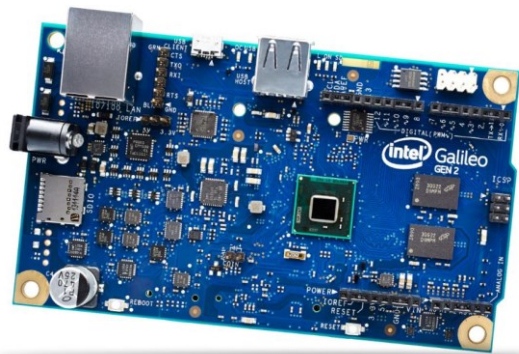
Stream millions of events per second

Stream Analytics provides out-of-the-box integration with Event Hubs to ingest millions of events per second. Together, Event Hubs and Stream Analytics let you process massive amounts of real-time data so you can make business decisions in real-time.

Real-time processing

Stream Analytics will process ingested events in real-time, comparing multiple real-time streams or comparing real-time streams together with historical values and models

Event Hubs - Connect the Things



Intel Galileo

Intel® Quark™ X1000

Win32 / C++

Qpid Proton - Apache Library

<http://qpid.apache.org/proton/>
<https://amqpnetlite.codeplex.com/>

GHI Fez Spider

72 Mhz 32-bit ARM7

.NET MF & Gadgeteer / C#

Amqp.Net Lite

Toradex Colibri T30

NVIDIA® Tegra™ 3

.NET CF / WEC2013 / C#

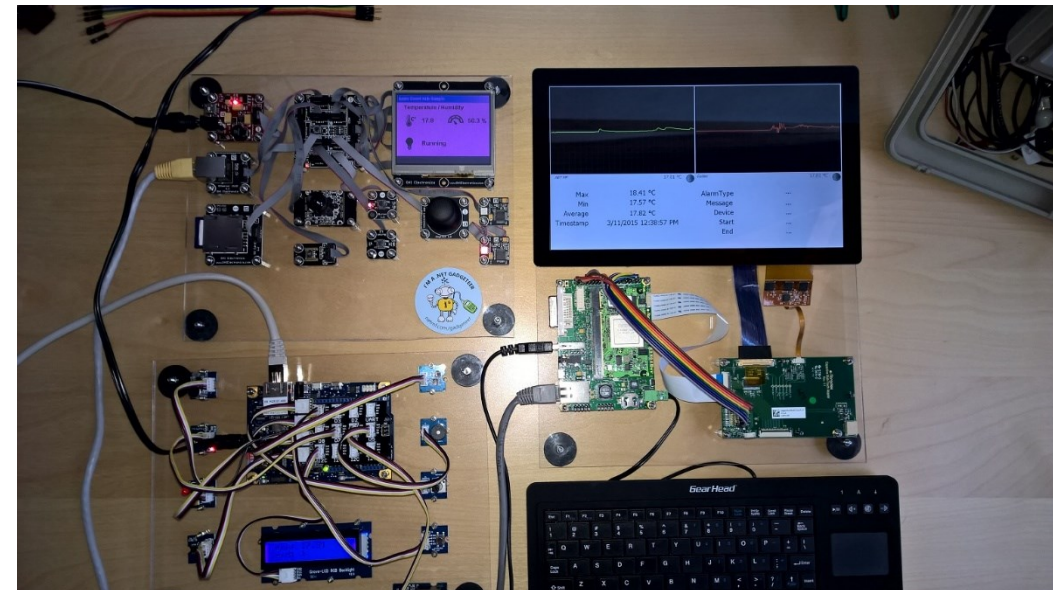
Amqp.Net Lite

 **Toradex**
Swiss. Embedded. Computing.

demo

IoT day 2015

connect things to azure
event hub



Event Hubs - pricing



	BASIC	Standard
Ingress events	€0.0209 per million events	€0.0209 per million events
Throughput unit (1 MB/s ingress, 2MB/s egress)	€0.0112/hr (~€9/mo)	€0.0224/hr (~€17/mo)
Publisher policies		√
Consumer groups	1 - Default	20
Maximum throughput units	20	20
Service Bus brokered connections	100 included	1000 included
Additional Service Bus brokered connections		√
Message Retention	1 day included	1 day included
Additional storage (up to 7 days)		√

<http://azure.microsoft.com/en-us/pricing/details/event-hubs/>



Stream millions of events per second into multiple applications

Enable applications to process events with variable load profiles

Connect millions of devices across platforms

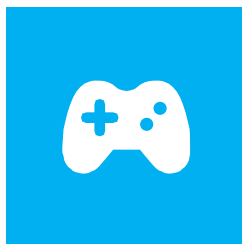
Available NOW !



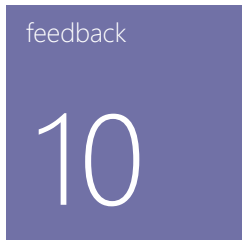
Blog <http://mircovanini.blogspot.com>



Email info@proxsoft.it



Web www.proxsoft.it



Twitter [@MircoVanini](https://twitter.com/MircoVanini)