

cdc.dev



CARIBBEAN DEVELOPERS CONFERENCE

A Top-Notch tech conference
right in the middle of paradise!



@caribbeandevconf





Clemens Vasters

Microsoft Corporation

Chief Messenger

     @clemensv

**CARIBBEAN
DEVELOPERS**
CONFERENCE

Azure Event Hubs The Architect's Cut

<https://github.com/clemensv/messaging>

Agenda

A very brief overview: “What is Azure Event Hubs?”

The Event Hubs Deep Dive

Clusters and Namespaces

Networking Capabilities

Authorization Models

Partitions and Log Storage

Message Broker Compute Model

Federation and Replication

Throughput and Latency

Observed Availability and Reliability

Q&A

What is Azure Event Hubs?

Platform-as-a-Service Event Stream Broker

Use the Apache Kafka[®] API, but with far lower cost and better performance.

Fully managed: You use the features, Azure deals with everything else

AMQP 1.0 standards compliant, Apache Kafka[®] wire-compatible

Polyglot Azure SDK and cross-platform client support

Industry-leading reliability and availability

Fast.

Azure Event Hubs vs. Apache Kafka®

Similar yet very different

	Azure Event Hubs	Apache Kafka
User Model	Partitioned event stream broker with high-availability replication	Partitioned event stream broker with high-availability replication
Architecture	Multi-tenant, 3-Tier Gateway/Broker/Storage cluster model, with tenant-isolation, all tiers independently scalable	Single-tenant monolith. Need to increase broker instances in a cluster to scale any dimension.
Implementation Language	C# and Native (C/C++)	Java
Cluster Manager	Azure Service Fabric (inline)	Apache Zookeeper (external); KRaft (inline, experimental)
Partition Mapping	Key hashing, client or server-side mapping of events	Key hashing, client-side mapping of events
Consumer Partition Ownership Coordination	Server-coordinated partition ownership (Kafka), client-coordinated ownership with external leader election. Parallel, direct partition reads.	Server-coordinated partition ownership
Server Workload Balancing	Dynamic and fully automated (100% hands-off). Broker resource allocation independent of partition count or ownership, flexible scaling.	Static assignment of partitions to broker instances requiring operator intervention for rebalancing.
Storage Model	Replicated log store, synchronous per-message flush-to-disk on all replicas	Replicated log store, asynchronous flush-to-disk controlled by host file system write cache settings.
Networking	Single endpoint access to all partitions, Public IP/DNS or Virtual Networking, Firewall.	Endpoint per broker instance. Multiple IPs required. Complex network management required.
Access Control	Token-based access policy model with unlimited publisher policies, Azure Active Directory role-based access control	Local accounts, federation extensibility.
Protocols	AMQP 1.0 (optional: WebSockets) HTTPS 1.1 Apache Kafka RPC	Apache Kafka RPC
Batching / Archives	Avro-packaged batch-packaging and archival to blob store	
Schema Registry	Schema Registry based on open CNCF Schema Registry API	(Proprietary from commercial vendors)

SKUs and Pricing

Azure Event Hubs *Standard:*

Shared-resources model

Blended capacity and consumption-based pricing.

Up to 40 MB/sec ingress:

- Throughput Units: 1TU = min(1000 msg/s, 1MB/s in), min(2000 msg/s, 2MB/s out)
- Capture
- First 12M events free, 12-25M \$x/1M, 25-50M \$x/1M, 50+ \$x/1M

Azure Event Hubs *Premium:*

Isolated-resources model

Capacity-based pricing model. No consumption charges.

Up to 120 MB/sec ingress:

- Premium Units: 1PU = ~5-10MB/sec throughput, no per-event charge, includes Capture

Azure Event Hubs *Dedicated:*

Dedicated-resources model

Capacity-based pricing model. No consumption charges. Full isolated cluster.

Beyond 120MB/sec ingress (>4 GB/sec in prod)

- Capacity Unit: 1CU \$4995/mo.
- 8CU required for availability zones.

We will discuss all these versions today and what the units mean

You're running Apache Kafka® in your own datacenter?

Azure Event Hubs is

more reliable and dependable with higher
24/365 uptime
(no service windows or weekend downtime)

far less costly to own
(no initial/recurring licensing fees as with commercial
Kafka distributions)

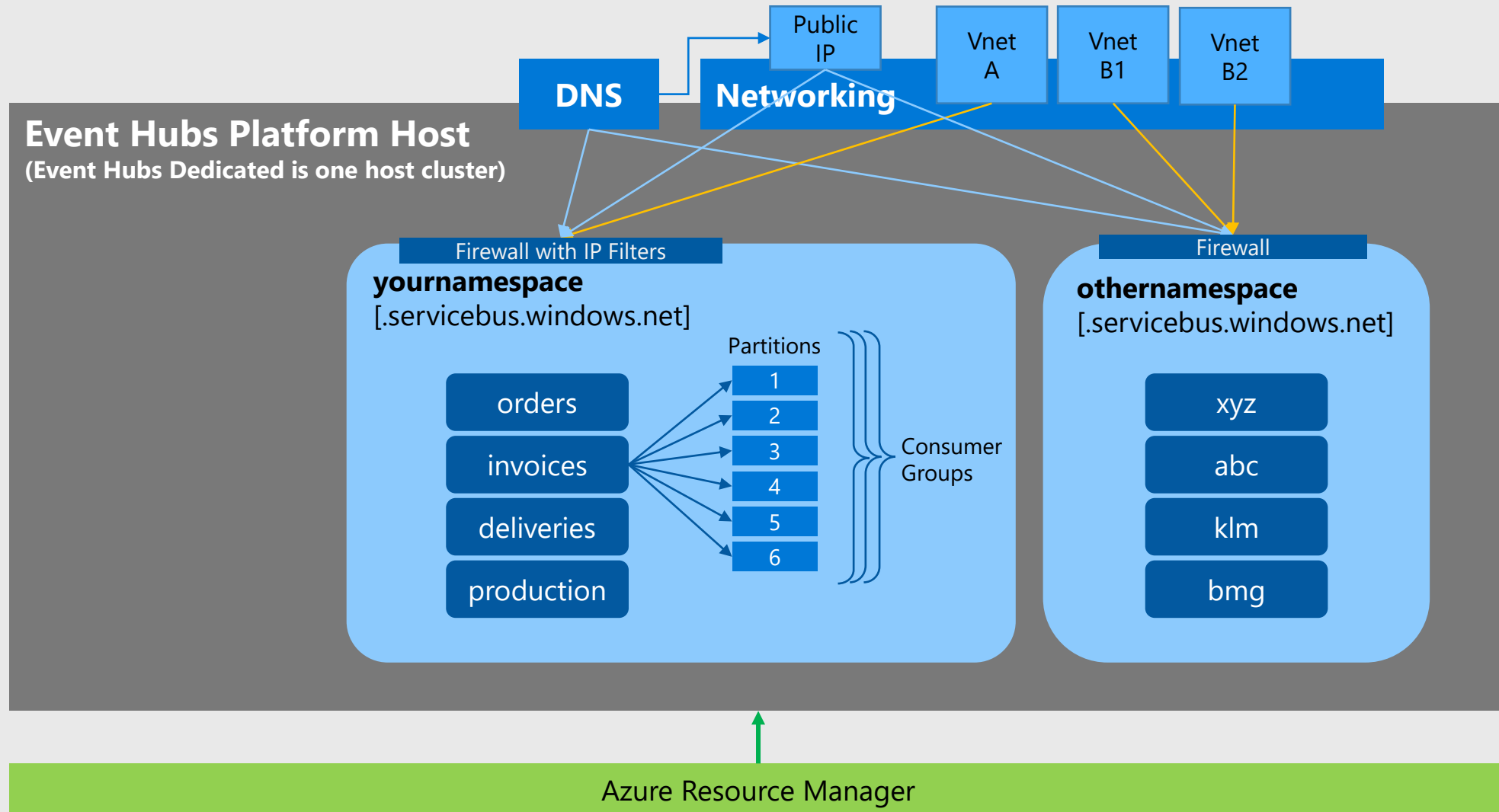
far less costly to operate
(fully managed, no hardware)

just a quick network hop away from your
existing workloads

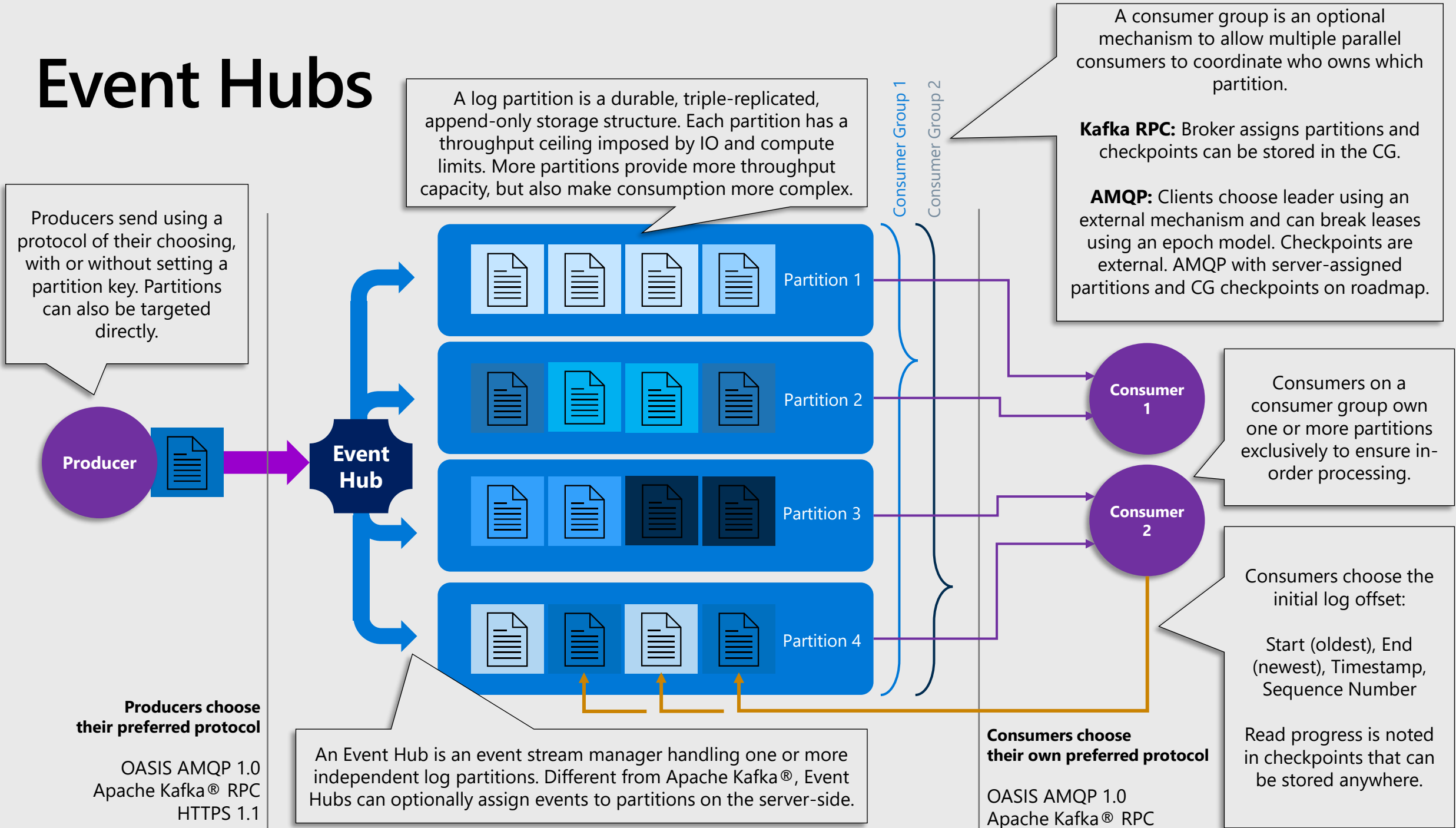
easy to integrate into your network via
VPN Gateway or Express Route



Event Hubs Namespaces: Virtual Brokers



Event Hubs

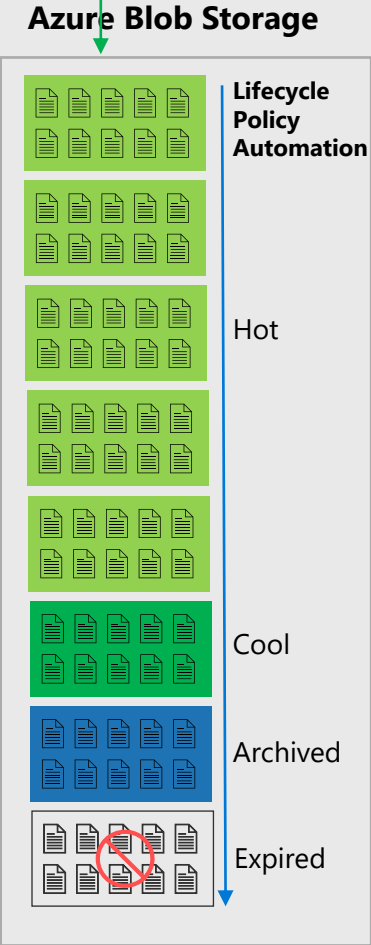
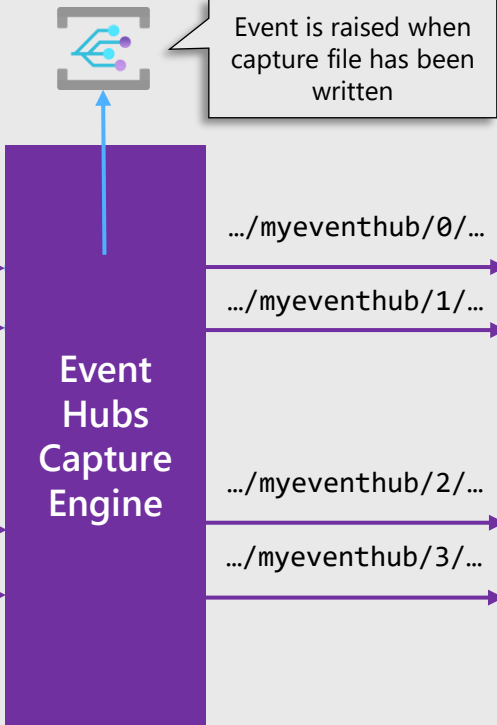
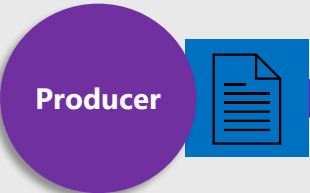


Event Hubs Capture

```
{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 Hubs Capture allows organizing torrents of telemetry in batches that can be easily consumed by batch-oriented analytics frameworks.



Producers choose their preferred protocol

OASIS AMQP 1.0
Apache Kafka® RPC
HTTPS 1.1

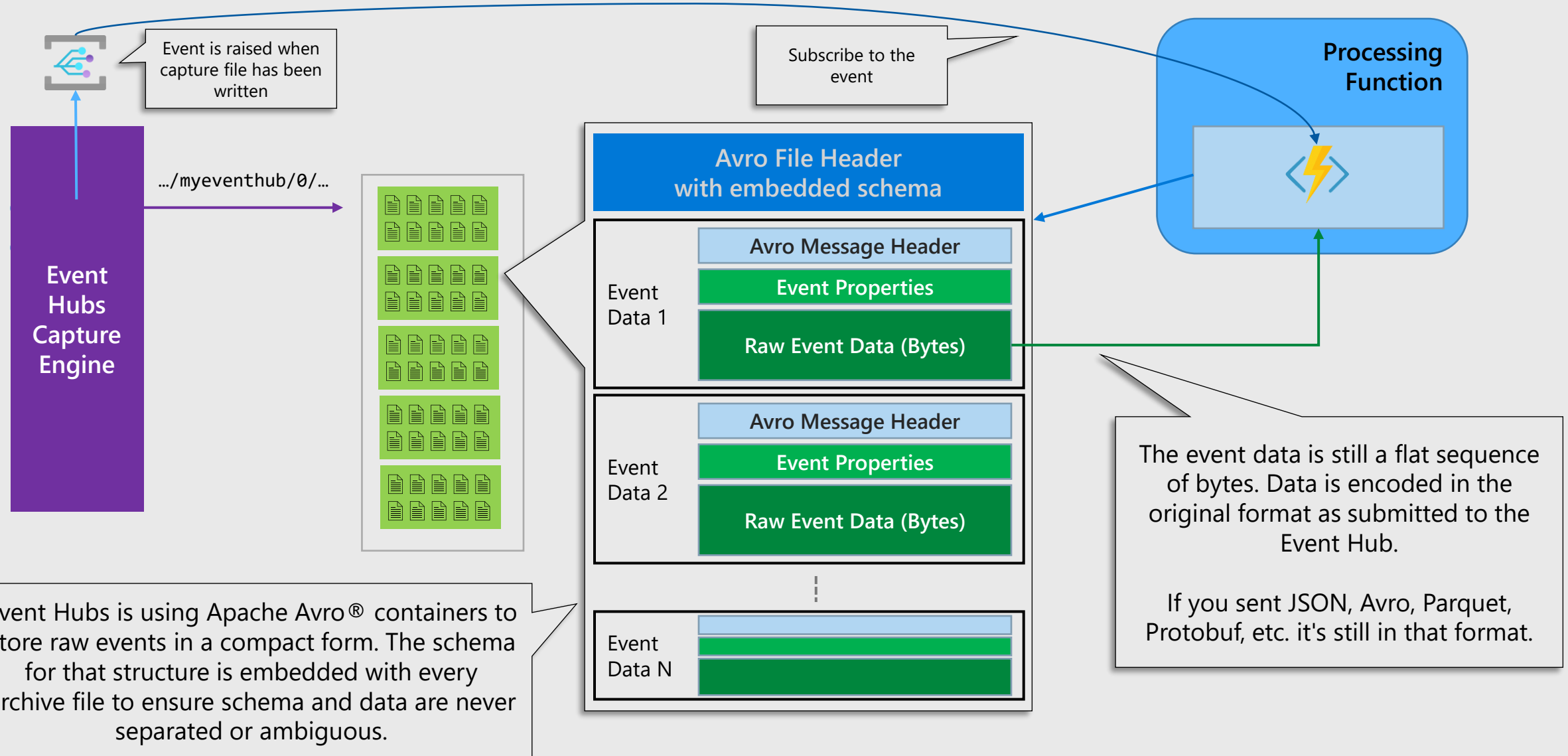
Capture throughput keeps up with the Event Hubs' ingress throughput capacity and provides automated low latency capture to batch storage.

Capture policy allows balancing latency and batch size.



Azure Data Lake Gen2

Event Hubs Capture Avro Archives



Event Hubs* is not...

... a publish/subscribe broker. Partitions are not subscriptions. They are chosen by the producer or the broker on ingress. There is also no server-side filtering.

→ [Azure Service Bus](#), [Azure Event Grid](#)

... a queue broker. Read progress over the log is handled by the client and there is no event-level ownership and delivery state handling.

→ [Azure Service Bus](#)

... a discrete event distribution engine. Event Hubs does not do push deliveries, and delivery failures need to be tracked individually.

→ [Azure Event Grid](#)

... a database or long-term event store. Event Hubs exists to catch, store and provide fast access to event data organized around time axis. As data ages (days, not months), you need better indexing.

→ [Azure Cosmos DB](#), [Azure SQL](#), [Azure Synapse](#), [Azure Data Explorer](#)

Apache Kafka® Clients

<https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=30736606>

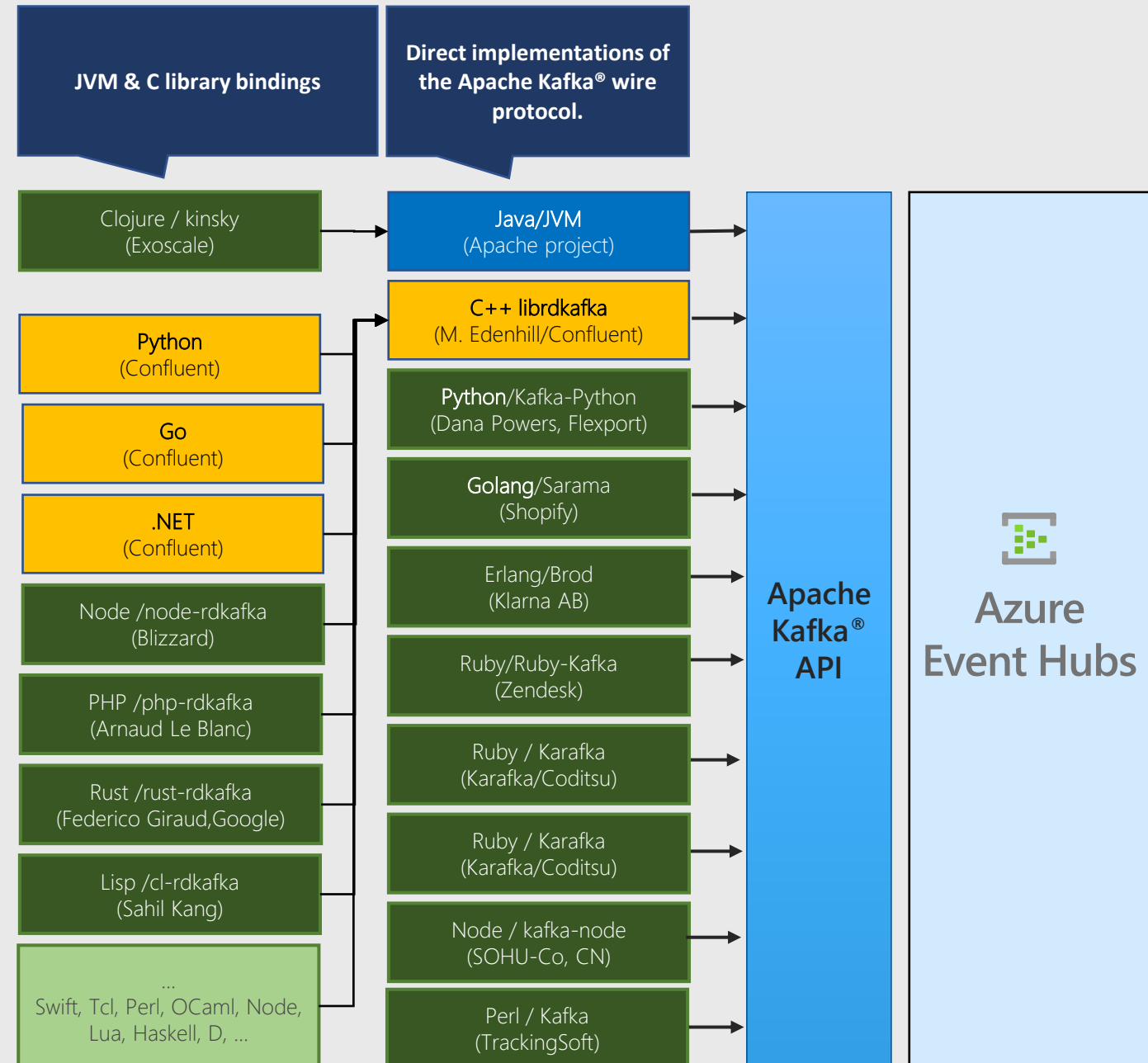
Azure Event Hubs is compatible with the producer and consumer APIs of all current Apache Kafka clients.

Only the Java/JVM client is part of the Apache Kafka project. All other clients are under various non-foundation ownerships and licenses. Azure does not provide QFE support for any of these clients but does provide support for service-side compatibility.

Apache Software Foundation

Non-Foundation, Community
/ Best Effort Support

Non-Foundation, Commercial
Support Option



Event Hubs SDK Support: Azure SDKs

.NET

Java

Java Spring Boot

Python

JavaScript/Typescript

Go

Docs / .NET /

Azure Event Hubs libraries for .NET

Article • 05/26/2021 • 2 minutes to read • 5 contributors

Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them to multiple consumers. This lets you process and analyze the massive amounts of data produced by your connected devices and applications. Once Event Hubs has collected the data, you can retrieve, transform, and store it by using any real-time analytics provider or with batching/storage adapters. If you would like to know more about Event Hubs, you may wish to review: [What is Event Hubs](#).

Java /

Azure Event Hubs client library for Java - Version 5.11.1

Azure / Developer / Java / Spring integration /

How to use the Spring Boot Starter for Apache Kafka with Azure Event Hubs

Article • 03/04/2022 • 8 minutes to read • 12 contributors

This article demonstrates how to configure a Java-based Spring Cloud Stream Binder created with the Spring Cloud Stream Binder to use [Apache Kafka](#) with Azure Event Hubs.

Prerequisites

Azure / Developer / Java / Spring integration /

How to create a Spring Cloud Stream Binder application with Azure Event Hubs

Article • 03/04/2022 • 7 minutes to read • 11 contributors

This article demonstrates how to configure a Java-based Spring Cloud Stream Binder application created with the Spring Cloud Stream Binder to use [Apache Kafka](#) with Azure Event Hubs.

Docs /

Azure Event Hubs client library for Python - Version 5.7.0

Article • 01/14/2022 • 13 minutes to read • 5 contributors

Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them to multiple consumers. This lets you process and analyze the massive amounts of data produced by your connected devices and applications. Once Event Hubs has collected the data, you can retrieve, transform, and store it by using any real-time analytics provider or with batching/storage adapters. If you would like to know more about Event Hubs, you may wish to review: [What is Event Hubs](#).

The Azure

- Emit
- Publish
- Subscribe
- Observe

Microsoft Azure Event Hubs Client for Golang

go report [A+](#) [reference](#) [build](#) [failing](#) [coverage](#) [72%](#)

Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them into multiple applications. This lets you process and analyze the massive amounts of data produced by your connected devices and applications. Once Event Hubs has collected the data, you can retrieve, transform and store it by using any real-time analytics provider or with batching/storage adapters.

Docs /

Azure Event Hubs client library for JavaScript - Version 5.7.0

Article • 01/14/2022 • 13 minutes to read • 5 contributors

Azure Event Hubs is a highly scalable publish-subscribe service that can ingest millions of events per second and stream them to multiple consumers. This lets you process and analyze the massive amounts of data produced by your connected devices and applications. Once Event Hubs has collected the data, you can retrieve, transform, and store it by using any real-time analytics provider or with batching/storage adapters. If you would like to know more about Azure Event Hubs, you may wish to review: [What is Event Hubs](#).

This article demonstrates how to send and receive events in your Node.js application.

Compatible Generic AMQP 1.0 Clients

Client Stack	URL
Apache Qpid Proton-C (Go, Python, Ruby, C+ +)	https://qpid.apache.org/proton/
Apache Qpid Messaging API	https://qpid.apache.org/components/messaging-api
Apache Qpid Proton-J	https://qpid.apache.org/proton/
Apache NMS AMQP	https://activemq.apache.org/components/nms/providers/amqp/
AMQP .NET Lite (all variants of .NET, incl. Nano & Micro)	https://github.com/Azure/amqpnetlite
Azure AMQP (our own server stack)	https://github.com/Azure/azure-amqp
Azure uAMQP C (Python, PHP)	https://github.com/Azure/azure-uamqp-c https://github.com/Azure/azure-uamqp-python
Rhea (NodeJS)	https://github.com/amqp/rhea
Go AMQP	https://github.com/Azure/go-amqp
Vert.X AMQP Client	https://vertx.io/docs/vertx-amqp-client/java/

Azure Event Hubs Deep Dive

Architecture

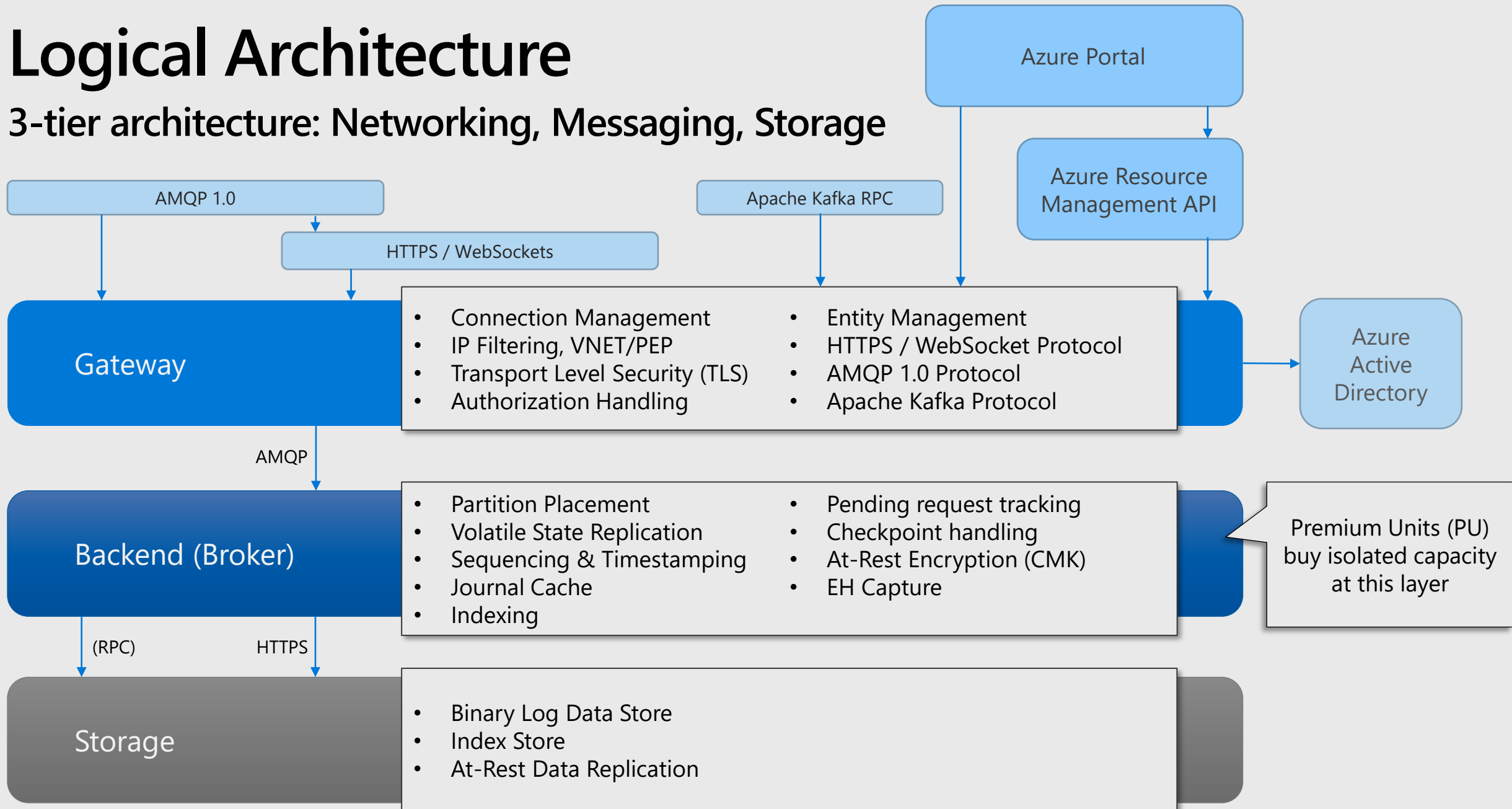
Internal Architecture

Protocols

Performance Metrics

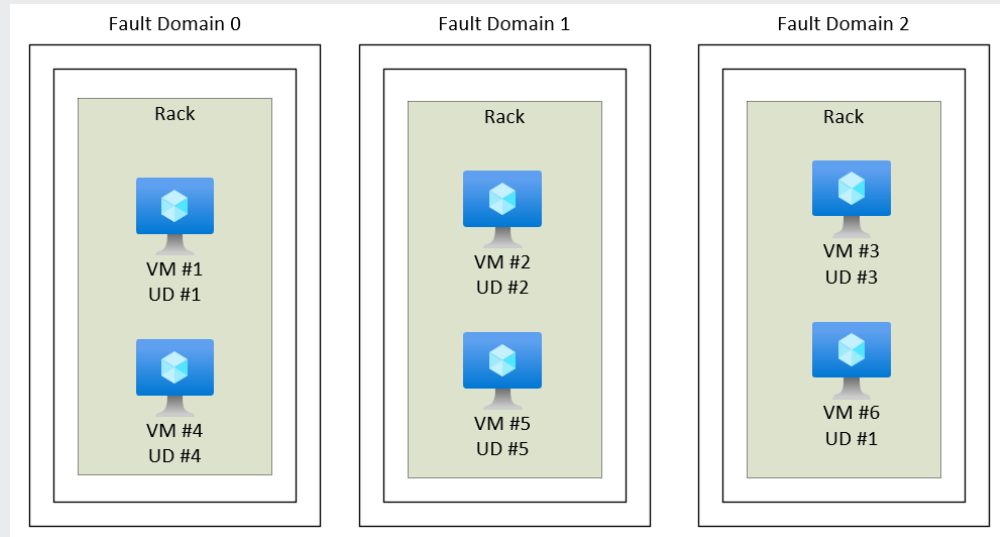
Logical Architecture

3-tier architecture: Networking, Messaging, Storage



Fault Domains & Availability Zones

High-Availability starts at the physical reality



Fault Domain Placement:

Cluster VMs are spread across at least 3 fault domains such that the loss of a rack or network poses no availability risk. Recovery from a fault domain failure is fully automated and the system maintains SLA.

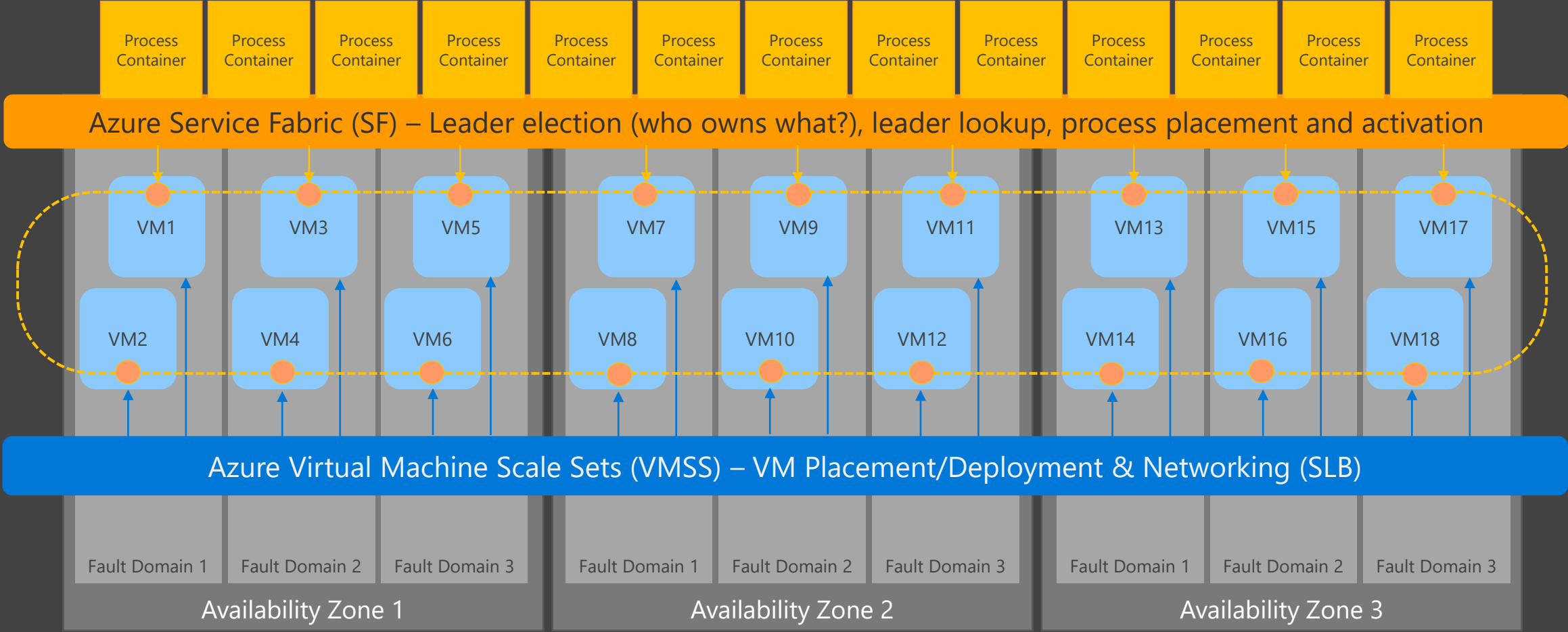


Availability Zones Placement:

Each cluster spans three availability zones and maintains SLA without any tolerance for data loss when one or two zones fail.

Backend and Gateway Clusters

Logical Architecture meets Placement



Backend and Gateway Clusters

Principles

Everything is automated. There's no manual intervention in placement of VMs or processes or workloads.

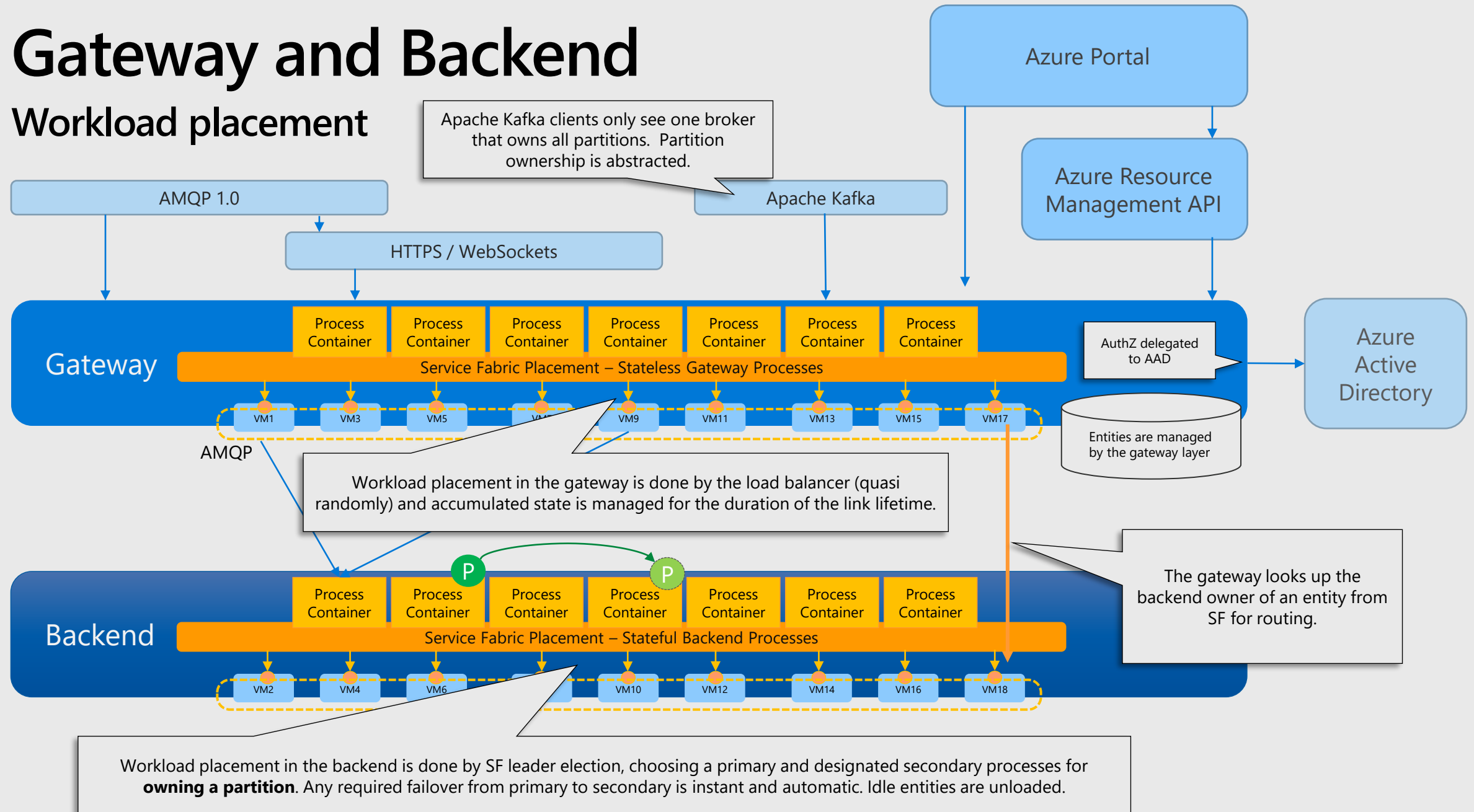
Each cluster has a well-known number of virtual machines. We don't use auto-scaling with VMs. We may (rarely) choose to initiate scale up/down to a different level.

Process placement and servicing/upgrade cycles are controlled by Azure Service Fabric.

Allocation of namespaces to clusters are based on well-established heuristics (Standard) or deterministic capacity limits (Premium). Customer-defined allocation in Dedicated.

Gateway and Backend

Workload placement



Backend: Event Hubs Premium.

Workload placement

Namespace PU are split across processes:

- 1 PU = 2 Proc (8GB Mem), 1 Core/Proc (2C)
- 2 PU = 2 Proc, 1 Core/Proc + 1 Core (3C)
- 4 PU = 2 Proc, 2 Core/Proc + 1 Core (5C)
- 8 PU = 4 Proc, 2 Core/Proc + 1 Core (9C)
- 16 PU = 4 Proc, 4 Core/Proc + 1 Core (17C)

Cores are exclusively mapped to a broker process.

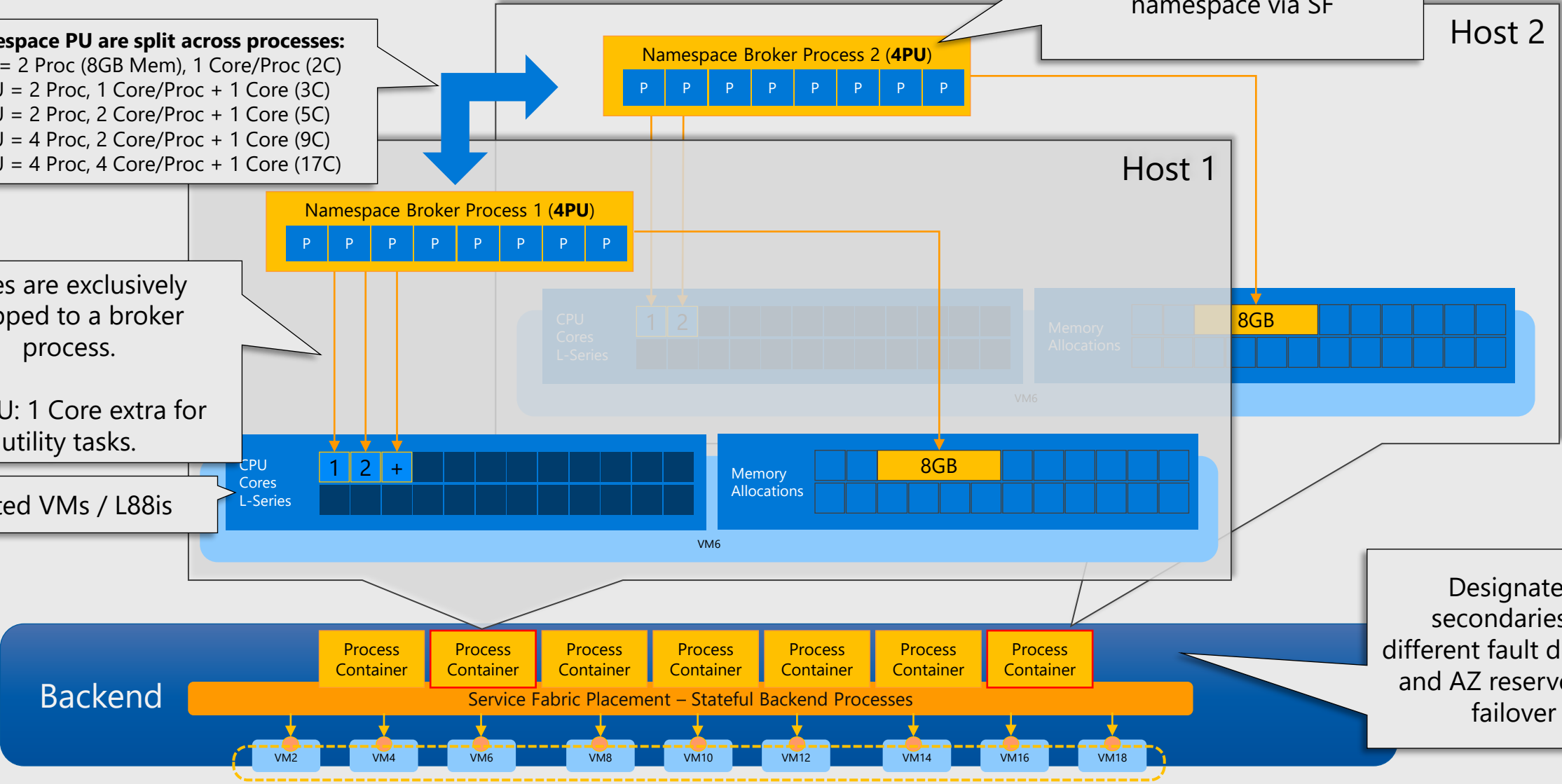
>=2PU: 1 Core extra for utility tasks.

Isolated VMs / L88is

Partition ownership is dynamically mapped to the process(es) associated with a namespace via SF

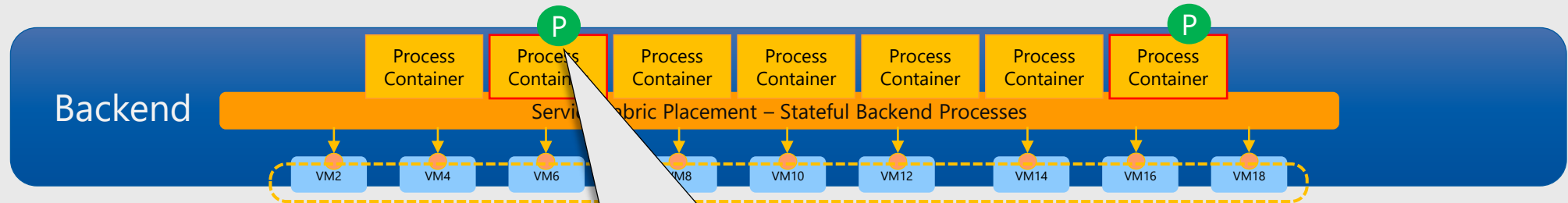
Host 2

Host 1



Designated secondaries in different fault domains and AZ reserved for failover

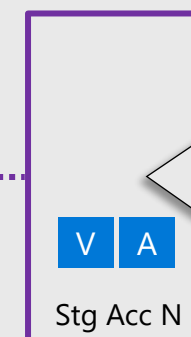
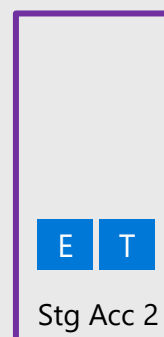
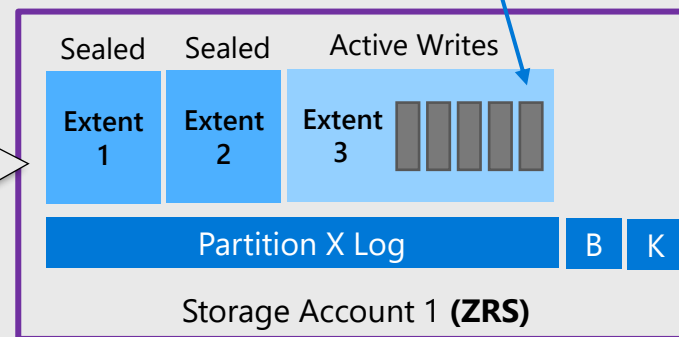
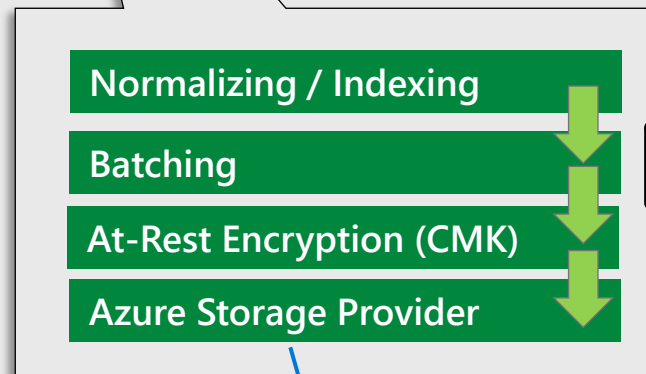
Storage Layer – Event Hubs Standard



Events are written to append blobs ("extents"). All writes are synchronously triple replicated across availability zones.

Extents are sealed after hitting a size threshold.

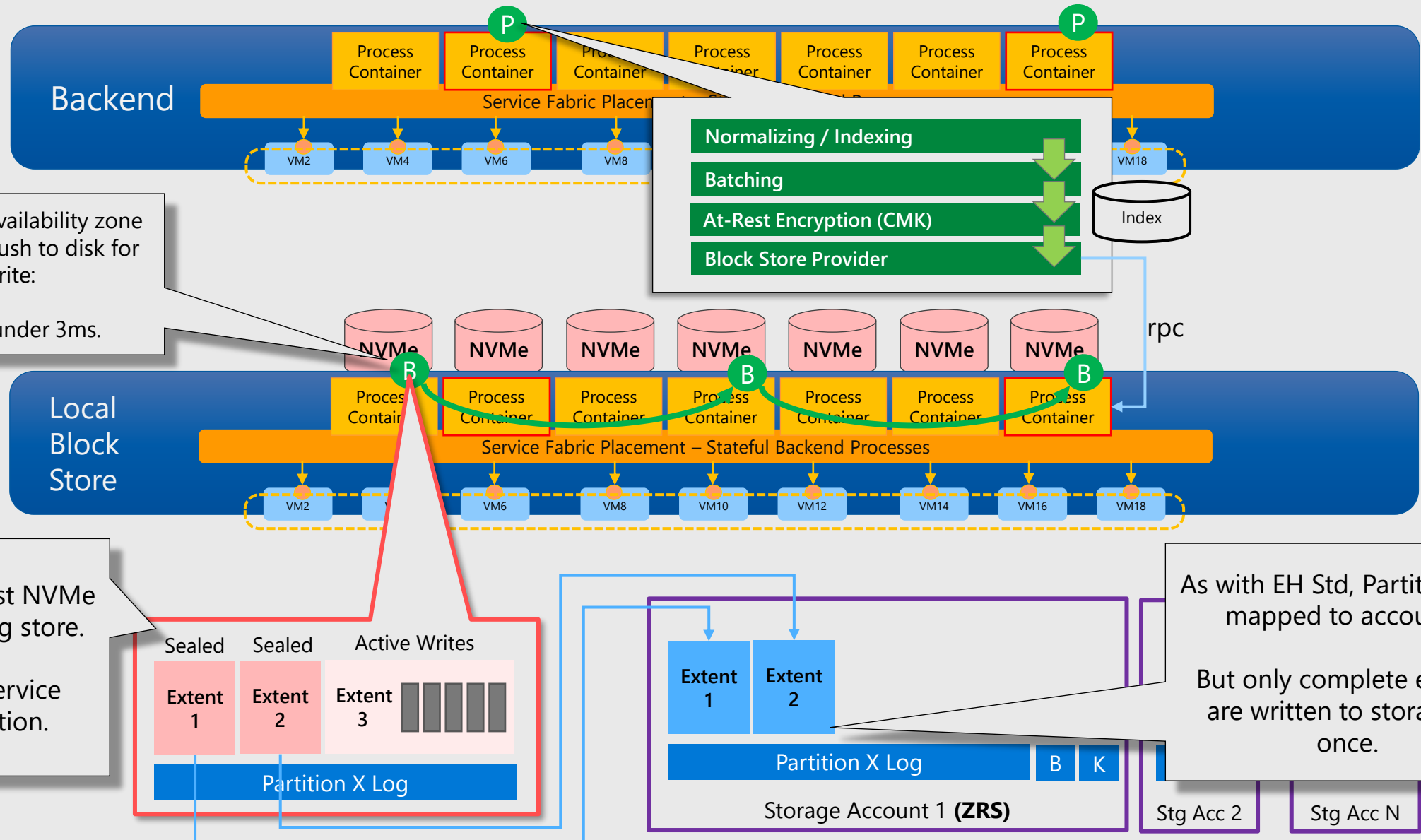
Expiry removes extents.



Each Event Hub Cluster uses numerous storage accounts to spread the load and overcome storage quota limitations.

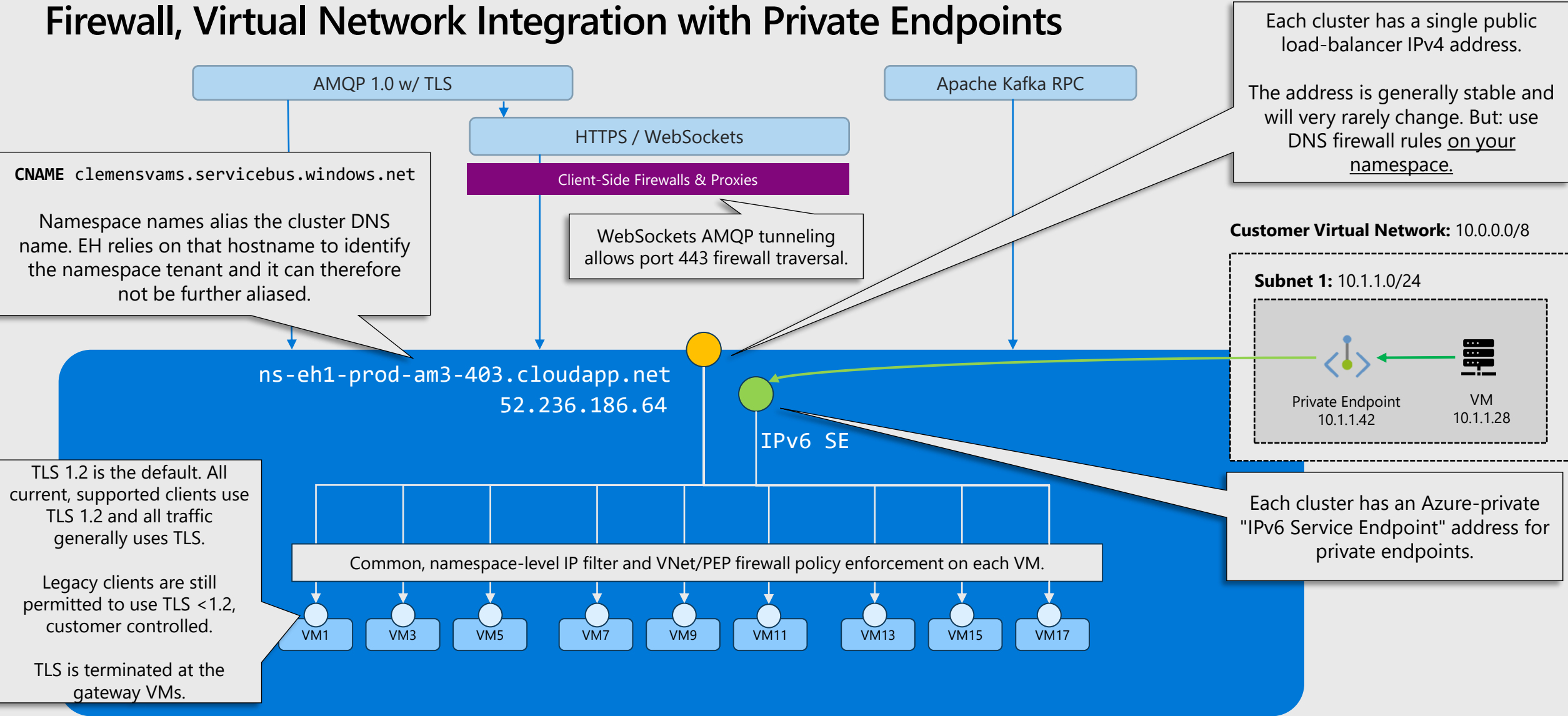
Partition storage is allocated to those storage accounts.

Storage Layer – Event Hubs Premium



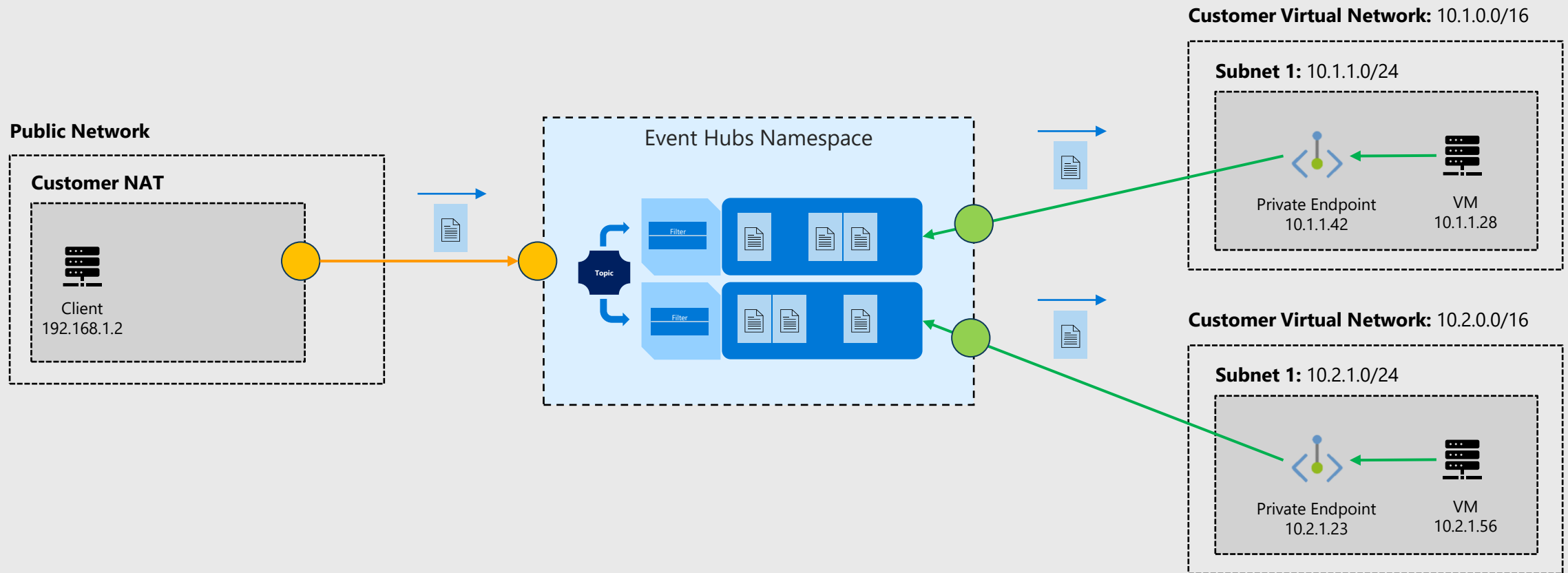
Networking Features

Firewall, Virtual Network Integration with Private Endpoints



Networking Features

Event Hubs namespaces can be attached to one or more virtual networks and the public IP address space concurrently and act as safe "Layer 7" stream bridges.



Authorization

Azure Active Directory RBAC and Shared Access Signatures (SAS)

Event Hubs has two authorization models.

SAS (Local): Simple model for external clients that cannot use Azure Active Directory for any reason.

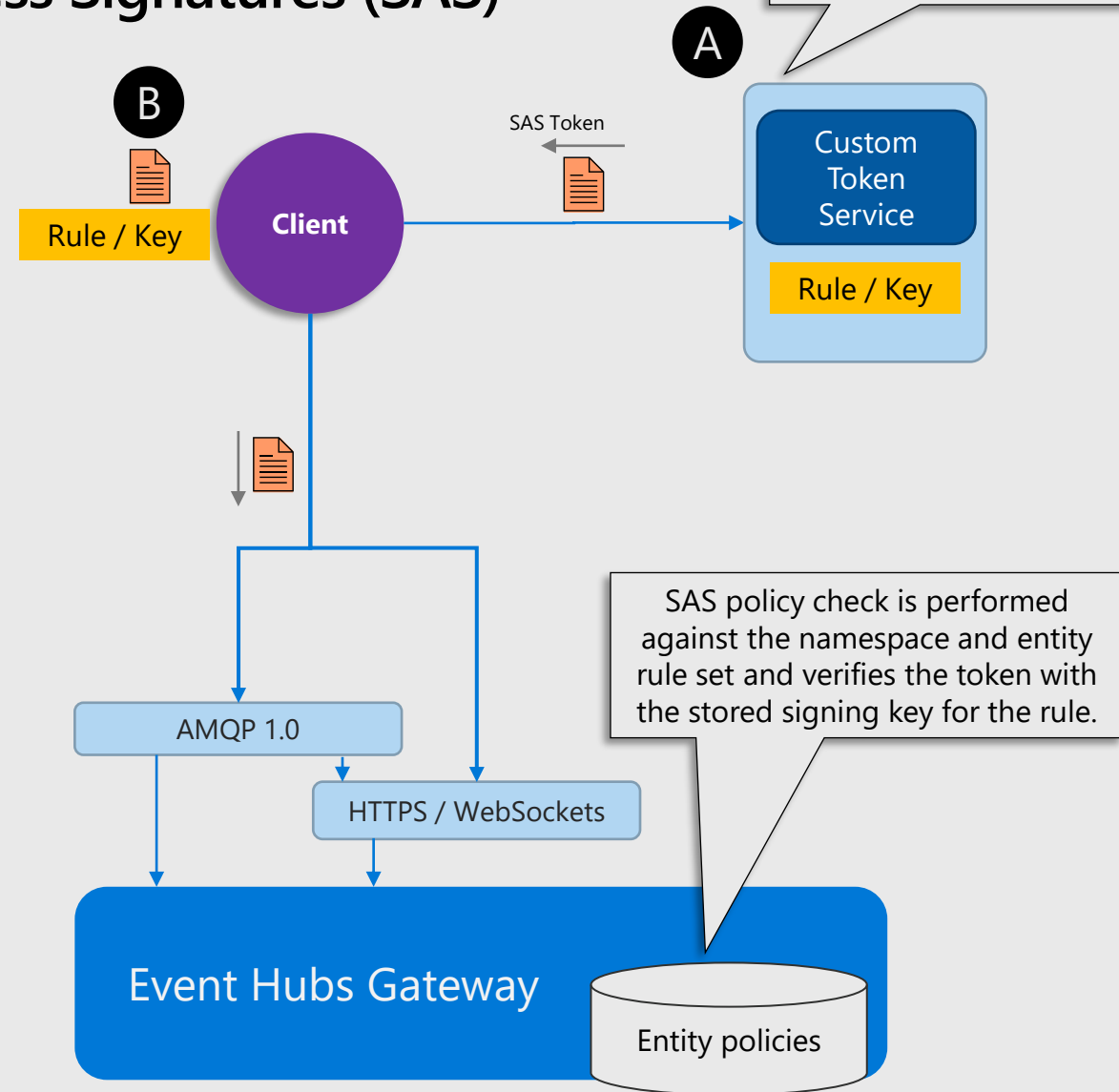
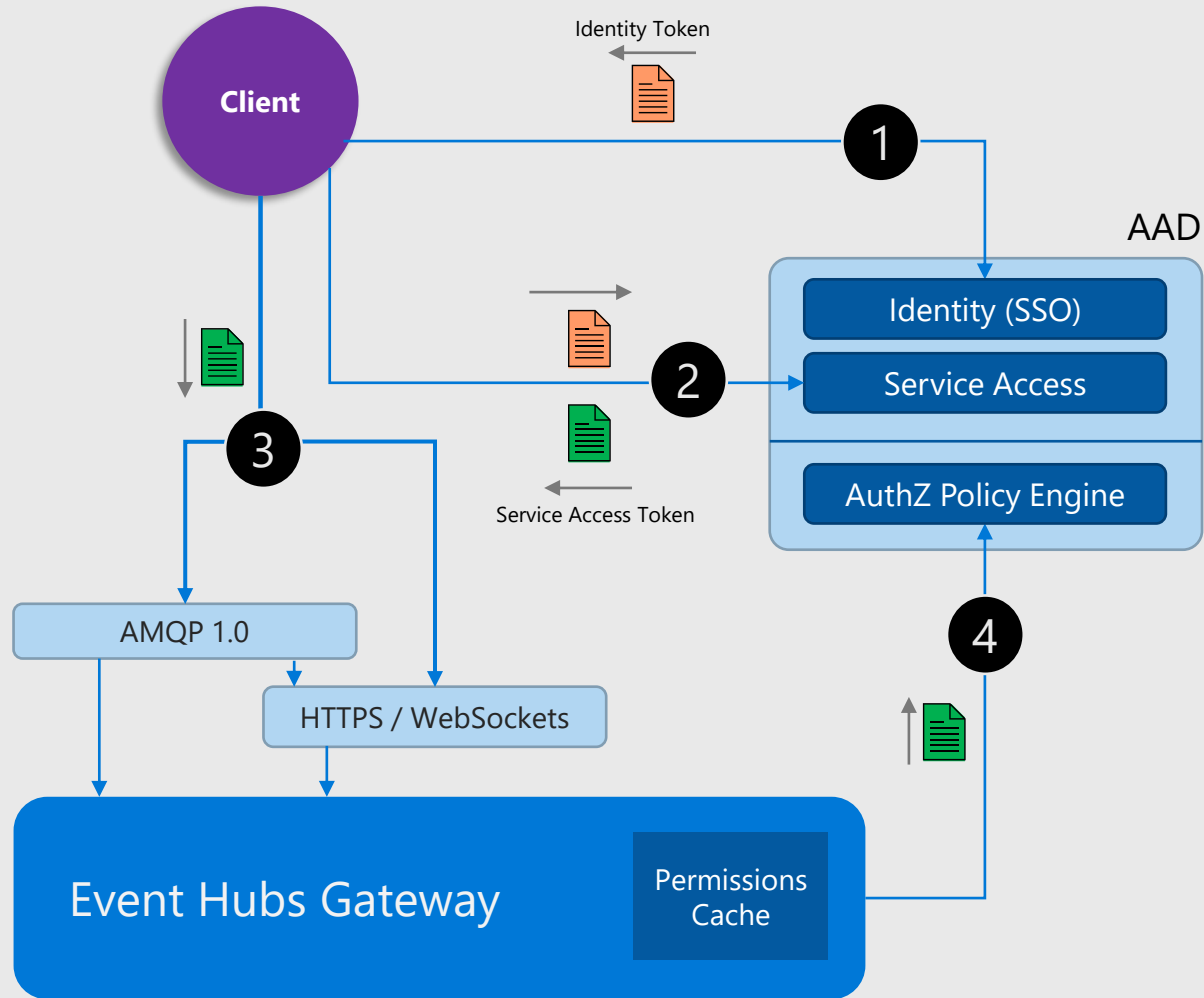
- Very simple. Rights at namespace or entity scope: "Send", "Listen" (Receive/Subscribe), "Manage"
- No accounts. 12 named rules per scope, each rule has a key. Rules combine a set of rights.
- HMACSHA256 signed **tokens** are issued using name and key and passed to the service.
- Tokens can be issued in the client or by some security token service holding the key
- *Can be turned off via the portal on the overview page under "Local Authentication"*

AAD RBAC: Integrated Azure identity model, also for service-level integration (managed identity)

- Very detailed, operation-level permission set that can be assigned to roles
- Several standard control- and data-plane roles (like Data Owner, Data Sender, Data Receiver)
- Roles can be assigned at the namespace and entity level

Authorization

Azure Active Directory RBAC and Shared Access Signatures (SAS)



Event Hubs Premium

Replication & Federation



Replication Tasks

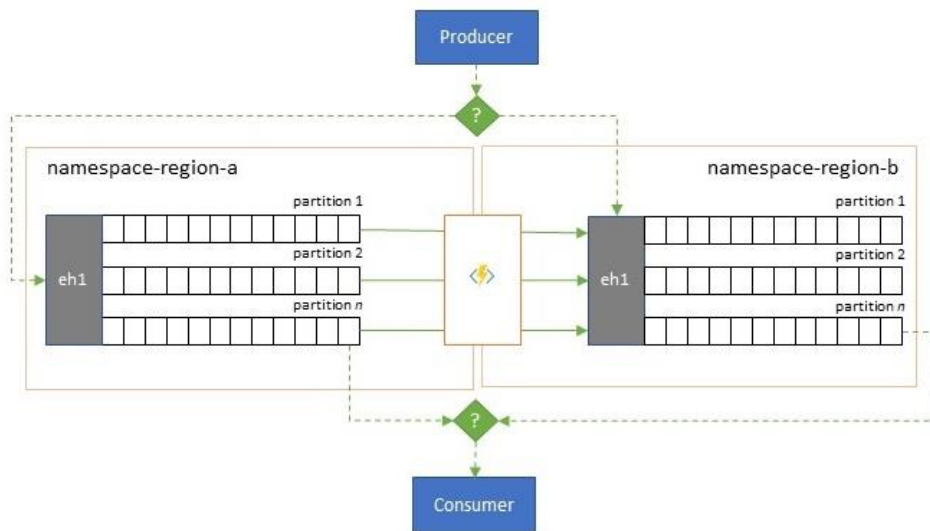
Providing code-free replication of data between Service Bus and Event Hubs

Problem

In many scenarios, event data needs to be moved or merged across regions.

Solution

Quick and simple replication between namespaces through templates available on Azure Portal's Tasks.



Automation Tasks to the rescue

Create

Select
template

Establish
connection

Review


Configure


Dashboard > Event Hubs > clemensvamsp >

Add a task

1 Select a template 2 Authenticate 3 Configure 4 Review + create

Select an automation template.

**Send monthly cost for resource**
Sends the cost of the resource every month.
[Select >](#)

**Replicate from Event Hub to Event Hub**
Replicate content from an Event Hub to another Event Hub
[Select >](#)

Dashboard > clemensvamsp >

Add a task

1 Select a template 2 Authenticate 3 Configure 4 Review + create

Selected template: Replicate from Event Hub to Event Hub

This task will connect to the following services. Select the connection to add your

Name	Status	Cor
Event Hubs (source)	Connected	Usi
Event Hubs (target)	Not connected	Cre

Connections

Display Name *
amr

Event Hubs Namespace *
clemensvamright

[Connect via connection string](#)

[Review + create](#) [Previous](#) [Next : Configure >](#)

Dashboard > clemensvamsp >

Add a task

1 Select a template 2 Authenticate 3 Configure 4 Review + create

Selected template: Replicate from Event Hub to Event Hub

Task name *
mytask

Source event hub instance *
eh1

Target event hub instance
replgroup

Target event hub instance *
eh2

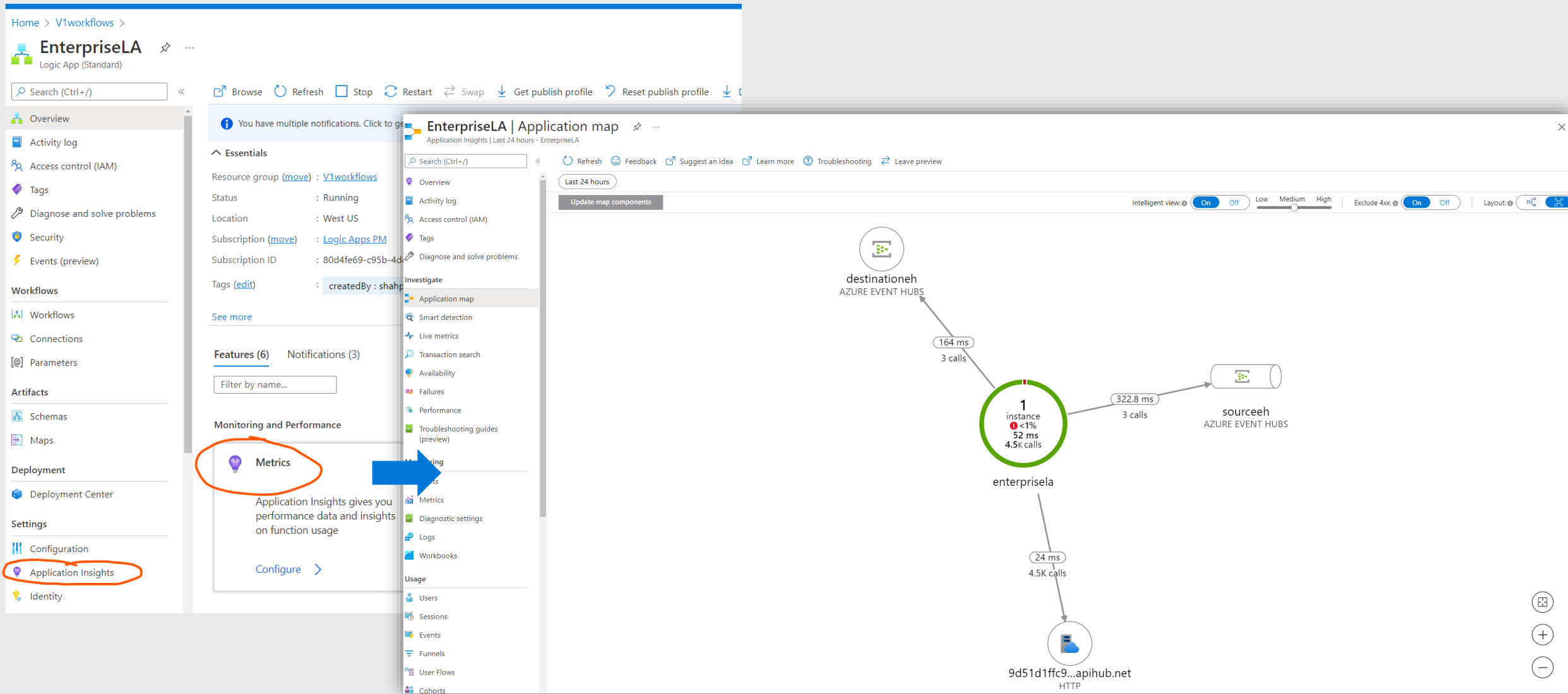
This task is powered by Logic Apps (Standard). [Learn more](#)

Logic App *
telemetrycopy
[Create new](#)

This task is billable. More information on pricing could be found here. [Learn more](#)

[Review + create](#) [Previous](#) [Next : Review + create >](#)

EventHub replication visualization



Home > devreldemoeventhub >

devreleventhub (devreldemoeventhub/devreleventhub)

Event Hubs Instance

Search (Ctrl+/)


+ Consumer group Delete Refresh

- Overview
- Access control (IAM)
- Diagnose and solve problems
- Settings
 - Shared access policies
 - Properties
 - Locks
- Entities
 - Consumer groups
- Features
 - Capture
 - Process data
- Automation
 - Tasks (preview)
 - Export template
- Support + troubleshooting
 - New Support Request

Essentials


Resource group (move)	: DBGGroup	Status	: Active
Location	: West US	Namespace	: devreldemoeventhub
Subscription (move)	: Visual Studio Enterprise	Created	: Monday, July 18, 2022 at 21:32:42 PDT
Subscription ID	: feaa8ca8-ba47-40fb-b2ce-7ddf1b49258b	Updated	: Monday, July 18, 2022 at 22:13:29 PDT
Partition Count	: 1	Message Retention	: 1 day

[JSON View](#)




Capture events

Use Capture to save your events to persistent storage.




Process data

Process data instantly with Azure Stream Analytics using no-code editor or SQL.



Connect

Authenticate with connection strings and SAS policies.



Checkpoint

Create consumer groups to checkpoint your events.

Event Hub Contents

1 CONSUMER GROUP

Event Hub Status

ACTIVE

Message Retention

1 DAY

Partition Count

1

Show data for the last: [1 hour](#) [6 hours](#) [12 hours](#) [1 day](#) [7 days](#) [30 days](#)

Requests

600

Messages

100

Throughput

100B

Process Event Hub data with Azure Stream Analytics

Process your Event Hub data using no-code drag and drop experience. (Preview)



Filter and ingest to Synapse SQL

- Preview Event Hub data
- Decide table schema
- Select Synapse SQL table

Start



Capture data to ADLS Gen2 in Parquet format

- Save events to ADLS Gen2 in Parquet
- Specify a time or size interval

Start



Materialize data in Cosmos DB

- Maintain a view of your data in Cosmos DB
- Select the fields to group by
- Define aggregations like count, sum, average
- Set a time period

Start



Filter and ingest to ADLS Gen2

- Preview Event Hub data
- Decide table schema
- Select ADLS Gen2 account

Start



Start with a blank canvas

- View incoming data and define schema
- Define transformations on your input data
- Select output to egress streaming data

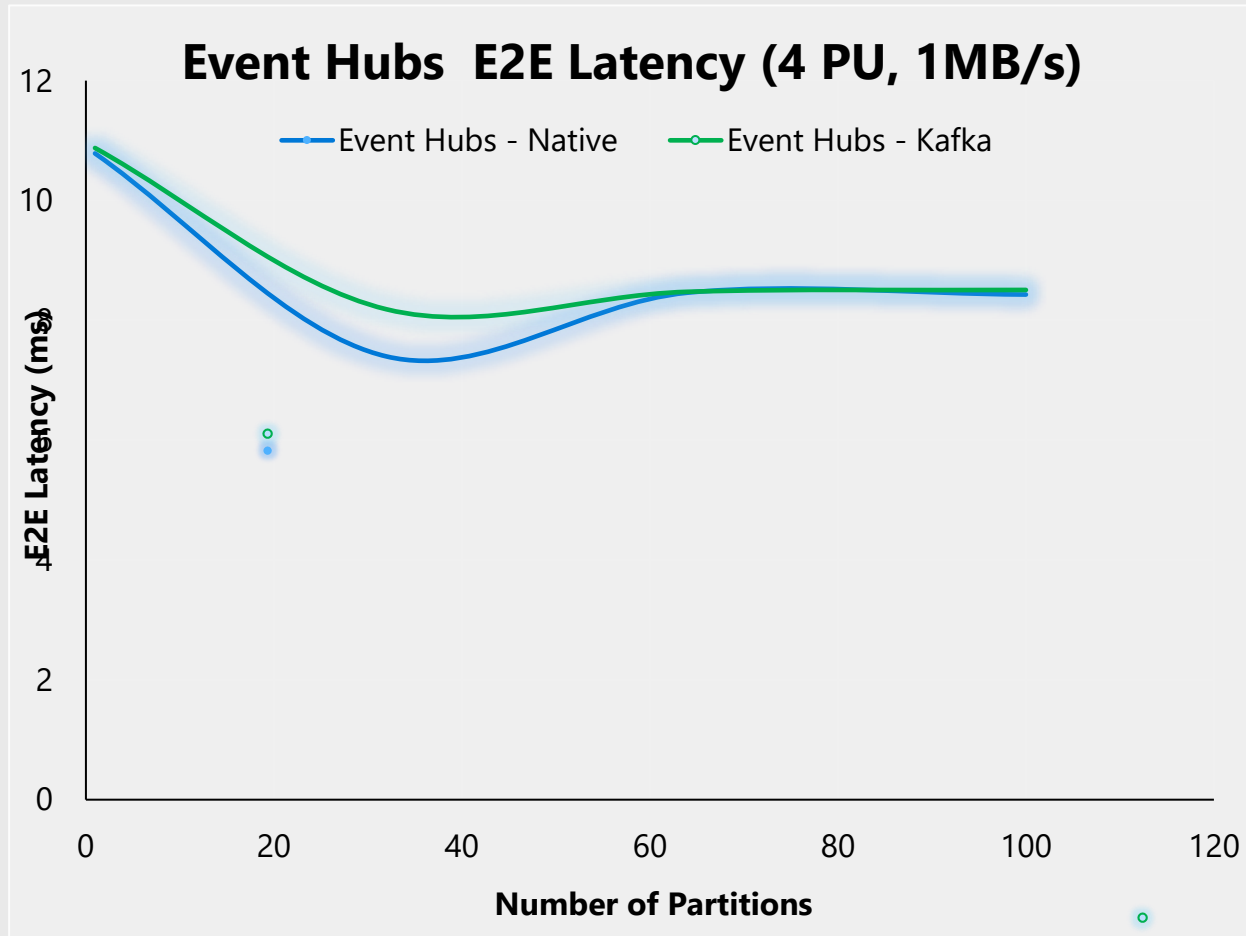
Start

Event Hubs Premium

Performance!



Event Hubs Premium is *Fast*!

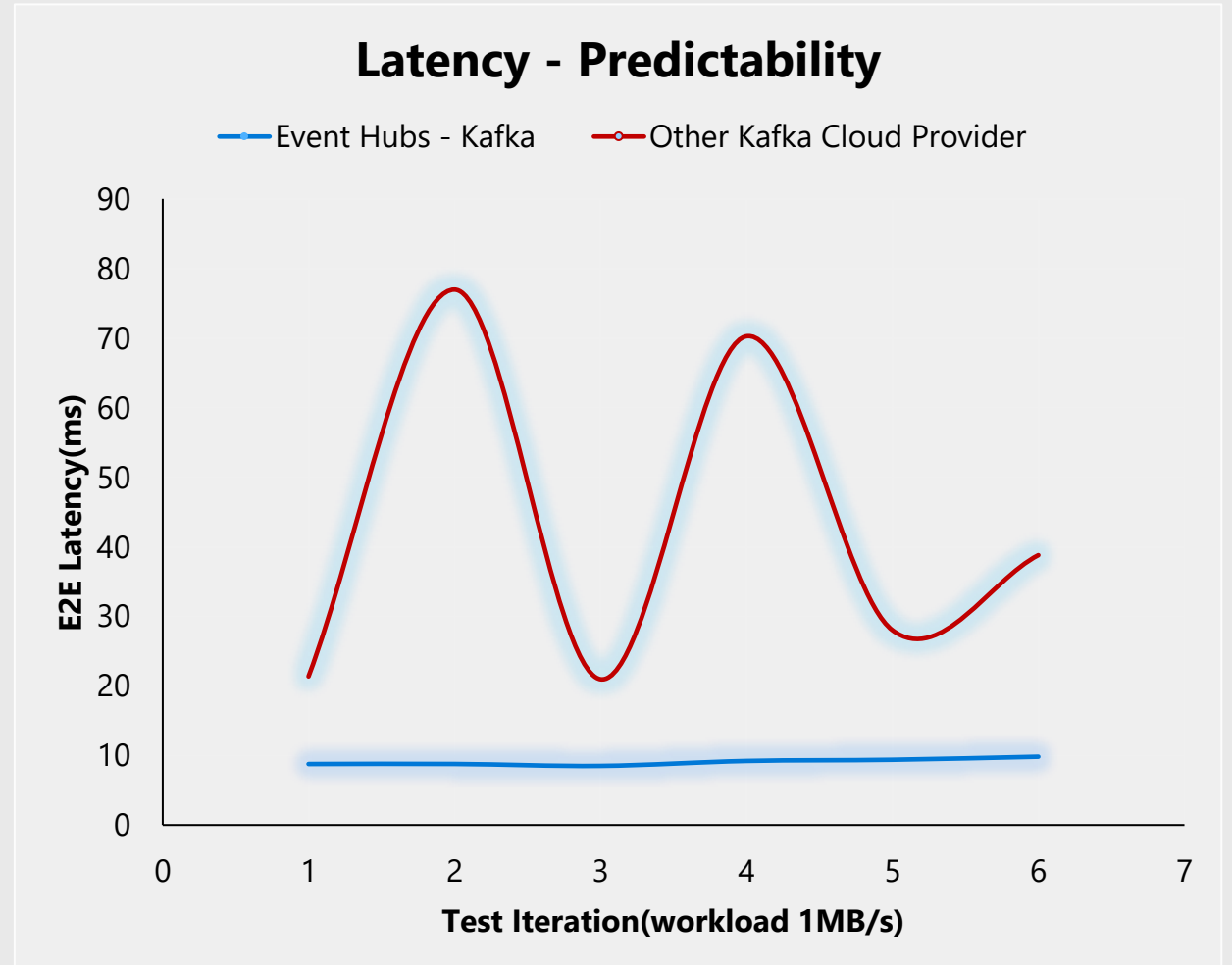


Event Hubs Premium end-to-end latency < 10ms for most Event Streaming workloads.

Performance Metric	Expected Range	Comments
Throughput	~5-10 MB/s per 1 PU	Maximum throughput can be achieved by scaling partitions For both Kafka and AMQP
Latency	< 10 ms	With 1 MB/s load, 4 PU namespace For both Kafka and AMQP

Event Hubs Premium – Predictability

Event Hubs Premium offers consistent end-to-end latency of < 10ms.



Effective Global Reliability

Uptime – Is the cluster available to accept connections?

100%

Reliability – Are individual operations succeeding?

>99,9999% weekly average

Azure Event Hubs

Platform-as-a-Service Event Stream Broker

Use the Apache Kafka[®] API, but with far lower cost and better performance.

Fully managed: You use the features, Azure deals with everything else

AMQP 1.0 standards compliant, Apache Kafka[®] wire-compatible

Polyglot Azure SDK and cross-platform client support

Industry-leading reliability and availability

Fast.

Thank You!

**CARIBBEAN
DEVELOPERS**
CONFERENCE

That was a lot, wasn't it?

Q&A



