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

Flow Architectures with Azure Event Hubs

https://github.com/clemensv/messaging

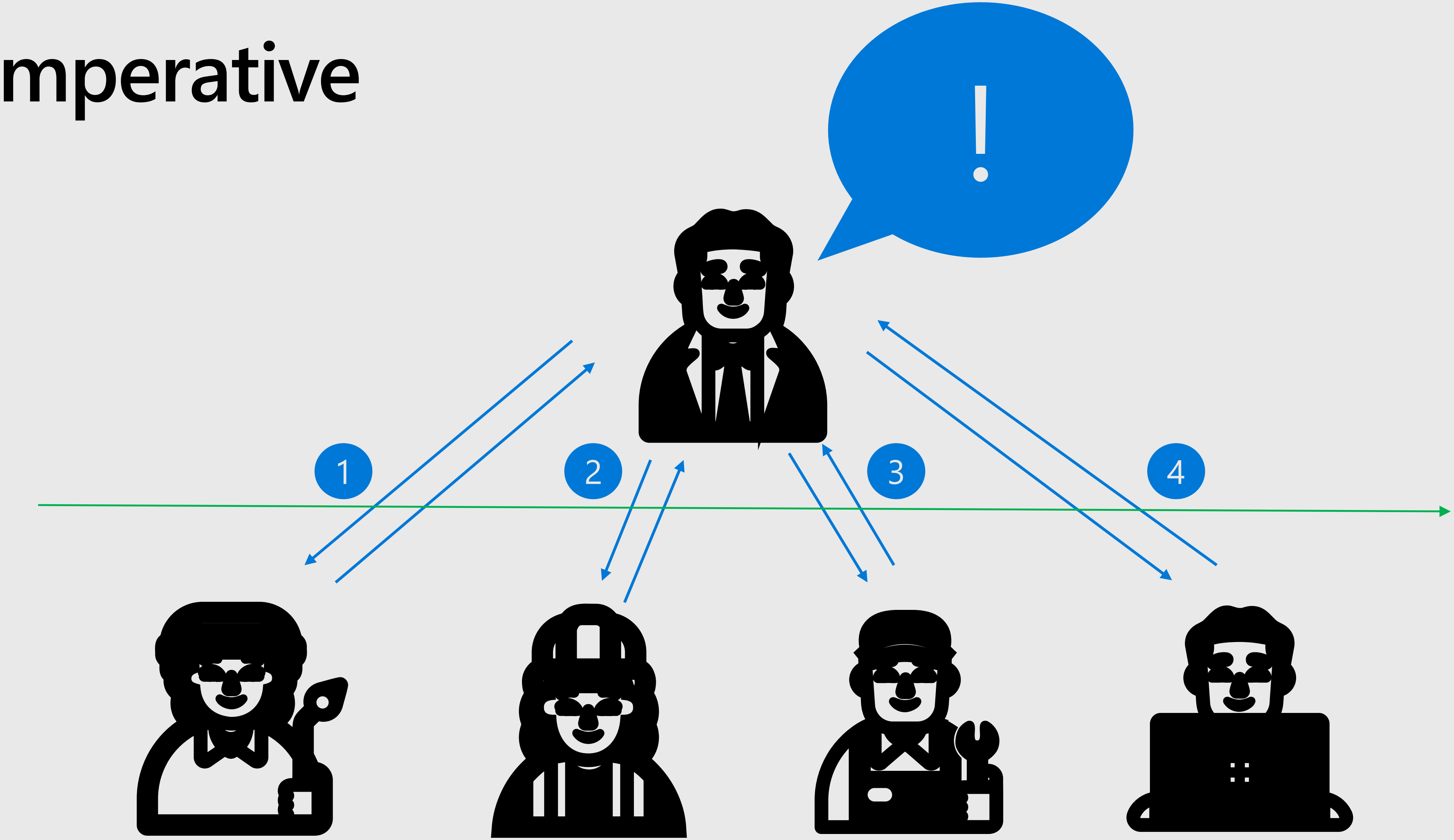
Event-Driven Flow

Observe, Decide, Act, Tell

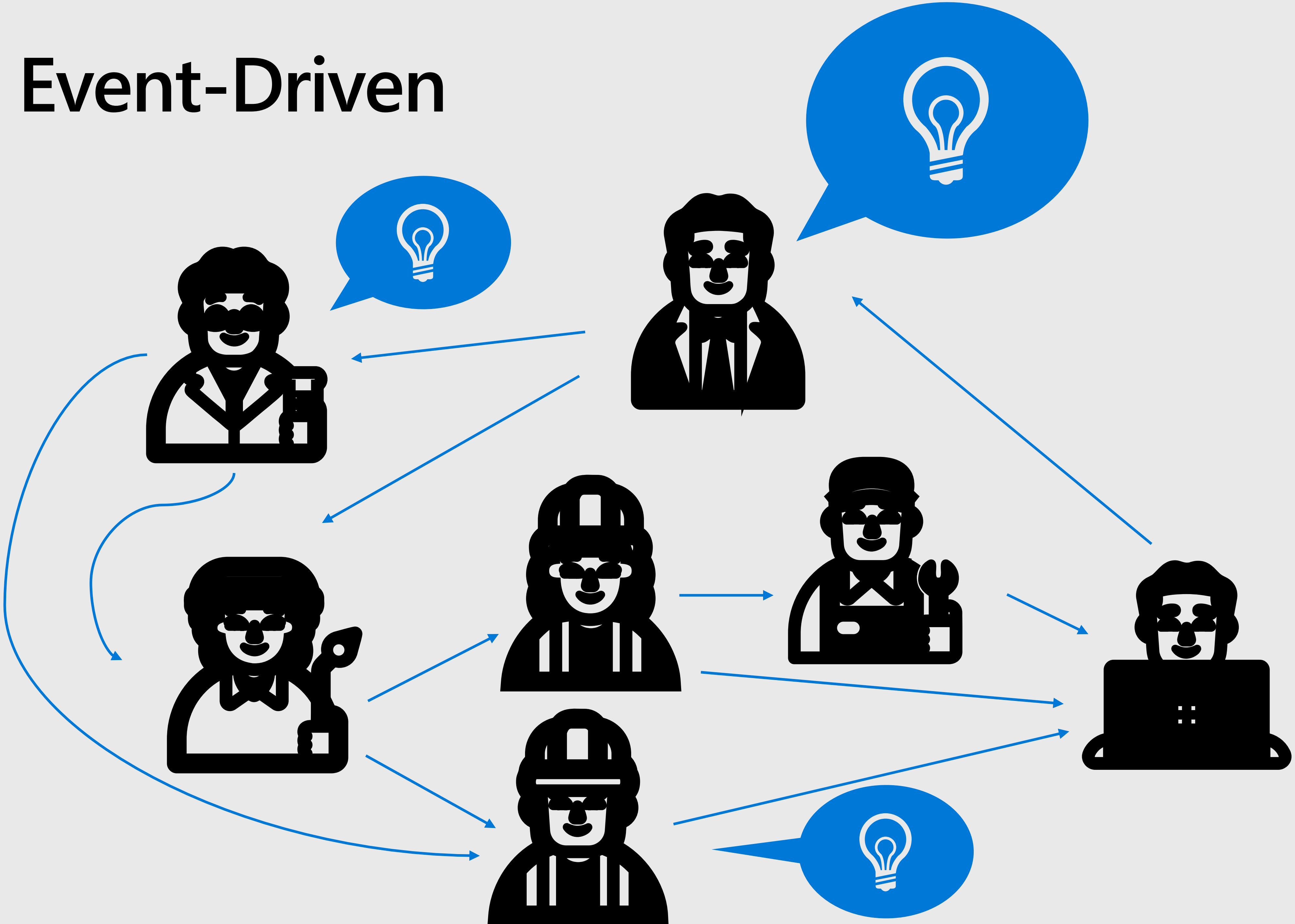
Imperative Flow

Listen, Obey, Execute, Respond

Imperative



Event-Driven



Imperative	Event-Driven
Execution plan with predetermined actor roles	No predetermined execution plan or actors.
Work is assigned, executed, and completion reported.	Work is volunteered reacting to events and completion is shared
Changes and extensions require modifying the execution plan.	Changes and extensions are local to actors or add/remove actors
Command & Control	Dynamic Collaboration

Most complex systems require both models in different areas

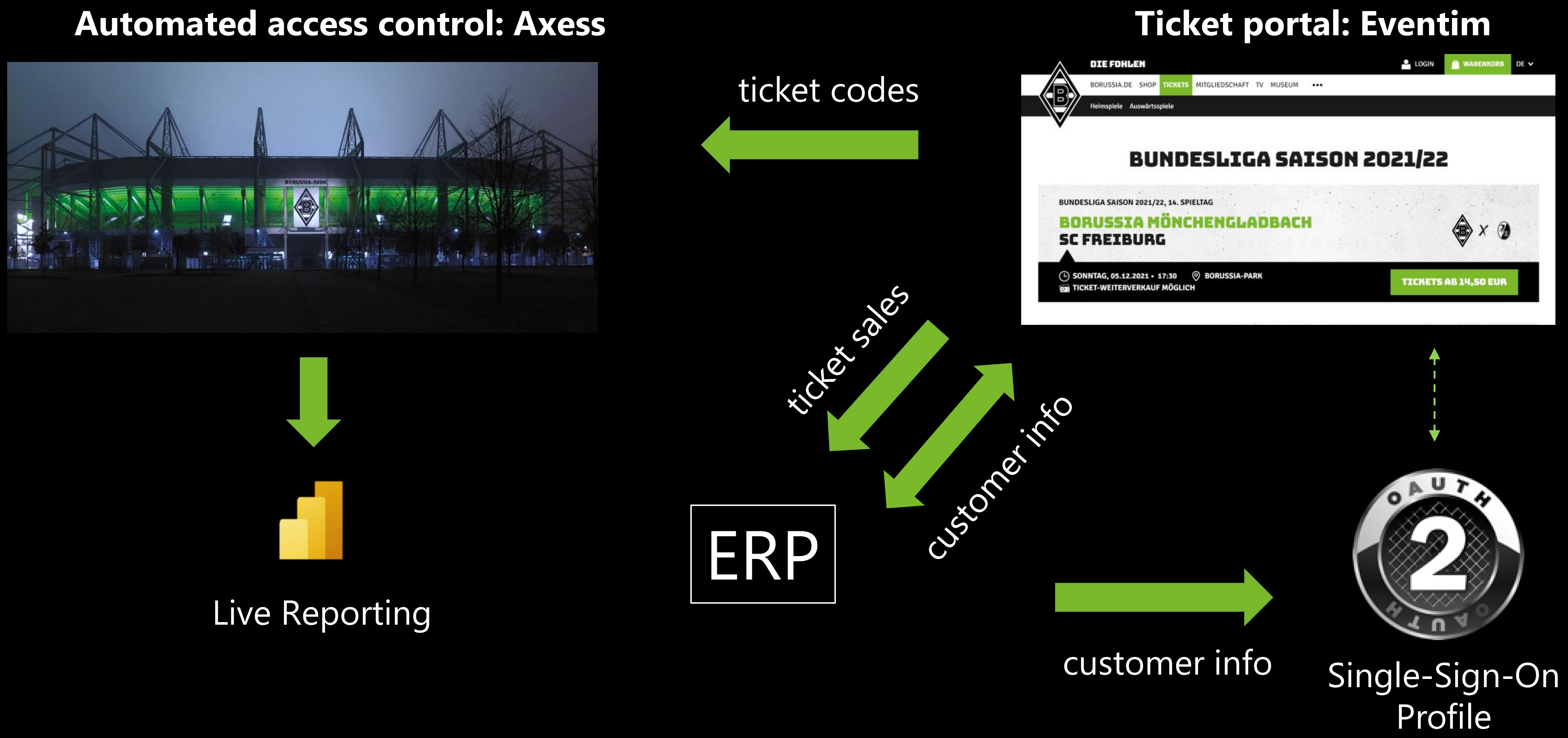
TICKETING INTEGRATION FLOW

BORUSSIA-PARK, MÖNCHENGLADBACH

DIE FOHLEN

TICKETING INTEGRATION FLOW

BORUSSIA / EVENTIM / AXESS



Automated access control: Axess

Ticket portal: Eventim

DIE FOHLEN

LOGIN WARENKORB DE

BORUSSIA.DE SHOP TICKETS MITGLIEDSCHAFT TV MUSEUM •••

Heimspiele Auswärtsspiele

BUNDESLIGA SAISON 2021/22

BUNDESLIGA SAISON 2021/22, 14. SPIELTAG

BORUSSIA MÖNCHENGLADBACH

SC FREIBURG

SONNTAG, 05.12.2021 • 17:30 BORUSSIA-PARK

TICKET-WEITERVERKAUF MÖGLICH

TICKETS AB 14,50 EUR

ticket codes

ticket sales

customer info

ERP

Live Reporting

customer info

Single-Sign-On Profile

Signals, Streams, and Jobs

Signal**:** The capture of an **occurrence** (statement of fact) during the operation of a software system

Event**:** A data record expressing a signal and its context**.** The context is expressed in metadata annotating the signal.

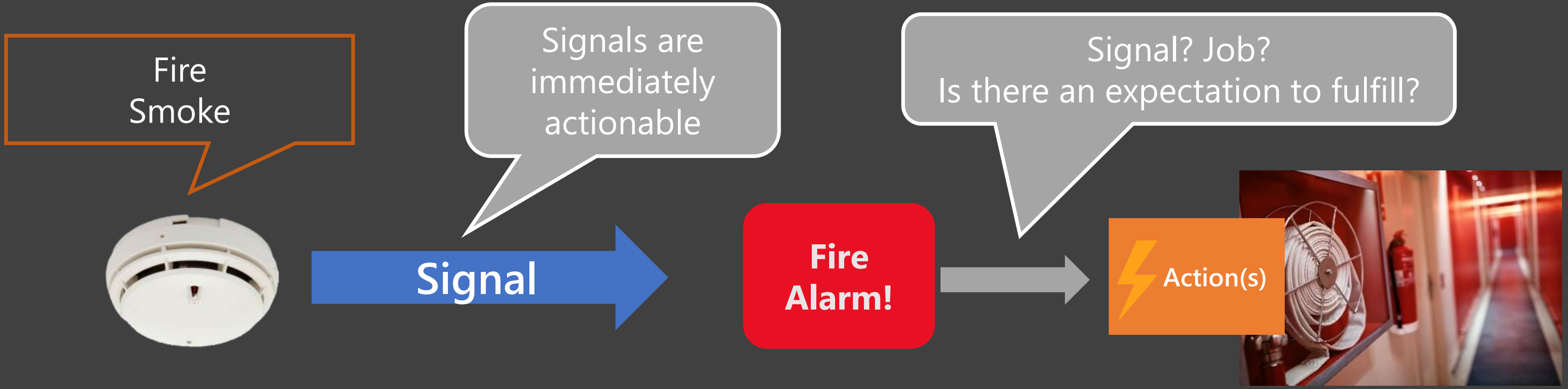
Event Stream**:** A chronological sequence of events belonging to the same context.

Job: *Not an event*. A description of a task that needs to be performed by some party. Preferably just once.

Signals

- Application deployment initiated
- Application deployment succeeded
- Application deployment failed
- Change committed
- Change proposed
- Change proposal verification succeeded
- Change proposal commented
- Database record inserted
- Database record updated
- File changed
- File created
- Venue seat ticket reserved
- Venue seat ticket released
- Venue seat ticket purchased
- CO PPM over threshold
- Room humidity read
- Room temperature read
- Smoke detected
- Object detected ahead
- Object velocity differential calculated
- Infrared movement detected
- Visual object classified
- Calendar entry created
- Calendar appointment reminder
- Vehicle passed induction loop
- Virtual machine started
- CPU utilization read
- Meeting scheduled
- Meeting attendee invited
- Meeting attendee accepted
- Mouse cursor moved
- Advertisement clicked
- Eye focus tracked
- Engine oil pressure read
- Engine coolant temperature read
- Engine RPM read
- Engine oil pressure threshold alert
- Tank pressure read
- Inventory item added to shelf
- Inventory item removed from shelf
- Inventory item scanned for sale
- Inventory item added
- Sales lead added
- Delivery address changed
- Package scanned
- Package delivered
- Package picked up
- Package loaded
- Package routed
- Pizza ordered
- Pizza prepared
- Pizza delivered
- Stock trade order issued
- Stock trade order executed
- Stock trade order settled
- Payment collected
- Aircraft rotated
- Aircraft position report
- Aircraft squawk code change
- Aircraft landed
- Account transfer scheduled
- Account transfer completed

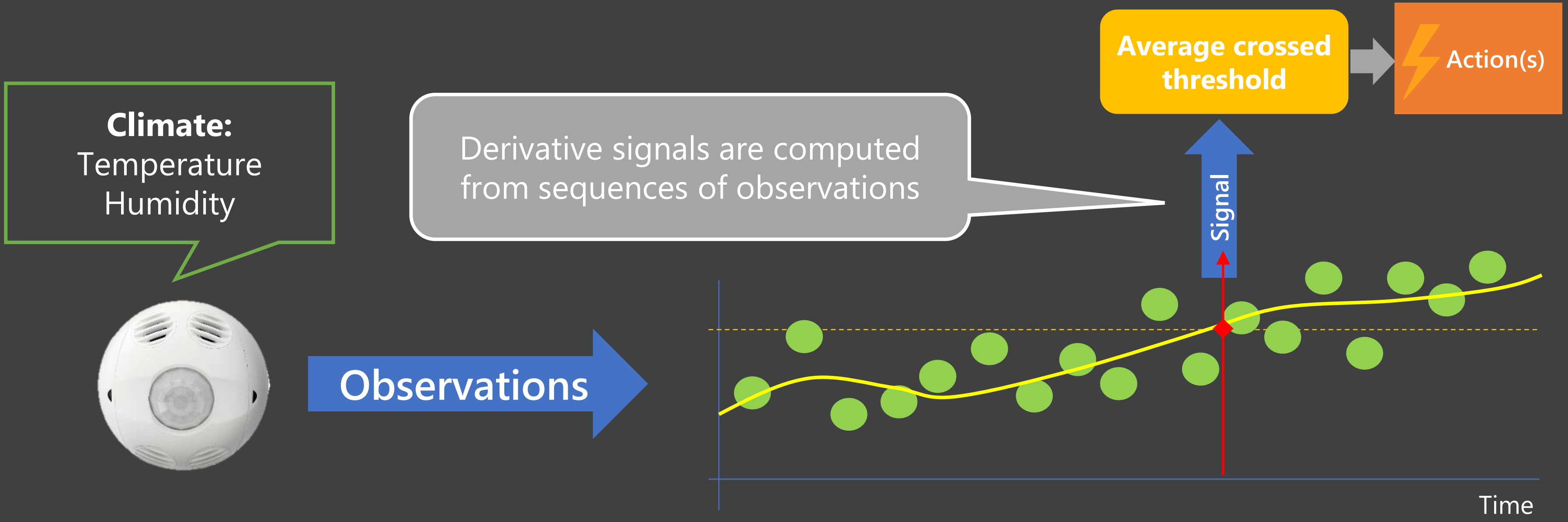
Events: Observations, Signals, Jobs



Fire
Smoke

Signals are immediately actionable

Signal? Job?
Is there an expectation to fulfill?

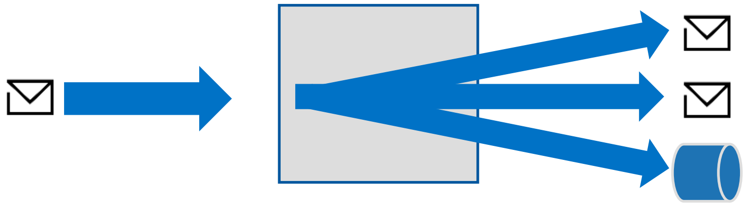


Climate:
Temperature
Humidity

Derivative signals are computed from sequences of observations

Discrete Event Router

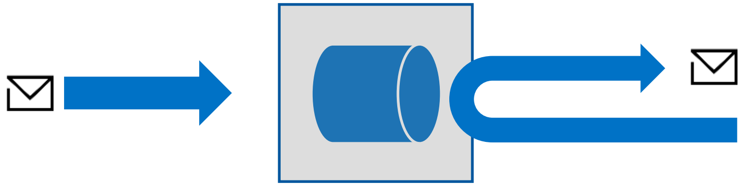
Azure Event Grid, AWS Event Bridge, Knative Eventing



Push-style distribution of discrete events to serverless workloads or other messaging infrastructures

Queue Pub/Sub Broker

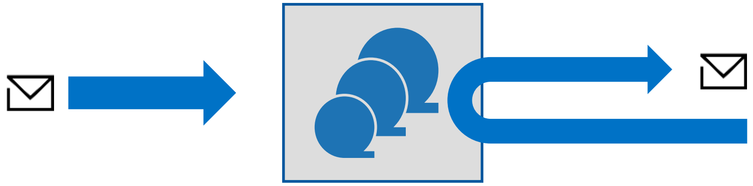
Azure Service Bus, AWS SQS/SNS, Google PubSub, Apache ActiveMQ, RabbitMQ, IBM MQ



Pull-style, queue-based transfer of jobs and control via message queues and topics

Event Stream Engine

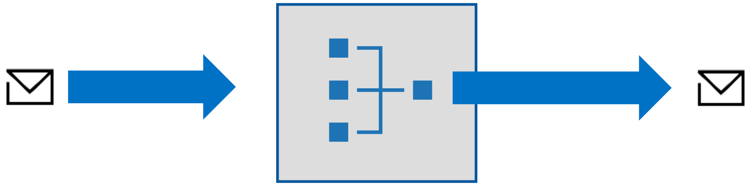
Azure Event Hubs, AWS Kinesis, Apache Kafka, Apache Pulsar, CNCF Pravega



Partitioned, high-volume, tape-drive-style sequential recording and unlimited, pull-style re-reads of event streams.

Event Stream Aggregator

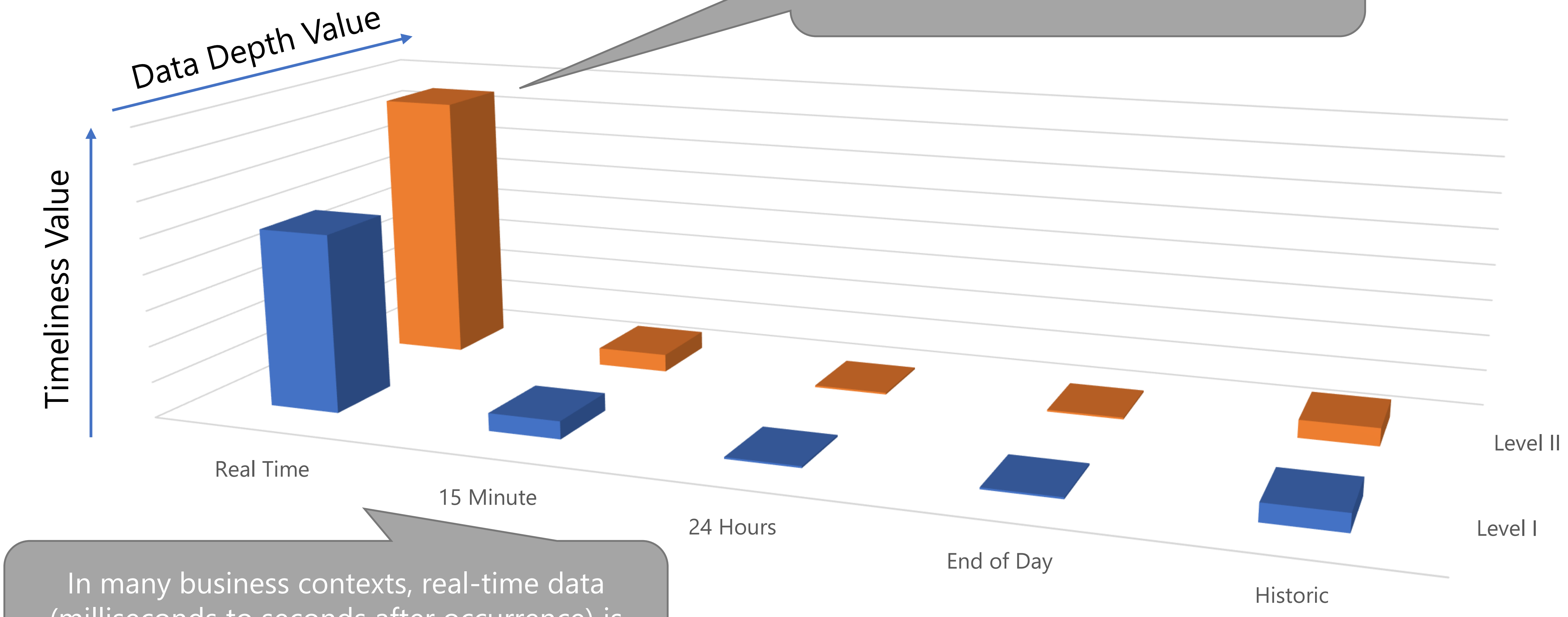
Azure Stream Analytics, AWS Kinesis Analytics, Apache Samza, Apache Flink, etc.



Stateful processing of event streams yielding event streams and discrete events as continuous output

Event Streams and Time(-liness)

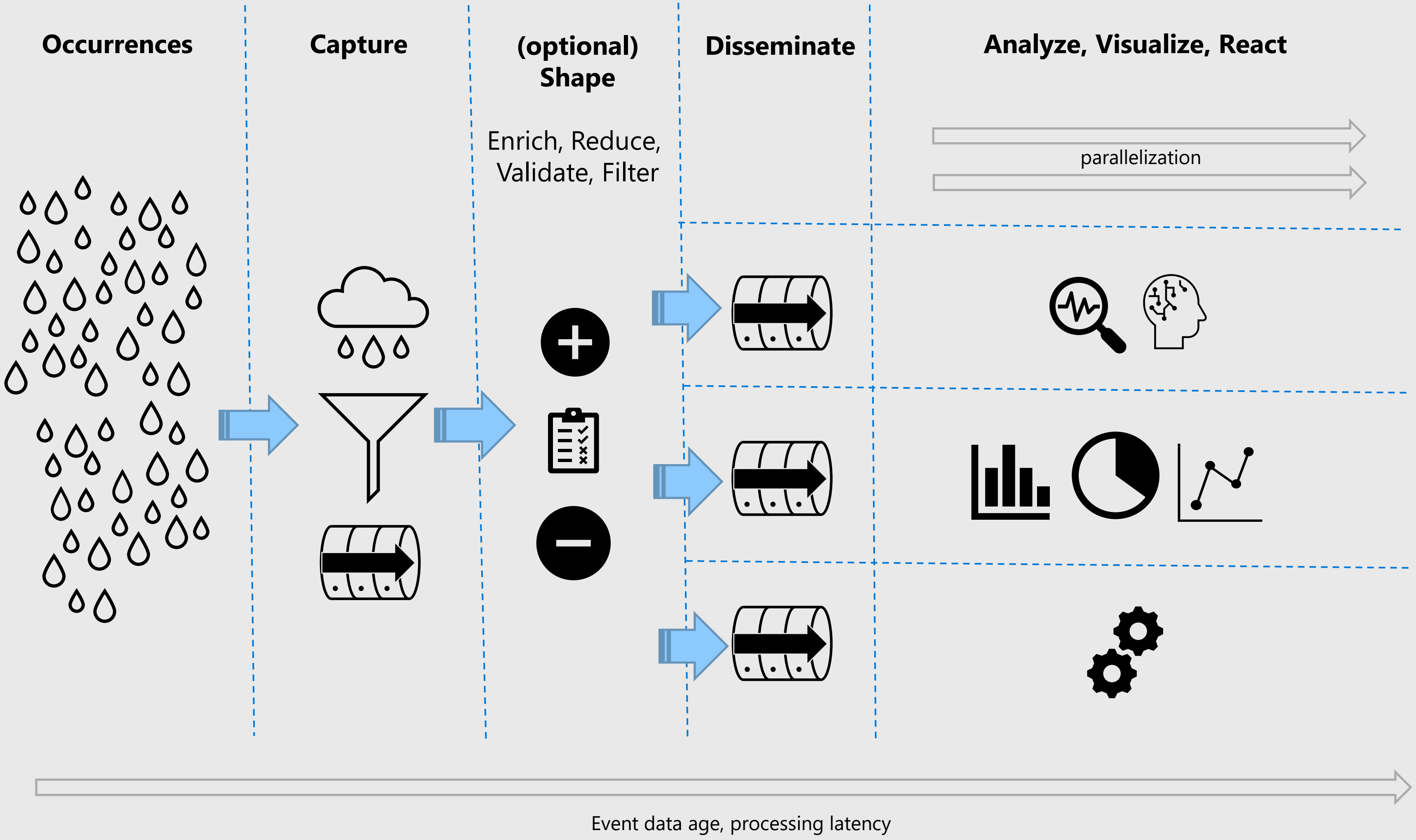
Event Data Value – Securities Markets



Data publishers provide more value by further enriching real-time data.

In many business contexts, real-time data (milliseconds to seconds after occurrence) is measurably most valuable when fresh.

Velocity Matters ➔ Parallelization Matters



Occurrences

Capture

(optional) Shape

Enrich, Reduce, Validate, Filter

Disseminate

Analyze, Visualize, React

Event data age, processing latency

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)

Event Hubs

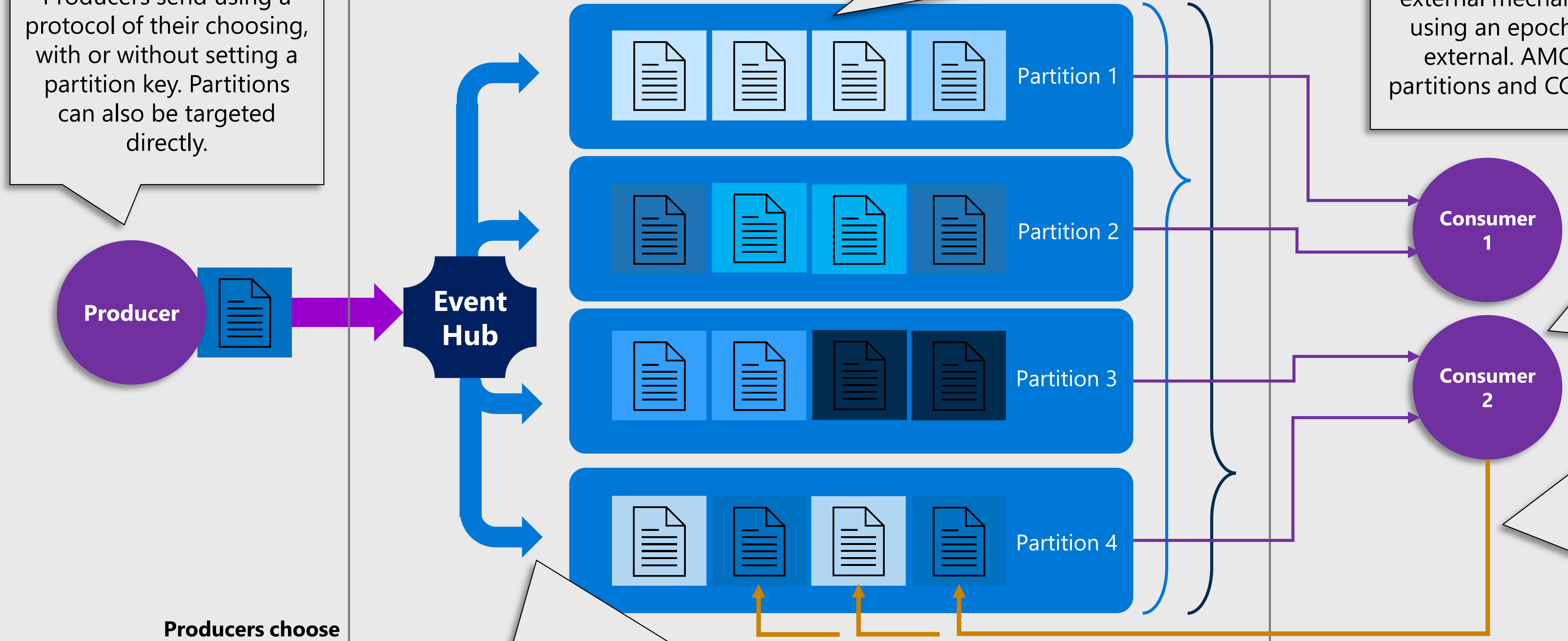
Producers send using a protocol of their choosing, with or without setting a partition key. Partitions can also be targeted directly.

A log partition is a durable, triple-replicated, append-only storage structure. Each partition has a throughput ceiling imposed by IO and compute limits. More partitions provide more throughput capacity, but also make consumption more complex.

A consumer group is an optional mechanism to allow multiple parallel consumers to coordinate who owns which partition.

Kafka RPC: Broker assigns partitions and checkpoints can be stored in the CG.

AMQP: Clients choose leader using an external mechanism and can break leases using an epoch model. Checkpoints are external. AMQP with server-assigned partitions and CG checkpoints on roadmap.



Consumers on a consumer group own one or more partitions exclusively to ensure in-order processing.

Consumers choose the initial log offset:

Start (oldest), End (newest), Timestamp, Sequence Number

Read progress is noted in checkpoints that can be stored anywhere.

Producers choose their preferred protocol

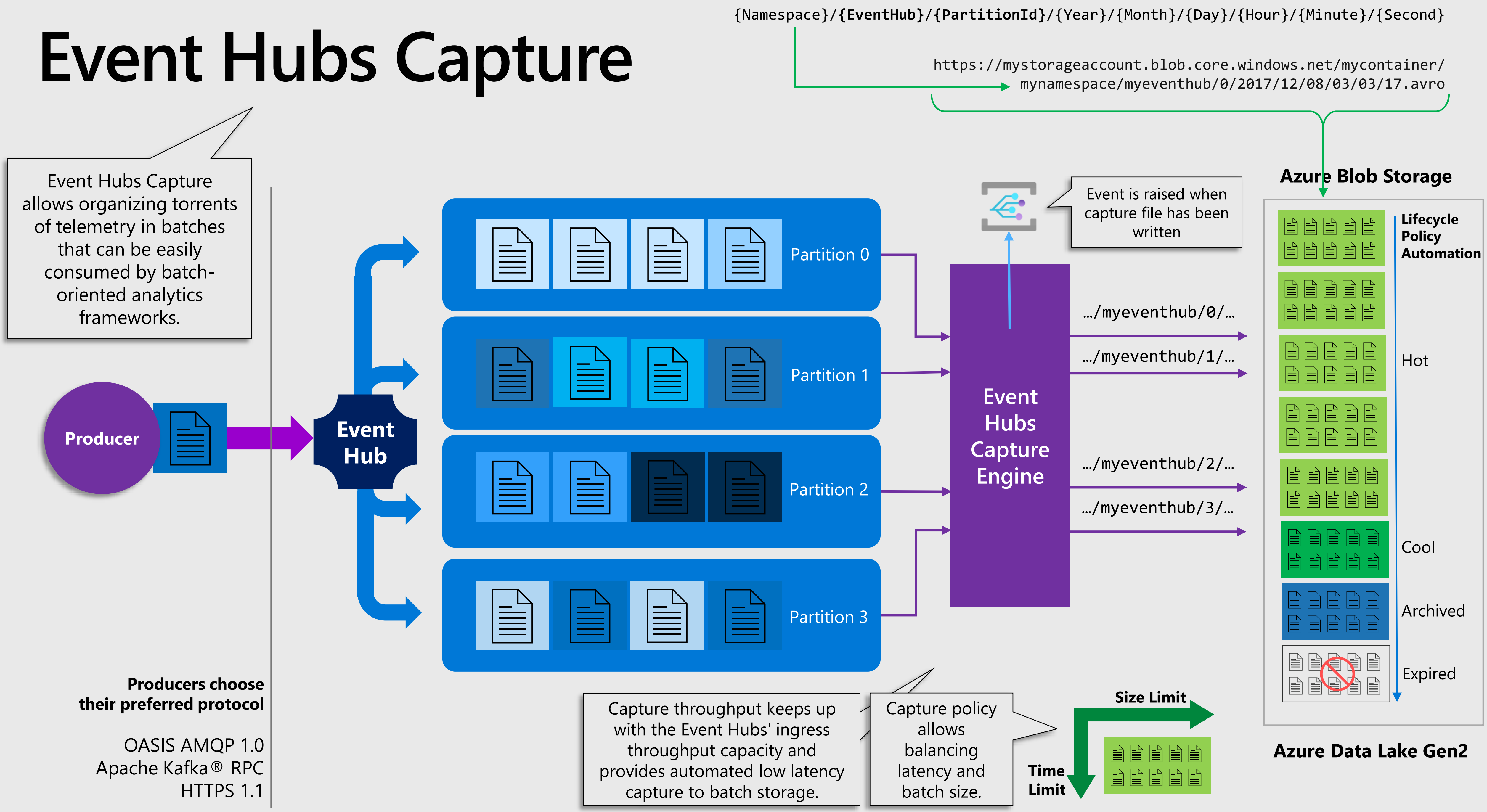
OASIS AMQP 1.0
Apache Kafka® RPC
HTTPS 1.1

An Event Hub is an event stream manager handling one or more independent log partitions. Different from Apache Kafka®, Event Hubs can optionally assign events to partitions on the server-side.

Consumers choose their own preferred protocol

OASIS AMQP 1.0
Apache Kafka® RPC

Event Hubs Capture



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

```
{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 is raised when capture file has been written

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.

Event Hubs* is not...

* Apache Kafka isn't any of these things, either.

... 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

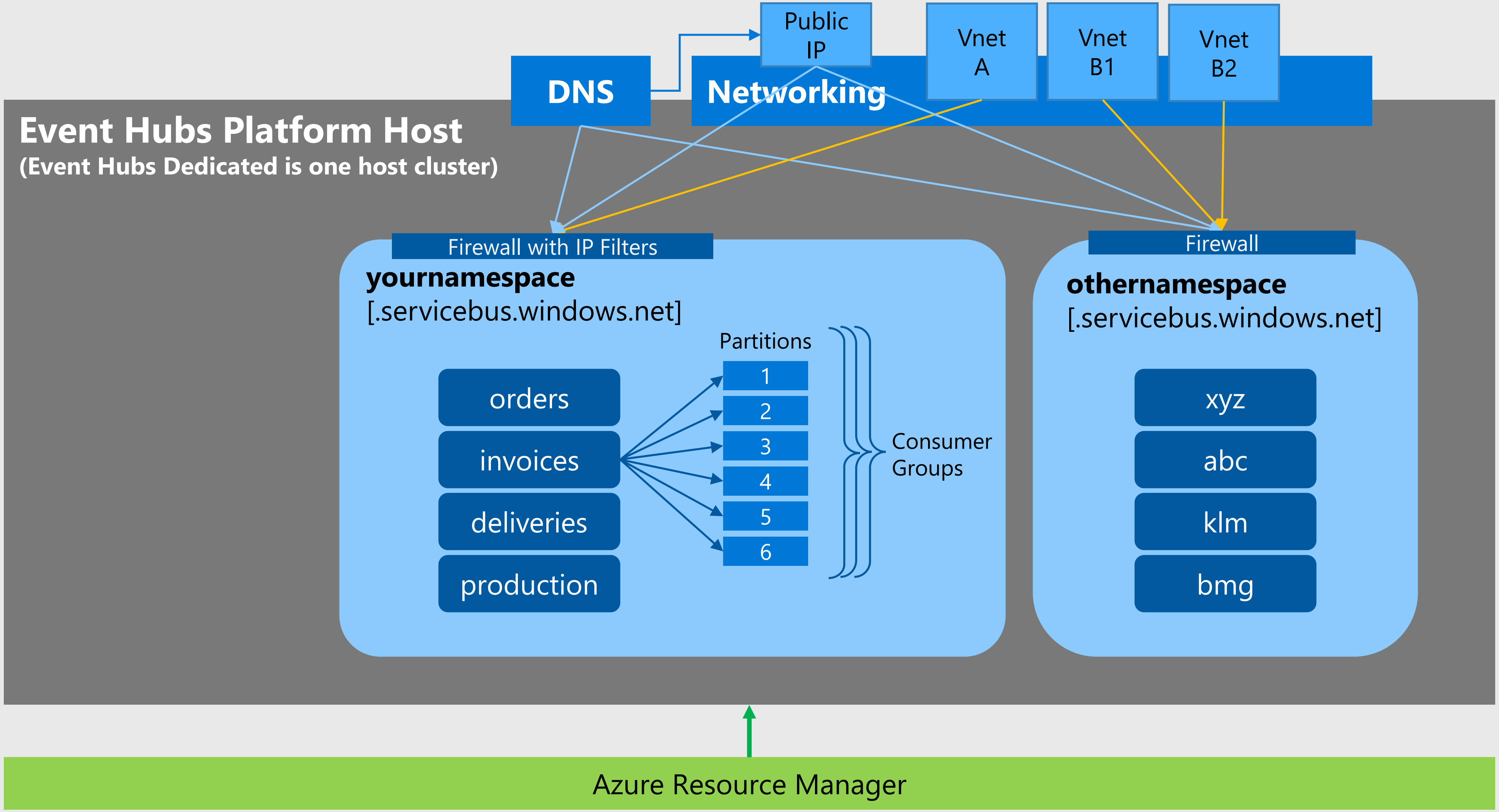
... a discrete event distribution engine. Event Hubs does not do push deliveries, and delivery failures need to be tracked individually.

→ 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.

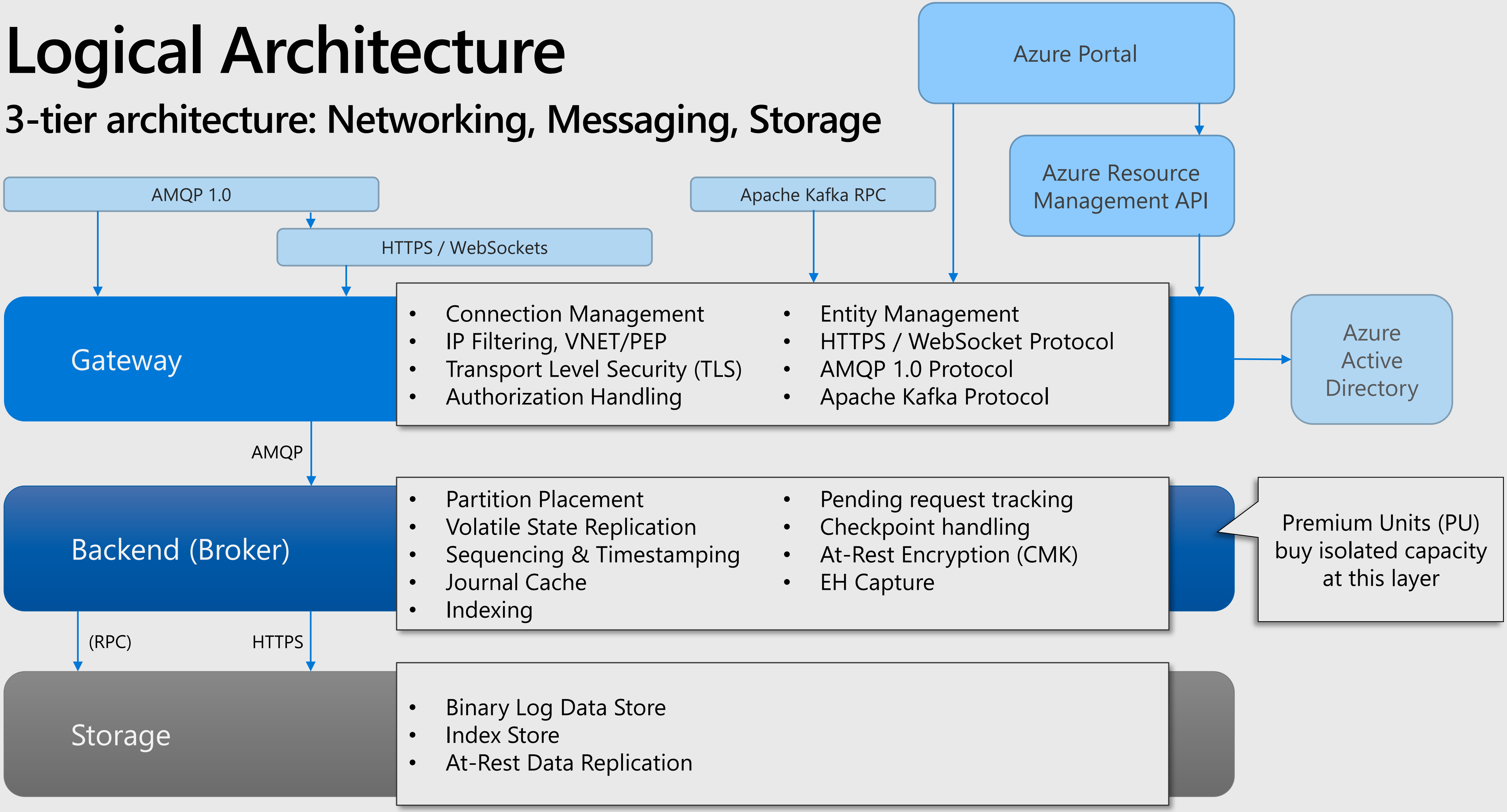
→ Azure Cosmos DB, Azure SQL, Azure Synapse, Azure Data Explorer

Event Hubs Namespaces: Virtual Brokers



Logical Architecture

3-tier architecture: Networking, Messaging, Storage



Gateway

- Connection Management
- IP Filtering, VNET/PEP
- Transport Level Security (TLS)
- Authorization Handling
- Entity Management
- HTTPS / WebSocket Protocol
- AMQP 1.0 Protocol
- Apache Kafka Protocol

Backend (Broker)

- Partition Placement
- Volatile State Replication
- Sequencing & Timestamping
- Journal Cache
- Indexing
- Pending request tracking
- Checkpoint handling
- At-Rest Encryption (CMK)
- EH Capture

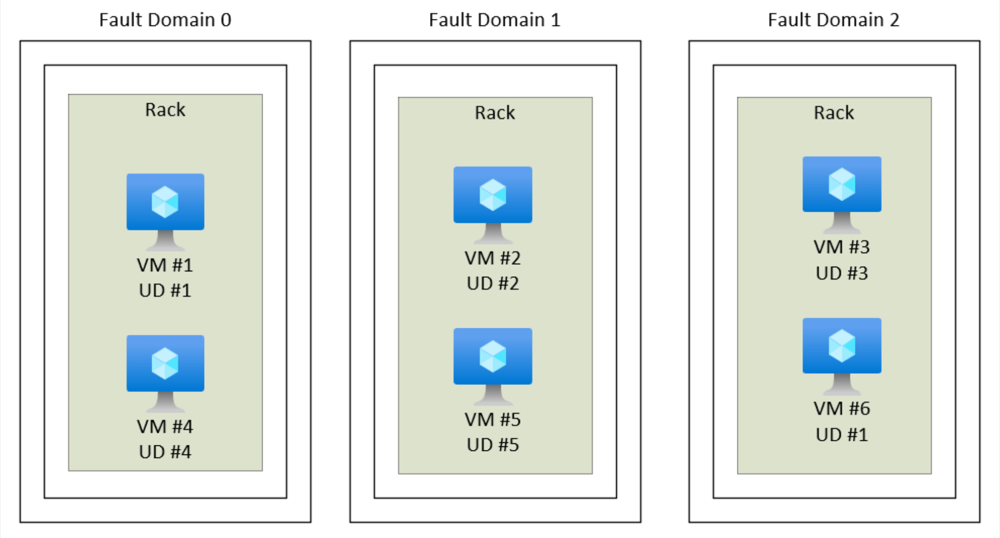
Premium Units (PU) buy isolated capacity at this layer

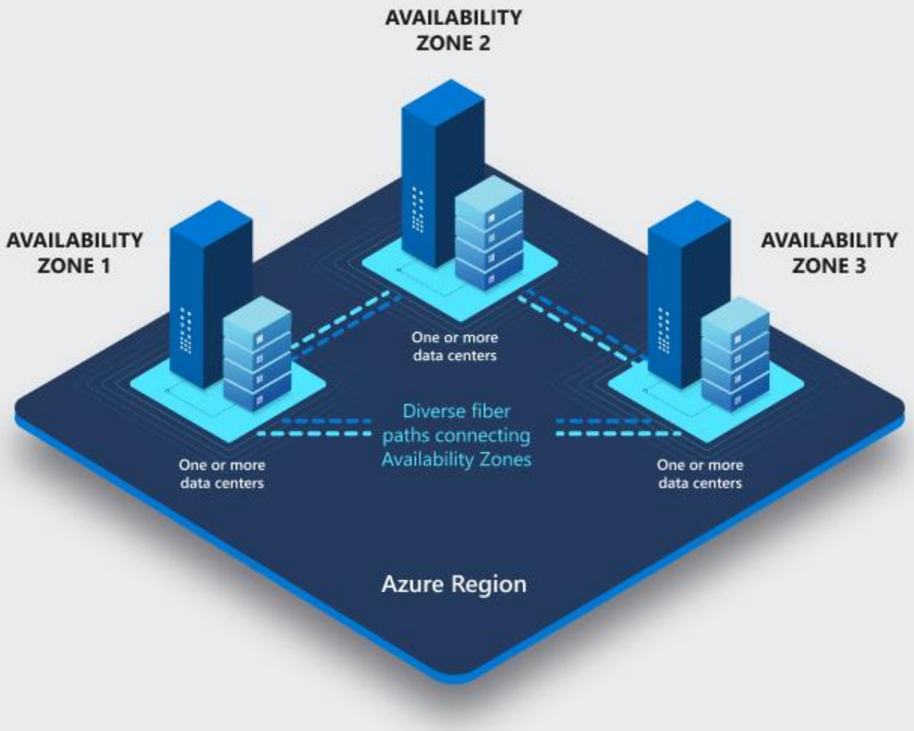
Storage

- Binary Log Data Store
- Index Store
- At-Rest Data Replication

Fault Domains & Availability Zones

High-Availability starts at the physical reality



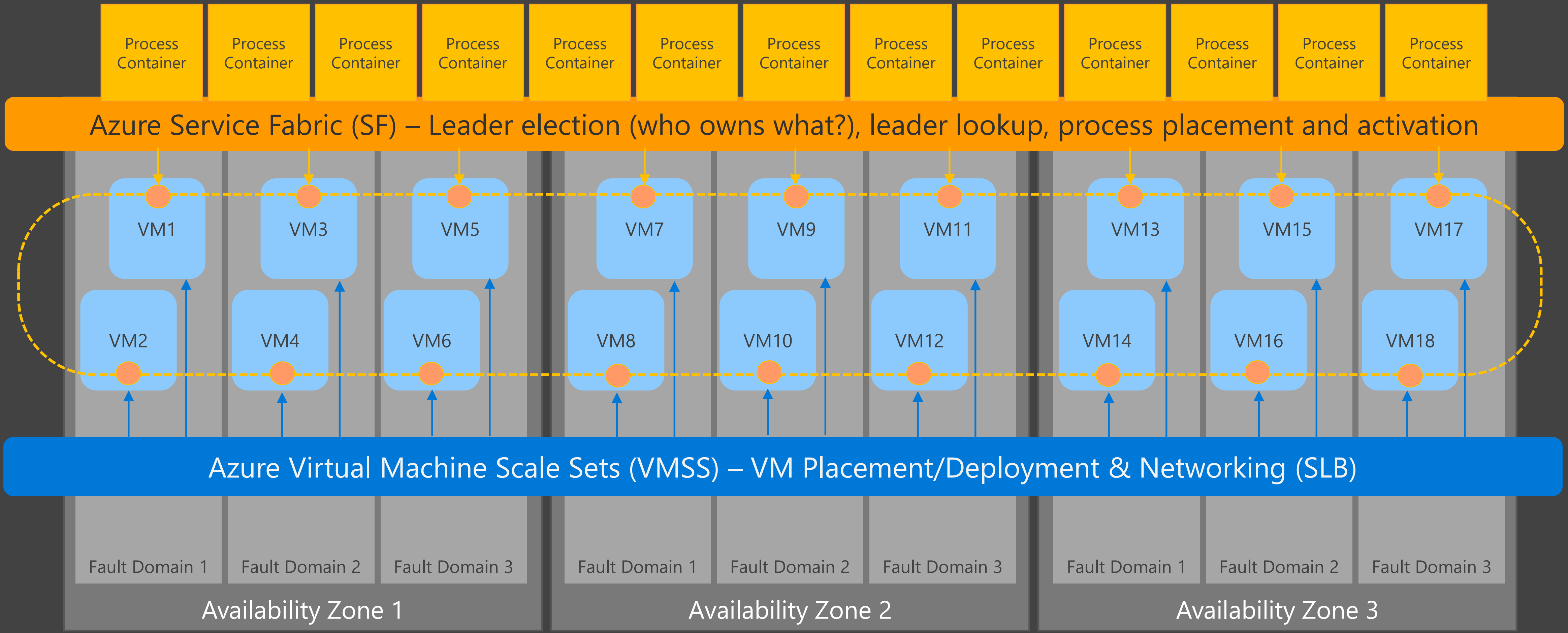


Fault Domain Placement:
Cluster VMs are spread across at least 3 fault domains such that the loss of a rack or network poses no availability risk. Recovery from a fault domain failure is fully automated and the system maintains SLA.

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



Process Container | Process Container | Process Container | Process Container | Process Container | Process Container | Process Container | Process Container | Process Container | Process Container | Process Container | Process Container | Process Container

Azure Service Fabric (SF) – Leader election (who owns what?), leader lookup, process placement and activation

VM1 VM2 VM3 VM4 VM5 VM6 VM7 VM8 VM9 VM10 VM11 VM12 VM13 VM14 VM15 VM16 VM17 VM18

Azure Virtual Machine Scale Sets (VMSS) – VM Placement/Deployment & Networking (SLB)

Fault Domain 1 | Fault Domain 2 | Fault Domain 3 — Availability Zone 1

Fault Domain 1 | Fault Domain 2 | Fault Domain 3 — Availability Zone 2

Fault Domain 1 | Fault Domain 2 | Fault Domain 3 — Availability Zone 3

Backend and Gateway Clusters

Principles

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

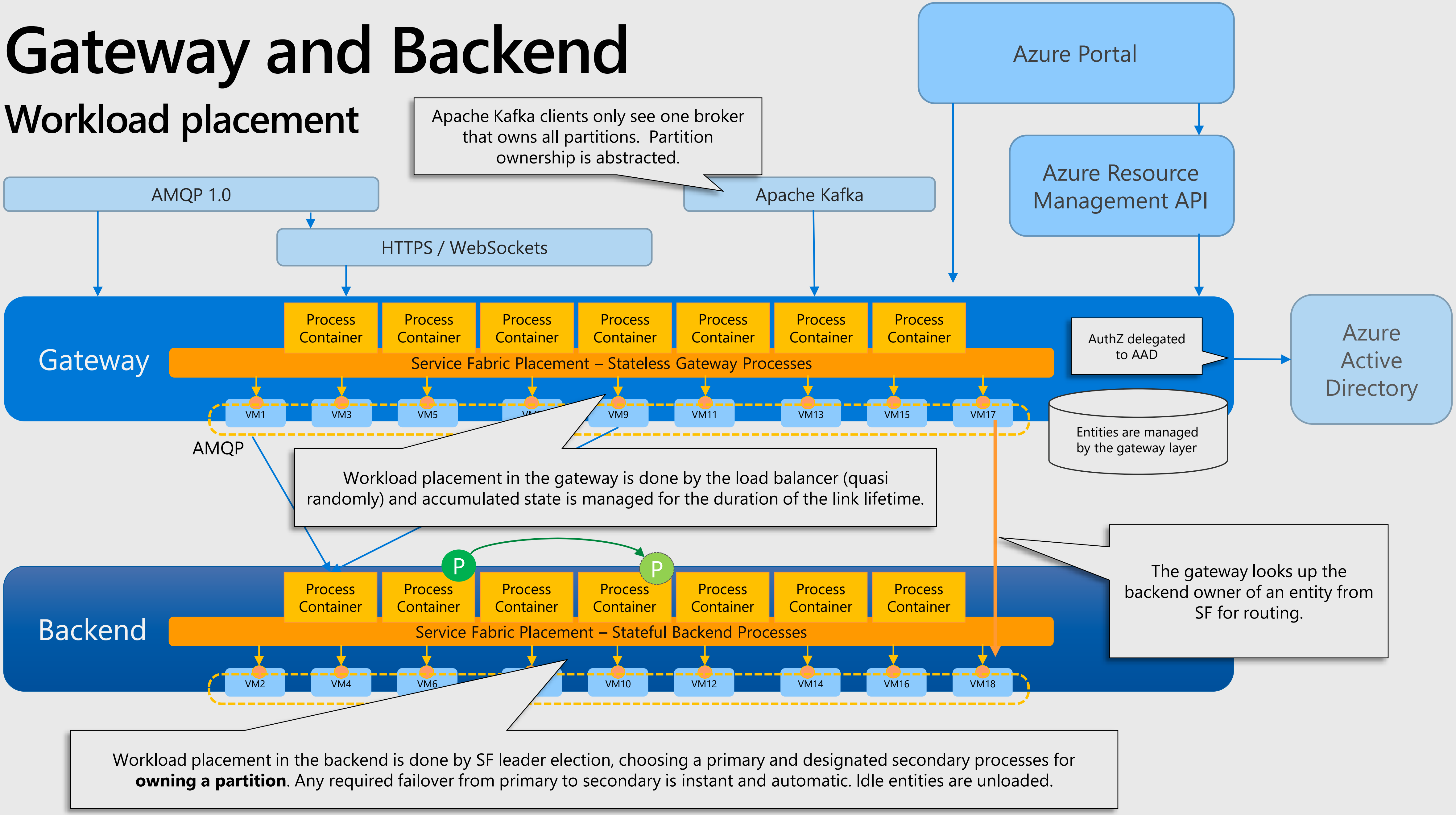
Each cluster has a well-known number of virtual machines. We don't use auto-scaling with VMs. We may (rarely) choose to initiate scale up/down to a different level.

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



Azure Portal

Azure Resource Management API

Azure Active Directory

Apache Kafka clients only see one broker that owns all partitions. Partition ownership is abstracted.

AMQP 1.0

HTTPS / WebSockets

Apache Kafka

Gateway

Process Container

Process Container

Process Container

Process Container

Process Container

Process Container

Process Container

Service Fabric Placement – Stateless Gateway Processes

AuthZ delegated to AAD

Entities are managed by the gateway layer

VM1 VM3 VM5 VM9 VM11 VM13 VM15 VM17

AMQP

Workload placement in the gateway is done by the load balancer (quasi randomly) and accumulated state is managed for the duration of the link lifetime.

The gateway looks up the backend owner of an entity from SF for routing.

P

P

Backend

Process Container

Process Container

Process Container

Process Container

Process Container

Process Container

Process Container

Service Fabric Placement – Stateful Backend Processes

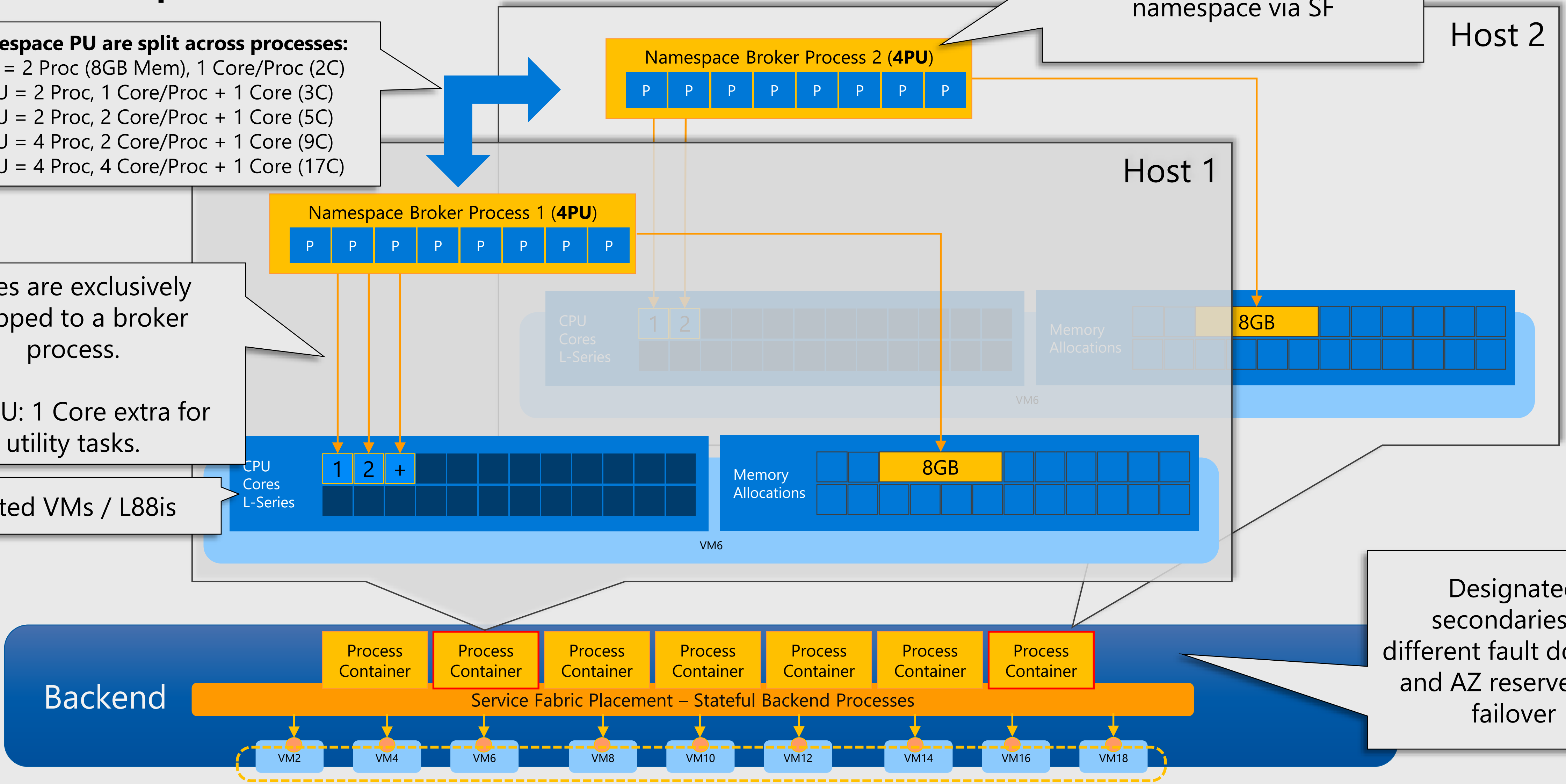
VM2 VM4 VM6 VM10 VM12 VM14 VM16 VM18

Workload placement in the backend is done by SF leader election, choosing a primary and designated secondary processes for **owning a partition**. Any required failover from primary to secondary is instant and automatic. Idle entities are unloaded.

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)



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

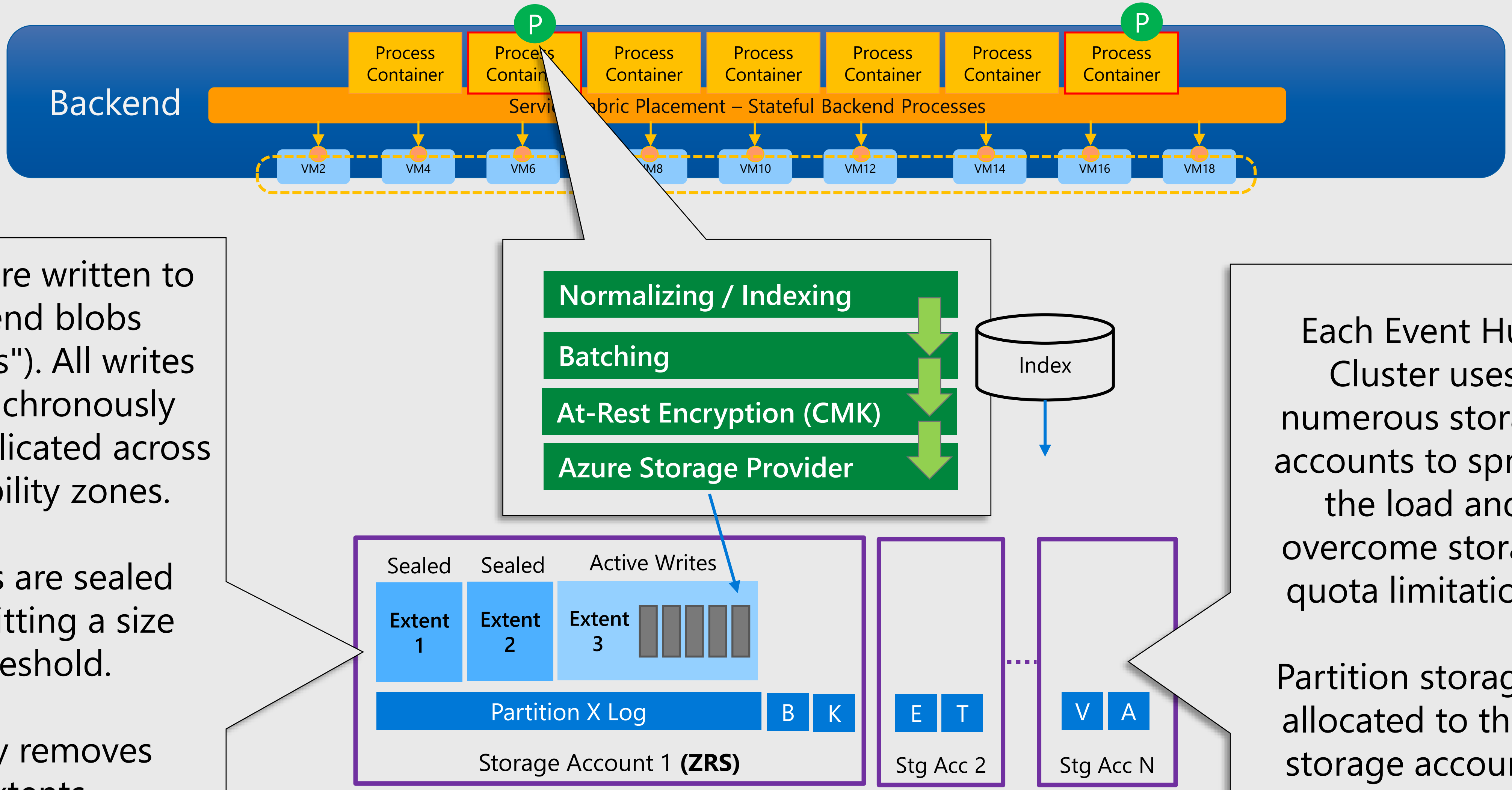
Cores are exclusively mapped to a broker process.

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

Isolated VMs / L88is

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.

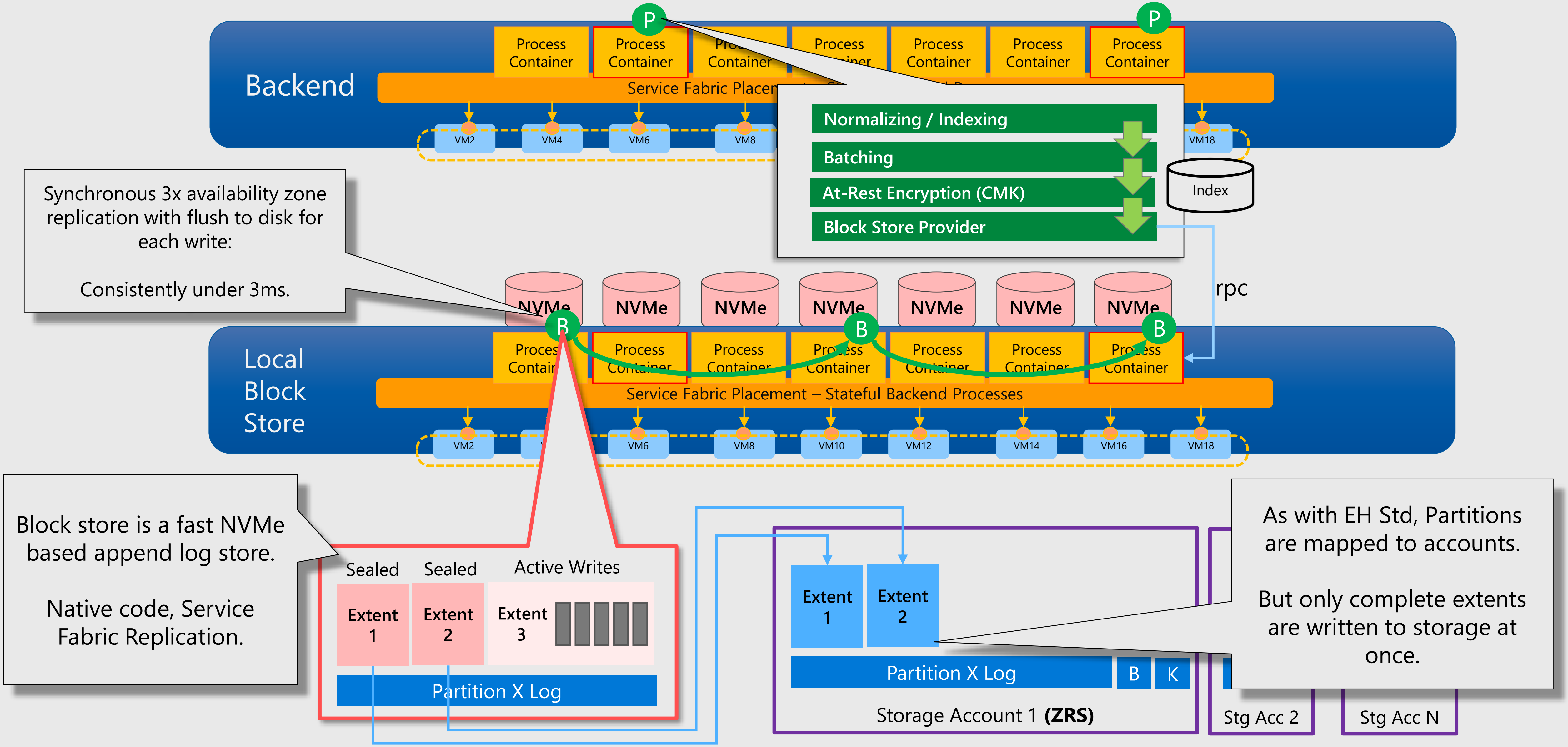
Extents are sealed after hitting a size threshold.

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



Synchronous 3x availability zone replication with flush to disk for each write:

Consistently under 3ms.

Block store is a fast NVMe based append log store.

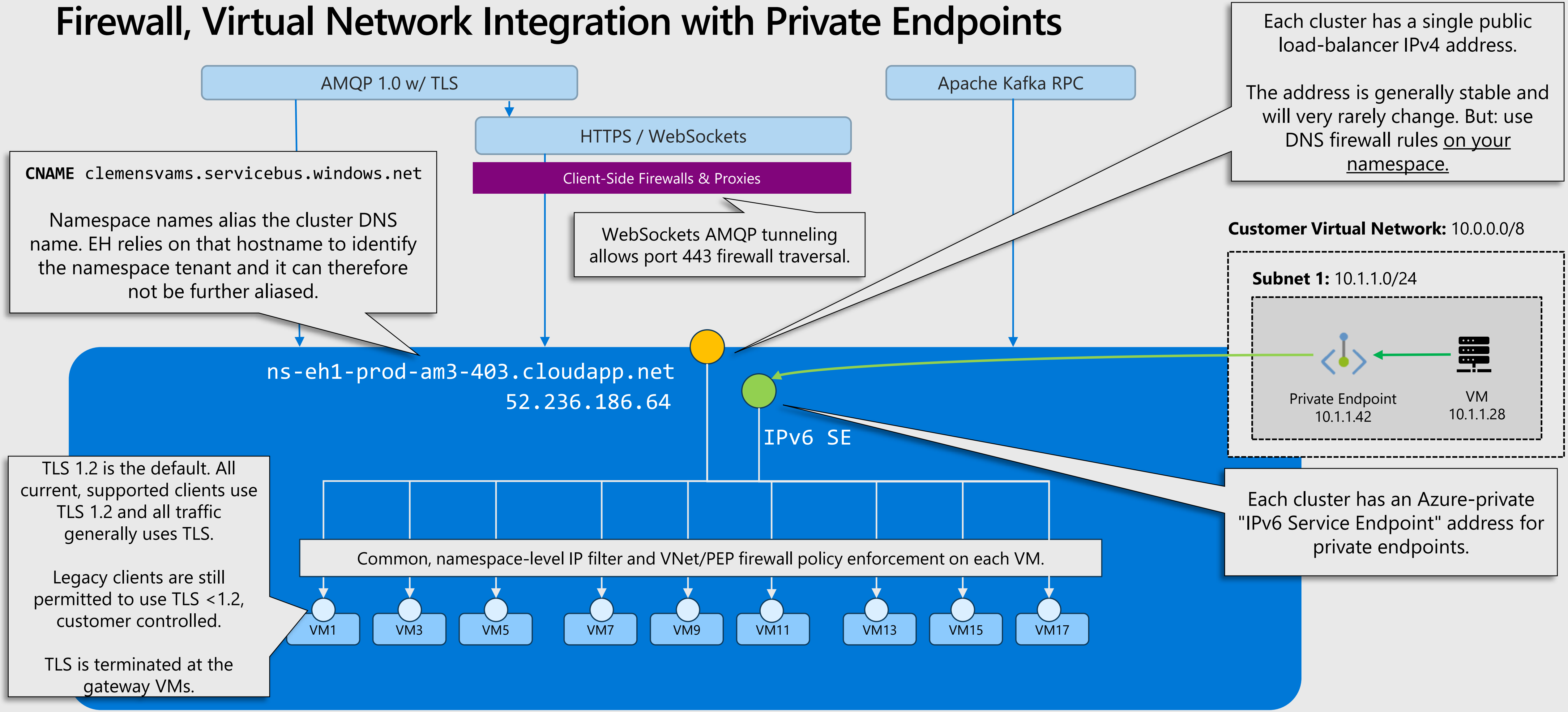
Native code, Service Fabric Replication.

As with EH Std, Partitions are mapped to accounts.

But only complete extents are written to storage at once.

Networking Features

Firewall, Virtual Network Integration with Private Endpoints


AMQP 1.0 w/ TLS
HTTPS / WebSockets
Client-Side Firewalls & Proxies
Apache Kafka RPC
ns-eh1-prod-am3-403.cloudapp.net
52.236.186.64
IPv6 SE
Common, namespace-level IP filter and VNet/PEP firewall policy enforcement on each VM.
VM1
VM3
VM5
VM7
VM9
VM11
VM13
VM15
VM17
Customer Virtual Network: 10.0.0.0/8
Subnet 1: 10.1.1.0/24
Private Endpoint
10.1.1.42
VM
10.1.1.28

CNAME `clemensvams.servicebus.windows.net`

Namespace names alias the cluster DNS name. EH relies on that hostname to identify the namespace tenant and it can therefore not be further aliased.

WebSockets AMQP tunneling allows port 443 firewall traversal.

Each cluster has a single public load-balancer IPv4 address.

The address is generally stable and will very rarely change. But: use DNS firewall rules <u>on your namespace.</u>

Each cluster has an Azure-private "IPv6 Service Endpoint" address for private endpoints.

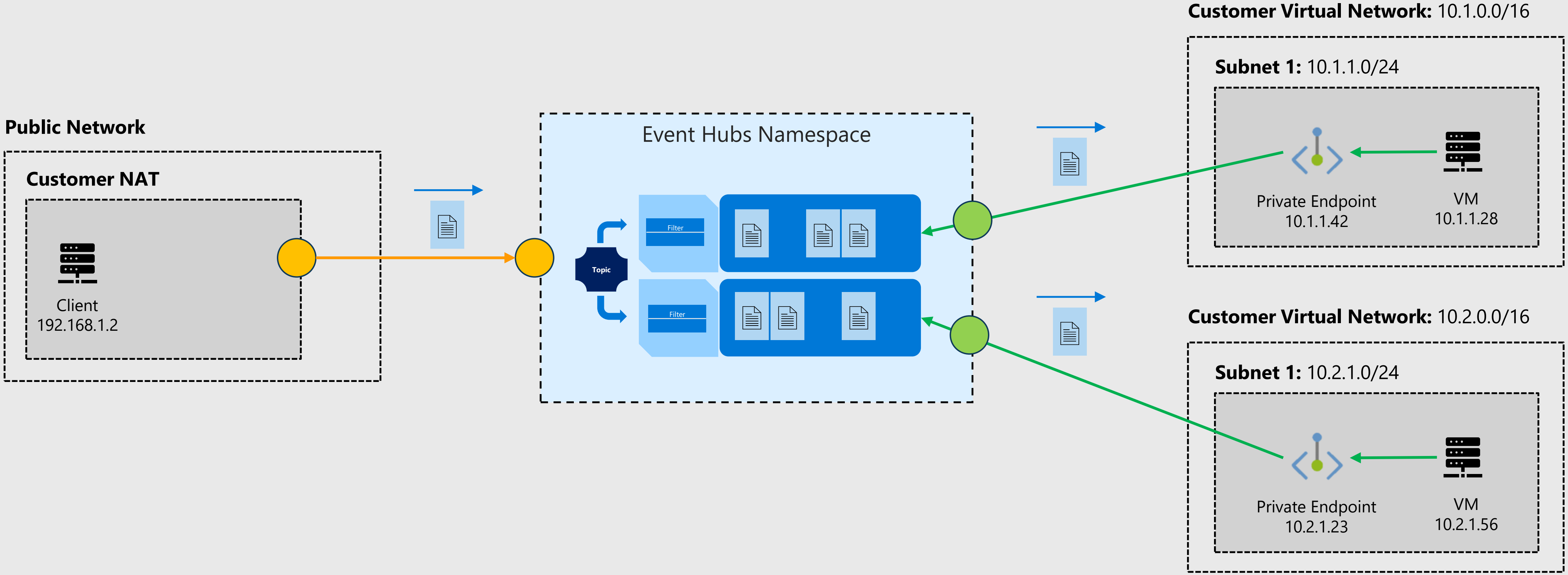
TLS 1.2 is the default. All current, supported clients use TLS 1.2 and all traffic generally uses TLS.

Legacy clients are still permitted to use TLS <1.2, customer controlled.

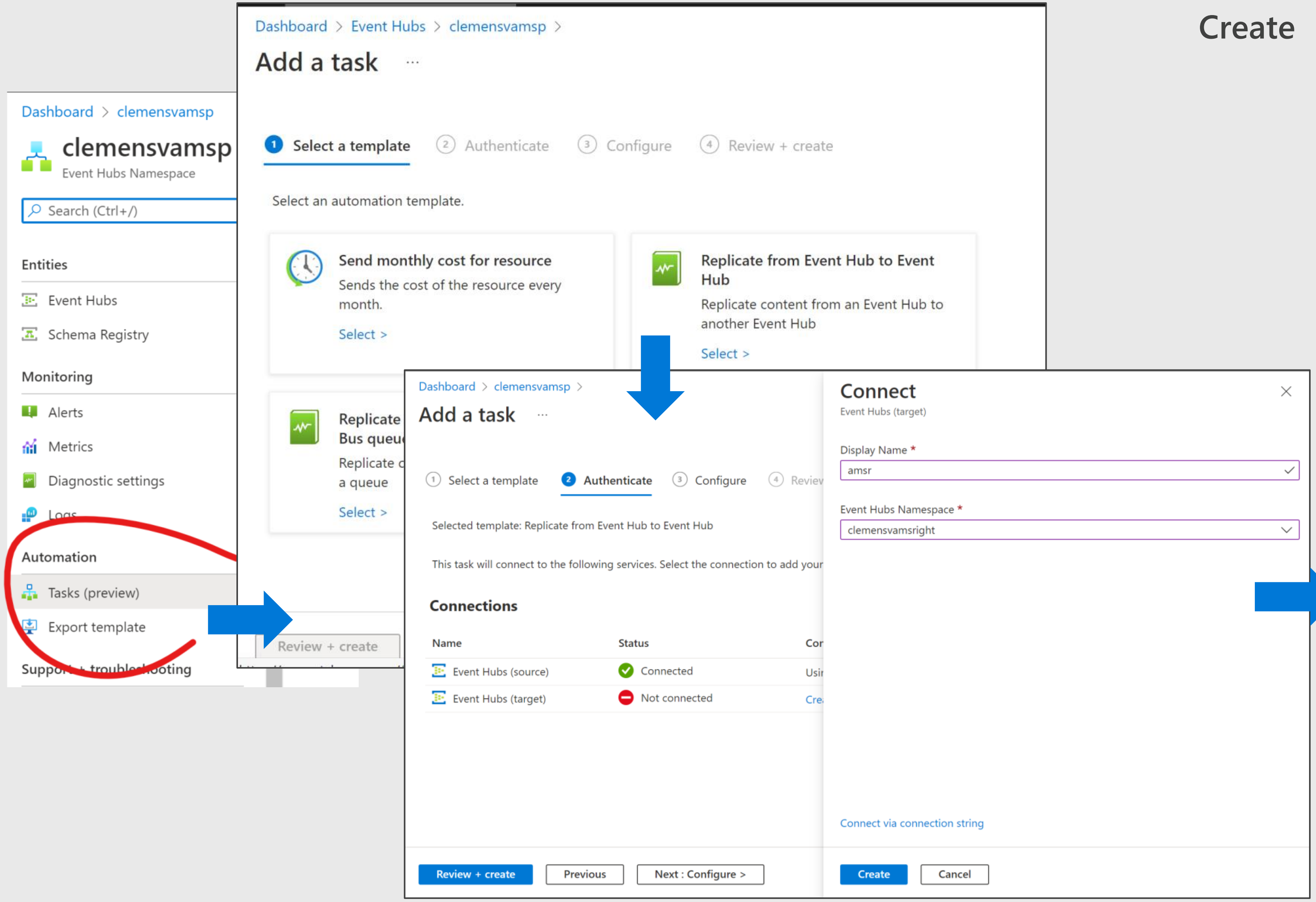
TLS is terminated at the gateway VMs.

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.



Flows: Automation Tasks



Dashboard > clemensvamsp

clemensvamsp

Event Hubs Namespace

Search (Ctrl+/)

Entities

- Event Hubs
- Schema Registry

Monitoring

- Alerts
- Metrics
- Diagnostic settings
- Logs

Automation

- Tasks (preview)
- Export template

Support + troubleshooting

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 >

Replicate ... Bus queu...
Replicate ... a queue
Select >

Review + create

Dashboard > clemensvamsp >

Add a task ...

1 Select a template 2 Authenticate 3 Configure 4 Review

Selected template: Replicate from Event Hub to Event Hub

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

Connections

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

Review + create | Previous | Next : Configure >

Connect

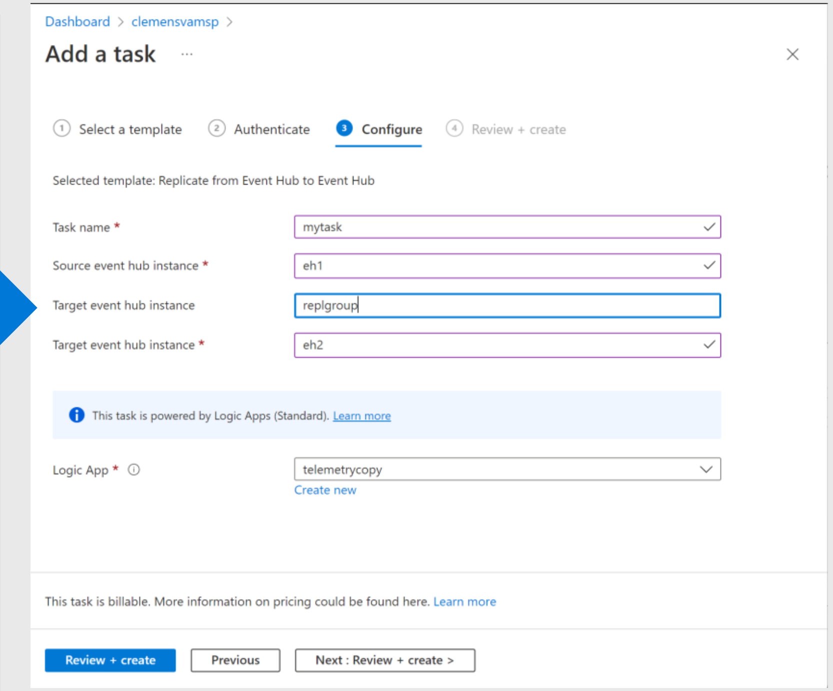
Event Hubs (target)

Display Name *: amsr

Event Hubs Namespace *: clemensvamsright

Connect via connection string

Create | Cancel



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

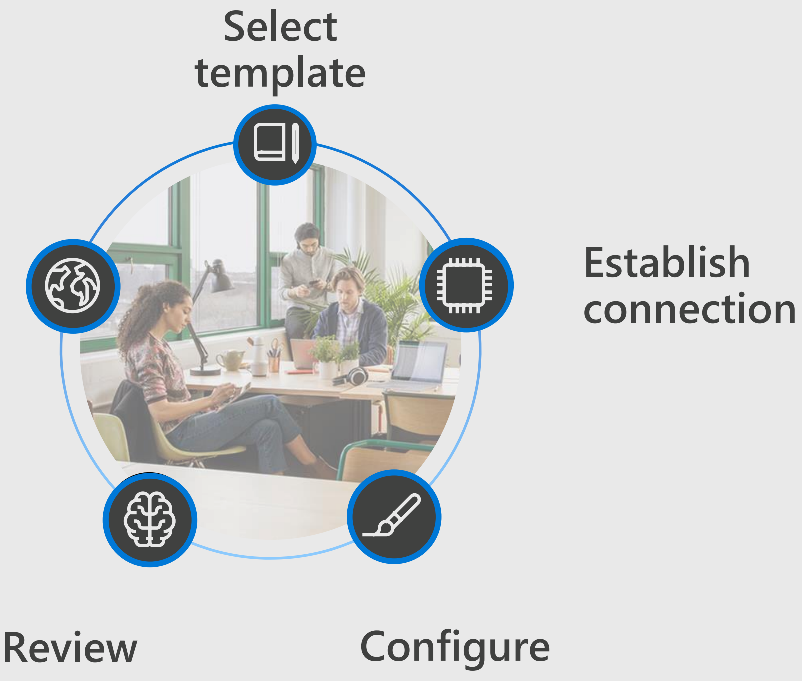
Task name *	mytask
Source event hub instance *	eh1
Target event hub instance	replgroup
Target event hub instance *	eh2

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

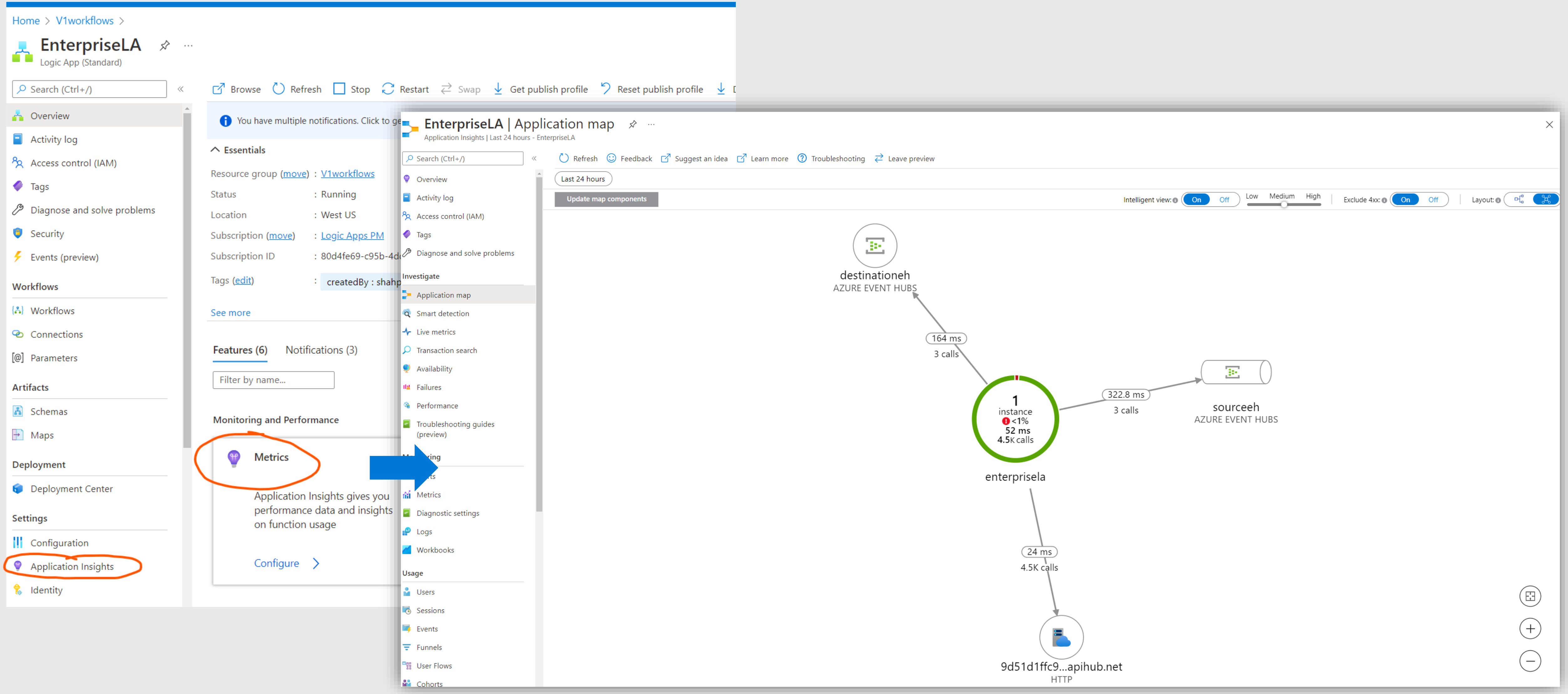
Logic App *: telemetrycopy
Create new

This task is billable. More information on pricing could be found here. Learn more

Review + create | Previous | Next : Review + create >



EventHub replication visualization



Home > V1workflows >

EnterpriseLA

Logic App (Standard)

Search (Ctrl+/)

Browse · Refresh · Stop · Restart · Swap · Get publish profile · Reset publish profile

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Security
Events (preview)

Workflows
Workflows
Connections
Parameters

Artifacts
Schemas
Maps

Deployment
Deployment Center

Settings
Configuration
Application Insights
Identity

You have multiple notifications. Click to ge

Essentials

Resource group (move)	: V1workflows
Status	: Running
Location	: West US
Subscription (move)	: Logic Apps PM
Subscription ID	: 80d4fe69-c95b-4d
Tags (edit)	: createdBy : shahp

See more

Features (6) · Notifications (3)

Filter by name...

Monitoring and Performance

Metrics

Application Insights gives you performance data and insights on function usage

Configure >

EnterpriseLA | Application map

Application Insights | Last 24 hours - EnterpriseLA

Search (Ctrl+/)

Refresh · Feedback · Suggest an idea · Learn more · Troubleshooting · Leave preview

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

Investigate
Application map
Smart detection
Live metrics
Transaction search
Availability
Failures
Performance
Troubleshooting guides (preview)

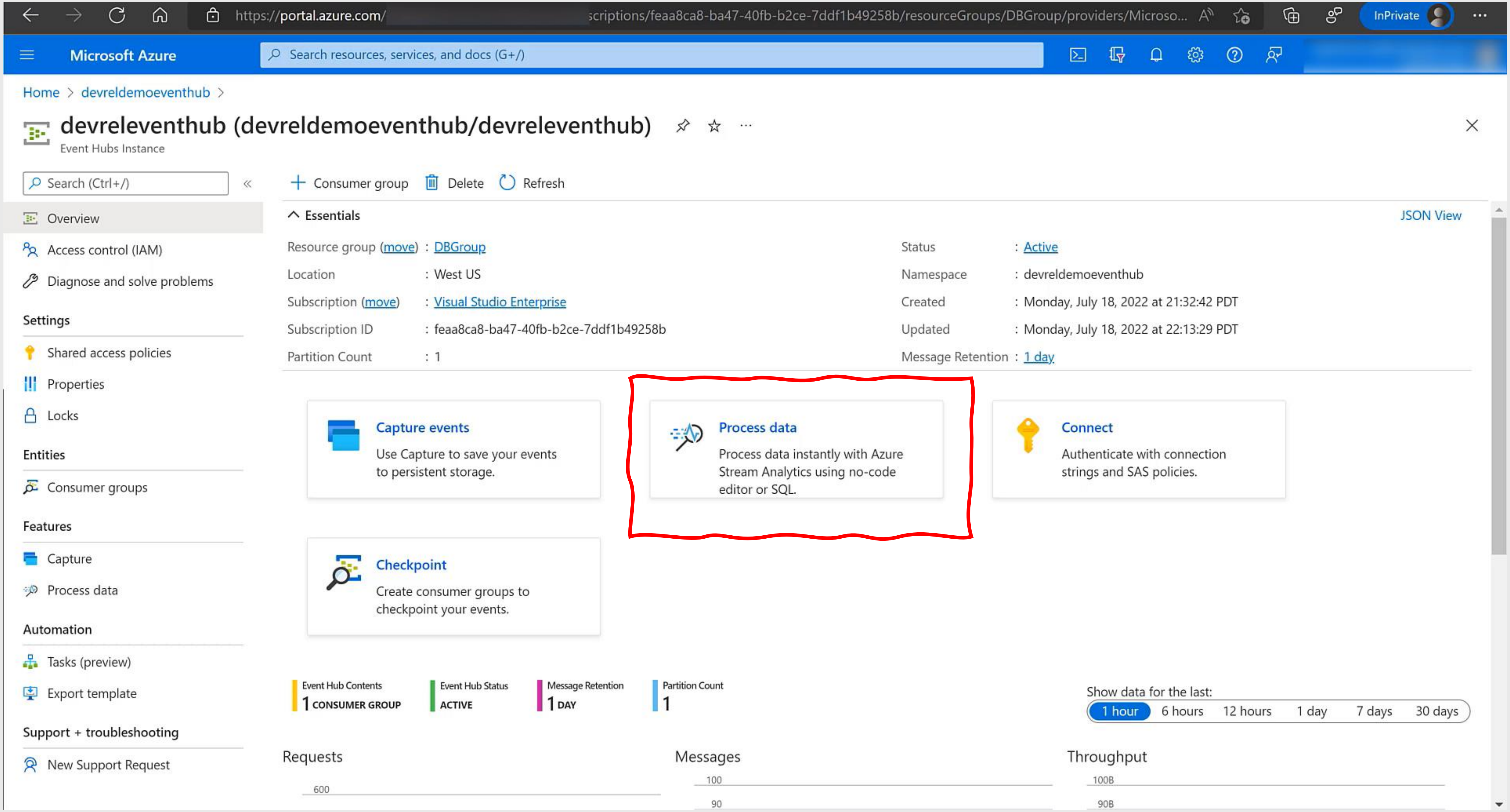
ring
ts
Metrics
Diagnostic settings
Logs
Workbooks

Usage
Users
Sessions
Events
Funnels
User Flows
Cohorts

Last 24 hours

Update map components

Intelligent view: On Off · Low Medium High · Exclude 4xx: On Off · Layout:



https://portal.azure.com/ [redacted] scriptions/feaa8ca8-ba47-40fb-b2ce-7ddf1b49258b/resourceGroups/DBGroup/providers/Microso...

InPrivate

Microsoft Azure

Search resources, services, and docs (G+/)

[redacted]

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

JSON View

Resource group (move)	: DBGroup	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

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	Event Hub Status	Message Retention	Partition Count
1 CONSUMER GROUP	ACTIVE	1 DAY	1

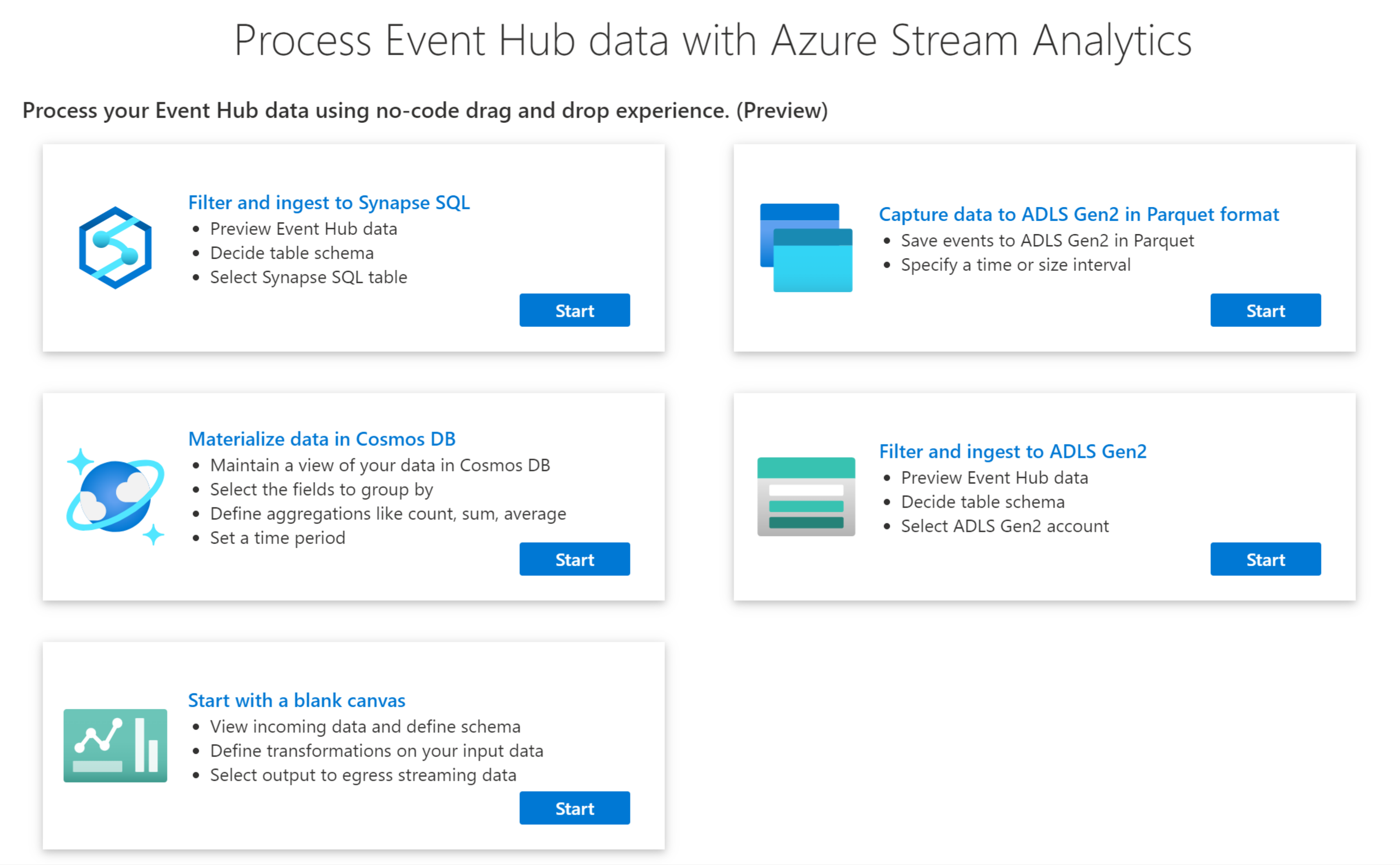
Show data for the last:

1 hour   6 hours   12 hours   1 day   7 days   30 days

Requests

Messages

Throughput



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

Capture data to ADLS Gen2 in Parquet format

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

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

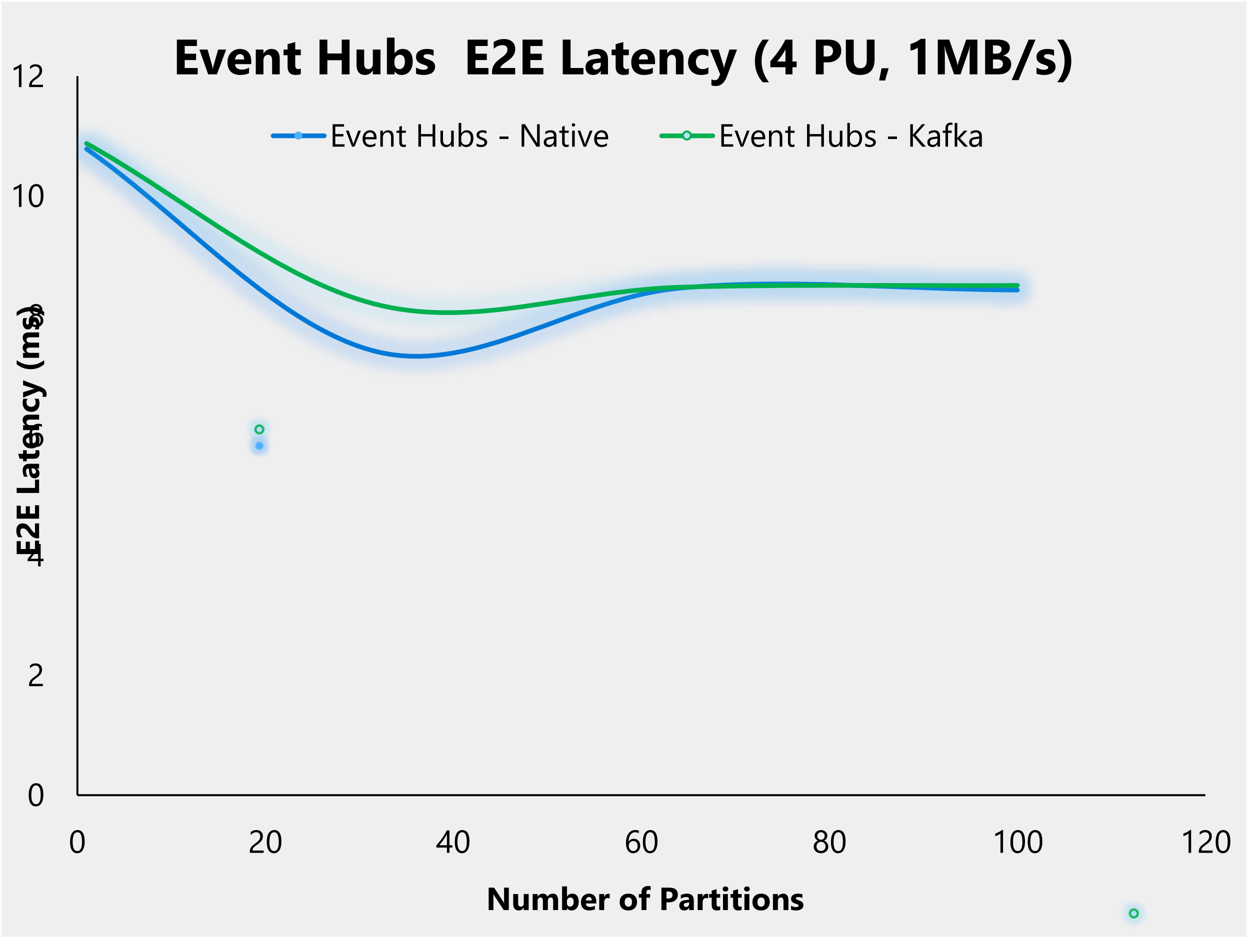
Filter and ingest to ADLS Gen2

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

Start with a blank canvas

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

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

Effective Global Reliability

Uptime – Is the cluster available to accept connections?

100%

Reliability – Are individual operations suceeding?

>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

Microsoft