

I N T E G R A T I N G C O M P O N E N T S U S I N G M E S S A G I N G S E R V I C E S

MODULE OVERVIEW

- EVENT MESSAGING
- INTEGRATION
- INTERNET OF THINGS (IOT)

EVENT MESSAGING

(STORAGE QUEUE, SERVICE BUS)

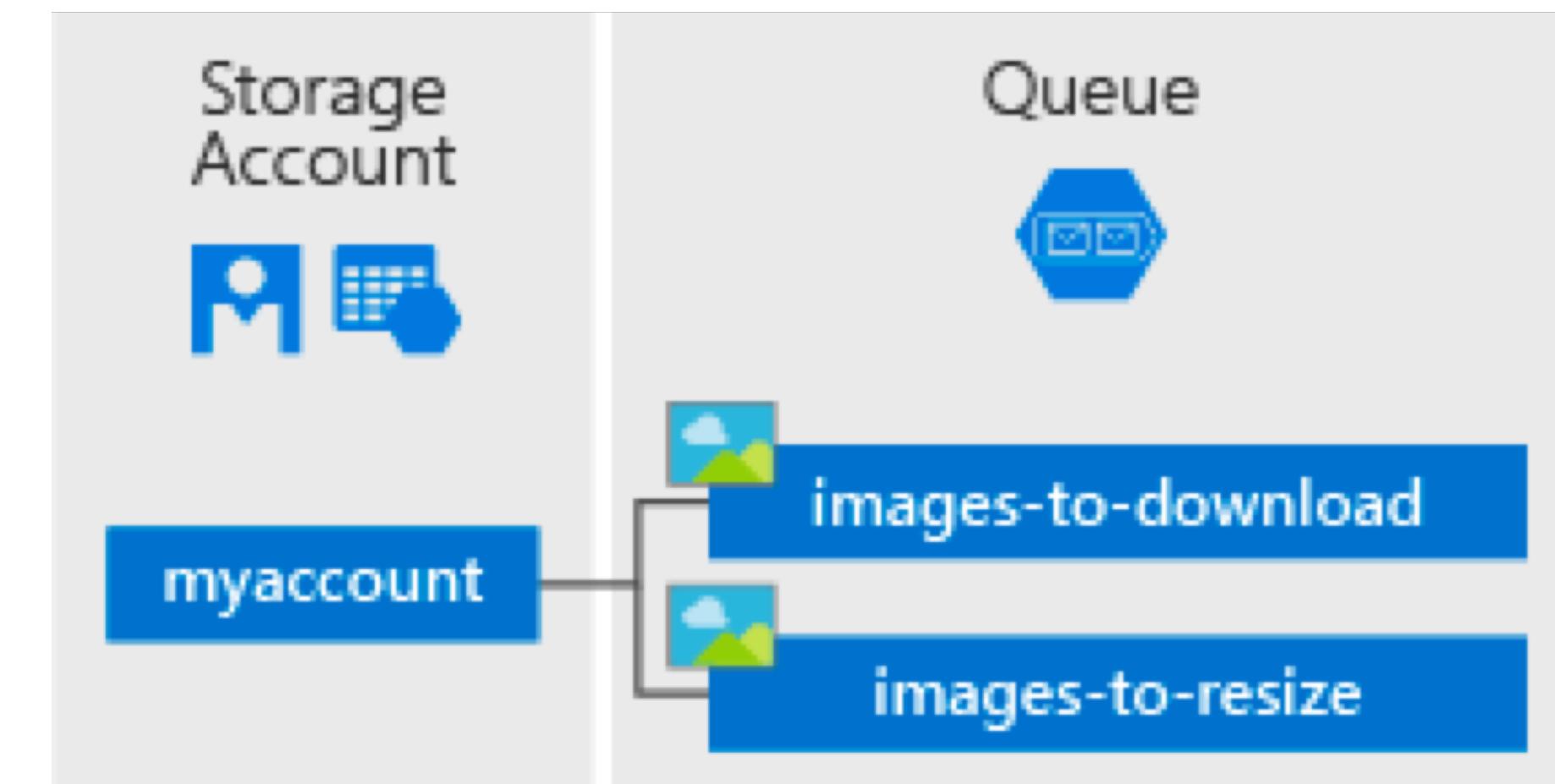
EVENT MESSAGING

- STORAGE QUEUES
- SERVICE BUS
- EVENT GRID

STORAGE QUEUES

- **MESSAGE** – UP TO 64 KB, OVER 80GB IN QUEUE
- **QUEUE** – NO GUARANTEE OF ORDER, [HTTPS://<STORAGE ACCOUNT>.QUEUE.CORE.WINDOWS.NET/<QUEUE>](https://<STORAGE ACCOUNT>.QUEUE.CORE.WINDOWS.NET/<QUEUE>)
- **ACCOUNT** – PART OF THE STORAGE ACCOUNT
- **QUEUE CLIENT** – ANY CODE, THAT CAN READ FROM QUEUE
- **SAS TOKEN** – ACCESS TO THE QUEUE MESSAGES

WHEN TO USE WHAT: [HTTPS://DOCS.MICROSOFT.COM/PL-PL/AZURE/SERVICE-BUS-MESSAGING/SERVICE-BUS-AZURE-AND-SERVICE-BUS-QUEUES-COMPARED-CONTRASTED](https://DOCS.MICROSOFT.COM/PL-PL/AZURE/SERVICE-BUS-MESSAGING/SERVICE-BUS-AZURE-AND-SERVICE-BUS-QUEUES-COMPARED-CONTRASTED)



STORAGE QUEUES

Comparison Criteria	Storage queues	Service Bus queues
Ordering guarantee	No For more information, see the first note in the "Additional Information" section.	Yes - First-In-First-Out (FIFO) (through the use of messaging sessions)
Delivery guarantee	At-Least-Once	At-Least-Once At-Most-Once
Atomic operation support	No	Yes
Receive behavior	Non-blocking (completes immediately if no new message is found)	Blocking with/without timeout (offers long polling, or the " Comet technique ") Non-blocking (through the use of .NET managed API only)
Push-style API	No	Yes OnMessage and OnMessage sessions .NET API.
Receive mode	Peek & Lease	Peek & Lock Receive & Delete
Exclusive access mode	Lease-based	Lock-based
Lease/Lock duration	30 seconds (default) 7 days (maximum) (You can renew or release a message lease using the UpdateMessage API.)	60 seconds (default) You can renew a message lock using the RenewLock API.

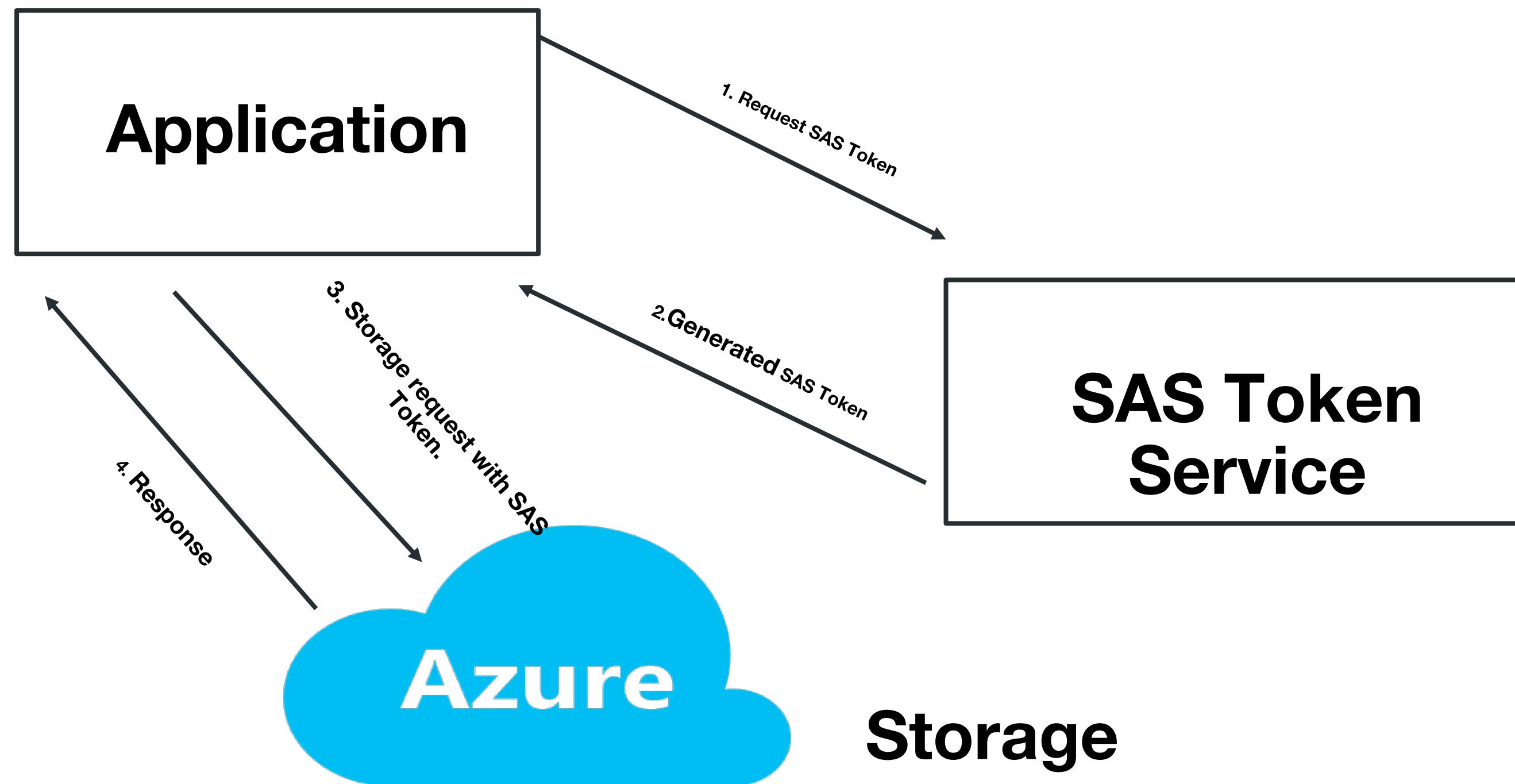
Batched receive	Yes (explicitly specifying message count when retrieving messages, up to a maximum of 32 messages)	Yes (implicitly enabling a pre-fetch property or explicitly through the use of transactions)
Batched send	No	Yes (through the use of transactions or client-side batching)

STORAGE QUEUES – HOW WE HANDLE THE QUEUE

- CREATE/DELETE QUEUE
- MEASURE QUEUE LENGTH
- INSERT MESSAGE INTO QUEUE
- RETRIEVE THE NEXT MESSAGE
- EXTEND MESSAGE LEASE
- PEEK AT THE NEXT MESSAGE
- UPDATE A MESSAGE
- DELETE A MESSAGE

THIS PROCESS IS VERY TYPICAL. CAN BE DONE ALMOST IN ANY LANGUAGE.

STORAGE QUEUES – HOW WE HANDLE THE QUEUE



<HTTPS://SAMF02.QUEUE.CORE.WINDOWS.NET/?>
SV=2018-03-28
&SS=Q
&SRT=SCO
&SP=P
&SE=2018-12-08T06:50:10Z
&ST=2018-12-07T22:50:10Z
&SPR=HTTPS
&SIG=HWOBNBTLGSWJAXGE9F%2FU716Y2J9DXIYZHHI24EF2L30%3D

<HTTPS://DOCS.MICROSOFT.COM/PL-PL/AZURE/STORAGE/QUEUES/STORAGE-NODEJS-HOW-TO-USE-QUEUES>

STORAGE QUEUES – HOW WE HANDLE THE QUEUE

Azure Queue storage scale targets

Resource	Target
Max size of single queue	500 TiB
Max size of a message in a queue	64 KiB
Max number of stored access policies per queue	5
Maximum request rate per storage account	20,000 messages per second assuming 1 KiB message size
Target throughput for single queue (1 KiB messages)	Up to 2000 messages per second

[HTTPS://DOCS.MICROSOFT.COM/EN-US/REST/API/STORAGESERVICES/CONSTRUCTING-AN-ACCOUNT-SAS?REDIRECTEDFROM=MSDN](https://docs.microsoft.com/en-us/rest/api/storageservices/constructing-an-account-sas?redirectedfrom=msdn)

[HTTPS://DOCS.MICROSOFT.COM/EN-US/REST/API/STORAGESERVICES/CONSTRUCTING-A-SERVICE-SAS](https://docs.microsoft.com/en-us/rest/api/storageservices/constructing-a-service-sas)

SERVICE BUS

SERVICE BUS IS A MANAGED MESSAGING INFRASTRUCTURE:

- MASSIVE IN SCALE AND COMPLETELY MANAGED
- ALLOWS YOU TO SCALE OUT YOUR APPLICATIONS AND CONSUMERS KNOWING THAT THE MESSAGING PLATFORM WILL SCALE OUT WITH YOUR APPLICATION

ALLOWS DECOUPLED COMPONENTS TO COMMUNICATE ASYNCHRONOUSLY AND SYNCHRONOUSLY.

IT IS A DIFFERENT OFFERING THAN STORAGE QUEUE, IT IS MORE ENTERPRISE READY THAN STORAGE QUEUE. IT'S WAY MORE EXPENSIVE, BUT THIS IS WORTH.

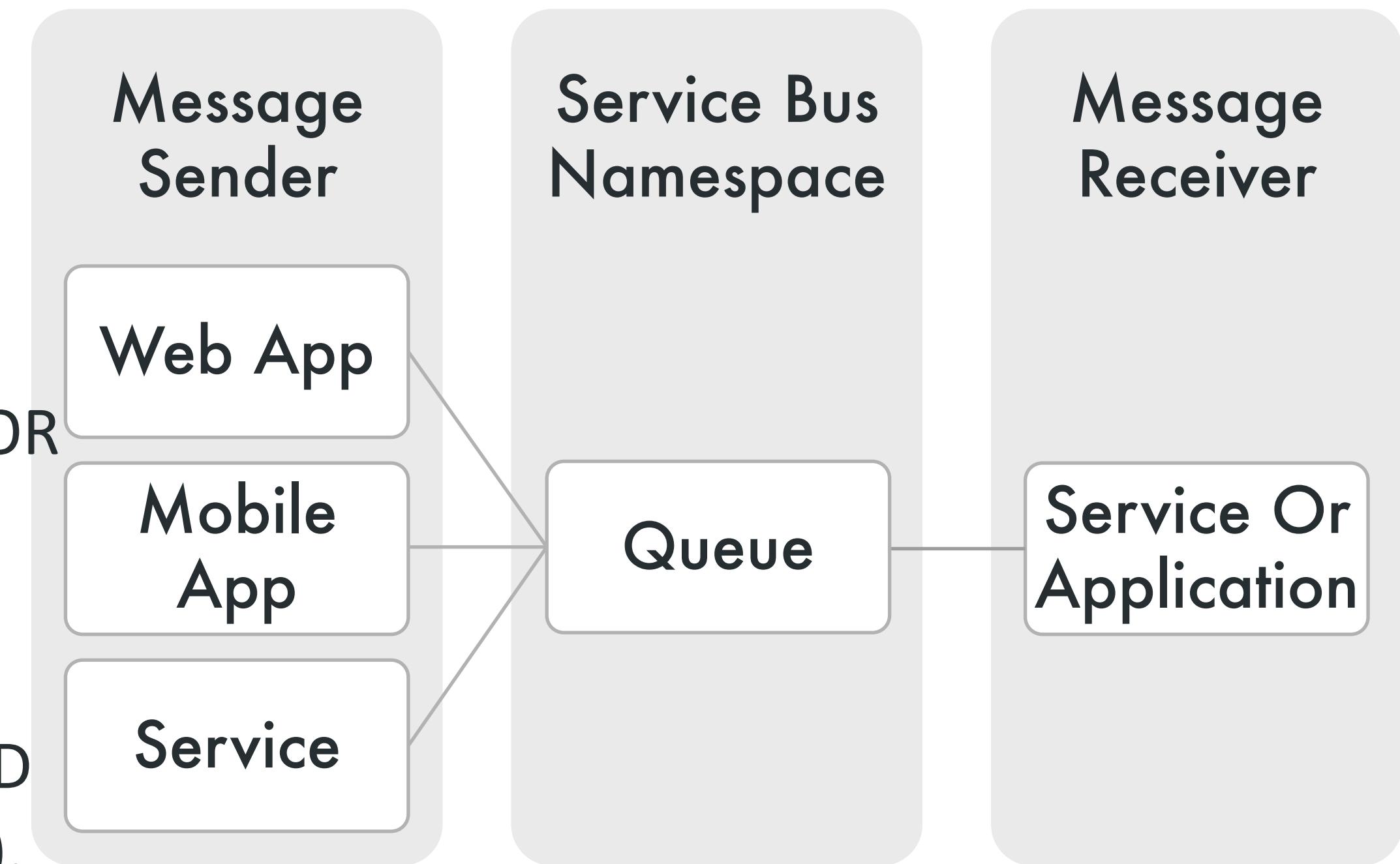
SERVICE BUS

SERVICE BUS QUEUES OFFER A BROKERED MESSAGING COMMUNICATION MODEL:

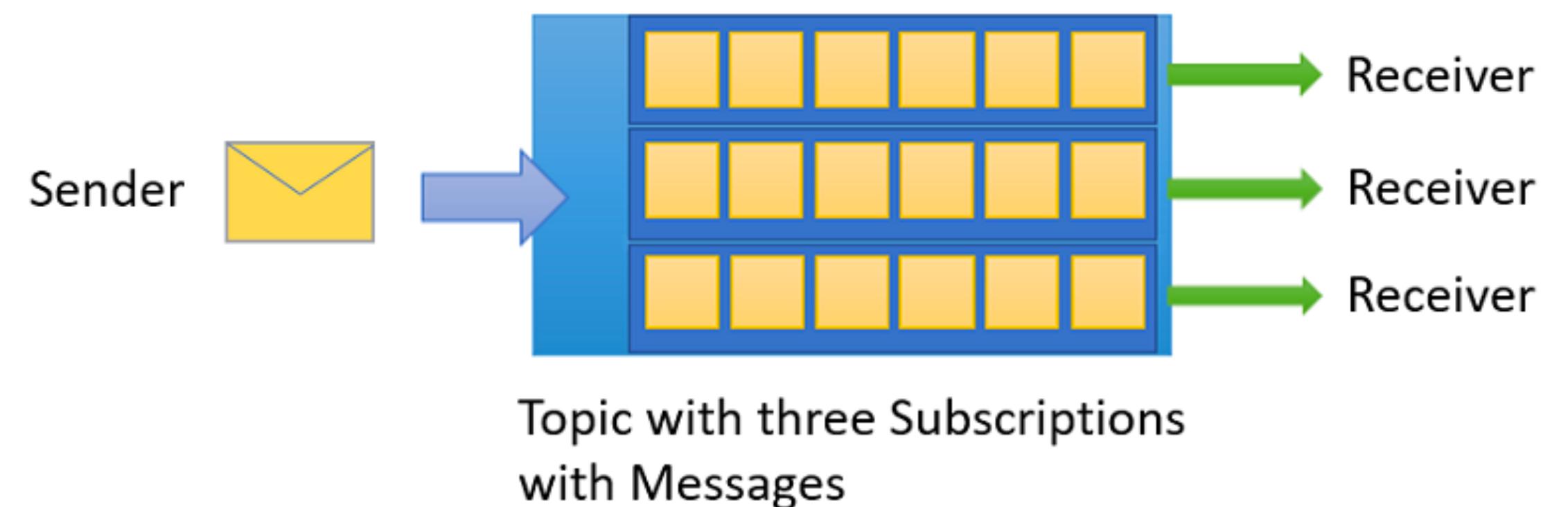
- DISTRIBUTED APPLICATIONS CAN SHARE MESSAGES IN A FIRST IN FIRST OUT (FIFO) PATTERN
- INDIVIDUAL MESSAGES ARE ONLY RECEIVED BY ONE MESSAGE CONSUMER

SAMPLE USE CASES:

- MESSAGING: TRANSFER BUSINESS DATA, SUCH AS SALES OR PURCHASE ORDERS, JOURNALS, OR INVENTORY MOVEMENTS.
- DECOUPLE APPLICATIONS: IMPROVE RELIABILITY AND SCALABILITY OF APPLICATIONS AND SERVICES (CLIENT AND SERVICE DO NOT HAVE TO BE ONLINE AT THE SAME TIME).
- TOPICS AND SUBSCRIPTIONS: ENABLE 1:N RELATIONSHIPS BETWEEN PUBLISHERS AND SUBSCRIBERS.
- MESSAGE SESSIONS: IMPLEMENT WORKFLOWS THAT REQUIRE MESSAGE ORDERING OR MESSAGE DEFERRAL.



SERVICE BUS – USAGE SCENARIOS



- TOPICS CAN HAVE MULTIPLE, INDEPENDENT SUBSCRIPTIONS.
- A SUBSCRIBER TO A TOPIC CAN RECEIVE A COPY OF EACH MESSAGE SENT TO THAT TOPIC.
- SUBSCRIPTIONS ARE NAMED ENTITIES, WHICH ARE DURABLY CREATED BUT CAN OPTIONAL EXPIRE OR AUTO-DELETE.
- YOU MAY NOT WANT INDIVIDUAL SUBSCRIPTIONS TO RECEIVE ALL MESSAGES SENT TO A TOPIC. IF SO, YOU CAN USE RULES AND FILTERS TO DEFINE CONDITIONS THAT TRIGGER OPTIONAL ACTIONS, FILTER SPECIFIED MESSAGES, AND SET OR MODIFY MESSAGE PROPERTIES.

SERVICE BUS – ADVANCED SCENARIOS

- MESSAGE SESSIONS
- AUTO-FORWARDING – CHAIN A QUEUE OR SUBSCRIPTION
- DEAD-LETTERING (DLQ)
- SCHEDULED DELIVERY
- MESSAGE DEFERRAL
- BATCHING
- TRANSACTIONS
- FILTERING AND ACTIONS
- AUTO-DELETE ON IDLE
- DUPLICATE DETECTION
- SAS, RBAC & MANAGED IDENTITY
- GEO-DISASTER RECOVERY
- SECURITY



[HTTPS://GITHUB.COM/AZURE/AZURE-SERVICE-BUS-DOTNET/](https://github.com/Azure/Azure-Service-Bus-DotNet/)

SERVICE BUS – ADVANCED SCENARIOS

Create queue

Service Bus

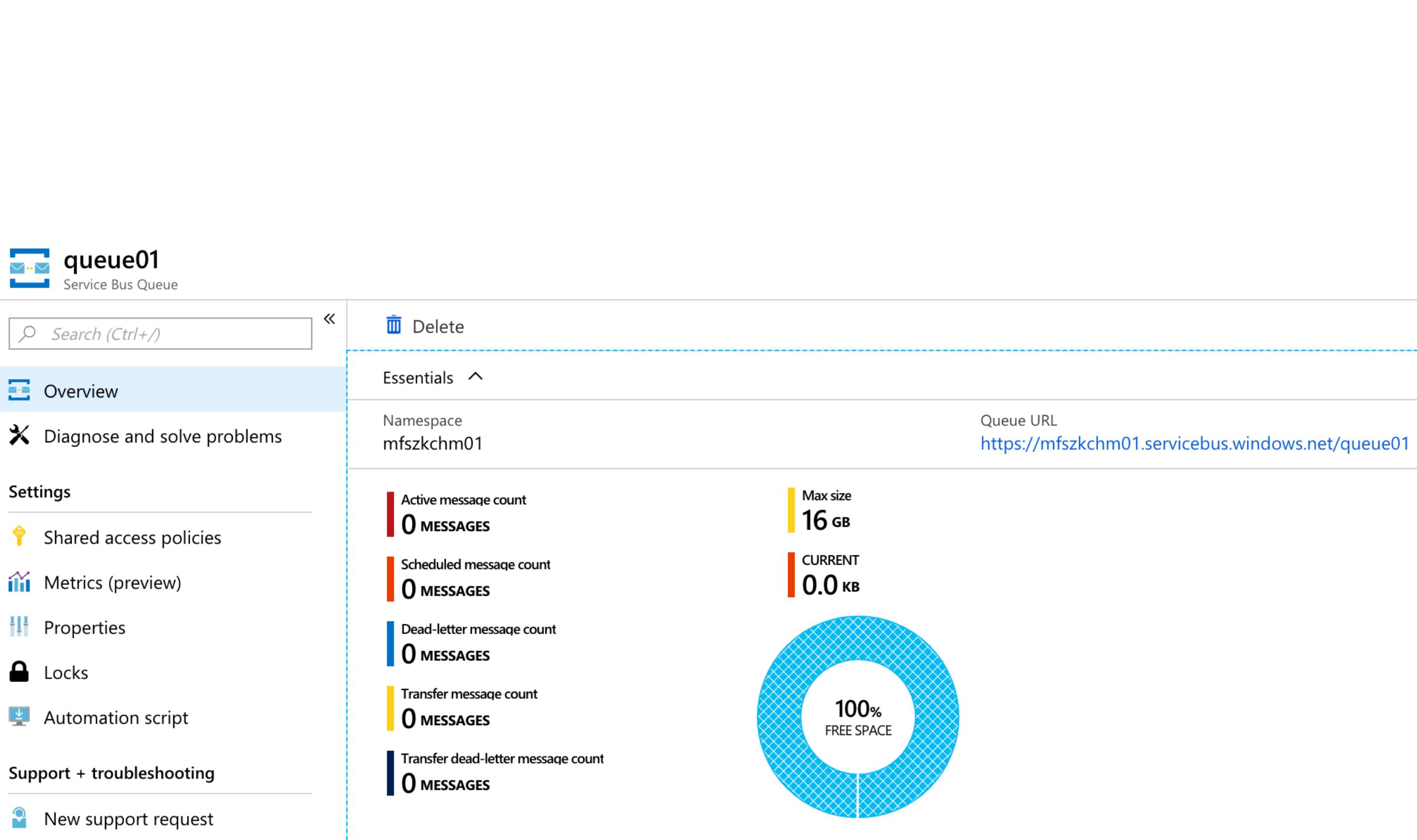
* Name

Max queue size

Message time to live Days: 14, Hours: 0, Minutes: 0, Seconds: 0

Lock duration Days: 0, Hours: 0, Minutes: 0, Seconds: 30

Enable duplicate detection Enable dead lettering on message expiration Enable sessions Enable partitioning



The screenshot shows the Azure portal interface for creating a Service Bus queue. On the left, there's a 'Create queue' dialog box with fields for name, max queue size, message time to live, lock duration, and various enablement options like duplicate detection and sessions. On the right, the 'queue01' overview page is displayed, showing basic queue statistics: Namespace (mfszkchm01), Queue URL (https://mfszkchm01.servicebus.windows.net/queue01), Active message count (0 MESSAGES), Scheduled message count (0 MESSAGES), Dead-letter message count (0 MESSAGES), Transfer message count (0 MESSAGES), and Transfer dead-letter message count (0 MESSAGES). A large circular progress bar indicates 100% free space.

SERVICE BUS RELAY

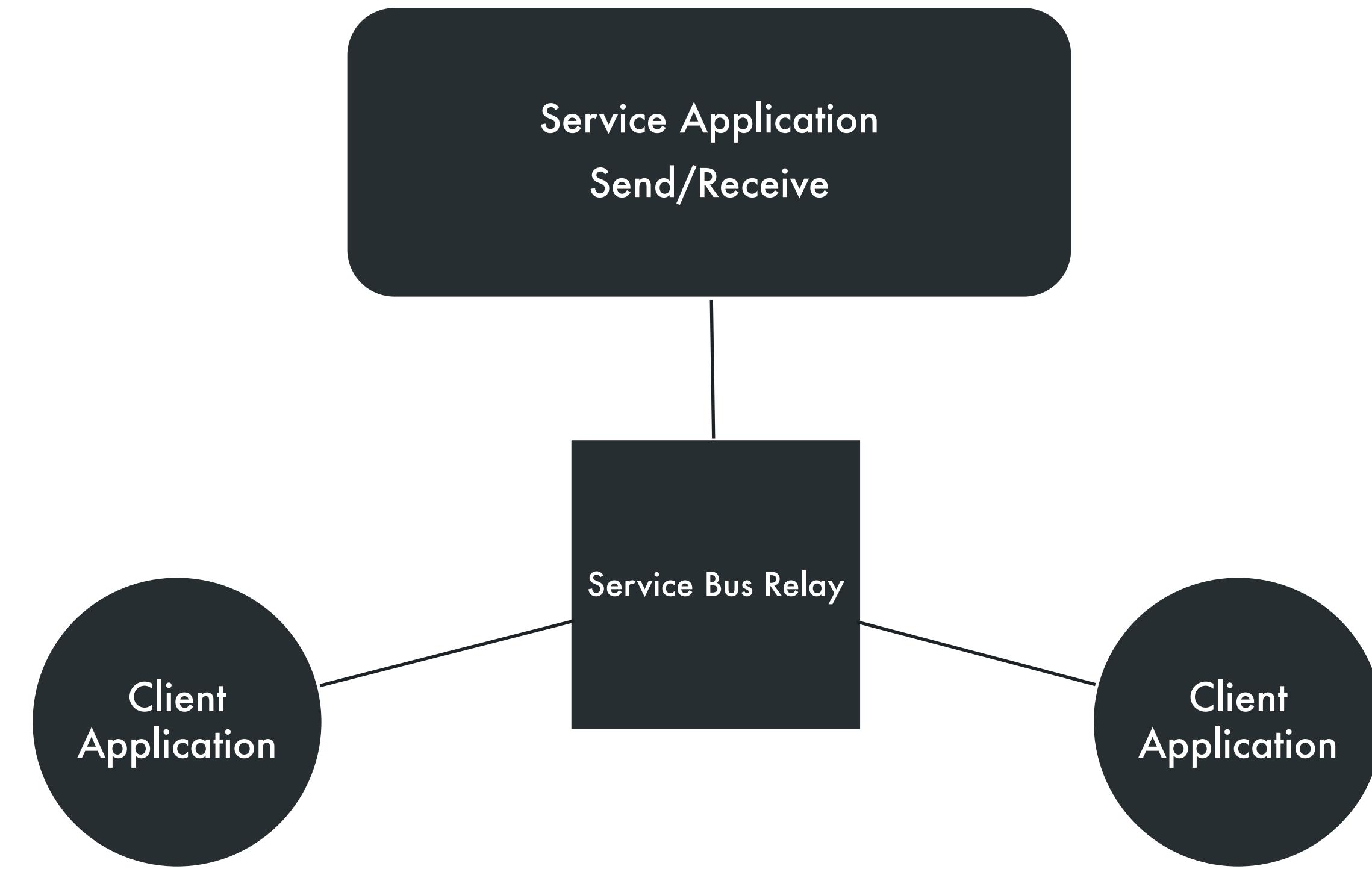
RELAYS PROVIDE A MECHANISM TO CONNECT DISTRIBUTED CLIENT APPLICATIONS OR CLOUD SERVICES TO ON-PREMISES ENDPOINT:

- IT ALLOWS FOR UNIDIRECTIONAL OR BI-DIRECTIONAL COMMUNICATION
- IT RELAYS MESSAGES DIRECTLY TO AN ENDPOINT WITHOUT ANY BROKERING OF THE MESSAGE

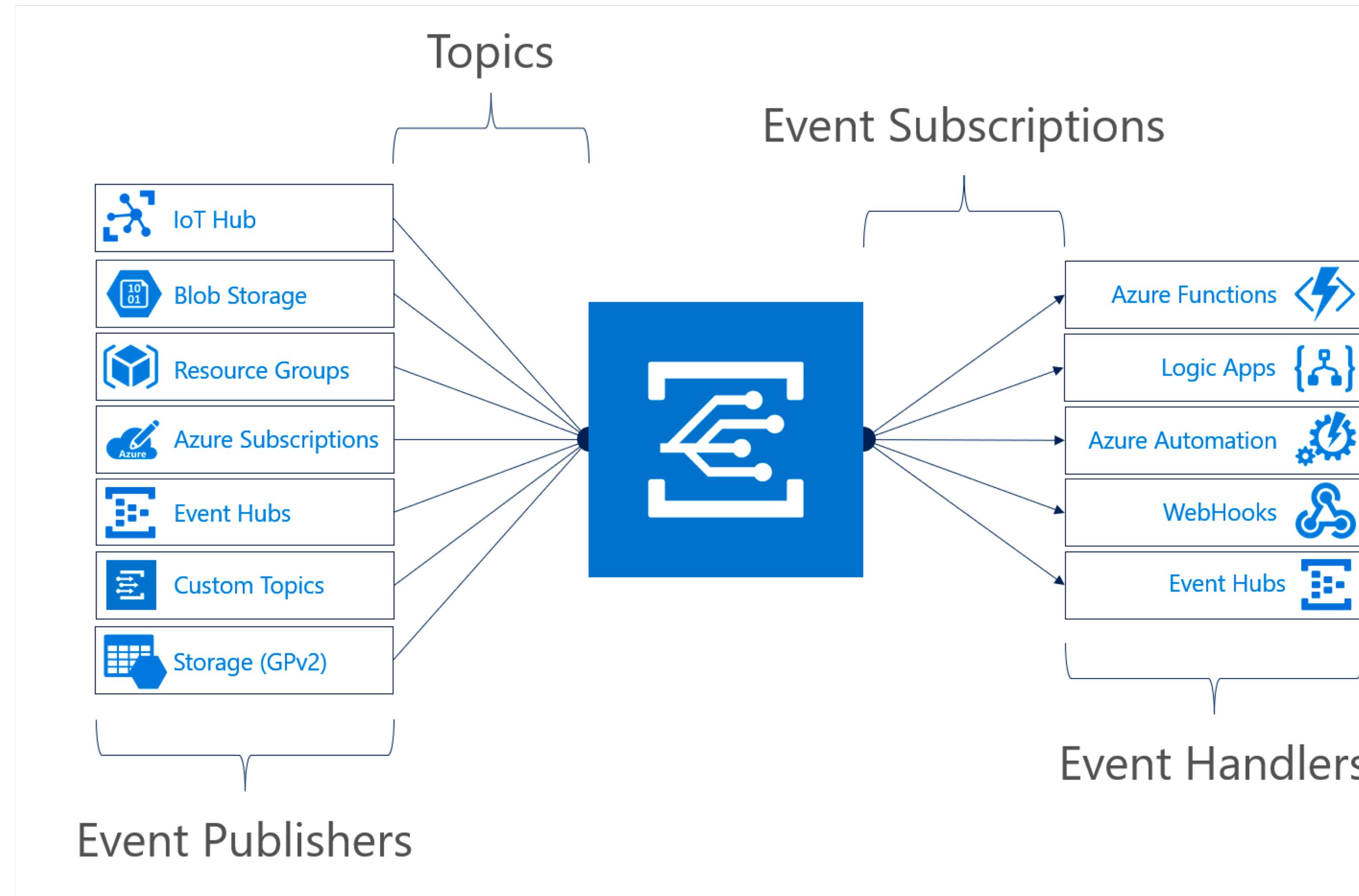
APPLICATIONS ESTABLISH AN OUTBOUND CONNECTION TO THE RELAY AND THE RELAY MANAGES THE TRANSPORT OF THE MESSAGES

EMPHASIZE THAT RELAYS PASS MESSAGES AROUND AND ARE NOT INTENDED FOR MESSAGE STORAGE. RELAYS CAN BE USED WITH CLOUD SERVICES.

SERVICE BUS RELAY



EVENT GRID – AUTOMATE OPERATIONS



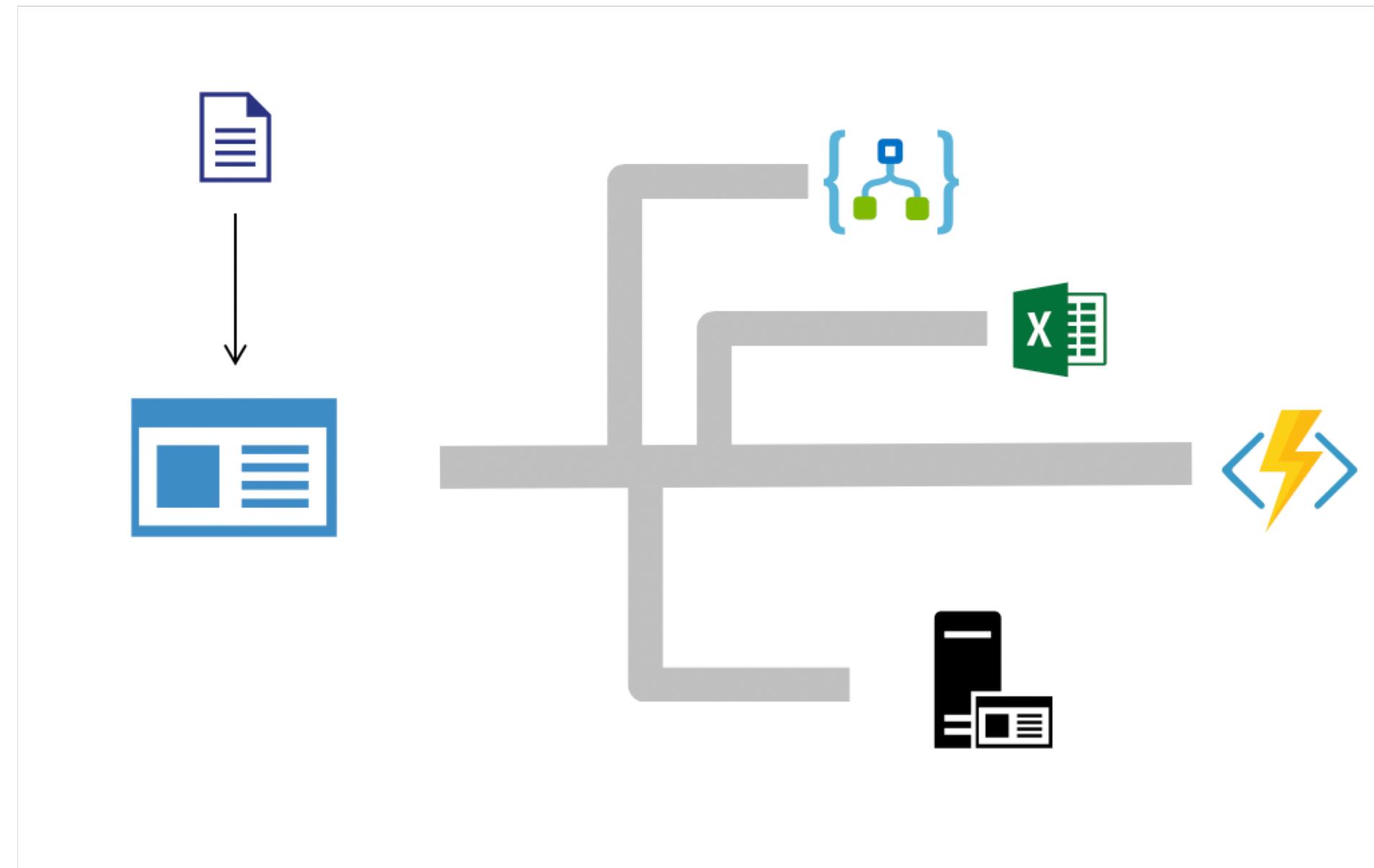
EVENT GRID – AUTOMATE OPERATIONS

EVENT GRID CAN PUBLISH ARM EVENTS INCLUDING:

- RESOURCE CREATION
- RESOURCE MODIFICATION/DELETION
- DEPLOYMENT OF MULTIPLE RESOURCES TO A RESOURCE GROUP
- CREATION OR DELETION OF A RESOURCE GROUP

AZURE SERVICES CAN RESPOND TO AN EVENT GRID RESOURCE-BASED EVENT BY PERFORMING AUTOMATION ACTIONS:

- A LOGIC APP CAN MODIFY A NEWLY CREATED DATABASE
- AZURE AUTOMATION CAN MANAGE A NEW VM
- METADATA ABOUT A RESOURCE DEPLOYMENT CAN BE STORED IN AZURE STORAGE USING AN AZURE FUNCTION

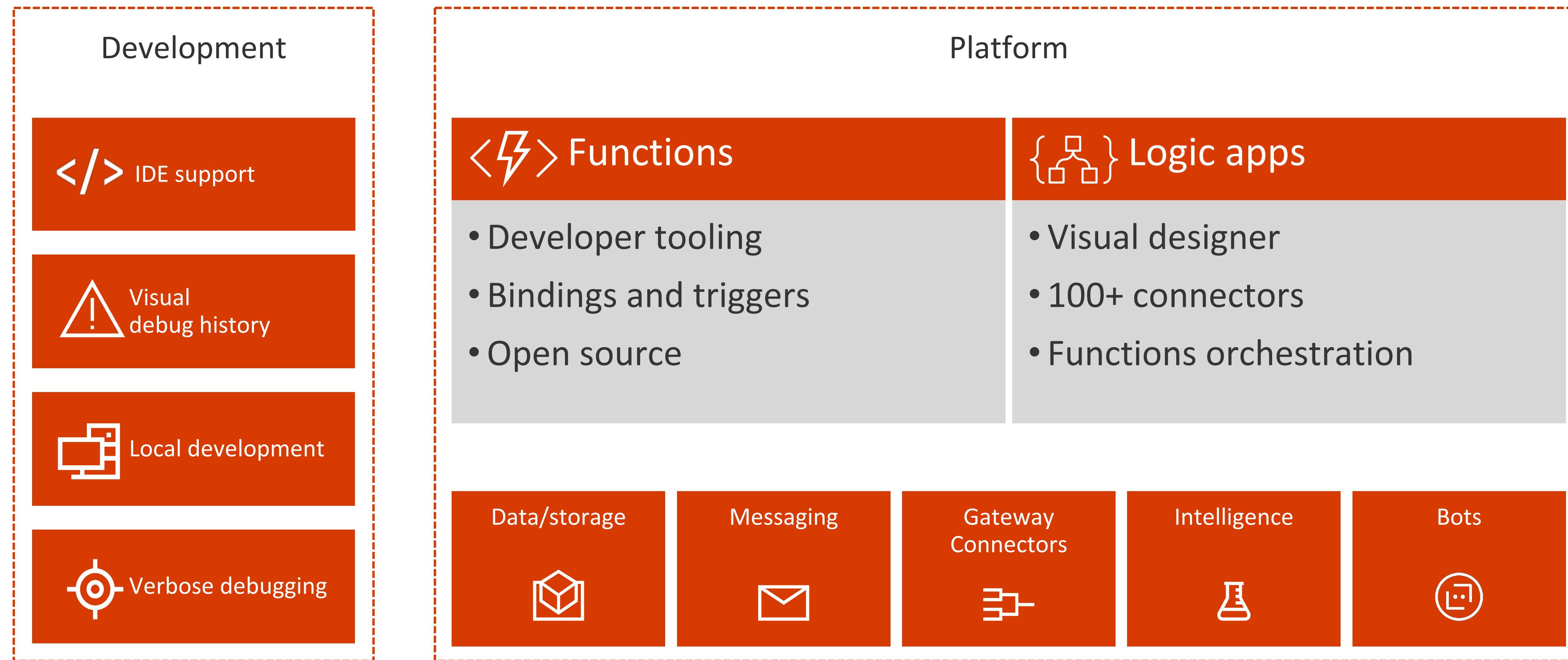


INTEGRATION

INTEGRATION

- SERVERLESS INTEGRATION
- NOTIFICATION HUB

INTEGRATION



INTEGRATION – LOGIC APPS

CLOUD APIs AND PLATFORM:

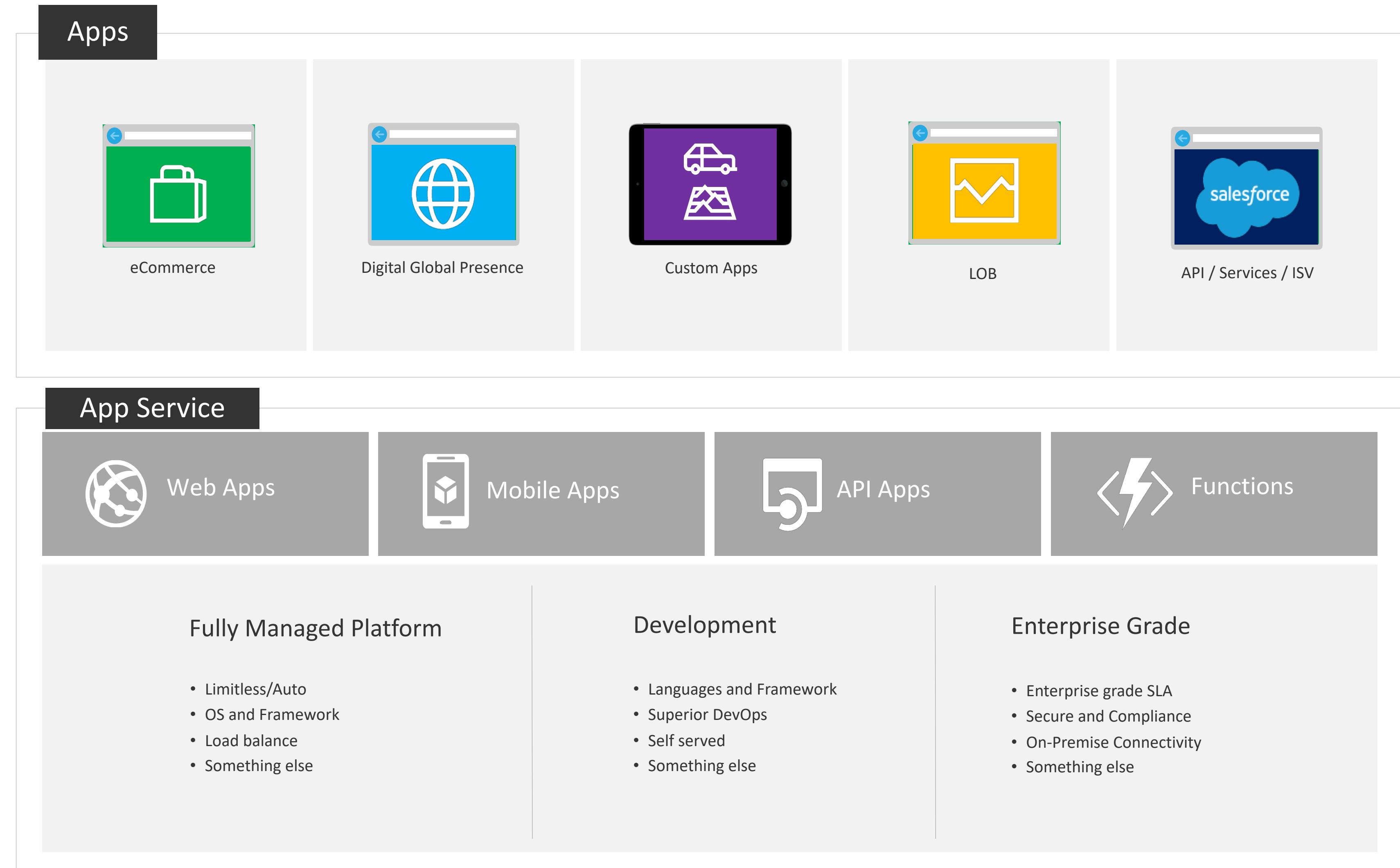
- SUPPORTS OVER 250 BUILT-IN CONNECTORS
- SCALES TO MEET YOUR NEEDS
- ENABLES RAPID DEVELOPMENT
- EXTENDS WITH CUSTOM APIs AND FUNCTIONS

API CONNECTIONS:

- AUTHENTICATE ONCE AND REUSE

- appFigures
- Asana
- Azure API Management
- Azure App Services
- Azure Automation
- Azure Cognitive Face API
- Azure Cognitive LUIS
- Azure Cognitive Text Analytics
- Azure Cognitive Vision
- Azure Data Lake Store
- Azure Document DB
- Azure Event Hub
- Azure Functions
- Azure Machine Learning
- Azure Resource Manager
- Azure Service Bus
- Azure SQL
- Azure Storage Blob
- Azure Storage Queues
- Basecamp
- Bing Search
- BitBucket
- Bitly
- Blogger
- Box
- Buffer
- Campfire
- Chatter
- Common Data Service
- Disqus
- DocuSign
- Dropbox
- Dynamics AX Online
- Dynamics CRM Online
- Dynamics CRM Service Bus
- Dynamics Financials
- Dynamics Operations
- Easy Redmine
- Eventbrite
- Facebook
- FreshBooks
- Freshdesk
- GitHub
- Gmail
- Google Calendar
- Google Contacts
- Google Drive
- Google Sheets
- Google Tasks
- GoTo Meeting
- GoTo Training
- GoTo Webinar
- Harvest
- HelloSign
- Infusionsoft
- JIRA
- Insightly
- Instagram
- Instapaper
- MailChimp
- Mandrill
- Medium
- Microsoft Project Online
- Microsoft Translator
- MSN Weather
- Muhimbi PDF
- Office 365
- Office 365 Users
- Office 365 Video
- OneDrive
- OneDrive for Business
- OneNote
- Outlook.com
- Outlook Tasks
- PagerDuty
- Pinterest
- Pipedrive
- Pivotal Tracker
- Power BI
- Project Online
- Redmine
- Salesforce
- Salesforce Chatter
- SendGrid
- SharePoint Online
- Slack
- SmartSheet
- SparkPost
- Stripe
- Survey Monkey
- Todoist
- Toodledo
- Trello
- Twilio
- Twitter
- Typeform
- UserVoice
- VS Team Services
- Webmerge
- Wordpress
- Wunderlist
- Yammer
- YouTube
- Zendesk

AZURE FUNCTION



AZURE FUNCTION – HOW IT WORKS

METHODS OF EXECUTION:

- TRIGGERS
- WEBHOOKS

LANGUAGE OF CHOICE:

- C#, NODE.JS, PYTHON, JAVA

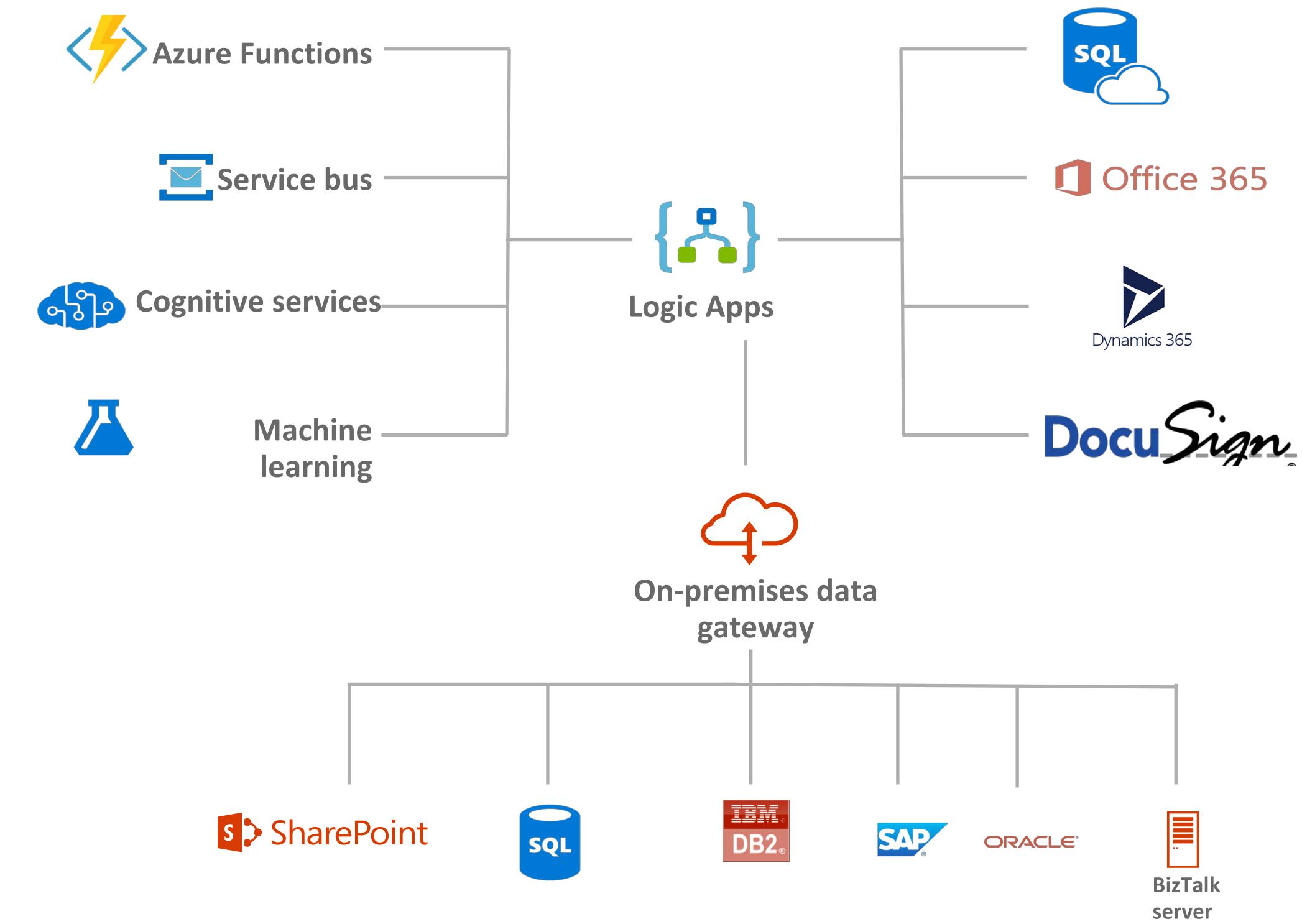
PRICING OPTIONS:

- DYNAMIC (PAY-PER-USE)
- APP SERVICE PLAN (DEDICATED)

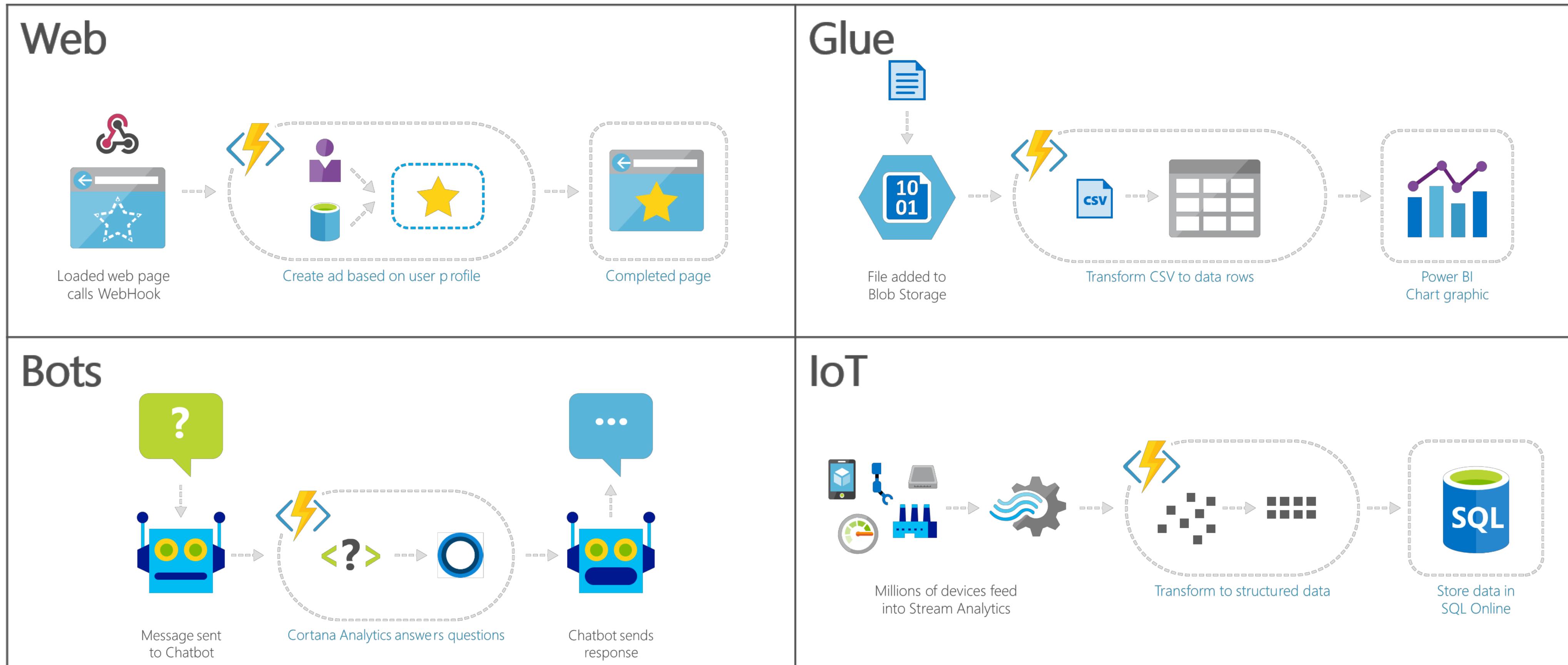
INTEGRATIONS:

- COSMOS DB
- EVENT HUBS
- MOBILE APPS (TABLES)
- NOTIFICATION HUBS, SERVICE BUS, STORAGE
- GITHUB (WEBHOOKS), ON-PREMISES (USING SERVICE BUS)

AZURE FUNCTION - INTEGRATION



AZURE FUNCTION - INTEGRATION



NOTIFICATION HUB

MANAGED INFRASTRUCTURE FOR SENDING PUSH
NOTIFICATIONS TO MOBILE DEVICES:

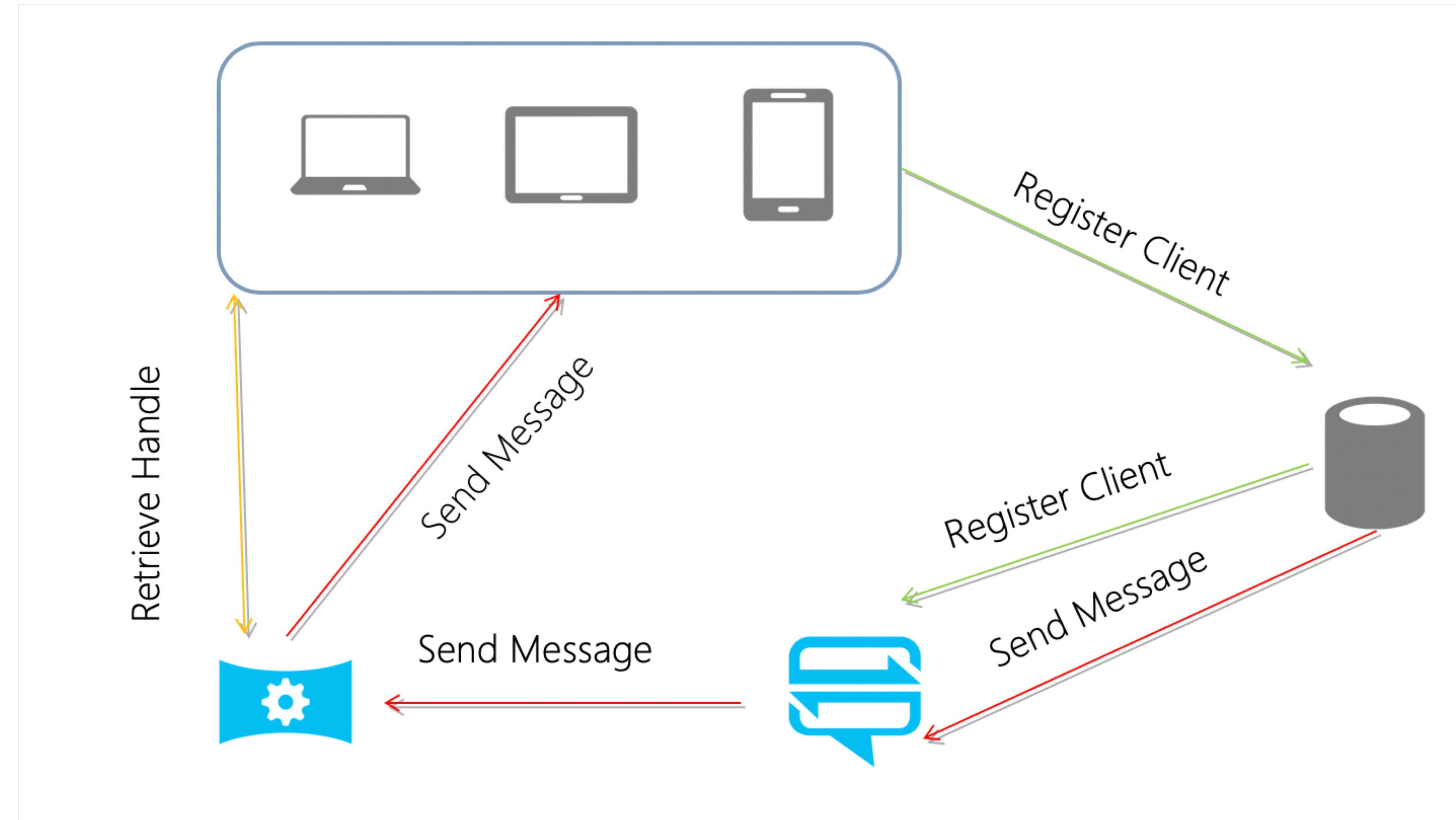
- MULTIPLATFORM
- SCALABLE

SIMPLE SDK:

- AVAILABLE ON MANY MAJOR MOBILE PLATFORMS
- BROADCAST TO MANY USERS OR TARGET SPECIFIC USERS
- YOU CAN BROADCAST A MESSAGE TO A LARGE GROUP OF USERS, SEND A MESSAGE TO A TARGETED SUBSET OF USERS OR EVEN NOTIFY A SPECIFIC USER. THE PLATFORM HANDLES THE BATCHING, TRANSPORT AND MANAGEMENT OF THE MESSAGES.

- SDKS AVAILABLE FOR MAJOR PLATFORMS
- TEMPLATE SUPPORT
- SUPPORT FOR FILTERING RECIPIENTS BY TAG

NOTIFICATION HUB



INTEGRATION - SUMMARY

- SERVERLESS INTEGRATION
- NOTIFICATION HUB

IOT (EVENT HUB)

IOT (EVENT HUB)

- EVENT HUB
- IOT HUB

EVENT HUB

How big is Big Data

1.8 Trillion

Requests per day
Azure Event Hubs

269 Billion

Emails per day

1.4 Billion

Active daily users on Facebook

3.8 Billion

Internet users

22 Billion

SMS messages per day

60 Billion

GB per day data created

5.2 Billion

Google searches per day

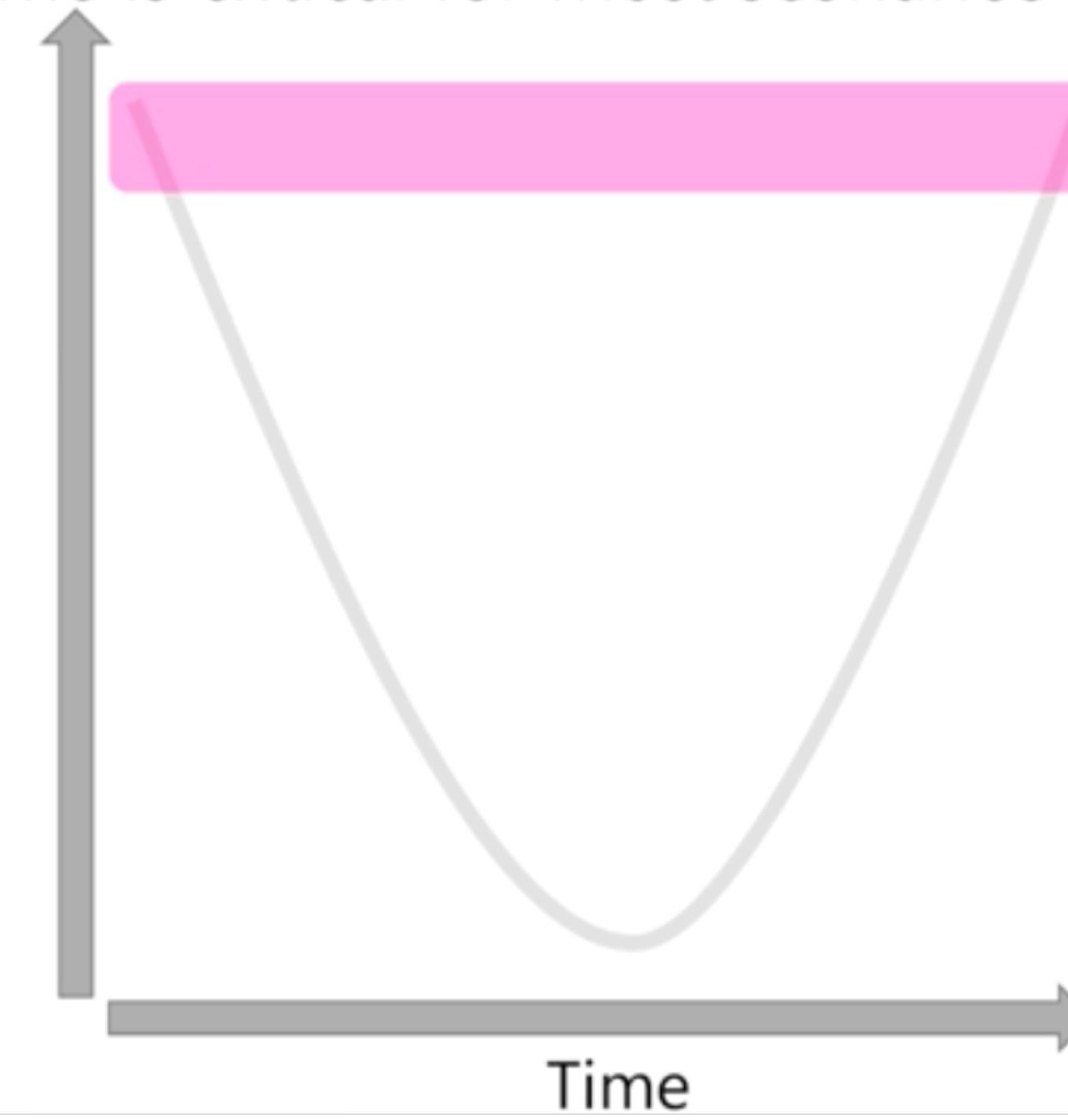
4 Million

Hours of content uploaded to
YouTube every day

Why faster is important: The Time Value of Data

The ability to use historical trends in real-time is critical for most scenarios

- Immediately valuable
- Loses value over time
- Regains value over long time



EVENT HUB

EVENT HUBS IS A PARTITIONED CONSUMER MESSAGING SERVICES

- PUBLISH AND SUBSCRIBE TO STREAMS OF RECORDS
- SIMILAR TO A MESSAGE QUEUE OR ENTERPRISE MESSAGING SYSTEM
- STORE STREAMS OF RECORDS IN A FAULT-TOLERANT MANNER
- PROCESS STREAMS OF RECORDS “AS THEY OCCUR”

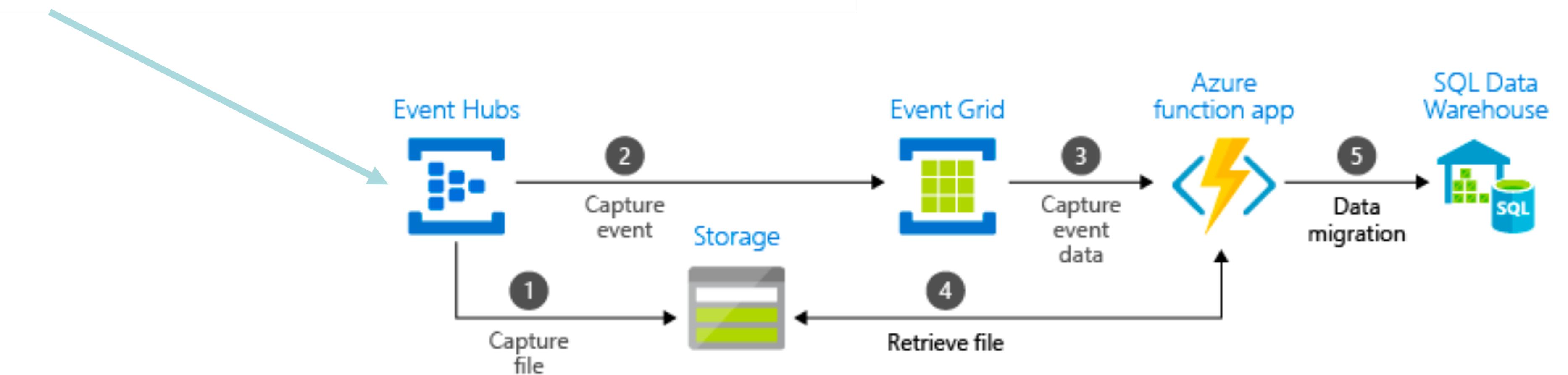
IDEAL FOR BUILDING APPLICATIONS THAT TRANSFORM OR REACT TO STREAMS OF DATA

- AZURE EVENT HUBS IS A BIG DATA STREAMING PLATFORM AND EVENT INGESTION SERVICE, CAPABLE OF RECEIVING AND PROCESSING MILLIONS OF EVENTS PER SECOND.
- EVENT HUBS CAN PROCESS AND STORE EVENTS, DATA, OR TELEMETRY PRODUCED BY DISTRIBUTED SOFTWARE AND DEVICES.
- DATA SENT TO AN EVENT HUB CAN BE TRANSFORMED AND STORED BY USING ANY REAL-TIME ANALYTICS PROVIDER OR BATCHING/STORAGE ADAPTERS.
- EVENT HUBS FOR APACHE KAFKA ENABLES NATIVE KAFKA CLIENTS, TOOLS, AND APPLICATIONS SUCH AS MIRROR MAKER, APACHE FLINK, AND AKKA STREAMS TO WORK SEAMLESSLY WITH EVENT HUBS WITH ONLY CONFIGURATION CHANGES.

EVENT HUB

Comparison of services

Service	Purpose	Type	When to use
Event Grid	Reactive programming	Event distribution (discrete)	React to status changes
Event Hubs	Big data pipeline	Event streaming (series)	Telemetry and distributed data streaming
Service Bus	High-value enterprise messaging	Message	Order processing and financial transactions



EVENT HUB

INPUT STREAMING:

- RECEIVES HIGH-VELOCITY MESSAGE STREAMS IN A MULTI-CONSUMER GROUP

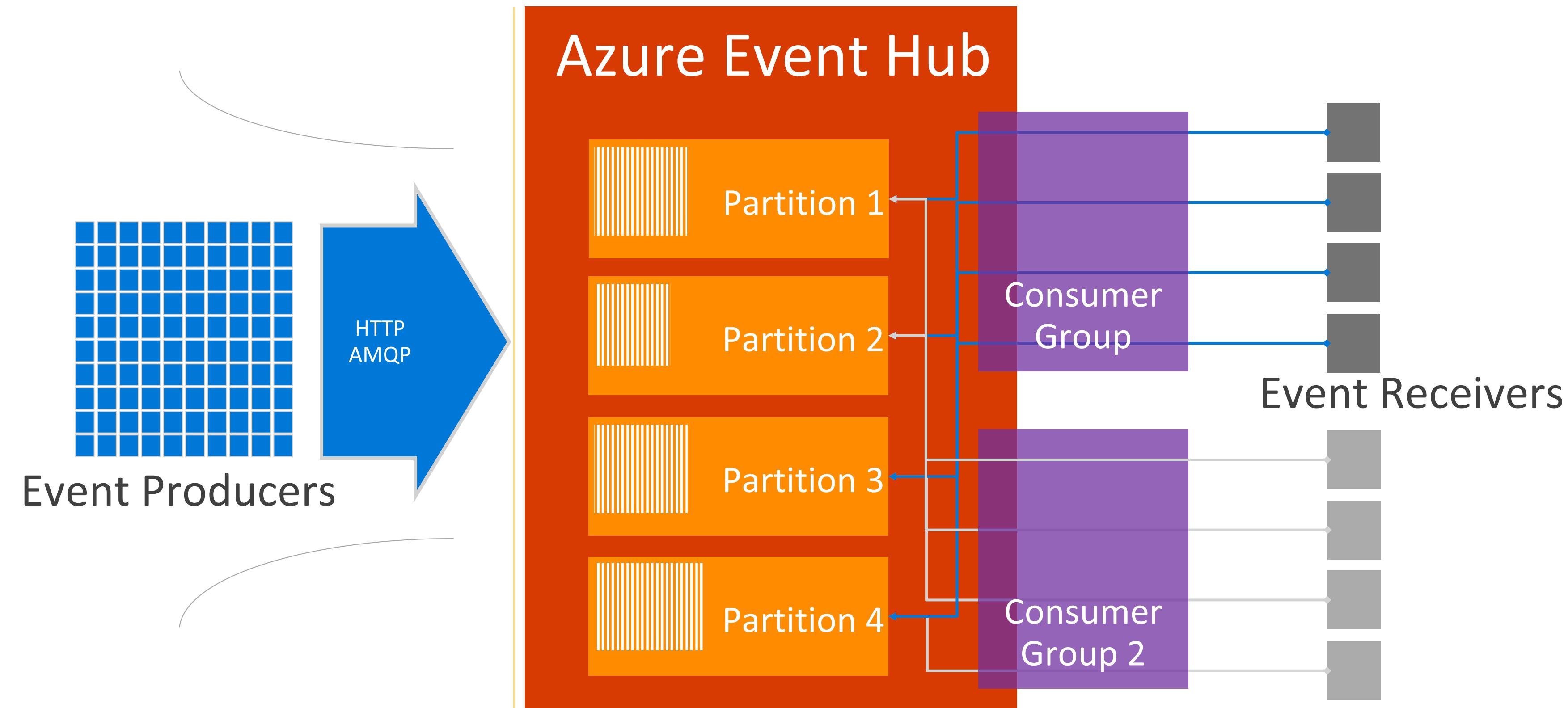
ISOLATED READ:

- STORES “POINTERS” FOR EACH READER SO THEY CAN RESUME AT A SPECIFIC POINT-IN-TIME IN READING TIME-BASED MESSAGES FROM THE QUEUE

OPEN PROTOCOLS:

- SUPPORTS AMQP 1.0
- REST API FOR MANAGEMENT

EVENT HUB – CONSUMER GROUPS

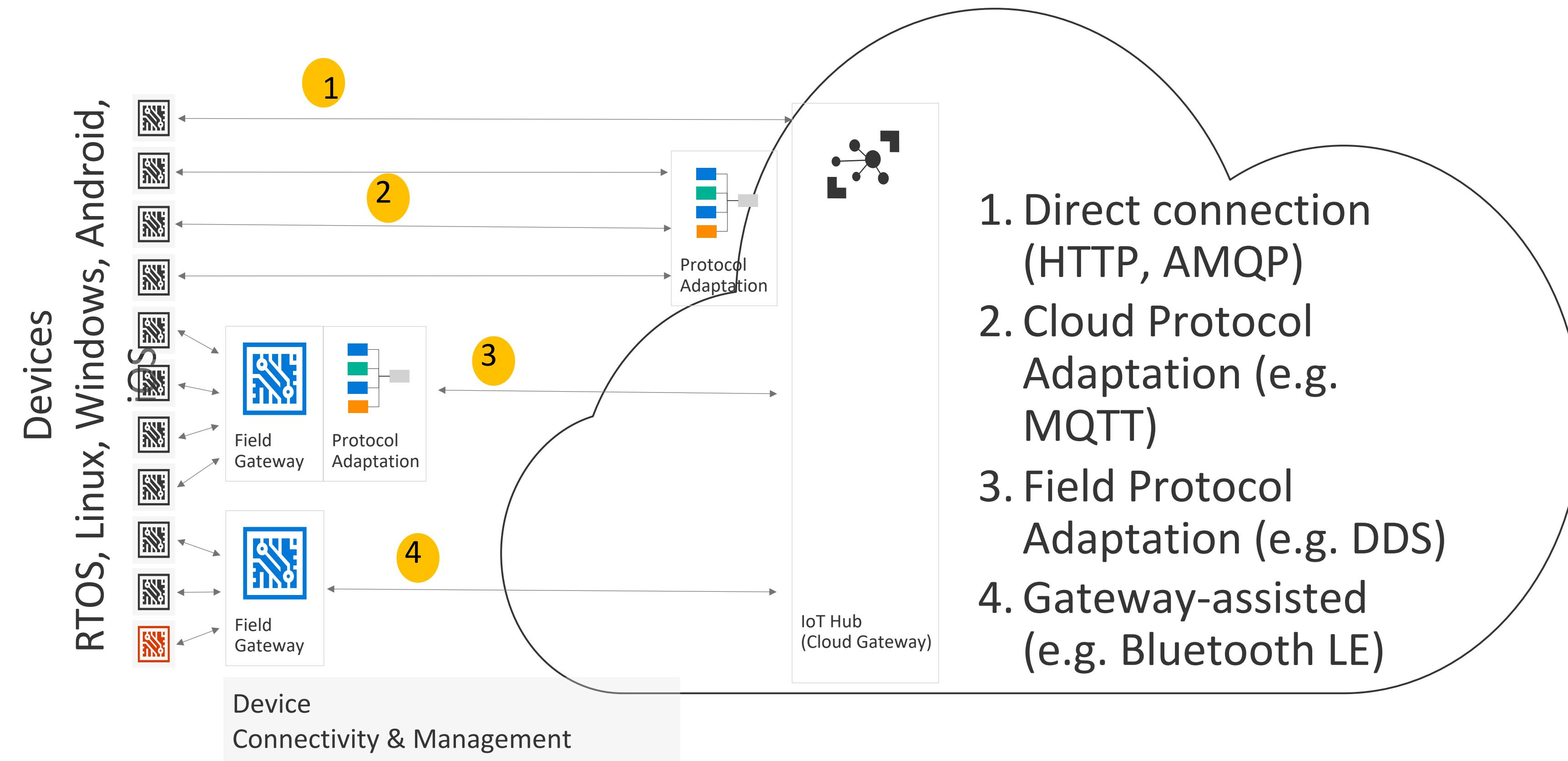


IOT HUB

IOT HUBS BUILDS ON THE FEATURES IN EVENT HUBS BY ADDING ADDITIONAL FUNCTIONALITY THAT IS COMMONLY NEEDED IN IOT APPLICATIONS

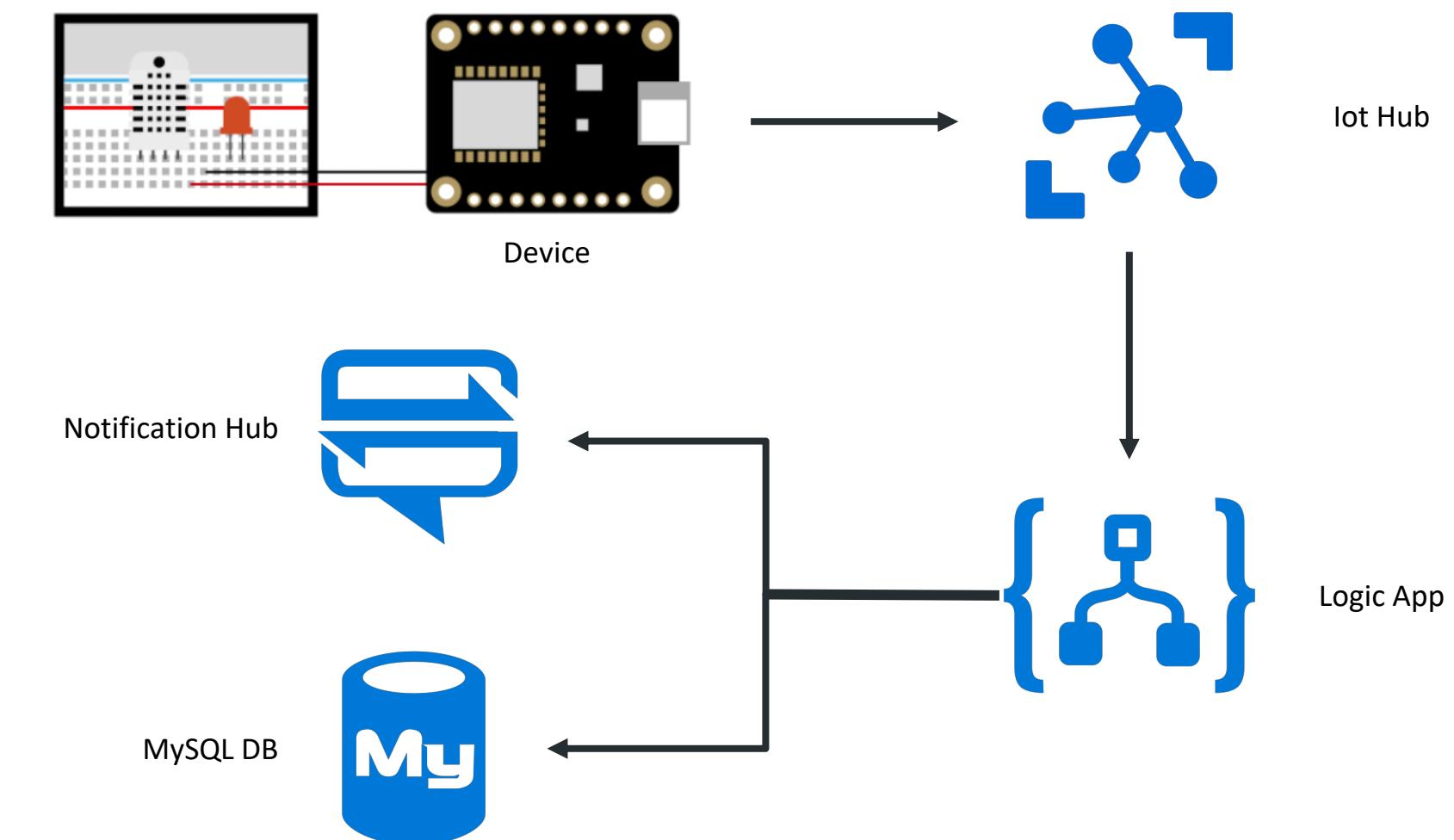
- SUPPORT ACROSS A WIDER VARIETY OF PLATFORMS AND SDKS EX. JAVASCRIPT AND JAVA SUPPORT, RTOS AND ARM PLATFORM SUPPORT
- DEVICE-FACING AND SERVICE-FACING SDKS FOR REGISTRATION AND MANAGEMENT
- IDENTITY AND ACCESS MANAGEMENT ACROSS ALL DEVICES CONNECTED TO HUB

IOT HUB

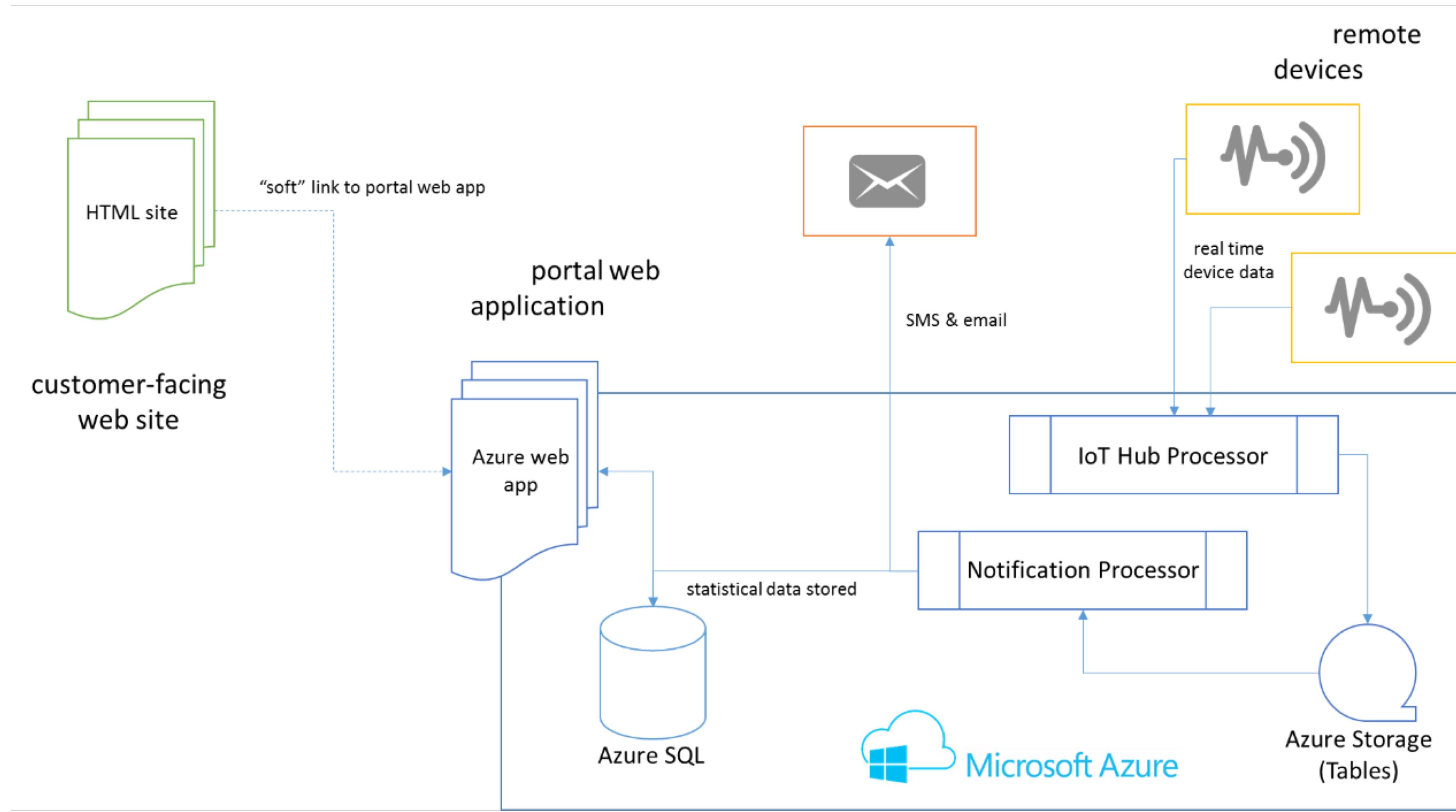


IOT HUB - SDK

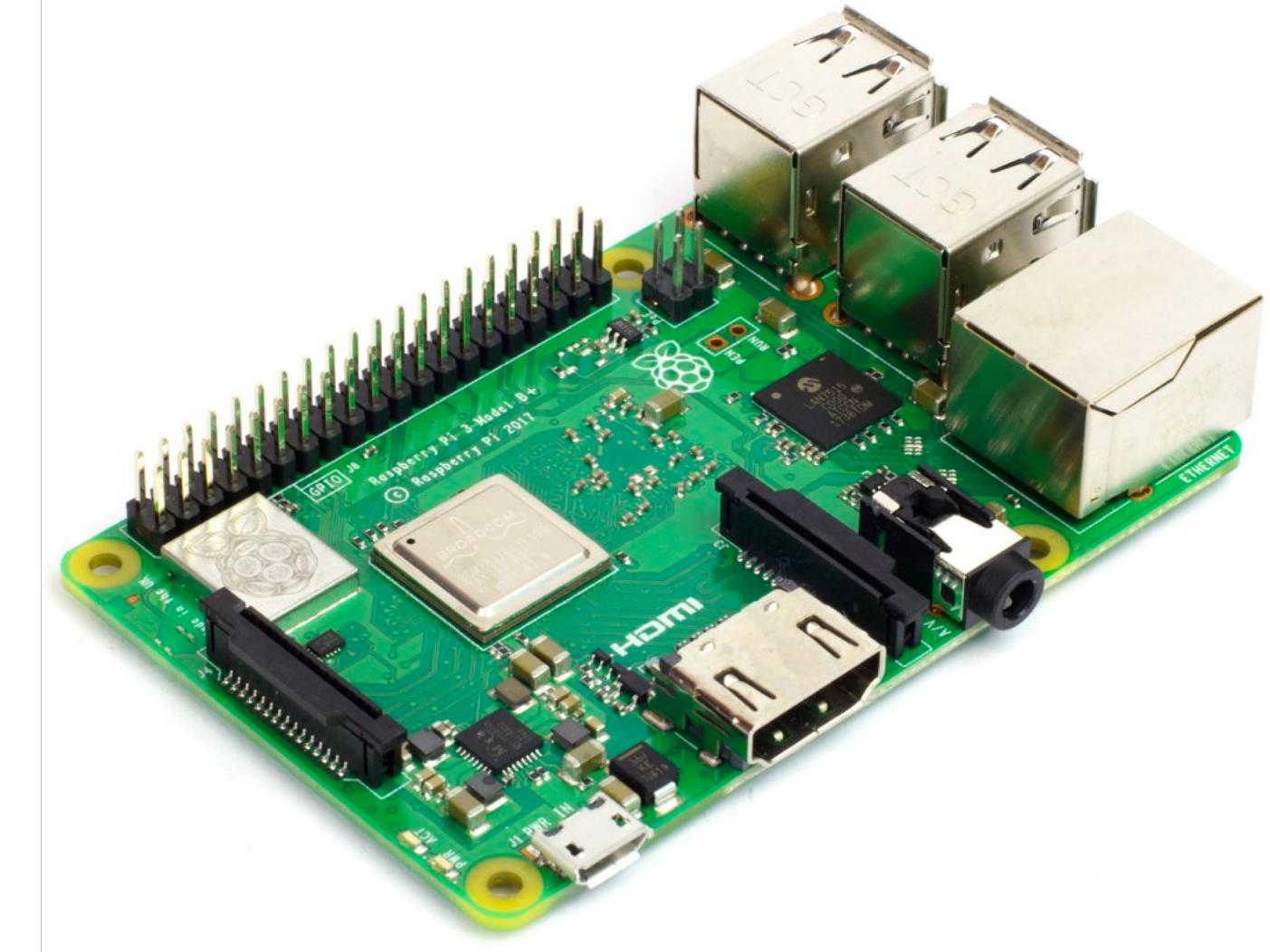
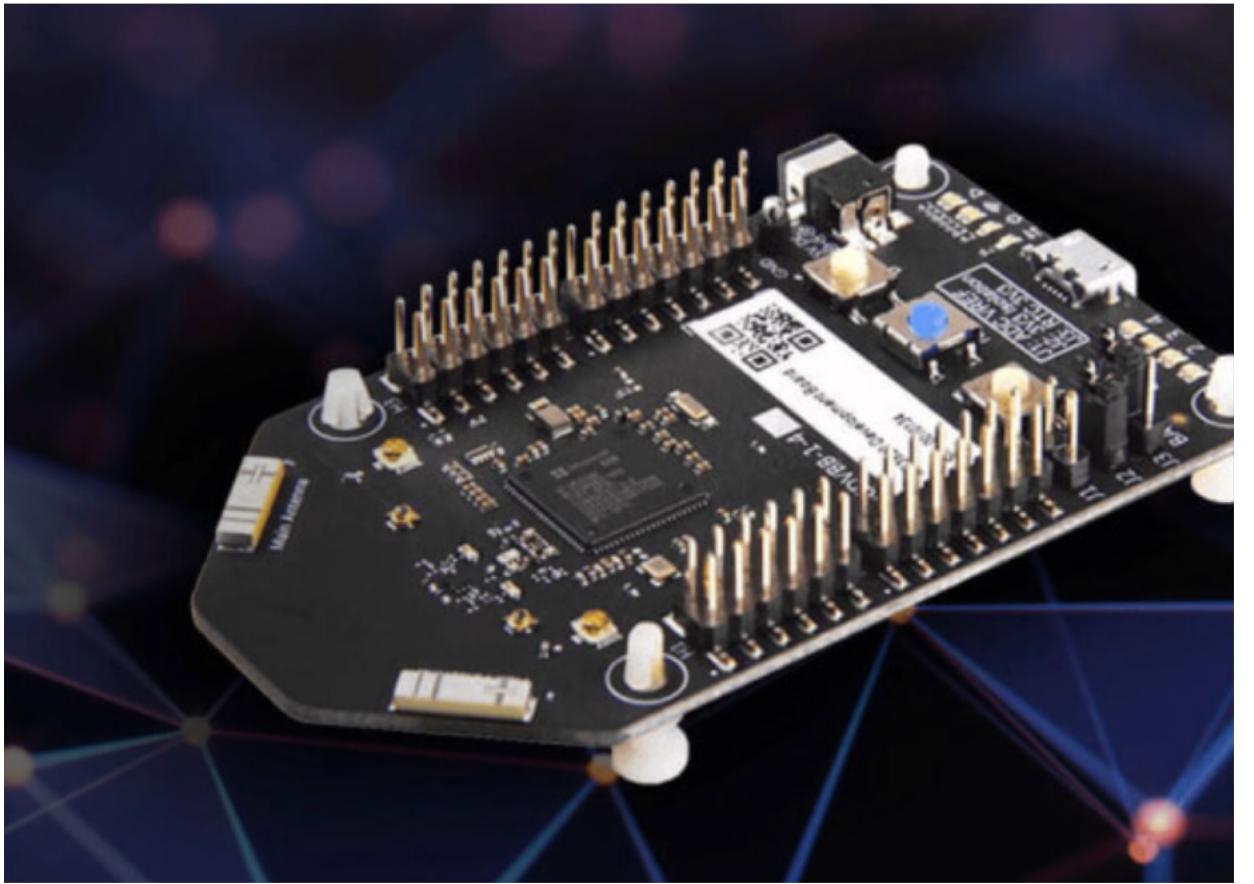
- ENABLE SIMPLE, SECURE DEVICE <-> CLOUD CONNECTIVITY & MANAGEMENT
- CLIENT “AGENT” SOFTWARE FOR DEVICES AND GATEWAYS
- LIBRARIES THAT OEMS/SIS/ISVS CAN USE IN NEW AND EXISTING SYSTEMS
- OPEN SOURCE SOFTWARE FRAMEWORK



IOT HUB - SAMPLE



IOT HUB - DEVICES



IOT - SUMMARY

- EVENT HUB
- IOT HUB