BIZTALK 360

PRESENTS YOU

IN ASSOCIATION WITH

Microsoft

INTEGRATE 2018

JUNE 4–6, 2018 etc.venues, London

Steef-Jan Wiggers

Azure & IoT Domain Lead

Serverless Messaging with Microsoft Azure

Platinum Sponsors

POWER OF INTEGRATION QUIBIQ codit

Gold Sponsors

affinus twoconnect Integrate.Everything.Now. REPLY SOLIDSOFT Bizbrains Integration accomplished. black marble

codit

Steef-Jan Wiggers

Azure & IoT Domain Lead

steefjan.wiggers@codit.eu

+31 653 12 29 57

@SteefJan

nl.linkedin.com/in/steefjan

Microsoft® MVP Most Valuable Professional

InfoQueue

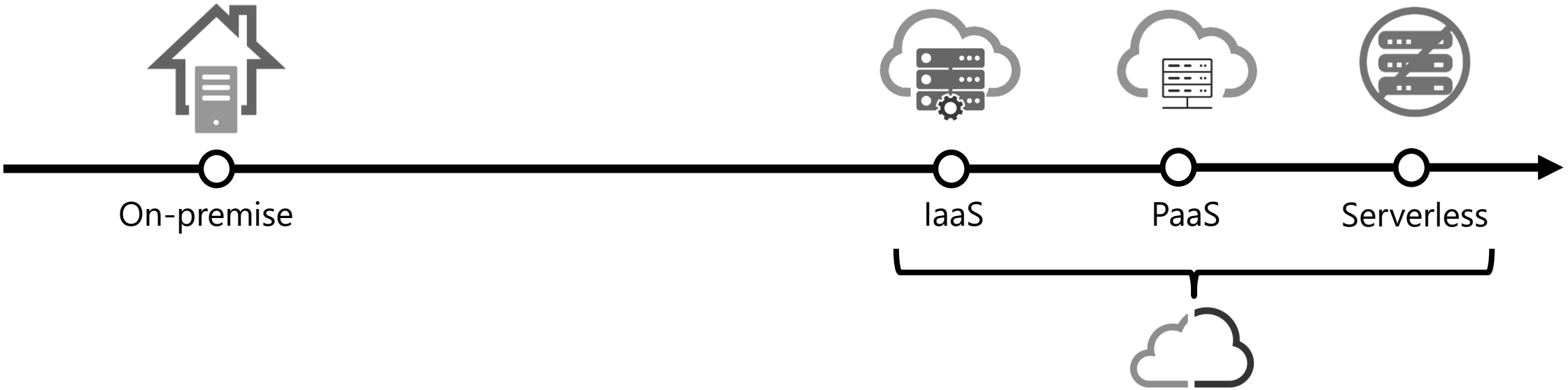
What can you expected in this session

- Serverless messaging
- Messaging In Azure
- Messaging scenarios
- Demo's

Serverless

Evolution to Serverless

- You can choose:
  - VM →
  - Containers →
  - Orchestrators →
  - PaaS →
  - Serverless



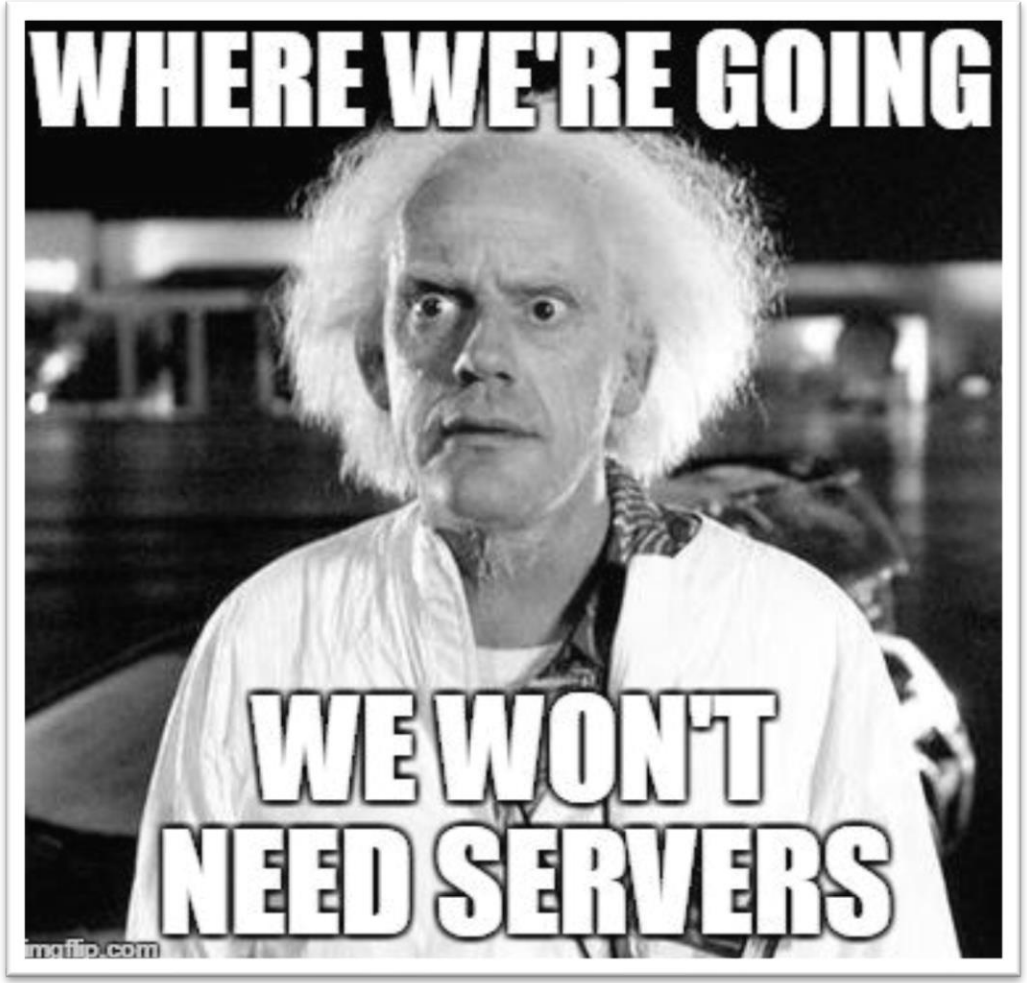
Defining Serverless

"Serverless computing is a ***cloud-computing execution model*** *in which the cloud provider* ***dynamically manages*** *the* ***allocation of machine resources****.* ***Pricing*** *is based on the* ***actual amount of resources consumed*** *by an application, rather than on pre-purchased units of capacity. It is a form of utility computing."*

Source (Wikipedia Serverless Computing):
https://en.wikipedia.org/wiki/Serverless_computing

Thoughts on Serverless

- Infinite scale requires atomic workloads and infinite resources.
- Reality is resources are always finite and many workloads not atomic.
- Messaging is for decoupling and transferring state or data secure and reliably.



Serverless in Azure

- Serverless Compute	→	Azure Functions
- Serverless Database	→	Azure Cosmos DB
- Serverless Events	→	Azure Event Grid
- Serverless Realtime	→	Azure Signal R Service
- Serverless Workflow	→	Logic Apps & Durable Functions
- Serverless IoT	→	IoT Hubs
- Serverless Analytics	→	Application Insights

Serverless Messaging

Is this serverless messaging?

Message: >*Down*

Message: >*Stay*

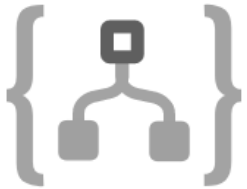
Message: >*Come*

What I mean is: You have options!

We have various messaging capabilities in Azure!

Notification Hubs

Mobile push notifications



Logic Apps

Workflow and LOB Integration

Service Bus & Azure Queues

Cloud messaging

Event Hubs

Telemetry stream ingestion

Event Grid

Event distribution

IoT Hub

IoT messaging and manage-ment

Relay

Discovery, Firewall/NAT Traversal

Azure Messaging Services

What do people do with messaging?

Financial Services	Utilities & Telecom	Logistics	Healthcare & Insurance	IoT & Telemetry

- Order processing
- Logging / telemetry
- Connected devices
- Notification / event driven systems

Messaging Services in Azure

- Azure Service Bus
- Event Hubs
- Event Grid
- Storage Queue

Building Blocks

Service Bus

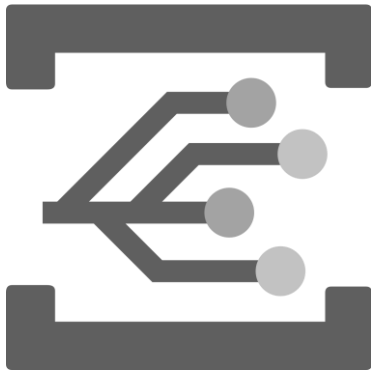
Enterprise messaging

Event Hubs

Big data streaming

Storage Queues

Simple task queues



Event Grid

Cross cloud reactive eventing

Enterprise Messaging

Business state transition

- Transfer of money and material
- Central arbiter of state / truth of transition or ownership
- Rich control of communication features
- You know a lot ahead of time
- You know the nouns:
  - Who
  - What
  - Where
- Competitive offering from Amazon: SQS, Amazon MQ, Google Pub/Sub

Eventing

Reacting dynamically to the world around you

- Cross App / Service / Cloud / Organization
- Light weight
- Low cost
- Few features, but important ones
- You probably don’t know the nouns and may not care

Cloud Pub/Sub Models

Push-Pull - Push-Push

Service Bus
Enterprise Messaging

- Queue Semantics
- Rich filters
- Instantaneous consistency
- Strict ordering
- JMS
- Geo-replication & Availability
- Rich broker features

Azure Event Grid
Reactive Programming

- Event raised by sources
- Independent individual messages
- Push mechanism
- Filtering and routing
- Pay as you go
- Fan out

Task Queue

Coordinate simple tasks across compute (workers)

- Low cost
- Pay as you Go
- Few features
- Easy to use
- Have as many queues as you like

Cloud Queues

Service Bus Queues - Storage Queues

Service Bus Queues

- Queue Semantics
- Rich filters
- Instantaneous consistency
- Strict ordering
- JMS
- Geo-replication & Availability
- Rich broker features

Storage Queues

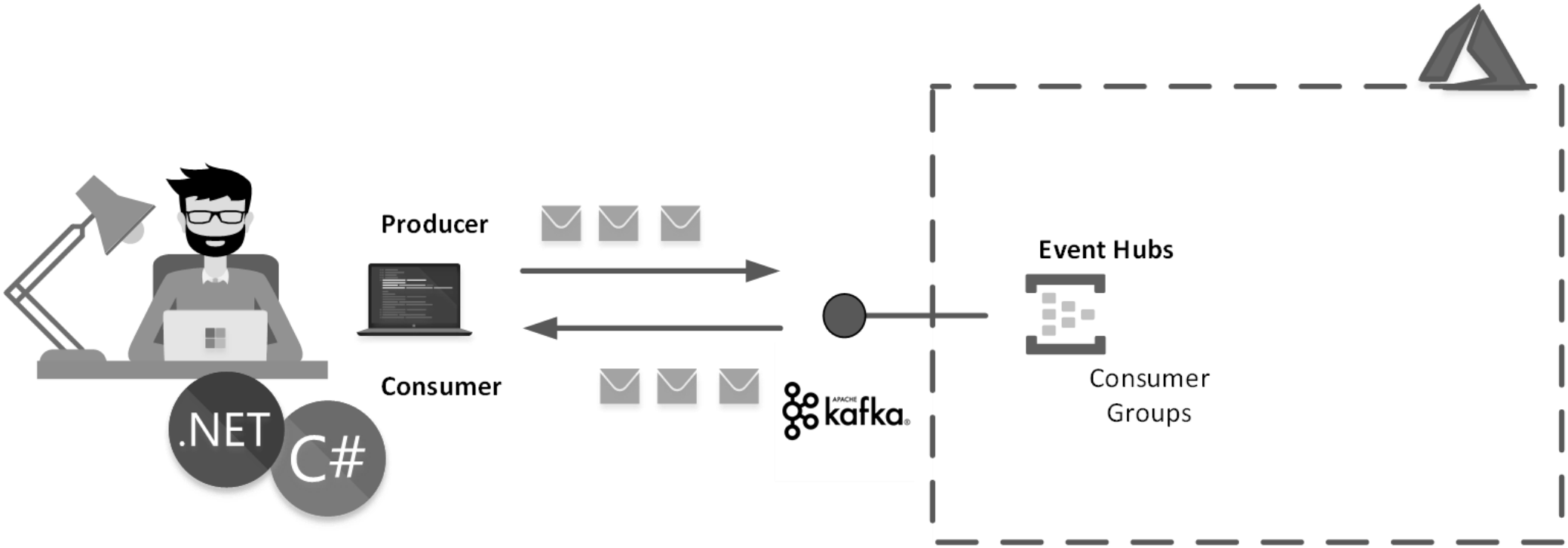
- Part of Storage infrastructure
- Simple REST-based interface
- Pull mechanism
- Reliable
- Persistence messaging
- Only queues

Big Data Streams

Flow data and telemetry in real-time

- Durable re-playable buffer
- Designed as a stream
- Events to an event hub via AMQP or HTTP
- Real-time and batch processing
- Feeding Big Data and Analytics
- Pre-purchase capacity in terms of throughput units
- Offers an Kafka Endpoint
- Competitive offering from Amazon Kinesis, Apache Kafka, and Google Pub/Sub

Kafka Endpoint - Demo



Brought to you by Wagner Silveira - https://goo.gl/LcjMQB

Produce - Consume

Consumer

```
ead 'Hi, welcome at Integrate - msg # 62 at 20180605_1137SS532' from: myeventhub [0] @1062
ead 'Hi, welcome at Integrate - msg # 63 at 20180605_1137SS532' from: myeventhub [0] @1063
ead 'Hi, welcome at Integrate - msg # 64 at 20180605_1137SS532' from: myeventhub [0] @1064
ead 'Hi, welcome at Integrate - msg # 65 at 20180605_1137SS532' from: myeventhub [0] @1065
ead 'Hi, welcome at Integrate - msg # 66 at 20180605_1137SS532' from: myeventhub [0] @1066
ead 'Hi, welcome at Integrate - msg # 67 at 20180605_1137SS532' from: myeventhub [0] @1067
ead 'Hi, welcome at Integrate - msg # 68 at 20180605_1137SS532' from: myeventhub [0] @1068
ead 'Hi, welcome at Integrate - msg # 69 at 20180605_1137SS532' from: myeventhub [0] @1069
ead 'Hi, welcome at Integrate - msg # 70 at 20180605_1137SS532' from: myeventhub [0] @1070
ead 'Hi, welcome at Integrate - msg # 71 at 20180605_1137SS532' from: myeventhub [0] @1071
ead 'Hi, welcome at Integrate - msg # 72 at 20180605_1137SS532' from: myeventhub [0] @1072
ead 'Hi, welcome at Integrate - msg # 73 at 20180605_1137SS532' from: myeventhub [0] @1073
ead 'Hi, welcome at Integrate - msg # 74 at 20180605_1137SS532' from: myeventhub [0] @1074
ead 'Hi, welcome at Integrate - msg # 75 at 20180605_1137SS532' from: myeventhub [0] @1075
ead 'Hi, welcome at Integrate - msg # 76 at 20180605_1137SS532' from: myeventhub [0] @1076
ead 'Hi, welcome at Integrate - msg # 77 at 20180605_1137SS532' from: myeventhub [0] @1077
ead 'Hi, welcome at Integrate - msg # 78 at 20180605_1137SS532' from: myeventhub [0] @1078
ead 'Hi, welcome at Integrate - msg # 79 at 20180605_1137SS532' from: myeventhub [0] @1079
ead 'Hi, welcome at Integrate - msg # 80 at 20180605_1137SS532' from: myeventhub [0] @1080
ead 'Hi, welcome at Integrate - msg # 81 at 20180605_1137SS532' from: myeventhub [0] @1081
ead 'Hi, welcome at Integrate - msg # 82 at 20180605_1137SS532' from: myeventhub [0] @1082
ead 'Hi, welcome at Integrate - msg # 83 at 20180605_1137SS532' from: myeventhub [0] @1083
ead 'Hi, welcome at Integrate - msg # 84 at 20180605_1137SS532' from: myeventhub [0] @1084
ead 'Hi, welcome at Integrate - msg # 85 at 20180605_1137SS532' from: myeventhub [0] @1085
ead 'Hi, welcome at Integrate - msg # 86 at 20180605_1137SS532' from: myeventhub [0] @1086
ead 'Hi, welcome at Integrate - msg # 87 at 20180605_1137SS532' from: myeventhub [0] @1087
```

Producer

```
7|2018-06-05 11:37:04.289|rdkafka#producer-1|SASL| [thrd:app]: Selected provider P
LAIN (builtin) for SASL mechanism PLAIN
7|2018-06-05 11:37:04.504|rdkafka#producer-1|SSL| [thrd:app]: Loading CA certifica
te(s) from file C:\Codit\Sample Projects\KafkaProducer\cacert.pem
7|2018-06-05 11:37:04.516|rdkafka#producer-1|BROKER| [thrd:app]: sasl_ssl://myeven
thubs360.servicebus.windows.net:9093/bootstrap: Added new broker with NodeId -1
7|2018-06-05 11:37:04.517|rdkafka#producer-1|BRKMAIN| [thrd::0/internal]: :0/inter
nal: Enter main broker thread
7|2018-06-05 11:37:04.517|rdkafka#producer-1|STATE| [thrd::0/internal]: :0/interna
l: Broker changed state INIT -> UP
7|2018-06-05 11:37:04.517|rdkafka#producer-1|BRKMAIN| [thrd:sasl_ssl://myeventhubs
360.servicebus.windows.net:9093/bootstrap]: sasl_ssl://myeventhubs360.servicebus.w
indows.net:9093/bootstrap: Enter main broker thread
7|2018-06-05 11:37:04.517|rdkafka#producer-1|CONNECT| [thrd:sasl_ssl://myeventhubs
360.servicebus.windows.net:9093/bootstrap]: sasl_ssl://myeventhubs360.servicebus.w
indows.net:9093/bootstrap: broker in state INIT connecting
7|2018-06-05 11:37:04.525|rdkafka#producer-1|CONNECT| [thrd:sasl_ssl://myeventhubs
360.servicebus.windows.net:9093/bootstrap]: sasl_ssl://myeventhubs360.servicebus.w
indows.net:9093/bootstrap: Connecting to ipv4#40.112.242.0:9093 (sasl_ssl) with so
cket 1124
7|2018-06-05 11:37:04.526|rdkafka#producer-1|STATE| [thrd:sasl_ssl://myeventhubs36
0.servicebus.windows.net:9093/bootstrap]: sasl_ssl://myeventhubs360.servicebus.win
```

.NET Code

```
var config = new Dictionary<string, object>
{
    {"bootstrap.servers", brokerList},
    {"api.version.request", "true" },
    {"security.protocol", "SASL_SSL"},
    {"sasl.mechanism", "PLAIN"},
    {"sasl.username", "$ConnectionString"},
    {"sasl.password", eventHubConnectionString},
    {"ssl.ca.location", caCertLocation },
    {"debug", "security,broker,protocol" }
};


using (var producer = new Producer<Null, string>(config, null, new StringSerializer(Encoding.UTF8)))
{
    Console.WriteLine($"{producer.Name} producing on {topicName}.");

    Console.WriteLine("Initiating Execution");
    for (var x = 0; x < 100; x++)
    {
        var msg = $"Hi, Welcome at Integrate - msg # {x} at {DateTime.Now:yyyMMdd_HHmmSSfff}";

        var deliveryReport = producer.ProduceAsync(topicName, null, msg);
        deliveryReport.ContinueWith(task =>
        {
            Console.WriteLine($"Partition: {task.Result.Partition}, Offset: {task.Result.Offset}");
        });
    }
```

Messaging considerations

- Protocol
- Format
- Size
- Security
- Frequency
- Reliability
- Monitoring
- Networking

Cloud Services

VMSS

VMs

Express Route

SQL DB

Cosmos DB

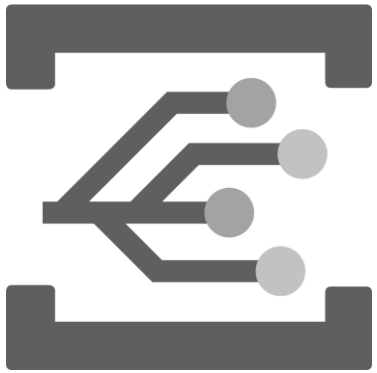
Logic Apps

ASA

SQL DW

Power BI

Data Lake



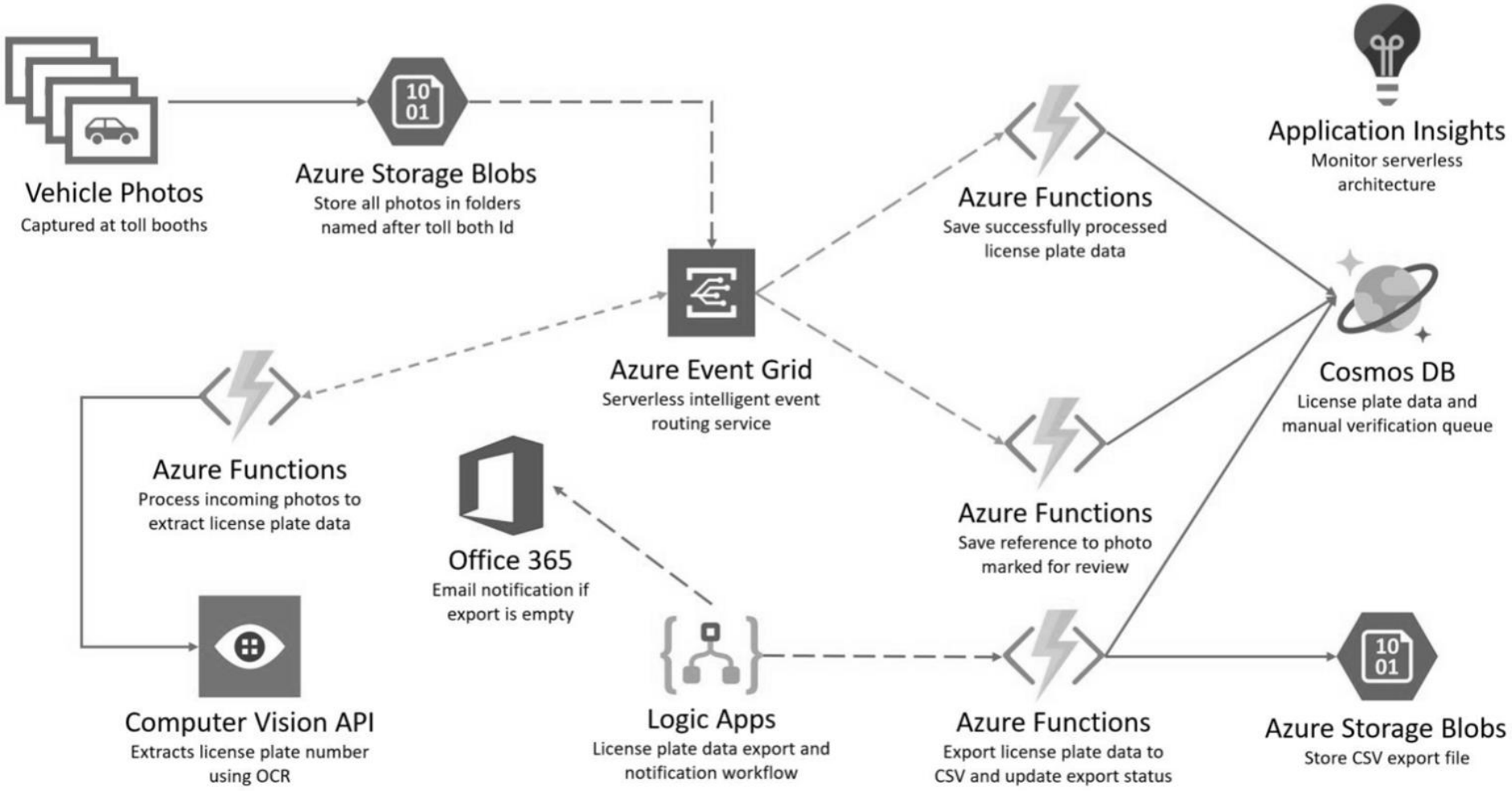
Functions

AKS

ACS

Messaging scenario's

Tollbooth License Plate Recognition

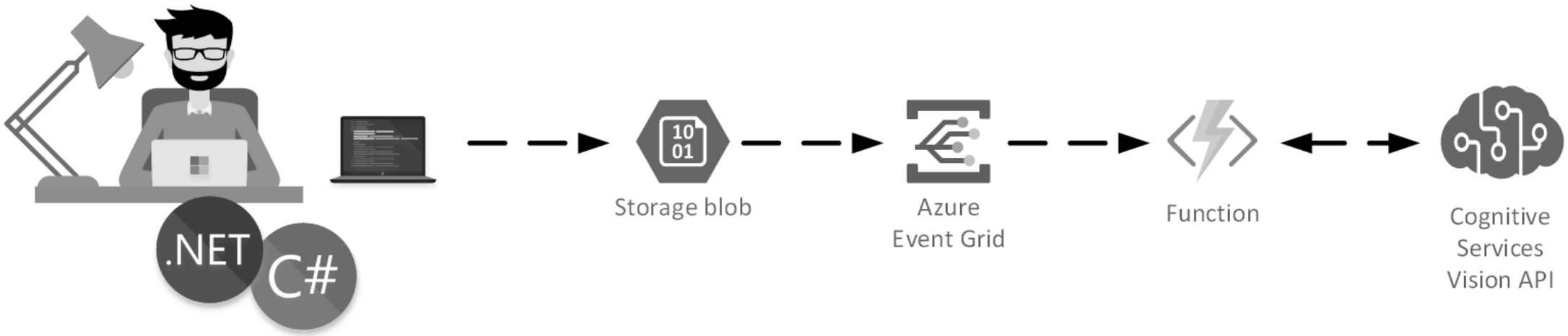


Smart solution

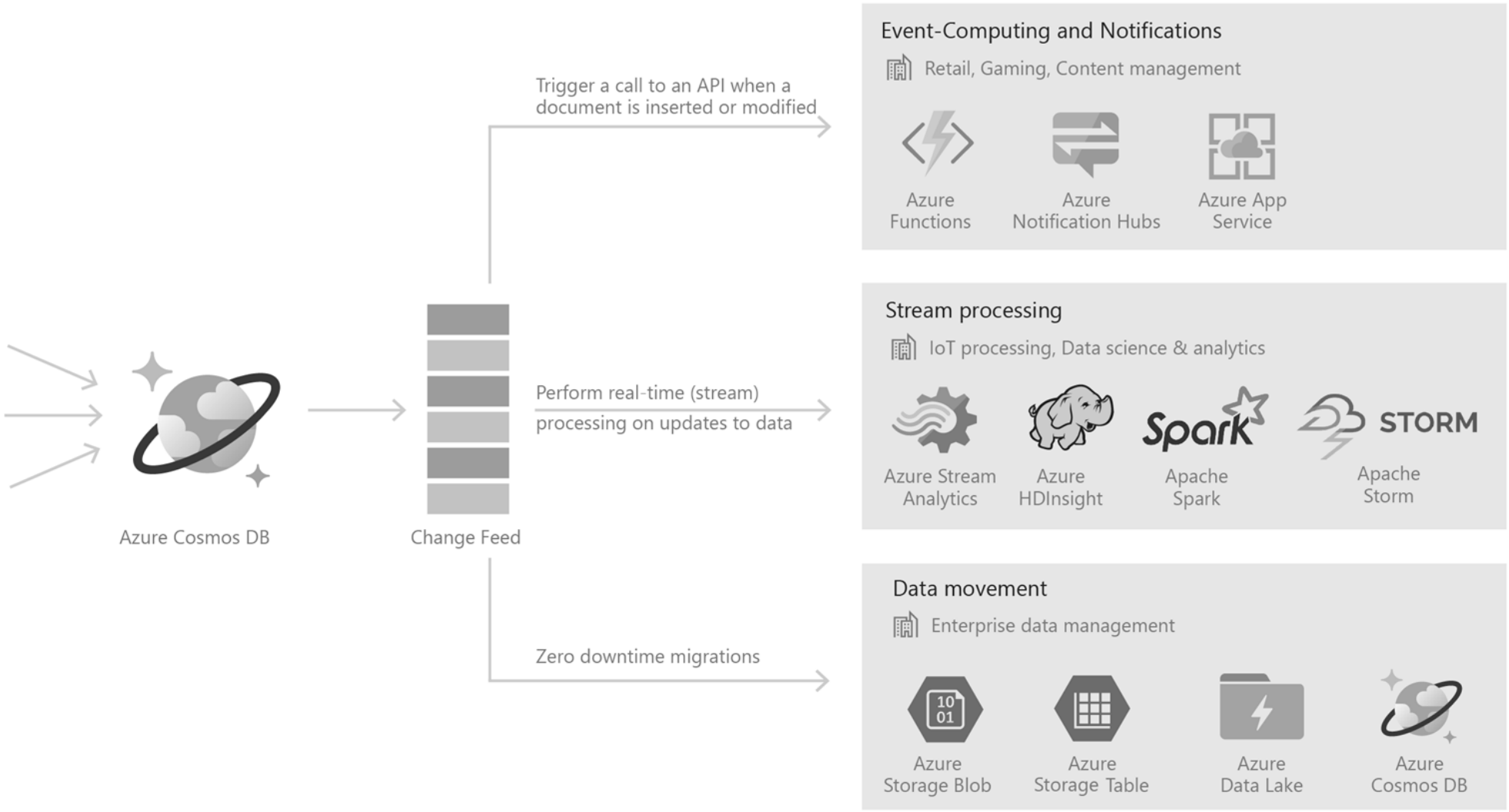
License plate OCR

- Use the Recognition Services Computer Vision API and its built-in OCR capabilities
- The image processing function can make a REST call to send the photo and read the JSON data in return

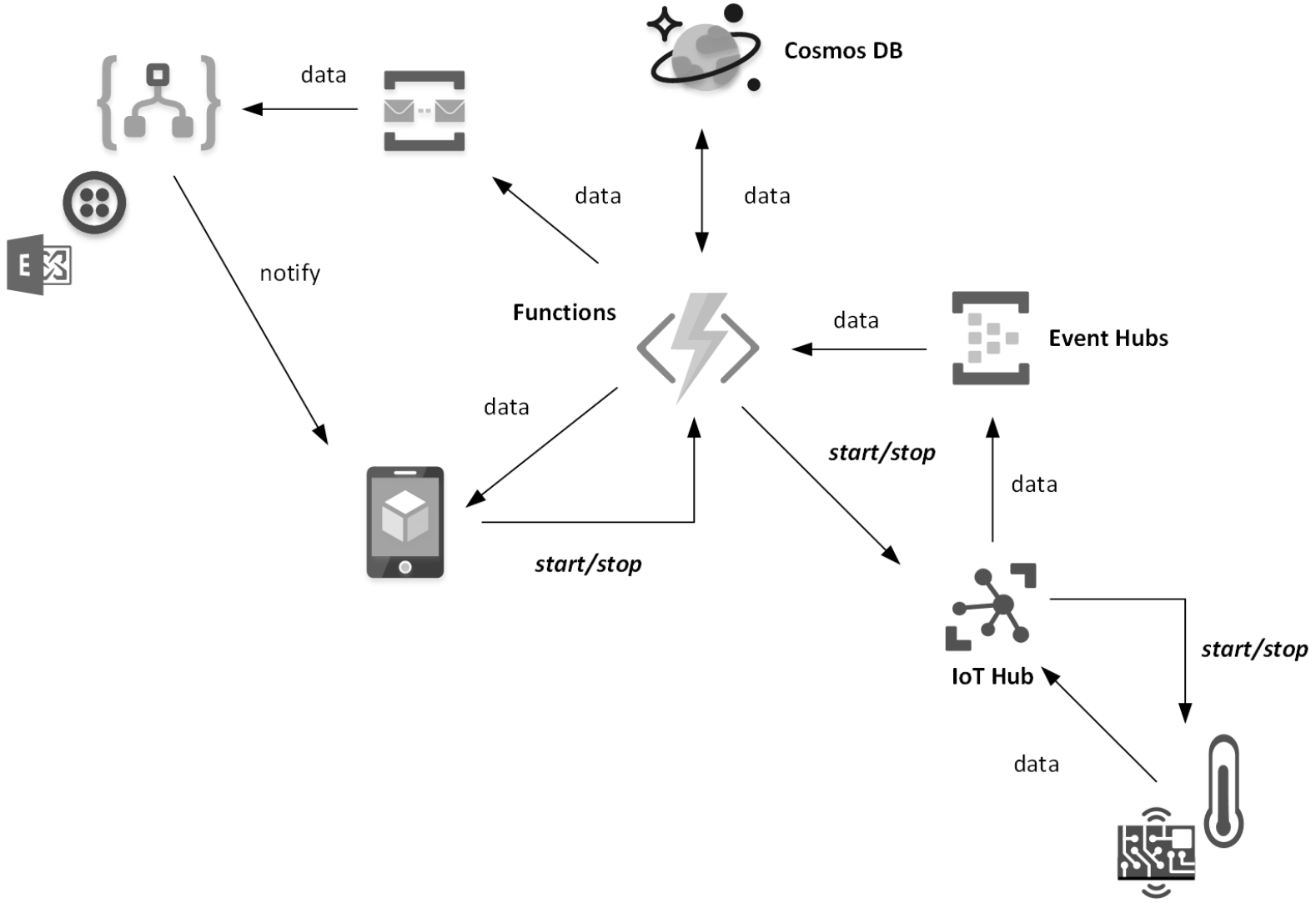
Smart Services - Demo



Change feed

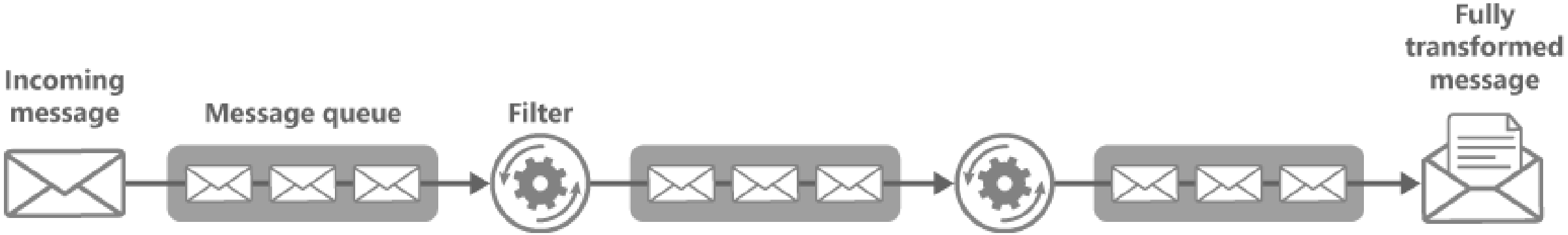

Azure Cosmos DB
Change Feed
Trigger a call to an API when a document is inserted or modified
Perform real-time (stream) processing on updates to data
Zero downtime migrations
Event-Computing and Notifications
Retail, Gaming, Content management
Azure Functions
Azure Notification Hubs
Azure App Service
Stream processing
IoT processing, Data science & analytics
Azure Stream Analytics
Azure HDInsight
Apache Spark
Apache Storm
Data movement
Enterprise data management
Azure Storage Blob
Azure Storage Table
Azure Data Lake
Azure Cosmos DB

Serverless Home Automation - Demo

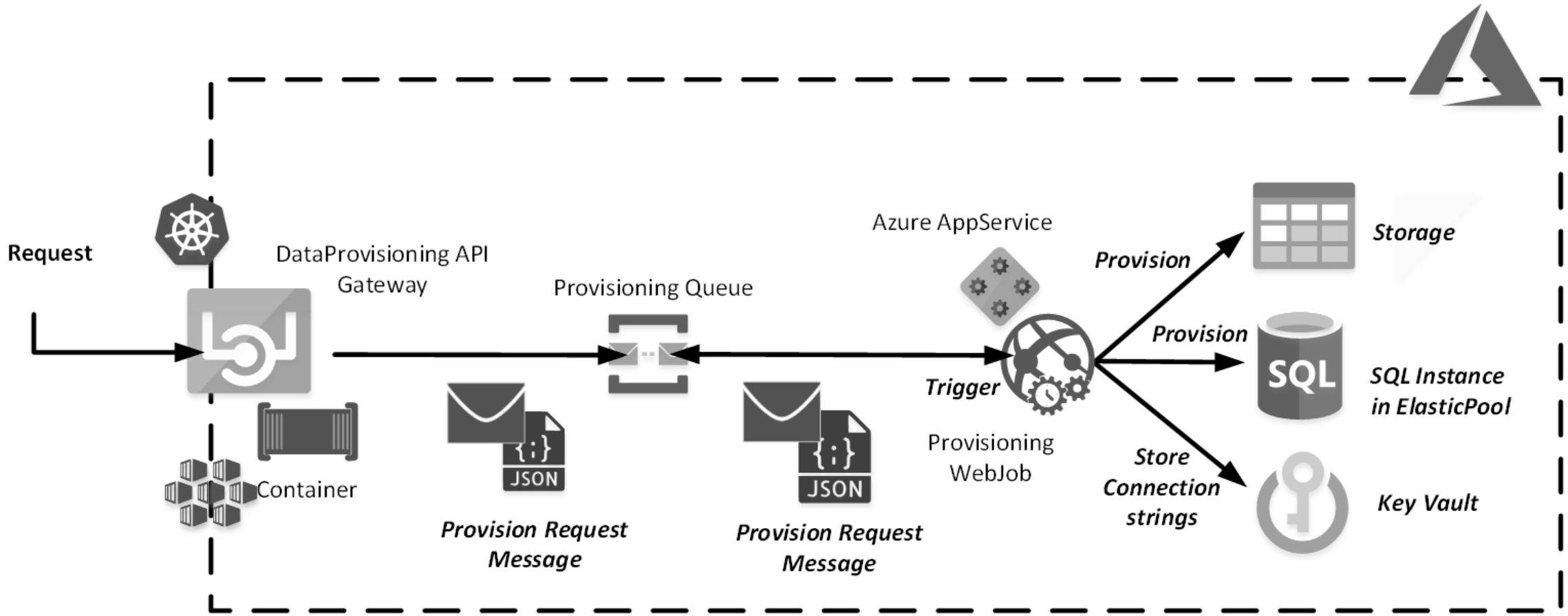


Pipes and filters

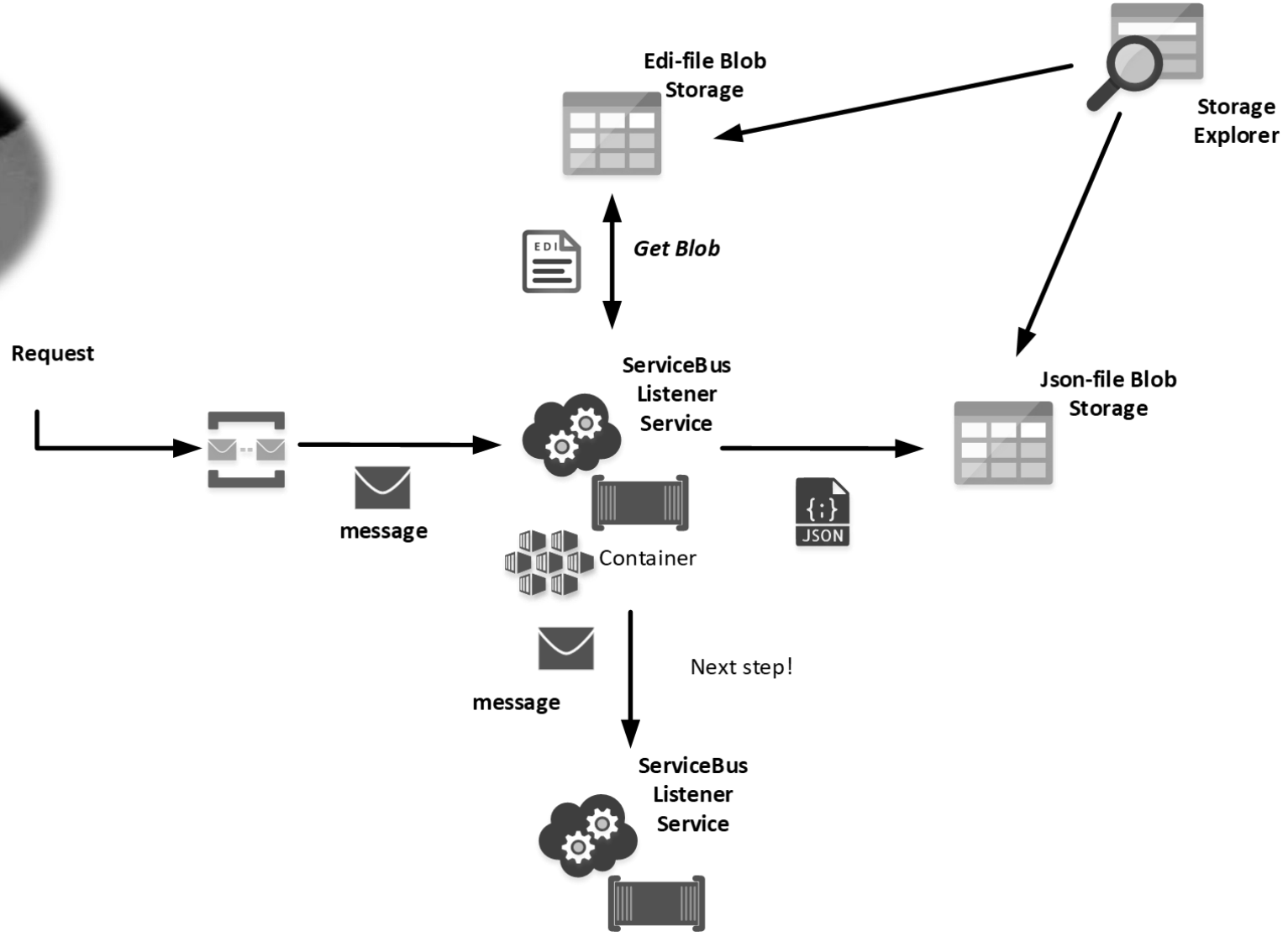
The processing required by an application can easily be broken down into a set of independent steps.



Microservices containers



Microservice processing - Demo



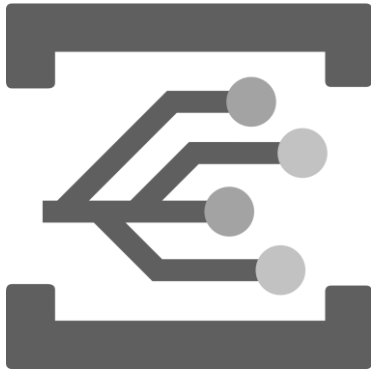
K8S – Microservice Log

Logs from demoservicebuslistener in demoservicebuslistener-127130596-c3j82

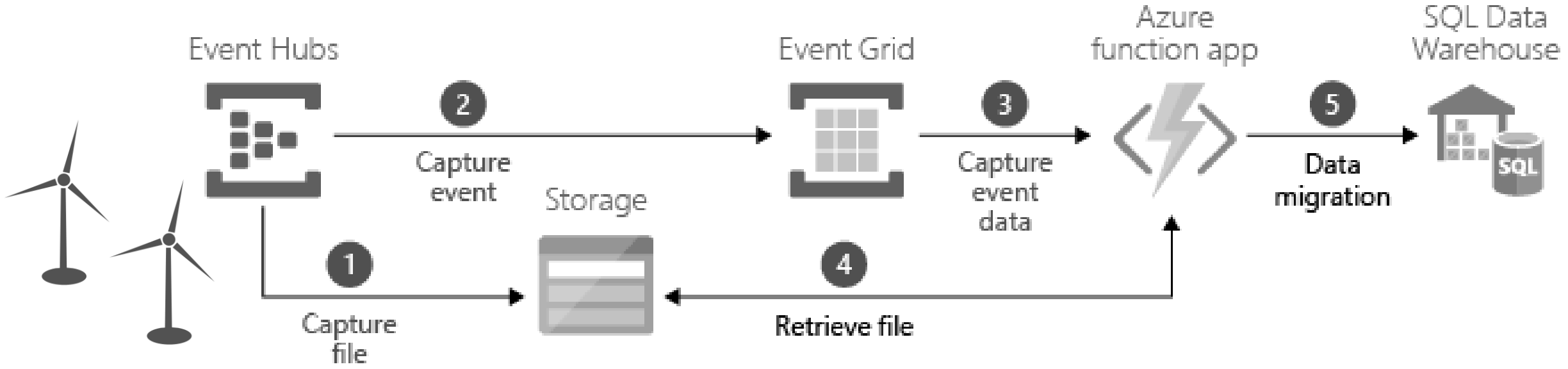
```
Message 84158cedeb4f4fcbb3d9bfdc8dcbdbd7 dequeued: {
 "MprId" : "10",
 "FileName" : "TestEDI.VEE",
 "Name": "TestEDI.json"
}
Message 84158cedeb4f4fcbb3d9bfdc8dcbdbd7 successfully processed
Message ef6ee3bbd7a34aabbbe62d58ac602b08 dequeued: {
 "MprId" : "89",
 "FileName" : "TestEDI.VEE",
 "Name": "TestEDI.json"
}
Message ef6ee3bbd7a34aabbbe62d58ac602b08 successfully processed
Message 8aaef190-d0aa-44d4-b2ef-420101ab95ce dequeued: {
 "MprId" : "13",
 "FileName" : "TestEDI.VEE",
 "Name": "TestEDI.json"
}
Message 8aaef190-d0aa-44d4-b2ef-420101ab95ce successfully processed
Message cfeeb8dd639c4ee58482e028766f4eb9 dequeued: {
 "MprId" : "79",
 "FileName" : "TestEDI.VEE",
 "Name": "TestEDI.json"
}
Message cfeeb8dd639c4ee58482e028766f4eb9 successfully processed
```

Logs from 6/1/18 8:18 AM to 6/5/18 9:18 AM

Use Azure Messaging Services together



Data and event pipeline - Demo



Summary

- Choose the right messaging service(s)
- Workloads
- Security and compliance
- Cross-platform
- Costs
- DevOps and Management

Happy Messaging!

Thank you!

Keep in touch.

Call or mail us. Ask us. Happy to help.