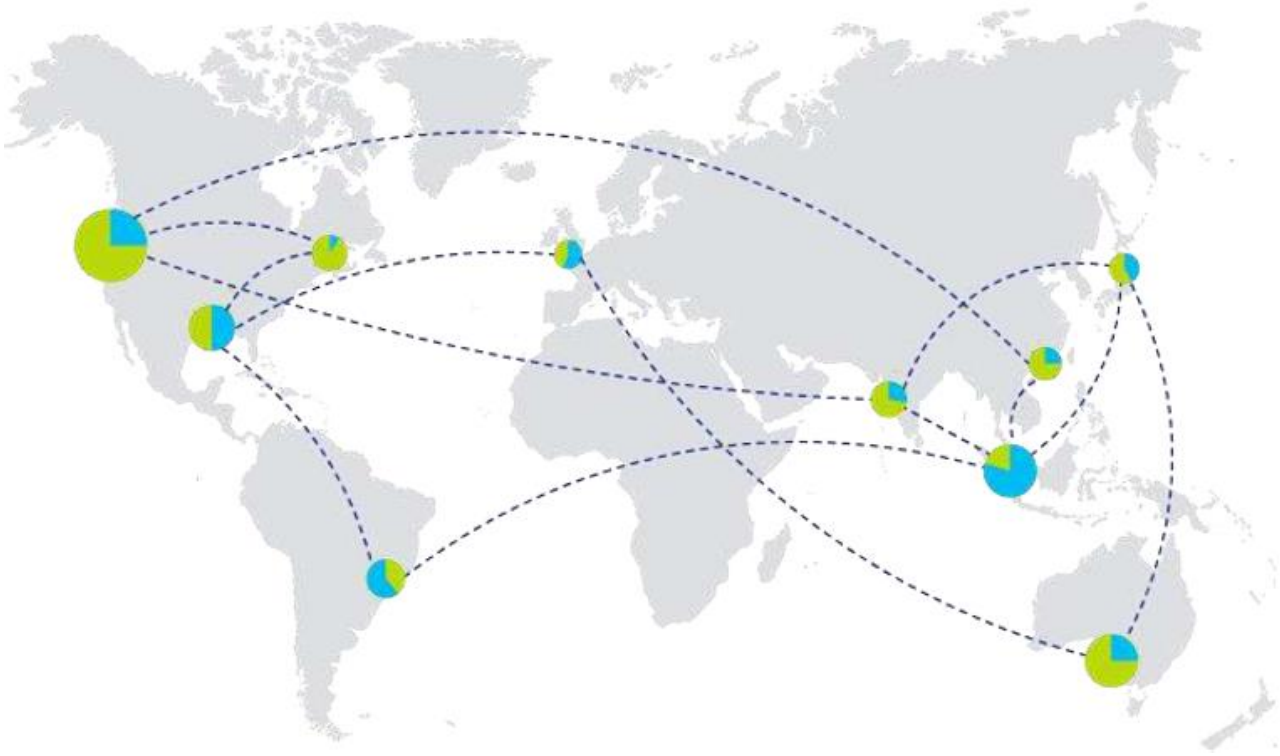
# Build Serverless App with Cosmos DB and Azure Functions

Mohammad Asif Waquar
@asifwaquar



# about me

*Asif Waquar*

https://www.linkedin.com/in/asifwaquar/

@asifwaquar

asifwaquar

asifwaquar

https://asifwaquar.com/

Microsoft® MVP Most Valuable Professional

NUS National University of Singapore | ISS INSTITUTE OF SYSTEMS SCIENCE

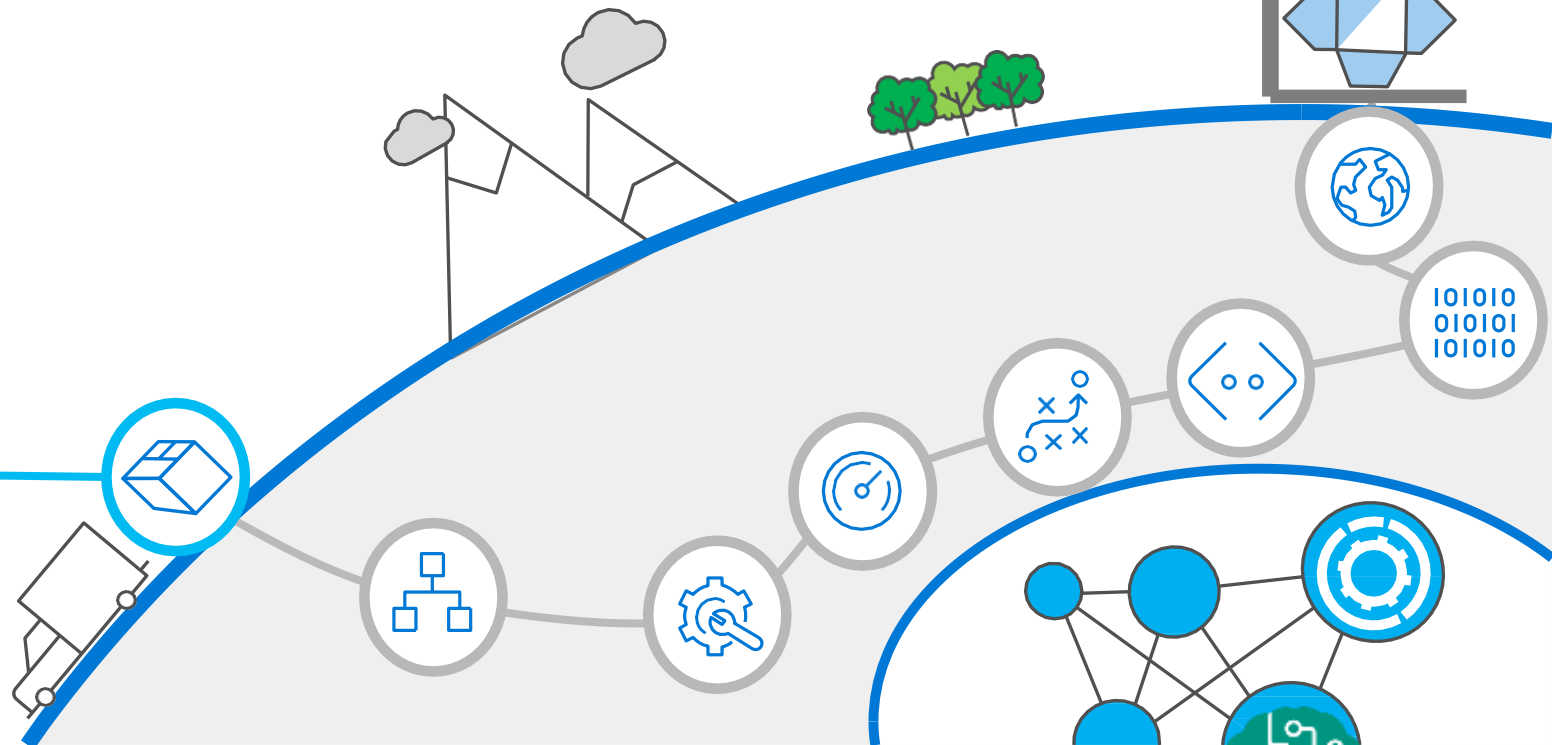
corda Certified r3

HYPERLEDGER FABRIC

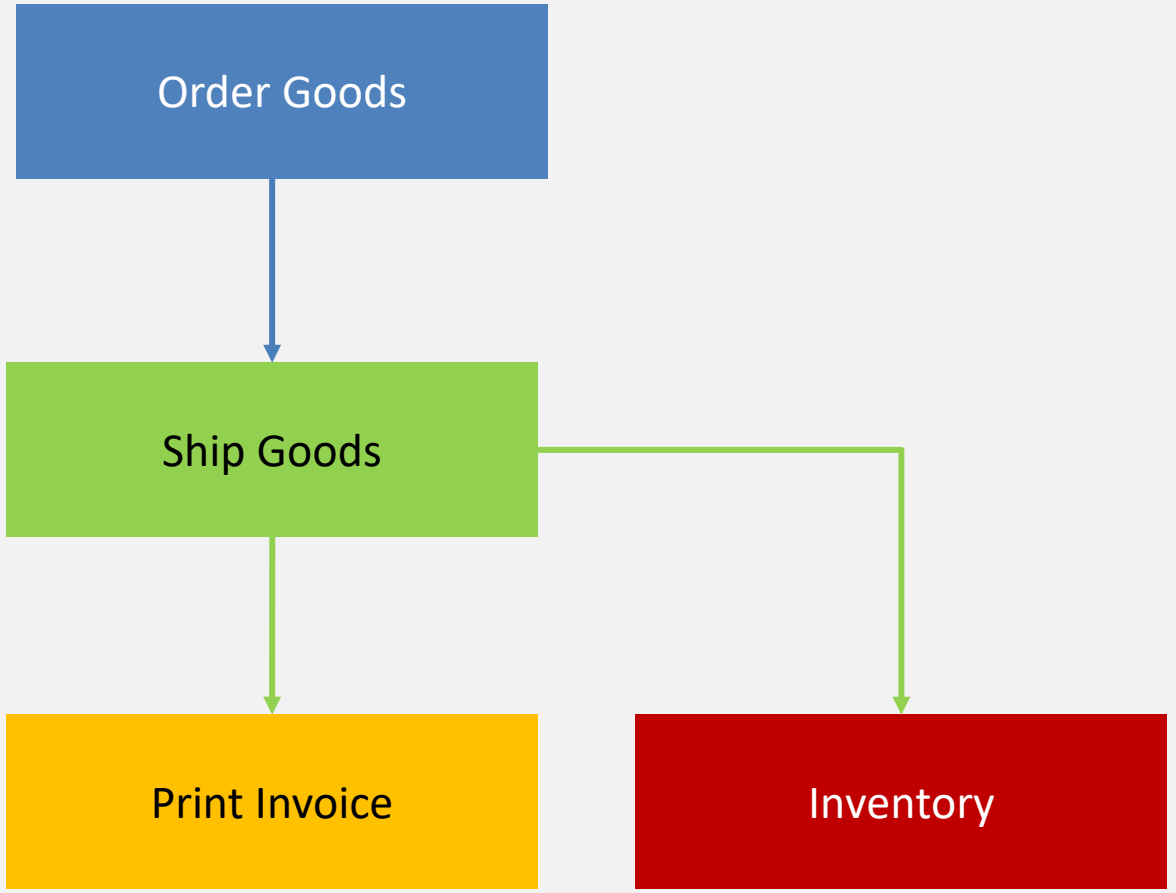
Microsoft CERTIFICATION EXAM DP-200

# Modern Apps ?

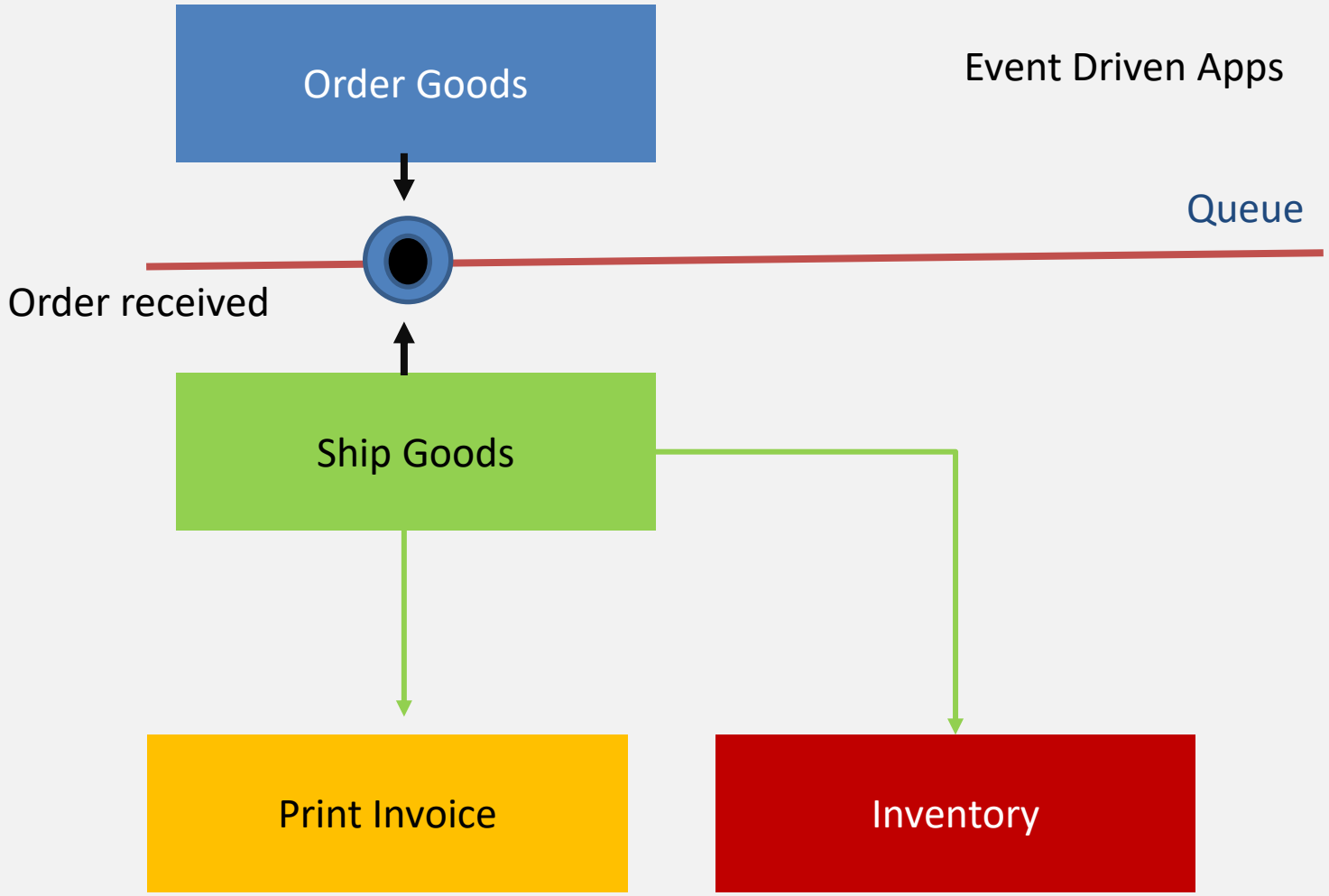
- ✓ Volume
- ✓ Variety
- ✓ Velocity



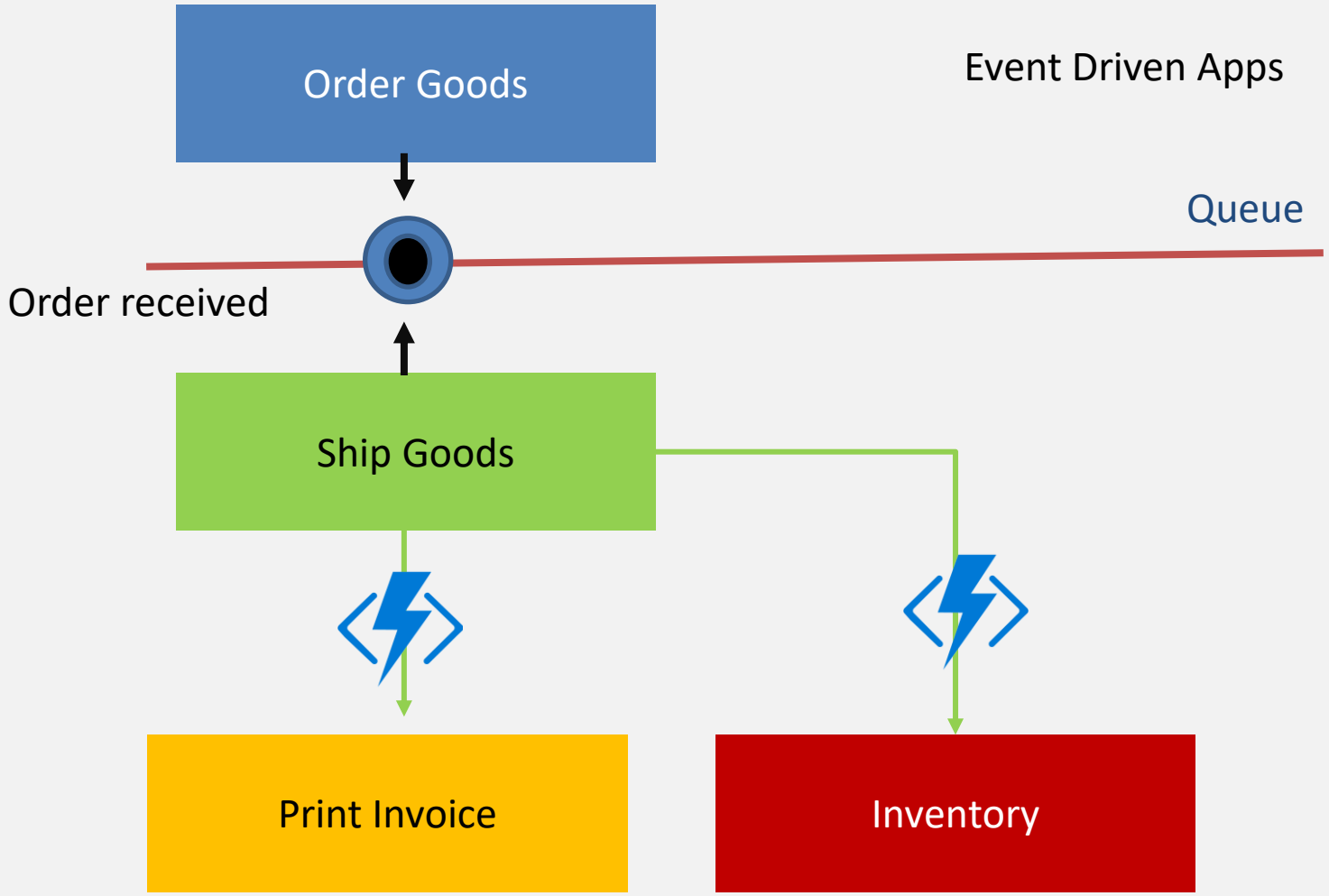
- ✓ Tight Coupling
- ✓ Integration with Other System
- ✓ CRUD Application Read/Writes Performance
- ✓ Update Conflicts
- ✓ Scaling Vertically



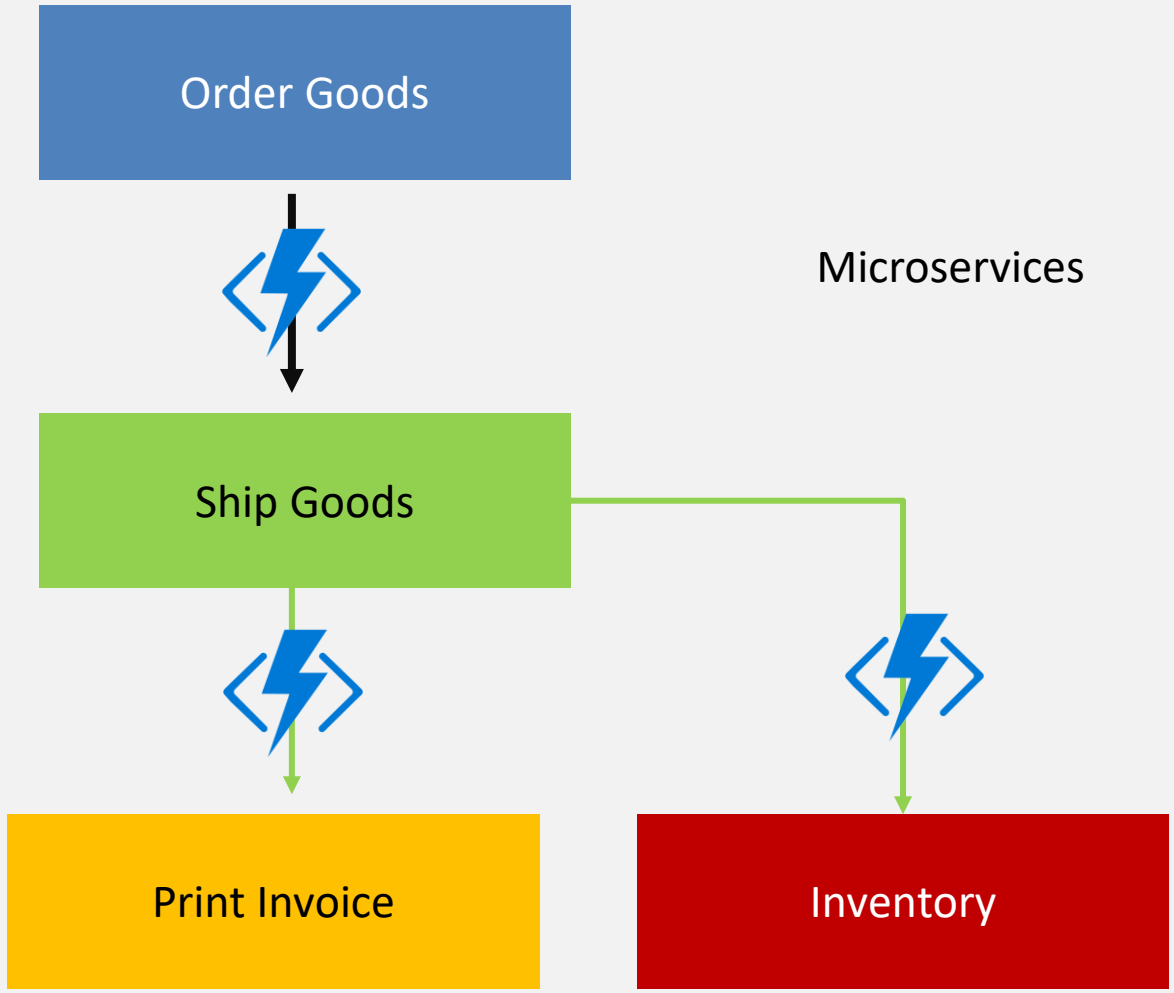
- ✓ Decoupling
- ✓ Auditing
- ✓ Debugging
- ✓ Integration



- ✓ Decoupling
- ✓ Auditing
- ✓ Debugging
- ✓ Integration



- ✓ Decoupling
- ✓ Fault Tolerance
- ✓ Stability
- ✓ Auditing
- ✓ Easy Update
- ✓ Versioning



# Azure Cosmos DB

A globally distributed, massively scalable, multi-model database service

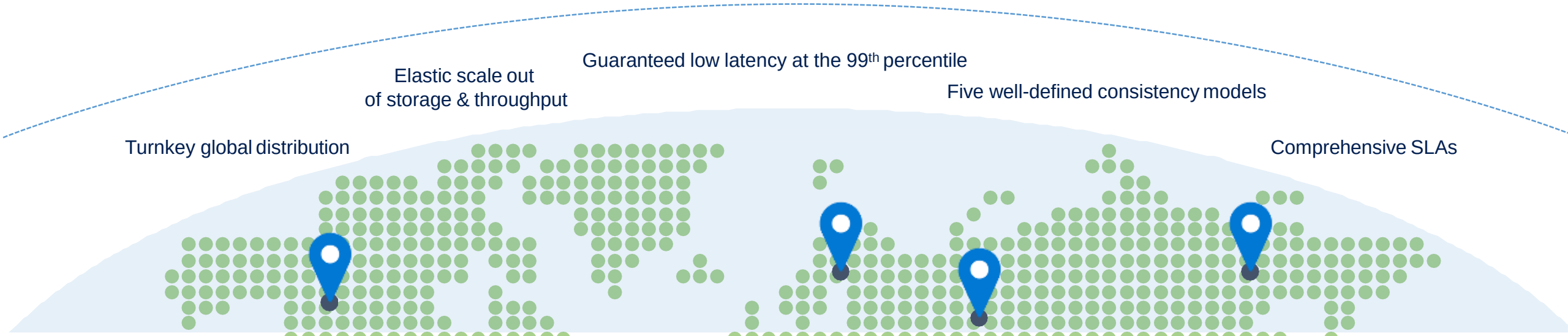
Elastic scale out
of storage & throughput

Guaranteed low latency at the 99th percentile

Five well-defined consistency models

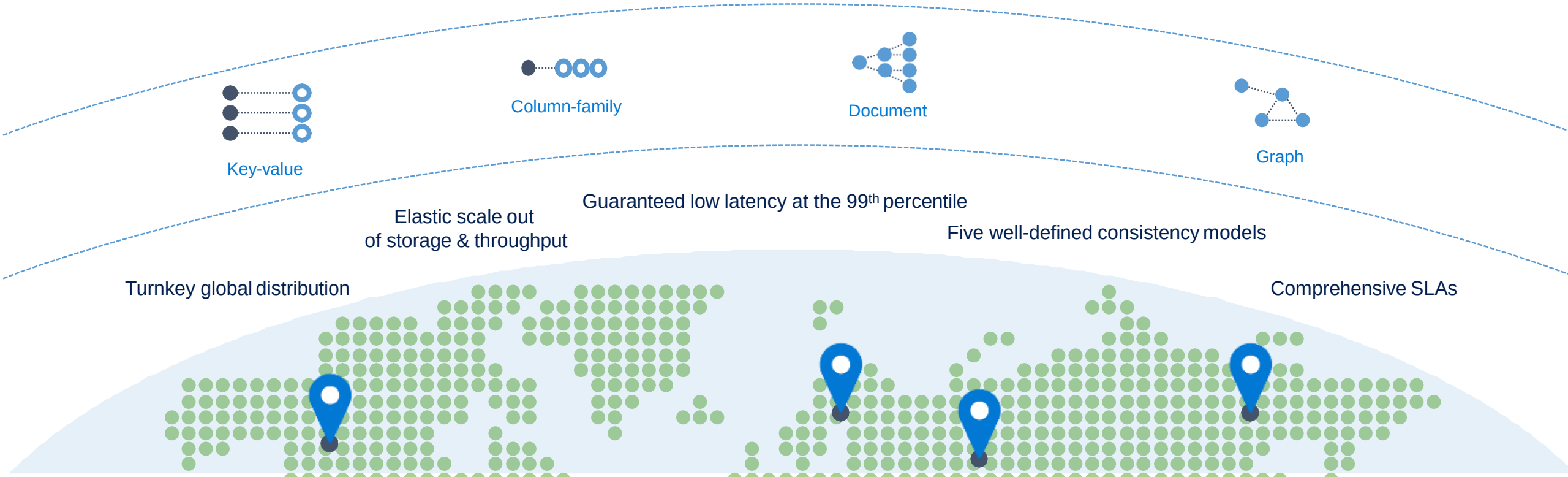
Turnkey global distribution

Comprehensive SLAs



# Azure Cosmos DB

A globally distributed, massively scalable, multi-model database service



# Azure Cosmos DB

A globally distributed, massively scalable, multi-model database service

Table API

cassandra

SQL

Cosmos DB's API for MongoDB

Gremlin

$G = (V, E)$

Key-value

Column-family

Document

Graph

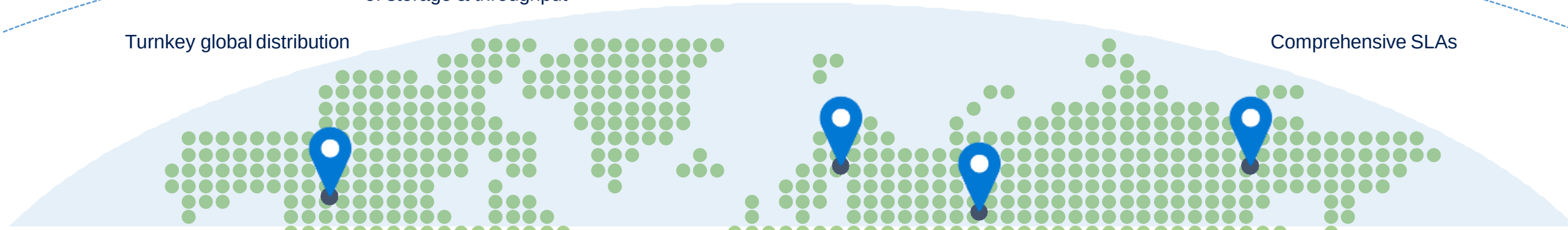
Elastic scale out of storage & throughput

Guaranteed low latency at the 99th percentile

Five well-defined consistency models

Turnkey global distribution

Comprehensive SLAs



# Features

- Multi-model data paradigm: key-value, document, graph, family of columns;
- Low latency for 99% of queries: less than 10 ms for read operations and less than 15 ms for (indexed) write operations;
- Designed for high throughput;
- Ensures availability, consistency of data, delay at SLA level of 99.999%;
- Configurable throughput;
- Automatic replication (master-slave);
- Automatic data indexing;
- Configurable levels of consistency of data. Five different levels (Strong, Bounded Staleness, Session, Consistent Prefix, Eventual);
- Run no-ETL analytics over the near-real time operational data stored in Azure Cosmos DB with Azure Synapse Analytics.

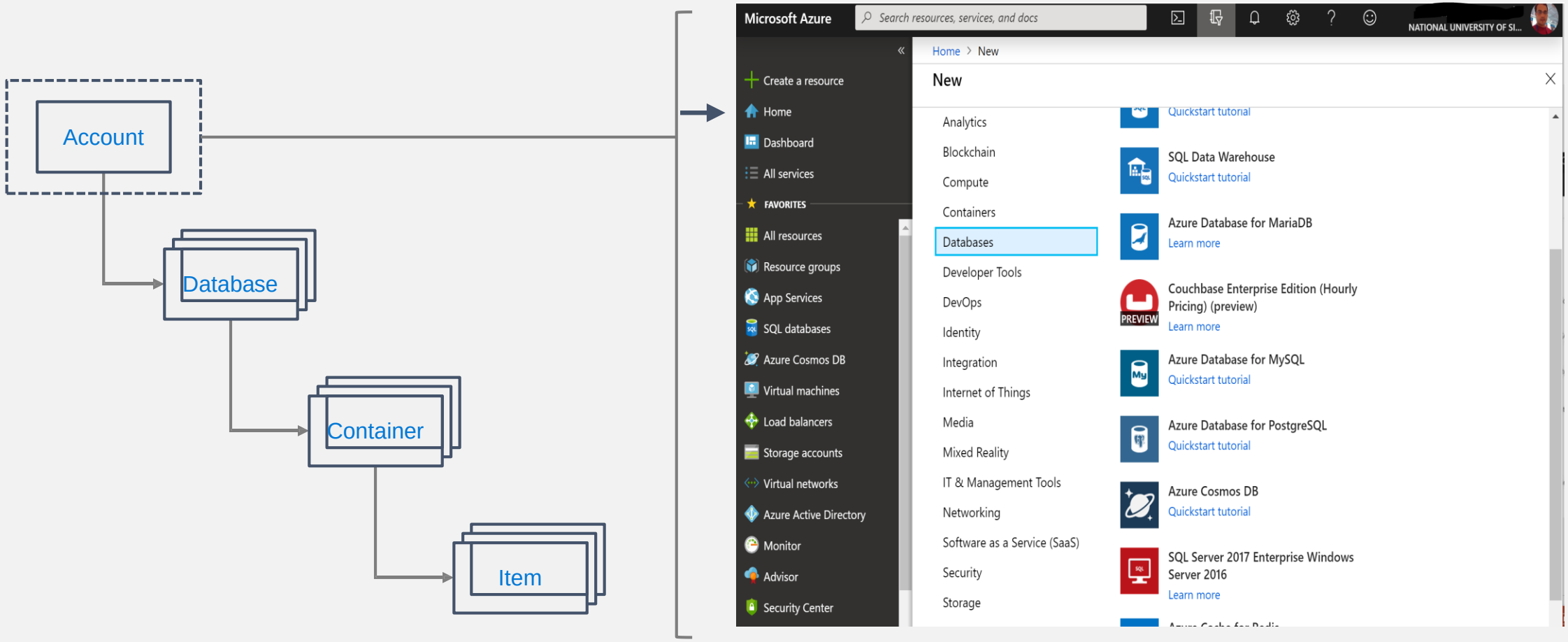
# How to start ?

- Get your Azure 30 day free trial
  http://azure.microsoft.com
- Free try of 30 days of Cosmos DB (no subscription or credit-card required)
  http://azure.microsoft.com/en-us/try/cosmosdb/
- For local development use the Local Emulator (no internet needed).
  http://aka.ms/cosmosdb-emulator

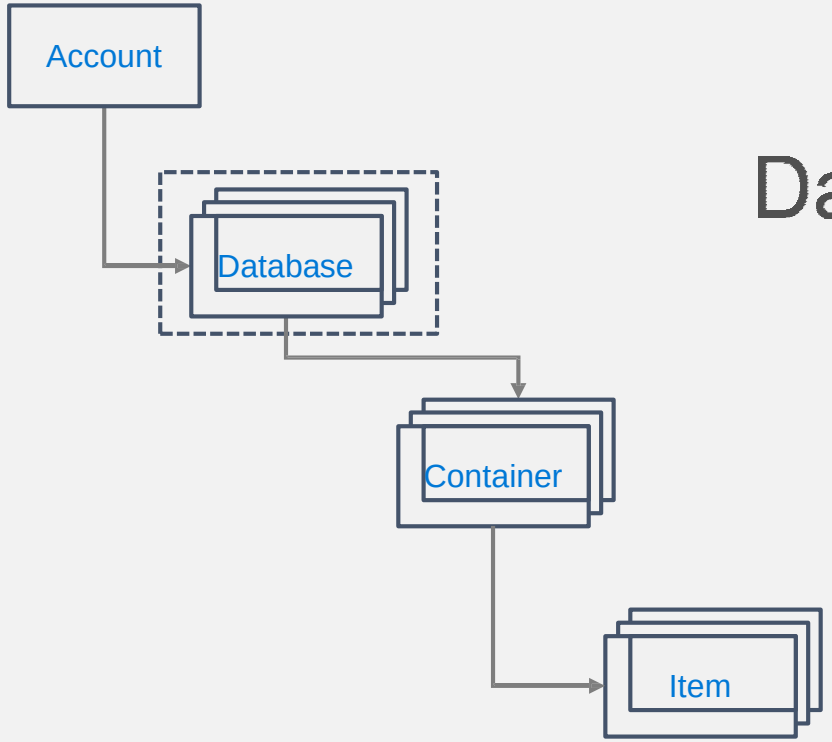
Take advantage of the Always free tier 5 GB and 400 RU/s (Request Units).

# Resource Model

# Creating Account

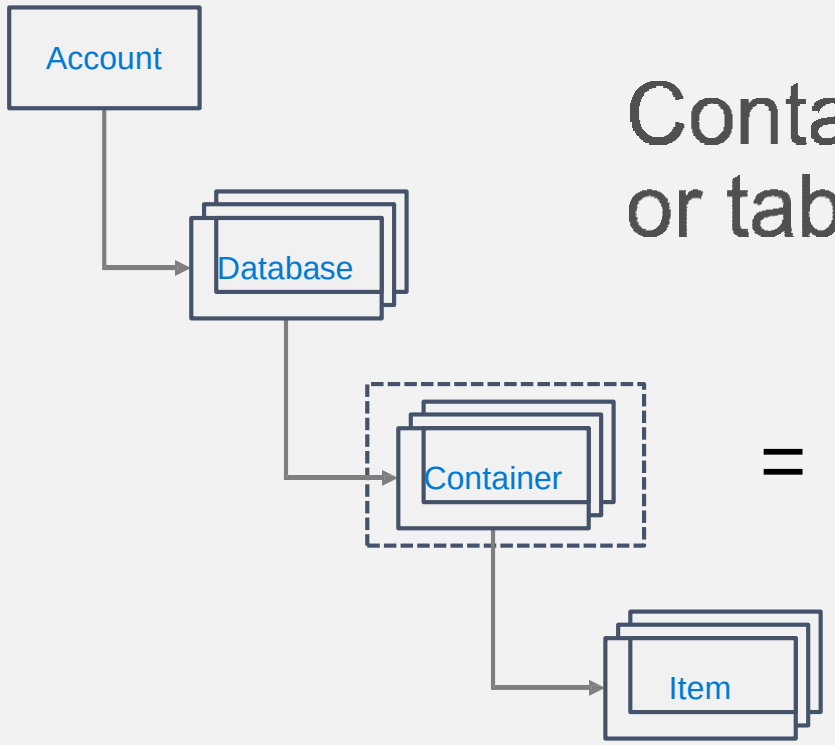


# Database Representations




Database has 1 or more Containers

# Container Representations




Container – can be a collection, graph or table

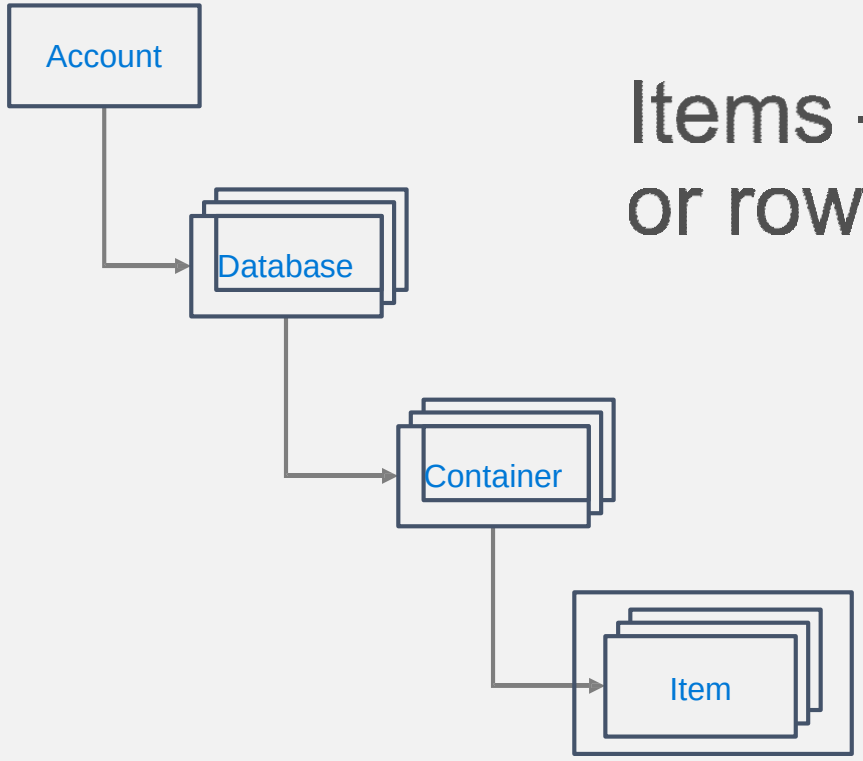
=

Collection

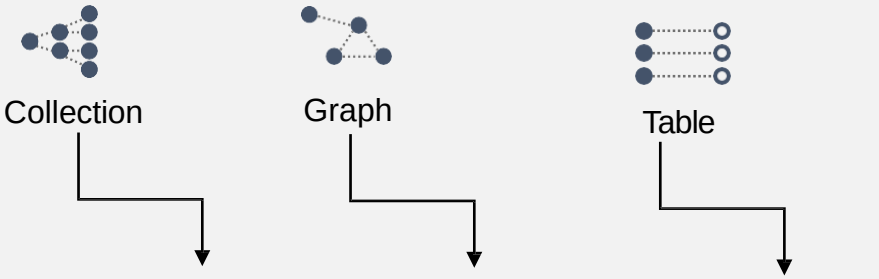
Graph

Table

# Item Representations




Items – can be document, vertices/edges, or row



# Container-Level Resources


Associate stored procedures, triggers, UDF with container

# Data Modelling: Relational vs. Document

| Relational Store | Document Store |
| --- | --- |
| Rows | Documents |
| Columns | Properties |
| Strongly-typed schemas | Schema-free |
| Highly normalized | Typically denormalized |

User Table

| UserID | Name | Dob |
| --- | --- | --- |
| 1 | John Smith | 8/30/1964 |

Holdings Table

| StockID | UserID | Qty | Symbol |
| --- | --- | --- | --- |
| 1 | 1 | 100 | MSFT |
| 2 | 1 | 75 | WMT |

Document

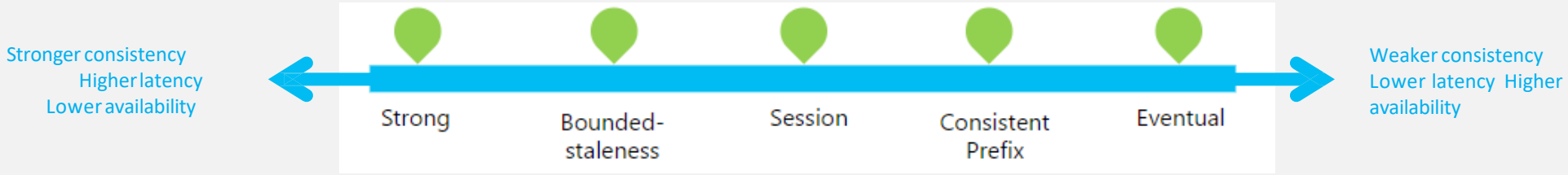
```
{
  "id": 1,
  "name": "John Smith",
  "dob": "1964-30-08",
  "holdings": [
     { "qty": 100, "symbol": "MSFT" },
     { "qty": 75, "symbol": "WMT" }
   ]
}
```

# Replication ,Consistency and Partitioning

# Replication and Consistency

How do you ensure consistent reads across replicas?

- Define a consistency level




Replication within a region

- Data moves extremely fast (typically, within 1ms) between neighboring racks

Global replication

- It takes hundreds of milliseconds to move data across continents

# Partitioning

Logical partition: **Stores all data associated with the same partition key value**

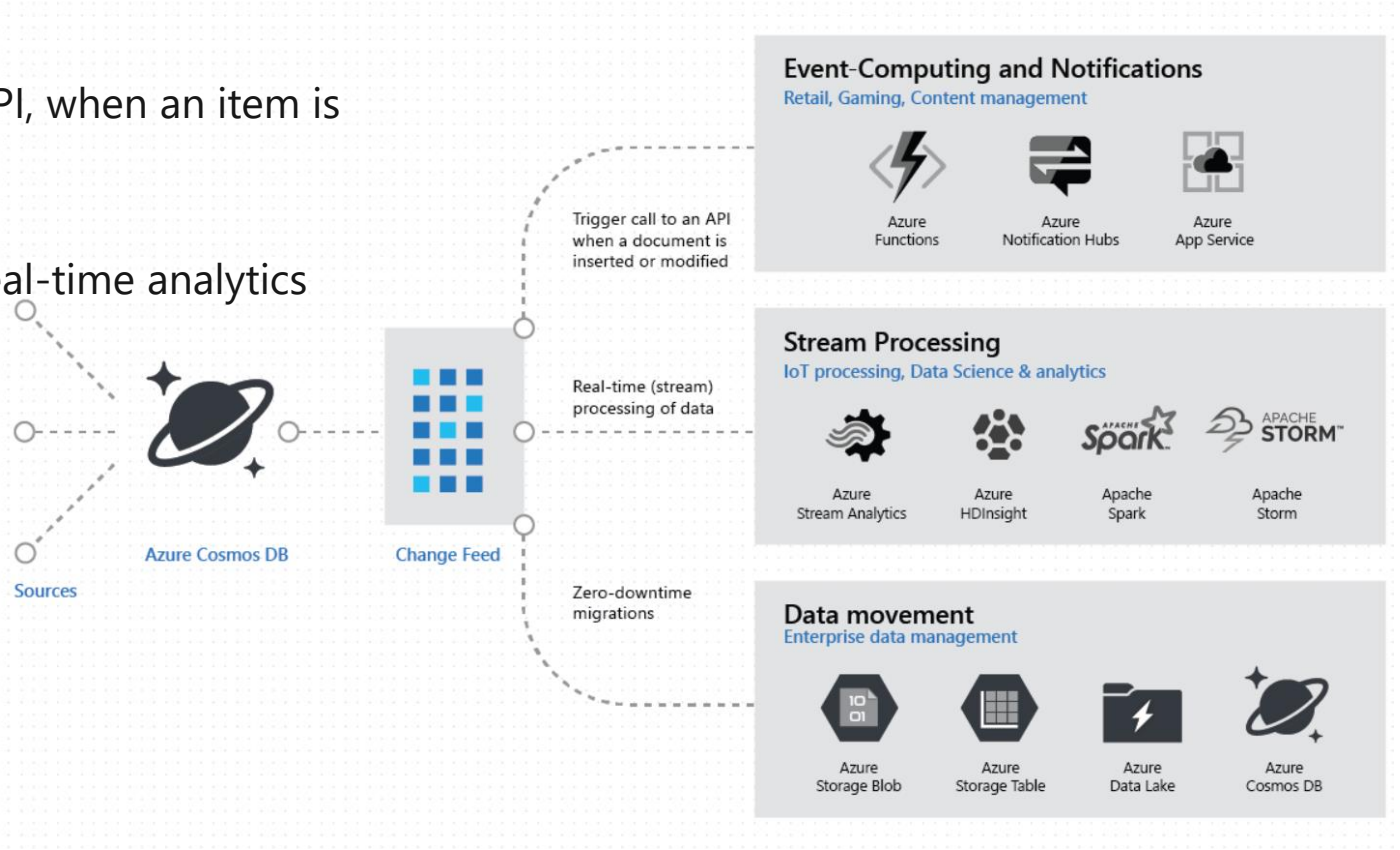
Physical partition: **Fixed amount of reserved SSD-backed storage + compute.**

**Cosmos DB distributes logical partitions among a smaller number of physical partitions.**

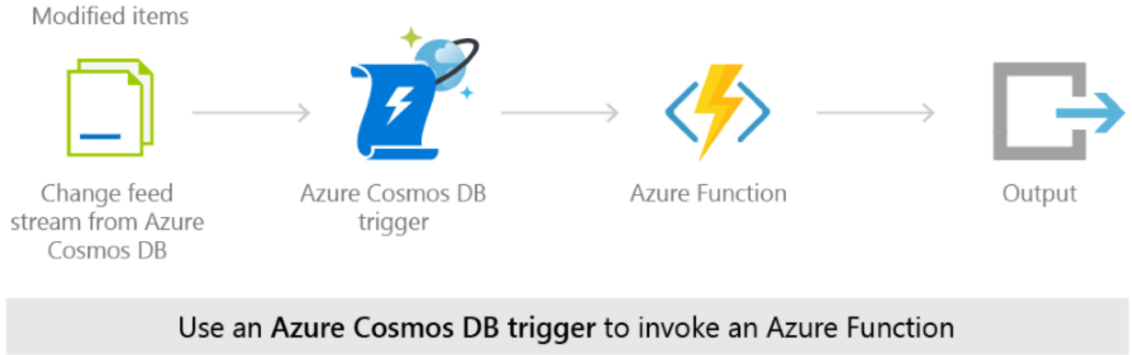
**From your perspective: define 1 partition key per container**

# Data Triggers using Azure Functions

- Triggering a notification or a call to an API, when an item is inserted or updated.
- Real-time stream processing for IoT or real-time analytics processing on operational data.
- Additional data movement by either synchronizing with a cache or a search engine or a data warehouse or archiving data to cold storage.

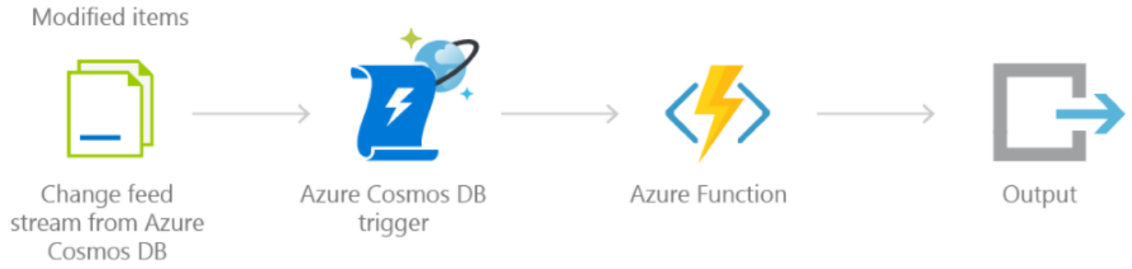


1.



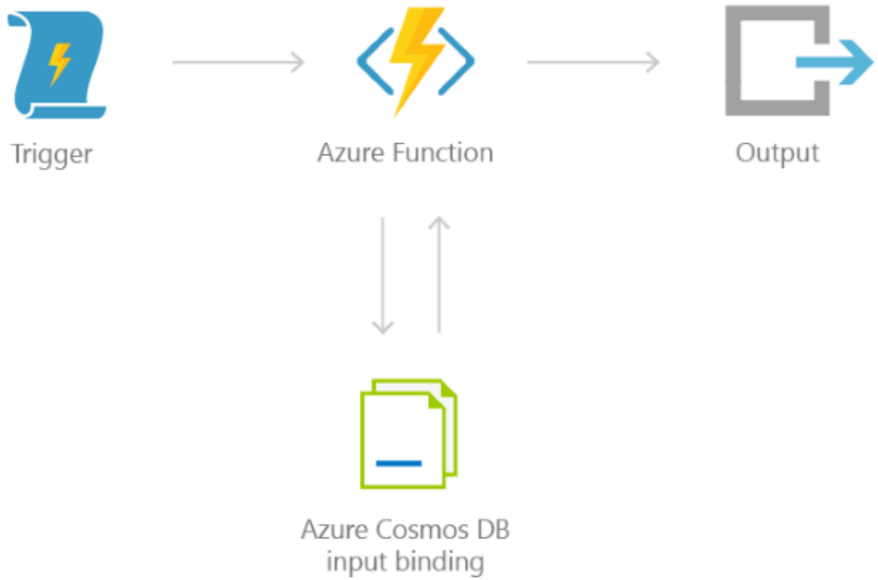

Use an **Azure Cosmos DB trigger** to invoke an Azure Function

1.



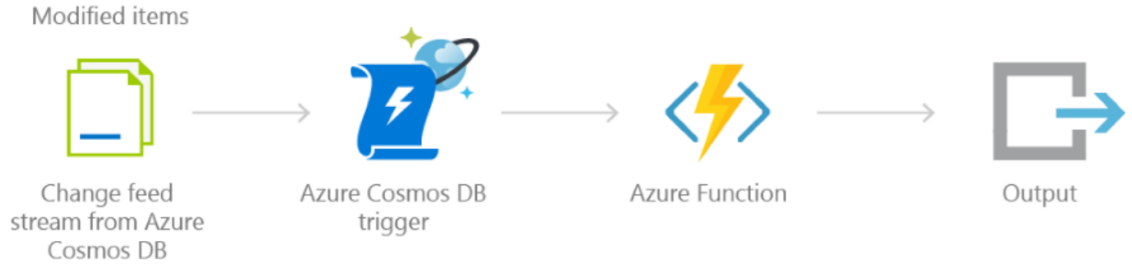

Use an **Azure Cosmos DB trigger** to invoke an Azure Function

2.



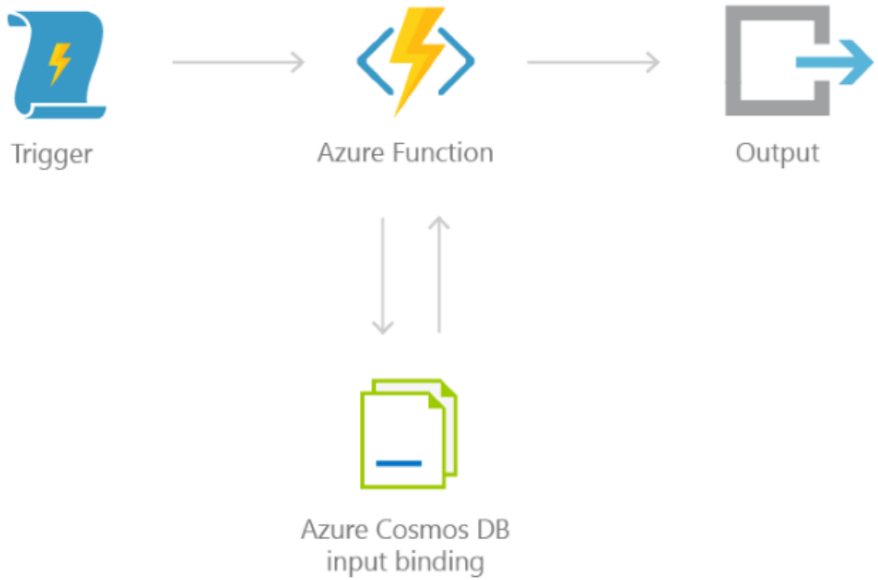

Use an **input binding** to get data from Azure Cosmos DB

1.




Use an **Azure Cosmos DB trigger** to invoke an Azure Function

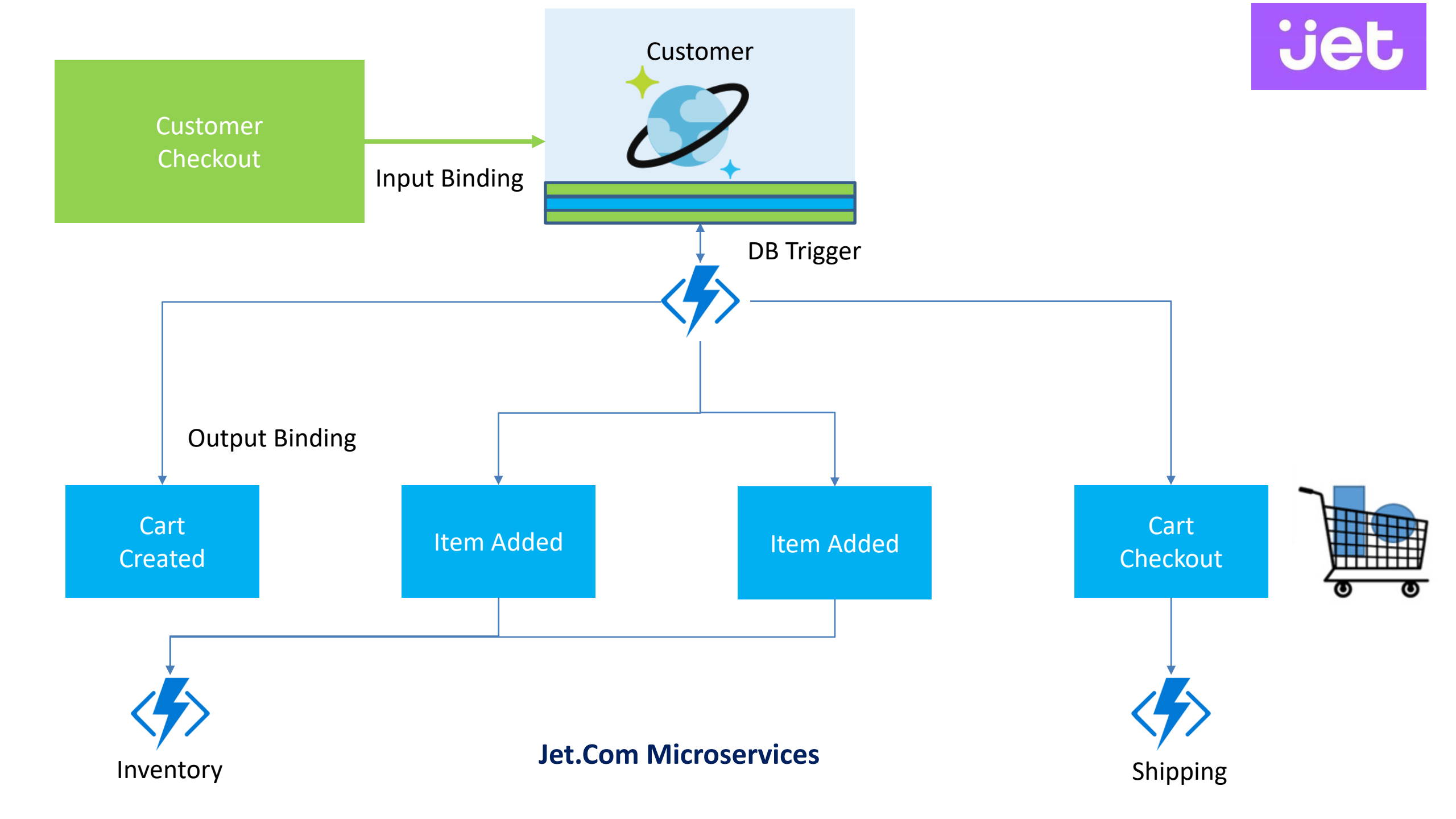
2.




Use an **input binding** to get data from Azure Cosmos DB

3.


Use an **output binding** to write data to Azure Cosmos DB


Customer
Checkout
Input Binding
Customer
DB Trigger
Output Binding
Cart
Created
Item Added
Item Added
Cart
Checkout
Inventory
Jet.Com Microservices
Shipping

```
public static class CosmosTrigger
{
    [FunctionName("CosmosTrigger")]
    public static void Run([CosmosDBTrigger(
        databaseName: "ToDoItems",
        collectionName: "Items",
        ConnectionStringSetting = "CosmosDBConnection",
        LeaseCollectionName = "leases",
        CreateLeaseCollectionIfNotExists = true)]IReadOnlyList<Document> documents,
        TraceWriter log)
    {
        if (documents != null && documents.Count > 0)
        {
            log.Info($"Documents modified: {documents.Count}");
            log.Info($"First document Id: {documents[0].Id}");
        }
    }
}
```

# Let’s see in action

# Important Links

Pricing Calculator
https://azure.microsoft.com/en-us/pricing/calculator/?service=cosmos-db#cosmos-db7aed2059-b457-48cc-a0e9-6744ce81096b

SQL API Query

https://docs.microsoft.com/en-us/azure/cosmos-db/sql-query-getting-started

Azure Cosmos Emulator

https://docs.microsoft.com/en-us/azure/cosmos-db/local-emulator#controlling-the-emulator

Data Migration Tool

http://www.microsoft.com/en-us/download/details.aspx?id=46436

# Questions?

Thank you