Cosmos DB Hands on Workshop

José René del Castillo &
David Sánchez

May 8th , 2019
Hotel Hyatt Regency Polanco

Microsoft

Agenda

8:00-9:00	Breakfast
9:00-10:30	Azure CosmosDB – What, Why & Customer Use cases
10:30-10:45	Break
10:45-11:30	Request Units & Billing, Reserved Capacity, Indexing & Best Practices
11:30-12:00	Lab 1 – Implement Cosmos DB
12:30-1:30	Lunch
1:30-3:00	Lab 2 – Creating and Connecting an App to CosmosDB [Contoso Demo]
3:00-3:15	Break
3:15-4:00	Lab 3 - Migrating SQL to Cosmos [Mexico Only]
4:00-5:30	Networking [Mexico Only]

Microsoft

CosmosDB – What, Why, Uses Cases


James Whittaker - @docjamesw
“We are reducing the world to data”

LAS APLICACIONES MODERNAS ENFRENTAN NUEVOS DESAFÍOS

Administrar y sincronizar datos distribuidos en todo el mundo

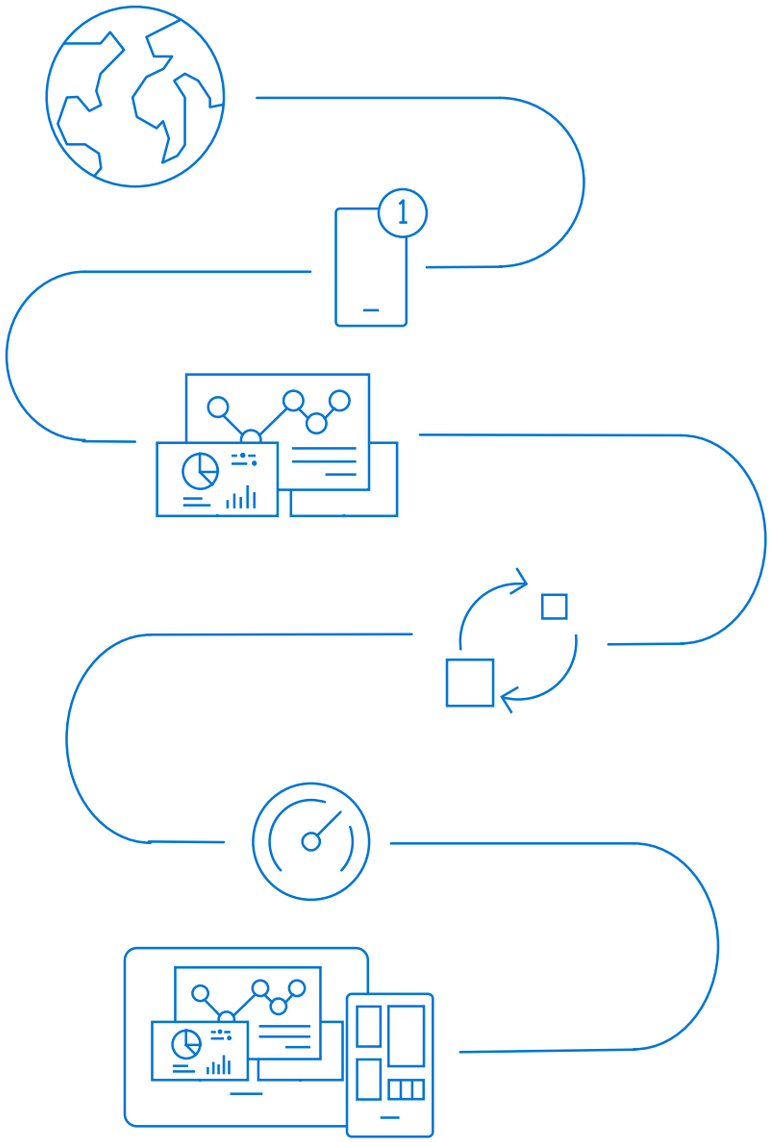
Entregar personalización que responde muy adecuadamente y en tiempo real

Procesamiento y análisis de muchos datos complejos

Escalación tanto de rendimiento como de almacenamiento con base en demanda global

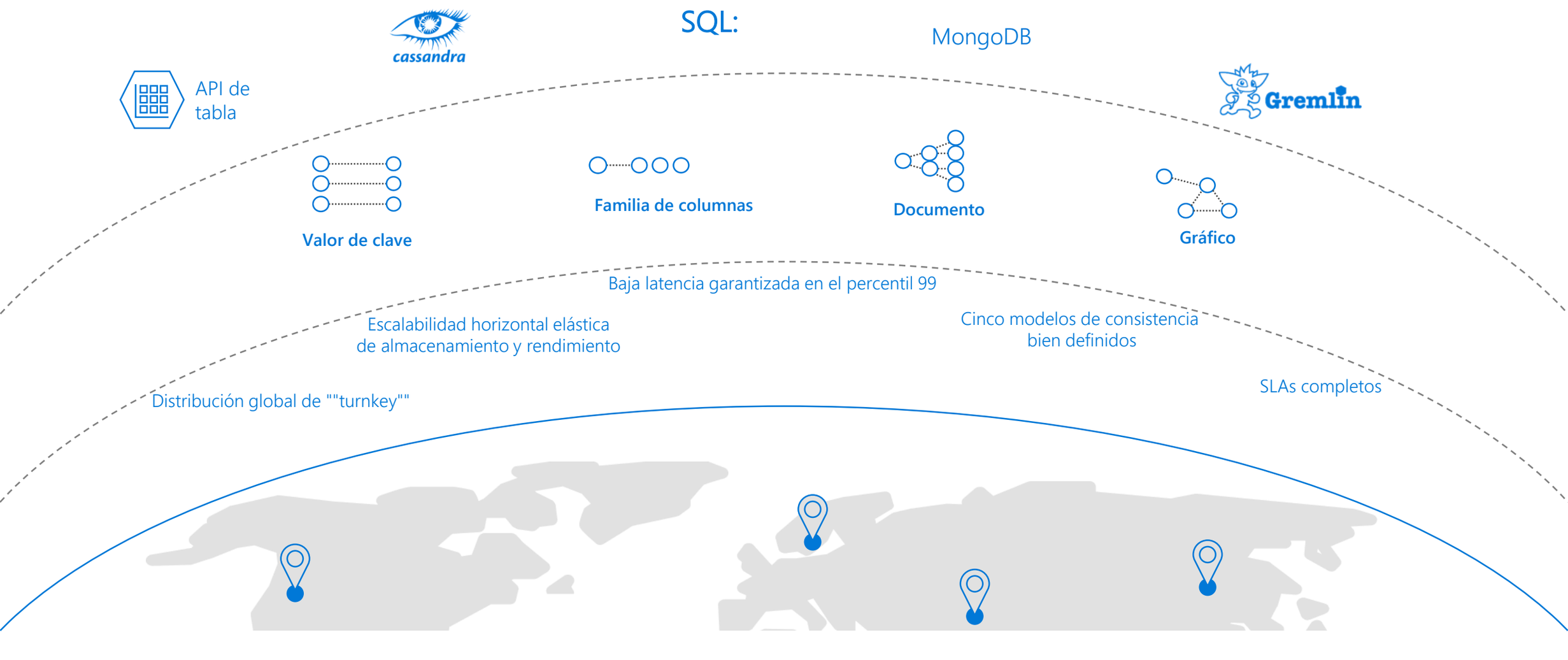
Entrega de baja latencia a usuarios globales

Modernización de aplicaciones y datos existentes



AZURE COSMOS DB

Un servicio de base de datos globalmente distribuido, escalable masivamente y multi-modelo



1. Performance around the world with a serverless architecture

ContosoAir wants to improve app performance for customers booking flights on the app around the world

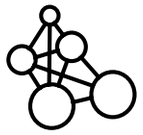
Previously the ContosoAir apps was deployed from a single region, leading to performance limitations

Performant and consistent app experience for all users across multiple regions

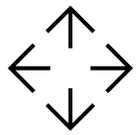
When developing accurate, globally distributed apps ContosoAir to consider the database that will best support its app. Some considerations will include:

Data close to where the apps users to ensure low latency and high performance

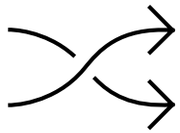
Managing complex schemas



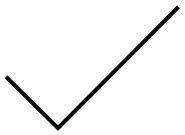
Ability to scale based on global demand



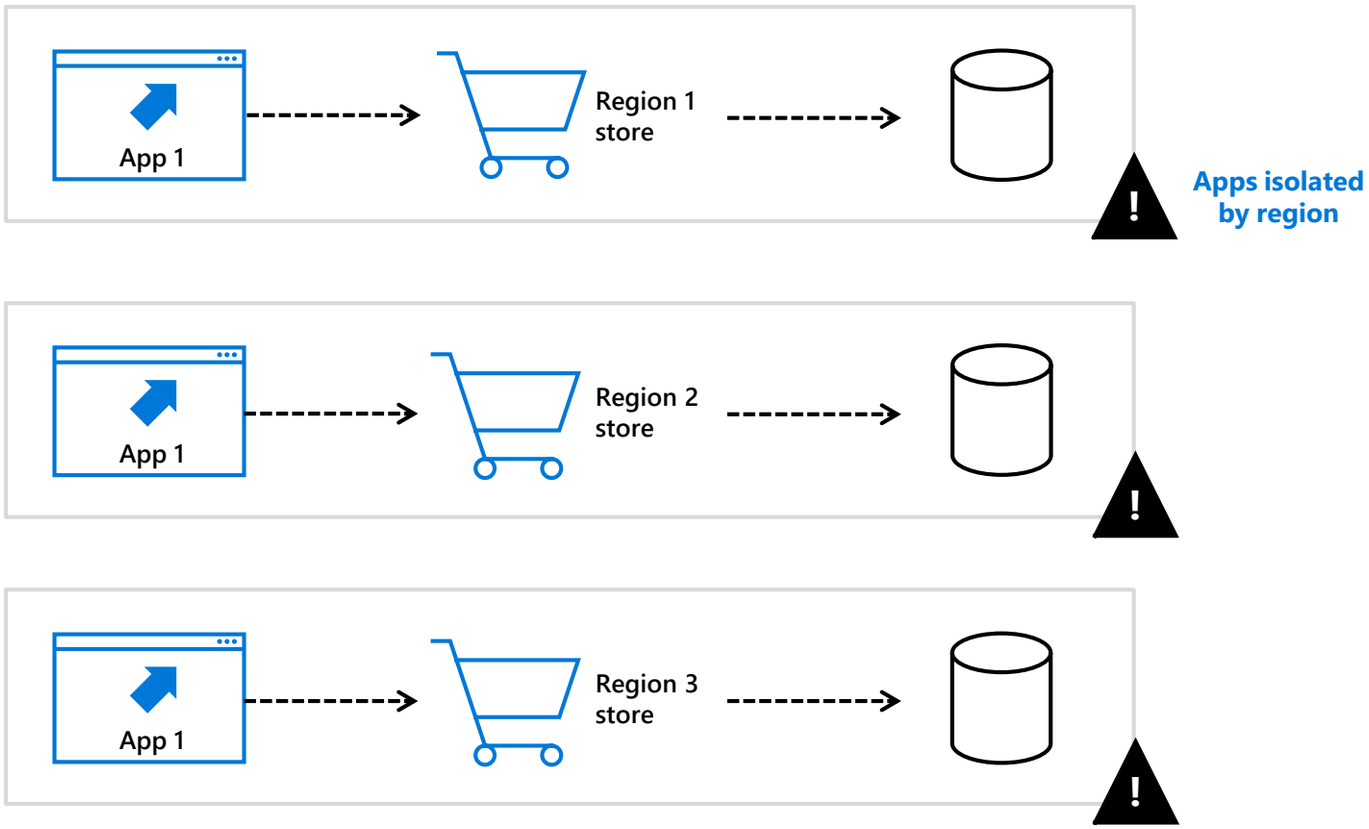
Choice of consistency models



An always-on system



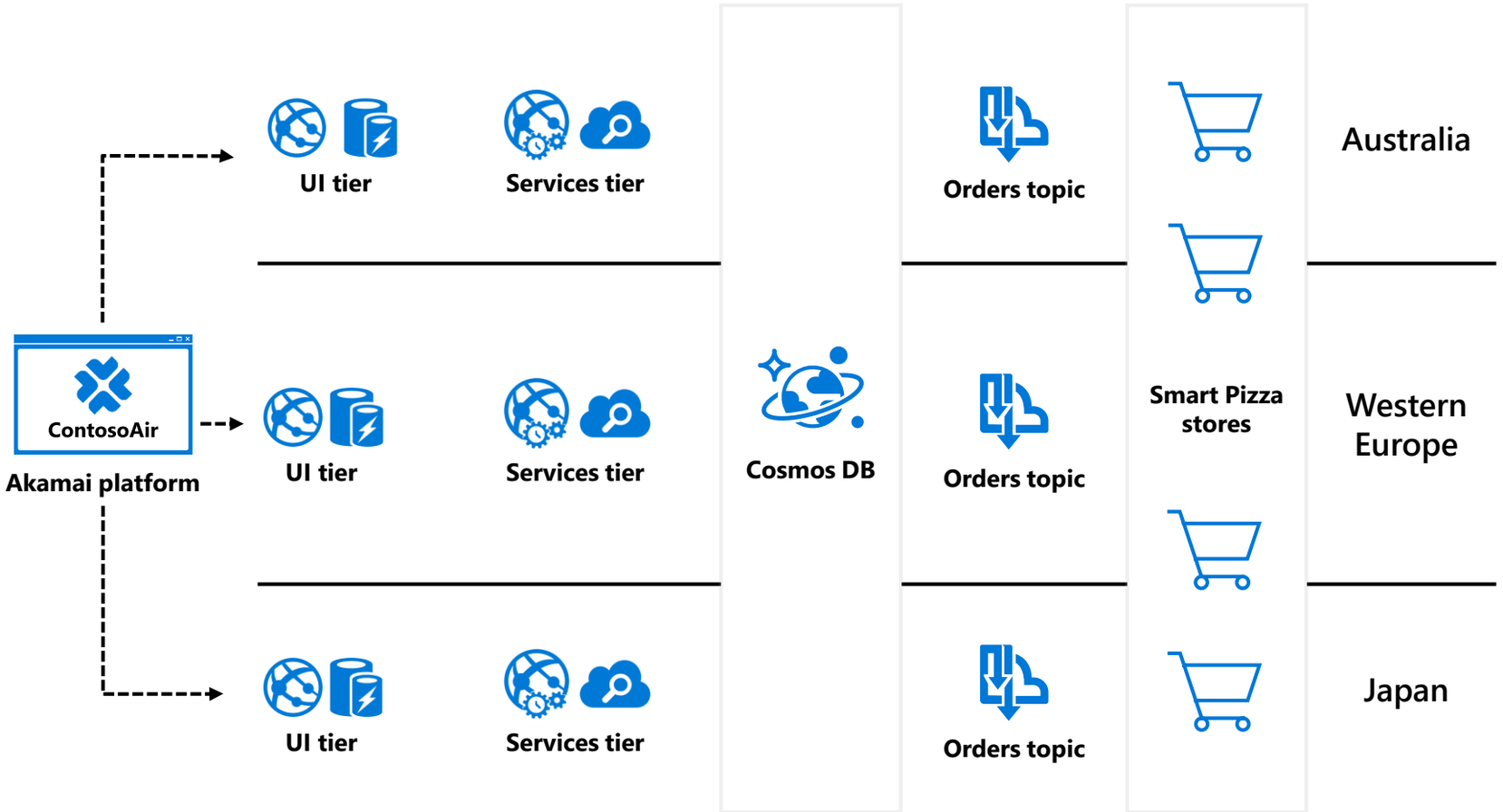
App not distributed across regions, users experience inconsistent



Challenges

- Unable to uniformly scale multiple regions
- Difficult to configure data by region

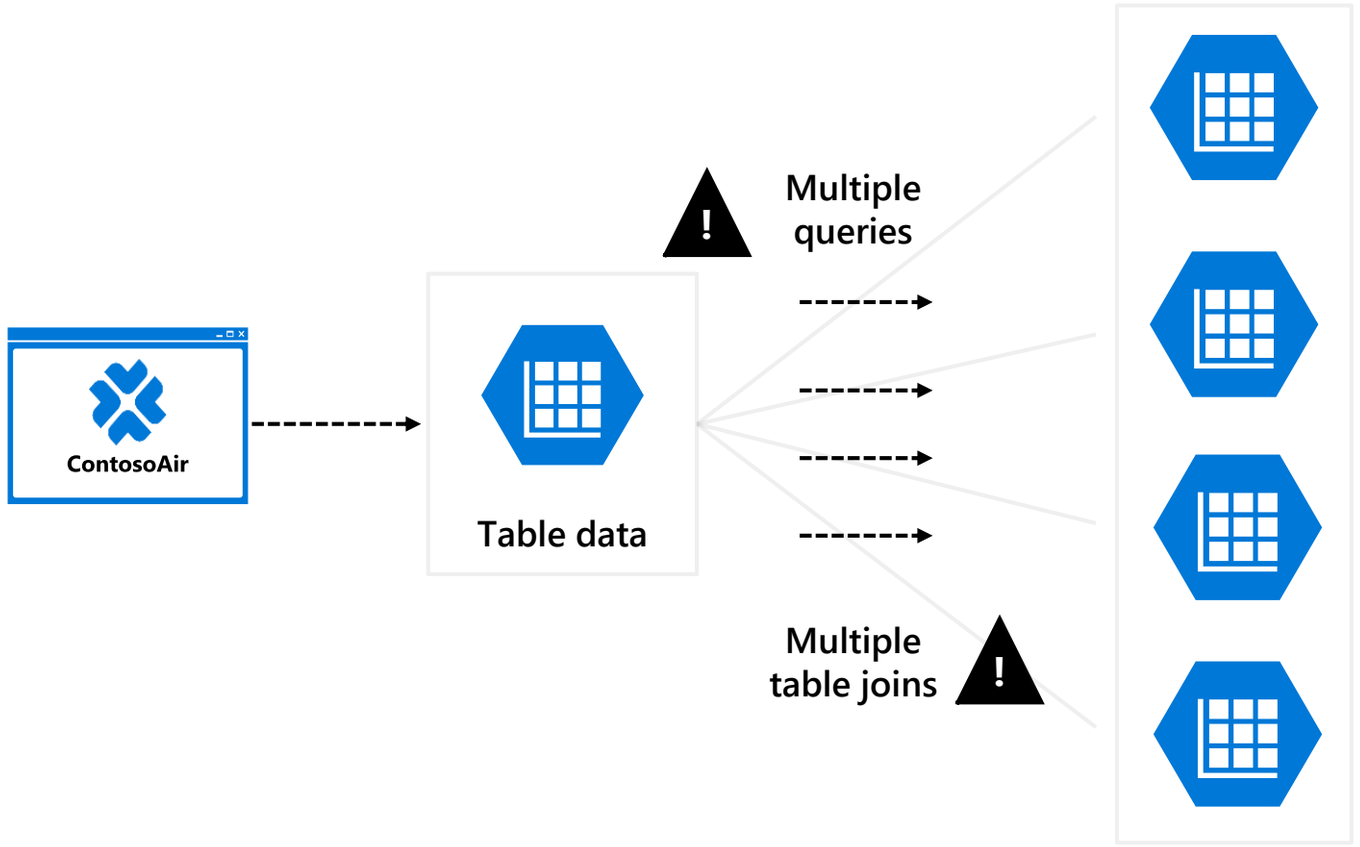
Global distribution across all regions improves user experience



Solutions

- Turnkey database replication across multiple data centers
- Policy-based geo-fencing for data governance and compliance
- Dynamically add or remove regions
- Low latency and improved app performance in all regions

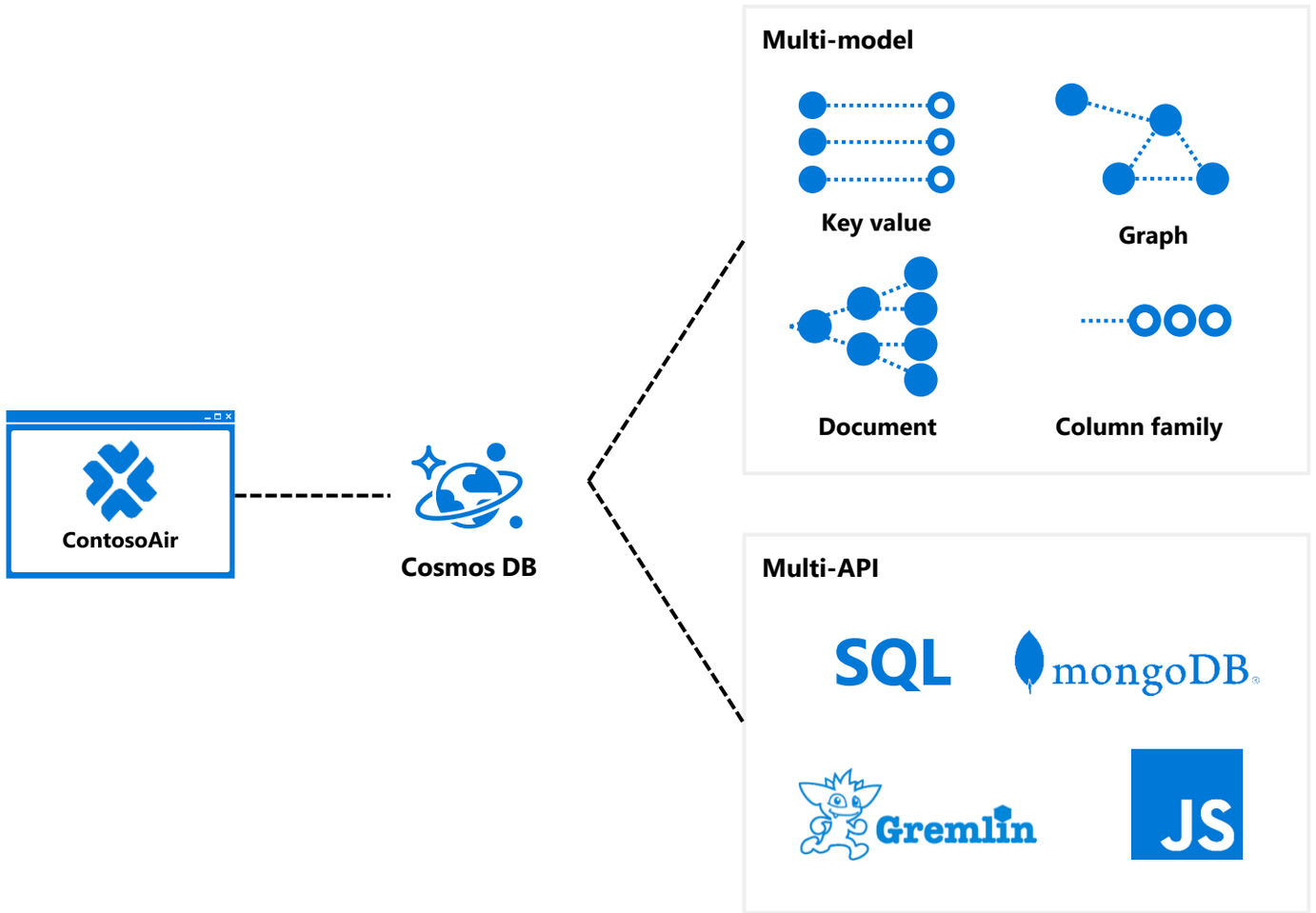
Increasingly complex data results in schema challenges



Challenges

- Requires complex and diverse data schemas
- Poor throughput due to multiple joins and queries

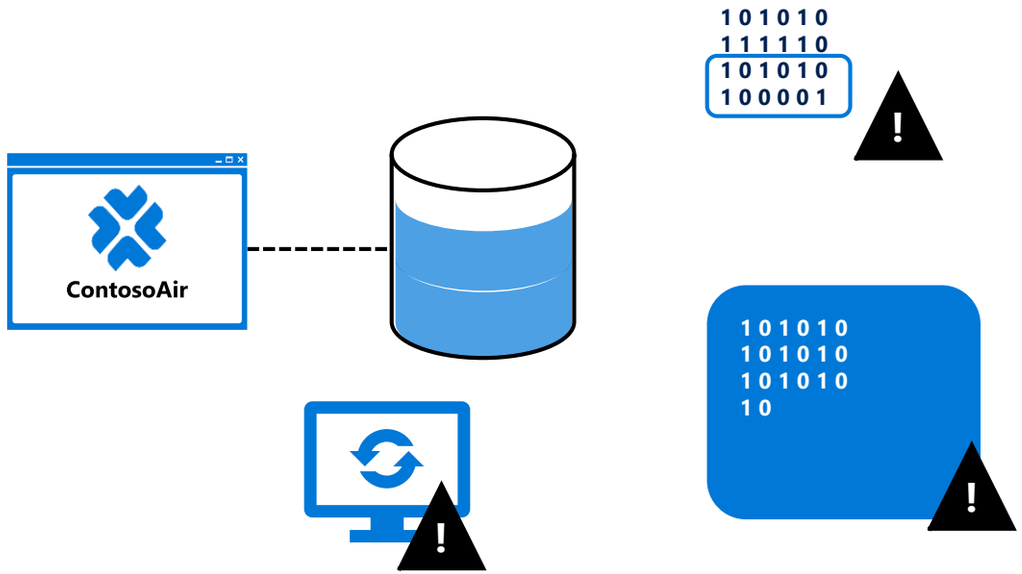
Analyze unstructured data without schema or index management



Solutions

- Use key-value, graph, and document data in one service
- Access data with your favorite API

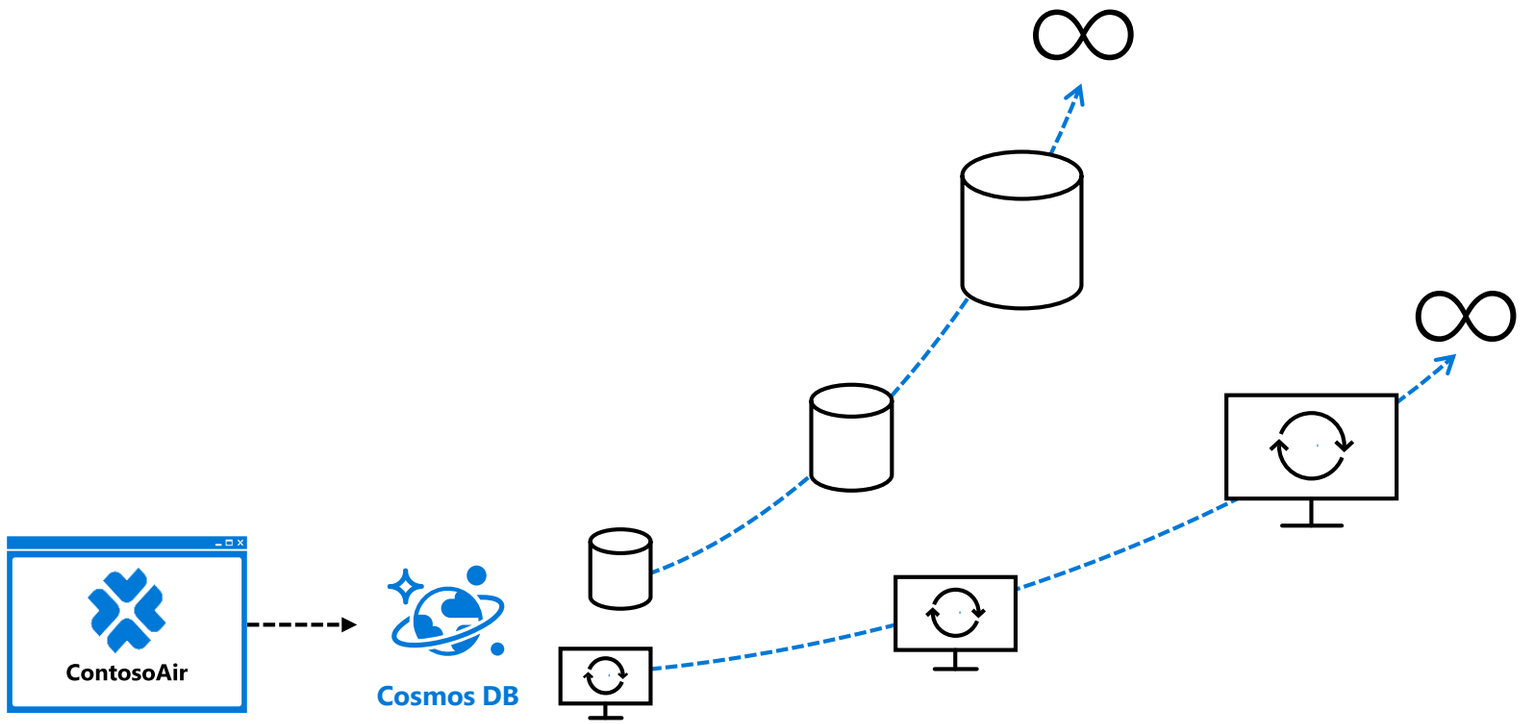
Apps limited ability to scaling does not respond to changes in demand



Challenges

- Fixed scaling results in under- or over-provisioning
- Throughput slows as scale expands

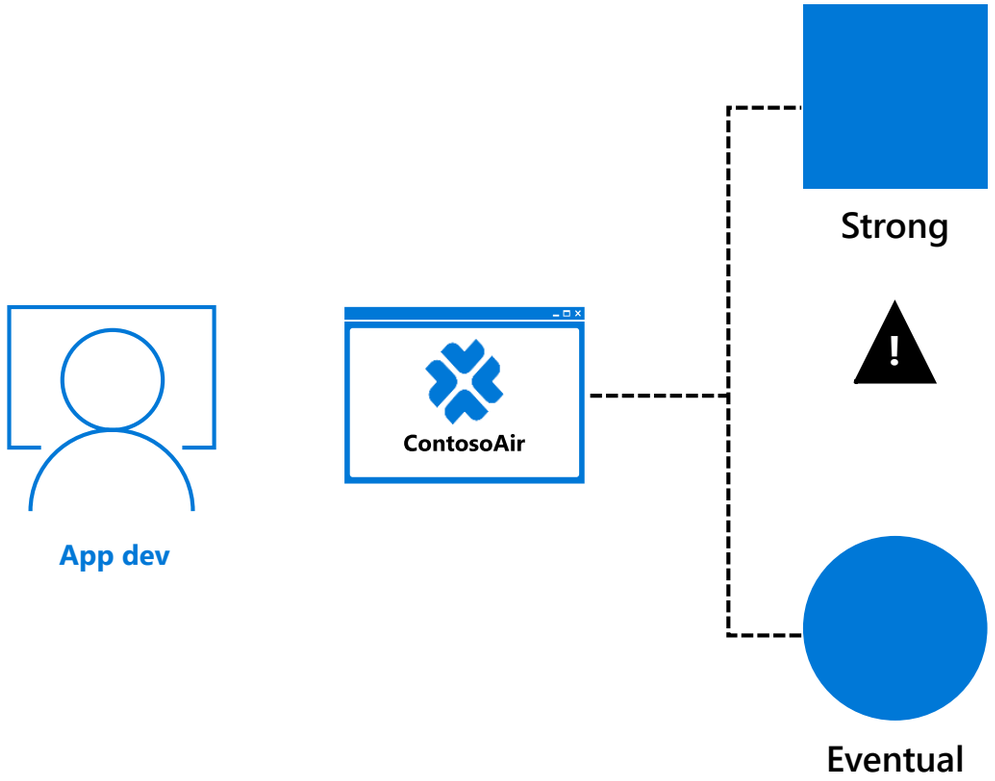
Infinite storage and throughput scale



Solutions

- Independent and elastic scale across regions
- Fully managed partitions and containers enable limitless scale

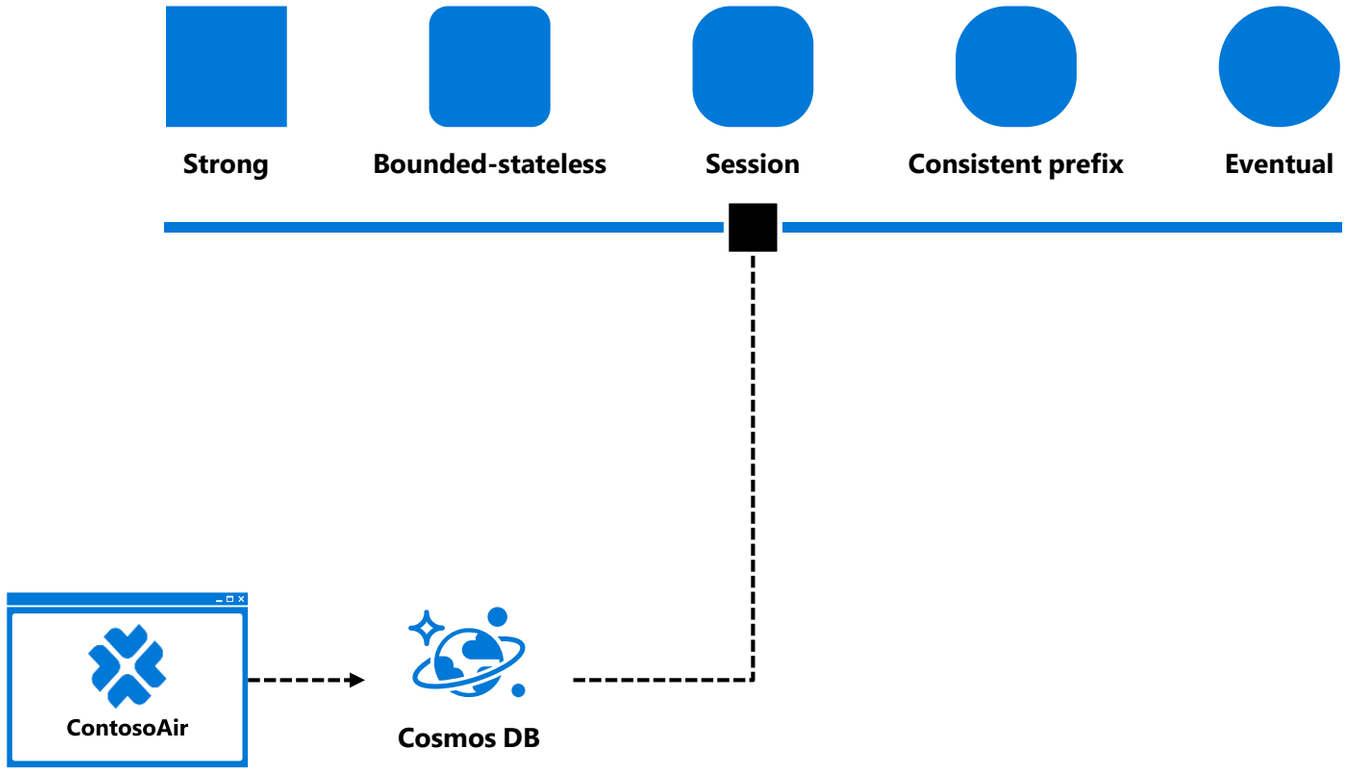
Must choose between extremes in consistency



Challenges

- Burdens app devs with tradeoffs between strong and eventual consistency

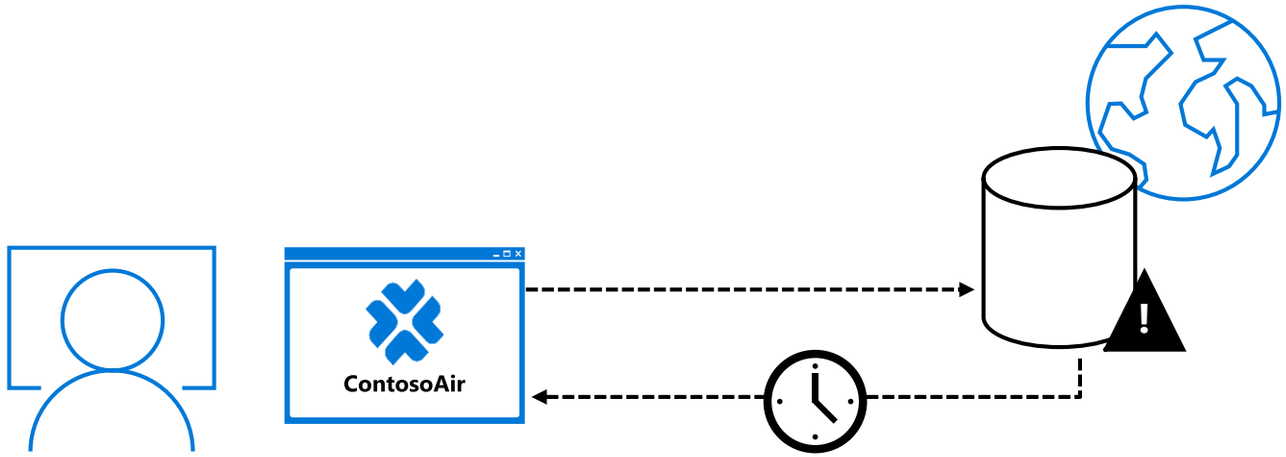
Well defined consistency models provide options



Solutions

- Five consistency levels with clear PACELC tradeoffs
- Levels can be overridden on a per-request basis

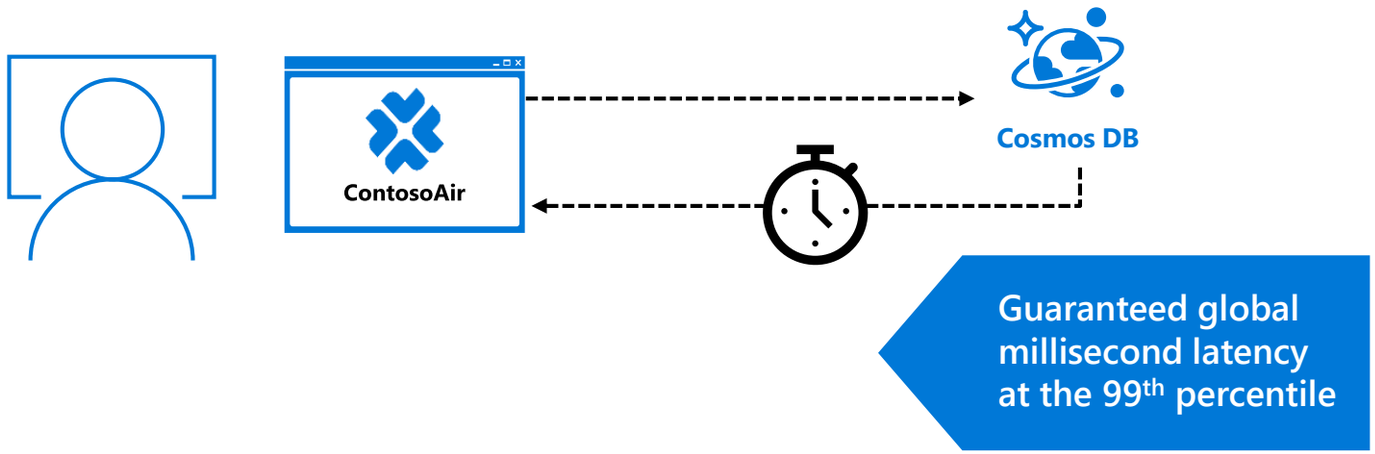
Well defined consistency models provide options



Challenges

- Reads and writes not served from local region
- Asynchronous indexing at sustain ingestion rates

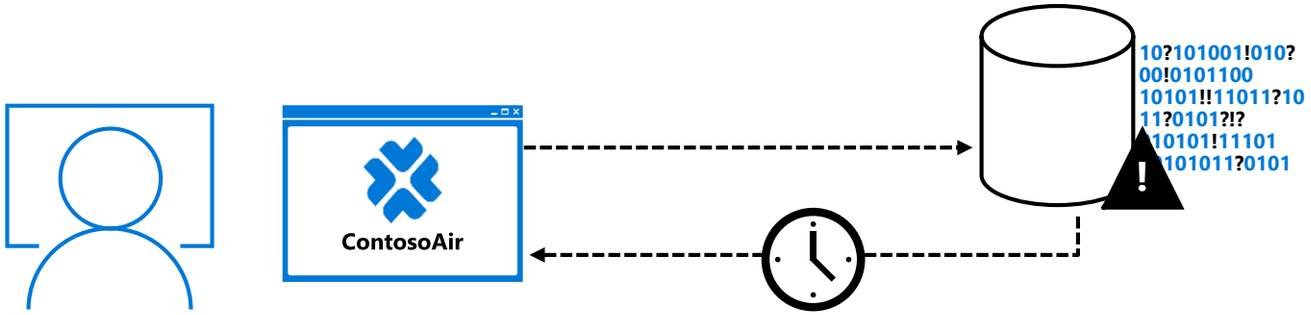
Well defined consistency models provide options



Solutions

- Write-optimized, latch-free database engine designed for SSDs
- Automatic indexing at sustained ingestion rates

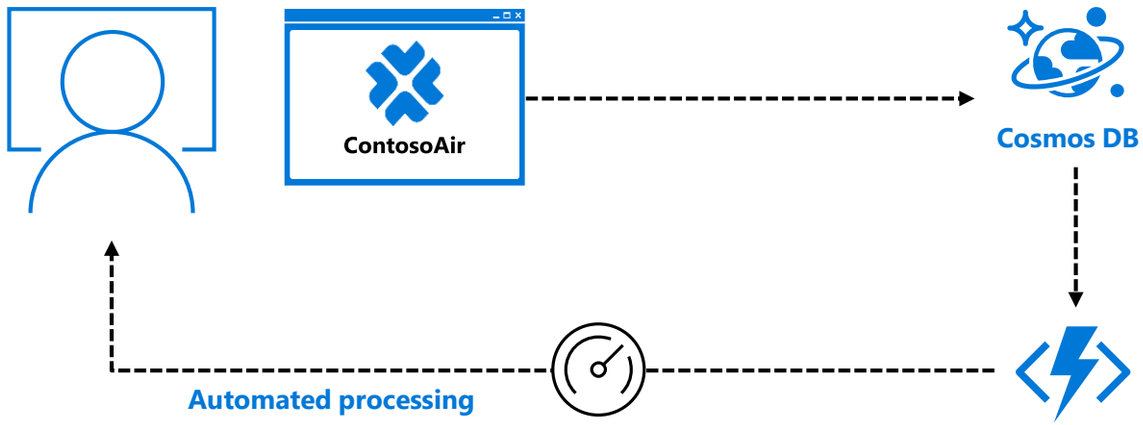
Coding inconsistences complicate app development and slows data retrieval



Challenges

- Difficult to search data due to tagging errors
- Data administration is complex and time-consuming

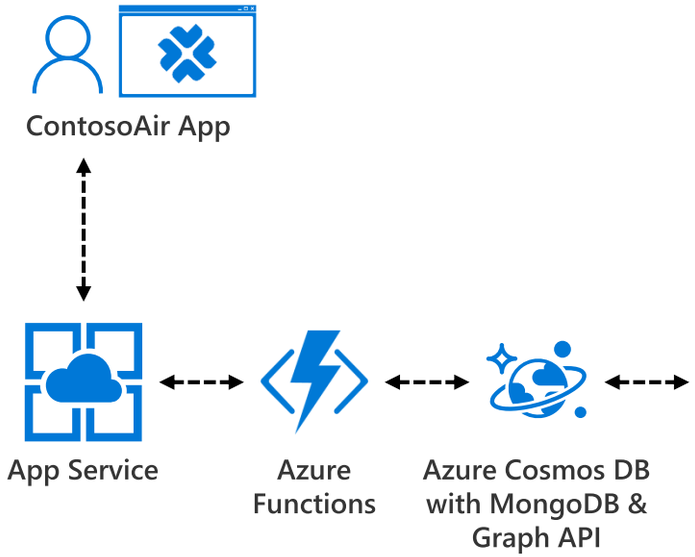
Azure Functions processes data with serverless code architecture

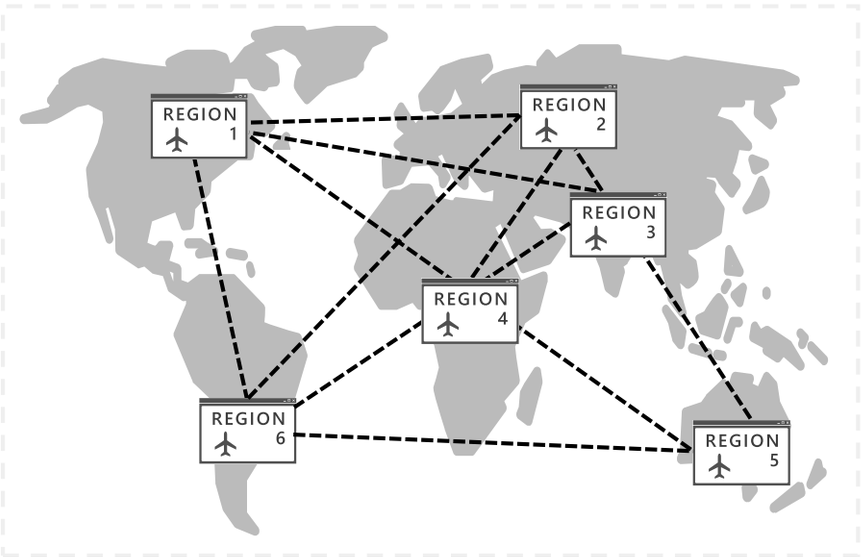


Solutions

- Time-based processing to clean up data based on customer business logic
- Execute serverless code that triggers automatic events

Global distribution improves availability and performance around the world





Solutions

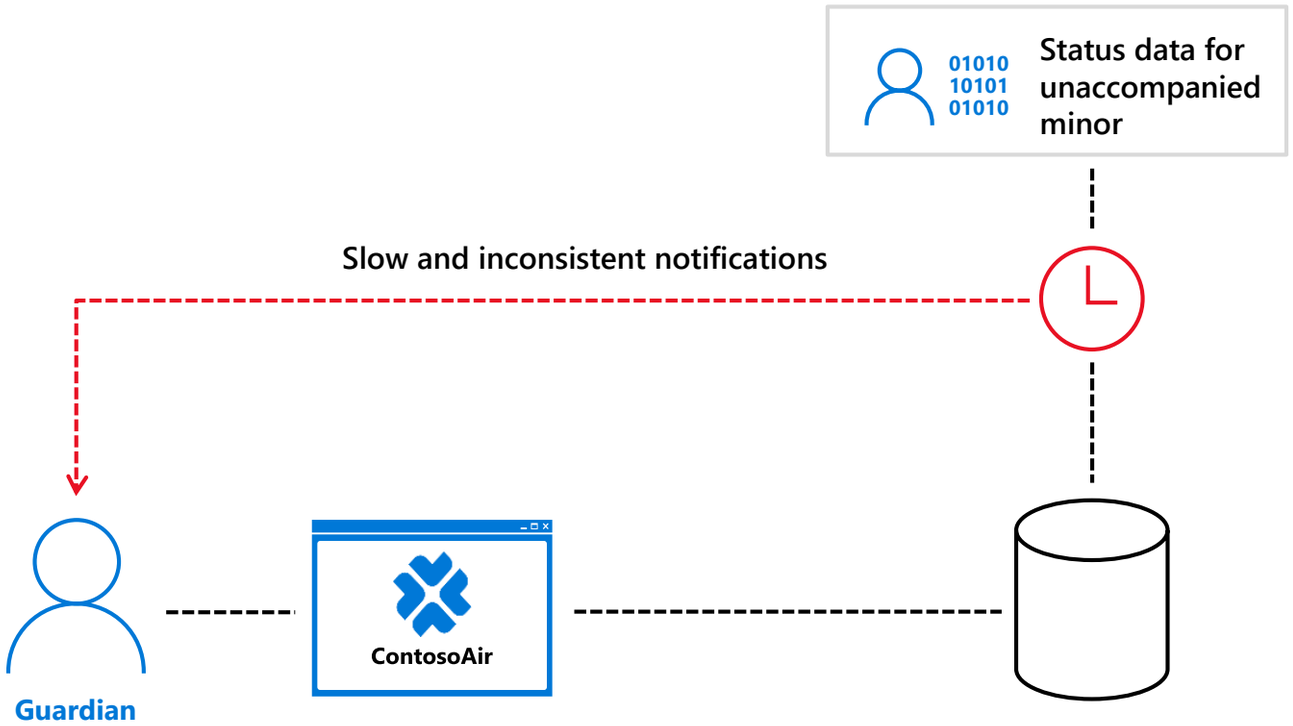
- Customers enjoy HA and performance anywhere with Azure Cosmos DB
- Regions can easily scale with customer demand through global database replication
- Azure Functions help streamline development in a serverless architecture

2. Greater customer choice and real-time notifications

Sending an unaccompanied minor on a flight can be a stressful and complex experience for a guardian

Contoso seeks to provide a better experience by enabling customers to select airline escorts and providing real-time status updates

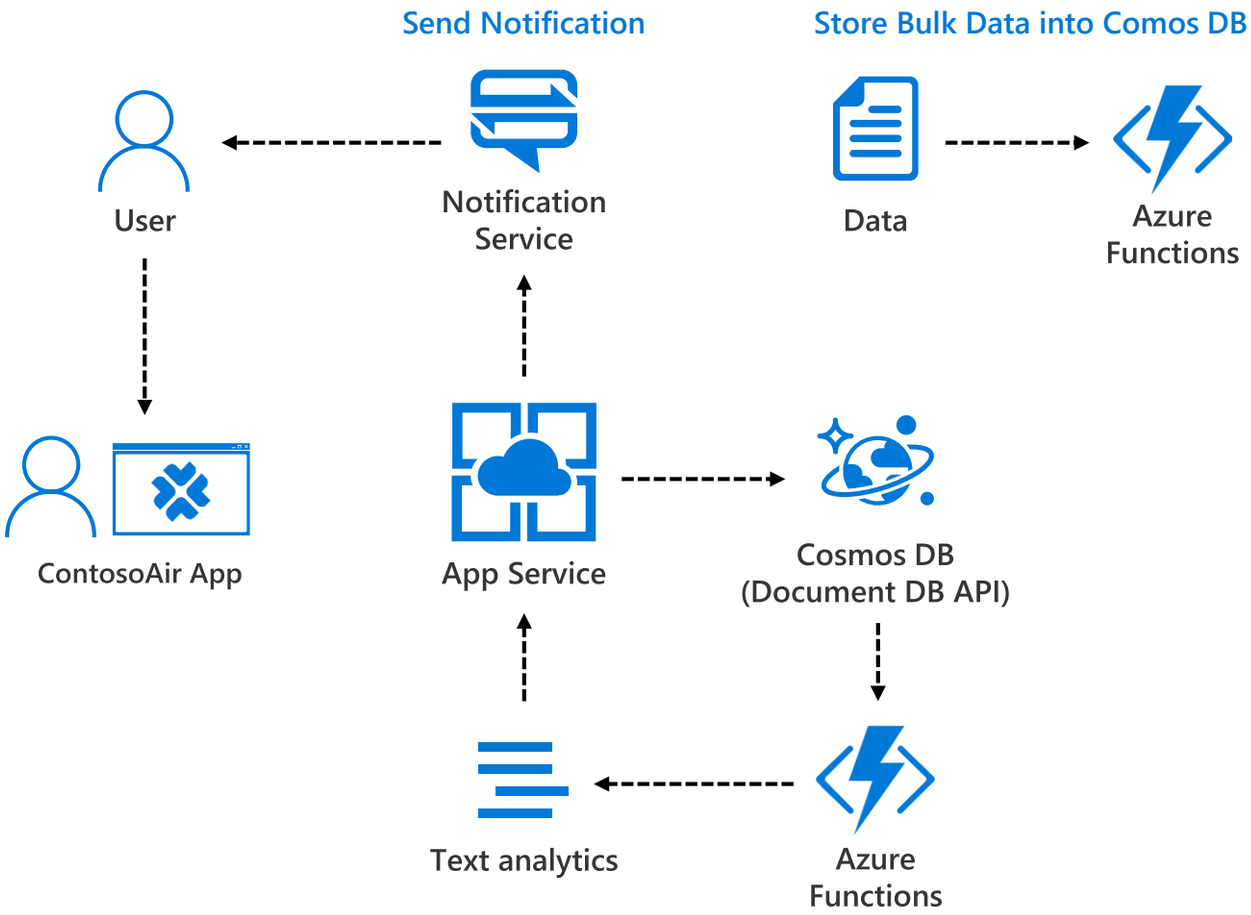
Legacy application migration



Challenges

- Slow notifications of key information inconvenience and frustrate customers

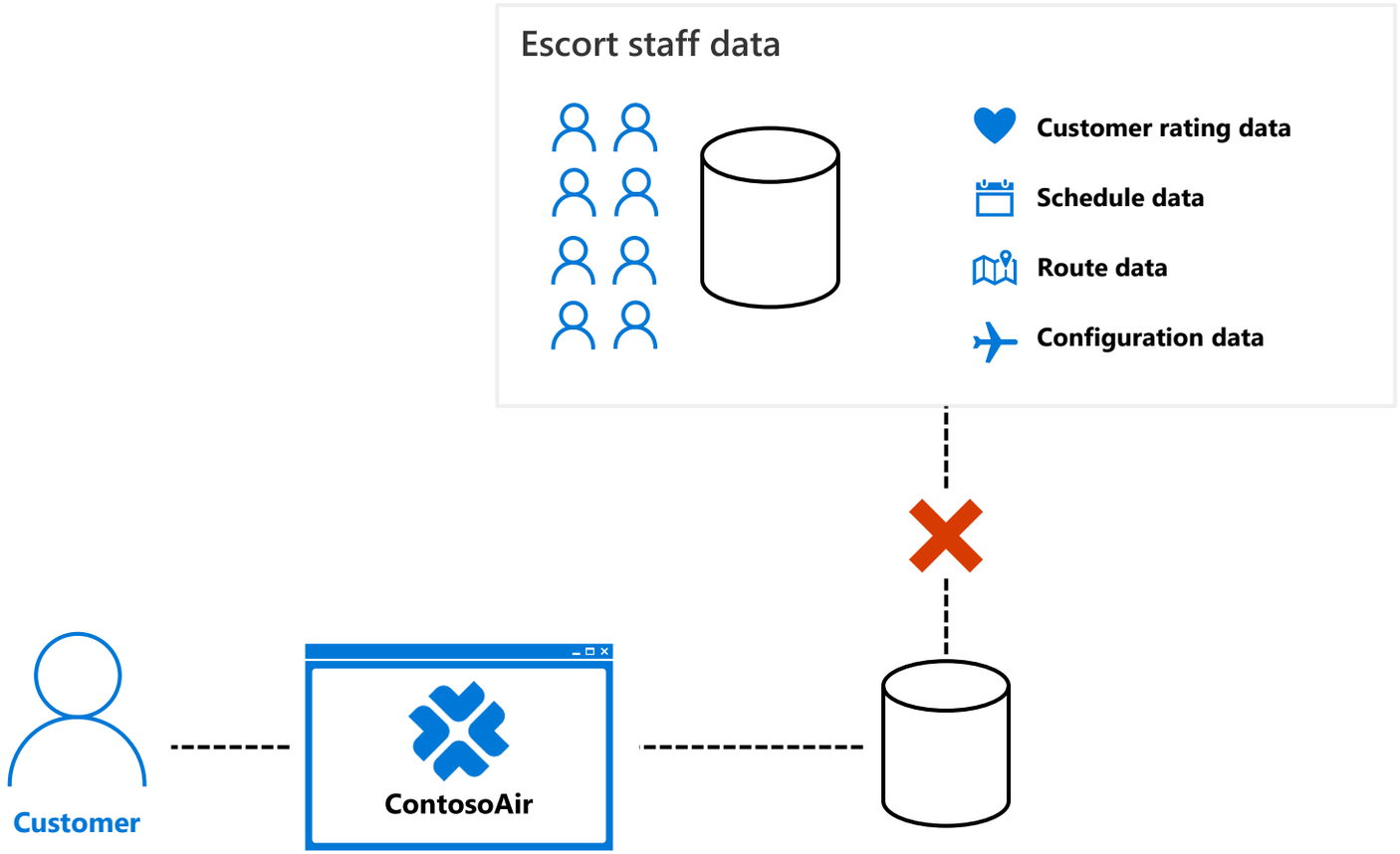
Real-time notifications improve customer awareness and satisfaction



Solutions

- Enables policy recommendations based upon customer behaviors
- Text Analytics API processes text to understand user sentiment

Inadequate data integration limits awareness for customers



Challenges

⚠ Customers have disparate data which create complex relationships

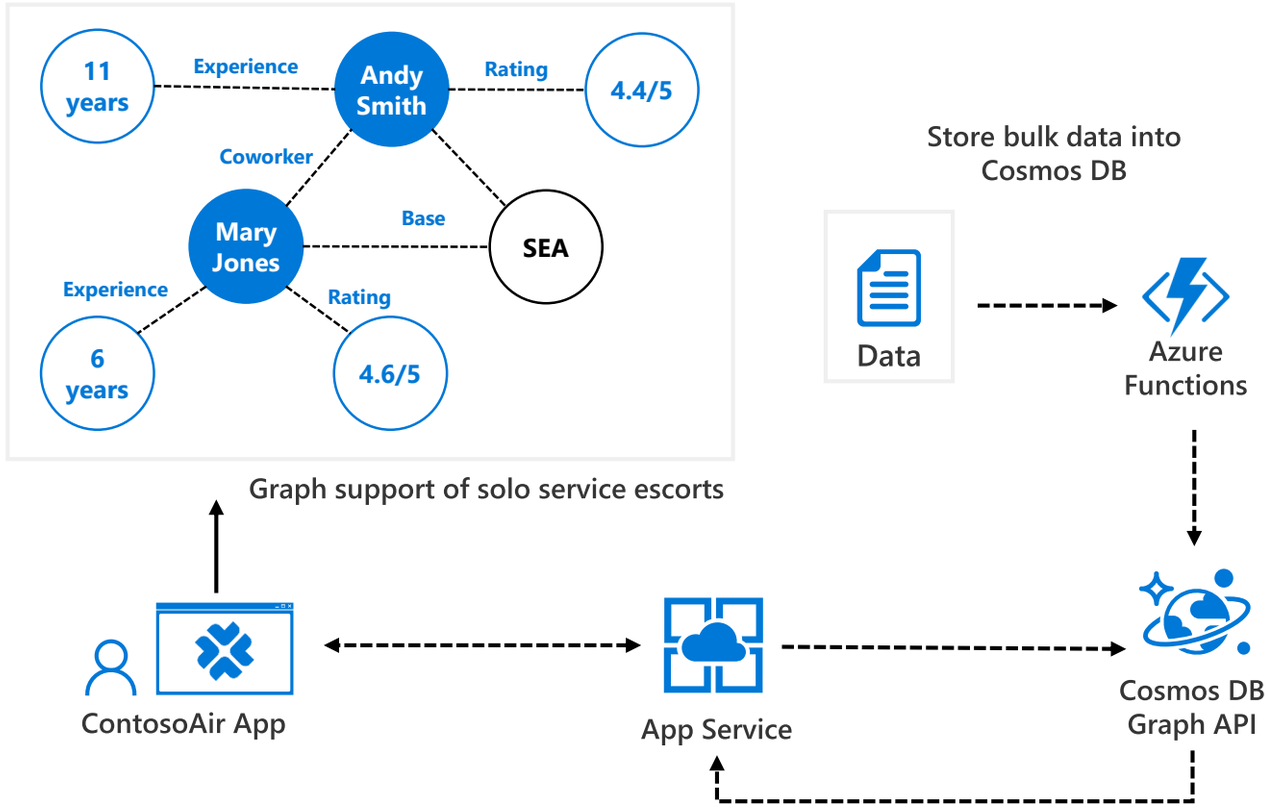
⚠ Querying over complex relationships can slow down queries making it challenging to get fast insights

Example:

cannot review airline escorts for unaccompanied minors

Customer dissatisfaction due to limited awareness and choice

Graph support shows codeshare flights for improved customer options



Solutions

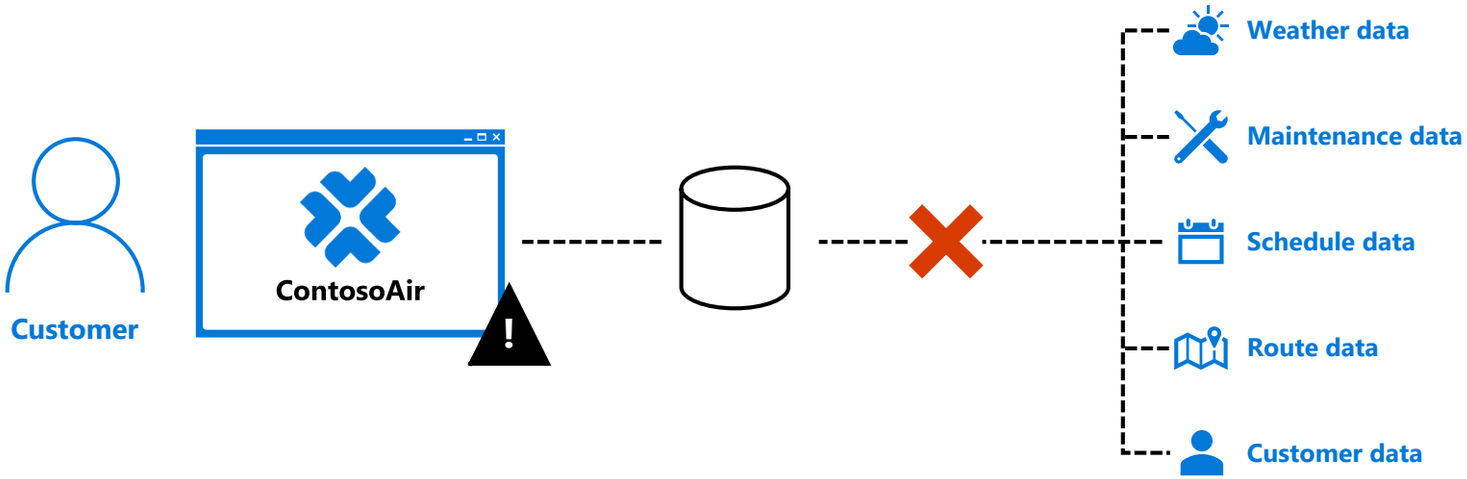
- Cosmos DB retrieves disparate airline staff data
- Graph API displays disparate staff data to help customers make informed decisions about escorts

3. Intelligent predictions based on complex data

Flight delays and cancellations result from a number of factors and are difficult to predict

ContosoAir can enhance the customer experience by forecasting flight delays and communicating with customers in real-time

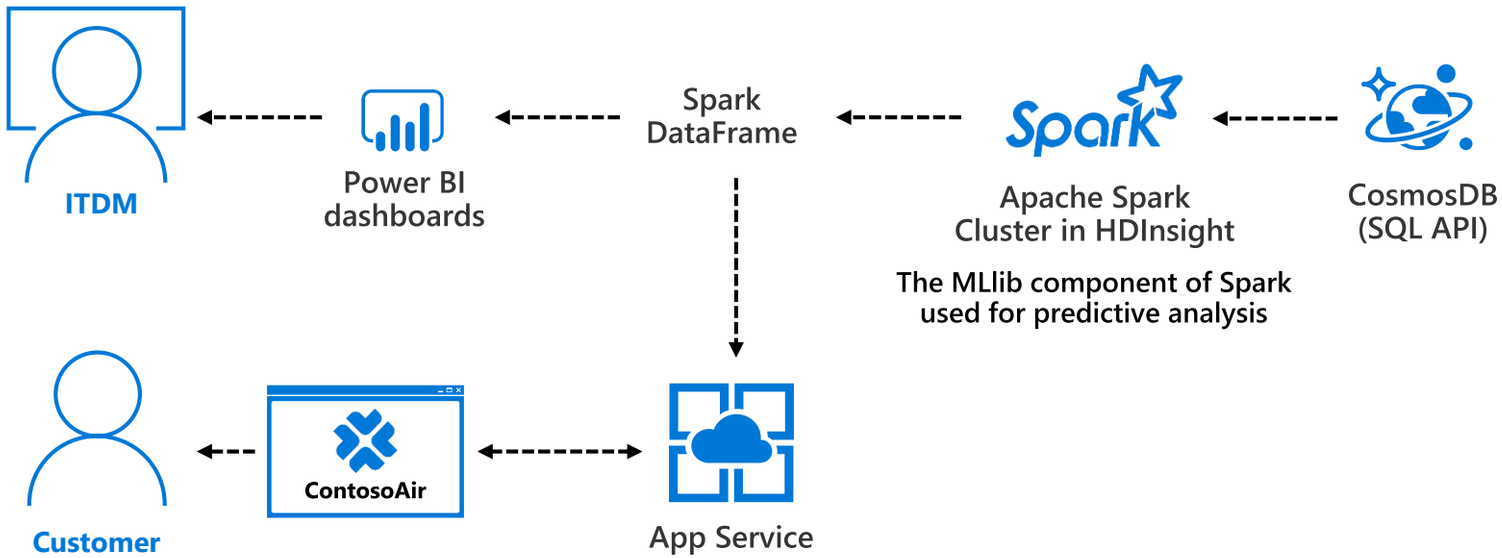
Unable to predict flight delays from diverse data sources



Challenges

- Customers do not receive forecasts of potential flight delays
- Customers cannot assess risk of flight delays when booking

Predictive analytics provide customers a complete view of flight itineraries



Solutions

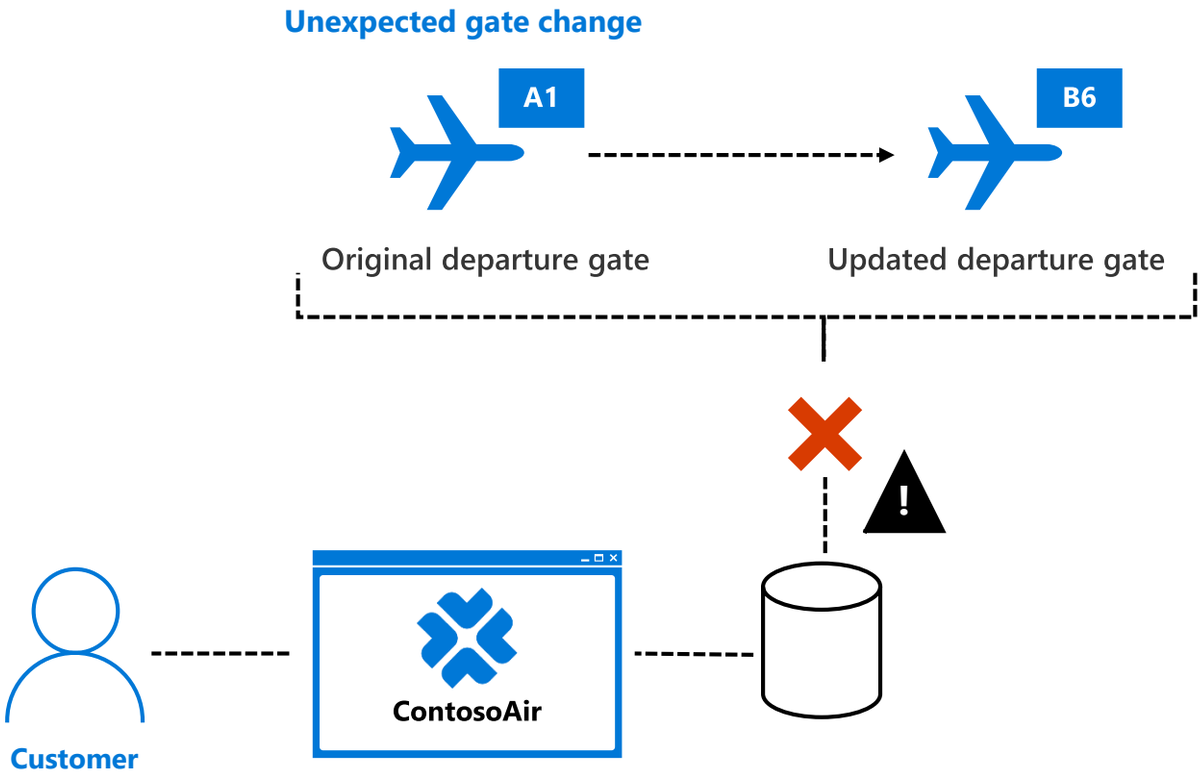
- Azure Cosmos DB and Machine Learning with Apache Sparks uncover risk of flight delays
- Embedded Power BI visualizations illustrate the risk of delays to customers in real-time

4. Improved customer awareness through event-driven scenarios

Airport gate assignments are occasionally changed with little notice, leading to customer confusion and frustration

ContosoAir needs to quickly learn and communicate unexpected information from diverse data sources to maintain customer satisfaction

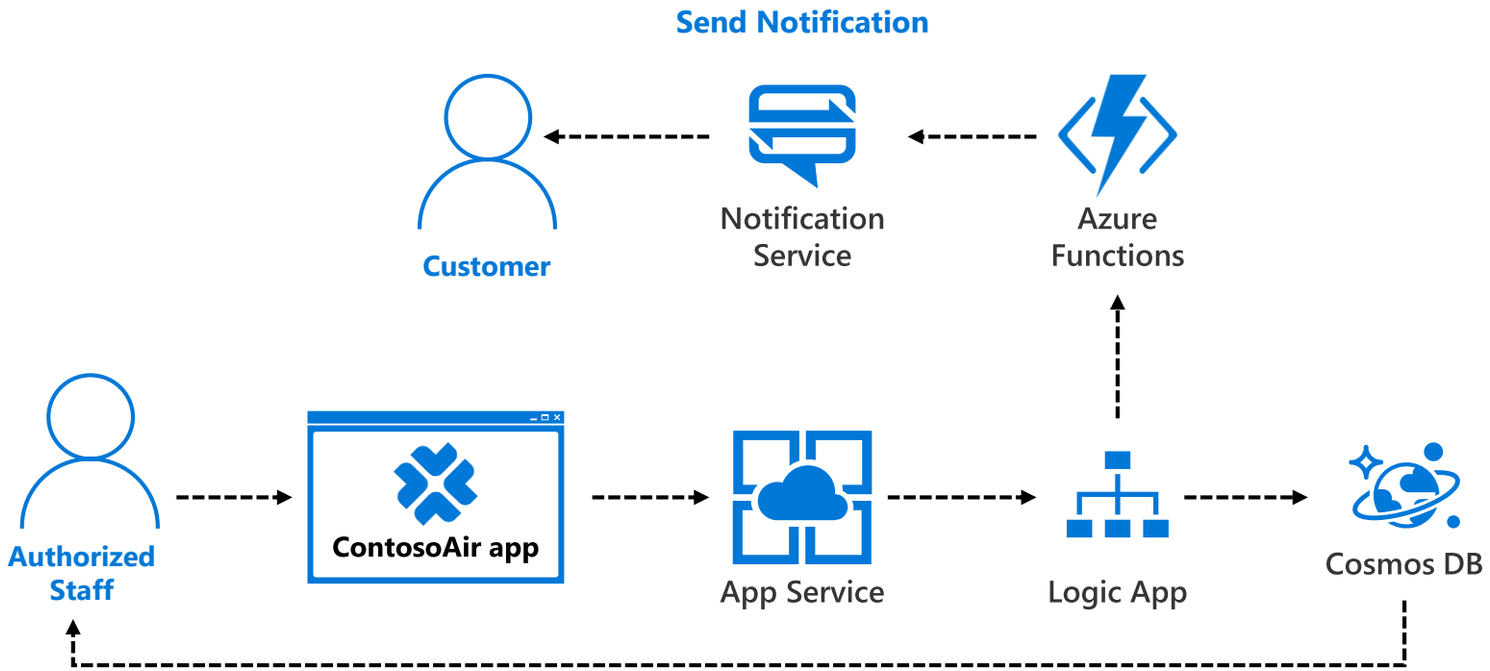
Unable to predict flight delays from diverse data sources



Challenges

- Difficult to quickly learn and communicate gate changes to the right customer
- Inadequate awareness leads to panicked customers and missed flights

Predictive analytics provide customers a complete view of flight itineraries



Solutions

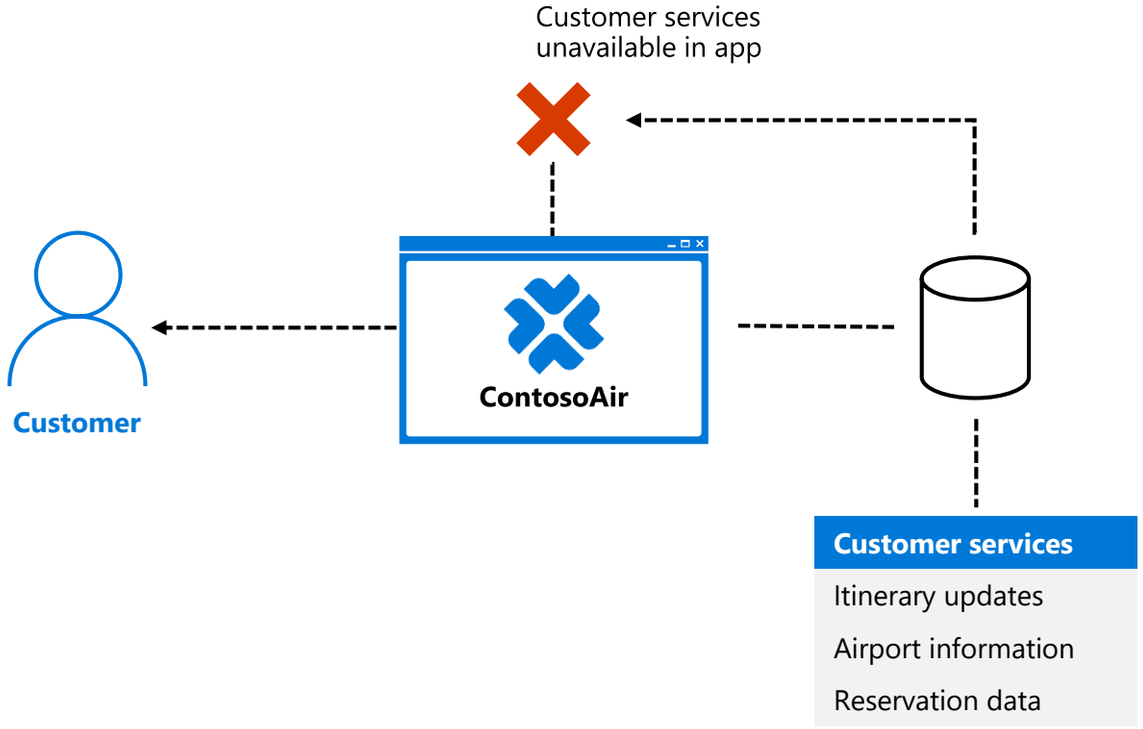
- Terminal gate information quickly retrieved from Cosmos DB through Azure Functions
- Customers received real-time updates with events triggering Azure Functions code

5. Responsive customer service through intuitive interactions

Airlines often struggle to provide reasonable customer service wait times or intelligent ways to gather and share customer feedback

ContosoAir seeks to automate services and enable intelligent feedback to improve customer satisfaction

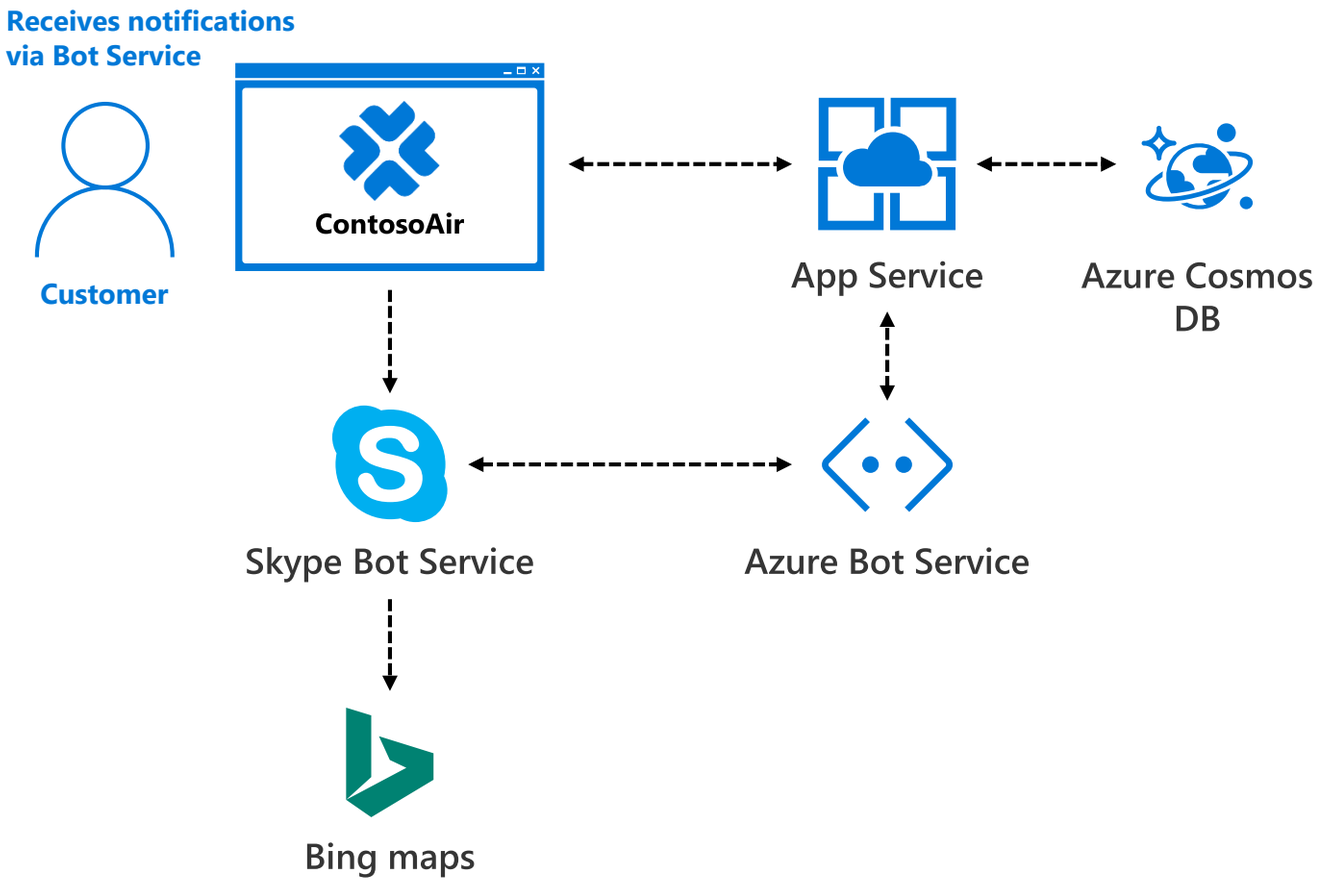
Limited customer service features directly available on app



Challenges

- Limited app functionality necessitates customer service calls
- Long customer service wait times

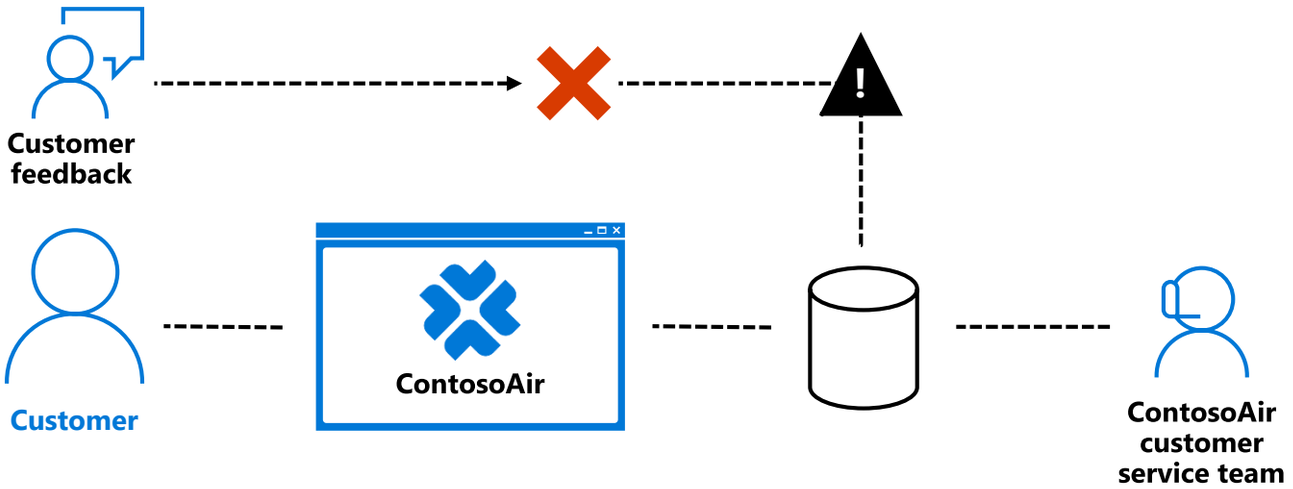
Intuitive and speedy customer service thanks to intelligent bots



Solutions

- Bot service automation unburdens customer service resources
- Bot service provides faster flight scheduling, reservation updates, and related services

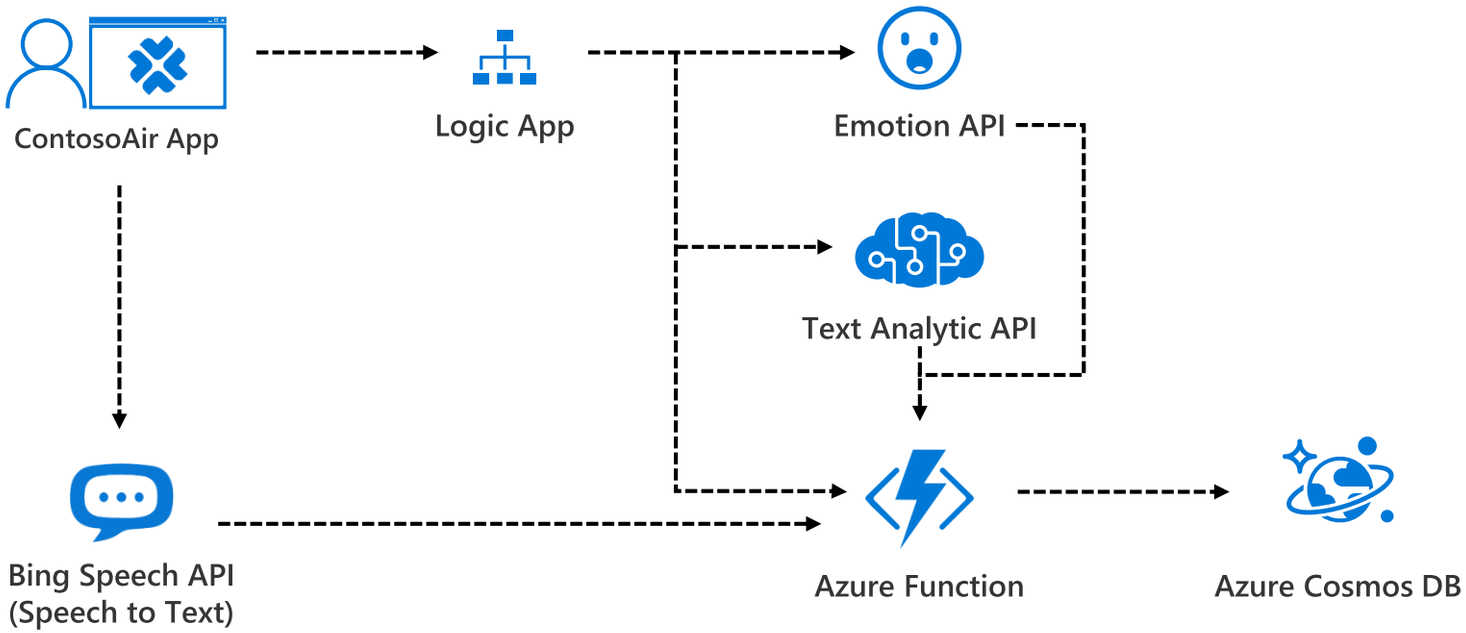
Delayed and incomplete understanding of customer sentiment hinders service



Challenges

- Unable to understand and apply complex customer feedback

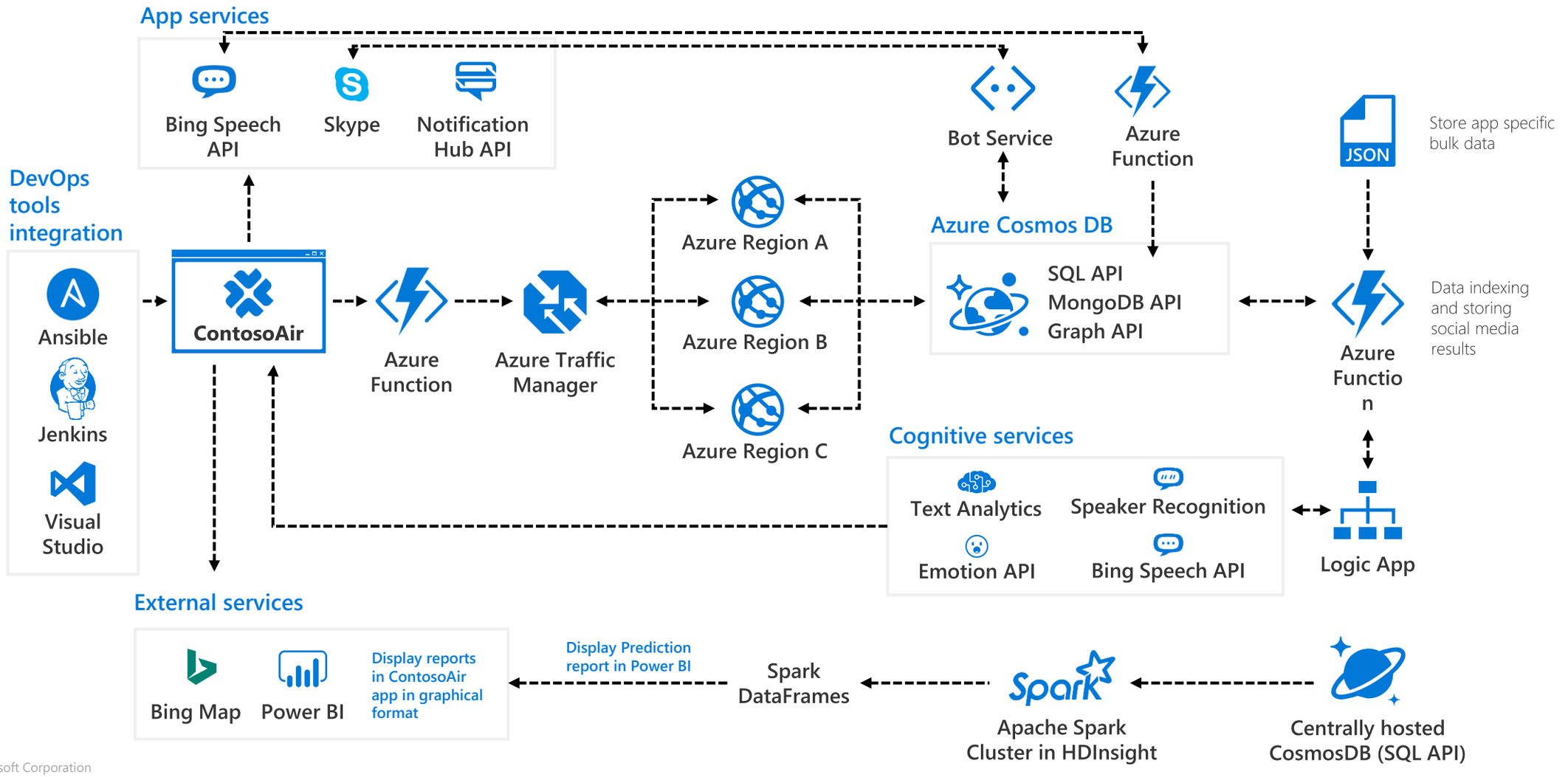
Transform sentiment into real-time insights with Cognitive Services



Solutions

- Bing Speech API converts customer voice data into written feedback
- Emotion API generates ratings based on customer facial expressions
- Text Analytics APIs codifies written customer feedback into ratings

ContosoAir provides real-time personalization around the world



PRINCIPALES 10 MOTIVOS POR LOS QUE LOS CLIENTES UTILIZAN AZURE COSMOS DB

La 1ª y única base de datos con capacidad "turnkey" de distribución global

Una base de datos con almacenamiento masivo/escalabilidad de rendimiento

Proporciona una latencia de milisegundos de un solo dígito en el percentil 99º en todo el mundo

Es compatible de forma nativa con diferentes tipos de datos a escala masiva

Ofrece 5 modelos de consistencia bien definidos para elegir la compensación adecuada de consistencia/latencia/rendimiento

Habilita aplicaciones inteligentes de misión crítica

Proporciona gran flexibilidad para optimizar la velocidad y el costo

Aborda las cargas de trabajo de big data con alta disponibilidad y confiabilidad

Proporciona seguridad multi-inquilino de nivel empresarial

Naturalmente lista para análisis y perfecta para arquitecturas orientadas a eventos

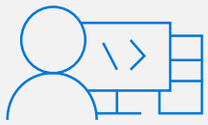
ACCIONANDO SOLUCIONES GLOBALES

Azure Cosmos DB fue construida para dar soporte a los patrones de aplicaciones y casos de uso modernos.

Permite a las organizaciones líderes de la industria liberar el valor de los datos, y responder a los clientes globales y la dinámica cambiante de negocios en tiempo real.

Datos distribuidos y disponibles en todo el mundo

Ponga sus datos donde están sus usuarios



Genere experiencias de cliente en tiempo real

Permita la personalización sensible a la latencia, la licitación y la detección de fraudes.

Ideal para juegos, IoT y comercio electrónico

Servicio rápido y predecible, incluso durante los picos de tráfico

Desarrollo simplificado con arquitectura sin servidor

Micro-servicios completamente administrados y conducidos por eventos con potencia de cómputo elástica

Ejecución de análisis de Spark sobre datos operacionales

Acelere la obtención de conocimiento a partir de datos rápidos y globales

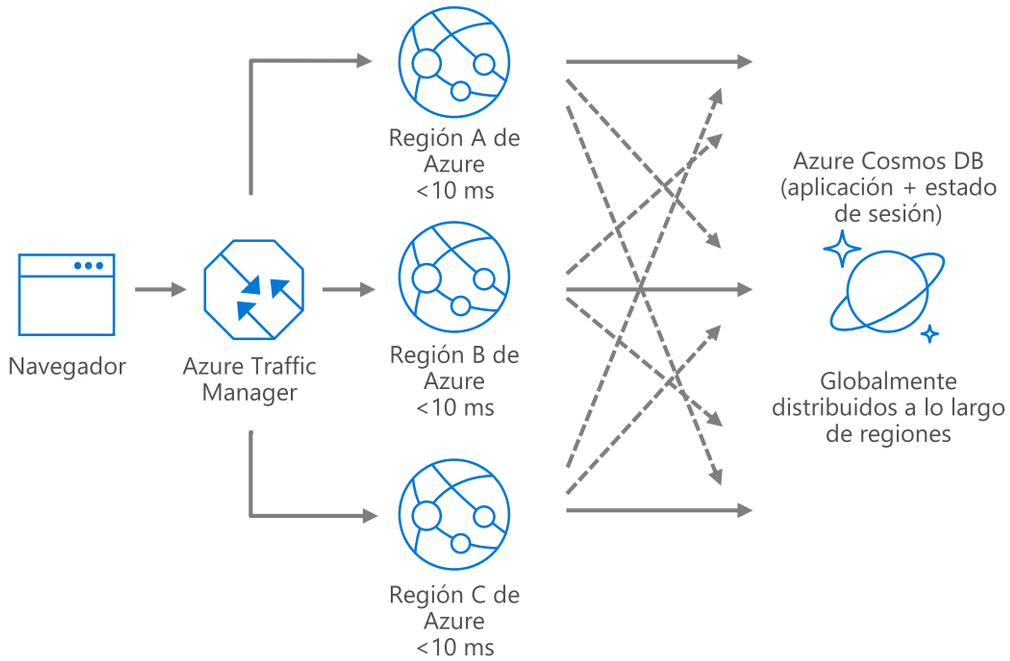
Datos NoSQL de levantar y desplazar

Cargas de trabajo de levantar y desplazar de MongoDB y de Cassandra

DATOS DISTRIBUIDOS Y DISPONIBLES EN TODO EL MUNDO

Ponga sus datos donde están sus usuarios para brindar acceso en tiempo real y servicio ininterrumpido a los clientes en cualquier parte del mundo.

- Replicación de datos global "turnkey" en todas las regiones de Azure
- Experiencia de baja latencia garantizada para usuarios globales
- Resiliencia para alta disponibilidad y recuperación ante desastres



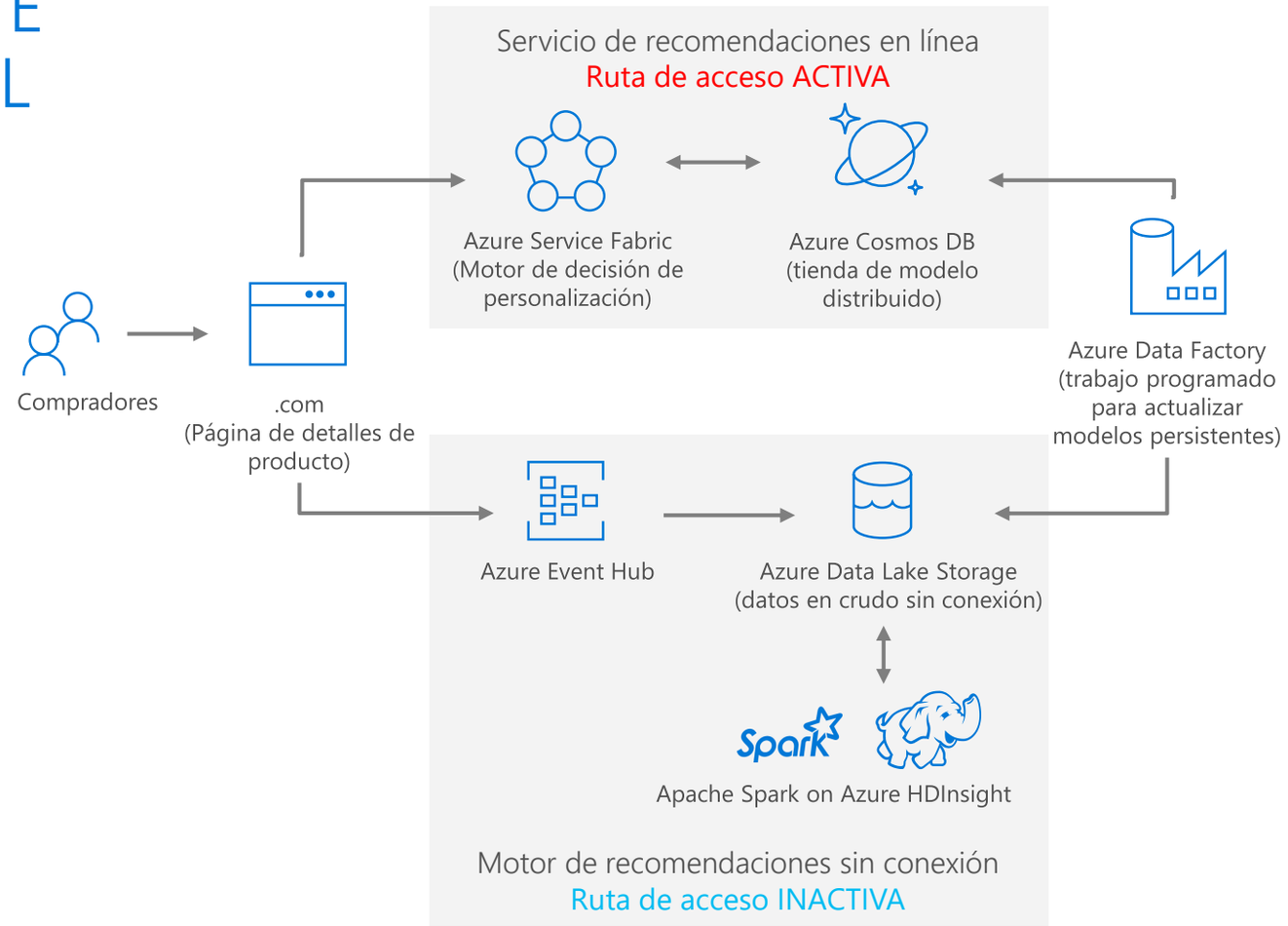
NEXT GAMES

Domino's

GENERE EXPERIENCIAS DE CLIENTE EN TIEMPO REAL

Ofrece aplicaciones sensibles a la latencia con personalización, licitación y detección de fraudes.

- Los modelos de aprendizaje de máquina generan recomendaciones en tiempo real a través de catálogos de productos
- Análisis de productos en milisegundos
- La baja latencia garantiza un alto rendimiento de las aplicaciones en todo el mundo
- Modelos de consistencia sintonizables para obtener conocimientos rápidos

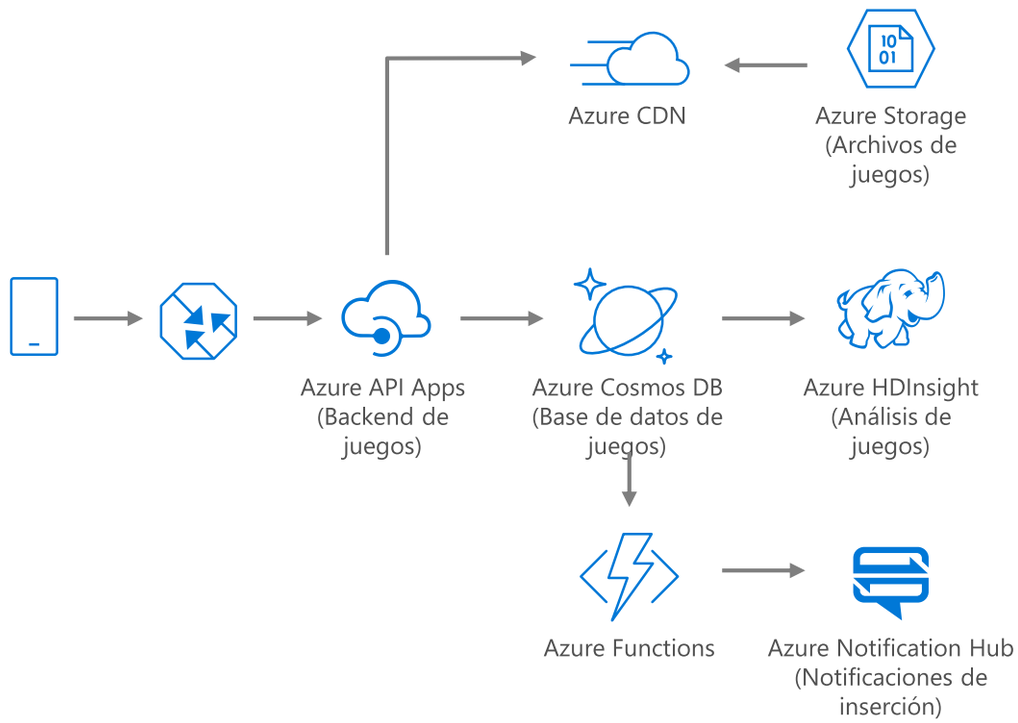


asos

IDEAL PARA JUEGOS Y COMERCIO ELECTRÓNICO

Mantener la calidad del servicio durante períodos de alto tráfico que requieren escalabilidad y rendimiento masivos.

- La escalación instantánea y elástica maneja las ráfagas de tráfico
- Experiencia de usuario global ininterrumpida
- Acceso y procesamiento de datos de baja latencia para bases de usuarios grandes y cambiantes
- Alta disponibilidad a lo largo de múltiples centros de datos

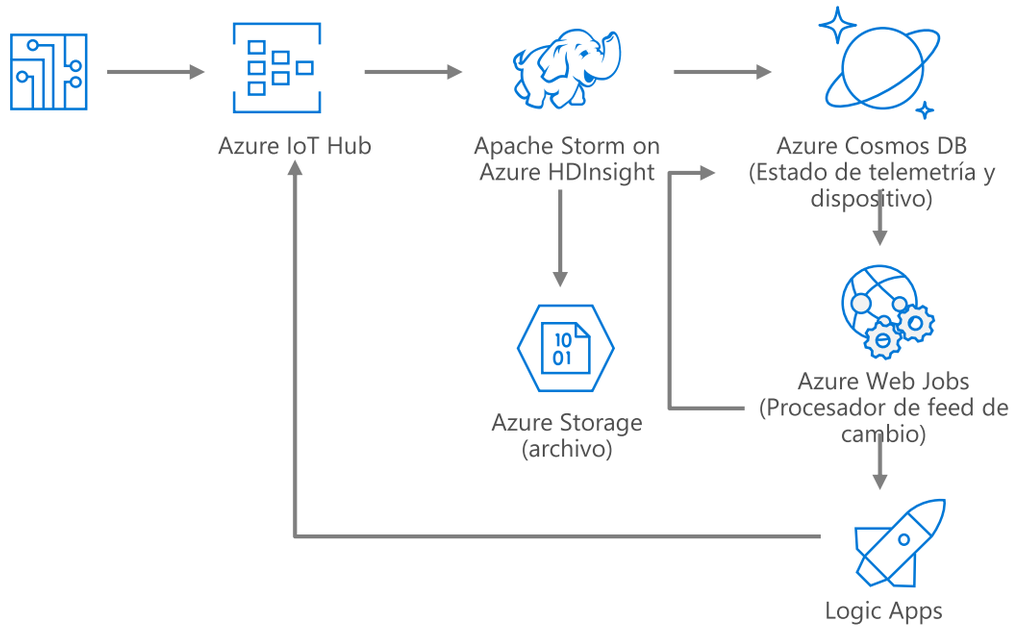


NEXT GAMES Domino's jet XBOX LIVE

TIENDAS DE TELEMETRÍA DE ESCALA MASIVA PARA IOT

Las cargas de trabajo de sensor de IoT que son diversas e imprevisibles requieren una plataforma de datos que responda adecuadamente

- Manipulación perfecta de cualquier salida o volumen de datos
- Los datos se ponen a disposición de manera inmediata y son automáticamente indexados
- Alto número de escrituras por segundo, con rendimiento estable de ingesta y consultas

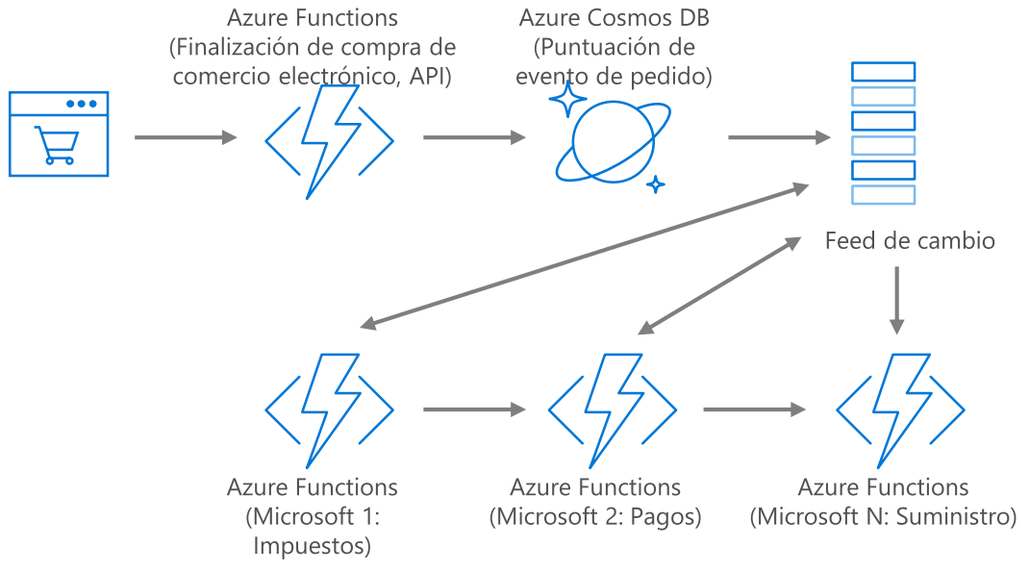


TOYOTA Honeywell LG CNS Johnson Controls

DESARROLLO SIMPLIFICADO CON ARQUITECTURA SIN SERVIDOR

Experimente una disminución en el tiempo de implementación, una mejor escalabilidad y la libertad de la administración de marco de trabajo con micro-servicios orientados hacia eventos.

- Manipulación perfecta de cualquier salida o volumen de datos
- Los datos se ponen a disposición de manera inmediata y son automáticamente indexados
- Alto número de escrituras por segundo, con rendimiento estable de ingesta y consultas
- Feeds de cambio resilientes y en tiempo real siempre conectados y accesible
- Integración nativa con Azure Functions

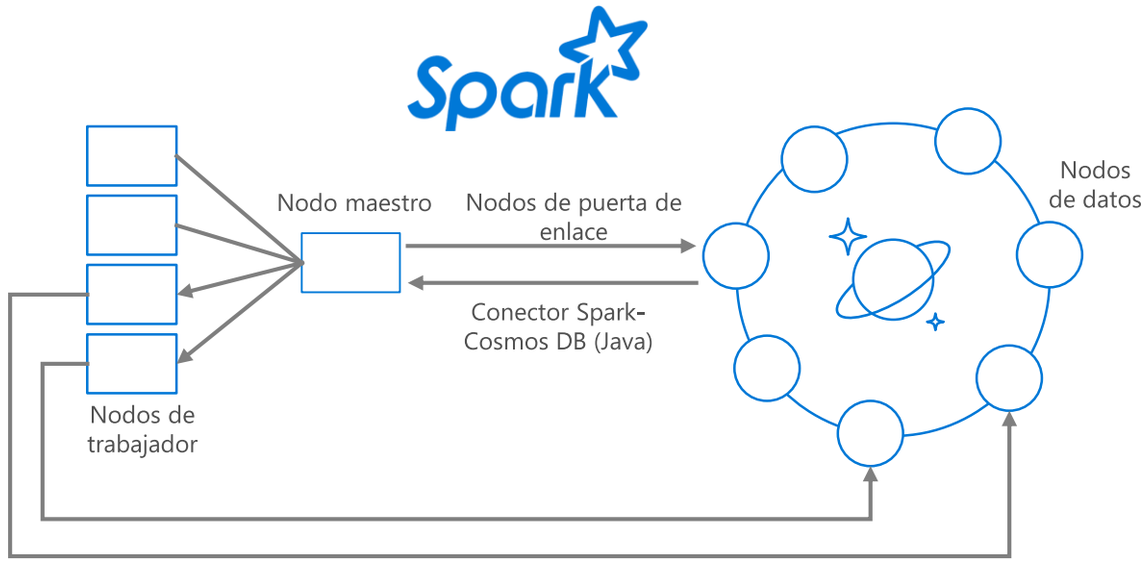


asos jet

EJECUCIÓN DE SPARK SOBRE DATOS OPERACIONALES

Acelerar el análisis de datos globales rápidamente cambiantes y de alto volumen.

- Procesamiento de big data en tiempo real a través de cualquier modelo de datos
- Aprendizaje automático a escala sobre datos distribuidos mundialmente
- Acelera las consultas analíticas con indexación automática y filtrado de empujar hacia abajo predicados
- Integración nativa con Spark Connector



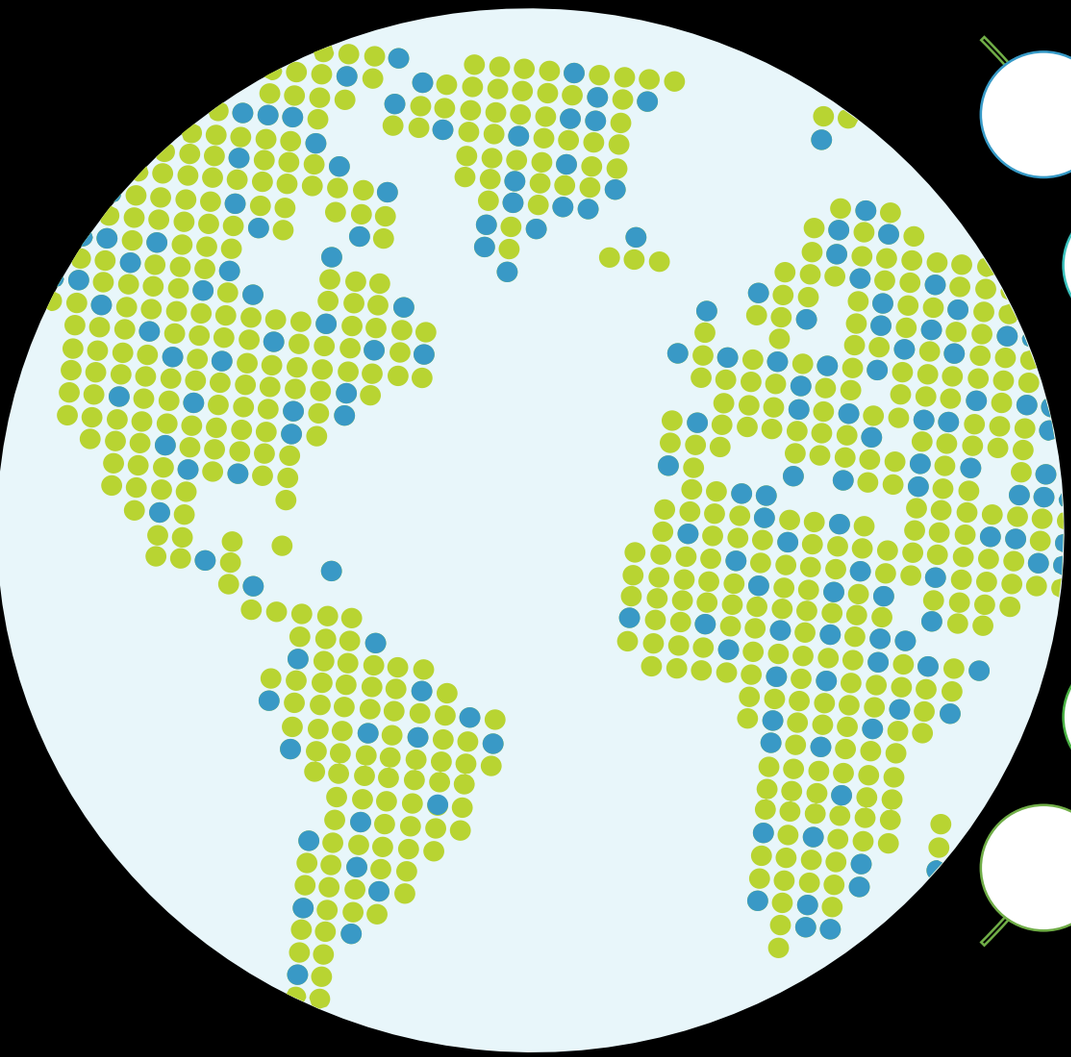
APLICACIONES NOSQL DE LEVANTAR Y DESPLAZAR

Facilite la modernización de datos con una migración de levantar y desplazar perfecta de cargas de trabajo de NoSQL hacia la nube.

- Las APIs de Azure Cosmos DB APIs para MongoDB y Cassandra traen datos de aplicaciones desde cualquier lugar hacia Azure Cosmos DB
- Aproveche las herramientas, controladores y bibliotecas existentes y continúe utilizando los SDKs de las aplicaciones existentes
- Geo-replicación "turnkey"
- No requiere infraestructura o administración de VM

Azure Cosmos DB

Cloud-born database for modern apps



- Operational database =
- Analytics database =
- Hot Updatable Data Lake =
- Database for Serverless =
- Database for AI =
- Database for IoT/Time-series data =

Microsoft

Break

Microsoft

Customer use cases

Azure Cosmos DB Use Cases

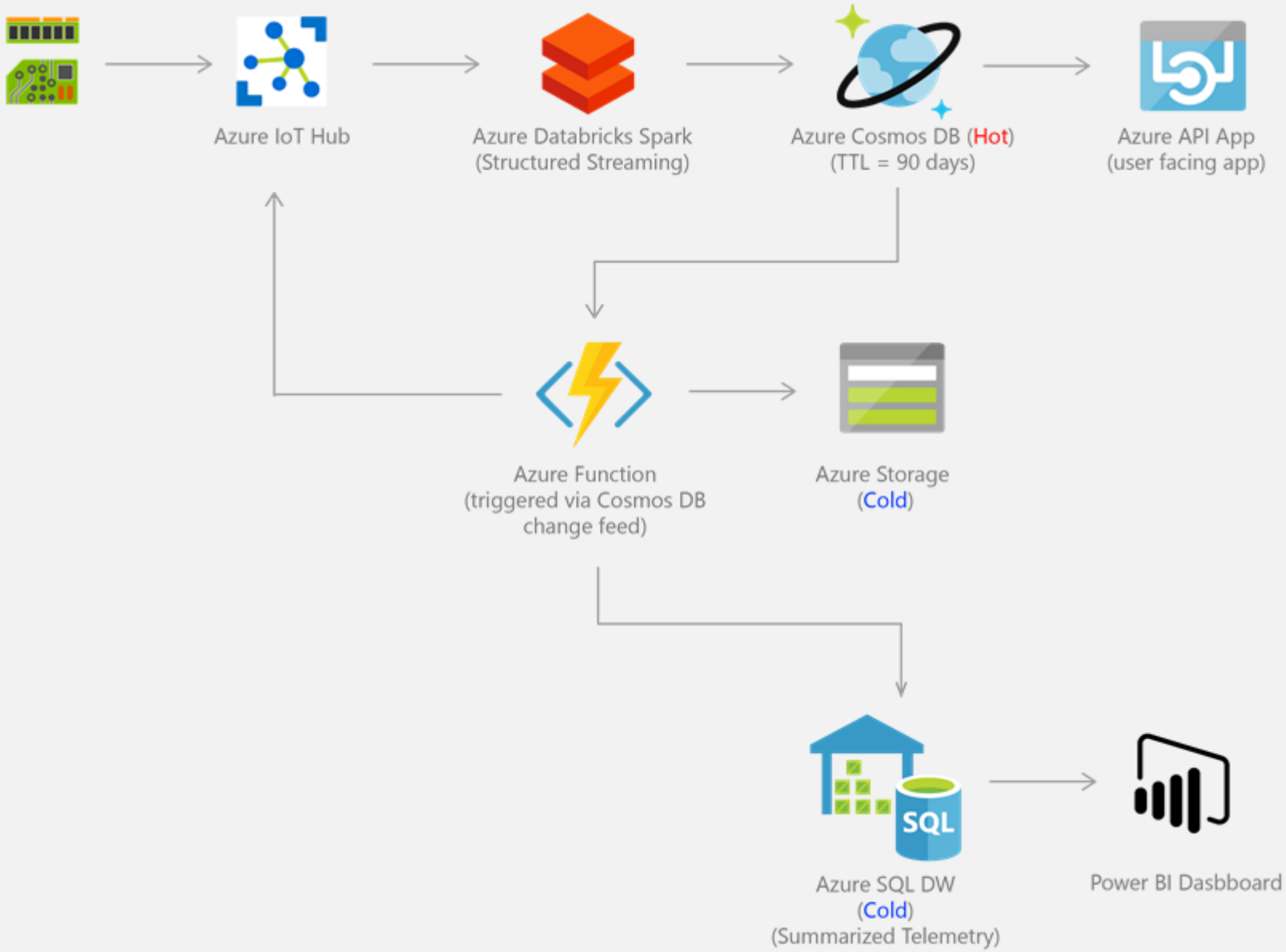
There are several use cases for Cosmos DB.

- Modern Applications
- Data migration
  - SQL => Cosmos DB
  - NoSQL => Cosmos DB
- Big Data
- Data Analytics
- IoT
- Games
- Mission Critical

XBOX LIVE

HALO 5
GUARDIANS

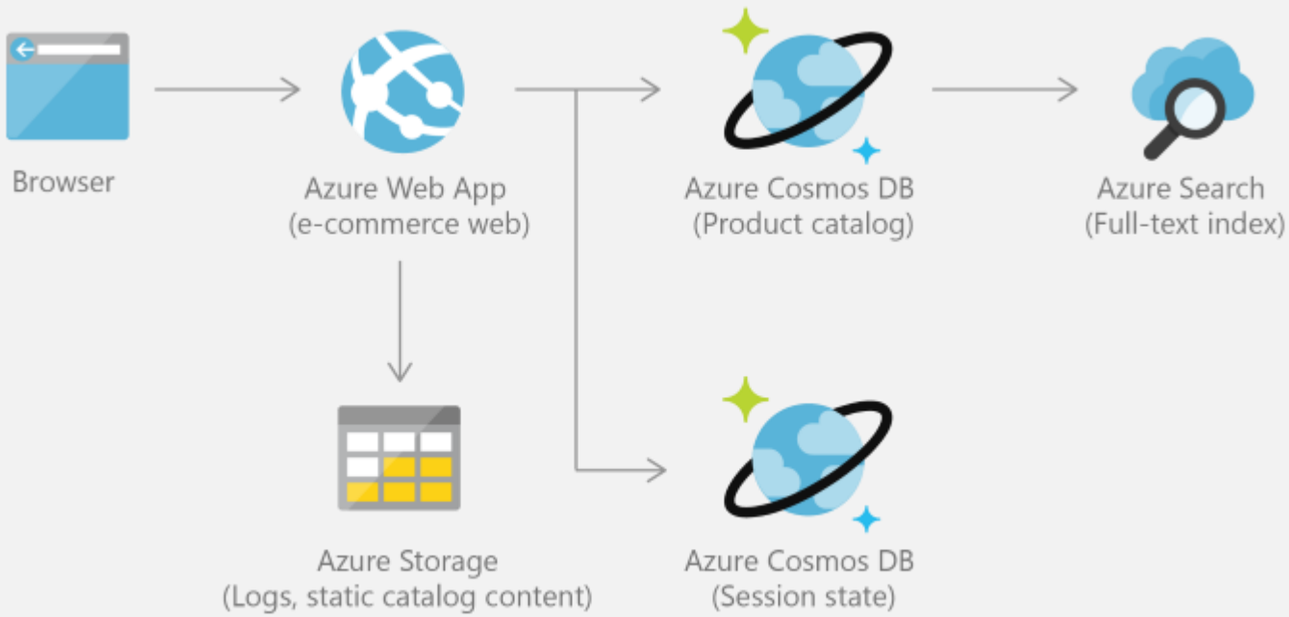
IoT and telematics



T · · Systems ·

targetbase®

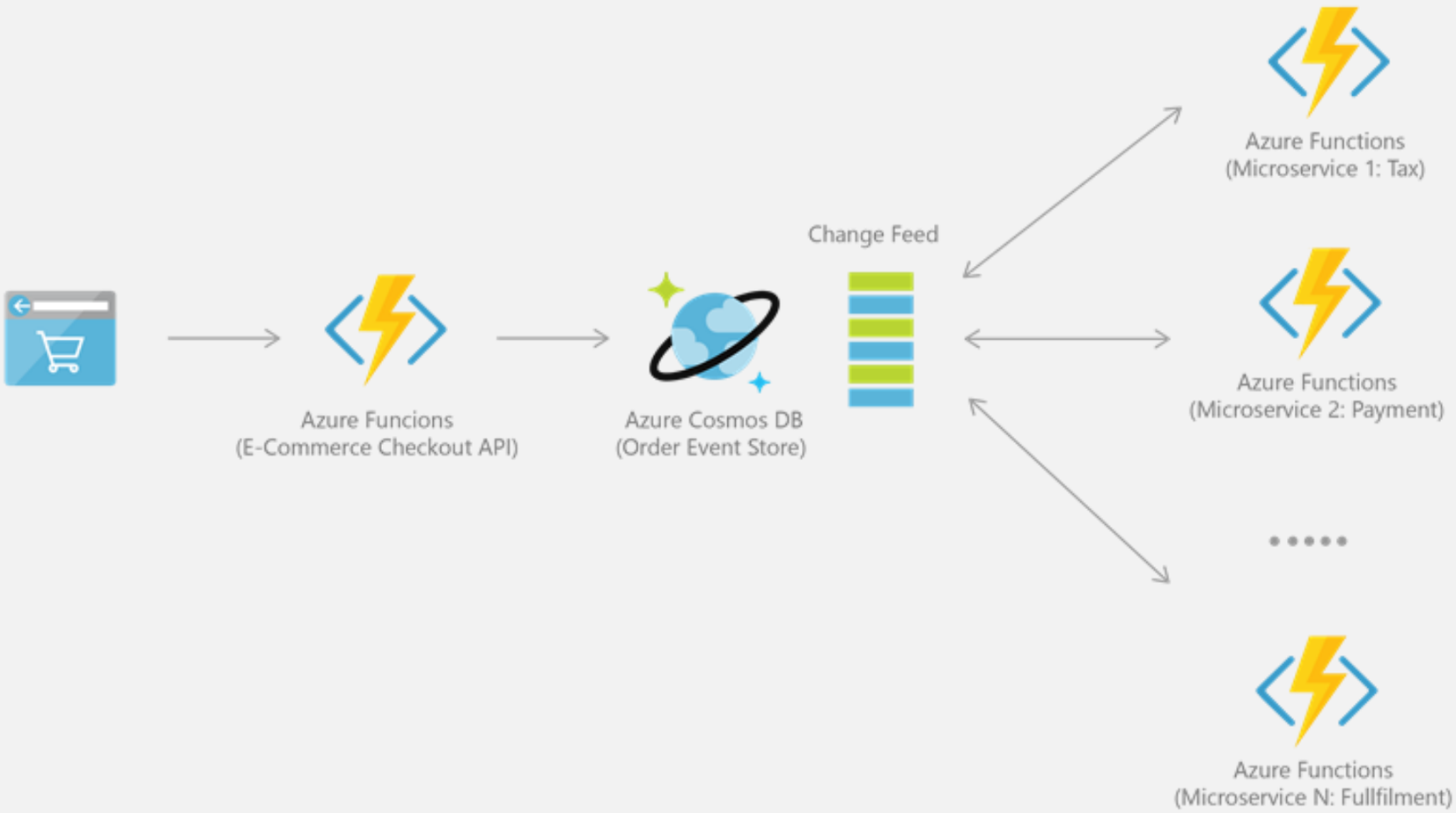
Retail and marketing



asos
discover fashion online

mapwize

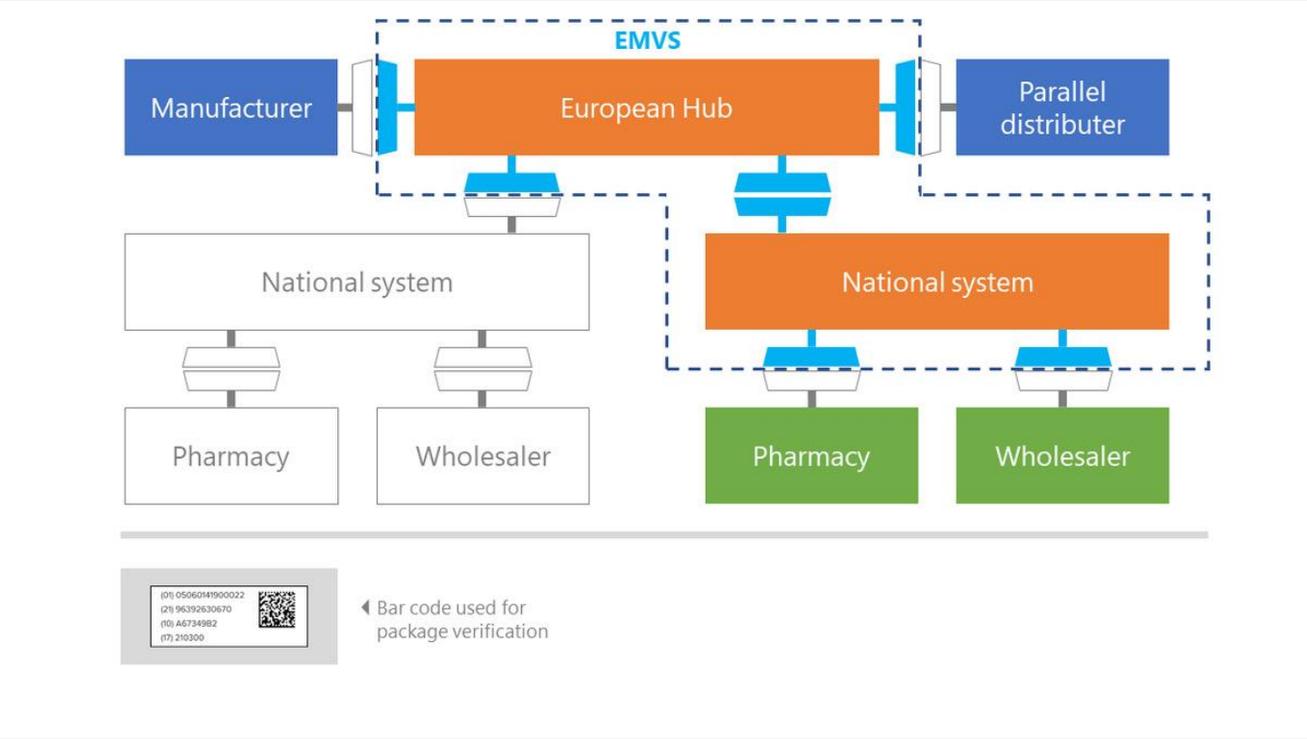
Retail and marketing



jet

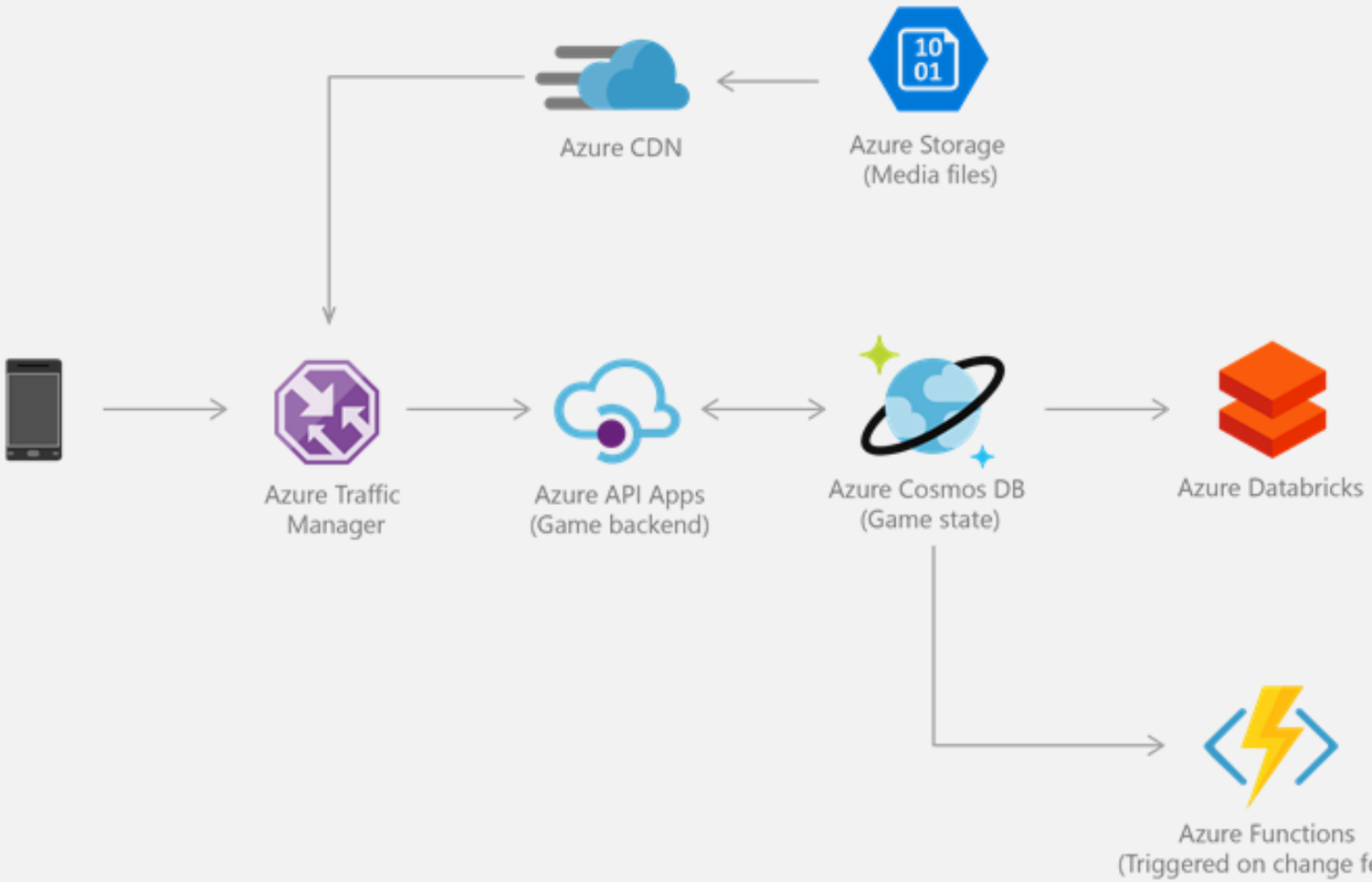
AFP HABITAT
Afiliados a una vida mejor

Retail and marketing



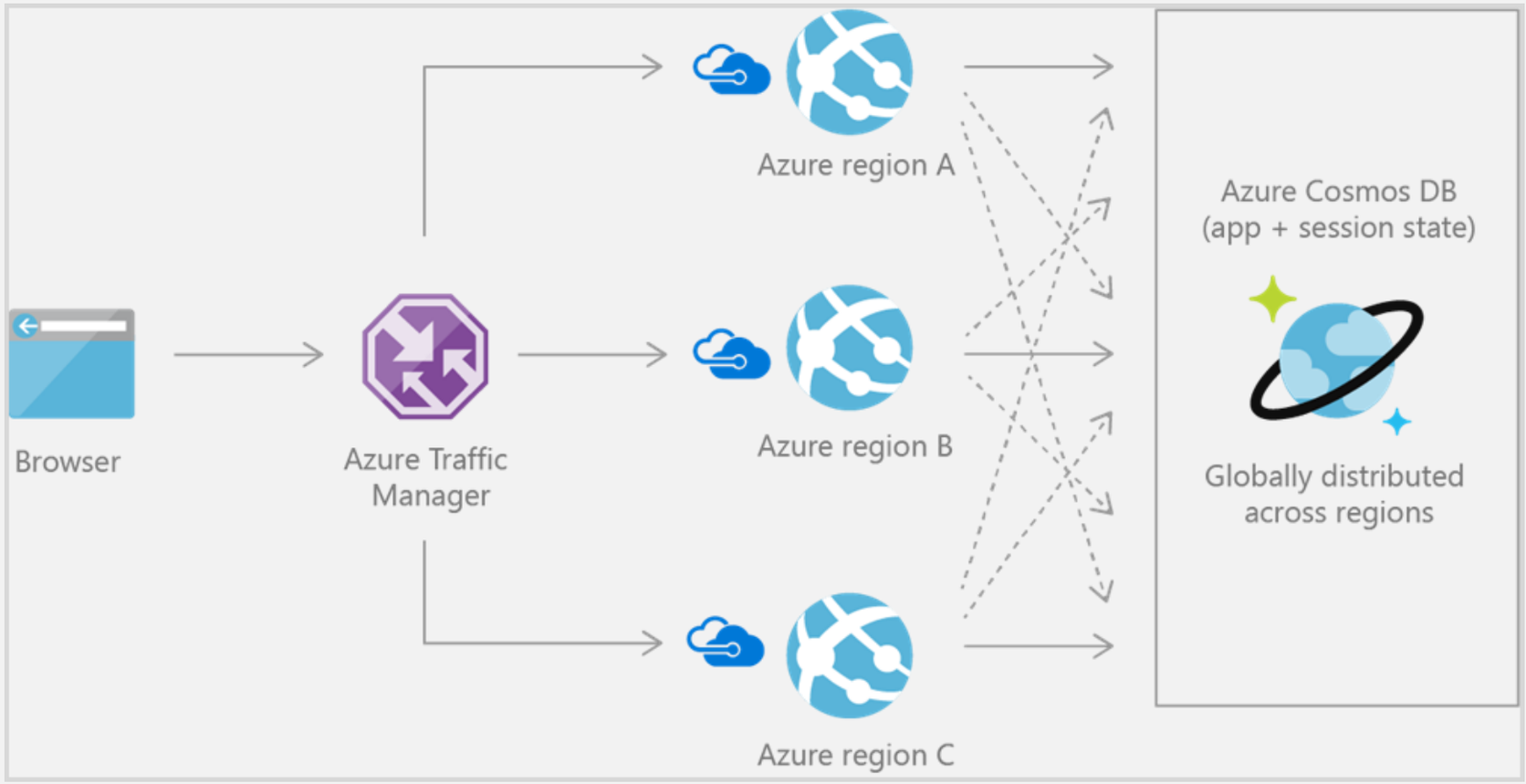
The EMVS uses Azure Cosmos DB to store data on an array of SSD disks, ensuring that the service is scalable, robust, and performant.
The impact of data loss in the EMVS is potentially severe. Although it is possible to recreate data by asking manufacturers to upload it again to the European Hub, this approach is expensive and disruptive. Even worse, if pack data is lost together with audit and transaction logs, the correct pack states cannot be recreated, and citizens could be put at risk.

Gaming



HALO 5
GUARDIANS

Web and mobile applications



SITEPRO

Flick 2 Know

NAVITIME

Mogile Tech

Microsoft

Lab 1 – Deploying CosmosDB

Request Units & Billing

Cosmos DB Total Cost of Ownership (TCO)

Azure Cosmos DB: Paper about the TCO of running a NoSQL database is available

November 19, 2016

Azure Cosmos DB Principal Engineering Manager Kirill Gavrylyuk recently published the paper The Total Cost of (Non) Ownership of a NoSQL Database Cloud Service. The paper compares the total cost of ownership (TCO) of running a NoSQL database in the following scenarios:

- An OSS NoSQL database like Cassandra or MongoDB hosted on-premises
- An OSS NoSQL database hosted on Azure Virtual Machines
- A managed NoSQL database as a service, such as Azure Cosmos DB

The paper uses the same scenarios, parameters, and assumptions used in a paper previously published by Amazon, and it yields similar conclusions. The main finding is that using a managed NoSQL cloud database is five to ten times more cost effective than the alternatives. In addition, a new finding in this paper is that Azure Cosmos DB is about 10 percent cheaper than DynamoDB due to the lower cost of write requests.

A qualitative comparison between Azure Cosmos DB, DynamoDB, and Cassandra also yielded the following findings:

- The TCO of Azure Cosmos DB is comparable to that of OSS Cassandra running on Azure D14v2 VMs for scenarios that involve high, sustained, predominantly write workloads with low storage needs.
- If more storage is needed, or the workload involves a balanced read/write mix, or the workload is bursty, the TCO of Azure Cosmos DB can be up to four times lower than that of OSS Cassandra running on Azure VMs.
- Azure Cosmos DB is up to three times cheaper than DynamoDB for the high-volume workloads examined.

Read the Azure blog post now, or jump right into the paper The Total Cost of (Non) Ownership of a NoSQL Database Cloud Service.

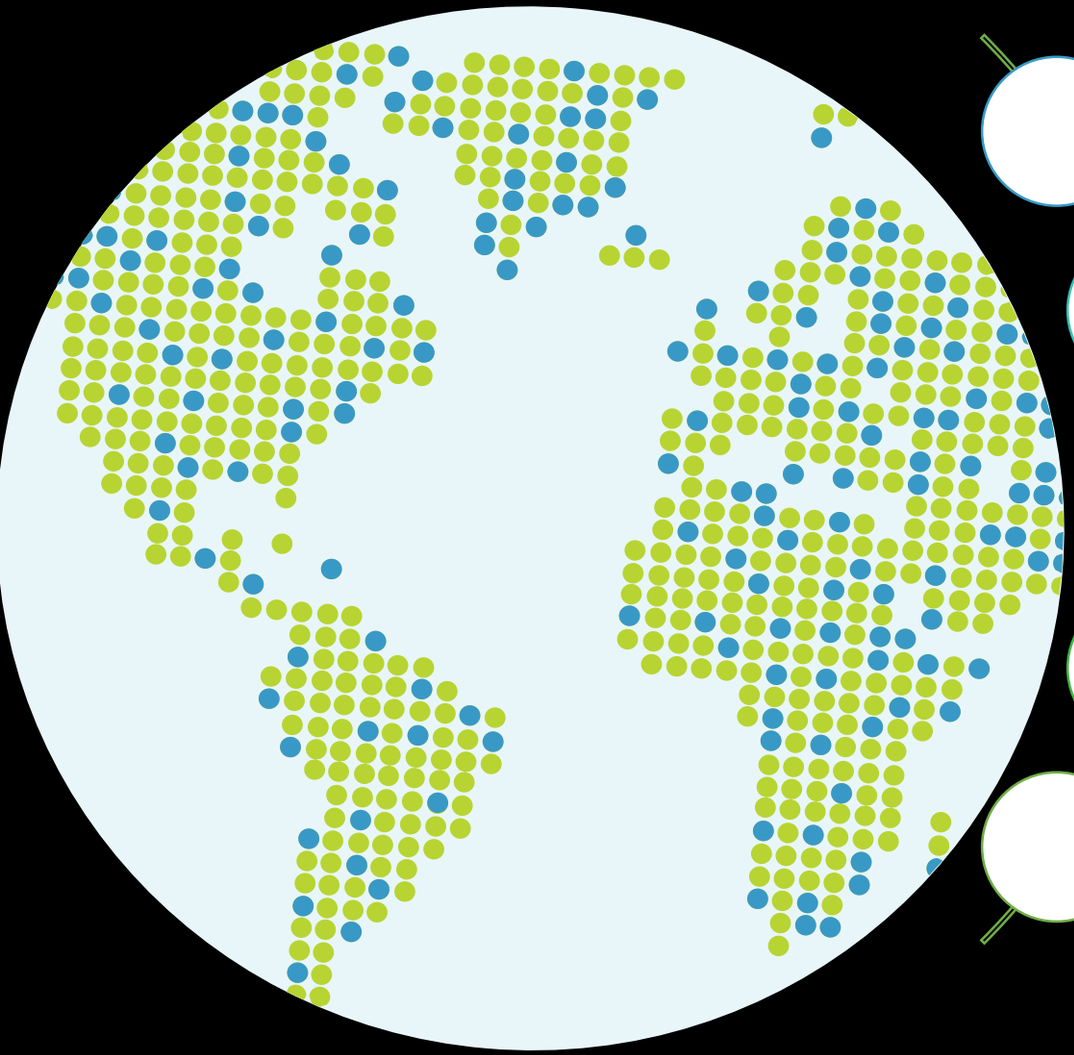
Related services

Azure Cosmos DB

Discover more Azure service updates

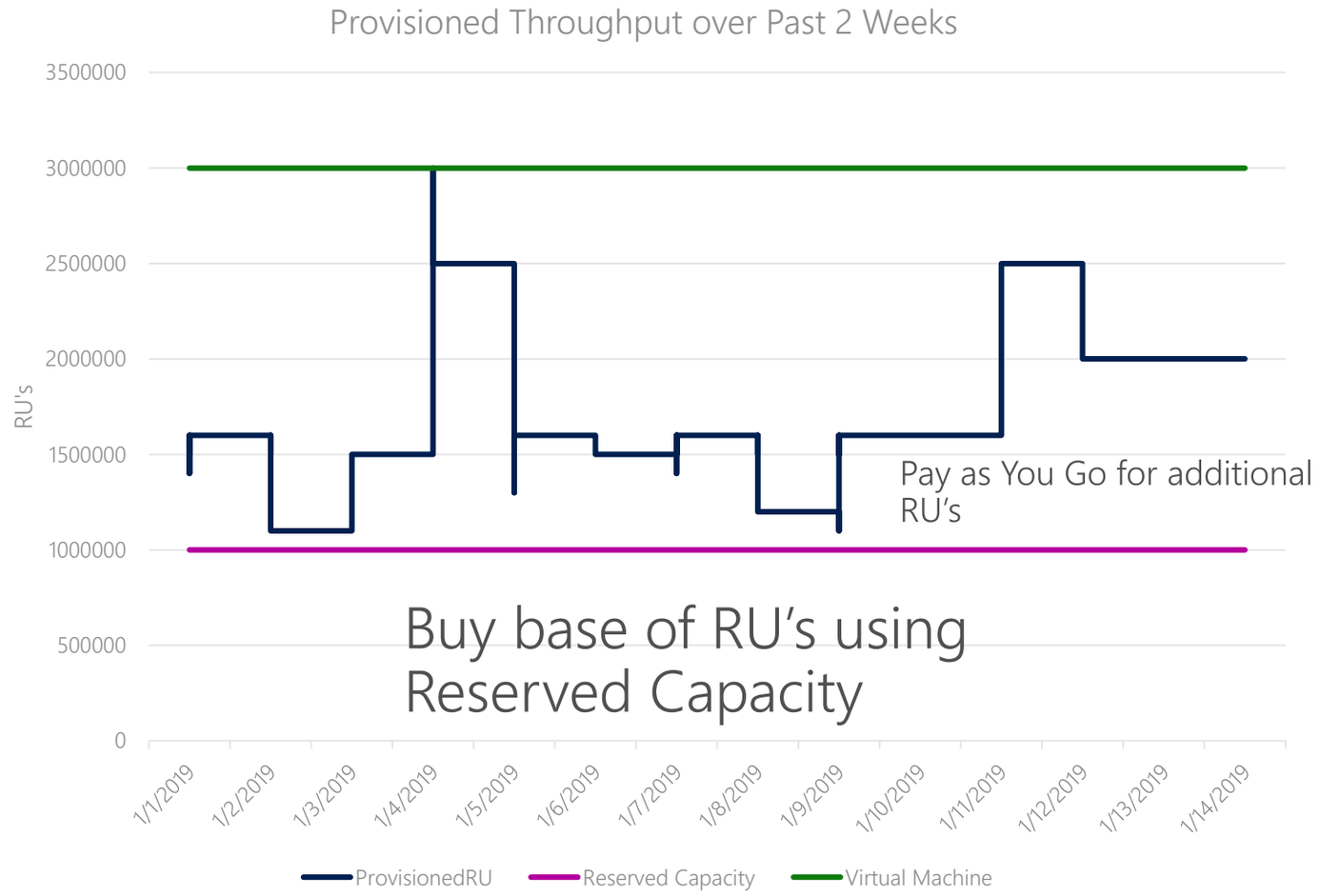
Azure Cosmos DB: TCO Factors

Deeply exploits cloud core properties and economies of scale



- Fine-grained multi-tenancy.
- No administration dev/ops required
- Superior elasticity
- Economy of Scale
- Cloud-optimized

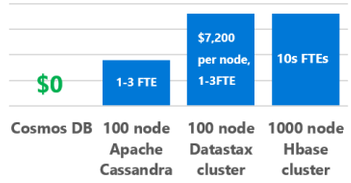
COST ADVANTAGES OF COSMOS DB OVER IAAS

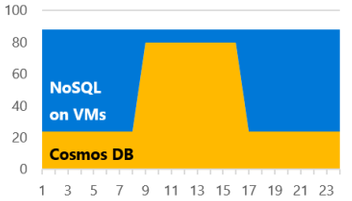


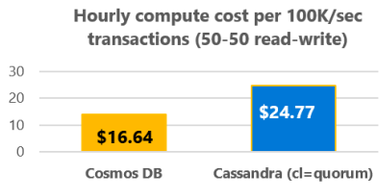
SAVE VS ON-PREM CASSANDRA

On-prem Cassandra TCO Challenges

Cassandra	Cosmos DB
High DevOps and license fees	No DevOps or license fees Up to 30% savings
No elasticity. Provision for the peak.	Instant and limitless elasticity Up to 40% savings
High datacenter and hardware maintenance costs	Economy of scale, lower cost Up to 40% savings

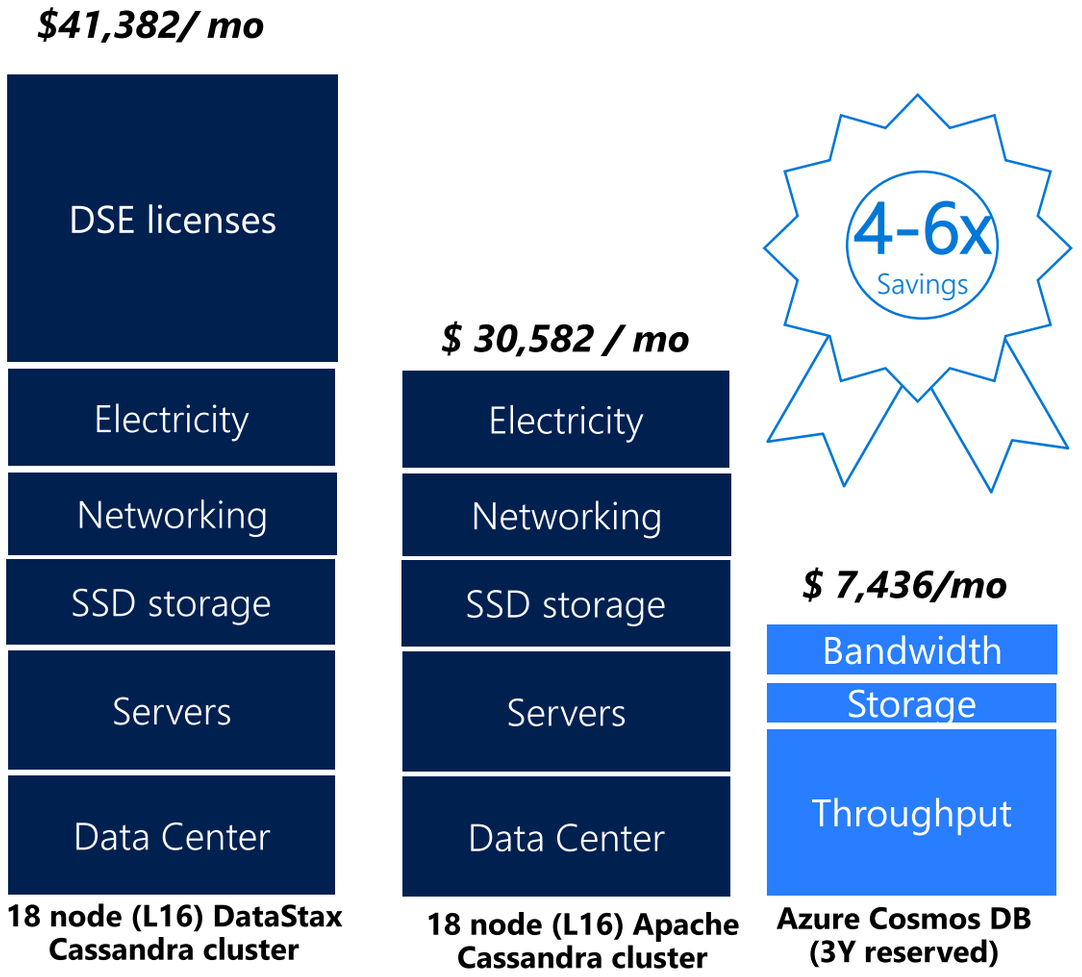






6TB workload, 80K TPS, 50-50 read-write split observed among our customers, off-peak to peak ratio is 40%, 1K objects. 4 proc, 4 cores/proc server nodes. Hardware, datacenter costs estimated using Azure TCO calculator for 5 years pro-rated monthly. DSE license assumed $7,200 / year per node (assumes 8 cores). Azure 3Y reserved price was used for Cosmos DB. Findings based on Microsoft calculations.

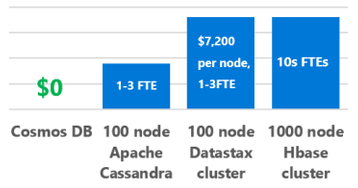
TCO Savings with Azure Cosmos DB

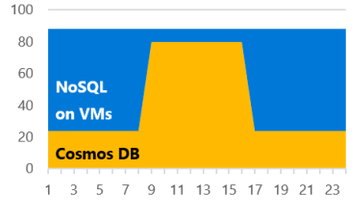


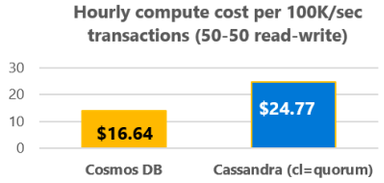
SAVE VS CASSANDRA ON IAAS

Cloud Cassandra TCO Challenges

Cassandra	Cosmos DB
High DevOps and license fees	No DevOps or license fees Up to 40% savings
Limited elasticity. Over-provisioning	Instant and limitless elasticity Up to 25% savings
Not Cloud-optimized. Poor compute, network, and storage utilization.	Azure-optimized. High compute and network utilization. Up to 40% savings

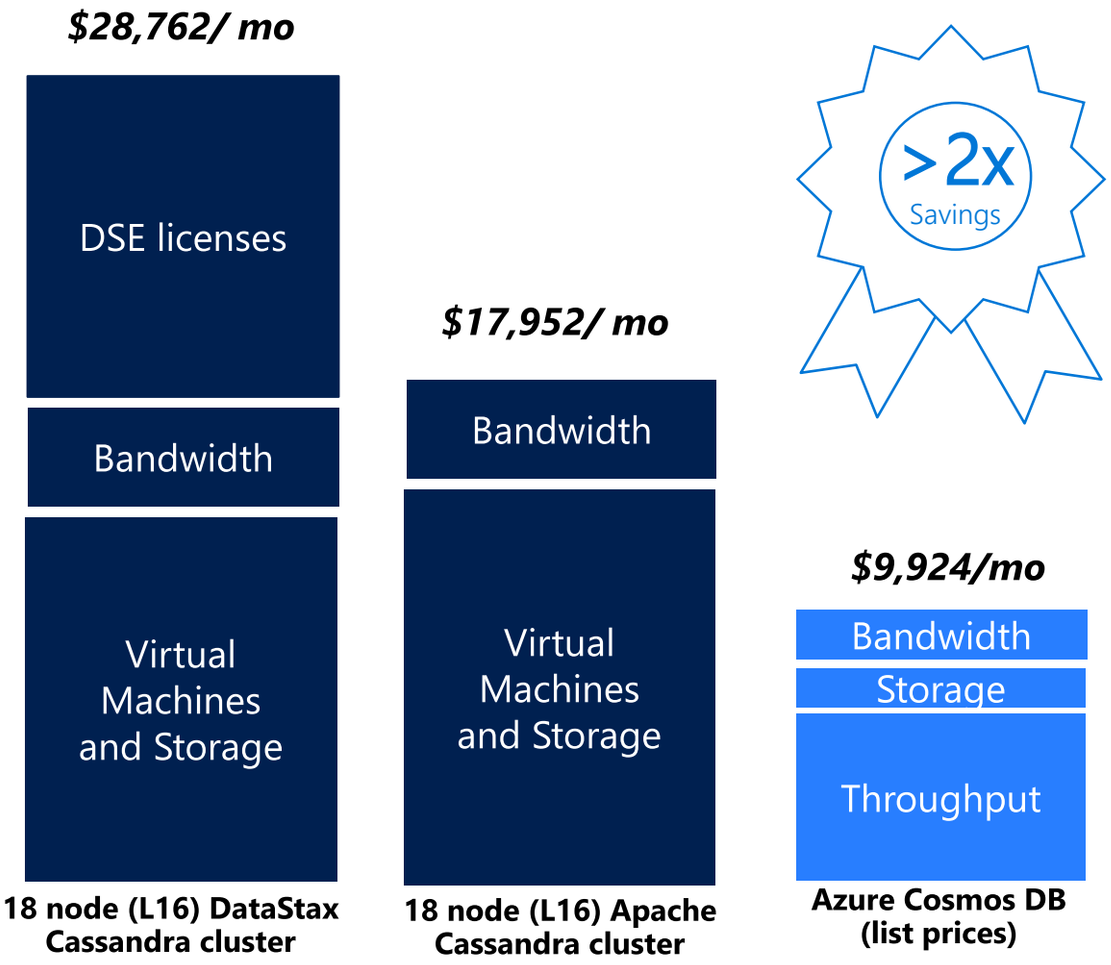






6TB workload, 80K TPS, 50-50 read-write split observed among our customers, off-peak to peak ratio is 40%, 1K objects. L16 VMs used per Azure guidance,. DSE license is $7,200 / year per node (assumes 8 cores). Azure monthly consumption prices were used. Findings based on Microsoft calculations.

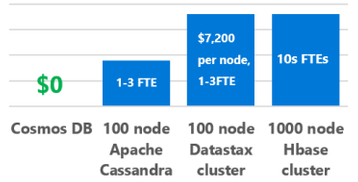
TCO Savings with Azure Cosmos DB

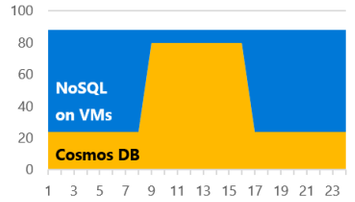


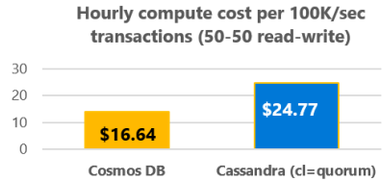
SAVE VS MONGODB ON IAAS

Cloud MongoDB TCO Challenges

MongoDB	Cosmos DB
High DevOps and license fees	No DevOps or license fees Up to 30% savings
Limited elasticity. Over-provisioning	Instant and limitless elasticity Up to 25% savings
Not Cloud-optimized. Poor compute, network, and storage utilization.	Azure-optimized. High compute and network utilization. Up to 40% savings

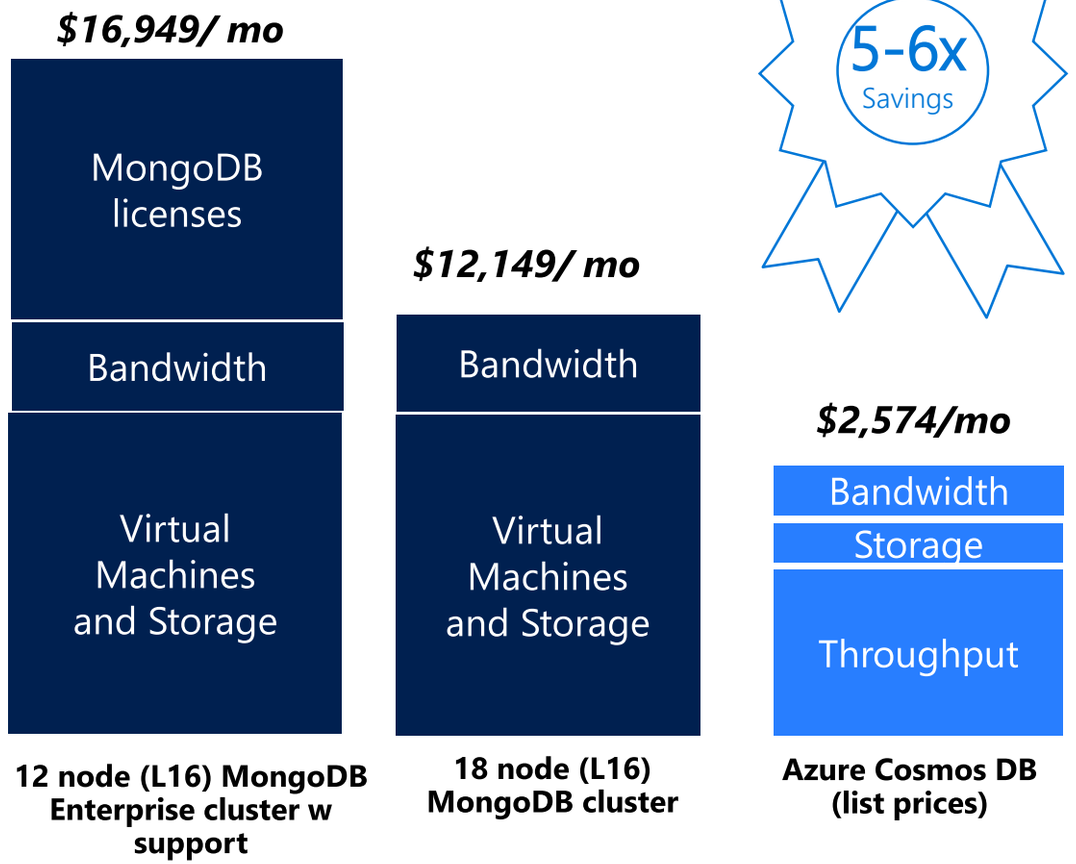






2TB workload, 20K TPS, 50-50 read-write split observed among our customers, off-peak to peak ratio is 40%, 1K objects. 4 shards, 3-way replication, L16 VMs used per Azure guidance,. MongoDB license assumed $4,800 / year per node. Azure monthly consumption prices were used. Workload is smaller than Cassandra due to observed reliability/performance challenges with MongoDB at high load. Findings based on Microsoft calculations.

TCO Savings with Azure Cosmos DB



BILLING MODEL

2 components: Storage + Throughput

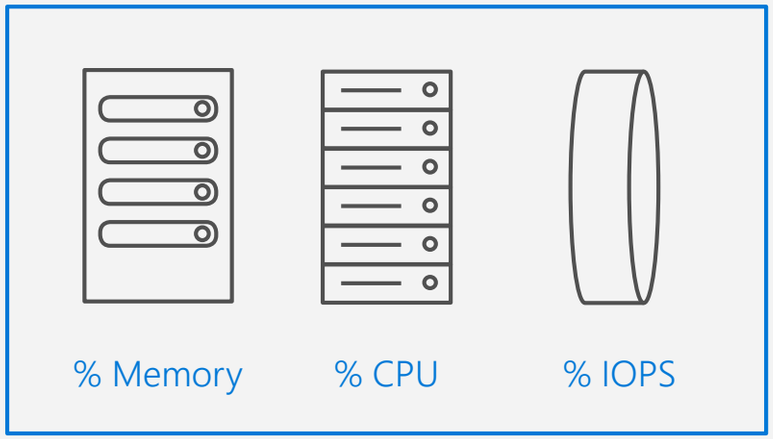
Pricing details

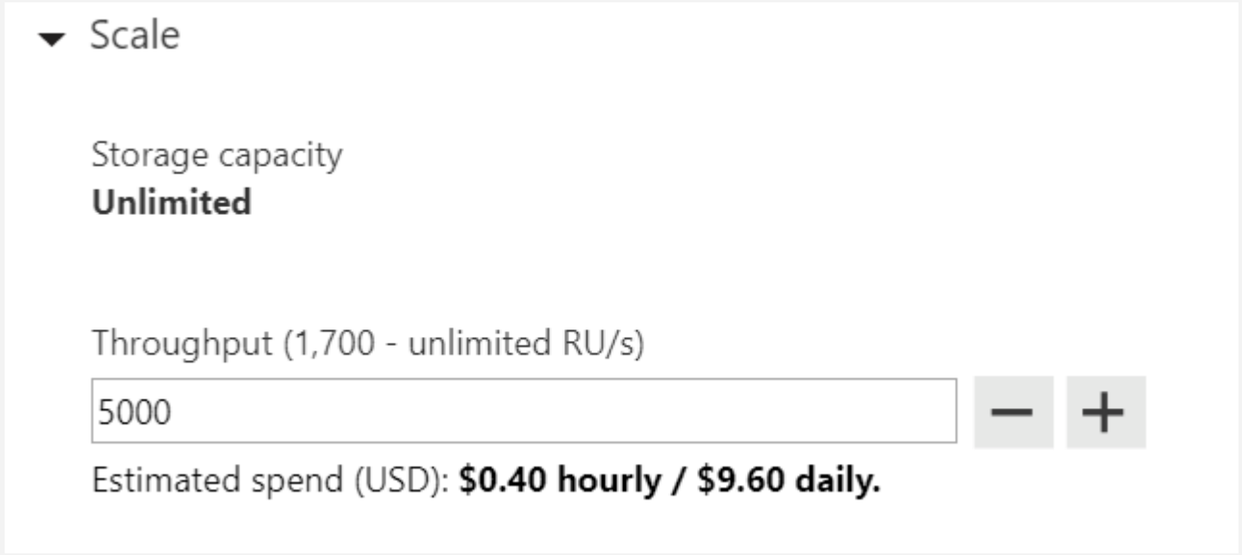
UNIT	PRICE
SSD Storage (per GB)	$0.25 GB/month
Reserved RUs/second (per 100 RUs, 400 RUs minimum)	$0.008/hour
Provisioned Throughput (multiple write regions) - 100 RU/sec	$0.016/hour

- Each collection has reserved throughput (RU/s)
- Collections in database can share throughput (400 RU/s minimum)
- Min 40 RU's per GB

REQUEST UNITS

Request Units (RUs) is a rate-based currency – e.g. 1000 RU/second





Abstracts physical resources for performing requests

REQUEST UNITS- RESERVED THROUGHPUT

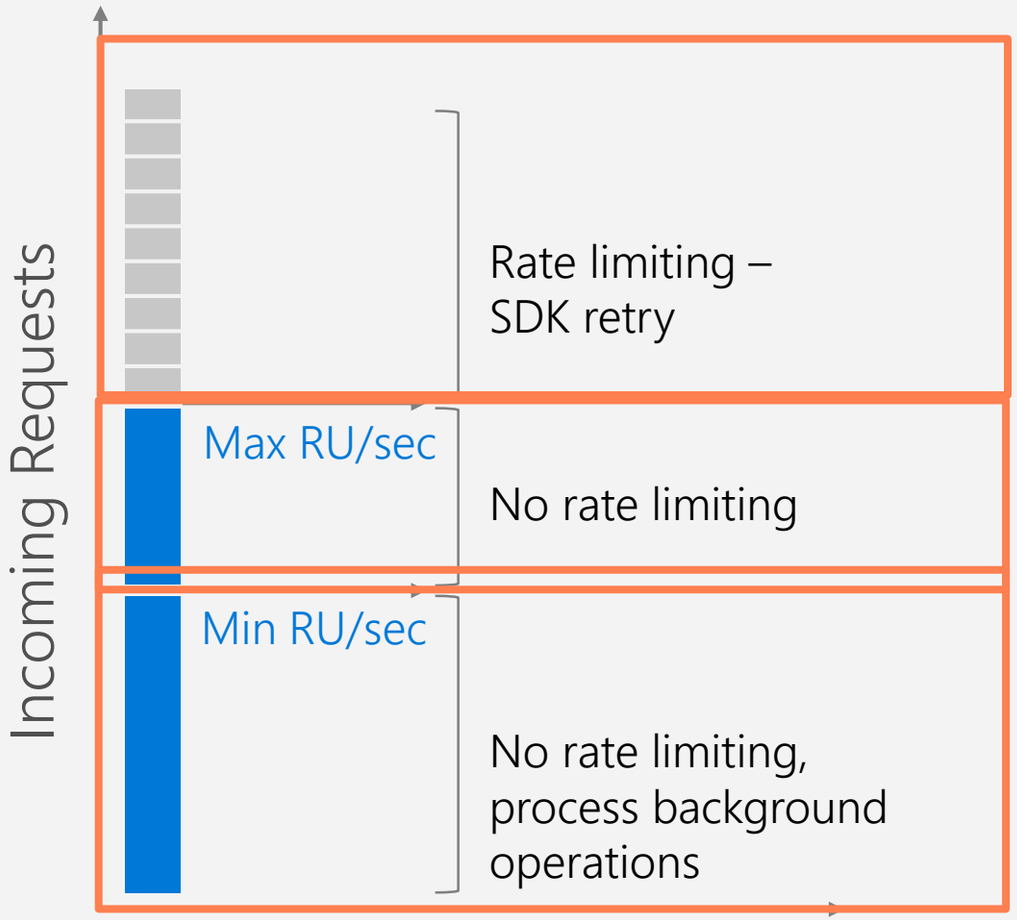
Provisioned in terms of RU/sec – e.g. 1000 RU/s

Billed for highest RU/s in 1 hour

Easy to increase and decrease on demand

Rate limiting based on amount of throughput provisioned

Background processes like TTL expiration, index transformations scheduled when quiescent



DATABASE VS COLLECTION LEVEL THROUGHPUT

Wherever possible, you should use **collection-level throughput**. This leads to predictable performance since each collection is guaranteed its provisioned RU's

Choose Database Level Throughput if you :

- Are migrating many collections and does not know how much throughput to set for each one
- Have many collections that are timeshared
- Do not want to refactor data/code to have fewer collections

ESTIMATING REQUIRED RU/S

Operation Type	# Requests per sec	# RU's per Request	RU's Needed
Write Single Document	10,000	10	100,000
Top Query #1	700	100	70,000
Top Query #2	200	100	20,000
Top Query #3	100	100	10,000
Total RU/s			200,000 RU/s

Guidance:

- **Identify query & access patterns –** e.g. top 5 queries, or # reads/writes per second
- Use '**Request Charge**' property from SDK + sample document to see # RU / operation
- POC / Load test -> Scale up, and scale down
- Scale 5-10x above the average number of RU's needed per second

DEMO: QUERY RU CONSUMPTION

Pricing Example – Do this exercise to ballpark cost + RU/s required

Storage Cost

Avg Record Size (KB)	1
Number of Records	100,000,000
Total Storage (GB)	100
Monthly Cost per GB	$0.25
Expected Monthly Cost for Storage	$25.00

Throughput Cost

Operation Type	Number of Requests per sec	Avg RU's per Request	RU's Needed
Create	100	5	500
Read	400	1	400

Total RU/sec	900
Hourly Cost per 100 RU/sec	$0.008
Monthly Cost per 100 RU/sec	$6.00
Expected Monthly Cost for Throughput	$54.00

Total Monthly Cost

[Total Monthly Cost] = [Monthly Cost for Storage] + [Monthly Cost for Throughput]
= $25 + $54
= $79 per month

* pricing may vary by region; for up-to-date pricing, see: https://azure.microsoft.com/pricing/details/cosmos-db

COSTS FOR MULTI-REGION ACCOUNTS

Single Region Write Accounts: Throughput: 10,000 RU's Storage: 1 TB

ITEM	USAGE (MONTH)	RATE	MONTHLY COST
Throughput bill for container in West US	10K RU/s * 24 * 31	$0.008 per 100 RU/s per hour	$595.20
Throughput bill for 3 additional regions - East US, North Europe, and East Asia	3 * 10K RU/s * 24 * 31	$0.008 per 100 RU/s per hour	$1,785.60
Storage bill for container in West US	1 TB	$0.25/GB	$256
Storage bill for 3 additional regions - East US, North Europe, and East Asia	3 * 1TB	$0.25/GB	$768
Total			$3,404.80

* pricing may vary by region; for up-to-date pricing, see: https://azure.microsoft.com/pricing/details/cosmos-db

COSTS FOR MULTI-REGION ACCOUNTS

Multi-Master Accounts: Throughput: 10,000 RU's Storage: 1 TB

ITEM	USAGE (MONTH)	RATE	MONTHLY COST
Throughput bill for container in West US (multiple regions write)	10K RU/s * 24 * 31	$0.016 per 100 RU/s per hour	$1,190.40
Throughput bill for 3 additional regions - East US, North Europe, and East Asia (multiple regions write)	(3 + 1) * 10K RU/s * 24 * 31	$0.016 per 100 RU/s per hour	$4,761.60
Storage bill for container in West US	1 TB	$0.25/GB	$256
Storage bill for 3 additional regions - East US, North Europe, and East Asia	3 * 1TB	$0.25/GB	$768
Total			$6,976

* pricing may vary by region; for up-to-date pricing, see: https://azure.microsoft.com/pricing/details/cosmos-db

EXAMPLE

ESTIMATING REQUIRED RU/S

Operation Type	# Requests per sec	# RU's per Request	RU's Needed
Top Query #1	9,900	10	
Top Query #2	100	10	
Total RU/s			
Total GB Stored			1,000 GB

Calculate the cost if:

a. Does not globally replicate data

b. Replicates to 1 additional region and uses multi-master

ESTIMATING REQUIRED RU/S

Operation Type	# Requests per sec	# RU's per Request	RU's Needed
Top Query #1	9,900	10	99,000
Top Query #2	100	10	1,000
Total RU/s			100,000 RU/s
Total GB Stored			1,000 GB

Calculate the cost if this customer:

a. Does not globally replicate data: 1 region and single-master

b. Replicates to 1 additional region and uses multi-master

COST CALCULATION EXAMPLE – SINGLE REGION

Single Region Write Accounts: Throughput: 100,000 RU's Storage: 1,000 GB

ITEM	USAGE (MONTH)	RATE	MONTHLY COST
Throughput bill for container in West US			
Throughput bill for additional regions			
Storage bill for container in West US			
Storage bill for additional regions			
Total			

* pricing may vary by region; for up-to-date pricing, see: https://azure.microsoft.com/pricing/details/cosmos-db

COST CALCULATION EXAMPLE – SINGLE REGION

Single Region Write Accounts: Throughput: 100,000 RU's Storage: 1,000 GB

ITEM	USAGE (MONTH)	RATE	MONTHLY COST
Throughput bill for container in West US	100K RU/s * 24 * 30	$0.008 per 100 RU/s per hour	$5,760
Throughput bill for additional regions	N/A	$0.008 per 100 RU/s per hour	0
Storage bill for container in West US	1,000 GB	$0.25/GB	$250
Storage bill for additional regions	N/A	$0.25/GB	0
Total			$6,010.00

* pricing may vary by region; for up-to-date pricing, see: https://azure.microsoft.com/pricing/details/cosmos-db

COST CALCULATION EXAMPLE – MULTI-REGION AND MULTI-MASTER

Multi-Master Accounts: Throughput: 100,000 RU's Storage: 1000GB

ITEM	USAGE (MONTH)	RATE	MONTHLY COST
Throughput bill for container in West US (multiple regions write)			
Throughput bill for 1 additional region - East US			
Storage bill for container in West US			
Storage bill for 1 additional region - East US			
Total			

* pricing may vary by region; for up-to-date pricing, see: https://azure.microsoft.com/pricing/details/cosmos-db

COST CALCULATION EXAMPLE – MULTI-REGION AND MULTI-MASTER

Multi-Master Accounts: Throughput: 100,000 RU's Storage: 1000 GB

ITEM	USAGE (MONTH)	RATE	MONTHLY COST
Throughput bill for container in West US (multiple regions write)	100K RU/s * 24 * 30	$0.016 per 100 RU/s per hour	$11,520
Throughput bill for 1 additional region - East US	(1 + 1) * 100K RU/s * 24 * 30	$0.016 per 100 RU/s per hour	$23,040
Storage bill for container in West US	1000 GB	$0.25/GB	$250
Storage bill for 1 additional region - East US	1 * 1000 GB	$0.25/GB	$250
Total			$35,060

* pricing may vary by region; for up-to-date pricing, see: https://azure.microsoft.com/pricing/details/cosmos-db

AZURE COSMOS DB RESERVED CAPACITY

Azure Cosmos DB Reserved Capacity helps you save money by pre-paying for one-year or three-years of Cosmos DB capacity allowing you to get a discount on the Cosmos DB resources, e.g., databases, containers (tables/collections/graphs) you use.

Azure Cosmos DB Reserved Capacity can significantly reduce your Cosmos DB costs—up to 65 percent on regular prices–with one-year or three-year upfront commitment.

Reserved Capacity provides a billing discount and does not affect the runtime state of your Cosmos DB resources.

	1 Year		3 Year	
Throughput	Single Region Writes	Multiple Regions Writes	Single Region Writes	Multiple Regions Writes
First 100K RU/s	20%	25%	30%	35%
Next 400K RU/s	25%	30%	35%	40%
Next 2.5M RU/s	30%	35%	45%	50%
Over 3M RU/s	45%	50%	60%	65%

Effective Discounts

Amount	1 Year – single region writes	3 Year - single region writes	1 Year - multiple region writes	3 Year - multiple region writes
20,000	15.00%	20.00%	25.00%	30.00%
50,000	15.00%	20.00%	25.00%	30.00%
100,000	20.00%	30.00%	25.00%	35.00%
500,000	24.00%	34.00%	29.00%	39.00%
1,000,000	27.00%	39.50%	32.00%	44.50%
2,000,000	28.50%	42.25%	33.50%	47.25%
3,000,000	29.00%	43.17%	34.00%	48.17%
5,000,000	35.40%	49.90%	40.40%	54.90%
10,000,000	40.20%	54.95%	45.20%	59.95%
20,000,000	42.60%	57.48%	47.60%	62.48%
30,000,000	43.40%	58.32%	48.40%	63.32%

RESERVED CAPACITY

Replaces EA discount (or other discounts)

Automatically applied across different accounts, regions, etc.

You pre-pay for Reserved Capacity

Single-master and multi-master accounts require separate reservations

Amount	1 Year – single region writes	3 Year - single region writes	1 Year - multiple region writes	3 Year - multiple region writes
20,000	15.00%	20.00%	25.00%	30.00%
50,000	15.00%	20.00%	25.00%	30.00%
100,000	20.00%	30.00%	25.00%	35.00%
500,000	24.00%	34.00%	29.00%	39.00%
1,000,000	27.00%	39.50%	32.00%	44.50%
2,000,000	28.50%	42.25%	33.50%	47.25%
3,000,000	29.00%	43.17%	34.00%	48.17%
5,000,000	35.40%	49.90%	40.40%	54.90%
10,000,000	40.20%	54.95%	45.20%	59.95%
20,000,000	42.60%	57.48%	47.60%	62.48%
30,000,000	43.40%	58.32%	48.40%	63.32%

INDEXING IN COSMOS DB

Cosmos DB uses an inverted index and allows you to search your database for any item regardless of schema structure

Indexing changes do not affect write availability or provisioned throughput but do *consume* RU's

Removing indexing on some fields can reduce both RU's used in writing and storage costs

INDEXING IN COSMOS DB

All fields indexed by default

You can:

- Remove index
- Change index to geo-spatial
- Change index to range (default)

INDEXING POLICY

Default – Index all fields with range indexing with max precision

Set at Collection Level

Indexing Mode:

- Consistent – updated synchronously on writes (recommended)
- None – no indexing

Automatic – True or False

Index kind – Range, spatial

Included Paths

Excluded Paths

BEST PRACTICES

Understand query patterns – which properties are being used?

Optimize for your most common operations

OPTIMIZING COSMOS DB COSTS

Costs can best be reduced by:

- Good data modeling choices
- Provisioning the optimal number of request unit and scaling up/down as needed
- Adjusting indexing policy
- Scaling Cosmos DB requests across different regions
- Purchasing Reserved Capacity

COMMON MISTAKES

COMMON MISTAKES

To ensure optimal CosmosDB performance and cost, you should ensure to:

- Make good data modeling choices
- Provision the optimal number of request unit and scaling up/down as needed
- Customize indexing policy
- Scale Cosmos DB requests across different regions
- Purchase Reserved Capacity

INEFFICIENT DATA MODELING DESIGN CHOICES

Symptoms:

- Many collections where entities are separated by collection (treated like relational database tables)
  - Ensure to take advantage of that fact that Cosmos DB is a NoSQL database and can have different types of documents in a single collection
- Misconception that Cosmos DB does not have cross-collection or cross-document joins
  - Ensure to take advantage of that fact that Cosmos DB is a NoSQL database and can have different types of documents in a single collection
- Excessively large documents (> 500 KB)
  - Writes and updates will consume a large amount of RU's
- Slow performance and throttled requests (even after increasing provisioned RU amount)
  - Check for "hot partitions" and pick a partition key that balances requests and data stored

Actions needed:

- 1-2 hour data modeling workshop should resolve most of these issues. Reach out to your account team or partner team to schedule a workshop.

INCORRECT PROVISIONING OPTIMAL NUMBER OF RU'S

Symptoms:

- Cosmos DB is slower than expected
  - You might be using default or minimum settings for RU's. If consumed RU's > provisioned RU's, we will throttle requests, leading to slower performance. You can view consumed vs provisioned RU's in the Azure portal.
- High Cosmos DB bill
  - You can probably optimize the number of provisioned RU's or scale it up/down on an hourly level. You can view consumed vs provisioned RU's in the Azure portal.

Actions needed:

- Estimate the right number of RU's given top writes/reads/queries RU consumption
- Scale RU's up/down – remember that you are only billed for the peak that they use in a given hour

INDEXING POLICY NOT OPTIMIZED

Symptoms:

- Writes use a large number of RU's
  - If documents have many fields that are indexed, it will be expensive to do writes
- Certain operations are slow or do not work
  - You might not have the right indexing type

Actions needed:

- Review Cosmos DB docs on optimizing indexing policy
- Modify the default indexing policy and consider removing indexes on unnecessary fields

NOT LEVERAGING HAVING COSMOS DB ACCOUNT REPLICATED IN MULTIPLE REGIONS

Symptoms:

- Misconception that Multi-region accounts is expensive
  - You may be routing all read requests to one single region and not leveraging both regions for read/writes
- Misconception that multi-master is expensive
  - Some customers may have multi-master enabled and not realize.

Actions needed:

- Use Cosmos DB docs on optimizing cost for multi-region accounts
- If your organization is highly cost-conscious, you can consider having the Cosmos DB account exist in only a single region (although not recommend)

LUNCH

Lab 2 – Creating and Connecting an App to CosmosDB

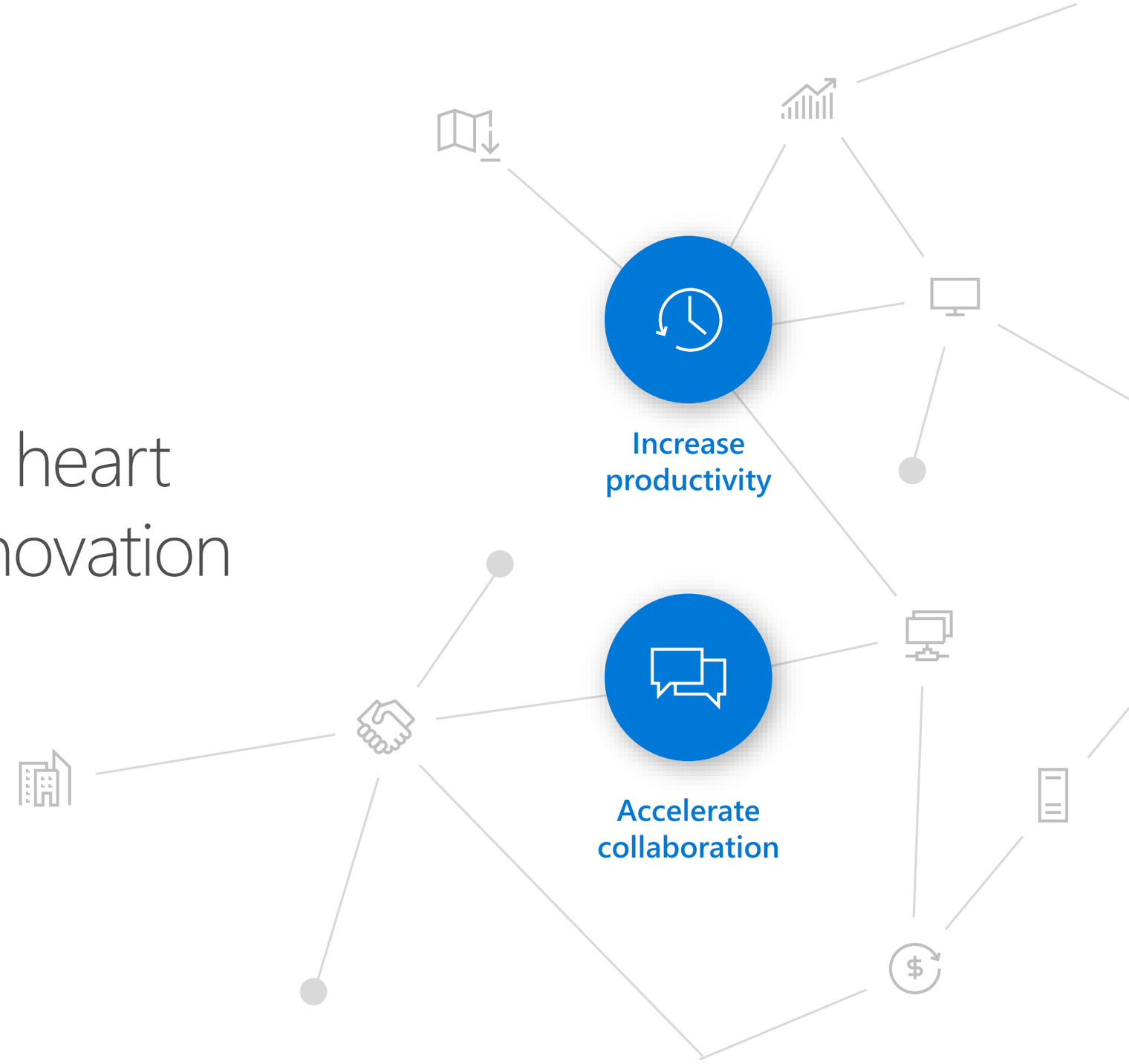
Git Hub Repo

Microsoft

Break

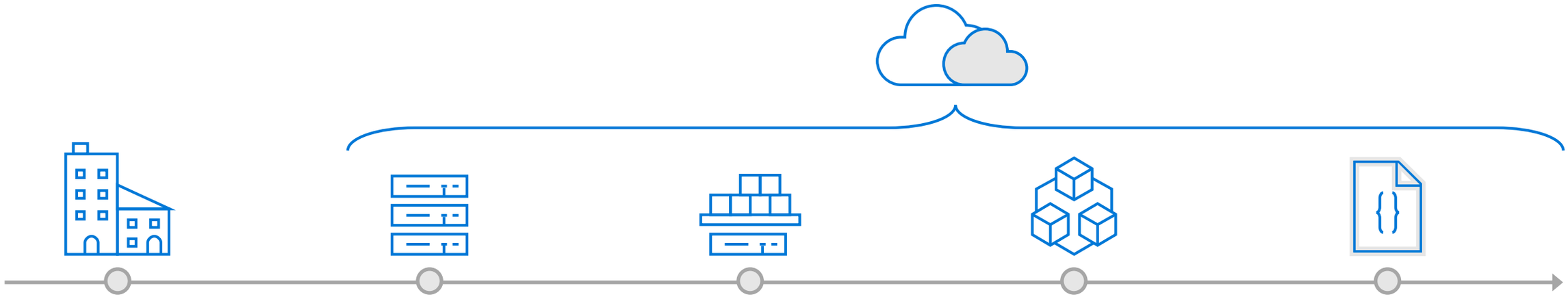
Using DevOps to Manage CosmosDB Apps

Developers are at the heart of your company's innovation



The way we build applications has changed

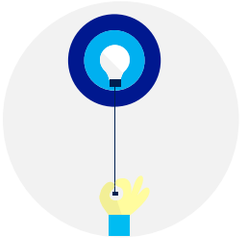
Past	Today
Long application cycles	**Rapid innovation**
Monolithic apps	**Loosely-coupled apps, microservices, and containers**
Servers and VMs	**Serverless**
Less data	**Big Data**
Desktop	**Mobile**
Distinct infrastructure and operations teams	**Service-focused DevOps teams**



IT challenges

There is extreme pressure on Ops teams to configure and provision apps and infrastructure quickly so that Dev teams can test and release new code

Highly integrated / **interdependent infrastructure** with **no end-to-end ownership** of a service



Complicated DR ("non modular")

Configuration Drift due to deficiencies in Information Technology Service Management (ITSM) **process implementation / delivery**

High service delivery lead time and over or under provisioned resources

Lack of **collaboration between Dev and Ops teams**

No room for **change windows**, running **out of support and issues with patching**

Developer challenges

There is extreme pressure on Dev teams to develop, test and release new release new code into production

Length of time to provision dev/test environments

Bugs are discovered late in release cycle

Increasing number of development languages to support

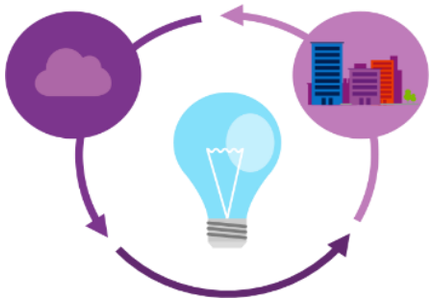
Lack of application telemetry and time to resolve issues

Lack of **collaboration between Dev and Ops teams**

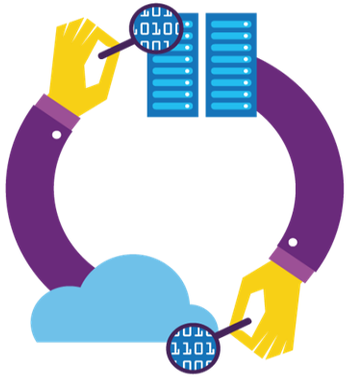
Releasing applications into production occurs infrequently and is complicated due to the number of new features releasing at once

What needs to change?

Shorten cycle times and deliver value faster



Improve quality and availability

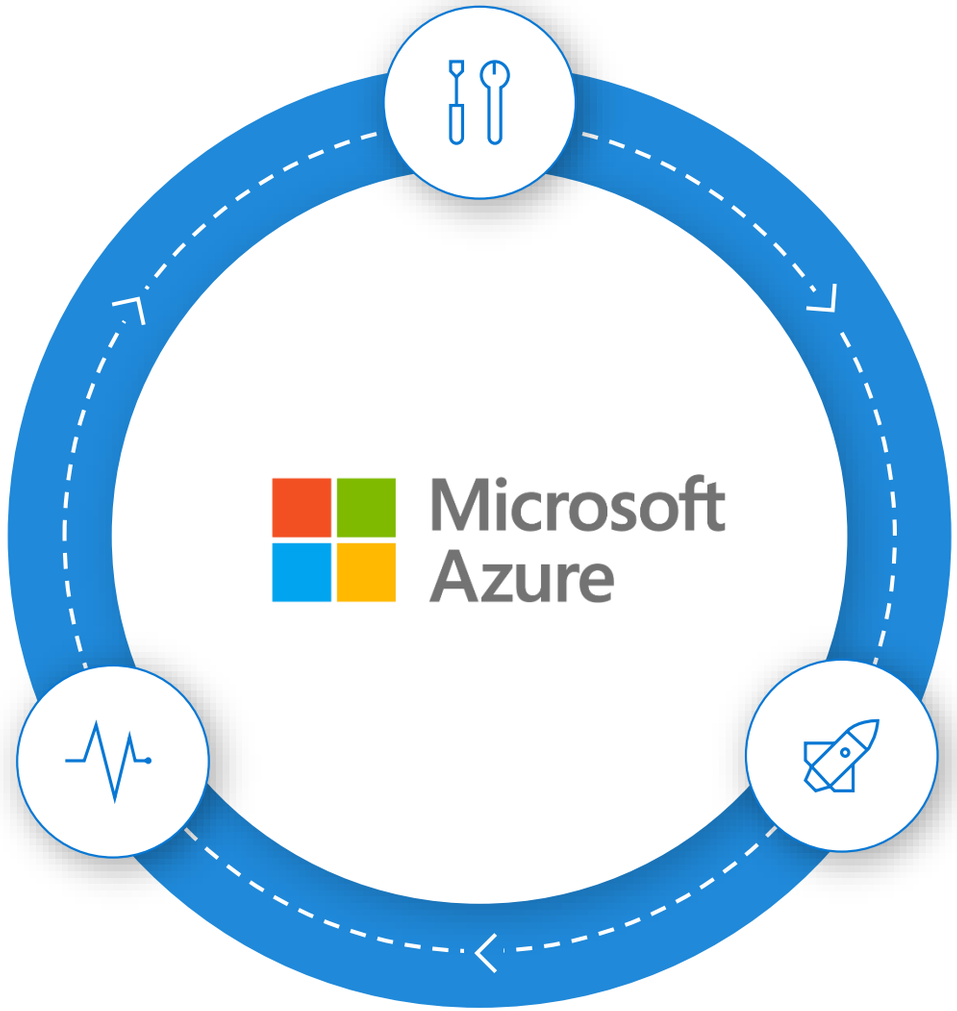


Optimize resources and **eliminate waste**

Deliver **innovation** with digital-era **velocity**

Innovate at scale.
Deliver with confidence.

Microsoft Azure is the cloud with DevOps tooling built in



Develop
GitHub
Visual Studio
Visual Studio Code

Operate
Azure Monitor
Azure Policy
Azure Automation
Azure Security Center

Deliver
Azure Boards
Azure Repos
Azure Pipelines
Azure Artifacts
Azure Test Plans
Azure Lab Services

Introducing Azure DevOps

Azure Boards

Deliver value to your users faster using proven agile tools to plan, track, and discuss work across your teams.

Azure Pipelines

Build, test, and deploy with CI/CD that works with any language, platform, and cloud. Connect to GitHub or any other Git provider and deploy continuously.

Azure Repos

Get unlimited, cloud-hosted private Git repos and collaborate to build better code with pull requests and advanced file management.

Azure Test Plans

Test and ship with confidence using manual and exploratory testing tools.

Azure Artifacts

Create, host, and share packages with your team, and add artifacts to your CI/CD pipelines with a single click.

https://azure.com/devops

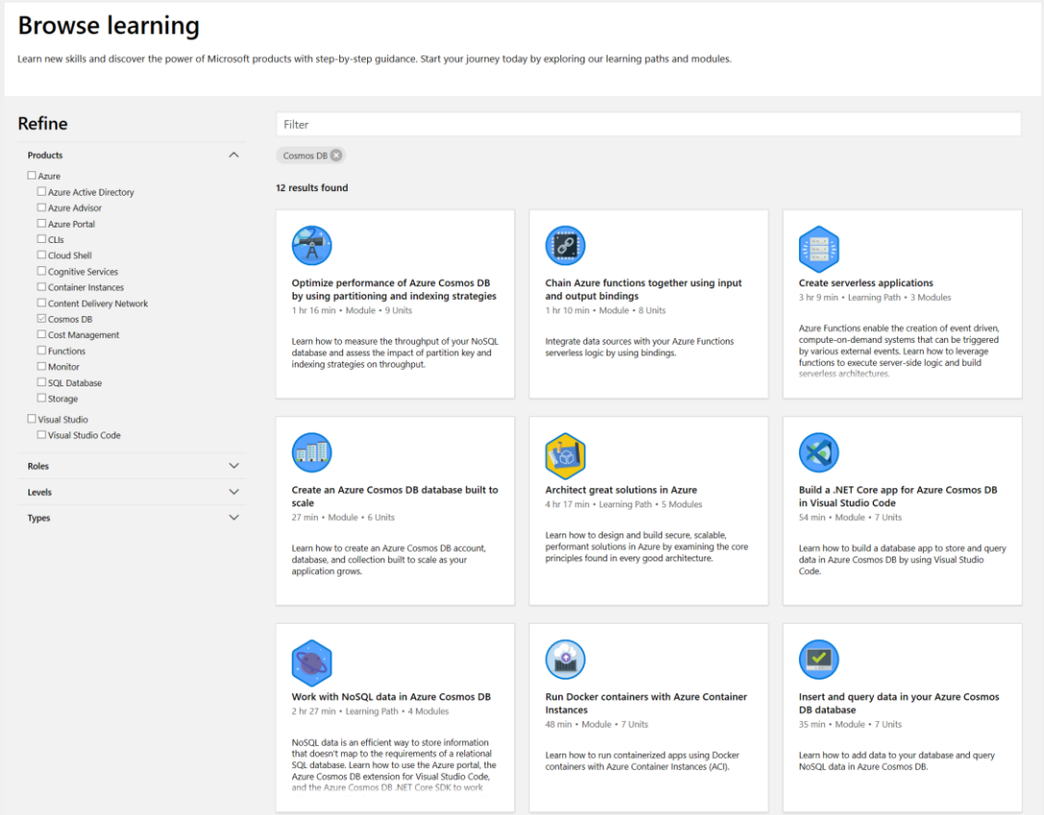
Lab 3 - Migrating SQL to CosmosDB

Lab - Migrating Mongo to CosmosDB using SQL API

Git Hub Repo

Learn new skills and discover the power of Microsoft products with step-by-step guidance. Start your journey today by exploring our learning paths and modules.

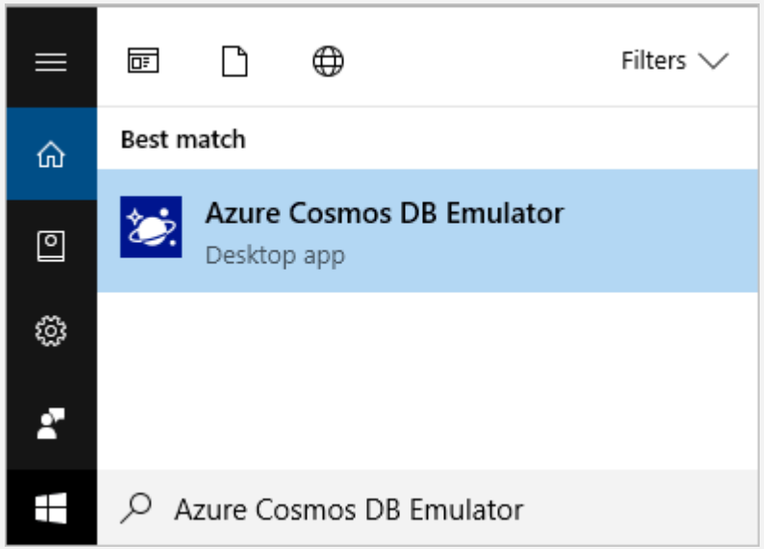
Azure Cosmos DB Learning Modules

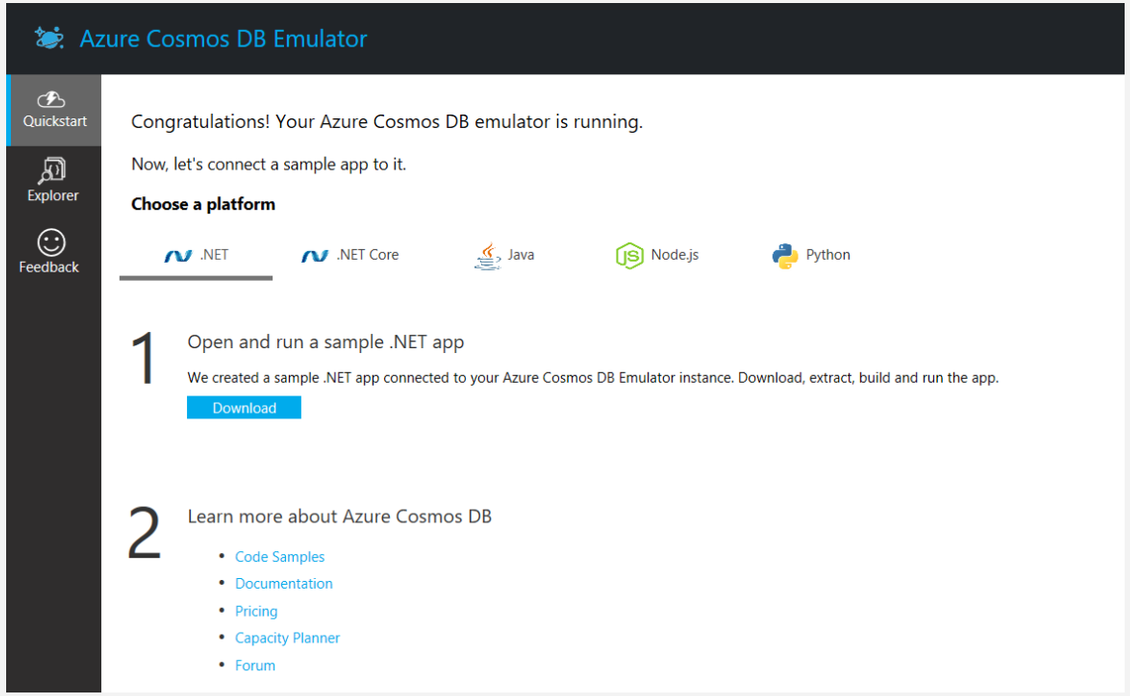


https://docs.microsoft.com/en-us/learn/browse/?products=azure-cosmos-db

TRY

Azure Cosmos DB Emulator

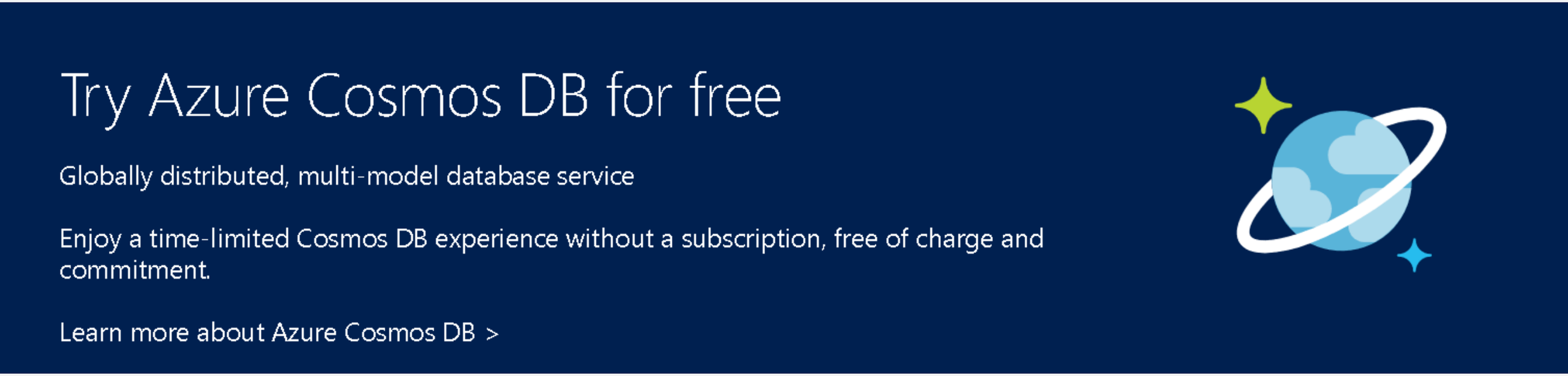




aka.ms/cosmosdb-emulator

TRY

Azure Cosmos DB gratis



azure.microsoft.com/try/cosmosdb/

Microsoft

askcosmosdb@microsoft.com

cosmosdb.com

@AzureCosmosDB
#CosmosDB

#azure-cosmosdb