

Azure Cosmos DB 的產業樣貌與實務應用

Eva Zhao (趙陽)

Microsoft, Asia Data & AI Senior Specialist

Agenda



- Azure Cosmos DB Overview
- Understand Azure Cosmos DB From Industry Use Case
 - FSI
 - Manufacturing
- Begin with Cosmos DB

Enterprises are embracing OSS and NoSQL DBaaS



#1

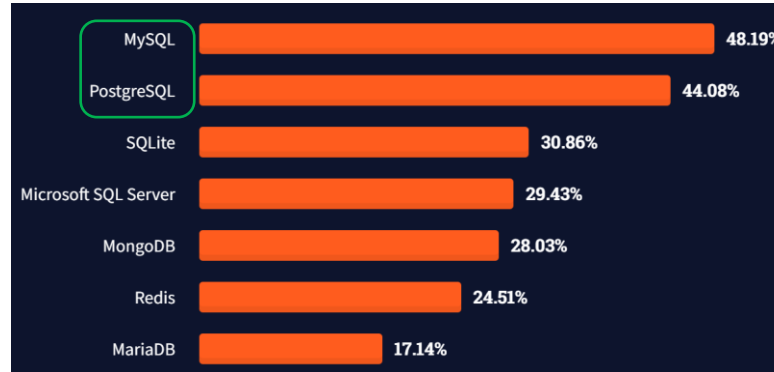
PostgreSQL DBMS of the Year 2020
and 2018 by DB-Engines



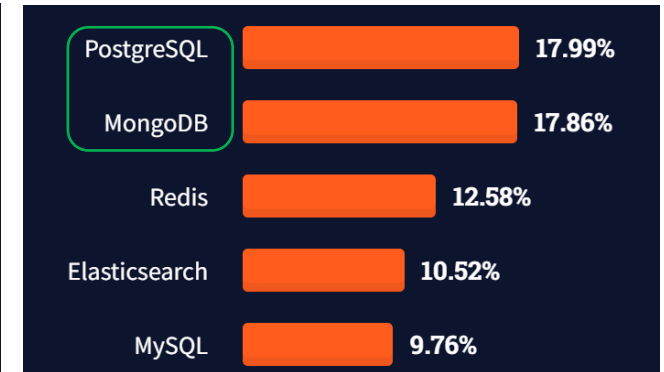
54%

MySQL #1 DBMS of the year 2019
used by 54% of developers

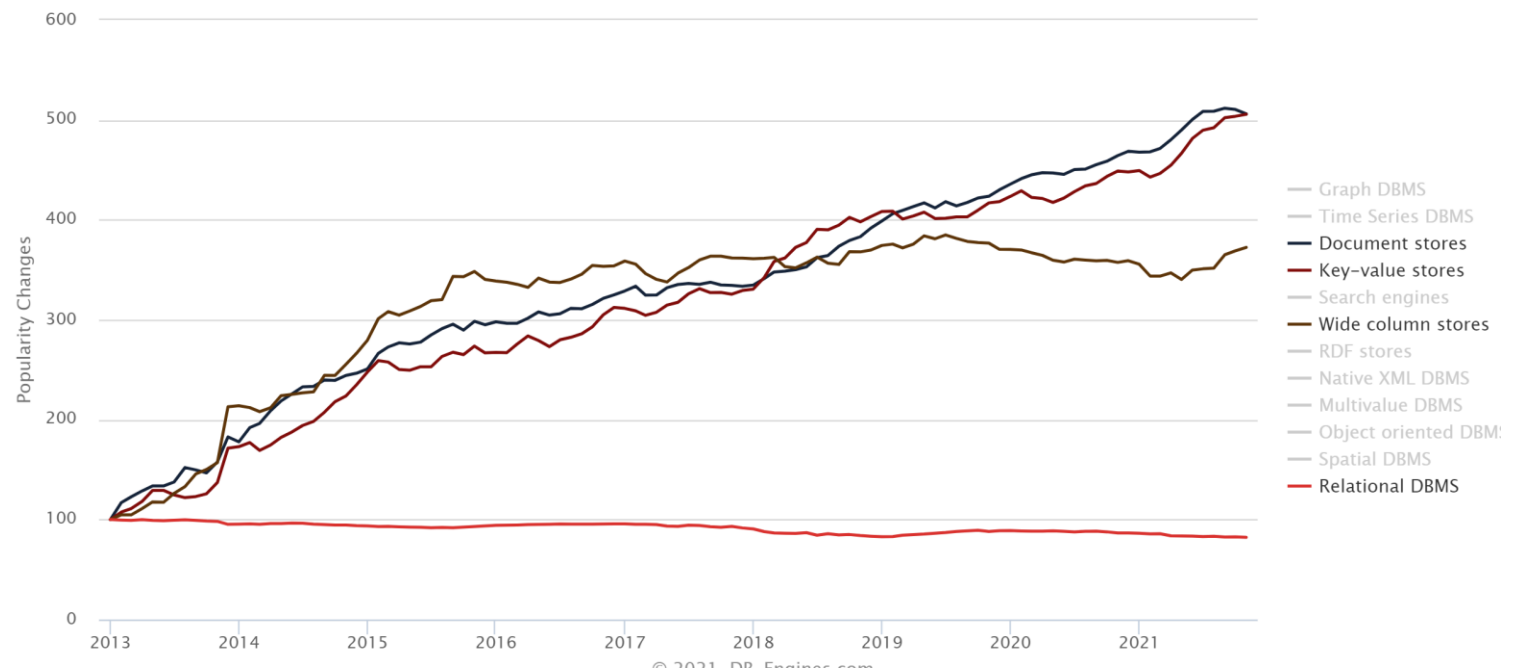
Most used Database by Professional Developers



Most wanted Database by Professional Developers



NoSQL vs. Relational Database Trends



Why NoSQL?



It 's all about real-time use cases!

Modern apps need to handle high
volume, variety, velocity of data
and serve it in **real-time**

- ✓ Respond in milliseconds
- ✓ Ultra low latency
- ✓ Always available 99.999%

Azure Cosmos DB is
uniquely positioned to meet these
business requirements



Deliver the best real-time customer experience

"On Black Friday, Azure Cosmos DB served 167 million requests in 24 hours. At our peak, we had 3,500 requests a second with an average response time of 48 milliseconds."



Deliver real-time personalized experiences

"Leveraging Azure Cosmos DB, we were able to develop a customized product matching algorithm for suggesting relevant products to customers."



Process events in real-time

"With Azure Cosmos DB, we can monitor 40 million endpoints in 150 countries, responding to 15 billion requests a day in less than one second."



Real-time payments, risk and fraud detection

"With Azure, we've brought our data into a single view which help us identify and evaluate risk."

Azure Cosmos DB Overview



Supports “all” popular NoSQL databases as managed DBaaS that provides “limitless” scalability, low latency and 5-nines availability with financially backed SLAs



Support all popular NoSQL databases



Native and open API support for MongoDB, Cassandra, Gremlin data and models

API for
MongoDB



SQL
Core API



Guaranteed speed and <10 ms latency anywhere



Guaranteed <10 ms latency backed by SLA. Automatic scaling with instant limitless elasticity.



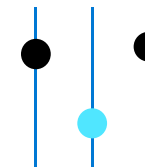
99.999% availability with zero RPO and RTO



Guaranteed 99.999% availability, zero RPO & RTO, backed by SLA and enterprise-level security.



Fully managed DBaaS experience



Auto-sharding, auto-scaling and auto-indexing for blazing fast queries and low TCO

<https://aka.ms/5whyCosmos>

Top use cases for Azure Cosmos DB



Use Case	Use Case Details	Why Azure Cosmos DB?	Alternatives	Optimization Tips	Customer Examples
Real-time telemetry	IoT - Logistics (track assets/inventory) - Smart cars (telematics) - Smart Cities - Smart Buildings User/Customer - Track digital actions (clicks; app events)	✓ Elastic, unlimited throughput (scale) ✓ Can easily ingest large volumes of data from write-heavy scenarios ✓ Real-time queries backed by <10 ms low-latency SLA ✓ Auto-indexing of semi-structured data	- Time-consuming and complex connection, integration, and management bundle of OSS tech (e.g., Redis, Cassandra, etc) - Maintain performance by actively managing, over-allocating, and over-paying for cluster	- Use Device ID or User ID as partition key - Customize index policy (vs auto-index all data) - Materialize View design pattern to optimize aggregates - Use local region for read/write data in SDK connection policy	Teijen Mercedes Benz Johnson Controls Exxon Mobil SitePro
Real-time recommendations and 360-degree customer view	Rapid response recommendations to enhance user experience (e.g., Netflix recommendations; e-commerce product suggestions)	✓ Spark integration offers ML and big data processing ✓ <10 ms low-latency SLA	- Build and maintain custom solutions; manage data integration (POS, CRM, etc.) - Static, non-responsive, or delayed recommendations based on stale data - Limited personalization (e.g., missing location, daypart, etc.)	- Custom configuration for Spark connector (e.g., batch size) - Scale up RU/s for batch processing then immediately scale down for steady-state - Use Product ID or User ID as partition key	ASOS Siemens Healthineers PureFacts
Mission-critical or geo-distributed applications	Deliver consistent real-time, always-on user experience anywhere in the world - Shopping cart/checkout - IoT device registry	✓ One-click, fully-managed data replication/distribution around the world ✓ <10 ms low-latency SLA ✓ 99.999% high-availability SLA ✓ Available in all Azure regions	- Set up and maintain infrastructure, networks, and databases globally - Manage data replication manually - Time, complexity, and cost to scale capacity and reach	- Use Session or Bounded Staleness pre-configured data consistency - Use local region for read/write data in SDK connection policy - Use multi-master writes/replication in production (not dev/test)	- Walmart/Jet.com - Citrix Chipotle Mexican Grill
NoSQL migration	Move from self-managed NoSQL (e.g., MongoDB, Cassandra) to PaaS solution and stop managing clusters.	✓ Full PaaS offering that handles all database/cluster management ✓ APIs for MongoDB, Cassandra, and other NoSQL enable migration with minimal code changes	- Create, manage, and tune clusters - Manage upgrades, patching, etc 24-7 internal team for monitoring, site reliability, and performance optimization	- Use Azure Data Migration Service to migrate to API for MongoDB - Use Azure Data Factory to migrate to SQL (core) API - Use Striim (or other preferred partner) for any migration	Bentley Systems Symantec (Norton)

Azure Cosmos DB Use Cases by Industry



<u>Retail</u>	<u>FSI</u>	Manufacturing & Logistics	Utilities & Telco	Healthcare	Hi-Tech & <u>Gaming</u>
Mobile Shopping App	Mobile Banking App	Order Tracking App	Mobile Utilities App	Mobile Health App	Mobile Gaming App
Personalization	Mobile payments	Device Telemetry (IoT)	Device Telemetry (IoT)	Claims settlement	User profiles
Recommendations	Transactions	Order Processing	Billing, payments	Prescription processing	Items inventory
Product Catalog	Fraud Detection	Inventory	Supply Chain Management	Billing, payments	Scoreboards
Order Processing	Credit Risk	Supply Chain Management	Real-time operational analytics	Supply Chain Management	Source code tracking
Supply Chain Management	Customer 360	Real-time operational analytics		Real-time operational analytics	
Real-time operational analytics					
Customer 360					



FSI: Real-time transactions

Industry Trends: Real-Time Transactions



Modernize Payments & Core Banking



42%

CAGR over the next 5 years for the US real-time payments transaction volume; projected to increase from \$734 million in 2019 to \$4.2 billion by 2024 ¹

69%

year-over-year growth in real-time payments in the United States in 2020 ²

1.9T

Projected number of cashless transactions in 2025; up from 1T in 2020 ³

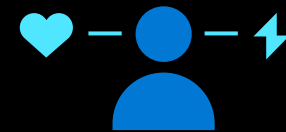
Top Challenges: Real-Time Transactions



Payment infrastructure **needs to be agile and mature** enough to capitalize on the benefits of the innovation



Need to **leverage enterprise data to drive greater business value**, develop new business models, and implement innovative pricing models



Need ways to **improve customer satisfaction and retention** by providing bridge from cash and check forms to real-time digital transactions

New digitally sophisticated market entrants are offering consumers innovative, value-added services that complement payments execution and fundamentally shift consumer expectations

Case Study: Real-Time Transactions



As one of a few US banks with a commercial banking license in China, East West operates in 120 locations throughout the United States and China. A few years ago, the bank was at an inflection point, where it needed a fresh, modern online presence to address the unique needs of its customer base. The bank chose Microsoft Azure to build a cloud-native, digital banking solution that it could quickly set up, tightly integrate, and elastically scale.

Approach

- Designed a brand-new architecture that is built to scale and highly available. The goal was lightning-fast processing times for banking transactions without batch processing.
- Azure Cosmos DB was picked to build semi-structured data from different datasets; *"We wanted the capability to ingest any type of data that comes into our system. Azure Cosmos DB gives us the ability to process transactions in real time and enables the services-based infrastructure that we require."* – Sunil Kumar, VP of Engineering, EastWest Bank

Outcomes

When a customer initiates a transaction, everything is done "over the wire" at the speed of light, within 20 milliseconds — the full spectrum of activities, including fraud protection. According to Rakesh Nambiar, CDO, *"Azure Stream Analytics, Azure Functions, Azure SQL Database, and Azure Cosmos DB are responsible for our biggest ROI"*



*"We were **focused on being real time, all the time, with no exceptions**. We started our Azure journey three-plus years ago with the question: How do we build something from scratch that can be scalable and highly available?"*

Rakesh Nambiar: Chief Digital Officer
East West Bank

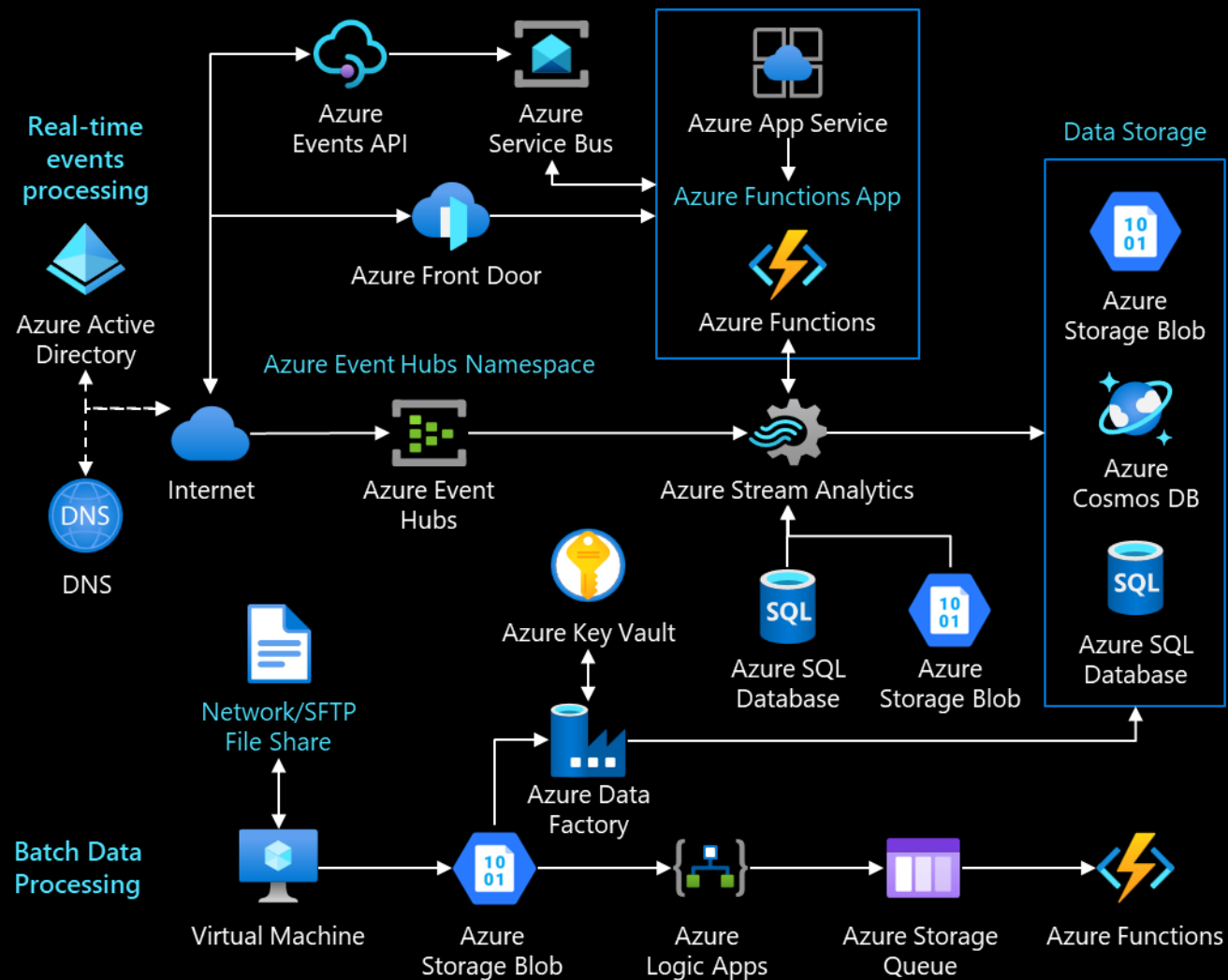


[Read Story](#)

Reference Architecture 1: Real-Time Transactions



Architecture for the Data Pipeline for Online Transactions



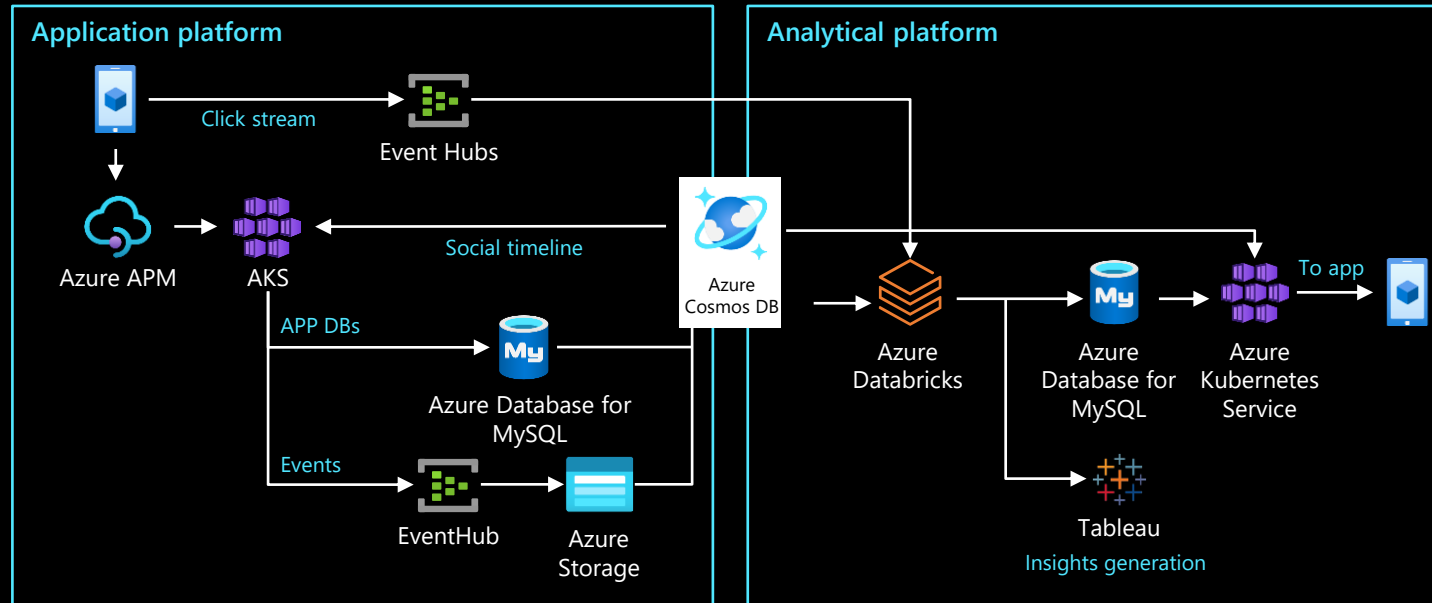
Key Highlights

- An Azure Stream Analytics job reads and processes events in Azure Event Hubs in real time. These are the orchestrators, and this is where the data gets processed.
- As the event is read, it goes through a set of checks to detect any frauds and to ensure that it complies with the business requirements. This is done by passing the event to Azure Functions for processing and fraud detection.
- After all this is done, the data finally lands in either Azure Cosmos DB or Azure SQL Database, based on the business rules.
- The design is for scalable and services-based infrastructure with disaster recovery across paired Azure Regions.

Reference Architecture 2: Real-Time Transactions



Digital Payment Platform Architecture for HSBC's PayMe App



Reference: [PayMe](#) social app allows consumers establish their personal networks while facilitating interactions and transactions with the people in their circle and business entities

Key Highlights

- In order to create more value for both businesses and consumers, Azure Cosmos DB is used for graph data modelled to store customer-merchant-transaction relationships.
- Massive amounts of structured and unstructured data from Azure Database for MySQL, Event Hubs, and Storage are streamed and transformed.
- The team designed an internally developed data ingestion process, feeding an analytical model called S.L.I.M (simple, lightly, integrated model), optimized for analytics queries performance, as well as making data virtually available to the analytics platform, using Azure Databricks Delta's unmanaged table capability.

Business Outcomes: Real-Time Transactions



Scale high volumes of payment data
at any time accurately and reliably



Centralized payment
pipelines with real-time
processing and analytics



Financial engine with
minimal dev ops



Fast, scalable,
on-demand, globally
distributed tech platform



Manufacturing: Real-time telemetry data

Industry Trends: Real-Time Telemetry Data



Build More Agile Factories

70%

of manufacturers may be using digital twins to conduct simulations and evaluations by 2022 ¹

50%

of U.S. organizations are investing in smart factories right now ²

5-20%

Loss in production due to unplanned downtime ³



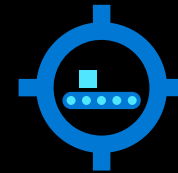
Top Challenges: Real-Time Telemetry Data



Need to **improve equipment effectiveness and asset productivity** by capturing and analyzing all connected shopfloor data



Need to **avoid unplanned downtime** due to unexpected machine failures



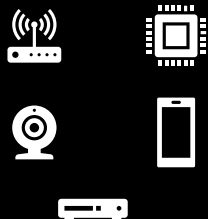
Deliver real-time platforms that enable factories to **closely monitor daily operation and perform real-time diagnostics remotely**

Leverage real-time data on cycle times, yields, utilization, and other factory insights to deliver more throughput and greater product quality with increased efficiencies and more uptime.

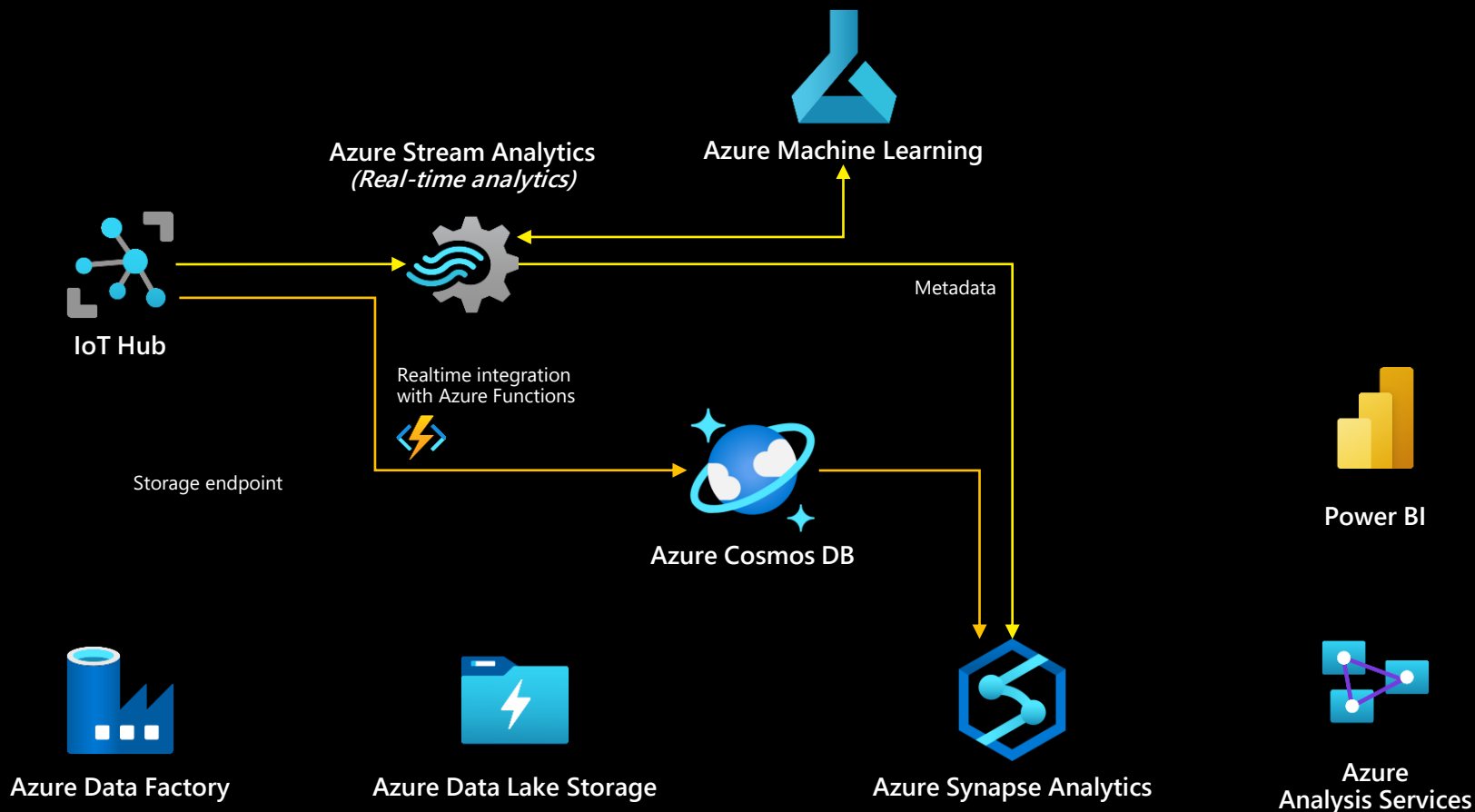
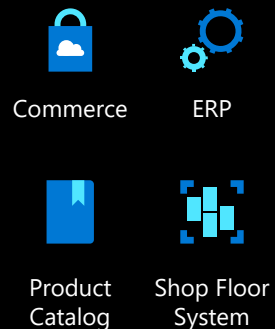
Reference Architecture1: Real-Time Telemetry Data



IoT Devices/Sensors



Manufacturing Systems



Case Study: Real-Time Telemetry Data



SitePro disrupted the oil and gas industry when it launched a fresh take on digital oilfield automation based on IoT tech. The company's innovative end-to-end approach automates everything from real-time data capture at the source to final payments and analysis. SitePro aggregates real-time streaming data from IoT sensors and controllers at facilities, pumps, valves, and wellheads. Site engineers can monitor and control every part of their field operations and solve issues from anywhere—enabling faster, safer, and more efficient operations.

Approach

- The SitePro platform manages high volumes of unstructured data from onsite IoT sensors and other monitoring devices. Sensor data is streamed to the SitePro edge API and ingested by the proprietary dataflow.
- The team used a combination of back-end Azure data services to store the real-time and historical data, including three managed databases—Azure Cosmos DB, Azure SQL Database, and Azure Cache for Redis.

Outcomes

- The new architecture of the SitePro solution has adapted the best from the original and has tweaked the back end to simplify the dataflow, creating an environment for smarter IoT decision-making.
- Adoption has been so high that at some points, the SitePro solution has controlled up to half of the fluid moving in the Permian Basin—the richest oil-producing area in the United States.

SITEPRO

*"Azure Cache for Redis was the only thing that had the throughput we needed. **Between Azure Cosmos DB and Azure Cache for Redis, we've never had congestion. They scale like crazy..**"*

Aaron Phillips

Founder, President

Co-CEO

SitePro

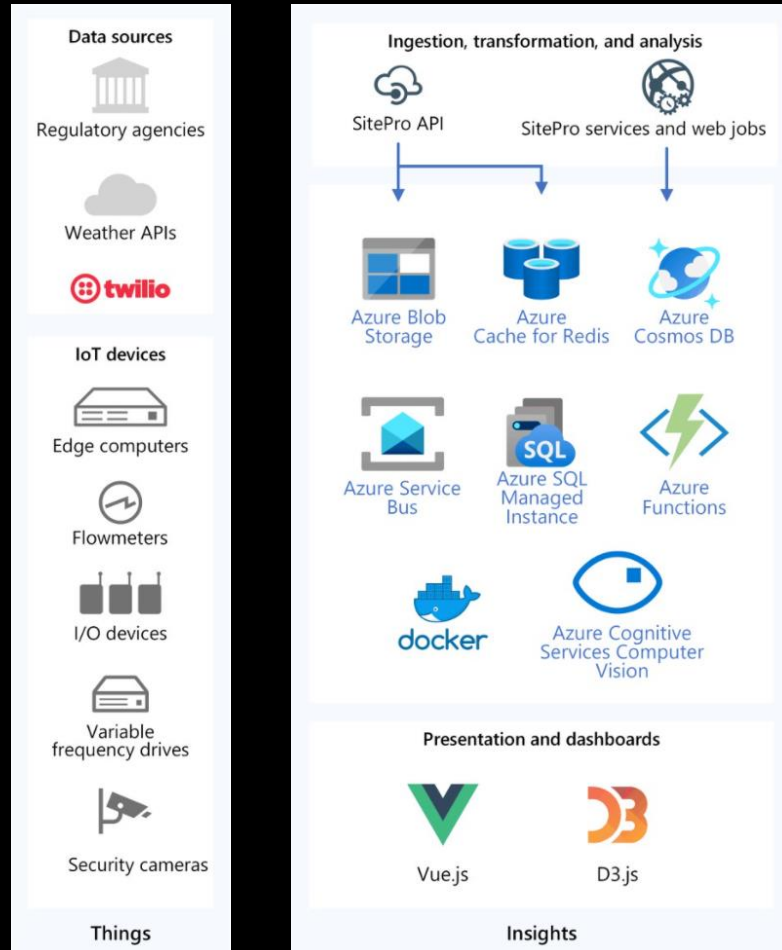


[Read Story](#)

Reference Architecture2: Real-Time Telemetry Data



Architecture



Key Highlights

- In this architecture, multiple services and web jobs capture real-time and historical data from hundreds of thousands of sensors, import weather and other external data, monitor communications, build reports, and provide relevant context for users
- Dozens of web jobs and app services in the background to handle change feeds and to pipe data to Azure Cosmos DB
- Key capabilities made possible due to the architecture include:
 - Web, mobile, and on-site access
 - Real-time IoT and intelligent control
 - Code-free configuration and deployment
 - Analytics suite
 - Field ticketing
 - Regulatory reporting
 - Camera streaming

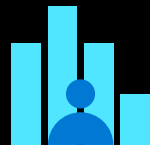
Business Outcomes: Real-Time Telemetry Data



Machine learning models
generate real-time
recommendations across
physical processes



Higher overall equipment
utilization and asset productivity



Improved employee productivity
with freeing up of time for
performing more value-added tasks



High app performance and
instant efficiency insights
worldwide with low latency

Demo: Azure Synapse Link for Cosmos DB



Microsoft Azure (Preview)

Search resources, services, and docs (G+)

srchi@microsoft.com
MICROSOFT

Home >

retail-cosmosdb | Data Explorer

Azure Cosmos DB account

Search (Ctrl+)

New Container

New Notebook

Connect to GitHub

Open Terminal

Reset Workspace

New SQL Query

Open Query

New Stored Procedure

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Quick start

Notifications

Data Explorer

Settings

Features

Replicate data globally

Default consistency

Firewall and virtual networks

Private Endpoint Connections

CORS

Keys

Add Azure Cognitive Search

SQL API

DATA

surfaceSalesDb

Product

RetailSales

StoreDemographics

transactionDb

NOTEBOOKS

Sample Notebooks (View Only)

My Notebooks

Welcome to Cosmos DB

Globally distributed, multi-model database service for any scale

Start with Sample

Get started with a sample provided by Cosmos DB

New Container

Create a new container for storage and throughput

New Notebook

Create a notebook to start querying, visualizing, and modeling your data

Common Tasks

New SQL Query

Open Query

New Stored Procedure

Scale & Settings

Recents

RetailSales Data

StoreDemographics Data

Product Data

Product Data

Tips

Data Modeling
Learn more about modeling

Cost & Throughput Calculation
Learn more about cost calculation

Configure automatic failover
Learn more about Cosmos DB high-availability

See more Cosmos DB documentation

Create and use Cosmos DB with simple steps



1. Create Azure Cosmos DB Account with an **API** and a **Capacity Model**
2. Create Database and Container with a configuration of **"Throughput"** and **"partition key" / "Shard key"**
3. Connect from application to insert/update items.

Microsoft Azure (Preview) Search resources, services, and docs (G+)

Dashboard > Create a resource >

Create an Azure Cosmos DB account

Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. [Learn more](#)

To start, select the API to create a new account. The API selection cannot be changed after account creation.

Azure Cosmos DB for NoSQL

Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.

[Create](#) [Learn more](#)

Azure Cosmos DB for MongoDB

Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB.

[Create](#) [Learn more](#)

Azure Cosmos DB for Apache Cassandra

Fully managed Cassandra database service for apps written for Apache Cassandra. Recommended if you have existing Cassandra workloads that you plan to migrate to Azure Cosmos DB.

[Create](#) [Learn more](#)

Azure Cosmos DB for Table

Fully managed database service for apps written for Azure Table storage. Recommended if you have existing Azure Table storage workloads that you plan to migrate to Azure Cosmos DB.

[Create](#) [Learn more](#)

Azure Cosmos DB for Apache Gremlin

Fully managed graph database service using the Gremlin query language, based on Apache TinkerPop project. Recommended for new workloads that need to store relationships between data.

[Create](#) [Learn more](#)

Azure Cosmos DB for PostgreSQL

Fully-managed relational database service for PostgreSQL with distributed query execution, powered by the Citus open source extension. Build new apps on single or multi-node clusters—with support for JSONB, geospatial, rich indexing, and high-performance scale-out.

[Create](#) [Learn more](#)

Create and use Cosmos DB with simple steps



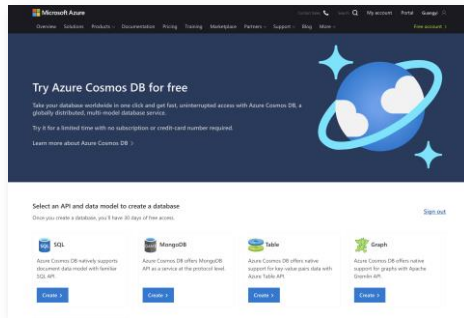
1. Create Azure Cosmos DB Account with an **API** and a **Capacity Model**
2. Create Database and Container with a configuration of **"Throughput"** and **"partition key" / "Shard key"**
3. Connect from application to insert/update items.

The screenshot shows the 'Create Azure Cosmos DB Account - Core (SQL)' page. The page is divided into sections: 'Project Details' and 'Instance Details'. In 'Project Details', the 'Subscription' is set to 'Microsoft Azure Internal Consumption' and the 'Resource Group' is empty with a 'Create new' link. In 'Instance Details', the 'Account Name' is 'Enter account name', the 'Location' is '(US) West US', and the 'Capacity mode' is set to 'Provisioned throughput' with a link to 'Learn more about capacity mode'. Below this, there are options for 'Apply Free Tier Discount' (set to 'Do Not Apply') and 'Limit total account throughput' (set to 'Limit the total amount of throughput that can be provisioned on this account'). At the bottom, there are three buttons: 'Review + create', 'Previous', and 'Next: Global Distribution'.

The screenshot shows the 'New Container' dialog in the Azure Cosmos DB Data Explorer. The dialog is open over the 'sparkdemo' account. The 'Database id' is set to 'Create new' with a text input field for a new database ID. The 'Database throughput (autoscale)' is set to 'Autoscale' with a 'Database Max RU/s' of 4000. The 'Container id' is set to 'e.g., Container1' and the 'Partition key' is set to 'e.g., /address/zipCode'. There is a 'Unique keys' section with an 'Add unique key' button. The background shows the 'Data Explorer' interface with a tree view of databases and containers.

Ready to build modern apps using the Azure Cosmos DB ?

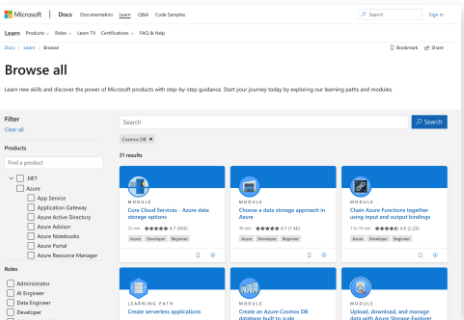
Get started today!



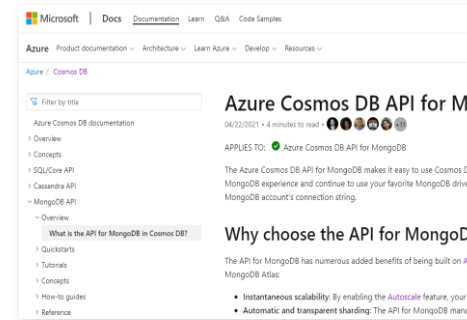
[Try Azure Cosmos DB](#) free or use free-tier to get started



[Learn](#) Azure Cosmos DB optimization & best practices



[Build your skills](#) with learning modules



[Read](#) Cosmos DB documentation



Thank You



Microsoft Azure Taiwan
Facebook 官方粉專



Azure Taiwan User Group
Facebook 交流社團

Eva Zhao (趙陽)

Microsoft, Asia Data & AI Senior Specialist