



Frederic Gisbert
Cloud Solution Architect
Azure Data & Analytics



Ali Bouhaddou
Cloud Solution Architect
Azure Data & Analytics

Training session DP900



DP-900

Partner Academy

Agenda











Describe non-relational data workloads

- describe the characteristics of non-relational data
- describe the types of non-relational and NoSQL data
- recommend the correct data store
- determine when to use non-relational data
- Describe non-relational data offerings on Azure
- identify Azure data services for non-relational workloads
- describe Azure Cosmos DB APIs
- describe Azure Table storage
- describe Azure Blob storage
- describe Azure File storage

Identify basic management tasks for non-relational data

- describe provisioning and deployment of non-relational data services
- describe method for deployment including the Azure portal, Azure Resource Manager templates, Azure PowerShell, and the Azure command-line interface (CLI)
- identify data security components (e.g., firewall, authentication, encryption)
- identify basic connectivity issues (e.g., accessing from on-premises, access with Azure VNets, access from Internet, authentication, firewalls)
- identify management tools for non-relational data

Describe non-relational data workloads

-  describe the characteristics of non-relational data
-  describe the types of non-relational and NoSQL data
-  recommend the correct data store
-  determine when to use non-relational data
-  Describe non-relational data offerings on Azure
-  identify Azure data services for non-relational workloads
-  describe Azure Cosmos DB APIs
-  describe Azure Table storage
-  describe Azure Blob storage
-  describe Azure File storage

How to choose between relational (SQL) and nonrelational (NoSQL) databases

| | NOSQL OR NONRELATIONAL | SQL OR RELATIONAL |
|--------------------|---|--|
| BEST FOR: | <ul style="list-style-type: none">•Handling large, unrelated, indeterminate, or rapidly changing data.•Schema-agnostic data or schema dictated by the app.•Apps where performance and availability are more important than strong consistency.•Always-on apps that serve users around the world. | <ul style="list-style-type: none">•Handling data that is relational and has logical and discrete requirements that can be identified in advance.•Schema that must be maintained and kept in sync between the app and database.•Legacy systems built for relational structures.•Apps requiring complex querying or multi-row transactions. |
| SCENARIOS: | <ul style="list-style-type: none">•Mobile apps.•Real-time analytics.•Content management.•Personalization.•IoT apps.•Database migration. | <ul style="list-style-type: none">•Accounting, finance, and banking systems.•Inventory management systems.•Transaction management systems. |
| SCALE: | <ul style="list-style-type: none">•Scales data horizontally by sharding across servers. | <ul style="list-style-type: none">•Scales data vertically by increasing server load. |
| DATA MODEL: | <ul style="list-style-type: none">•Database types: key-value, document, columnar, and graph databases.•Stores data depending on database type. | <ul style="list-style-type: none">•Database type: tables of rows, grouped into relations.•Uses Structured Query Language (SQL).•Stores data as rows in tables; related data stored separately and joined for complex queries. |

Document data stores

Relevant Azure service:

- [Azure Cosmos DB \(SQL API\)](#)



A document data store **manages a set of named string fields and object data values** in an entity that's referred to as a *document*.



These data stores typically store data in the form of JSON documents. **Each field value could be a scalar item**, such as a number, **or a compound element**, such as a list or a parent-child collection.



The **data in the fields of a document can be encoded in various ways**, including XML, YAML, JSON, BSON, or even stored as plain text. The fields within documents are exposed to the storage management system, enabling an application to query and filter data by using the values in these fields.

| Key | Document |
|------|---|
| 1001 | <pre>{ "CustomerID": 99, "OrderItems": [{ "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" }</pre> |
| 1002 | <pre>{ "CustomerID": 220, "OrderItems": [{ "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" }</pre> |

Columnar data stores

Relevant Azure service:

- [Cosmos DB Cassandra API](#)
- [HBase in HDInsight](#)



A columnar or column-family data store organizes data into columns and rows.

Columns are divided into groups known as column families.

Each column family holds a set of columns that are logically related and are typically retrieved or manipulated as a unit

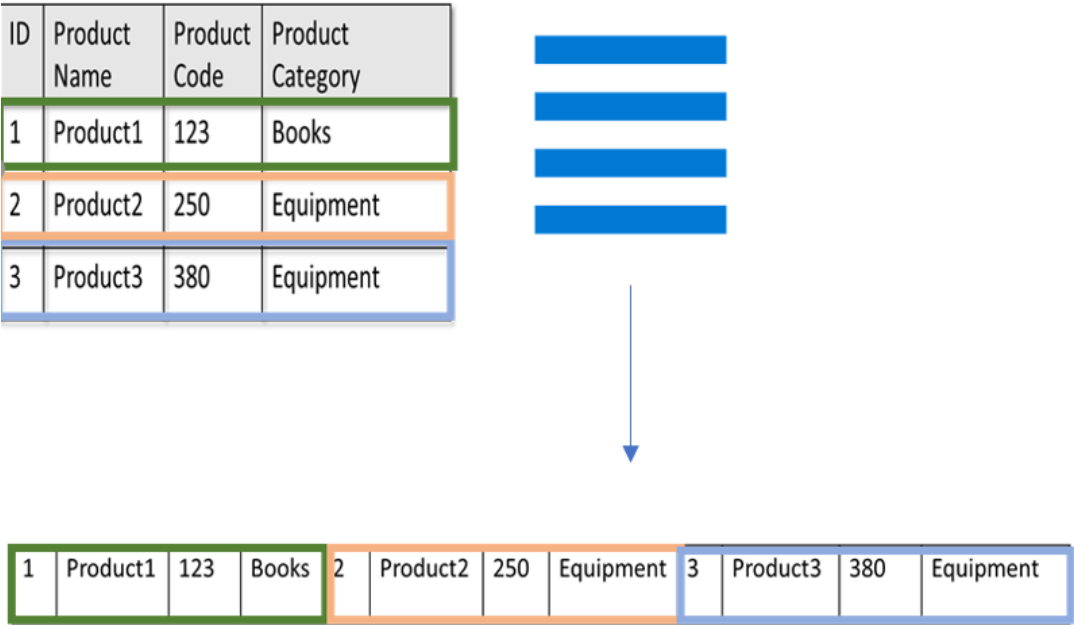
On disk, all of the columns within a **column family** are **stored together in the same file**, with a specific number of rows in each file. With large data sets, this approach creates a performance benefit by reducing the amount of data that needs to be read from disk when only a few columns are queried together at a time.

| CustomerID | Column Family: Identity |
|------------|--|
| 001 | First name: Mu Bae Last name: Min |
| 002 | First name: Francisco Last name: Vila Nova Suffix: Jr. |
| 003 | First name: Lena Last name: Adamczyk Title: Dr. |

| CustomerID | Column Family: Contact Info |
|------------|--|
| 001 | Phone number: 555-0100 Email: someone@example.com |
| 002 | Email: vilanova@contoso.com |
| 003 | Phone number: 555-0120 |

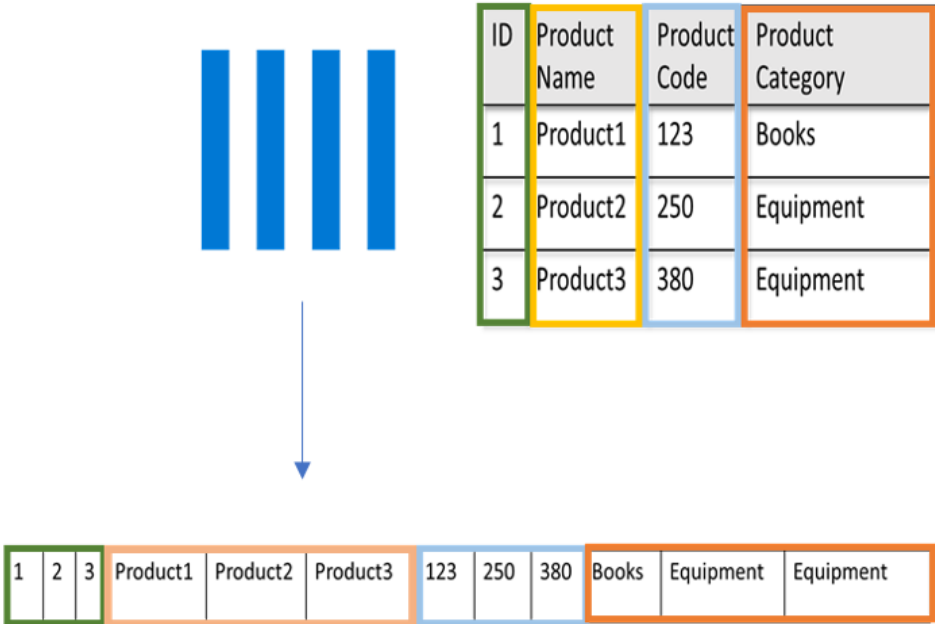
NB: wide-column stores Vs columnar databases

Wide-column or column family databases



Disk- Logical Representation

Columnar databases



Disk- Logical Representation

Key/value data stores

Relevant Azure services:

- [Azure Cosmos DB Table API](#)
- [Azure Cache for Redis](#)
- [Azure Table Storage](#)



key/value store is essentially a large hash table. You associate each data value with a unique key, and the key/value store uses this key to store the data by using an appropriate hashing function. The hashing function is selected to provide an even distribution of hashed keys across the data storage.



Most key/value stores only **support simple query, insert, and delete operations.** To **modify a value (either partially or completely), an application must overwrite the existing data for the entire value.** In most implementations, reading or writing a single value is an atomic operation. If the value is large, writing may take some time.



Key/value stores are **highly optimized for applications performing simple lookups using the value of the key**, or by a range of keys, but are **less suitable for systems that need to query data across different tables of keys/values**, such as joining data across multiple tables.

| Key (Customer ID) | Value (Customer Data) | | | | | |
|----------------------|--------------------------|-----|-------------|--------------|-----|-----------|
| C1 | AAAAA | BBB | 101 | Block Street | YY | 999 888 |
| C2 | MM | NN | 21 A Street | 5 B Avenue | | |
| C3 | DDD | EEE | FFF | 111 | 222 | 66 C Road |

Graph data stores

Relevant Azure service:

- [Azure Cosmos DB Graph API](#)



graph data store manages two types of information, nodes and edges.

<<Nodes>> represent entities, and <<edges>> specify the relationships between these entities.

Both nodes and edges can have properties that provide information about that node or edge, similar to columns in a table.

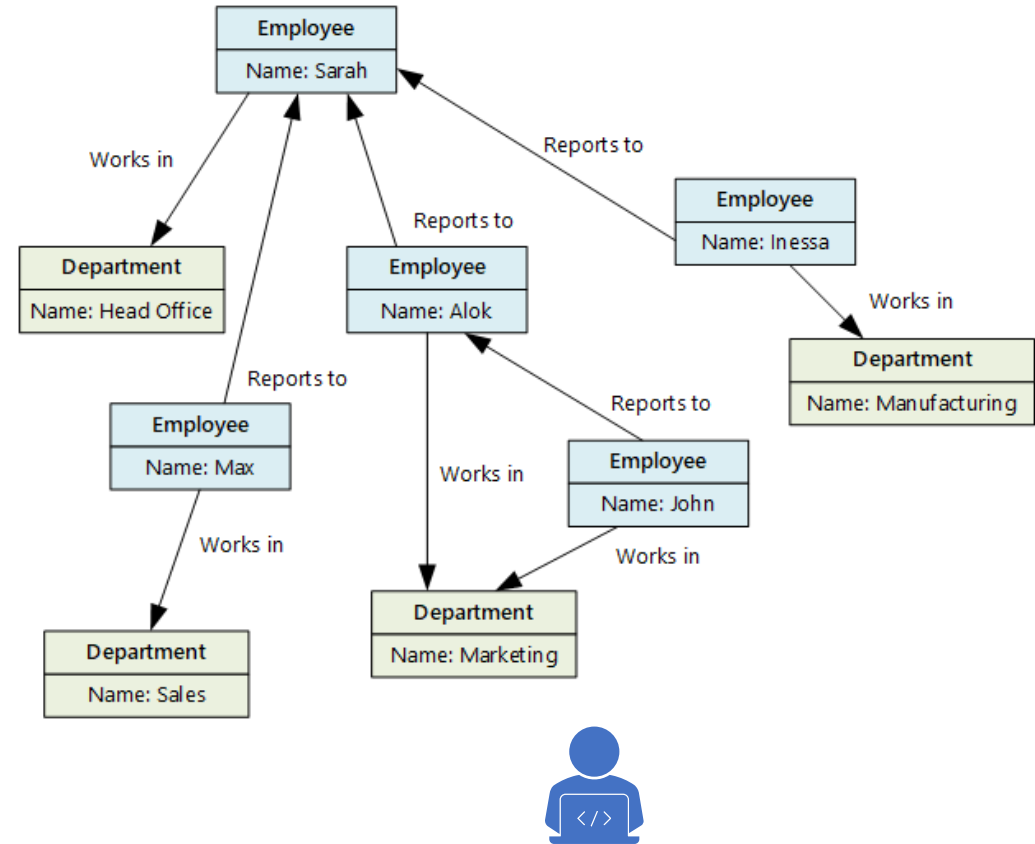
Edges can also have a direction indicating the nature of the relationship.



The purpose of a graph data store is to allow an application to efficiently perform queries that traverse the network of nodes and edges, and to analyze the relationships between entities.

For large graphs with lots of entities and relationships, you can perform complex analyses quickly.

Many graph databases provide a query language that you can use to traverse a network of relationships efficiently.



Time series data stores

Relevant Azure services:

Azure DataExplorer

- [Azure DataExplorer](#)
- [Azure Time Series Insights](#)
- [OpenTSDB with HBase on HDInsight](#)



Time series data is a **set of values organized by time**, and a time series data store is optimized for this type of data.

Time series data stores must **support a very high number of writes**, as they typically **collect large amounts of data in real time** from a large number of sources.



Time series data stores also **handle out-of-order and late-arriving data, automatic indexing of data points**, and optimizations for queries described in terms of **windows of time**



This last feature **enables queries to run across millions of data points and multiple data streams quickly**, in order to support time series visualizations, which is a common way that time series data is consumed

| timestamp | deviceid | value |
|-------------------------|----------|-------|
| 2017-01-05T08:00:00.123 | 1 | 90.0 |
| 2017-01-05T08:00:01.225 | 2 | 75.0 |
| 2017-01-05T08:01:01.525 | 2 | 78.0 |

Object data stores

Relevant Azure services:

- [Azure Blob Storage](#)
- [Azure Data Lake Store](#)
- [Azure File Storage](#)



Object data stores are optimized **for storing and retrieving large binary objects** or blobs such as **images, text files, video and audio streams**, large application data objects and documents, and virtual machine disk images.



An object consists of the stored data, some metadata, and a unique ID for accessing the object.

Object stores are designed to support files that are individually very large, as well provide large amounts of total storage to manage all files.



Some **object data stores replicate a given blob across multiple server nodes**, which **enables fast parallel reads**.

This process, in turn, enables the scale-out querying of data contained in large files, because multiple processes, typically running on different servers, can each query the large data file simultaneously.

| path | blob | metadata |
|--------------------------------|------------------|-----------------------|
| /delays/2017/06/01/flights.csv | 0XAABBCCDDEEF... | {created: 2017-06-02} |
| /delays/2017/06/02/flights.csv | 0XAADDCCDDEEF... | {created: 2017-06-03} |
| /delays/2017/06/03/flights.csv | 0XAEBBDEDDEEF... | {created: 2017-06-03} |

External index data stores

Relevant Azure service:

- [Azure Cognitive Search](#)

| id | search-document |
|---------|---|
| 233358 | {"name": "Pacific Crest National Scenic Trail", "county": "San Diego", "elevation": 1294, "location": {"type": "Point", "coordinates": [-120.802102, 49.00021]}} |
| 801970 | {"name": "Lewis and Clark National Historic Trail", "county": "Richland", "elevation": 584, "location": {"type": "Point", "coordinates": [-104.8546903, 48.1264084]}} |
| 1144102 | {"name": "Intake Trail", "county": "Umatilla", "elevation": 1076, "location": {"type": "Point", "coordinates": [-118.0468873, 45.9981939]}} |



External index data stores **provide the ability to search for information held in other data stores and services.**

An external index acts as a secondary index for any data store, and **can be used to index massive volumes of data and provide near real-time access to these indexes.**













The **indexes are created by running an indexing process.** This can be performed using a pull model, triggered by the data store, or using a push model, initiated by application code. Indexes can be multidimensional and may support free-text searches across large volumes of text data.



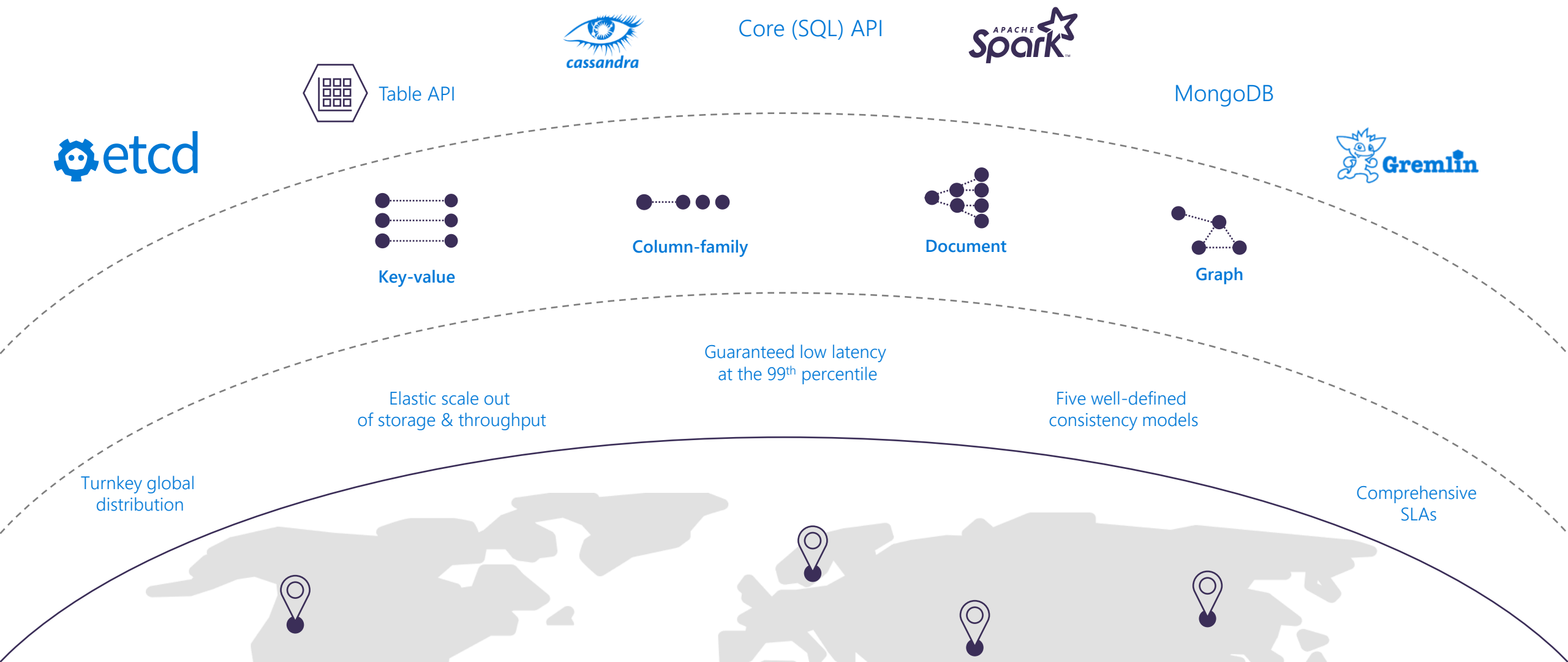
External index data stores are **often used to support full text and web-based search.** In these cases, searching can be exact or fuzzy. A **fuzzy search** finds documents that match a set of terms and calculates how closely they match.

Some external indexes also support **linguistic analysis** that can return matches based on synonyms, genre expansions (for example, matching "dogs" to "pets"), and stemming (for example, searching for "run" also matches "ran" and "running").

Describe non-relational data workloads

-  describe the characteristics of non-relational data
-  describe the types of non-relational and NoSQL data
-  recommend the correct data store
-  determine when to use non-relational data
-  Describe non-relational data offerings on Azure
-  identify Azure data services for non-relational workloads
-  describe Azure Cosmos DB APIs
-  describe Azure Table storage
-  describe Azure Blob storage
-  describe Azure File storage

What is...AZURE COSMOS DB?



AZURE COSMOS DB

FAST NoSQL DATABASE WITH OPEN APIs FOR ANY SCALE

Azure Cosmos DB is a fully managed NoSQL database for modern app development with SLA-backed speed and availability, automatic and instant scalability, and open source APIs for MongoDB, Cassandra, and other NoSQL engines.



Guaranteed speed at any scale

Gain unparalleled SLA-backed speed and throughput, fast global access, and instant elasticity.



Simplified application development

Build fast with open source APIs, multiple SDKs, schemaless data, and no-ETL analytics over operational data.



Mission-critical ready

Guarantee business continuity, 99.999% availability, and enterprise-level security for every application.



Fully managed and cost-effective

End-to-end database management with serverless and automatic scaling matching your application and TCO needs.

Top use cases for Azure Cosmos DB

| Use Case | Use Case Details | Why Azure Cosmos DB? | Alternatives | Optimization Tips | Customer Examples |
|---|--|---|--|---|---|
| Real-time telemetry | <ol style="list-style-type: none"> IoT <ul style="list-style-type: none"> Logistics (track assets/inventory) Smart cars (telematics) Smart Cities Smart Buildings User/Customer <ul style="list-style-type: none"> Track digital actions (clicks; app events) | <ul style="list-style-type: none"> ✓ 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 | <ul style="list-style-type: none"> • 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 | <ol style="list-style-type: none"> 1. Use Device ID or User ID as partition key 2. Customize index policy (vs auto-index all data) 3. Materialize View design pattern to optimize aggregates 4. Use local region for read/write data in SDK connection policy | <ul style="list-style-type: none"> • 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) | <ul style="list-style-type: none"> ✓ Spark integration offers ML and big data processing ✓ <10 ms low-latency SLA | <ul style="list-style-type: none"> • 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.) | <ol style="list-style-type: none"> 1. Custom configuration for Spark connector (e.g., batch size) 2. Scale up RU/s for batch processing then immediately scale down for steady-state 3. Use Product ID or User ID as partition key | <ul style="list-style-type: none"> • ASOS • Siemens Healthineers • PureFacts |
| Mission-critical or geo-distributed applications | Deliver consistent real-time, always-on user experience anywhere in the world <ul style="list-style-type: none"> • Shopping cart/checkout • IoT device registry | <ul style="list-style-type: none"> ✓ 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 | <ul style="list-style-type: none"> • Set up and maintain infrastructure, networks, and databases globally • Manage data replication manually • Time, complexity, and cost to scale capacity and reach | <ol style="list-style-type: none"> 1. Use Session or Bounded Staleness pre-configured data consistency 2. Use local region for read/write data in SDK connection policy 3. Use multi-master writes/replication in production (not dev/test) | <ul style="list-style-type: none"> • 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. | <ul style="list-style-type: none"> ✓ Full PaaS offering that handles all database/cluster management ✓ APIs for MongoDB, Cassandra, and other NoSQL enable migration with minimal code changes | <ul style="list-style-type: none"> • Create, manage, and tune clusters • Manage upgrades, patching, etc • 24-7 internal team for monitoring, site reliability, and performance optimization | <ol style="list-style-type: none"> 1. Use Azure Data Migration Service to migrate to API for MongoDB 2. Use Azure Data Factory to migrate to SQL (core) API 3. Use Striim (or other preferred partner) for any migration | <ul style="list-style-type: none"> • Bentley Systems • Symantec (Norton) |

When should the customer use Azure Cosmos DB?

One or more of these requirements...

- Flexible schema

- Sensor data may change over time



- Scalability

- Especially for writes

- Our use case needs high #writes/sec

- Cosmos guarantees throughput for any workload



- Low latency (fast)

- Cosmos guarantees <10ms at P99 for reads, writes of 1kb

- Can use as fast cache for key/val lookups



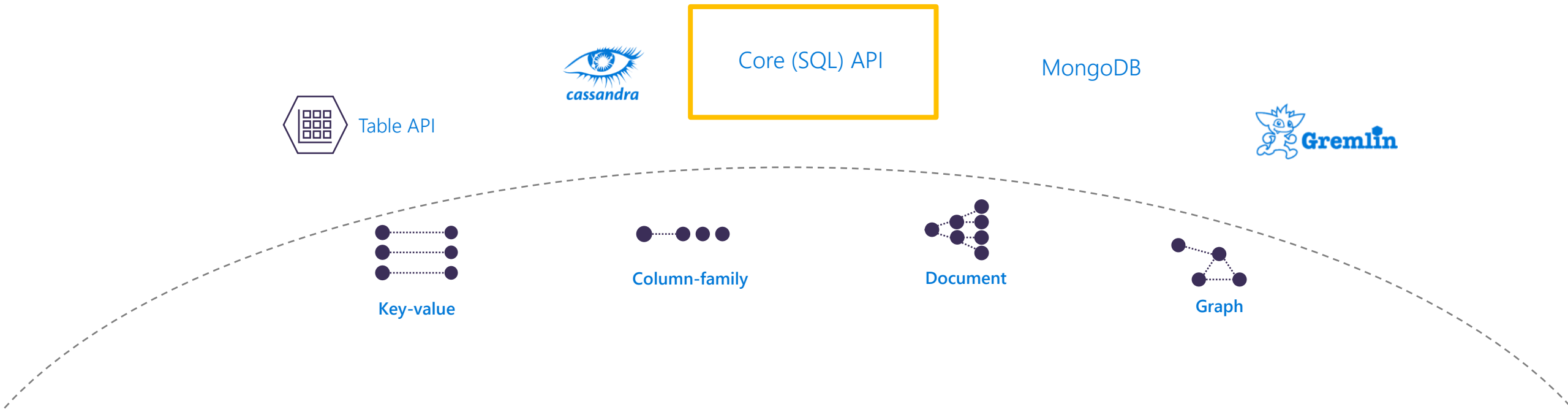
- High availability

- 99.999% HA with 2+ regions




Which API should the customer use for their app?

- For all new workloads – **Core (SQL) API**
 - Best developer experience – Cosmos builds interface, service, and SDKs
 - Gremlin API for graph data
- Other APIs for migration or DB platform preference/requirement



Production readiness checklist

- Add a second region for 99.999% High Availability

 **cosmos-friday** - Replicate data globally
Azure Cosmos DB account

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Quick start

Notifications

Data Explorer

Settings


Replicate data globally


Default consistency


Firewall and virtual networks


Private Endpoint Connections

CORS

 Save


 Discard

 Manual Failover

 Automatic Failover

Click on a location to add or remove regions from your Azure Cosmos DB account.

* Each region is billable based on the throughput and storage for the account. [Learn more](#)





Configure regions

Multi-region writes [ⓘ]

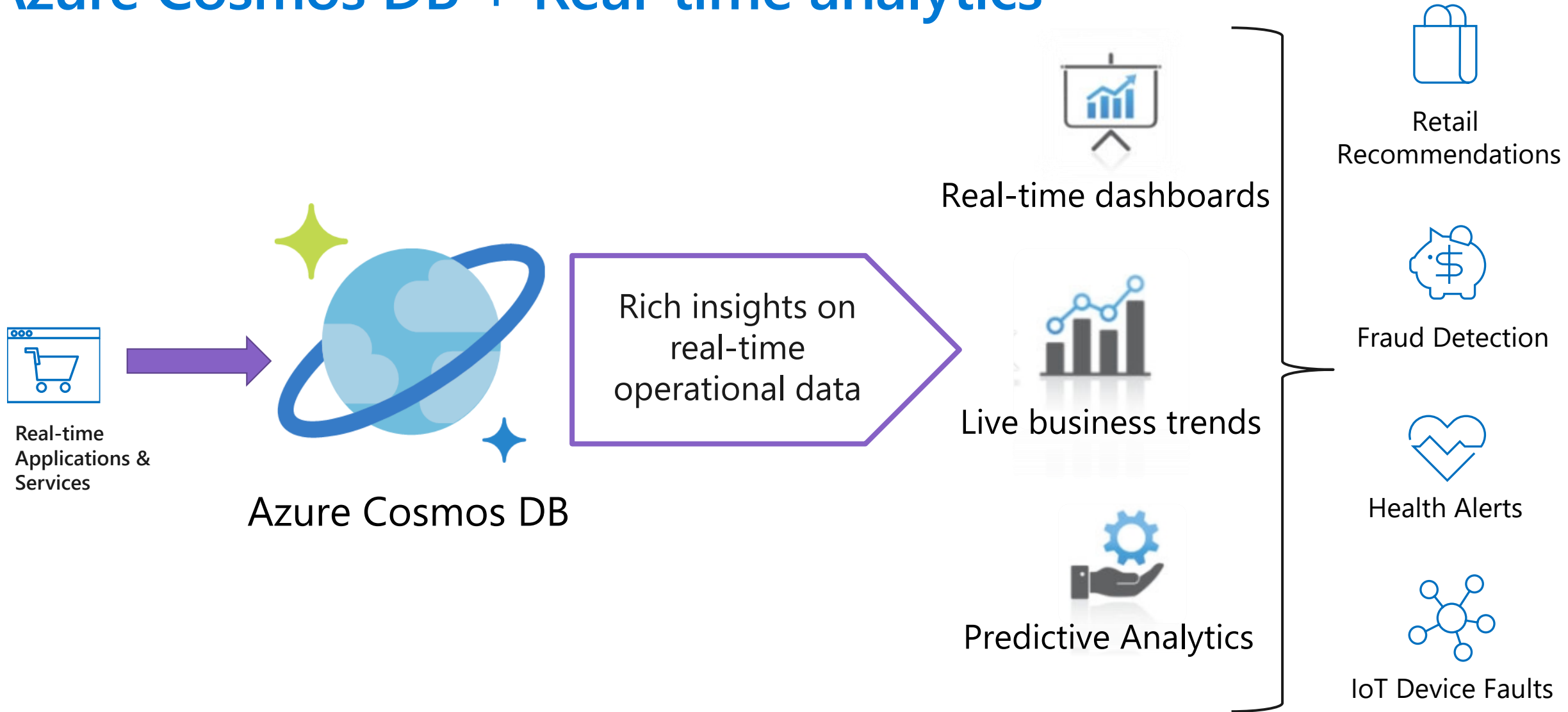
Disable

Enable

Configure the regions for reads, writes and availability zone (supported in selected regions and can only be configured when a new region is added). [+ Add region](#)

| Regions | Reads Enabled | Writes Enabled | Availability Zo... | Action |
|-----------|---------------|----------------|--------------------------|---|
| East US 2 | ✓ | ✓ | |  |
| West US 2 | ✓ | ✓ | <input type="checkbox"/> |  |

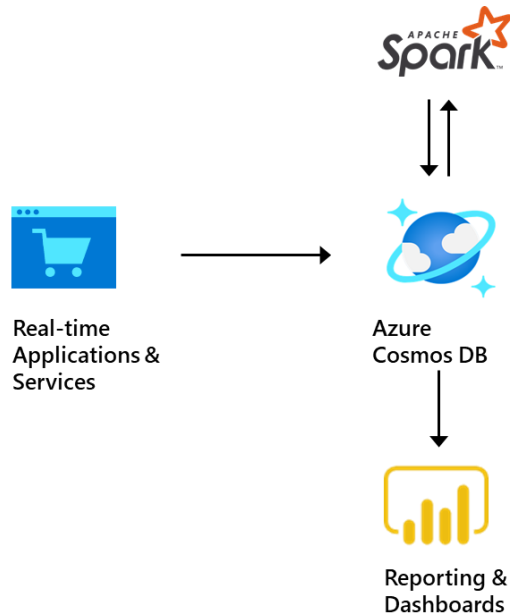
Azure Cosmos DB + Real-time analytics



Transform your business with actionable insights

Azure Cosmos DB analytics : Options

Run OLTP & OLAP workloads on the same database

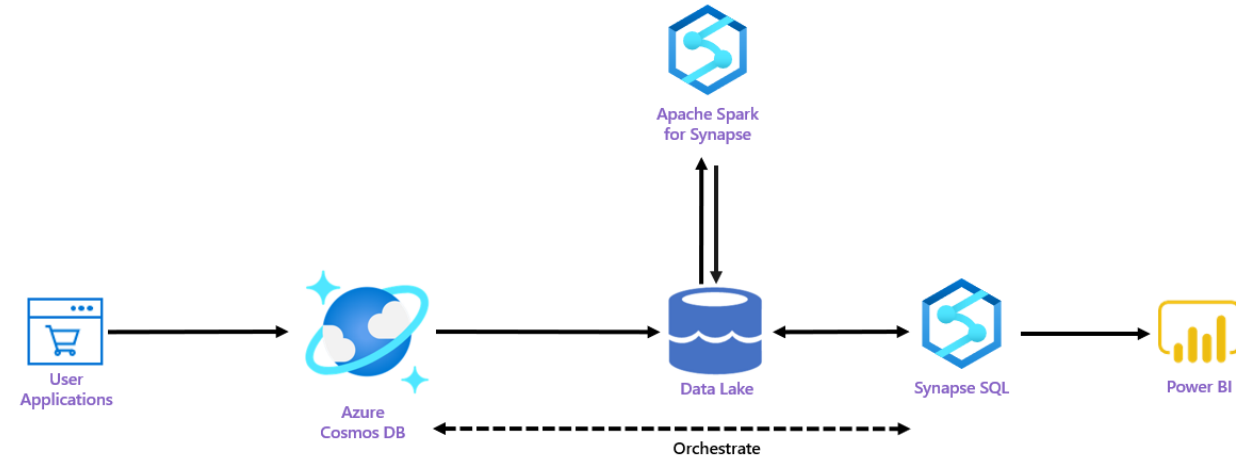


HUGE performance impact on the OLTP workloads at scale



As volume of operational data increases, latency of analytical queries increases and more resource intensive

Separating OLTP & OLAP



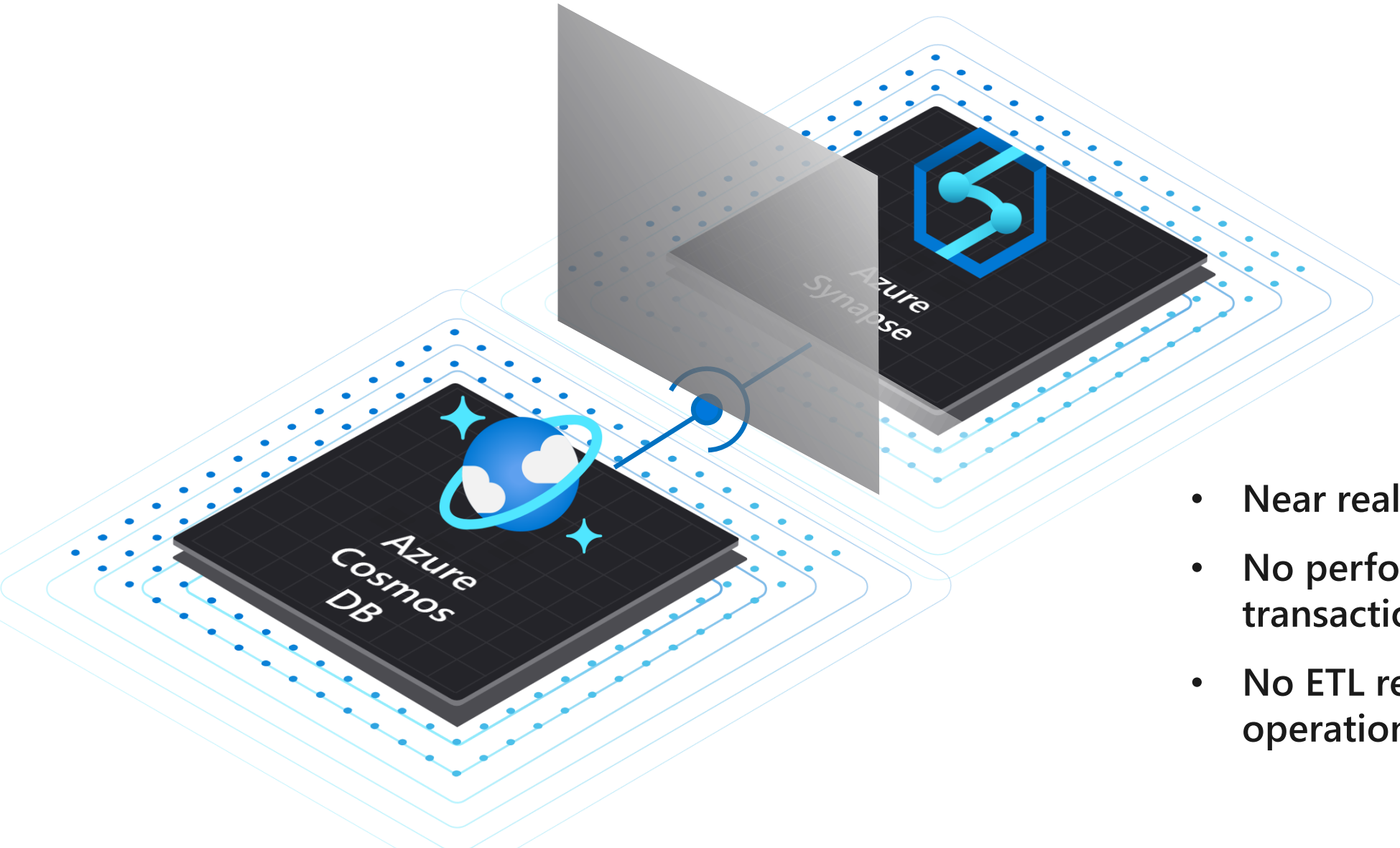
Ingest data periodically from Azure Cosmos DB to Data Lake

Delayed insights on data

Manage data formats and storage layer for analytics

Azure Synapse Link for Azure Cosmos DB

Breaking down the barrier between OLTP & OLAP



- Near real-time data analytics
- No performance impact on transactional workloads
- No ETL required to analyze operational data

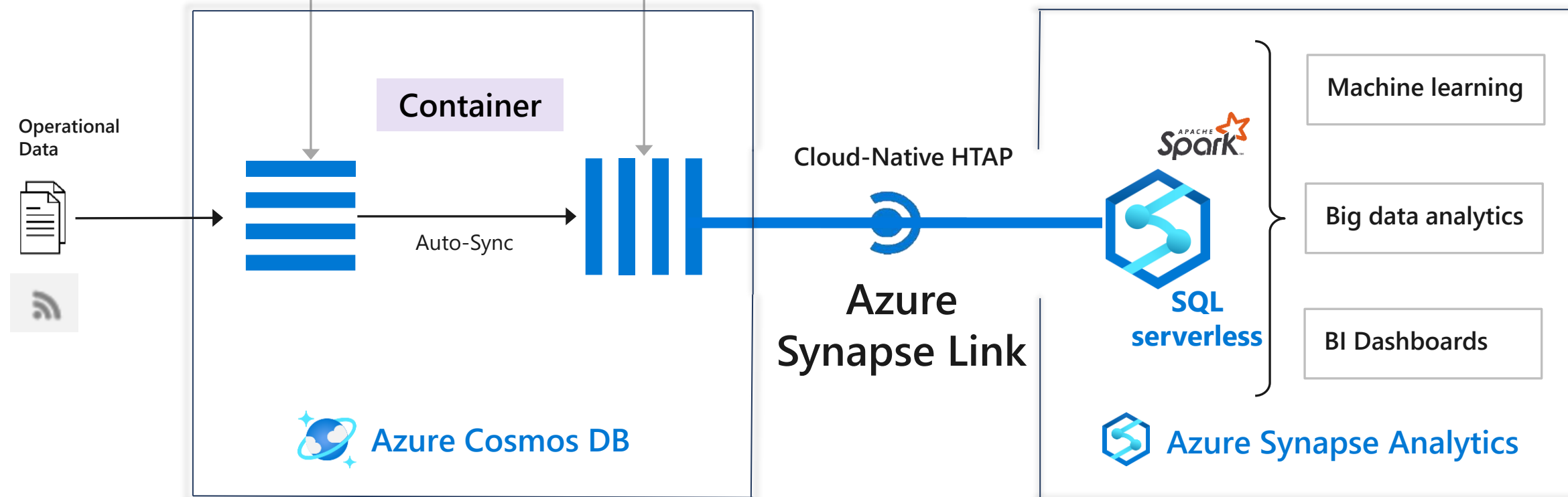
Azure Synapse Link : How it works?

Transactional Store

Row store optimized for transactional operations

Analytical Store

Column store optimized for analytical queries

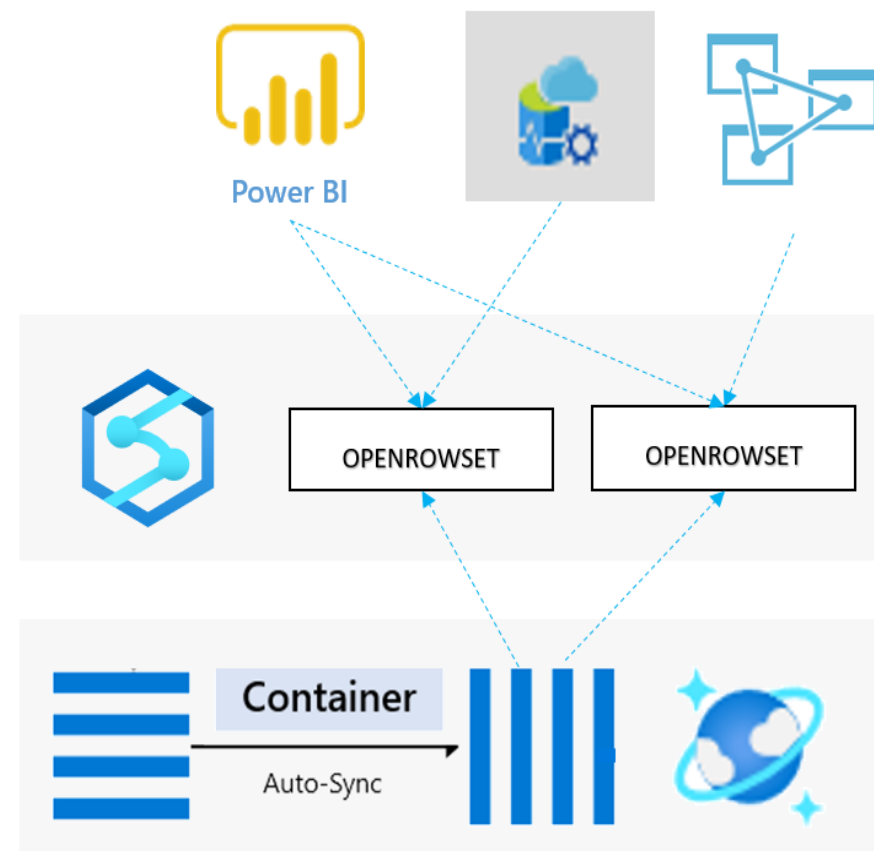


Generate near real-time insights on your operational data

Azure Synapse Link : Query Azure Cosmos DB Data in Synapse SQL serverless



- Run *analytical T-SQL queries*:
Query Azure Cosmos DB data in place, in seconds
- Build rich *near real-time dashboards*:
Using Power BI integrated in Azure Synapse
- Use wider range of *BI and ad-hoc querying* tools:
Integrated using T-SQL interface
- Build a *logical data warehouse* :
Analyze unified view of data across Azure Cosmos DB,
Azure Data Lake Storage & Azure Blob Storage



Available in public preview, starting October 2020

>>

Publish all 1 Validate all Refresh Discard all

>>

101 Query CosmosDB ●

Run Undo | Publish Query plan

Connect to SQL on-demand Use database SQLServerlessDB

```
1
2  -- Schema inference
3  SELECT *
4  FROM OPENROWSET ('CosmosDB',
5   'Account=retailplatform-cosmosdb;Database=RetailSalesDemoDB;region=WestUS;Key=9bNC7Se5qOkbBZyTy7DvaQ7puhosbny4EcZ24P3XT3sZCLFX7Acj0zOnhYoYJskmj1PO
6   Products
7 )
8 AS p
9
```

Results Messages

View Table Chart Export results

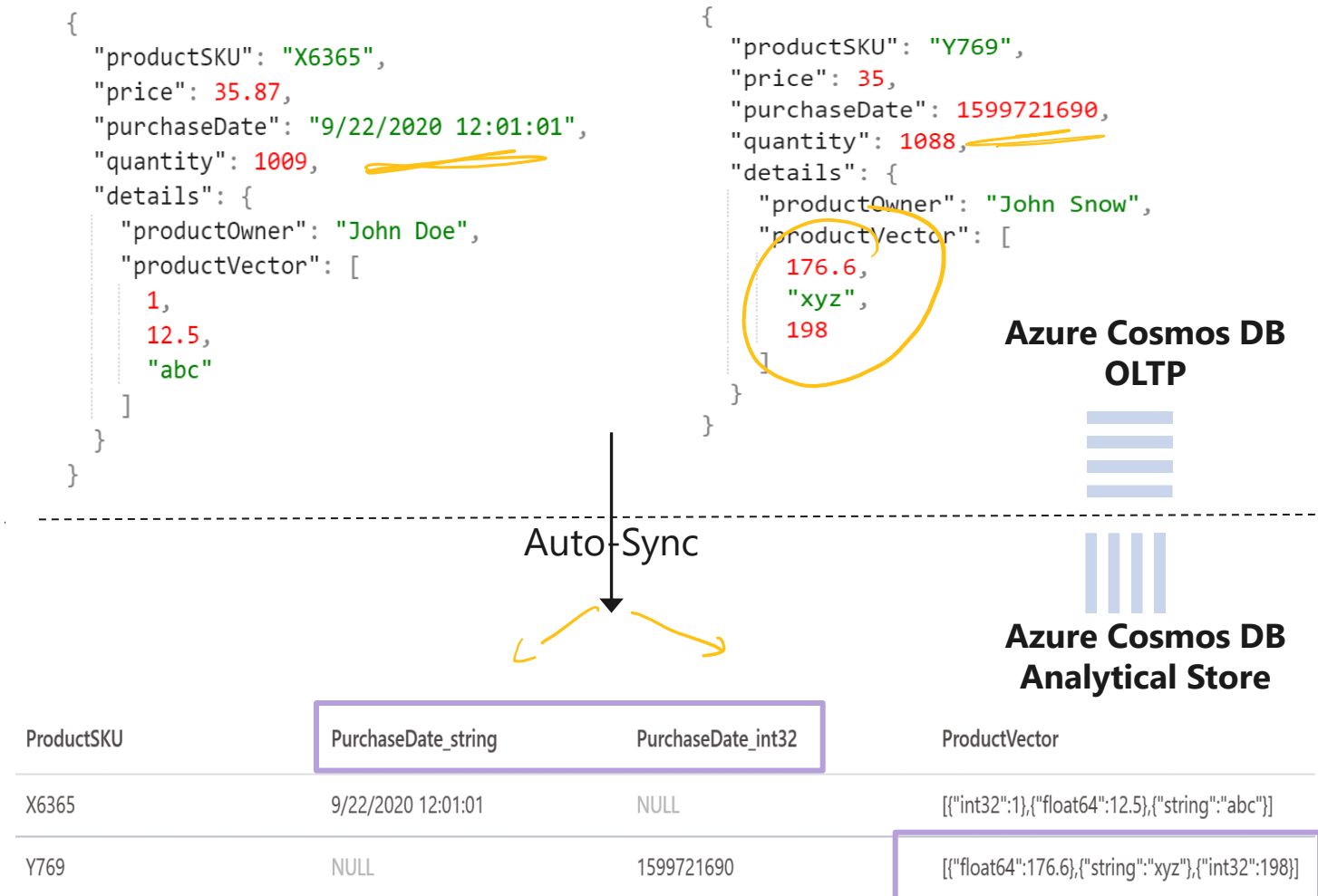
| _id | _ts | _etag | Id | ProductCode | WholeSaleCost |
|-------------------------------------|------------|-----------------------------------|-----------------------------|-----------------|---------------|
| {"objectId":"0x5F57255744D5E16..."} | 1599546713 | "0c01eec2-0000-0700-0000-5f57..." | NWY1NzI1NTc0NGQ1ZTE2MDYz... | surface.go | 220.55 |
| {"objectId":"0x5F57255744D5E16..."} | 1599546713 | "0c01efc2-0000-0700-0000-5f57..." | NWY1NzI1NTc0NGQ1ZTE2MDYz... | surface.pro7 | 400.83 |
| {"objectId":"0x5F57255744D5E16..."} | 1599546713 | "0c01f0c2-0000-0700-0000-5f57..." | NWY1NzI1NTc0NGQ1ZTE2MDYz... | surface.laptop3 | 623.15 |

00:00:09 Query executed successfully.

Azure Synapse Link : Support for Azure Cosmos DB API for MongoDB

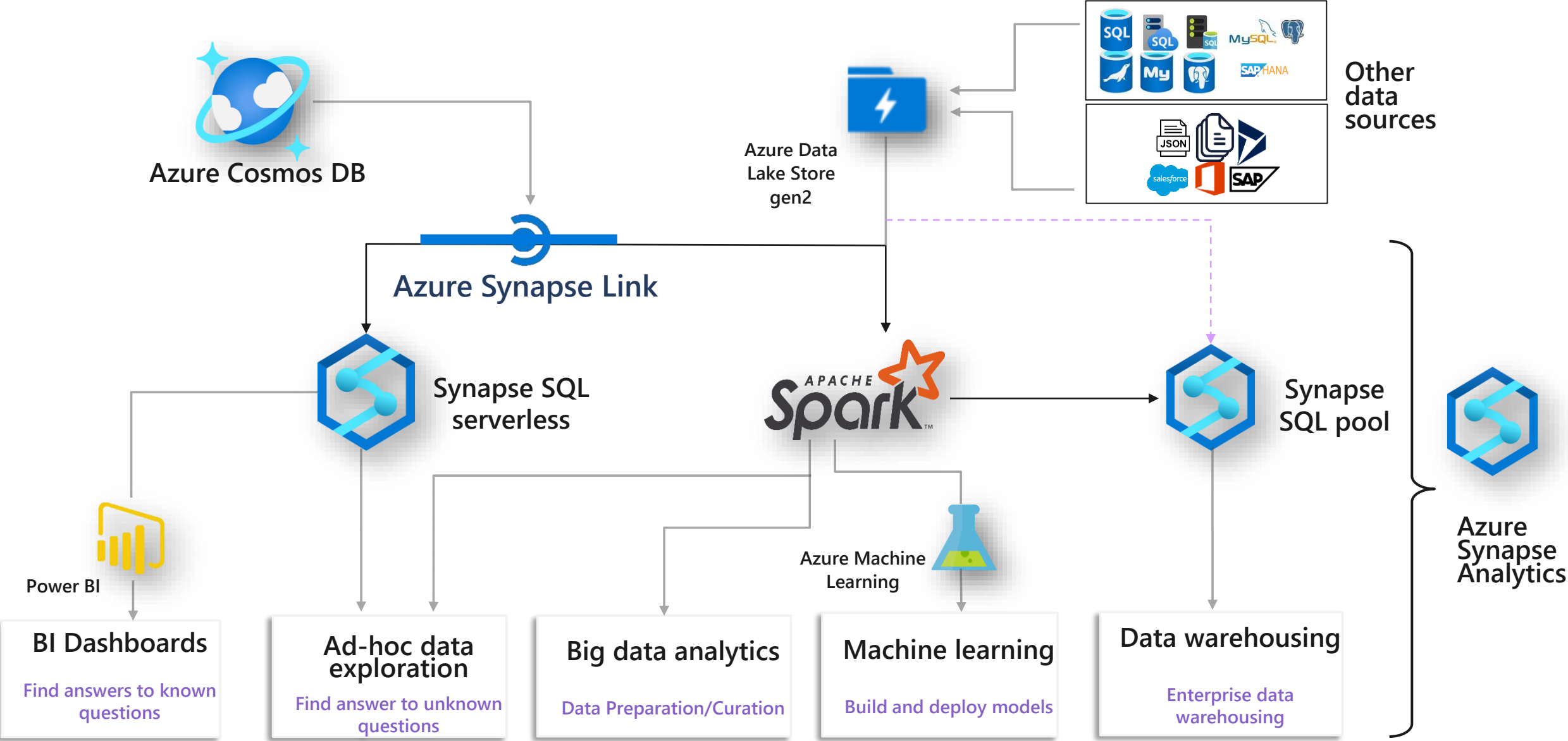
NEW

- Cloud native HTAP for MongoDB data
- Run Synapse T-SQL & Spark queries against MongoDB:
 - No impact to OLTP workloads
 - No need to ETL to OLAP store
- Analytics against highly polymorphic MongoDB BSON schemas:
 - Full-fidelity schema representation in Azure Cosmos DB analytical store (fully managed in auto-sync)



Available now in public preview

Analytics + BI patterns with Azure Synapse Link



Describe non-relational data workloads











-  describe the characteristics of non-relational data
-  describe the types of non-relational and NoSQL data
-  recommend the correct data store
-  determine when to use non-relational data
-  Describe non-relational data offerings on Azure
-  identify Azure data services for non-relational workloads
-  describe Azure Cosmos DB APIs
-  describe Azure Table storage
-  describe Azure Blob storage
-  describe Azure File storage

Table storage concepts

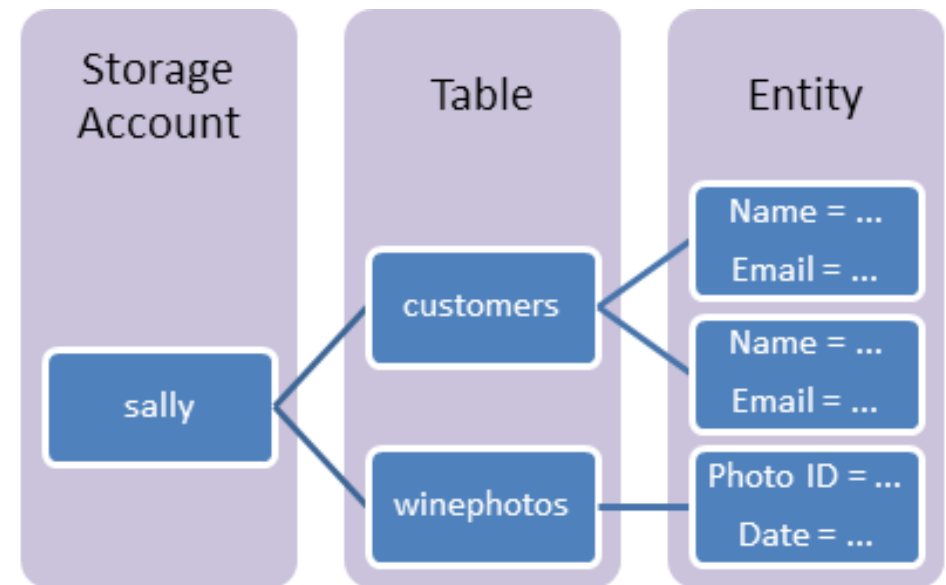
•**URL format:** Azure Table Storage accounts use this format: <http://<storage account>.table.core.windows.net/<table>>
Azure Cosmos DB Table API accounts use this format: <http://<storage account>.table.cosmosdb.azure.com/<table>>
You can address Azure tables directly using this address with the OData protocol.
For more information, see [OData.org](http://odata.org).

•**Accounts:** All access to Azure Storage is done through a storage account. For more information about storage accounts, see [Storage account overview](#).
All access to Azure Cosmos DB is done through a Table API account. See [Create a Table API account](#) for details creating a Table API account.











•**Table:** A table is a collection of entities. Tables don't enforce a schema on entities, which means a **single table can contain entities that have different sets of properties.**

•**Entity:** An **entity is a set of properties, similar to a database row.** An entity in Azure Storage can be up to 1MB in size. An entity in Azure Cosmos DB can be up to 2MB in size.

•**Properties:** A property is a name-value pair. **Each entity can include up to 252 properties to store data.** Each entity also has three **system properties that specify a partition key, a row key, and a timestamp.** Entities with the same partition key can be queried more quickly, and inserted/updated in atomic operations. An **entity's row key is its unique identifier within a partition.**



Describe non-relational data workloads

-  describe the characteristics of non-relational data
-  describe the types of non-relational and NoSQL data
-  recommend the correct data store
-  determine when to use non-relational data
-  Describe non-relational data offerings on Azure
-  identify Azure data services for non-relational workloads
-  describe Azure Cosmos DB APIs
-  describe Azure Table storage
-  describe Azure Blob storage
-  describe Azure File storage

blob storage concepts



Évolutivité, durabilité et disponibilité

Excellente durabilité, avec géoréplication et flexibilité pour une mise à l'échelle en fonction des besoins



Sécurité

Authentification auprès d'Azure Active Directory et contrôle d'accès en fonction du rôle (RBAC), plus chiffrement au repos et protection avancée contre les menaces



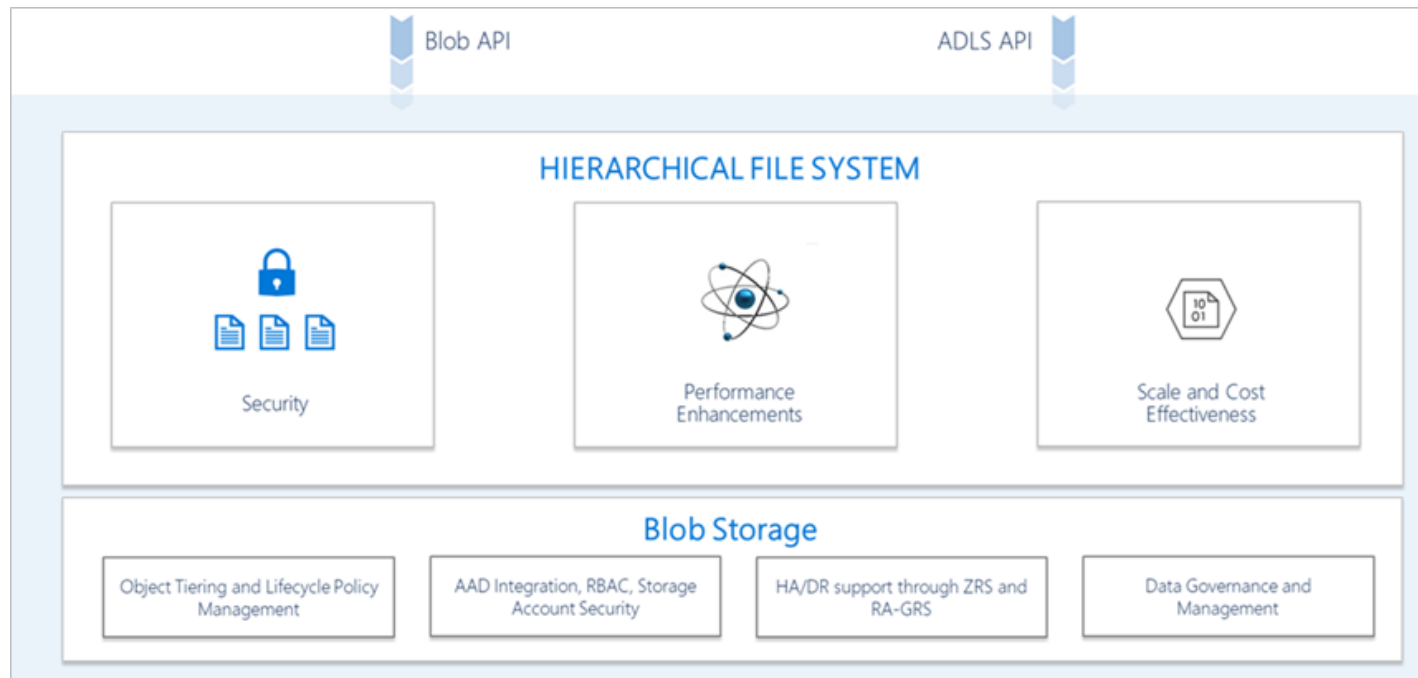
Optimisé pour les lacs de données

Espace de noms de fichier et prise en charge de l'accès multiprotocole permettant des charges de travail analytiques pour obtenir des insights sur les données













Gestion complète des données

Gestion du cycle de vie de bout en bout, contrôle d'accès basé sur des stratégies et stockage immuable (WORM)



Describe non-relational data workloads

-  describe the characteristics of non-relational data
-  describe the types of non-relational and NoSQL data
-  recommend the correct data store
-  determine when to use non-relational data
-  Describe non-relational data offerings on Azure
-  identify Azure data services for non-relational workloads
-  describe Azure Cosmos DB APIs
-  describe Azure Table storage
-  describe Azure Blob storage
-  describe Azure File storage

Azure File Storage



Partages de fichiers serverless

Système de fichiers distribués entièrement conçu pour être hautement disponible et durable (pas de machines virtuelles IaaS ni de fichier NAS masqués)



Coût total de possession optimisé

Différents niveaux sont utilisés pour adapter vos partages à vos besoins en matière de performances et à des prix compétitifs



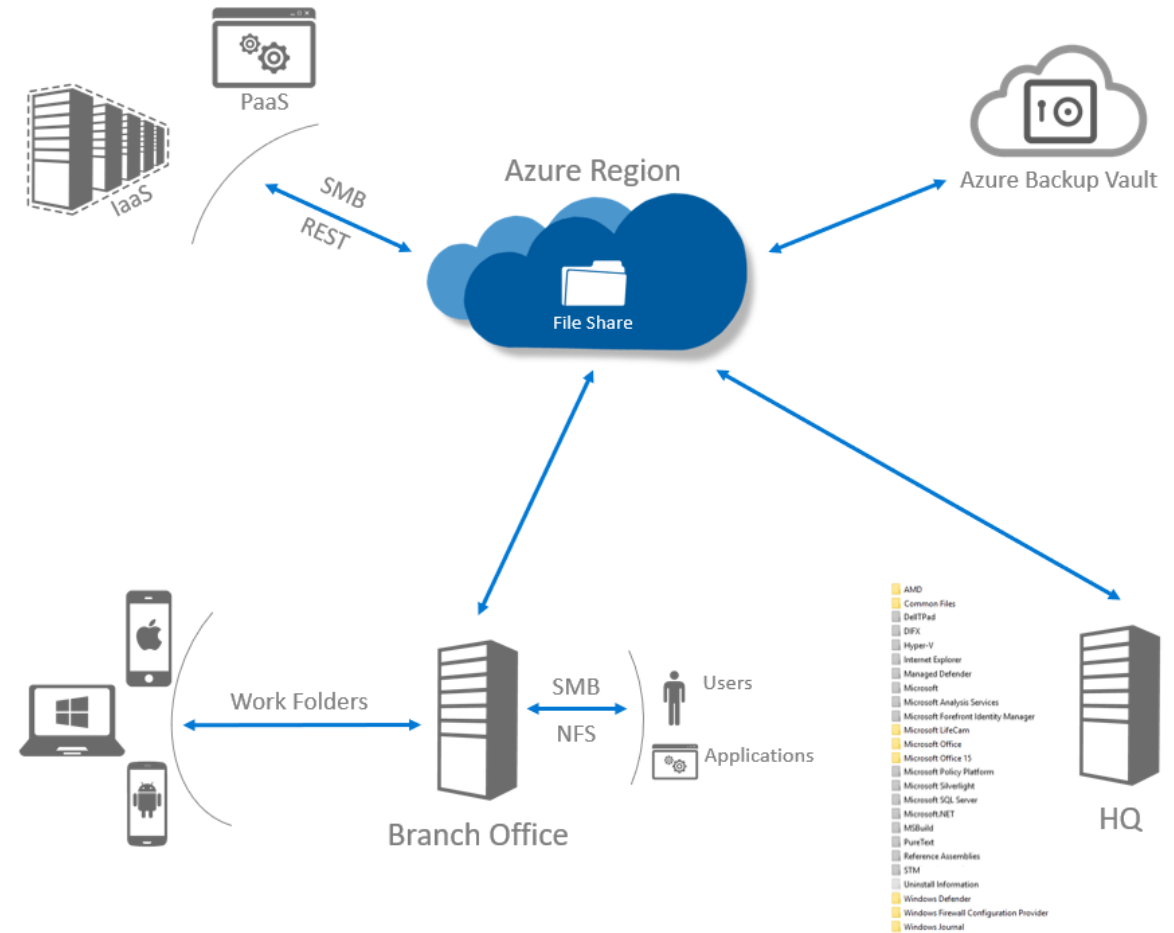
Conçu pour les scénarios hybrides avec File Sync

Transformez vos serveurs Windows en caches de vos partages cloud sans compromettre les performances locales



Prise en charge de plusieurs protocoles

Prend en charge plusieurs protocoles permettant de stocker les données à partir de différentes sources



Identify basic management tasks for non-relational data



describe provisioning and deployment of non-relational data services



describe method for deployment including the Azure portal, Azure Resource Manager templates, Azure PowerShell, and the Azure command-line interface (CLI)



identify data security components (e.g., firewall, authentication, encryption)

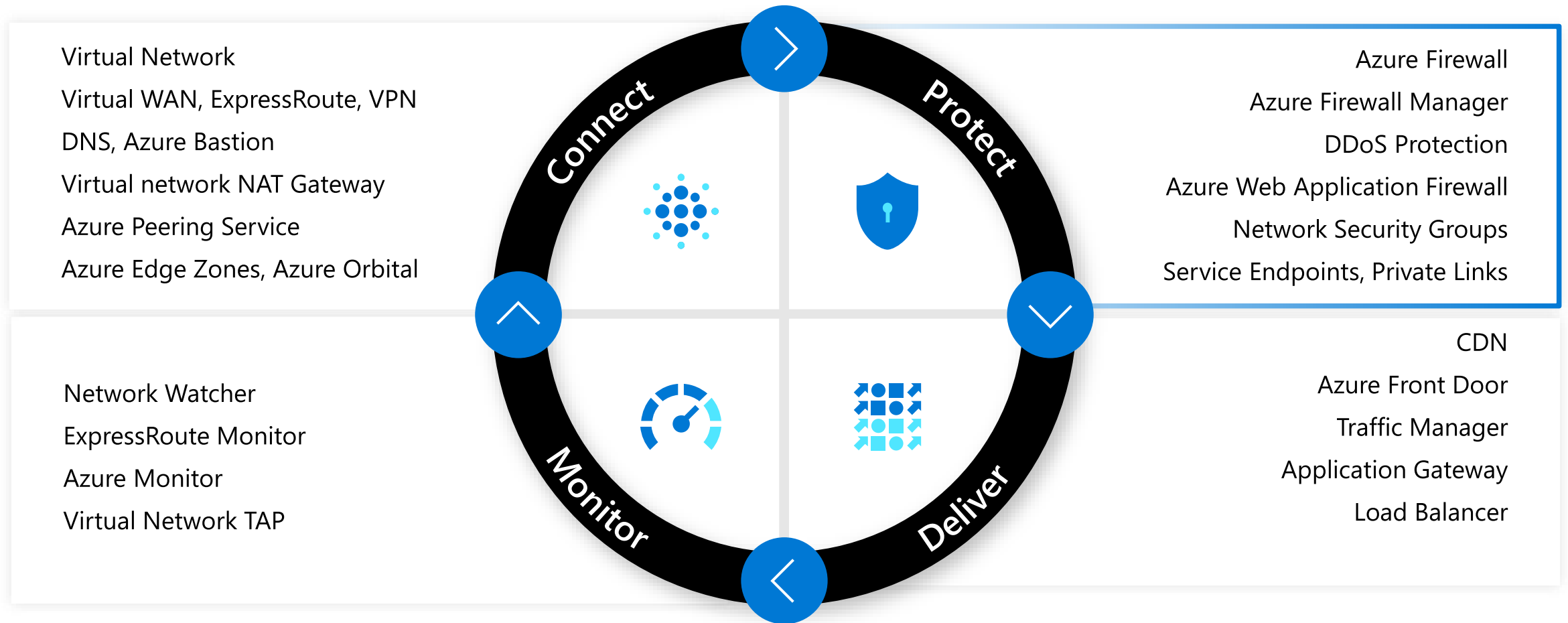


identify basic connectivity issues (e.g., accessing from on-premises, access with Azure VNets, access from Internet, authentication, firewalls)



identify management tools for non-relational data

Services réseau Azure



Services de protection réseau



DDoS protection

Protection DDOS adaptée aux modèles de trafic de vos applications



Web Application Firewall

Protection centralisée des applications Web entrantes contre les attaques et les vulnérabilités courantes



Azure Firewall

Filtrage réseau et application centralisé (L3-L7) sortant et entrant (non HTTP / S)



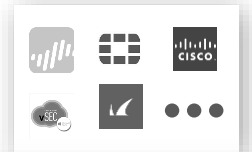
Network Security Groups

Filtrage distribué du trafic sur les réseaux entrants et sortants (L3-L4) sur une VM, un conteneur ou un sous-réseau



Private link

Limitez l'accès aux ressources de service Azure (PaaS) à votre réseau virtuel uniquement



Security Appliances

Tirez parti de vos compétences, processus et licences existants en ajoutant des technologies à partir de la Place de marché Azure

Protection des applications

Segmentation

Et plus...

Azure DDoS Protection Standard

Cloud scale DDoS protection for Azure

1

Azure global network

2

Adaptive tuning

3

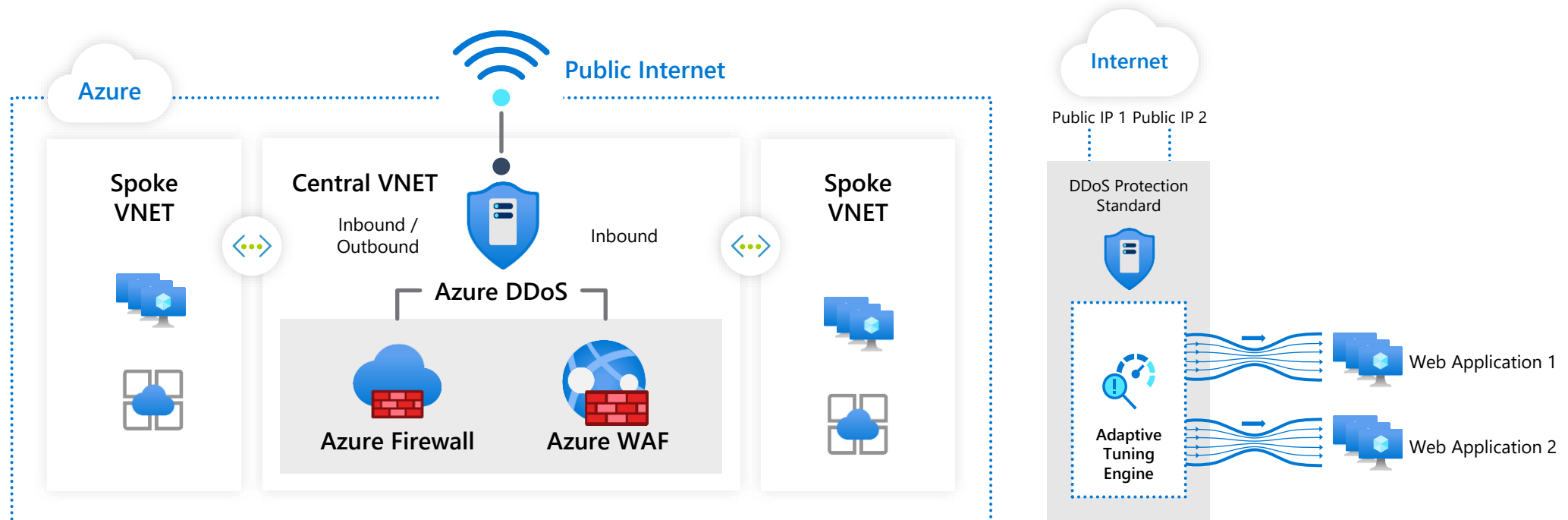
Attack analytics and metrics

4

DDoS Rapid Response (DRR)

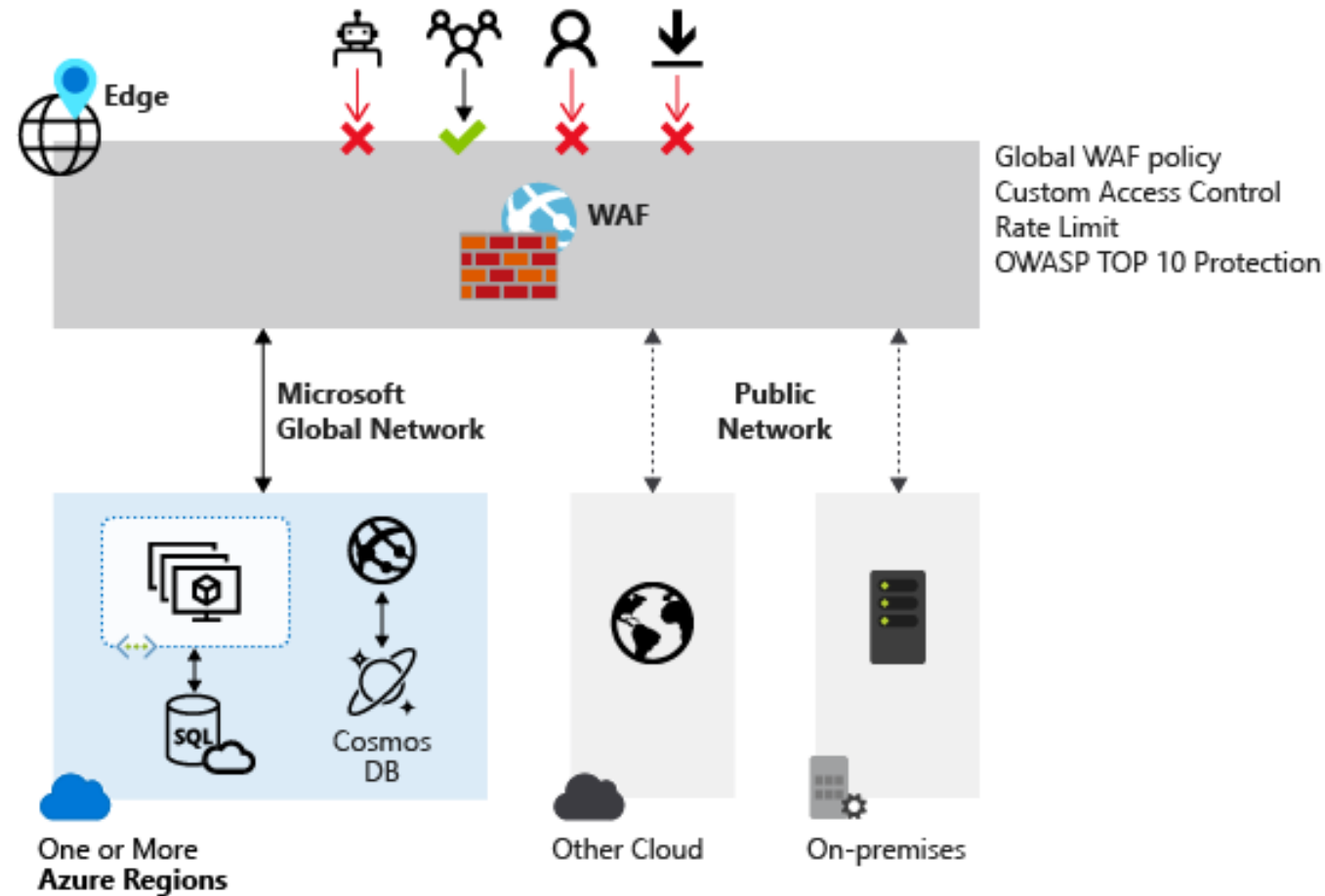
5

SLA guarantee and cost protection



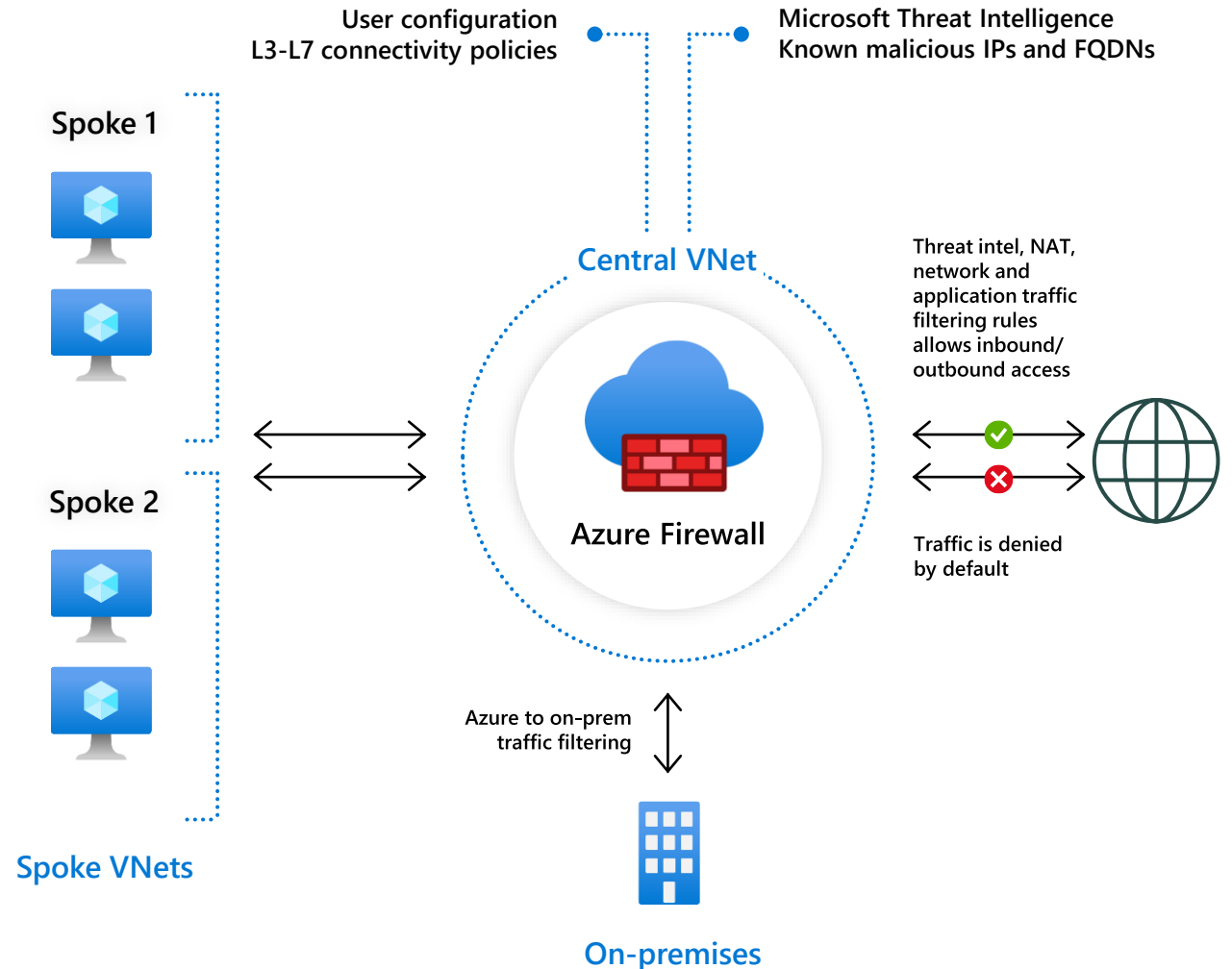
Web Application Firewall

- Une protection centralisée de vos applications Web contre les vulnérabilités et les exploits courants (SQL injection, cross-site scripting...)
- Une solution WAF peut réagir plus rapidement à une menace de sécurité en corrigeant de manière centralisée une vulnérabilité connue
- Service supporté : Azure Application Gateway, Azure Front Door et Azure CDN de Microsoft (Public preview)
- WAF possède des fonctionnalités personnalisées pour chaque service spécifique



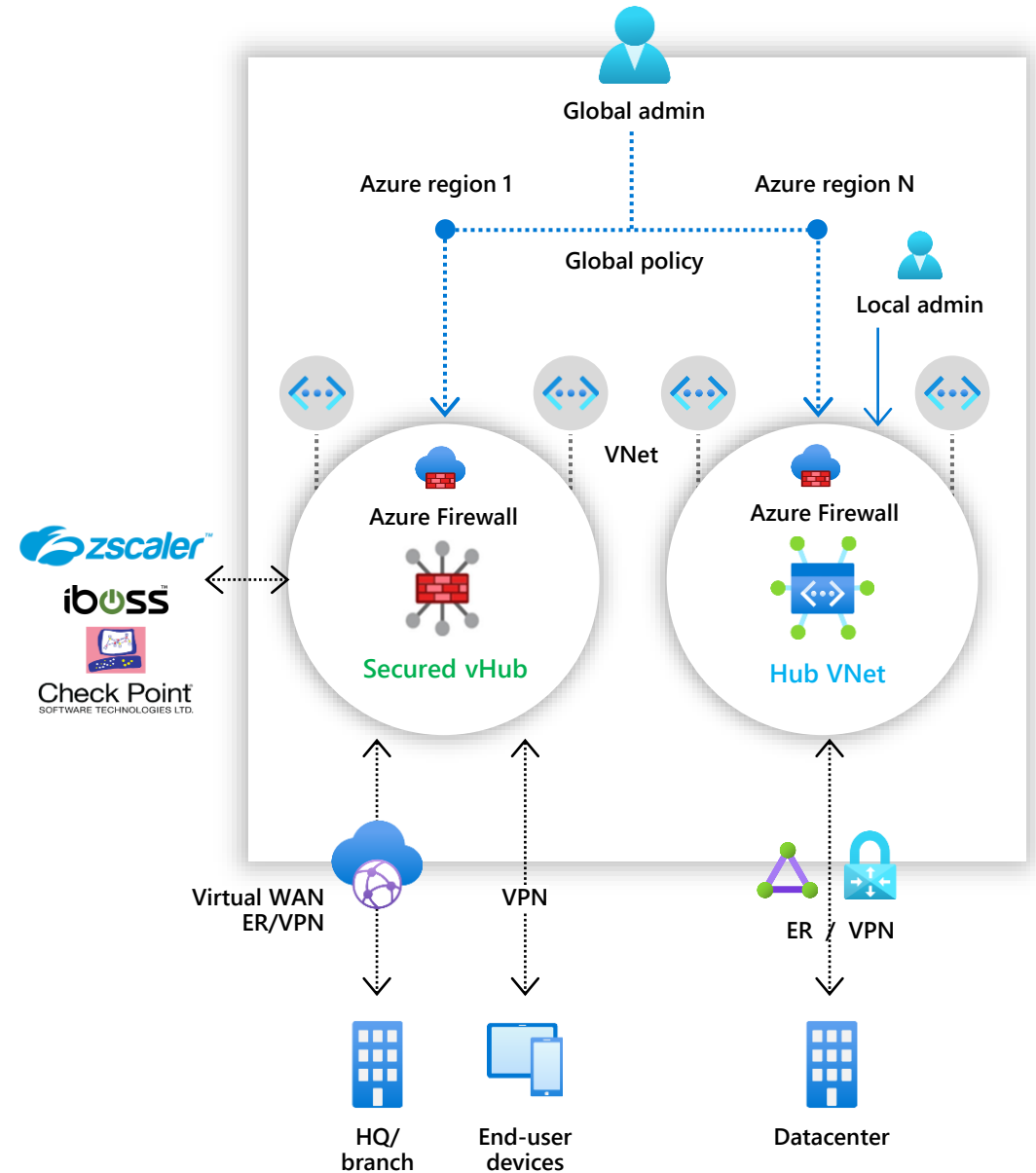
Azure Firewall

- Firewall as a Service
- Statefull (L3,4,7) hautement disponible et scalable
- Complète les NSG et autres mécanismes de sécurité
- Filtrage par FQDN des flux pour contrôler le trafic sortant HTTP(S)
- Gestion centralisée (multi abonnements) des règles de filtrage
- Passerelle SNAT pour les flux sortants. NAT entre le VNet et l'IP publique
- Journalisation dans Azure Monitor



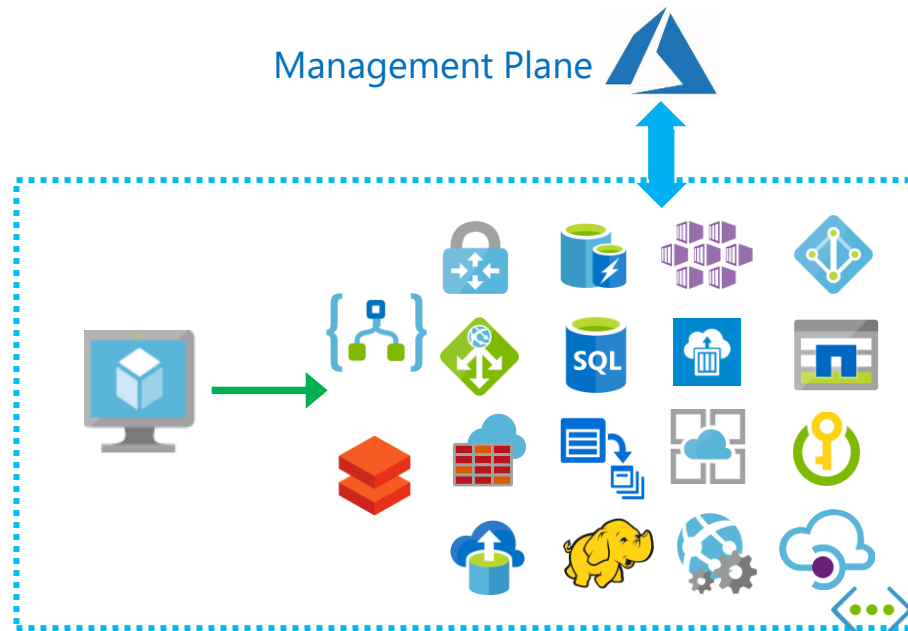
Azure Firewall Manager

- Un nouveau service qui fournit une **console centralisée** pour déployer et configurer plusieurs instances d'Azure Firewall
- Déployer les stratégies de pare-feu Azure via différents **Secure Virtual Hubs** et de **Azure Hub Vnet**
- Intégration avec des solutions tierces de SECurity as a Service
- Couvre différentes régions et différents abonnements Azure pour la gouvernance et la protection du trafic
- Est nécessaire pour gérer les hubs sécurisés (vWAN)



Modèles d'intégration réseau des services Azure

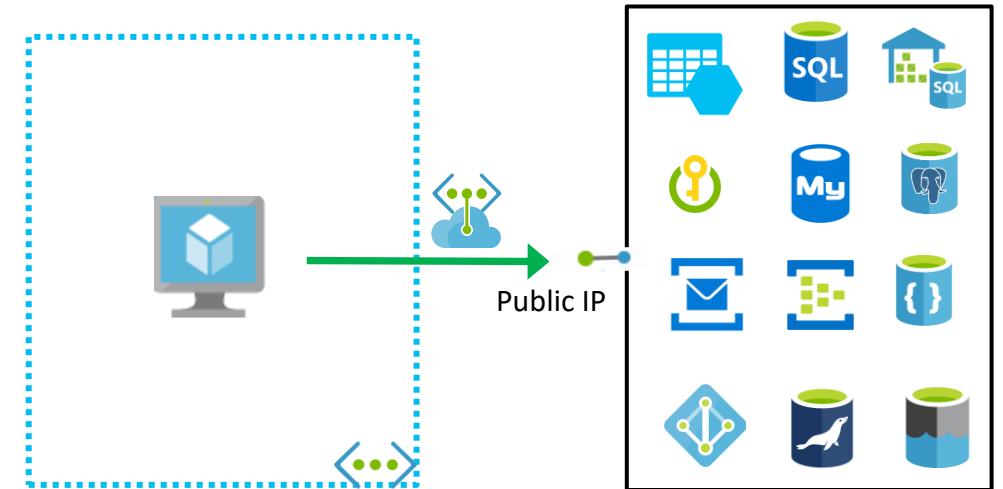
Instances de service exécutées
dans le VNet du client



Accès privé aux ressources PaaS
Gestion exposée à l'aide d'adresses IP publiques
Configuration réseau complexe

Services dédiés

Service s'exécutant en dehors du
VNet du client

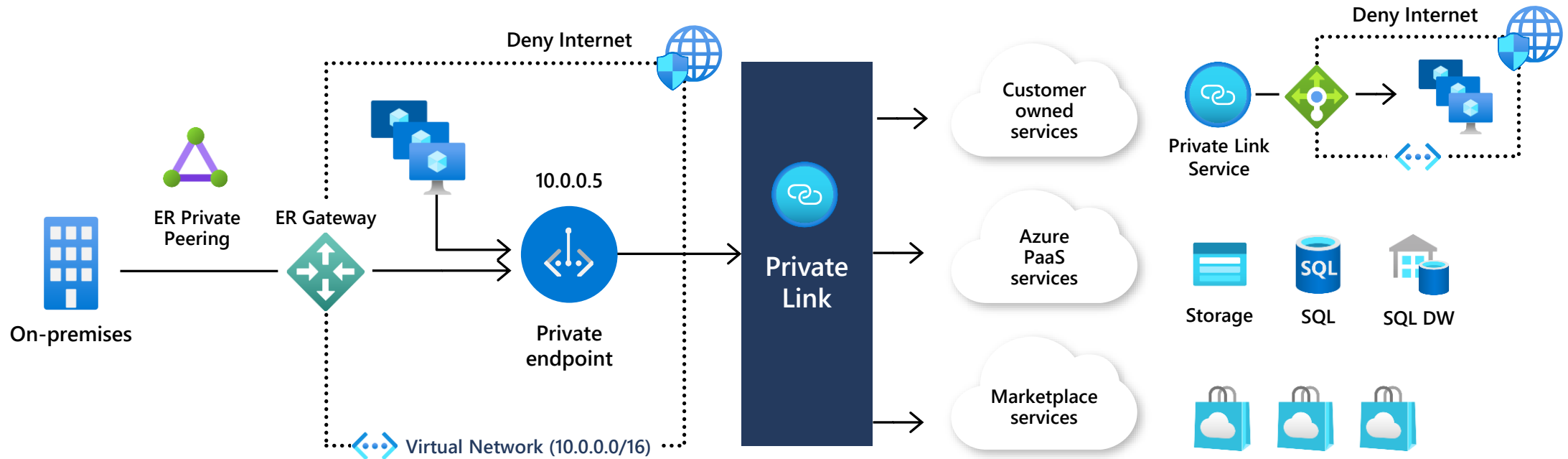


Accès public aux ressources PaaS
Plan de gestion isolé
Ressource PaaS sécurisée pour le VNet du client

Services partagés

Azure Private Link

Highly secure and private connectivity solution for Azure Platform



Private access from Virtual Network resources, peered networks and on-premise networks

In-built Data Exfiltration Protection

Predictable private IP addresses for PaaS resources

Unified experience across PaaS, Customer Owned and marketplace Services

Azure Bastion

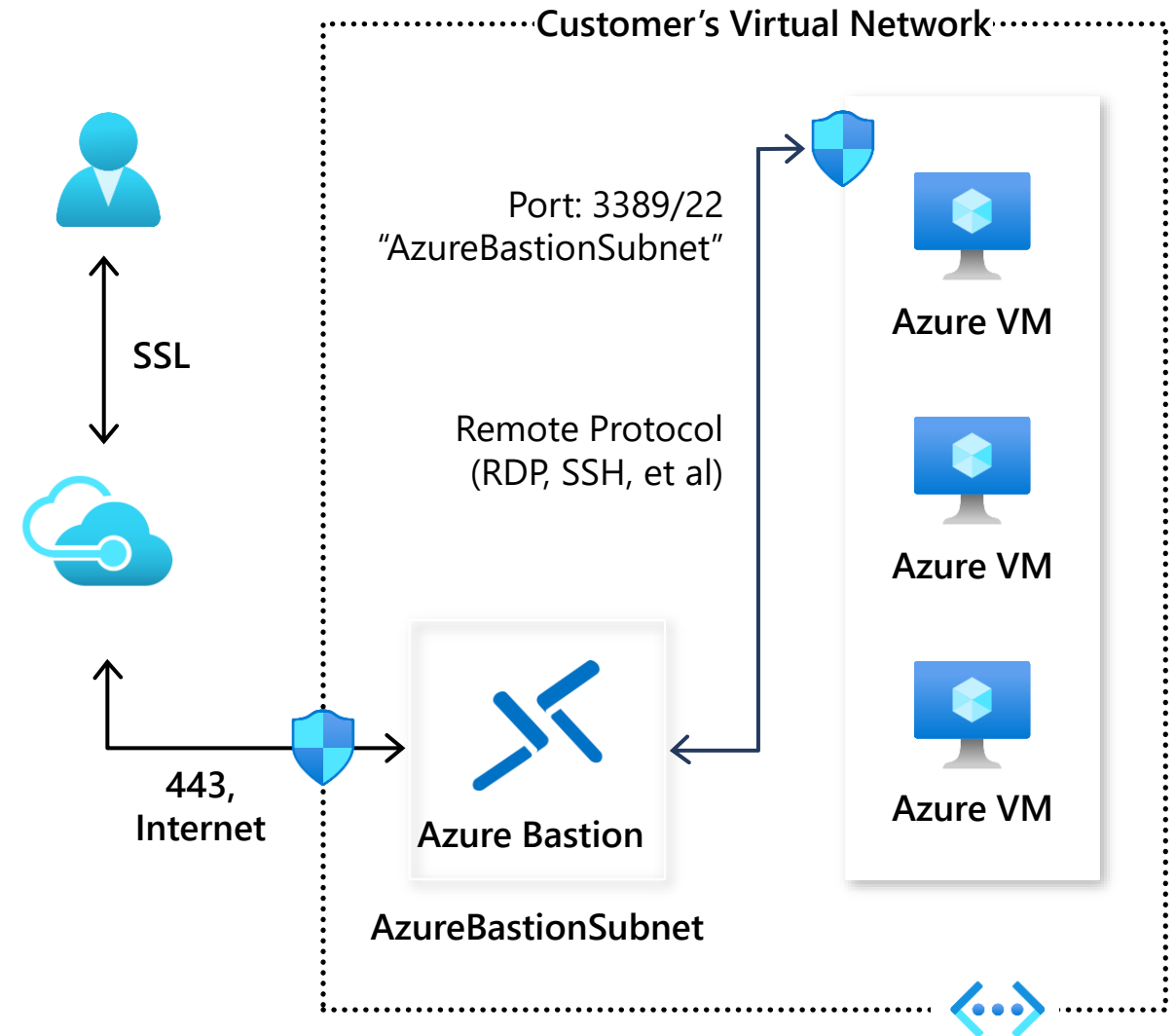
Secure and seamless RDP and SSH access to your virtual machines using zero trust

RDP/SSH to your workload using HTML5 standards-based web-browser, directly in Azure Portal

Resources can be accessed without public IP addresses

Supported Azure resources include VMs, VM Scale Sets, Dev-Test Labs

No agent required



How it all works together

