Cloud Computing

Caltech | Center for Technology & Management Education

Designing Infrastructure Solutions on Azure

Powerd by simplilearn

Design a Solution for Relational Data

A Day in the Life of an Azure Architect

You are an Azure architect for a company that needs a fully managed platform as a service (PaaS) database engine that can handle most database maintenance tasks like updating, patching, backups, and monitoring without requiring user interaction. They also want to make sure that the data is secure. They are also on the lookout for a relational database engine that is optimized for IoT and IoT Edge installations. The company is searching for a solution that can stream,
process, and analyze relational and non-relational data.

Finally, your organization has asked you to help them with a fully managed NoSQL database service for modern app development.

To achieve all of the above, along with some additional features, we will be learning a few concepts in this lesson that will help you find a solution to the given scenario.

Learning Objectives

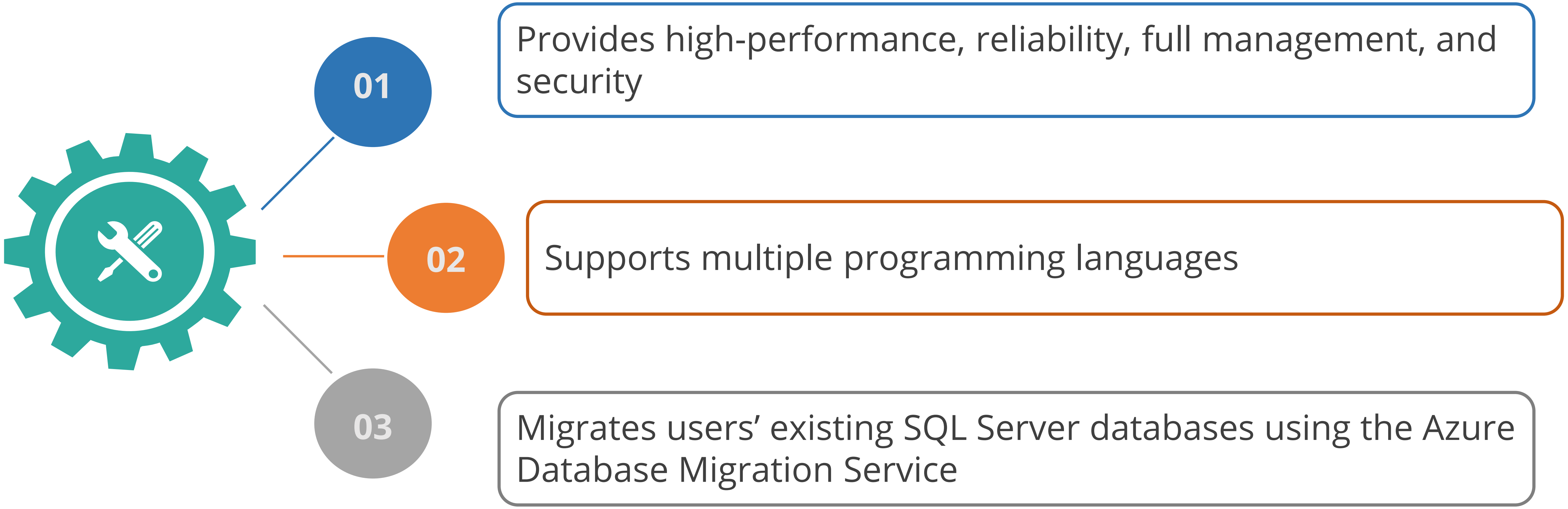
By the end of this lesson, you will be able to:

- Describe Azure SQL database
- Analyze an appropriate data platform based on requirements
- Illustrate Azure data storage
- Design for Azure SQL Edge and Azure Cosmos DB
- Implement design security for data

Select an Appropriate Data Platform Based on Requirements

Azure SQL Database

Azure SQL Database is a fully managed platform as a service (PaaS) database engine.



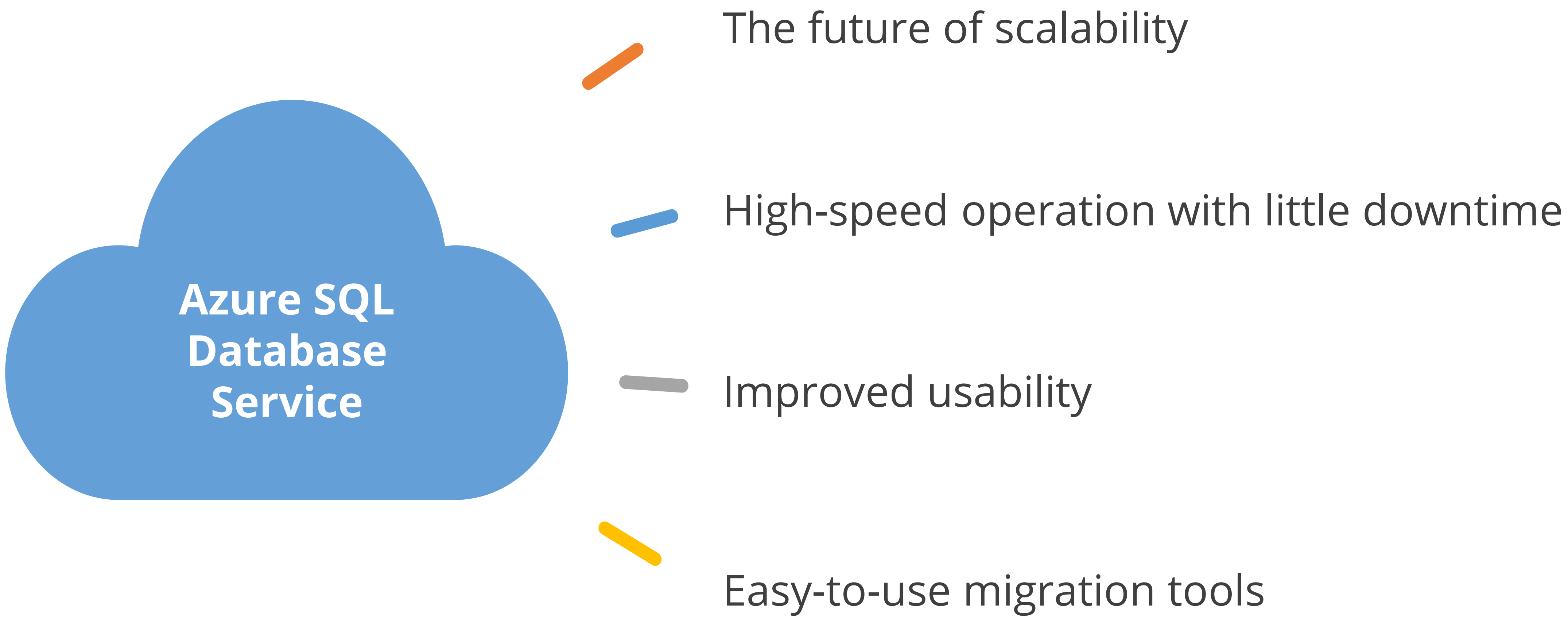
01 Provides high-performance, reliability, full management, and security

02 Supports multiple programming languages

03 Migrates users’ existing SQL Server databases using the Azure Database Migration Service

Azure SQL Database Service

The benefits of Azure SQL Database Service are:



Azure SQL Database Service

- The future of scalability
- High-speed operation with little downtime
- Improved usability
- Easy-to-use migration tools

Azure SQL Database

Azure SQL Database deployment options for a database:

01

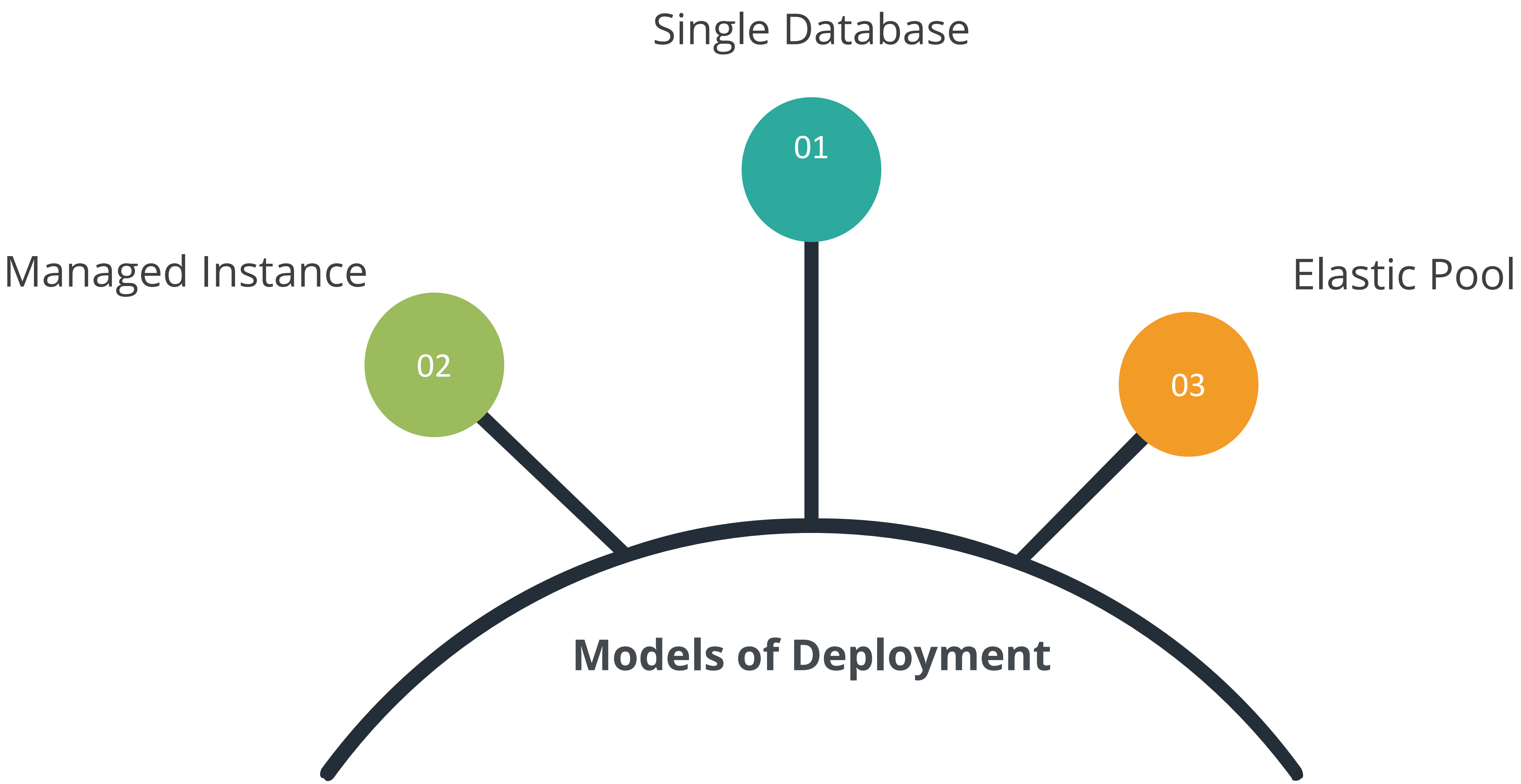
Single database represents a fully managed, isolated database.

02

Elastic pool is a collection of single databases with a shared set of resources, such as CPU or memory.

Deployment Models

There are three types of deployment models:



Deployment Models

Single Database

A single database is maintained by an SQL Database server and deployed on an Azure VM, and this is the simplest deployment method.

Managed Instance

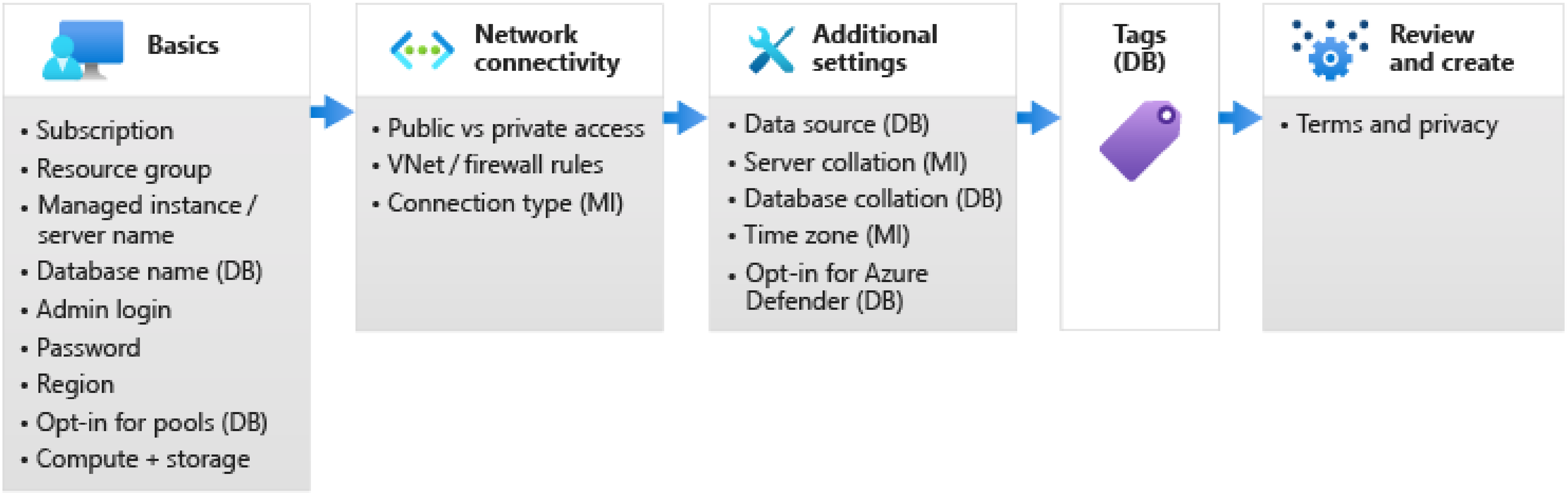
It is a database instance that is fully managed, which makes migrating on-premise SQL databases simple.

Elastic Pool

It is a collection of interconnected databases that pool resources, and individual databases can be added and removed from an elastic pool.

Deploy Azure SQL Database

There are five sections in the Azure portal to filled out during a deployment for Azure SQL Database and Azure SQL Managed Instance:



Basics

- Subscription
- Resource group
- Managed instance / server name
- Database name (DB)
- Admin login
- Password
- Region
- Opt-in for pools (DB)
- Compute + storage

Network connectivity

- Public vs private access
- VNet / firewall rules
- Connection type (MI)

Additional settings

- Data source (DB)
- Server collation (MI)
- Database collation (DB)
- Time zone (MI)
- Opt-in for Azure Defender (DB)

Tags (DB)

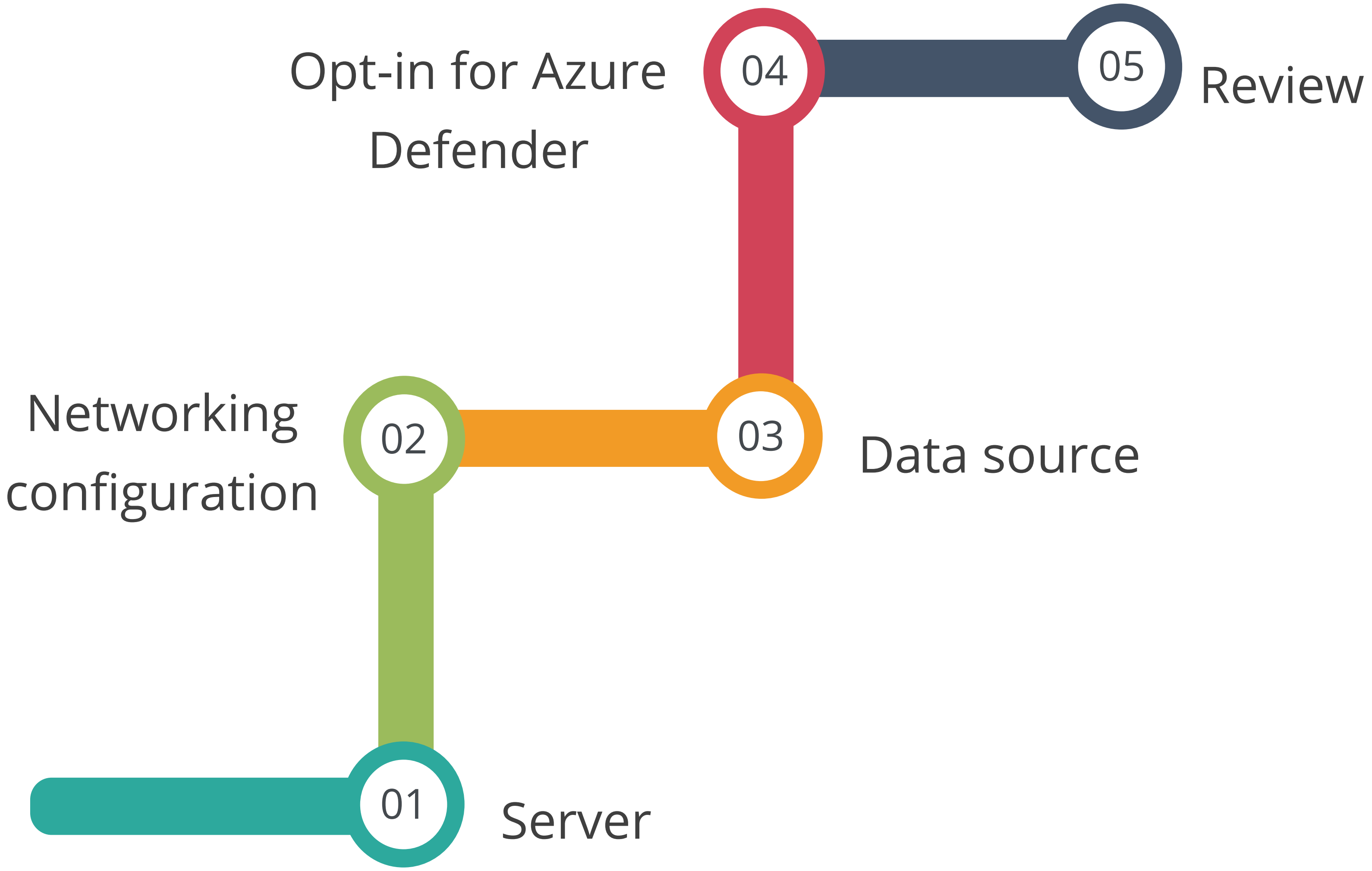
Review and create

- Terms and privacy

image source: https://docs.microsoft.com/en-in/

Deploy Azure SQL Database

These are the five options from the sections in the Azure portal:



01 Server

02 Networking configuration

03 Data source

04 Opt-in for Azure Defender

05 Review

Deploy Azure SQL Database

These are the explanations for the options:

Server

The name of the server must be unique across Azure.

Networking Configuration

A public endpoint can be allowed to link to a controlled instance over the internet.

Data Source

The user deploys the instance first and then the databases within it.

Opt-in for Azure Defender

After the instance has been deployed, the user can allow Azure Defender.

Review

The user can check the deployment choices in the Review section.

Recommending the Right Data Store

There are different options to choose from among SQL and NoSQL databases:

Relational database management systems

Key or value stores

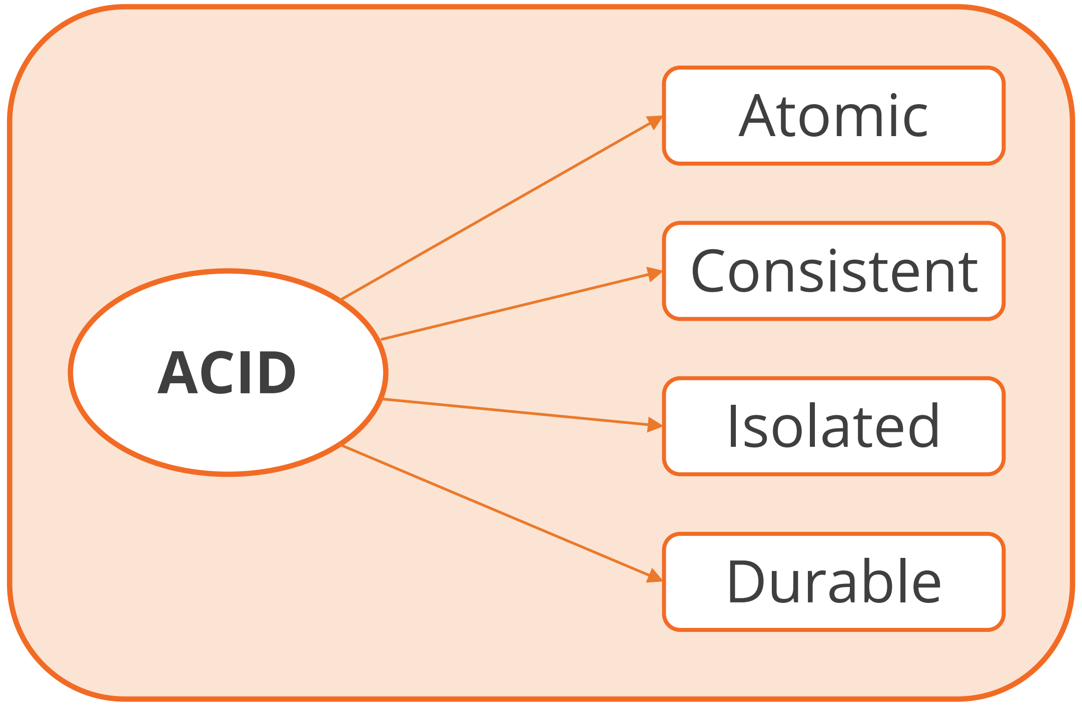
Different database systems:

Graph databases

Document databases

Relational Database Management Systems

Relational databases organize data as a series of two-dimensional tables with rows and columns.

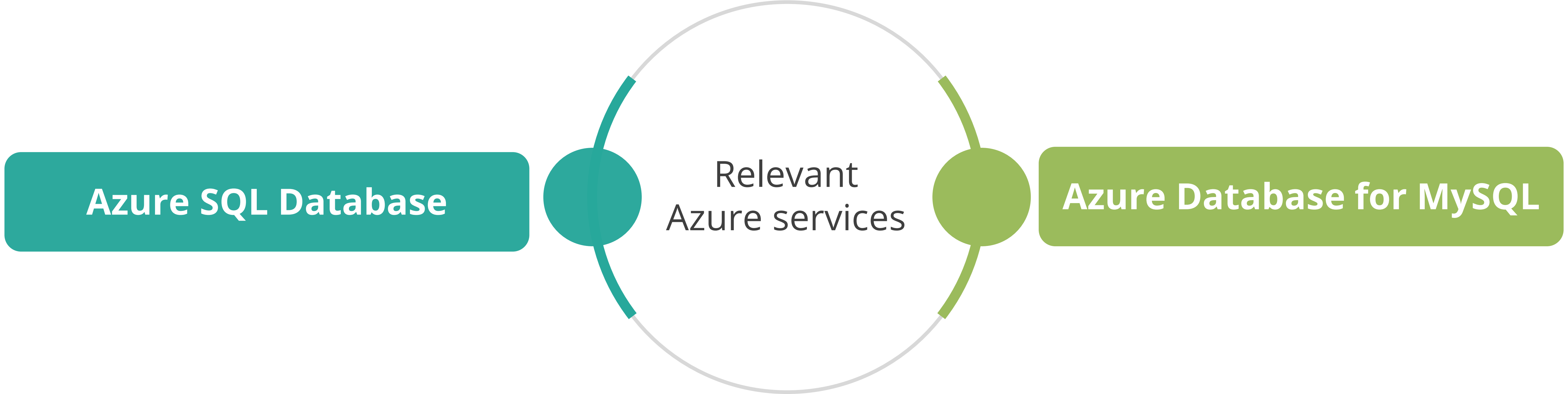


Each table has its own columns, and every row in a table has the same set of columns.

RDBMS implements a transactionally consistent mechanism that conforms to the ACID model for updating information.

Relational Database Management Systems

An RDBMS is useful when strong consistency guarantees are important.



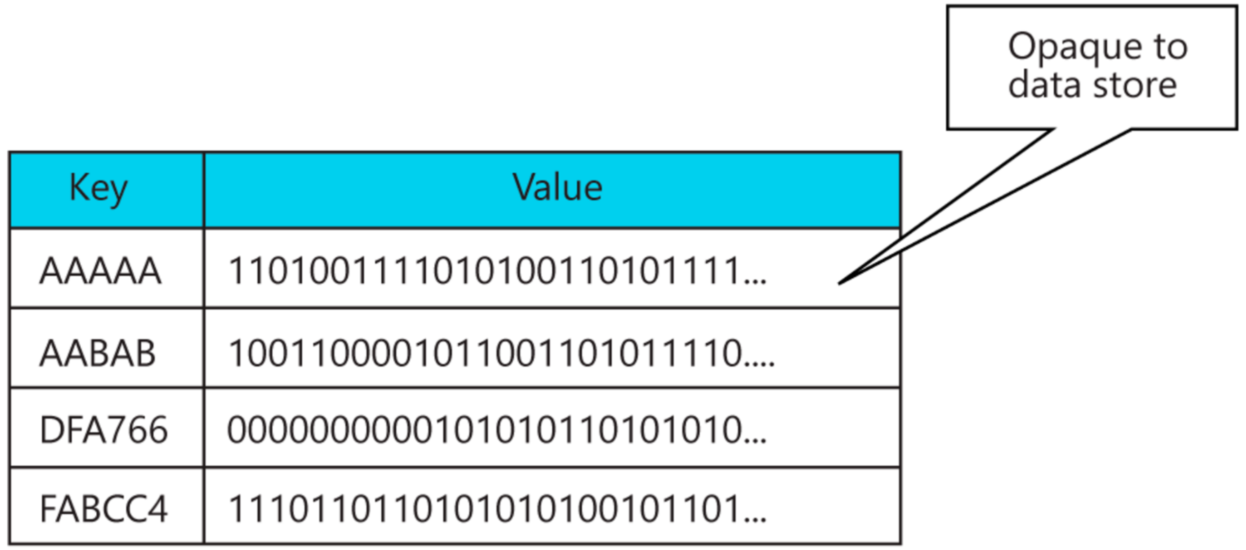
Azure SQL Database

Relevant Azure services

Azure Database for MySQL

Key or Value Stores

A key or value store is a large hash table.



Key	Value
AAAAA	1101001111010100110101111...
AABAB	1001100001011001101011110....
DFA766	0000000000101010110101010...
FABCC4	1110110110101010100101101...

Opaque to data store

- It associates each data value with a unique key.
- Most key or value stores only support simple query, insert, and delete operations.
- An application can store arbitrary data as a set of values.
- Some key or value stores impose limits on the maximum size of values.

Key or Value Stores

These are the features of key or value stores:

Cosmos DB

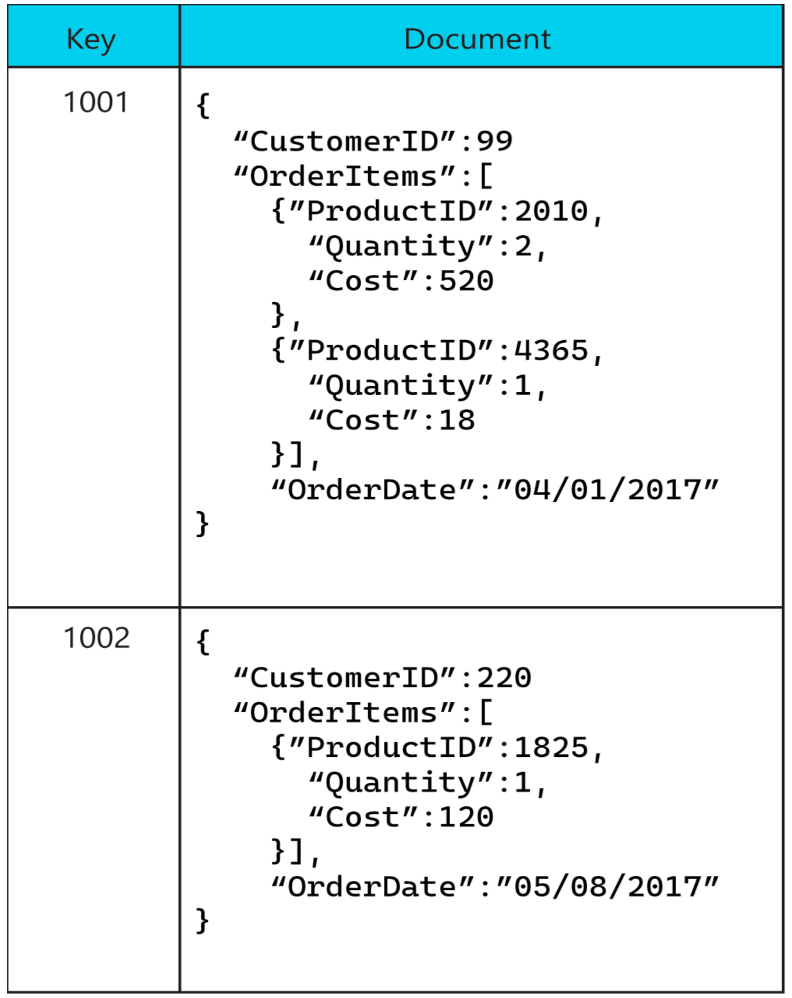
Azure Cache for Redis

Azure Tables

Document Databases

It is conceptually like a key or value store, except that it stores a collection of named fields and data.

Data fields of a document can be encoded, including XML, YAML, JSON, and BSON.



Key	Document

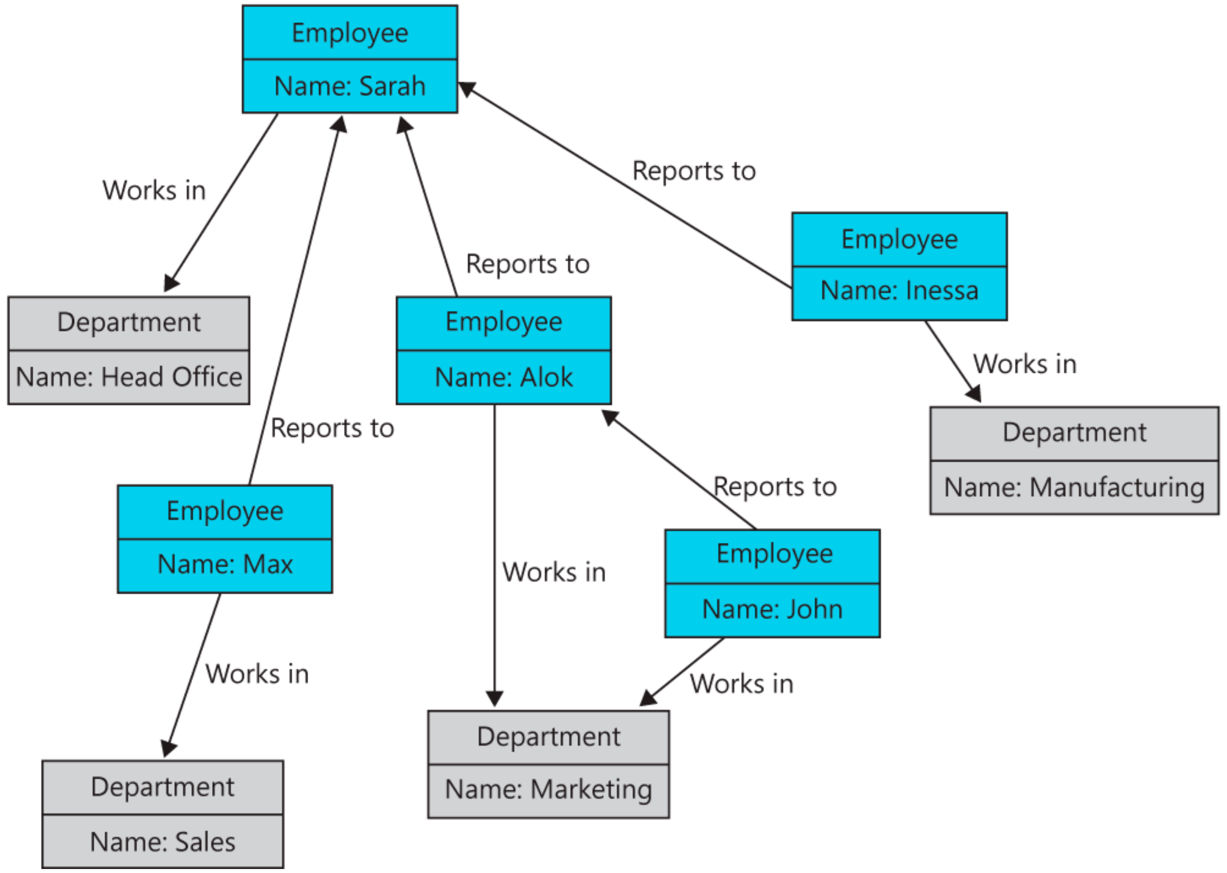
1001	{
	“CustomerID”:99
	“OrderItems”:[
	{”ProductID”:2010,
	“Quantity”:2,
	“Cost”:520
	},
	{”ProductID”:4365,
	“Quantity”:1,
	“Cost”:18
	}],
	“OrderDate”:”04/01/2017”
	}

1002	{
	“CustomerID”:220
	“OrderItems”:[
	{”ProductID”:1825,
	“Quantity”:1,
	“Cost”:120
	}],
	“OrderDate”:”05/08/2017”
	}

- A document store does not require that all documents have the same structure.
- **Relevant Azure services:** Cosmos DB

Graph Databases

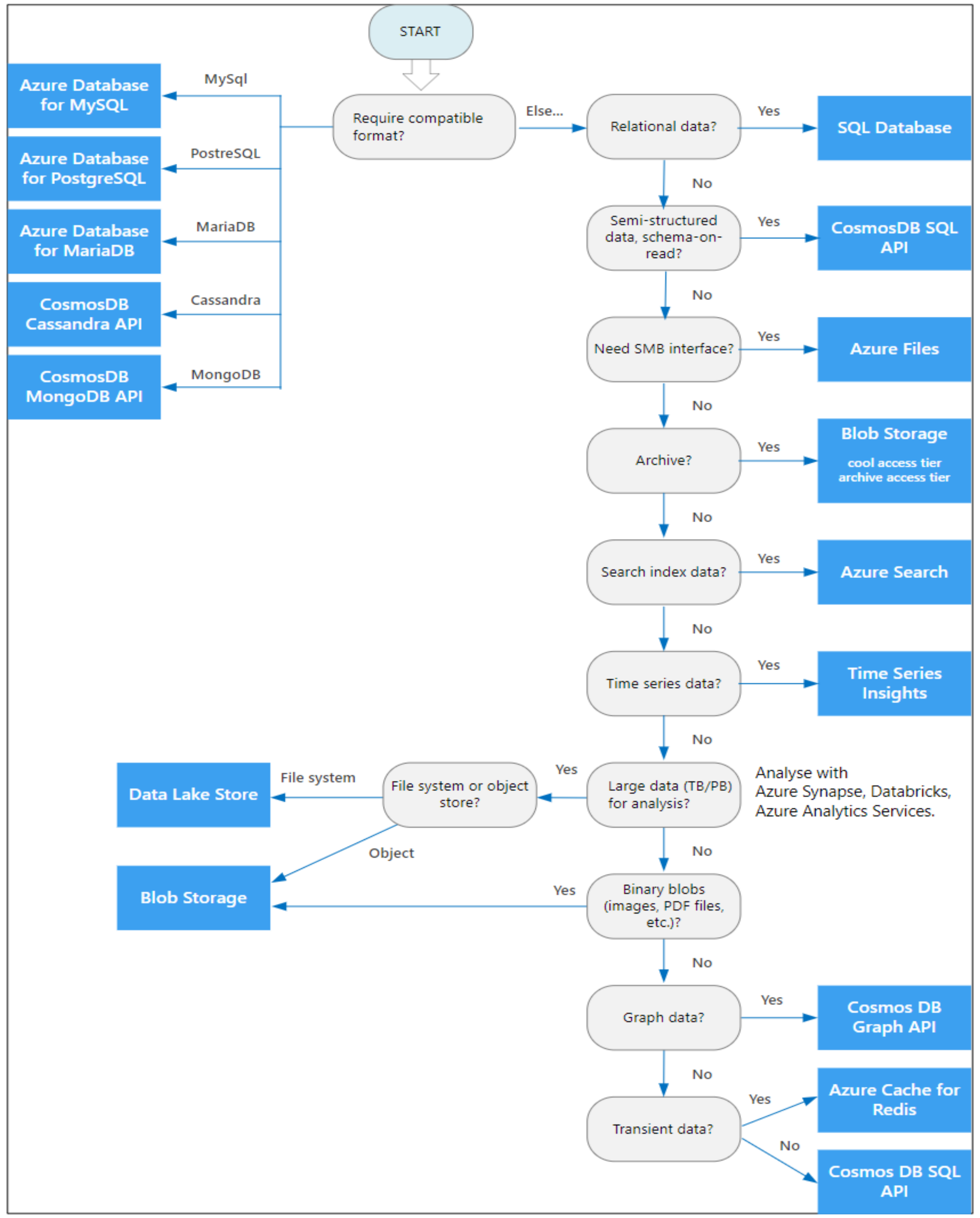
A graph database stores two types of information: nodes and edges.



- Nodes store data entities.
- Edges specify the relationships between nodes.
- Both nodes and edges can have properties that provide information about a particular node.
- Edges can also have a direction indicating the nature of the relationship.
- **Relevant Azure services**: Cosmos DB

Select a Candidate Data Store

Azure offers several managed data storage solutions; each one provides different features and capabilities.



- If the application consists of multiple workloads, evaluate each workload separately.
- A complete solution may incorporate multiple data stores.

Recommend Database Service Tier Sizing

Azure SQL Database and Azure SQL Managed Instance

The types of Azure SQL database & SQL managed instance service tiers are:

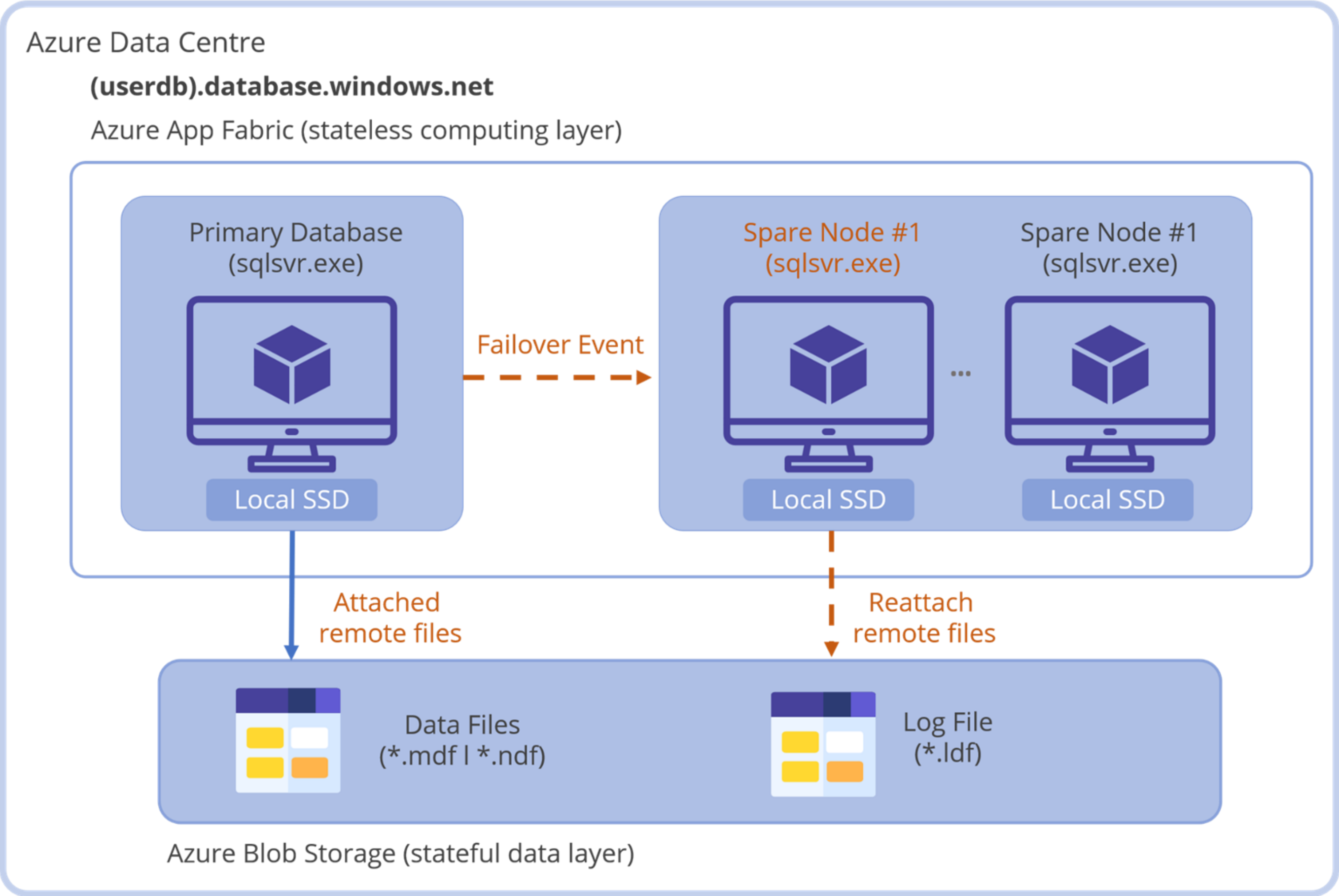
General purpose	Business critical	Hyperscale
Budget-oriented workloads	Low latency workloads	Business workloads

General Purpose Service Tier

This service tier comes with 99% availability.

System Diagram

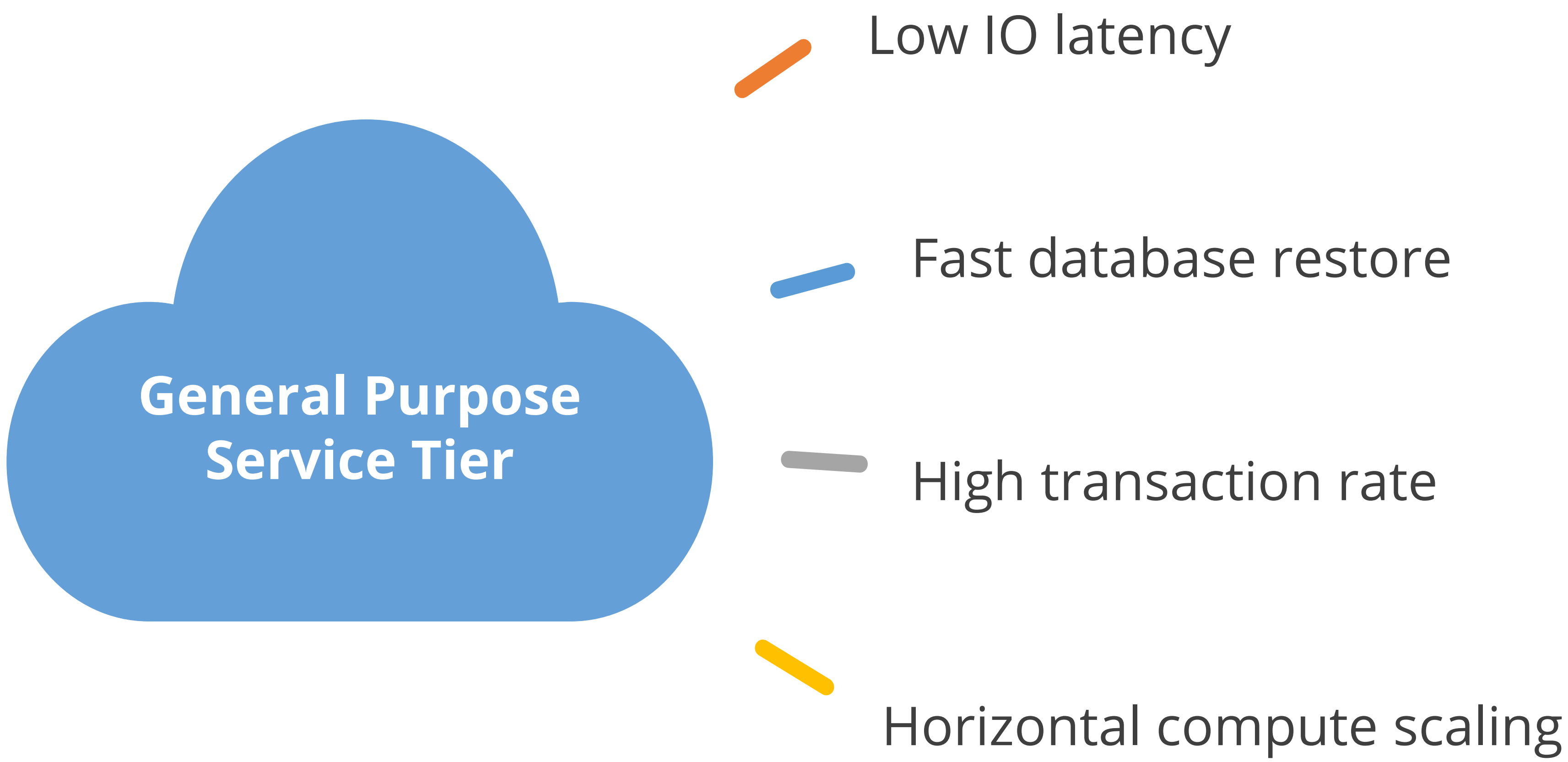
General Purpose – Azure SQL database



It is a default service tier that is suitable for most generic workloads.

General Purpose Service Tier

The benefits of General Purpose service tier are:

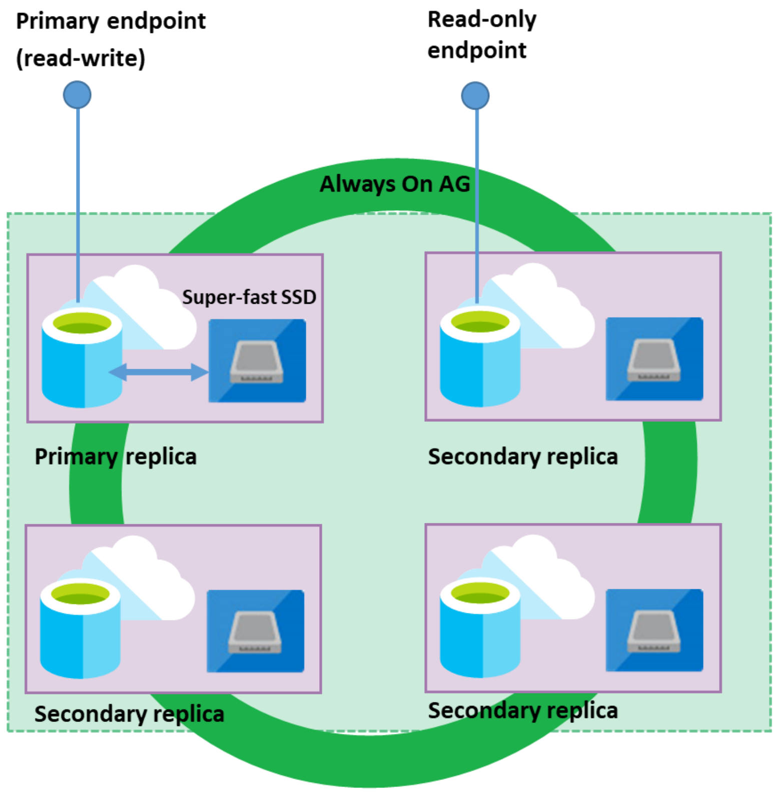


General Purpose Service Tier

- Low IO latency
- Fast database restore
- High transaction rate
- Horizontal compute scaling

Business Critical Service Tier

The Business Critical Service Tier is based on a cluster of database engine processes.

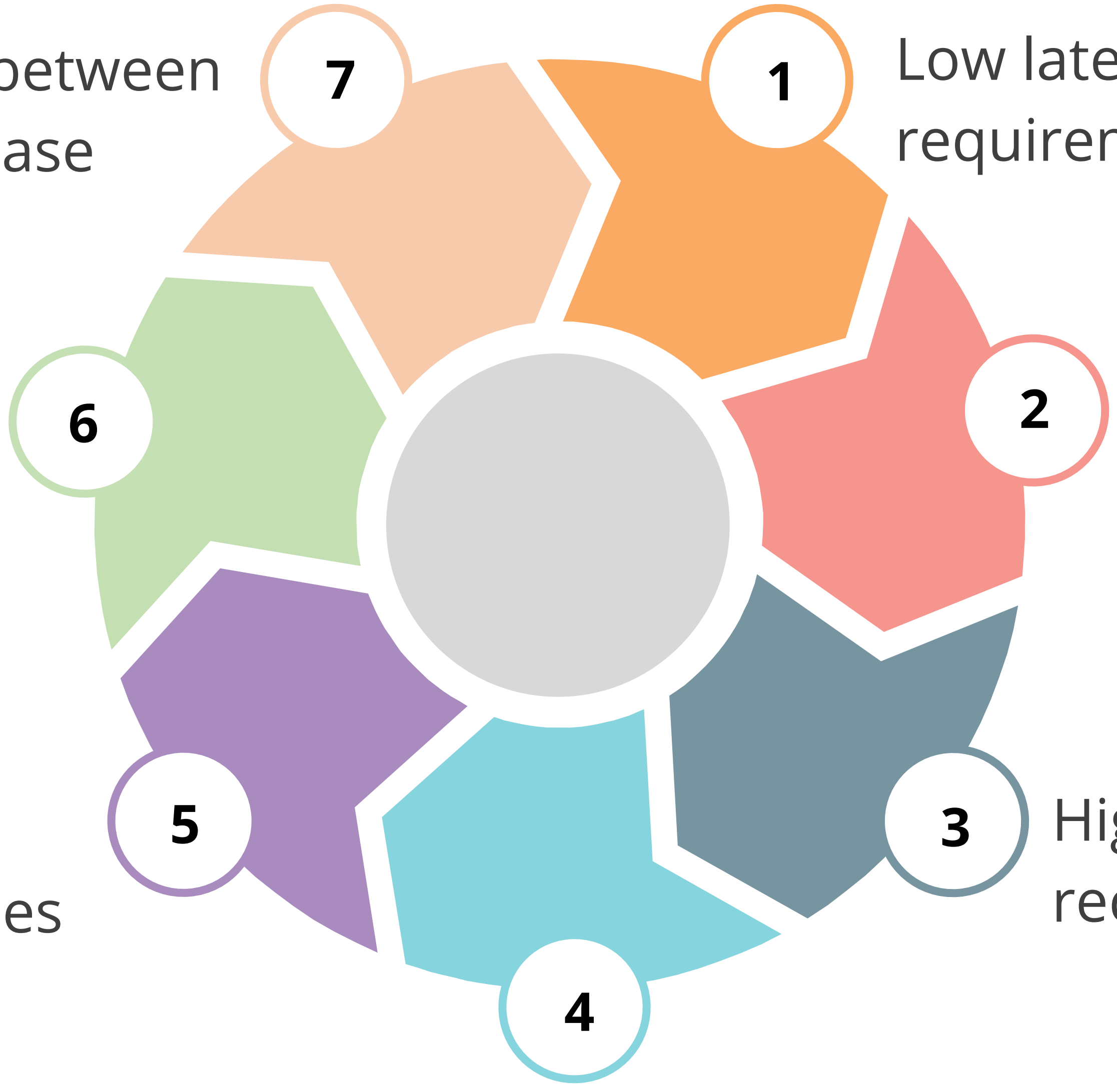


Business Critical service tier: collocated compute and storage

Premium availability is enabled in this service tier.

Business Critical Service Tier

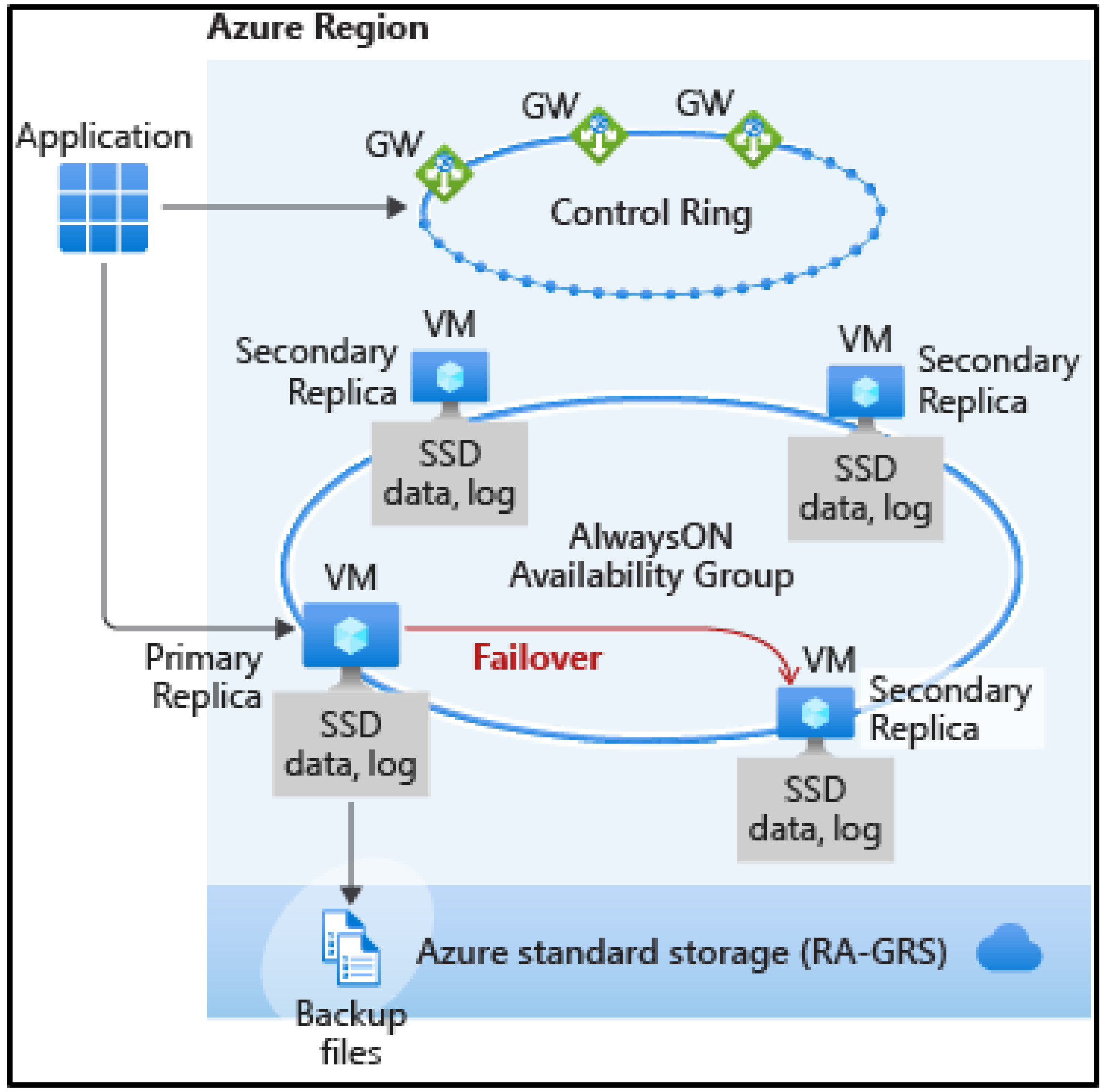
Business critical service tier can be chosen when there are:



1. Low latency requirements
2. Higher availability
3. Higher resiliency and faster recovery from failure
4. Fast geo-recovery
5. More time consumed in reporting and analytic queries
6. Long running transactions that modify data
7. Frequent communication between application and database

Business Critical Service Tier Availability

Business critical high availability service tier is created when a four-node cluster is formed.



Source: https://docs.microsoft.com/

Service Tier Comparison

The difference between the service tiers is given in the table below:

Availability	Resource type	General Purpose	Business Critical	Hyperscale
Compute size	SQL database	1 to 80 vCores	1 to 80 vCores	1 to 80 vCores
Storage size	SQL database	5 GB - 4 TB	5 GB - 4 TB	Upto 100 TB
TempDB size	SQL managed instance	24 GB per vCore	Upto 4 TB	N/A
In-memory OLTP	SQL managed instance	N/A	Available	N/A
Database size	SQL database	5 GB - 4 TB	Upto 100 TB	5 GB - 4 TB

Service Tier Comparison

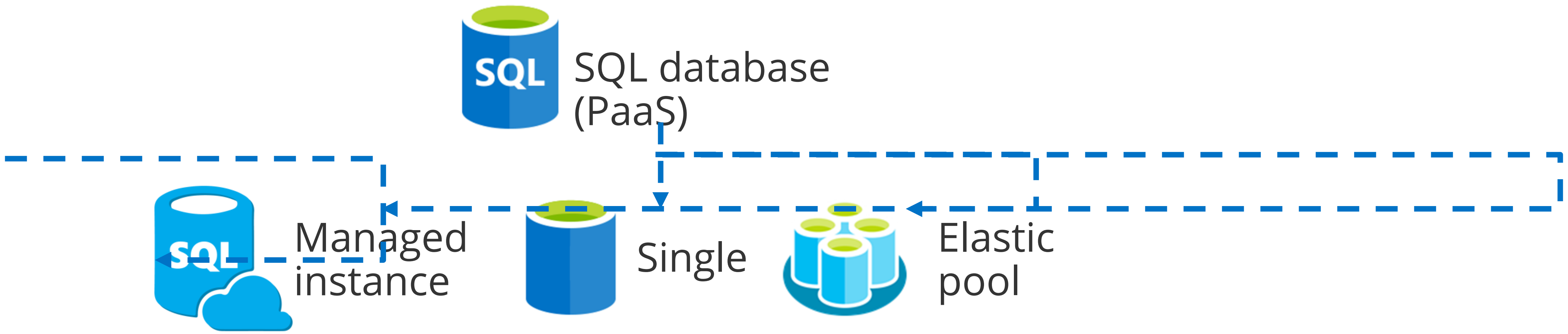
General Purpose	Hyperscale	Business Critical
Least expensive	Less expensive	Most expensive
Latency 5-10 milliseconds	Scales compute resources up and down very quickly	Latency 1-2 milliseconds
99.99 % availability	Instant backups and fast database restores	99.95 % availability
Maximum size 4TB	Maximum size 100TB	Local SSDs on four-node cluster

The service tier characteristics might be different in SQL database and SQL managed instance.

Recommend a Solution for Database Scalability

Dynamically Scale Azure SQL Database and Managed Instance

SQL database enables the users to change resources allocated to the databases.



SQL managed instance is a new deployment option that enables frictionless migration for SQL apps and modernization in a fully managed service.

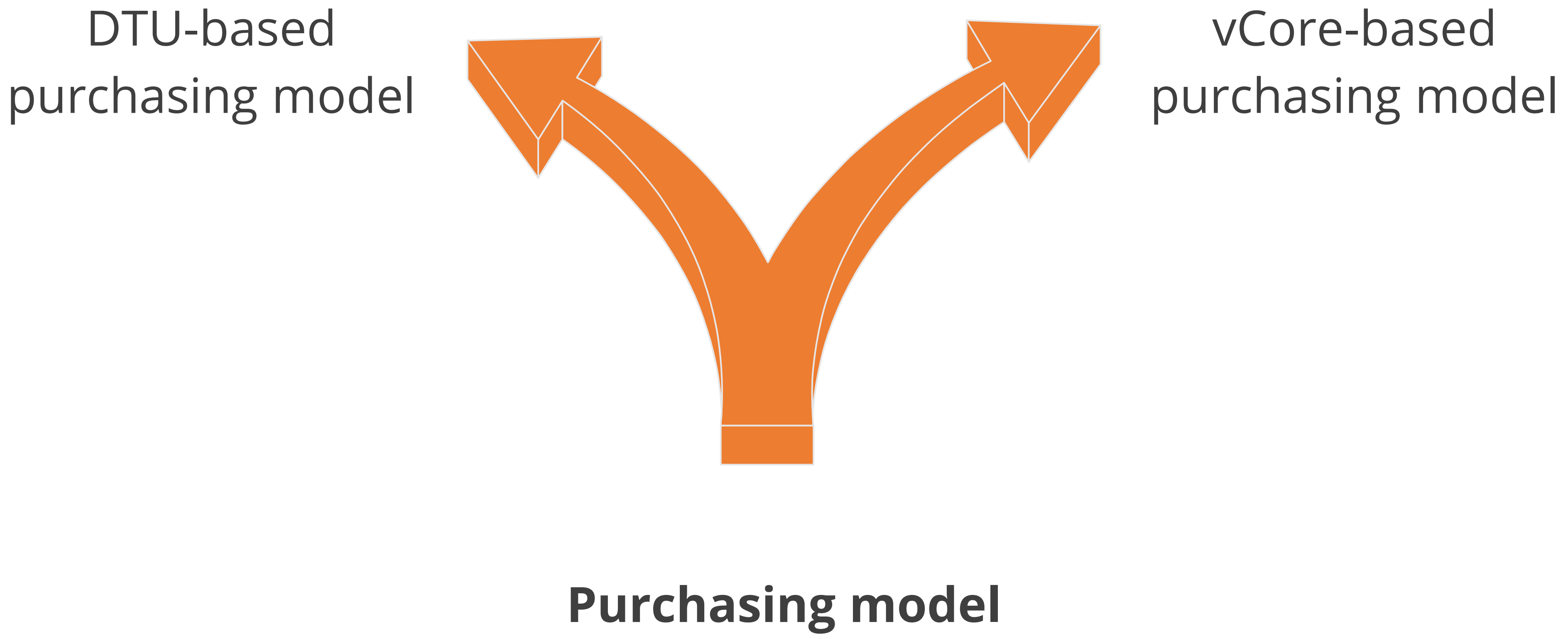
Dynamically Scale Azure SQL Database and Managed Instance

These are the benefits of Scaling:

- It is cost effective.
- It is more versatile.
- It mitigates performance issues.
- It handles the incoming workload well.

Azure SQL Database and Managed Instance Pricing

Azure SQL database and SQL managed instance offers two types of purchasing model:



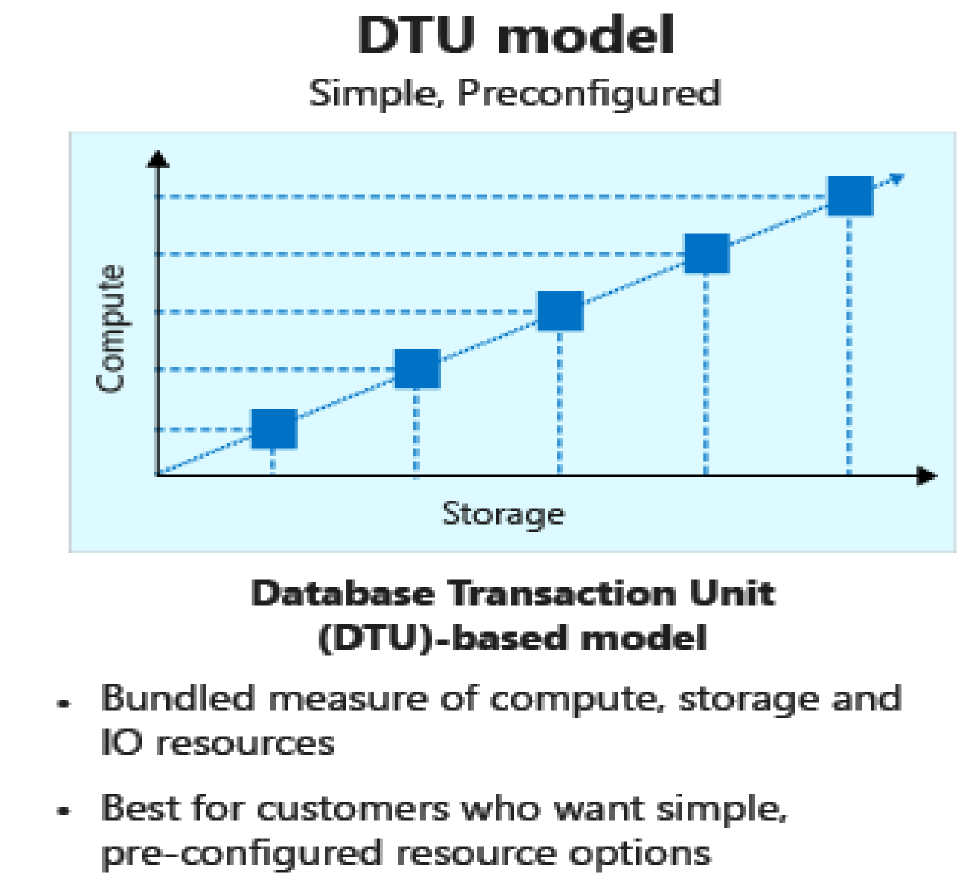
DTU-based purchasing model

vCore-based purchasing model

Purchasing model

The DTU-Based Purchasing Model

A Database Transaction Unit (DTU) is a composite metric that includes CPU, memory, reads, and writes.



DTU model

Simple, Preconfigured

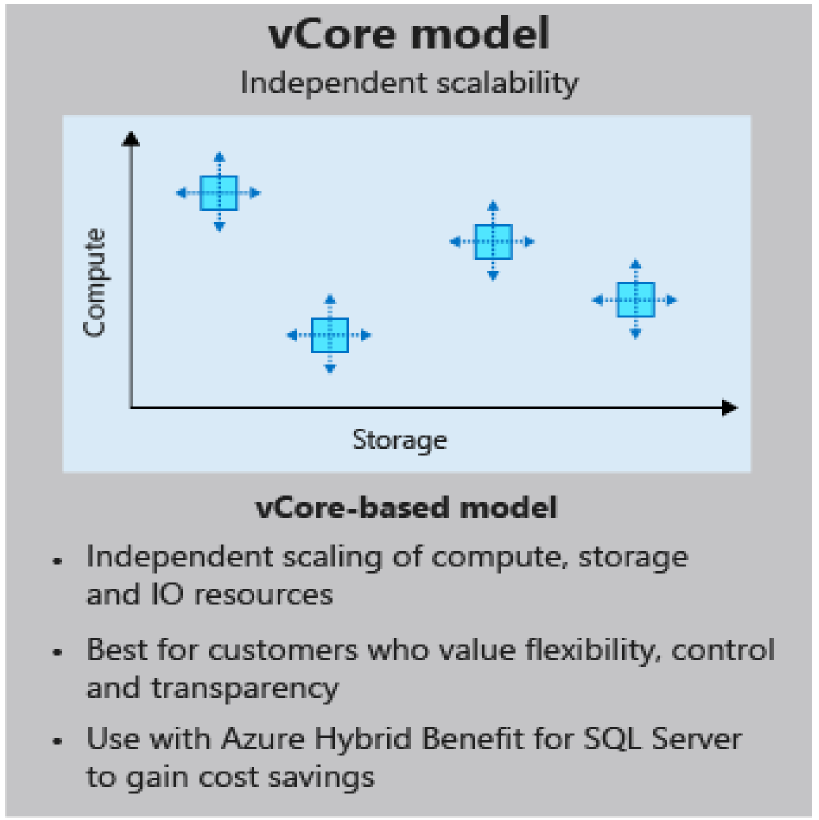
Database Transaction Unit (DTU)-based model

- Bundled measure of compute, storage and IO resources
- Best for customers who want simple, pre-configured resource options

The DTU-based purchase model provides a list of preconfigured compute resource bundles.

The vCore-Based Purchasing Model

The vCore-based model helps in optimizing price.



vCore model

Independent scalability

vCore-based model

- Independent scaling of compute, storage and IO resources
- Best for customers who value flexibility, control and transparency
- Use with Azure Hybrid Benefit for SQL Server to gain cost savings

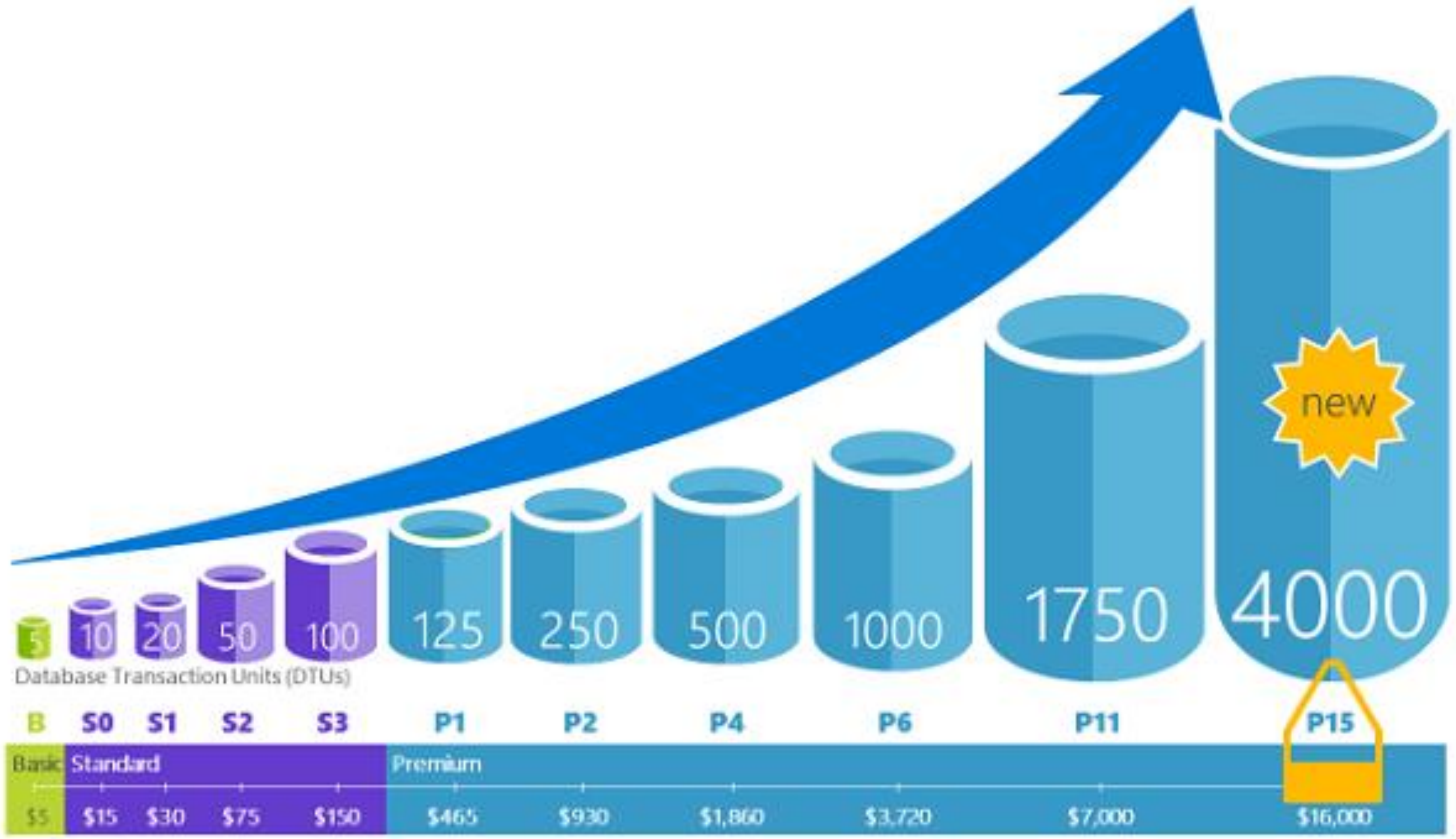
The vCore-based pricing strategy allows users to choose computing and storage resources individually.

Scaling Azure SQL Database and Managed Instance

Scaling the database can be done via the Azure portal using a slider.

Scale Single Databases in Azure SQL Database

Azure SQL Database allows the users to scale the databases dynamically.



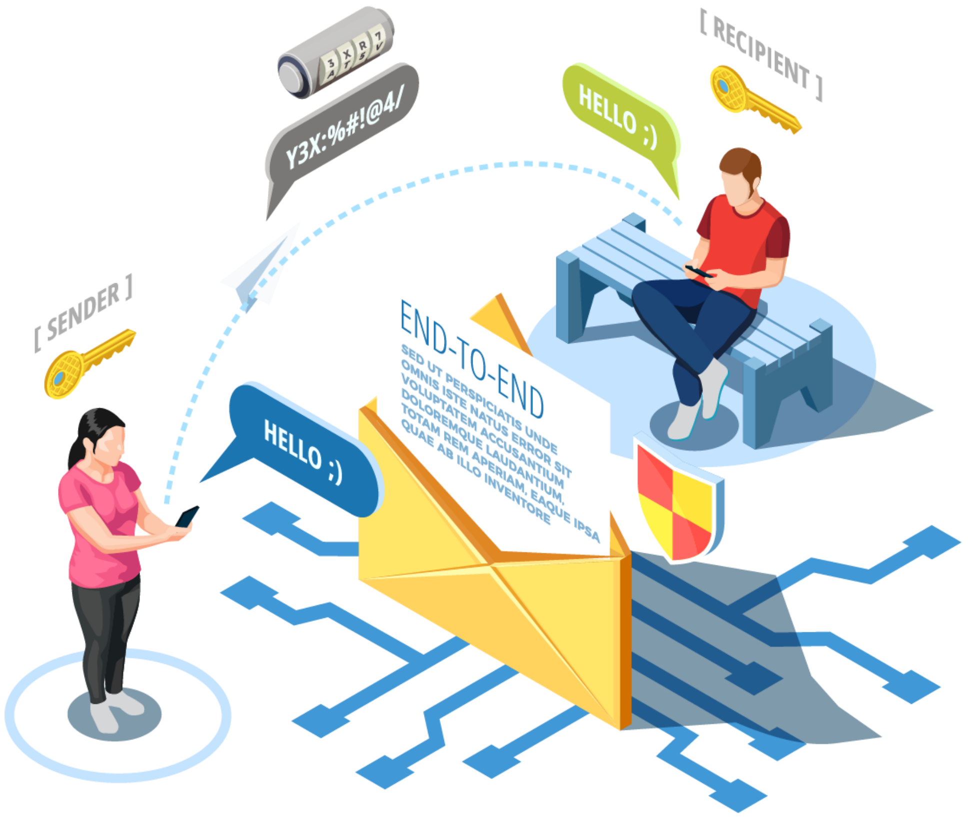
Scale Single Databases in Azure SQL Database

The user can determine the maximum number of resources that will be assigned to each database using either of the purchasing models.

The user can set a maximum resource limit for each group of databases in an elastic pool.

Transport Layer Security

Transport Layer Security(TLS) encrypts communication between a client application and an Azure Storage account.


Y3X:%#!@4/
[RECIPIENT]
HELLO ;)
[SENDER]
HELLO ;)
END-TO-END
SED UT PERSPICIATIS UNDE OMNIS ISTE NATUS ERROR SIT VOLUPTATEM ACCUSANTIUM DOLOREMQUE LAUDANTIUM, TOTAM REM APERIAM, EAQUE IPSA QUAE AB ILLO INVENTORE

TLS is a standard cryptographic protocol that assures client and service privacy and data integrity over the Internet.

Transparent Data Encryption

SQL Server, Azure SQL Database, and Azure Synapse Analytics data files are encrypted using transparent data encryption (TDE).

Encrypting data at rest is the terminology for this type of encryption.

Transparent Data Encryption

Precautions can be taken to secure a database, such as:

- Creating a safe system
- Encrypting sensitive information
- Constructing a firewall to protect the database servers

Azure Security Audit

The audit methodology is based on Microsoft's own best practice recommendations for Azure, and it primarily covers the following topics:

- Improve user’s Microsoft Defender for Cloud security score
- Use an industry accepted benchmark to assess an organization's current security posture

Azure Security Audit

- Conduct internal and external compliance audits on a regular basis, as well as regulatory compliance attestations
- Examine the policy's specifications
- For a comprehensive perspective of the technical Azure Governance setup, use Azure Governance Visualizer

Assisted Practice

Azure SQL DB **Duration: 10 Min.**

Problem Statement:

You've have been assigned a project to construct an Azure SQL DB that will automate most database maintenance tasks such as updating, patching, backups and monitoring without user involvement.

Assisted Practice: Guidelines

Steps to create an Azure SQL DB are:

1. Login to your Azure portal
2. Search and select Azure SQL
3. Select the SQL database tile
4. Fill in the required fields and create SQL DB

Design Security for Data

Design Security For Data

Data on the Azure platform is always encrypted, and Azure provides customers with strong data security, both by default and as customer options.

Data at Rest

Data in Transit

Data in Use

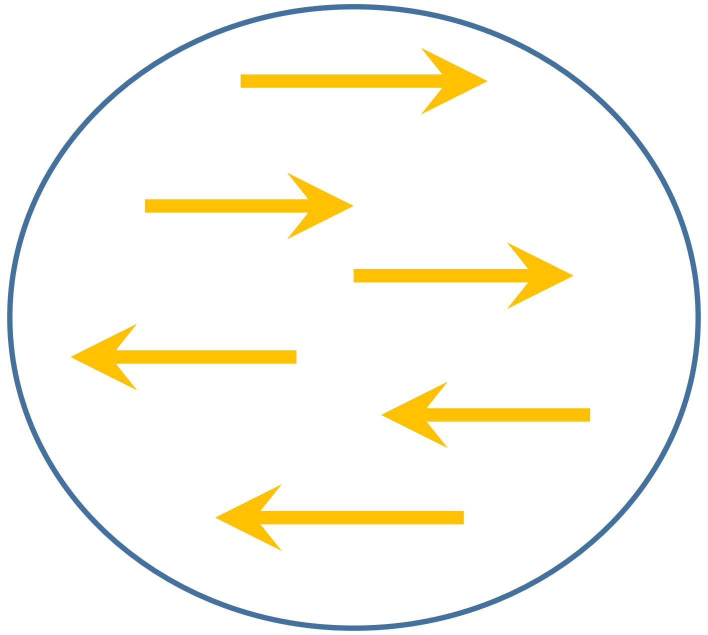
Data at Rest

Data at rest is static data stored on hard drives that are archived or not frequently accessed or modified, and the protection at rest aims to secure the inactive data stored on devices or networks.

- Available for the services across the software as a service (SaaS)
- Available for the platform as a service (PaaS)
- Available for the infrastructure as a service (IaaS) cloud models

Data in Transit

Data in transit is the data actively moving from one location to another, such as across the internet or through a private network.



- Data loss prevention (DLP) technology
- Backup and recovery solutions
- Identity and access management technology

Data in Use

Data in use is to protect the confidentiality and integrity of data while it is in use, in addition to existing protection to encrypt data in transit and at rest.

Available on a new DC-series of virtual machines

Design for Azure SQL Edge

Azure SQL Edge

Azure SQL Edge is an optimized relational database engine geared for IoT and IoT Edge deployments.

- It provides capabilities to create a high-performance data storage and processing layer for IoT applications.
- Azure SQL Edge can stream, process, and analyze relational and non-relational data such as JSON, graph, and time-series data.

Azure SQL Edge

Azure SQL Edge provides the same Transact-SQL (T-SQL) programming surface area that makes the development of applications or solutions easier and portability among IoT Edge devices, data centers, and the cloud straightforward.

Advanced Security of Azure SQL Edge

The features that provide advanced security of Azure SQL Edges are:

- Transparent data encryption and Always Encrypted features that provide data security
- Role-based access control (RBAC) and attribute-based access control (ABAC)
- Compliance with security rules with the help of data classification

Supported Features of Azure SQL Edge

The supported features of Azure SQL Edge are:

1

SQL streaming based on the same engine that powers Azure Stream Analytics and provides real-time data streaming capabilities in Azure SQL Edge

2

The T-SQL function call **Date_Bucket** for Time-Series data analytics

3

Machine learning capabilities through the ONNX runtime, included with the SQL engine

Unsupported Features of Azure SQL Edge

The Unsupported features of Azure SQL Edge are:

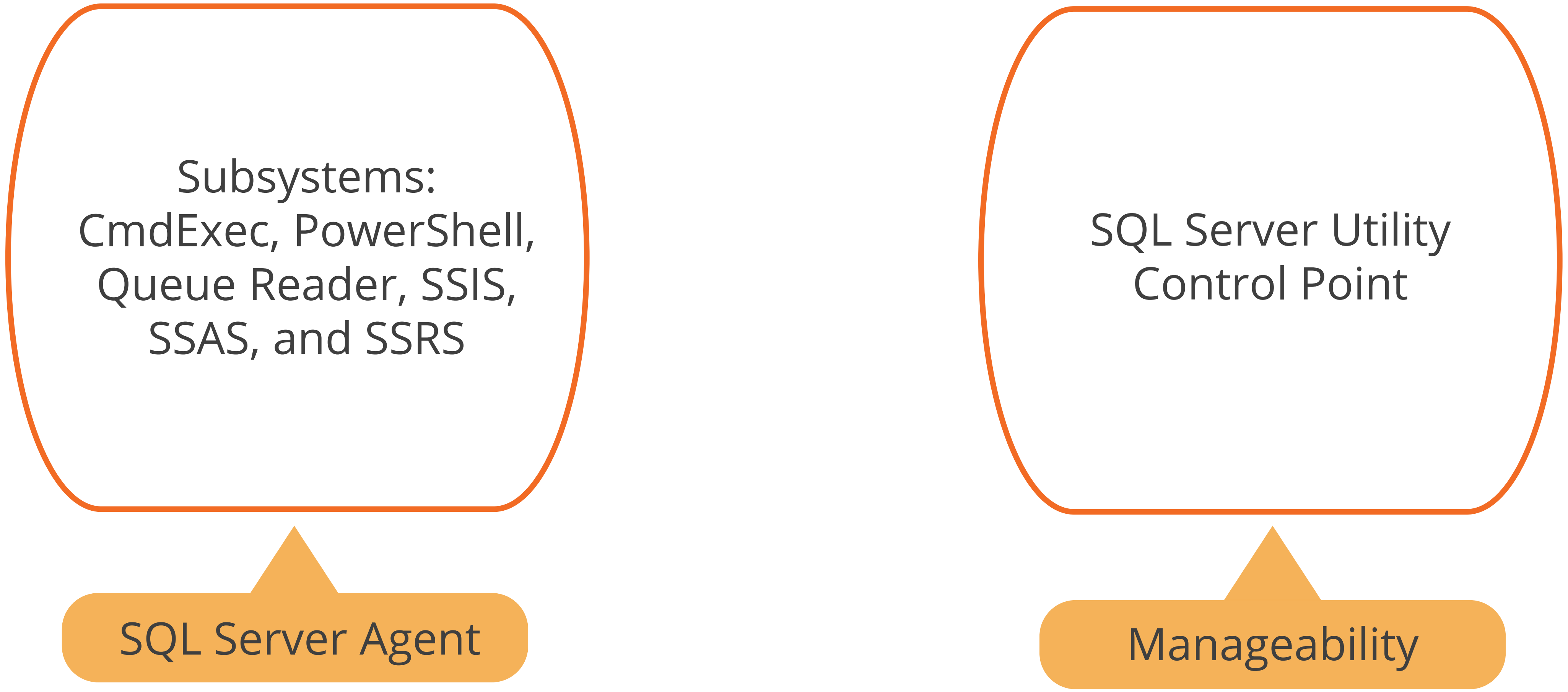
Database Design

Full-text indexes and search, related DDL commands and Transact-SQL functions, catalog views, and dynamic management views

Database Engine

CLR assemblies, related DDL commands and Transact-SQL functions, catalog views, and dynamic management views

Un-Supported Features of Azure SQL Edge



Subsystems: CmdExec, PowerShell, Queue Reader, SSIS, SSAS, and SSRS

SQL Server Agent

SQL Server Utility Control Point

Manageability

Azure SQL Edge Editions

Azure SQL Edge is available with two different editions or software plans such as:

Azure SQL Edge Developer

For development

Azure SQL Edge

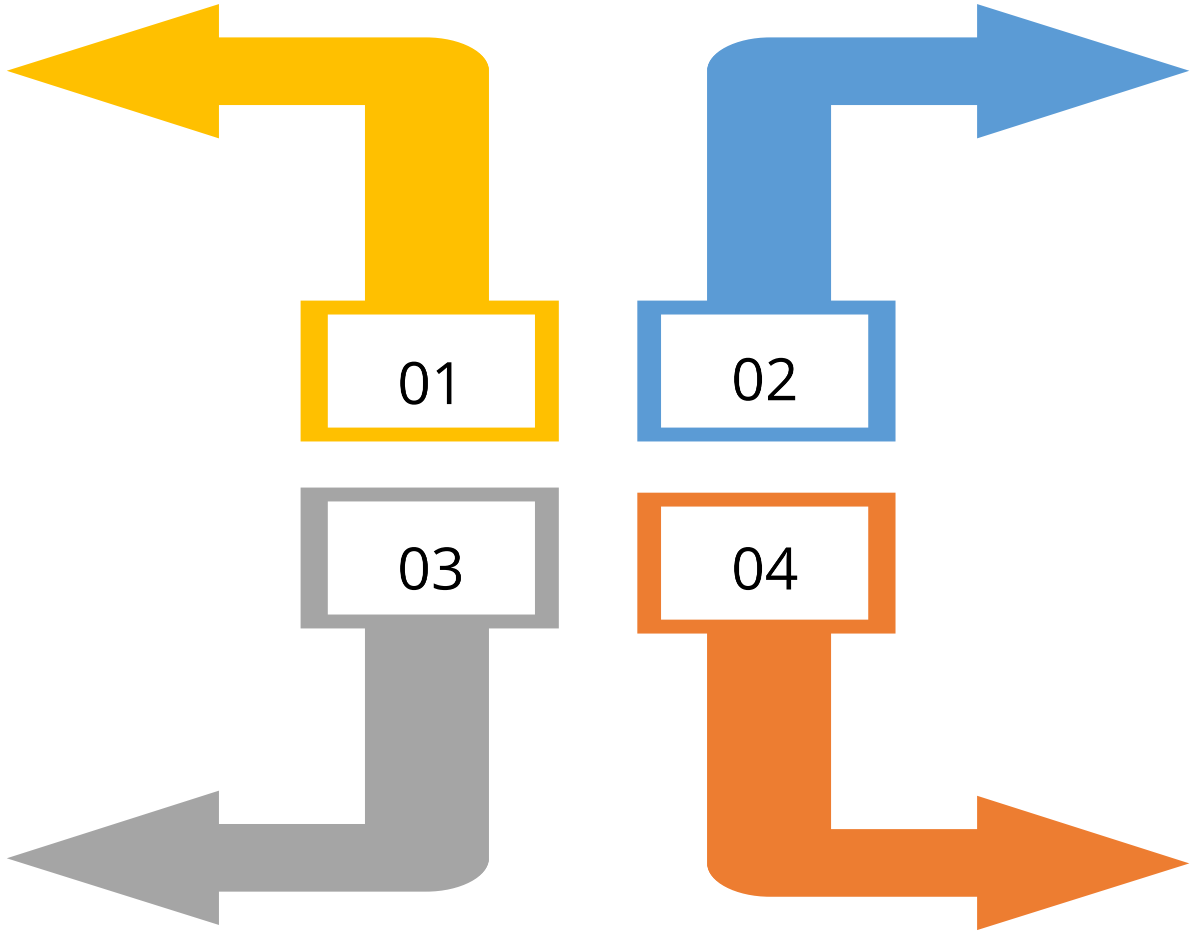
For production

Azure SQL Edge Working

Azure SQL Edge makes it easier and more productive to design and maintain applications. Users can build fantastic apps and solutions for their IoT Edge needs using all of their familiar tools and abilities.

Azure SQL Edge Working

A user can develop in SQL Edge using tools such as:



01 SQL Server Management Studio

02 Azure Data Studio

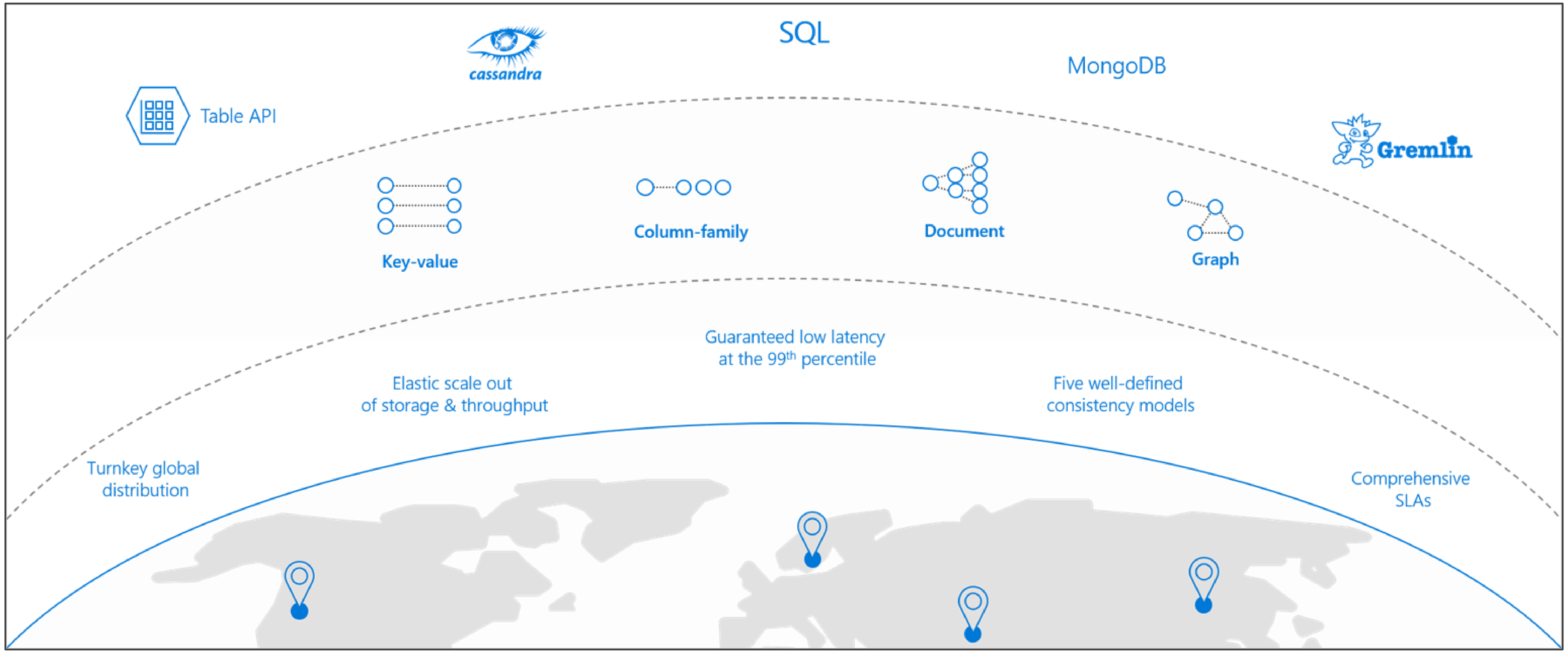
03 SQL Server Data Tools in Visual Studio

04 Visual Studio Code

Design for Azure Cosmos DB

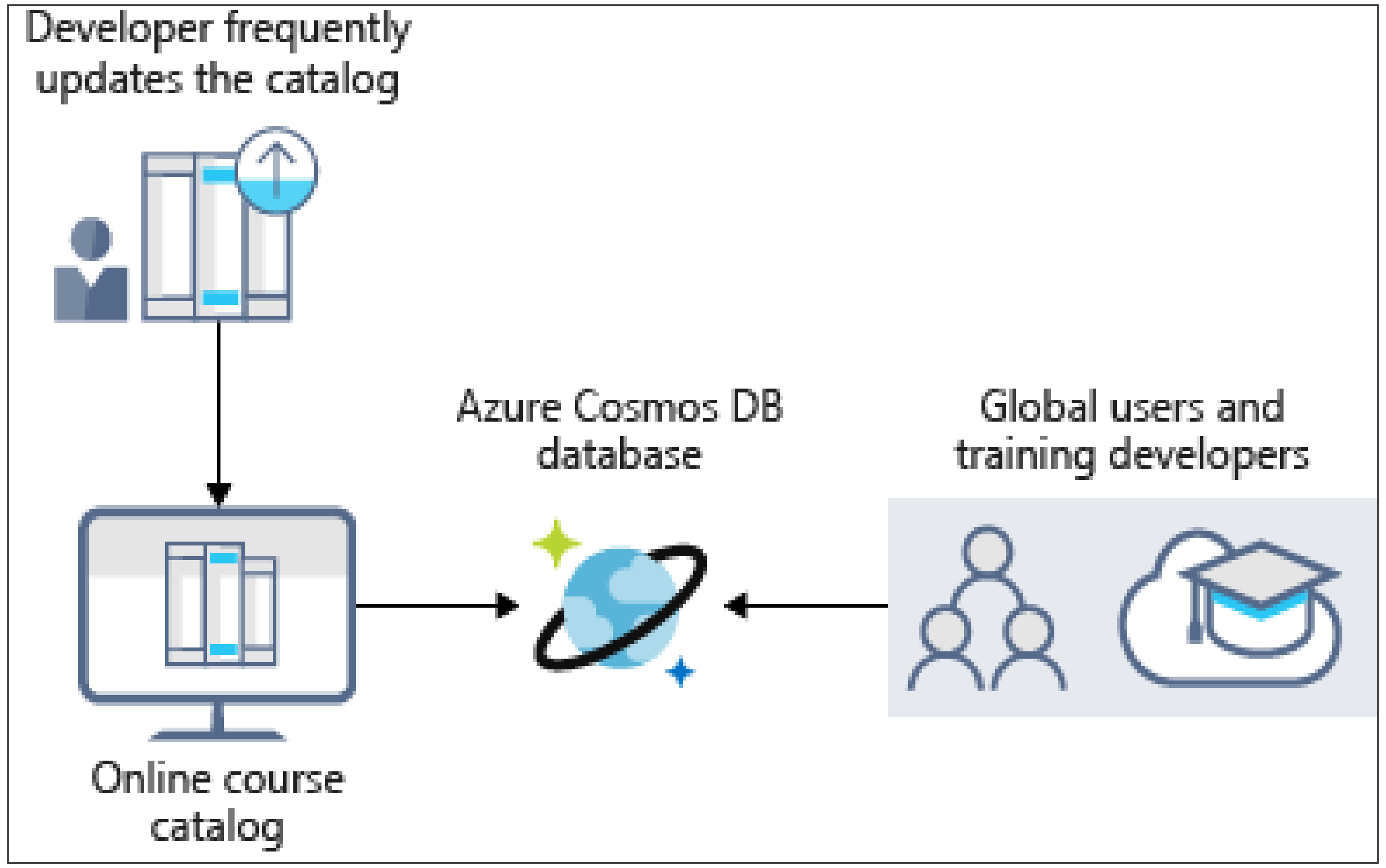
Azure Cosmos DB

Azure Cosmos DB is a globally distributed and elastically scalable database.


cassandra
SQL
MongoDB
Table API
Gremlin
Key-value
Column-family
Document
Graph
Guaranteed low latency at the 99th percentile
Elastic scale out of storage & throughput
Five well-defined consistency models
Turnkey global distribution
Comprehensive SLAs

Azure Cosmos DB

The key benefits of Azure Cosmos DB are:

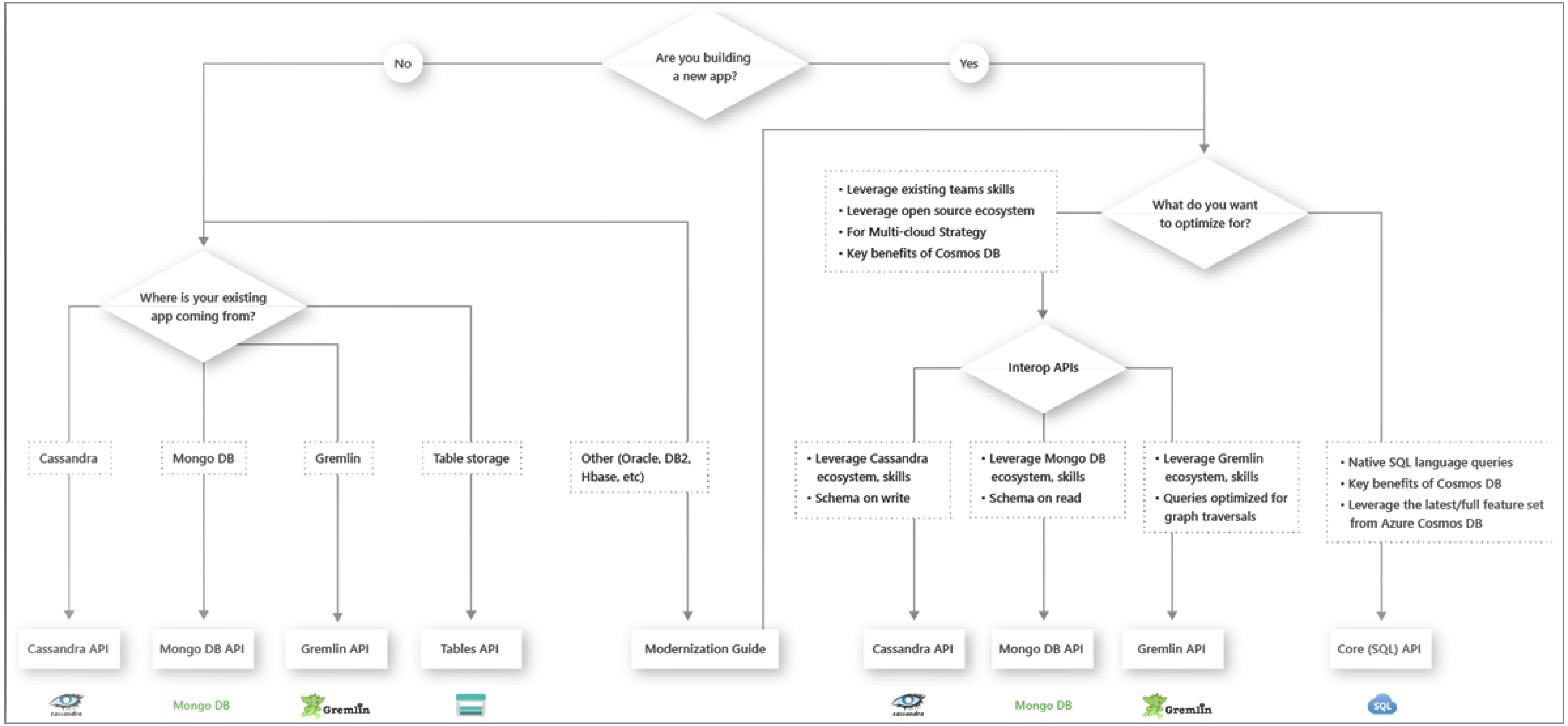


Developer frequently updates the catalog
Azure Cosmos DB database
Global users and training developers
Online course catalog

- Geo-replication
- Elastic scaling of throughput and storage worldwide
- Five well-defined consistency levels

Database API in Azure Cosmos DB

Decision tree to choose an API in Azure Cosmos DB :



Are you building a new app?
No
Yes
Where is your existing app coming from?
Cassandra
Mongo DB
Gremlin
Table storage
Other (Oracle, DB2, Hbase, etc)
Cassandra API
Mongo DB API
Gremlin API
Tables API
Modernization Guide
What do you want to optimize for?
• Leverage existing teams skills
• Leverage open source ecosystem
• For Multi-cloud Strategy
• Key benefits of Cosmos DB
Interop APIs
• Leverage Cassandra ecosystem, skills
• Schema on write
• Leverage Mongo DB ecosystem, skills
• Schema on read
• Leverage Gremlin ecosystem, skills
• Queries optimized for graph traversals
• Native SQL language queries
• Key benefits of Cosmos DB
• Leverage the latest/full feature set from Azure Cosmos DB
Cassandra API
Mongo DB API
Gremlin API
Core (SQL) API
Mongo DB
Gremlin
Mongo DB
Gremlin
SQL

Image Source: https://docs.microsoft.com/

Powerd by simplilearn

Caltech Center for Technology & Management Education

Database API in Azure Cosmos DB

Azure Cosmos DB offers multiple database APIs, such as:

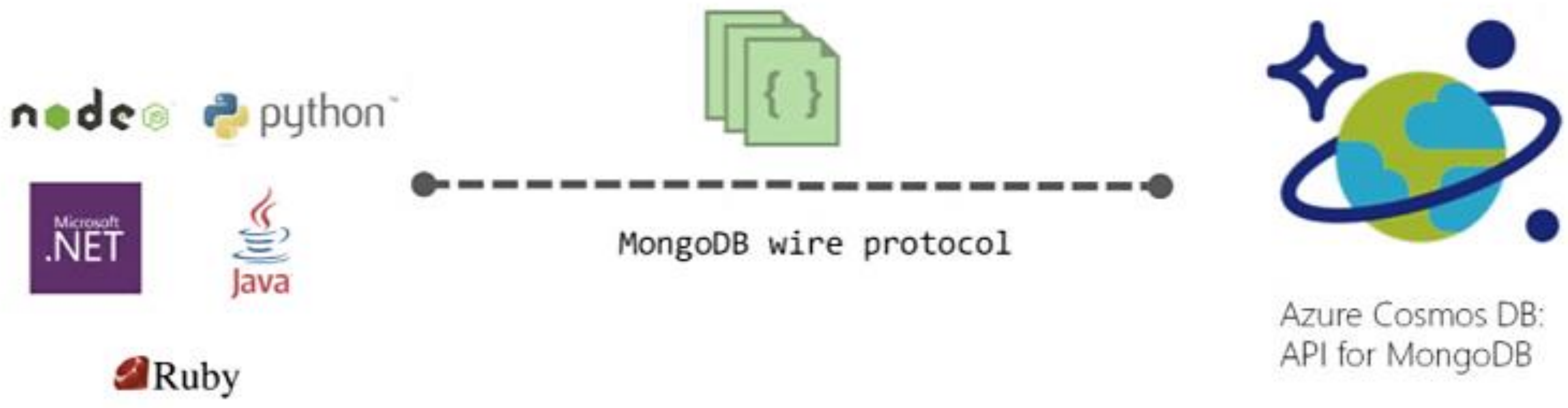
- Core (SQL) API
- API for MongoDB
- Cassandra API
- Gremlin API

Core (SQL) API

The feature that is being rolled out to Azure Cosmos DB is first available on SQL API accounts.

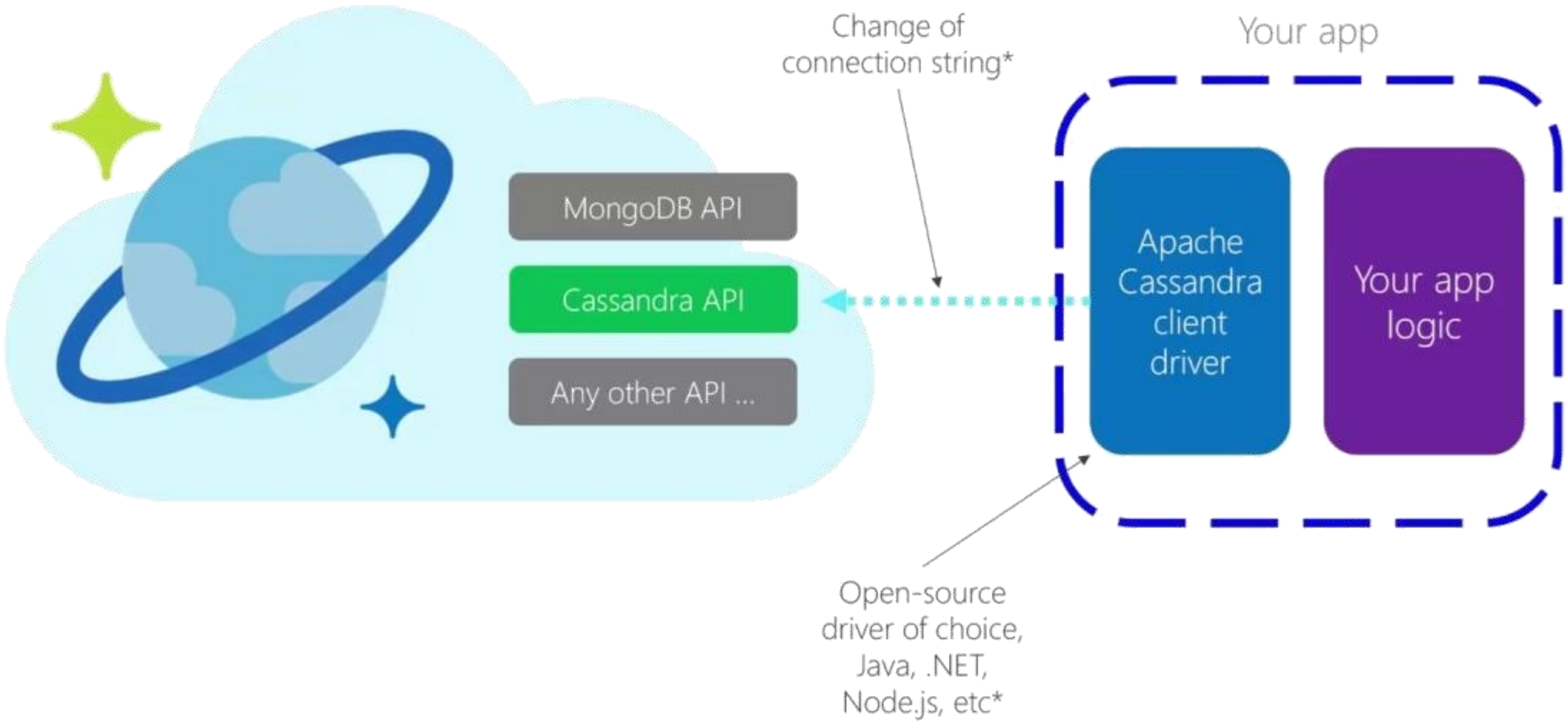
API for MongoDB

The API for MongoDB enables the use of existing client drivers by adhering to the MongoDB wire protocol.



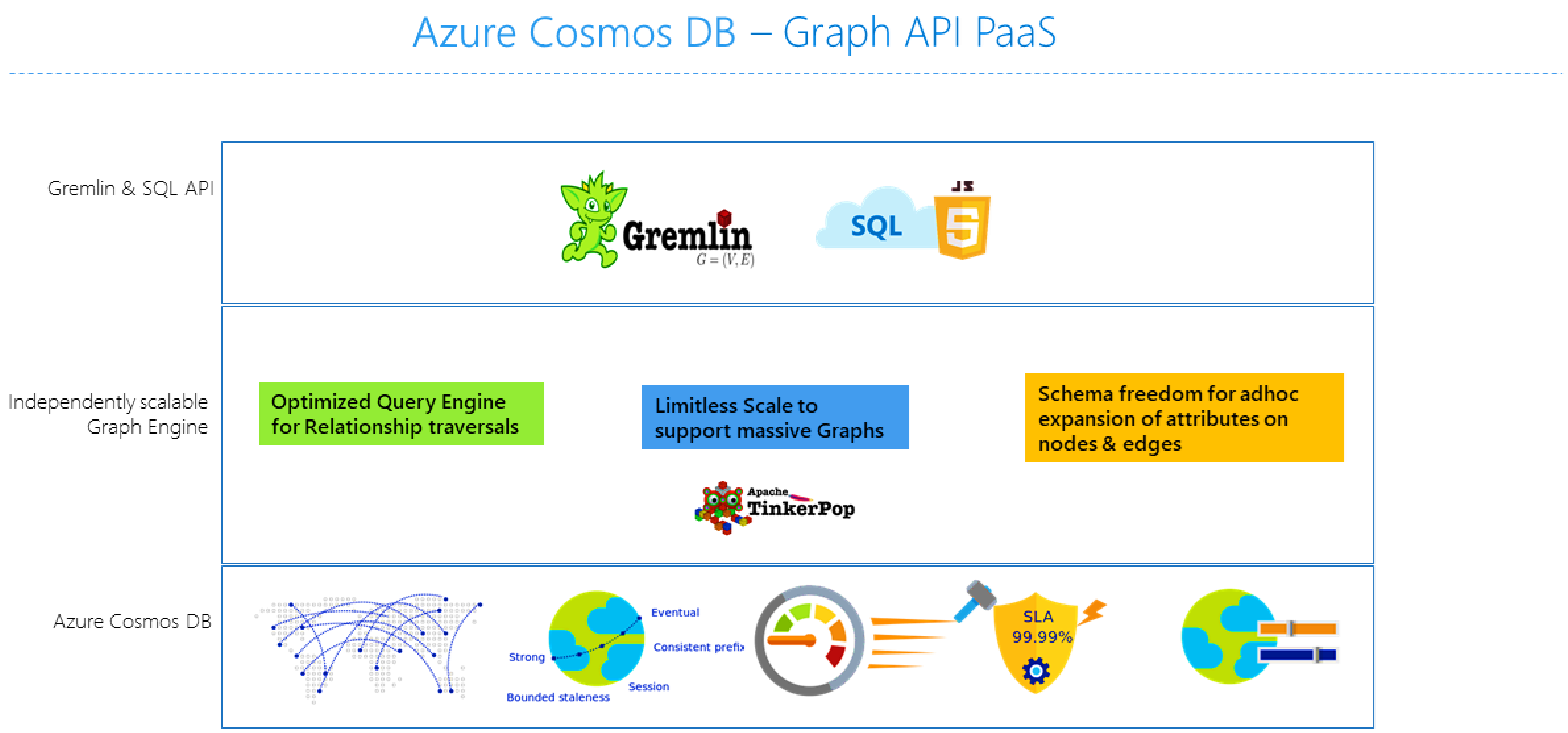
Cassandra API

Cassandra API in Azure Cosmos DB aligns with the philosophy of approaching distributed NoSQL databases, and the API stores data in a column-oriented schema.


MongoDB API
Cassandra API
Any other API ...
Change of connection string*
Your app
Apache Cassandra client driver
Your app logic
Open-source driver of choice, Java, .NET, Node.js, etc*

Gremlin API

Azure Cosmos DB's Gremlin API combines the power of graph database algorithms with highly scalable, managed infrastructure.



Azure Cosmos DB – Graph API PaaS

Gremlin & SQL API

Independently scalable Graph Engine

Optimized Query Engine for Relationship traversals

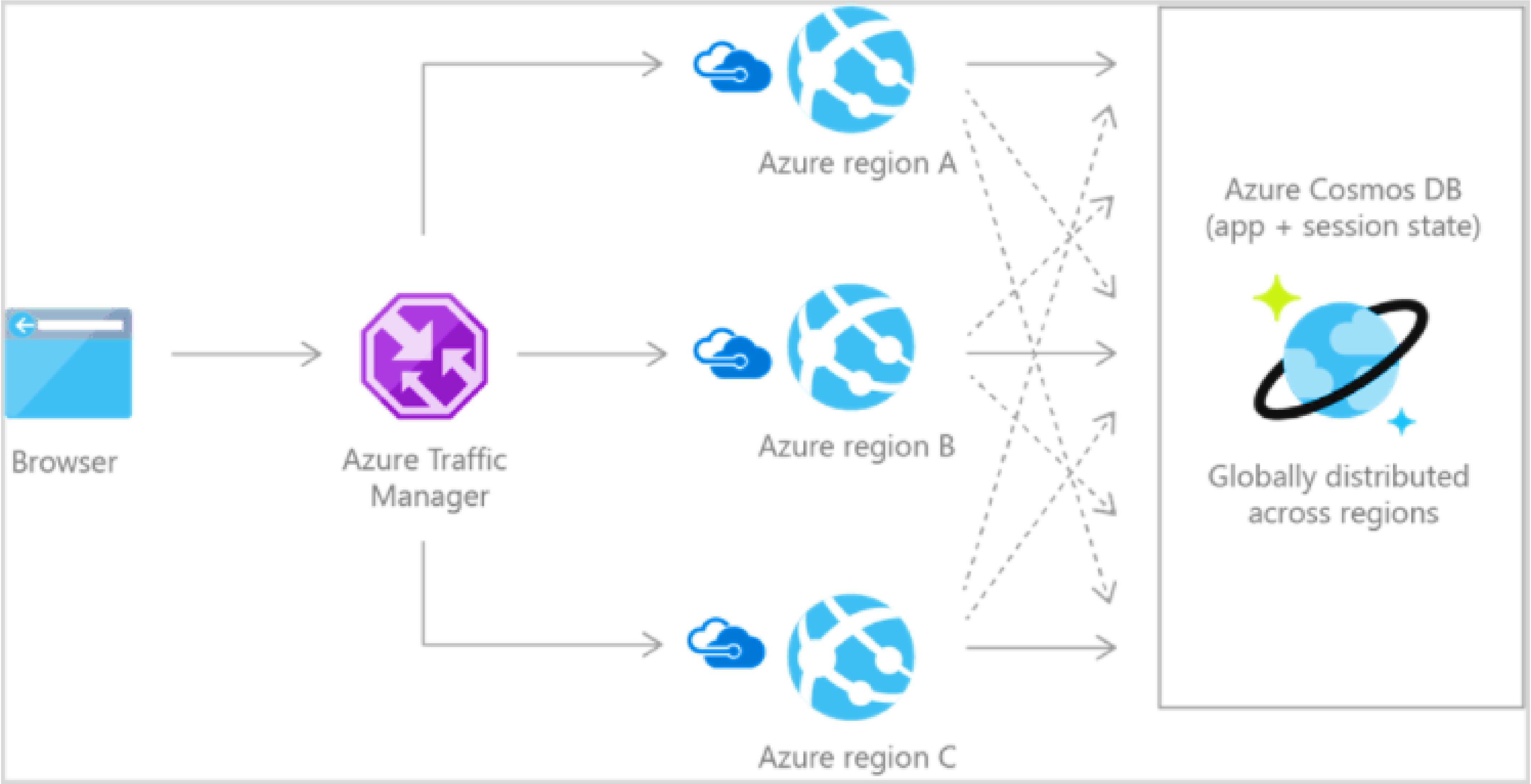
Limitless Scale to support massive Graphs

Schema freedom for adhoc expansion of attributes on nodes & edges

Azure Cosmos DB

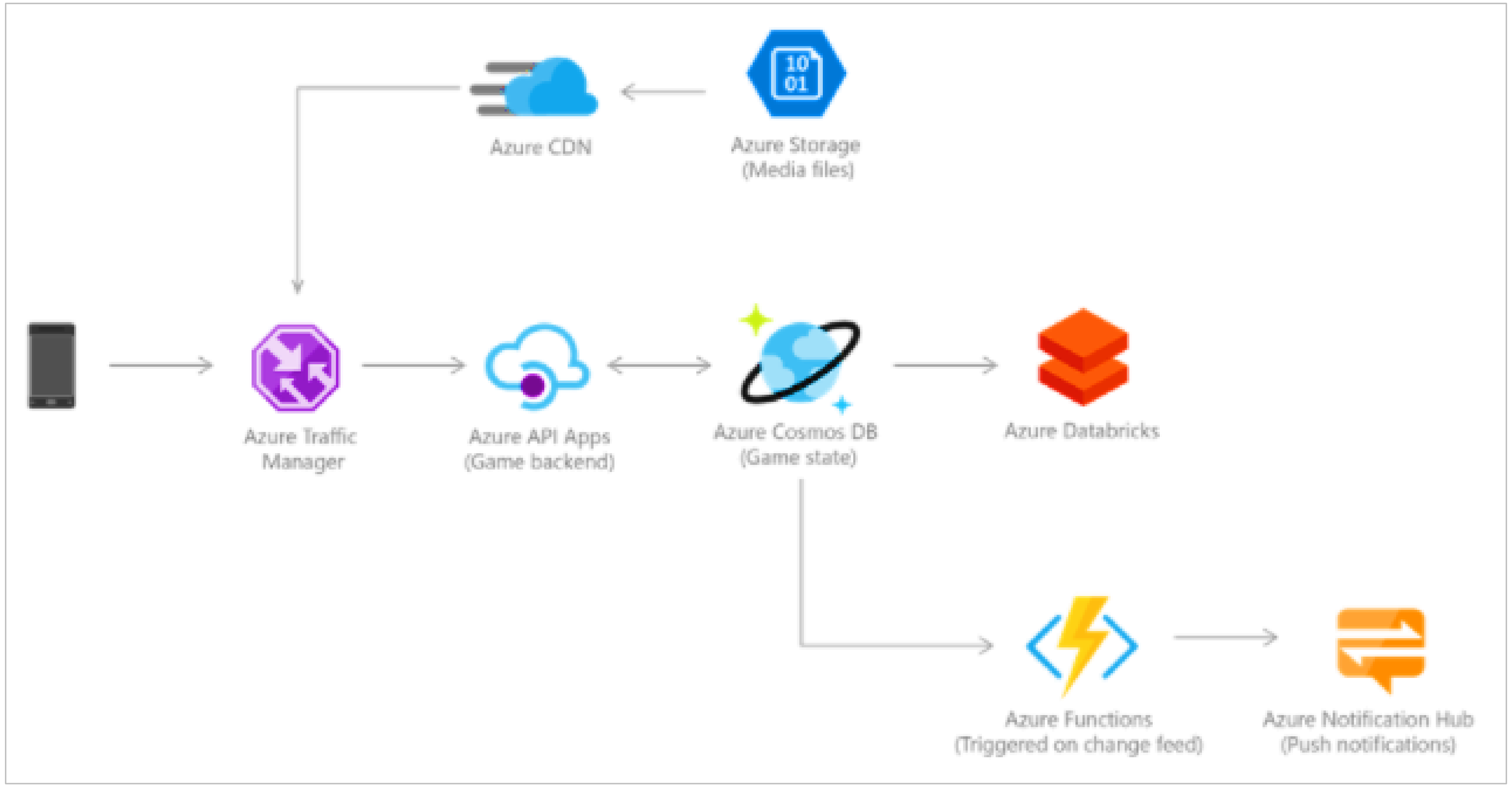
Use Case of Azure Cosmos DB

Applications that integrate with third-party social networks must respond to changing schemas from these networks.


Browser
Azure Traffic Manager
Azure region A
Azure region B
Azure region C
Azure Cosmos DB (app + session state)
Globally distributed across regions

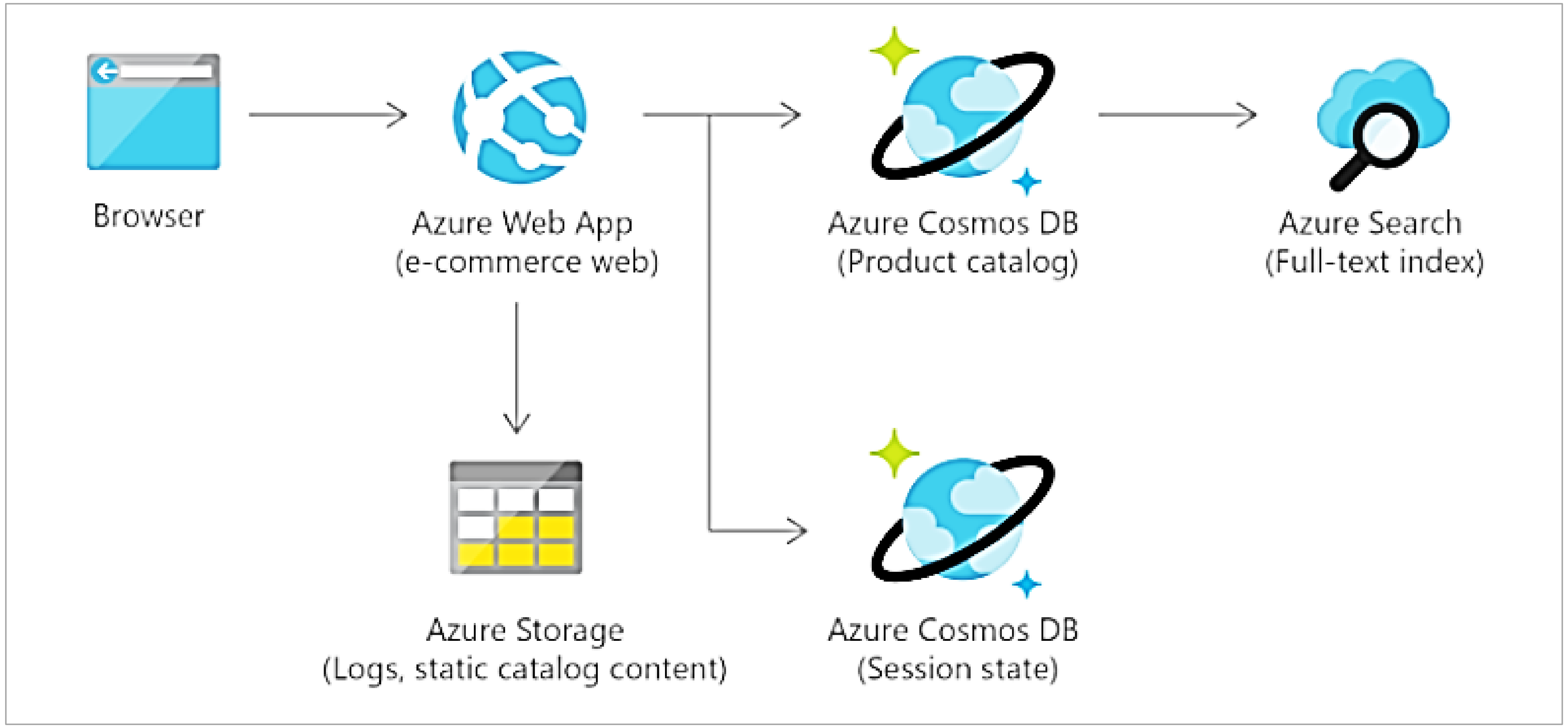
Use Case of Azure Cosmos DB

Modern games perform graphical processing on mobile or console clients but rely on the cloud for in-game content.



Use Case of Azure Cosmos DB

Microsoft's e-commerce sites, such as the Windows Store and Xbox Live, rely heavily on Azure Cosmos DB.



Azure Cosmos DB Table

The features of Azure table storage and Azure Cosmos Table API are:

Features	Azure Table storage	Azure Cosmos DB Table API
Latency	Fast but no upper bounds on latency	Single-digit millisecond latency for reads and writes
Query	Execution uses index for primary key and scans otherwise	Queries that can take advantage of automated indexing on properties
Indexing	Only primary index on Partition Key and Row Key	Automatic indexing on all properties by default without index management
Global distribution	Single region with one optional readable secondary read region for high availability	Turnkey global distribution from one to any number of regions

Azure Cosmos DB Table

Features	Azure Table storage	Azure Cosmos DB Table API
Throughput	Scalability limit of 20,000 operations per second	Highly scalable with dedicated reserved throughput per table that is backed by SLAs and Accounts with no upper limit
Consistency	Strong within primary region and Eventual within secondary region	Five well-defined consistency levels to trade off availability, latency, throughput, and consistency based on needs.
SLAs	99.99% availability, depending on the replication strategy	Comprehensive SLAs covering availability, latency, throughput, and consistency.
Pricing	Consumption-based	Available in both consumption-based and provisioned capacity modes

Storage Management Tools

Manage Tiered Storage Using Azure Tools

There are various tools available to manage Azure Storage:

Azure Portal

Azure Storage Explorer

Azure CLI

Azure PowerShell

Manage Tiered Storage Using Azure Tools

The following commands are used to manage the Azure tools:

```
az storage blob upload
az storage blob list
az storage blob download
az storage blob set-tier
```

```
$Storage= “StorageAccountName”
$Key = “Storage Account Key”
$Container = “Blob Container”
$Blobs = Get-AzureStorageblob -Container $Container
$blob.icloudblob.setstandardblobtier(“Cool”)
```

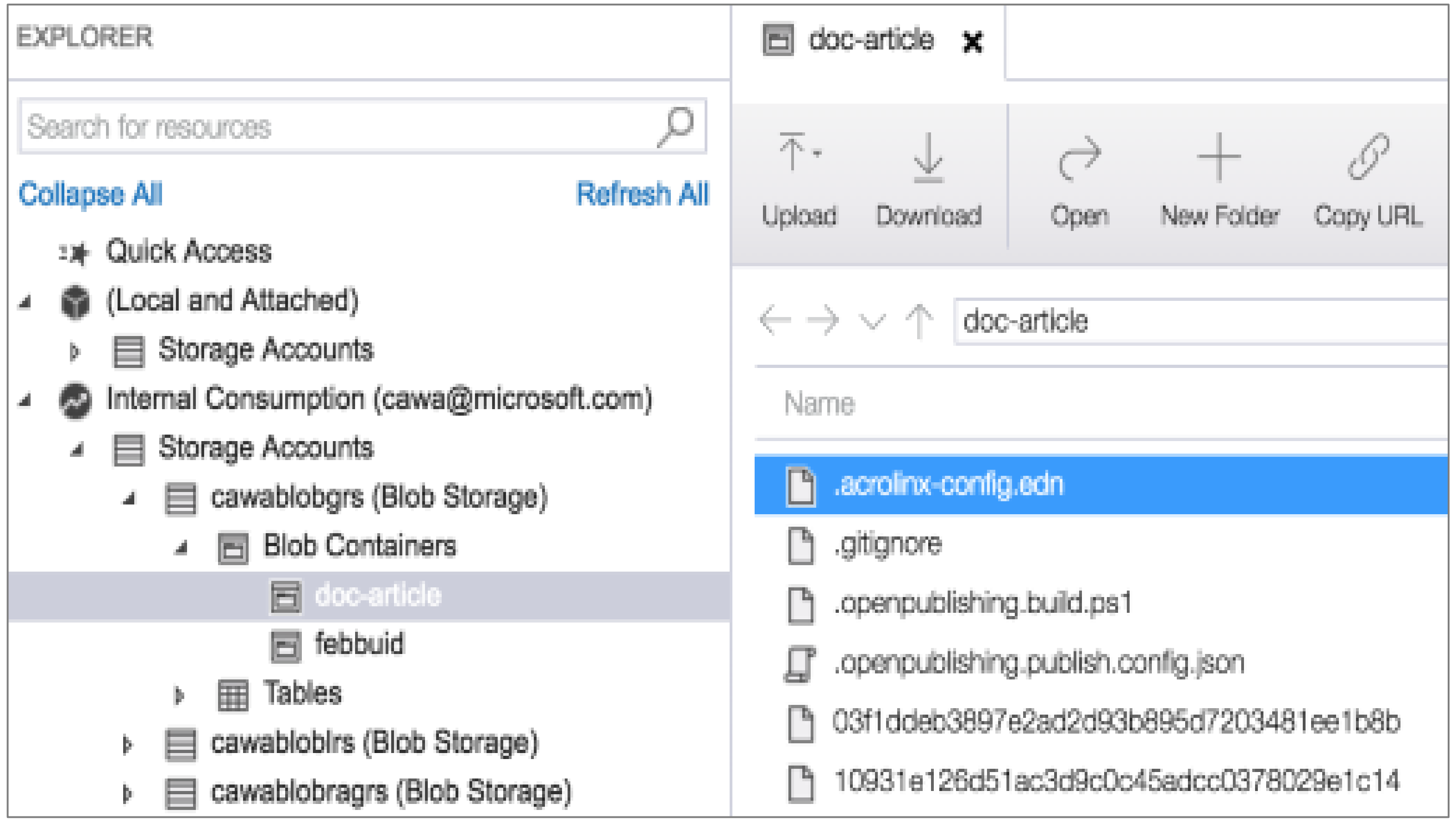
Comparing Tools for Working with Azure Storage

The table given below shows the comparison of tools that can run on various platforms:

Azure Storage Client Tool	Supported Platform	Block Blob	Page Blob	Append Blob	Tables	Queues	Files
Azure Portal	Web	Yes	Yes	Yes	Yes	Yes	Yes
Azure Storage Explorer	Windows, OSX	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft Visual Studio Cloud Explorer	Windows	Yes	Yes	Yes	Yes	Yes	No

Azure Storage Explorer

Azure Storage Explorer is a single software for Windows, macOS, and Linux that makes working with Azure Storage data simple.



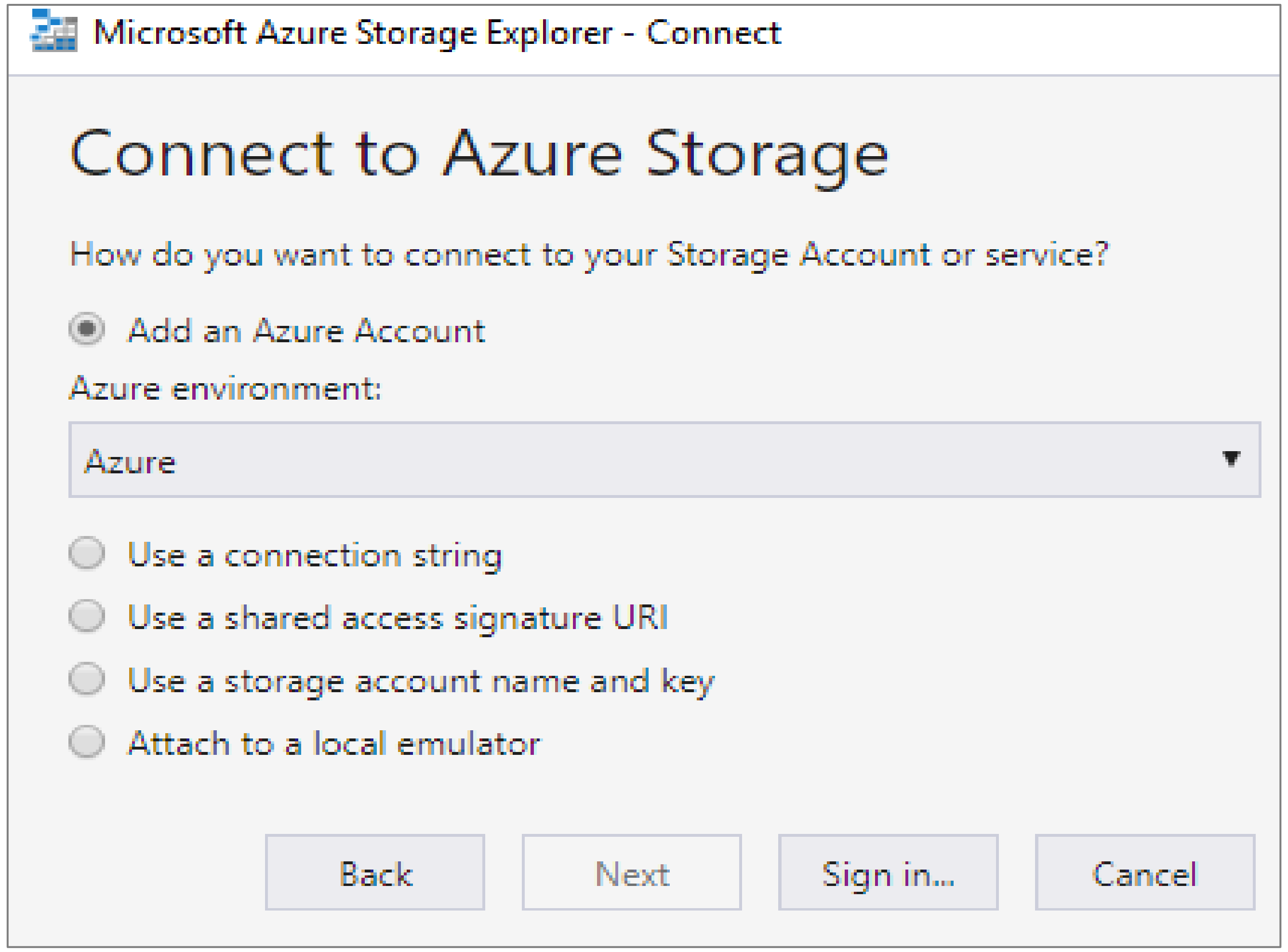
Quick Access
(Local and Attached)
Storage Accounts
Internal Consumption (cawa@microsoft.com)
Storage Accounts
cawablobgrs (Blob Storage)
Blob Containers
doc-article
febbuid
Tables
cawablobIrs (Blob Storage)
cawablobragrs (Blob Storage)

.acrolinx-config.edn
.gitignore
.openpublishing.build.ps1
.openpublishing.publish.config.json
03f1ddeb3897e2ad2d93b895d7203481ee1b8b
10931e126d51ac3d9c0c45adcc0378029e1c14

- Access multiple accounts and subscriptions
- Create, delete, view, and edit storage resources
- View and edit Blob, Queue, Table, File, Cosmos DB storage, and Data Lake Storage
- Obtain shared access signature (SAS) keys
- Available for Windows, Mac, and Linux

Azure Storage Connection

The procedure for connecting to Azure storage is given below:



Microsoft Azure Storage Explorer - Connect

Connect to Azure Storage

How do you want to connect to your Storage Account or service?

● Add an Azure Account

Azure environment: Azure

○ Use a connection string

○ Use a shared access signature URI

○ Use a storage account name and key

○ Attach to a local emulator

Options

- Connect to an Azure subscription
- Work with local development storage (emulator)
- Attach to external storage
- Attach a storage account or storage service by using a shared access signature
- Connect to an Azure Cosmos DB account by using a connection string

Assisted Practice

Managing Cosmos DB

Duration: 10 Min.

Problem Statement:

You need to create a Cosmos DB account and identify different options for deploying and configuring Cosmos DB. You also need to add some data to the Cosmos DB to check its functionality further by running queries.

Assisted Practice: Guidelines

Steps to manage storage access keys:

1. Creating a Cosmos DB account
2. Creating a database
3. Adding some data to the cosmos DB database
4. Querying the data using SQL APIs

Design solution for Database

Duration: 25 min.

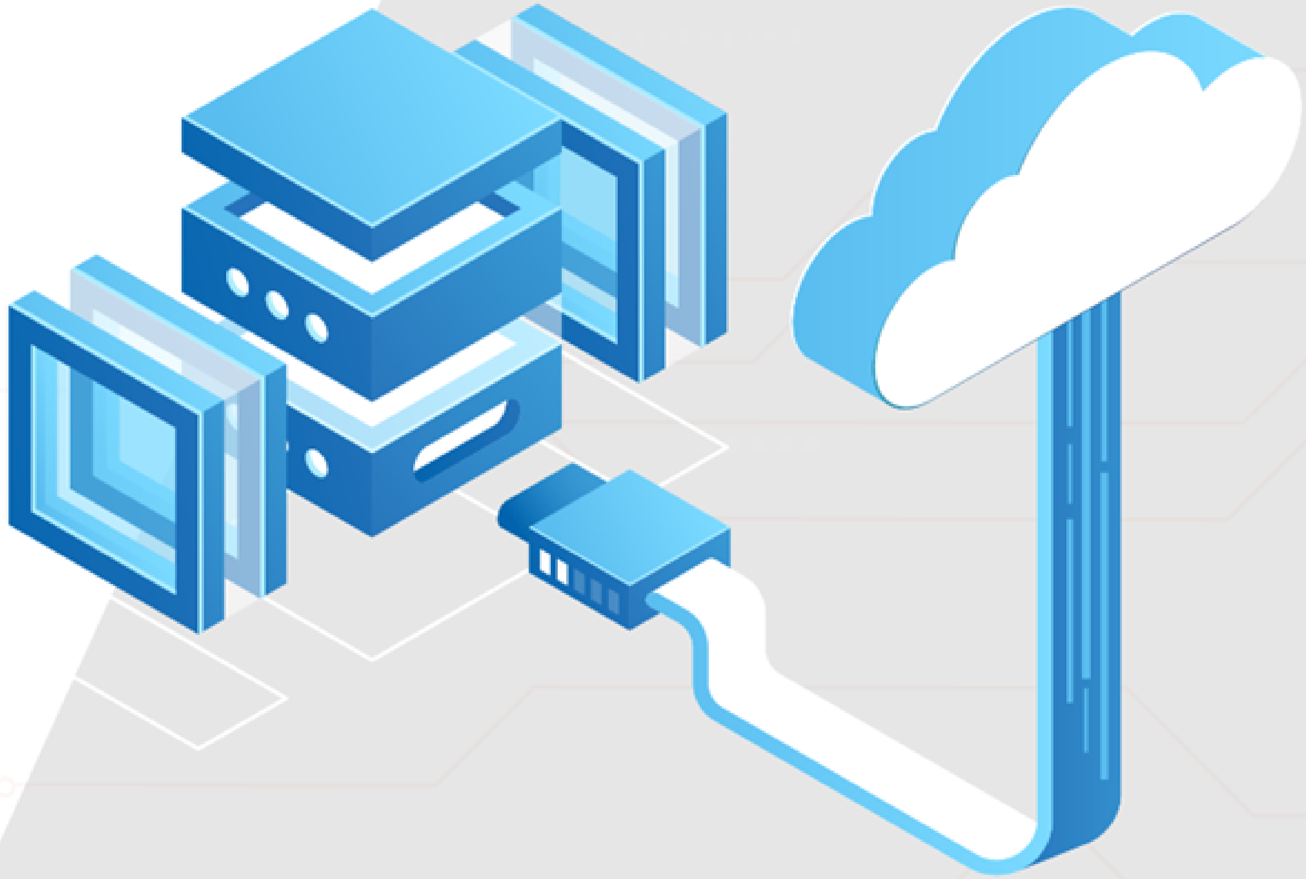
Project Agenda: Design a solution for database

Description: You have been given a project to move a database from on-premise to the cloud. The data is in a document format. You need to choose the right database option. You also need to ensure that you have support for querying items using the Structured Query Language (SQL) syntax. Suggest a solution for this migration.

Perform the following:

1. Creating an Azure Cosmos DB account by selecting Core (SQL) API option.

LESSON-END PROJECT



Thank you