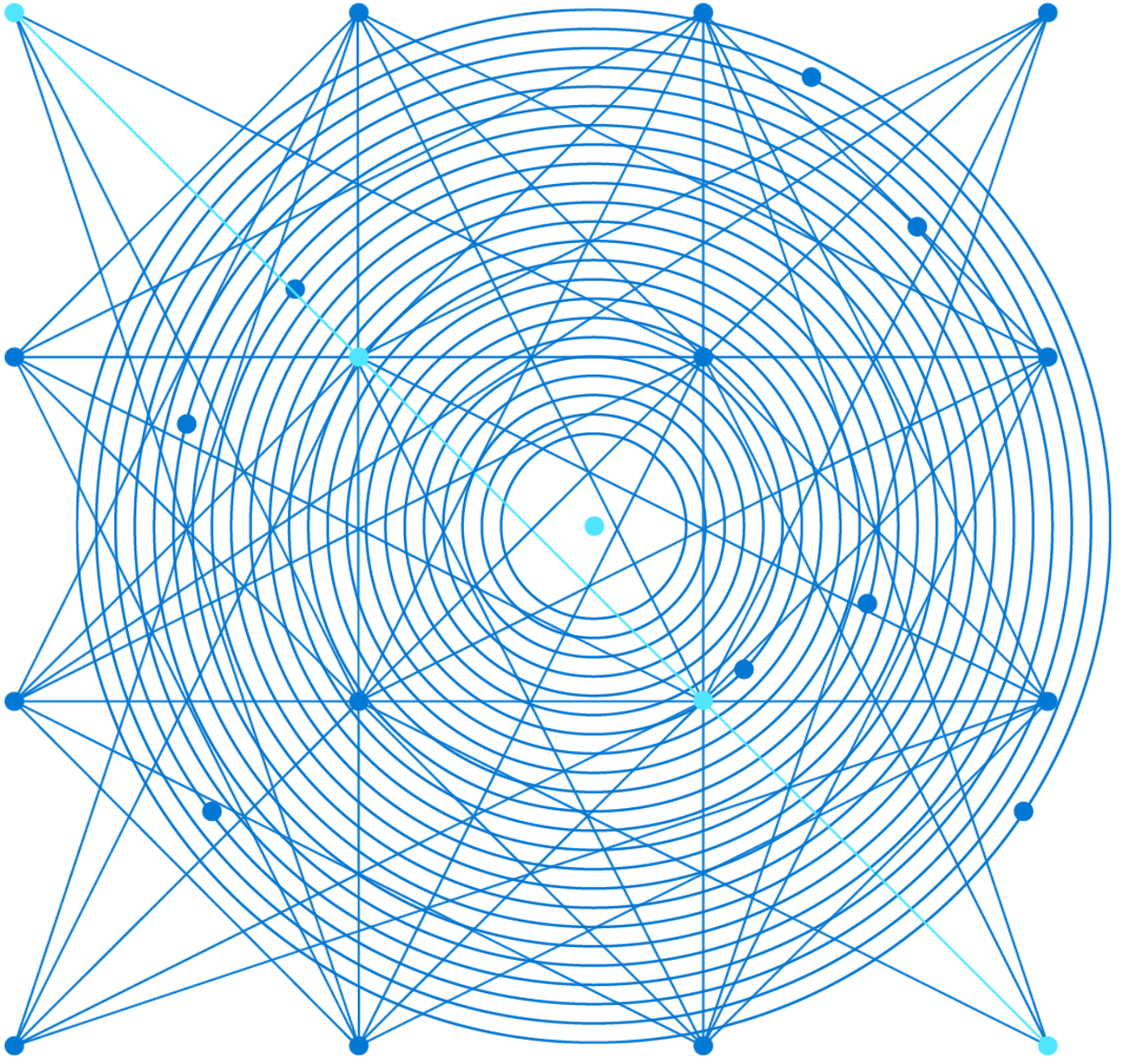
AZ-303: Microsoft Azure Architect Technologies



Module 7: Implement NoSQL Databases

Azure Storage Account Tables and CosmosDB APIs

Learning Objectives

You will learn the following:

- Configure Storage Account Tables
- Select Appropriate Cosmos DB APIs

Configure Storage Account Tables

Azure Table Storage (1 of 2)

- Azure Table is a NoSQL datastore ideal for storing structured, non-relational data.

- Common use of table storage
  - Storing TBs of structured data capable of serving web scale applications
  - Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
  - Quickly querying data using a clustered index
  - Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries

Azure Table Storage (2 of 2)

Table storage concepts

- URL format:

  http://<storage account>.table.core.windows.net/<table>

- Accounts
- Table
- Entity
- Properties

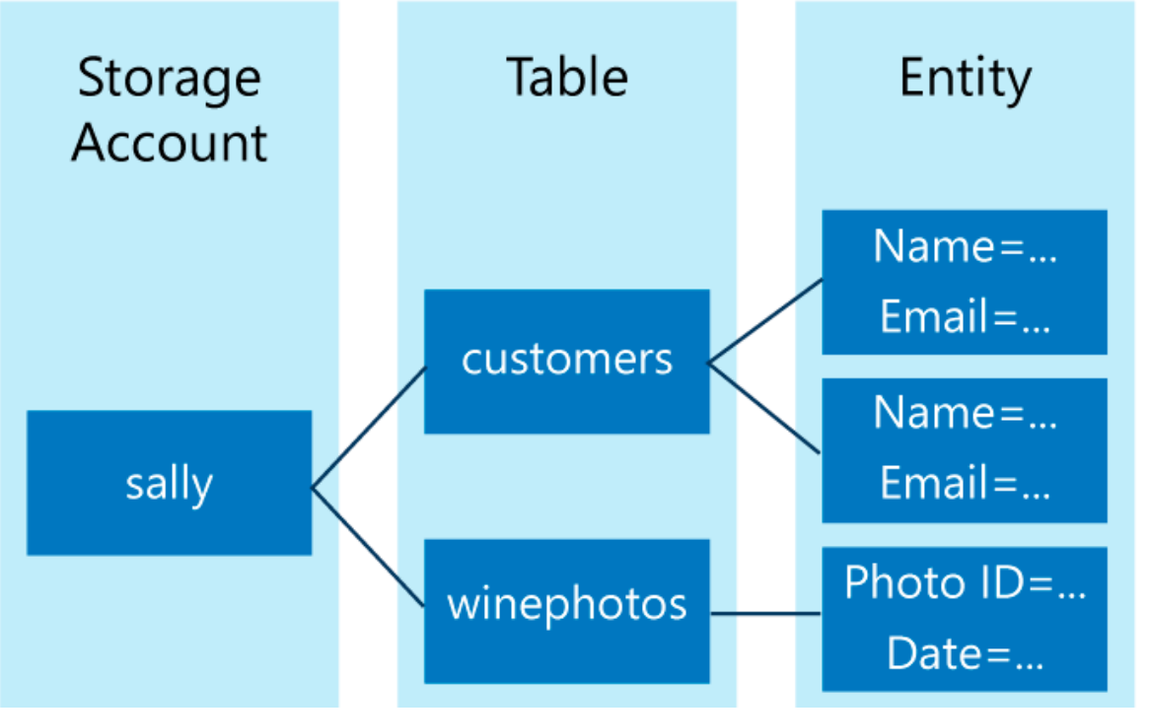


Table Service Data Model (1 of 2)

- Storage account and table service **https://*myaccount*.table.core.windows.net**

- Tables store data as collections of entities
  - Entities are similar to rows. An entity has a primary key and a set of properties.
  - A property is a name, typed-value pair, similar to a column.
  - An entity can have up to 255 properties, including 3 system properties

- Table names must conform to the following rules
  - Table names must be unique within an account
  - Table names may contain only alphanumeric characters
  - Table names cannot begin with a numeric character
  - Table names are case-insensitive
  - Table names must be from 3 to 63 characters long
  - Some table names are reserved, including "tables"

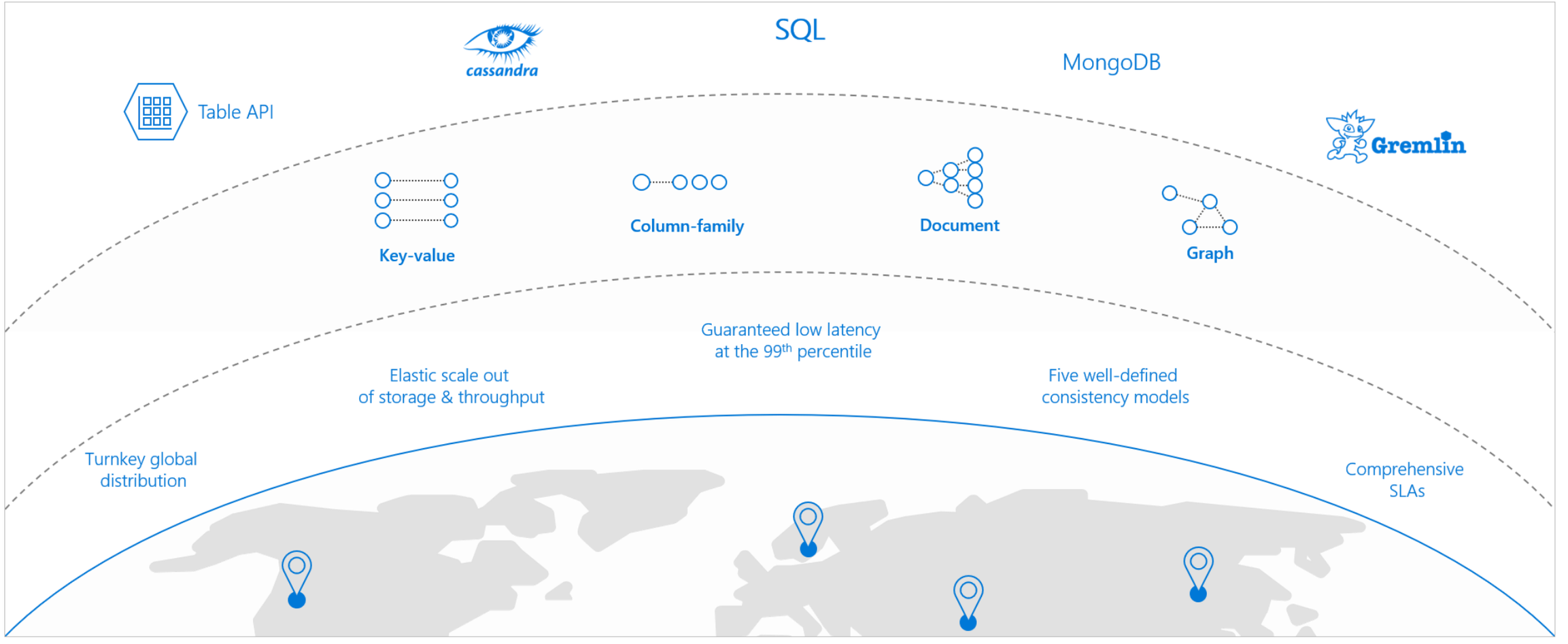
Table Service Data Model (2 of 2)

- System properties (automatically included for every entity in a table)
  - **PartitionKey** property: serves as the basis for table partitioning
  - **RowKey** property: a unique identifier for an entity within a given partition
  - **Timestamp** property: DateTime value that is maintained on the server side to record the time an entity was last modified

Select Appropriate Cosmos DB APIs

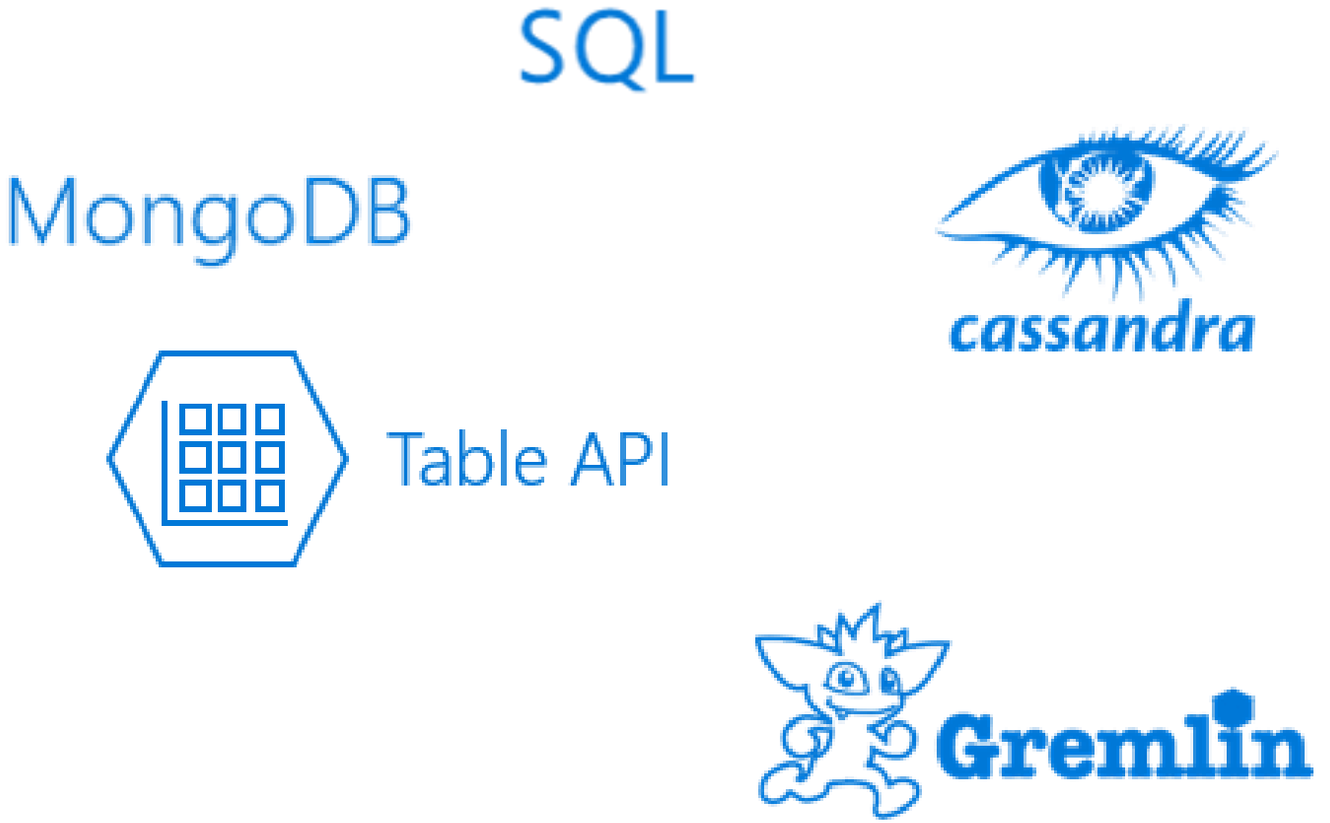
Overview of Azure Cosmos DB (1 of 2)

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

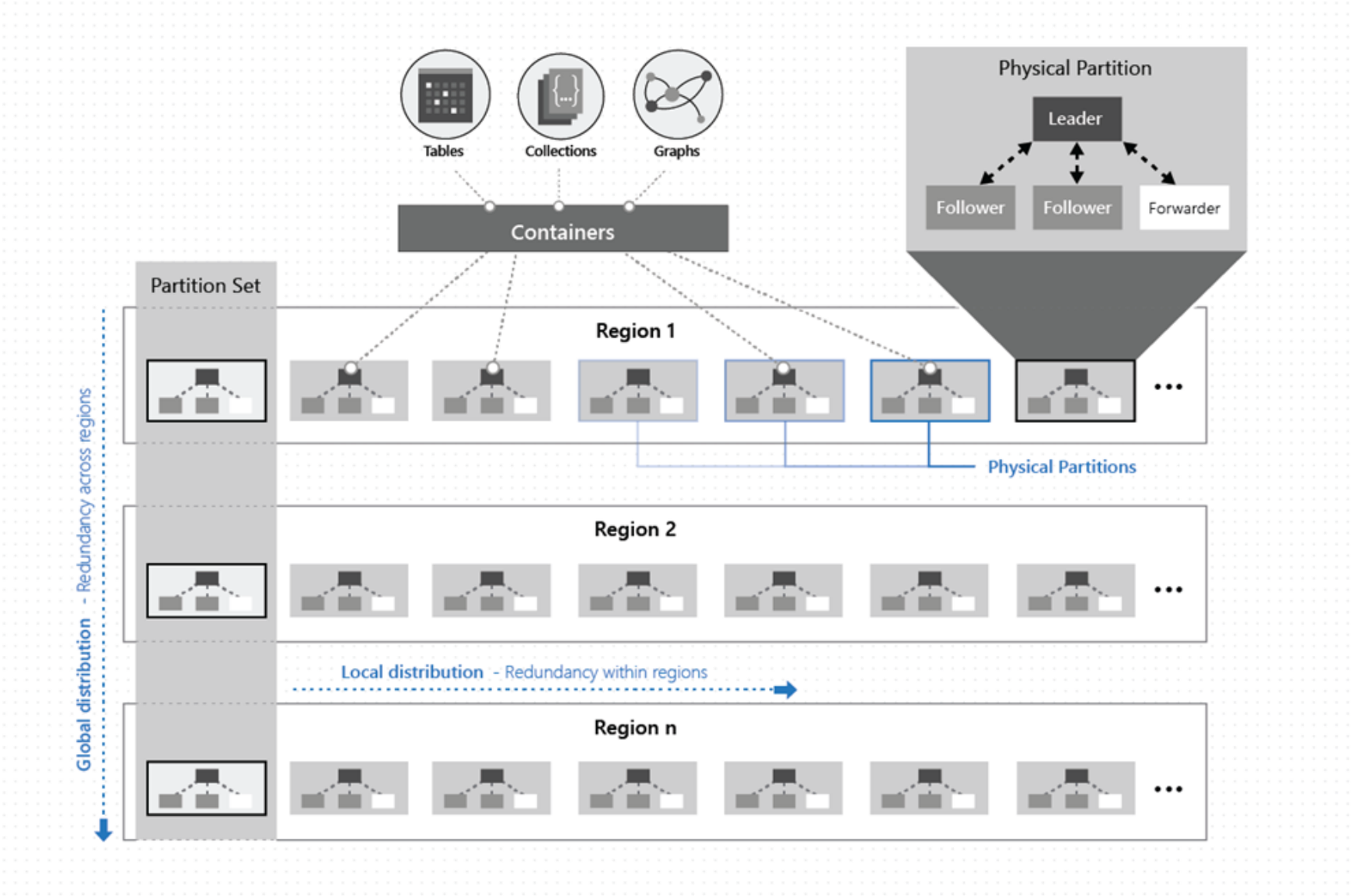


Overview of Azure Cosmos DB (2 of 2)

- Cosmos DB supports a number of APIs
  - SQL
  - MongoDB
  - Cassandra
  - Table API
  - Gremlin



High Availability with Cosmos DB (1 of 3)



High Availability with Cosmos DB (2 of 3)

- SLAs for availability

Operation type	Single region	Multi-region (single region writes)	Multi-region (multi-region writes)
Writes	99.99	99.99	99.999
Reads	99.99	99.999	99.999

- High availability with Cosmos DB in the event of regional outages
  - With Cosmos DB, before a write operation is acknowledged to the client, the data is durably committed by a quorum of replicas within the region that accepts the write operations
  - Multi-region accounts configured with multiple-write regions will be highly available for both writes and reads

High Availability with Cosmos DB (3 of 3)

- Multi-region accounts with a single-write region (write region outage):
  - During a write region outage, the Cosmos account automatically promotes a secondary region to be the new primary write region
  - Any write data that was not replicated when the region failed, is made available through the conflicts feed.
  - When the impacted write region recovers, it becomes automatically available as a read region

- Multi-region accounts with a single-write region (read region outage):
  - During a read region outage, Cosmos accounts using any consistency level or strong consistency with three or more read regions will remain highly available for reads and writes
  - The impacted region is automatically disconnected and will be marked offline
  - If none of the regions in the preferred region list is available, calls automatically fall back to the current write region

- Availability Zone support

Overview of Azure Cosmos DB Supported APIs (1 of 2)

MongoDB API

- Acts as a massively scalable MongoDB service powered by the Azure Cosmos DB platform
- Compatible with existing MongoDB libraries, drivers, tools, and applications

Table API

- A key-value database service built to provide premium capabilities to existing Azure Table storage applications without making any app changes

Gremlin API

- A fully managed, horizontally scalable graph database service
- Easy-to-build and run applications that work with highly connected datasets supporting Open Graph APIs (based on the Apache TinkerPop specification, Apache Gremlin)

Overview of Azure Cosmos DB Supported APIs (2 of 2)

Cassandra API

- Globally distributed Apache Cassandra service powered by the Azure Cosmos DB platform
- Compatible with existing Apache Cassandra libraries, drivers, tools, and applications

SQL API

- JavaScript and JavaScript Object Notation (JSON) native API based on the Azure Cosmos DB database engine
- Provides query capabilities rooted in SQL
- Queries for documents based on their identifiers or make deeper queries based on properties of the document, complex objects, or the existence of specific properties
- Supports the execution of JavaScript logic within the database in the form of stored procedures, triggers, and user-defined functions

Azure Cosmos DB Cassandra API

Benefits of using Apache Cassandra API for Azure Cosmos DB

- No operations management
- Open source standard
- Performance management
- Ability to use existing tools and code
- Throughput and storage elasticity
- Global distribution and availability
- Choice of consistency
- Enterprise grade
- Event sourcing

Azure Cosmos DB API for MongoDB

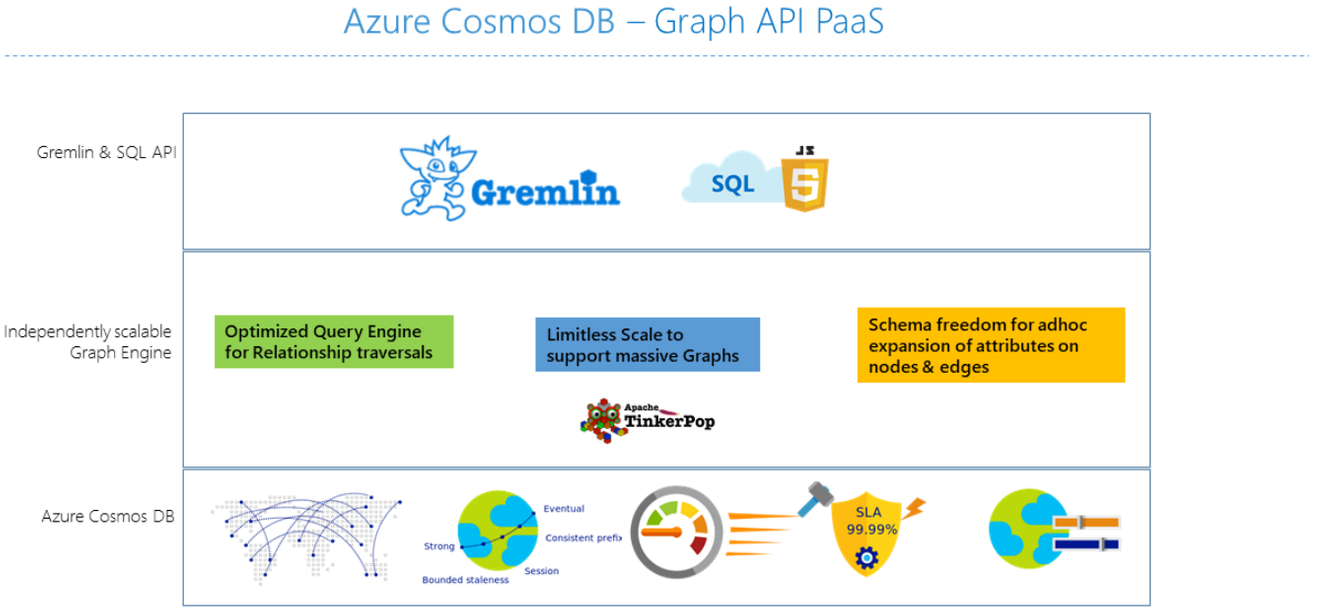
- Wire protocol compatibility
- Key benefits
  - The ability to for migrate applications to Cosmos DB while preserving significant portions of the existing application logic
  - Keeping applications portable and continuing to remain cloud vendor-agnostic
  - Financially backed SLAs for the common NoSQL APIs powered by Cosmos DB
  - Turnkey, global distribution with multi-master replication

Azure Cosmos DB Gremlin API

- Features of Azure Cosmos DB graph database
  - Elastically scalable throughput and storage
  - Multi-region replication
  - Fast queries and traversals with the most widely adopted graph query standard
  - Fully managed graph database
  - Automatic indexing
  - Tunable consistency levels

- Graph databases

- Property graph objects
  - Vertices
  - Edges
  - Properties



Azure Cosmos DB table API

Azure Cosmos DB provides the Table API for applications that are written for Azure Table storage and that need premium capabilities such as

- Turnkey global distribution.
- Dedicated throughput worldwide.
- Single-digit millisecond latencies at the 99th percentile.
- Guaranteed high availability.
- Automatic secondary indexing.

If you currently use Azure Table Storage, you gain the following benefits by moving to the Azure Cosmos DB Table API

- Latency
- Throughput
- Global distribution
- Indexing
- Query
- Consistency
- Pricing
- SLAs

Module Review Questions

Microsoft Azure

Online Role-based training resources:

Microsoft Learn

https://docs.microsoft.com/en-us/learn/

Microsoft Azure

Thank you.