DP-203 Microsoft Azure Data Engineer

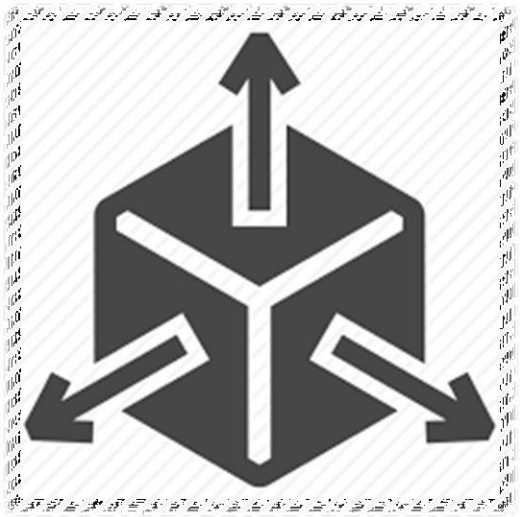
NoSQL - CosmosDB

28th July 2021

Vinodkumar Bhovi

databoq®

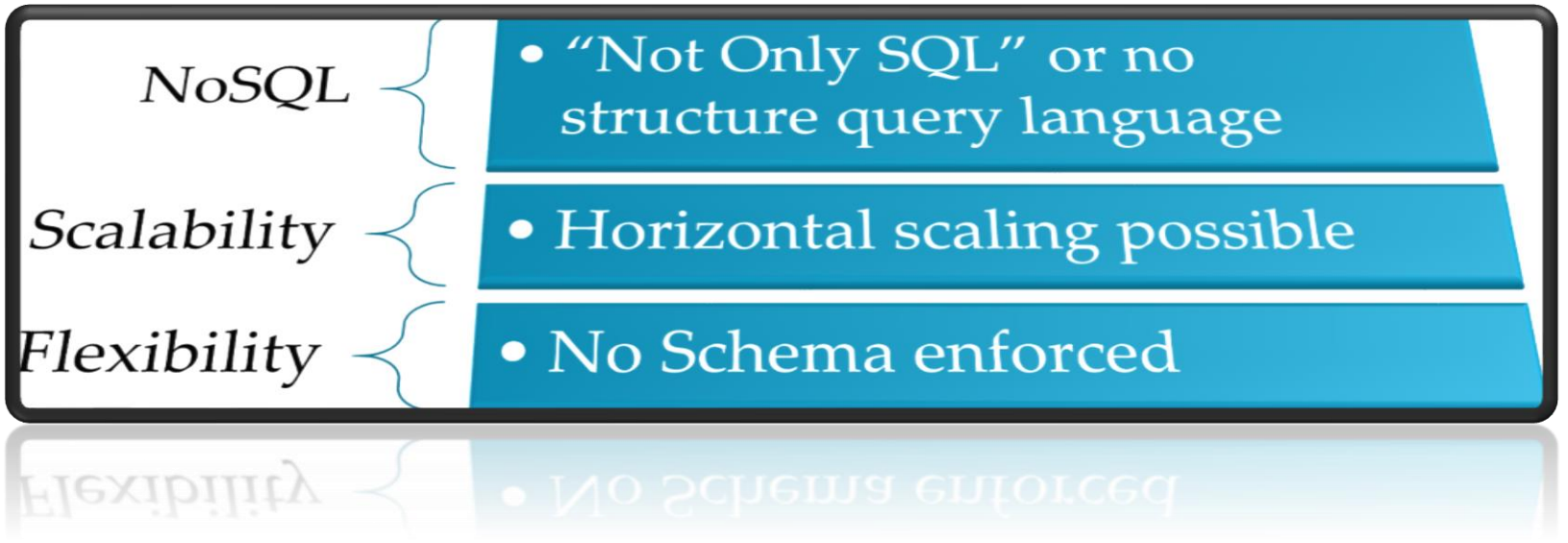
RDBMS were lacking



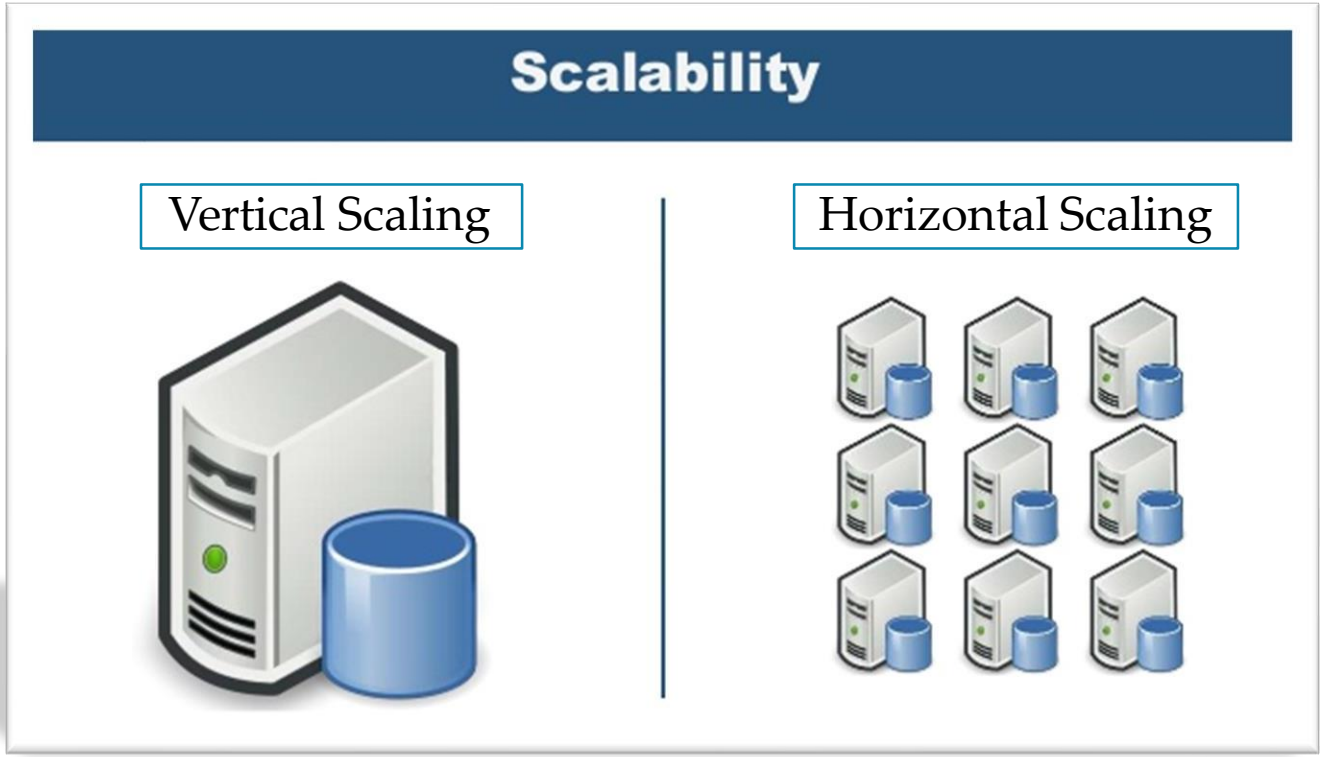
Scalability

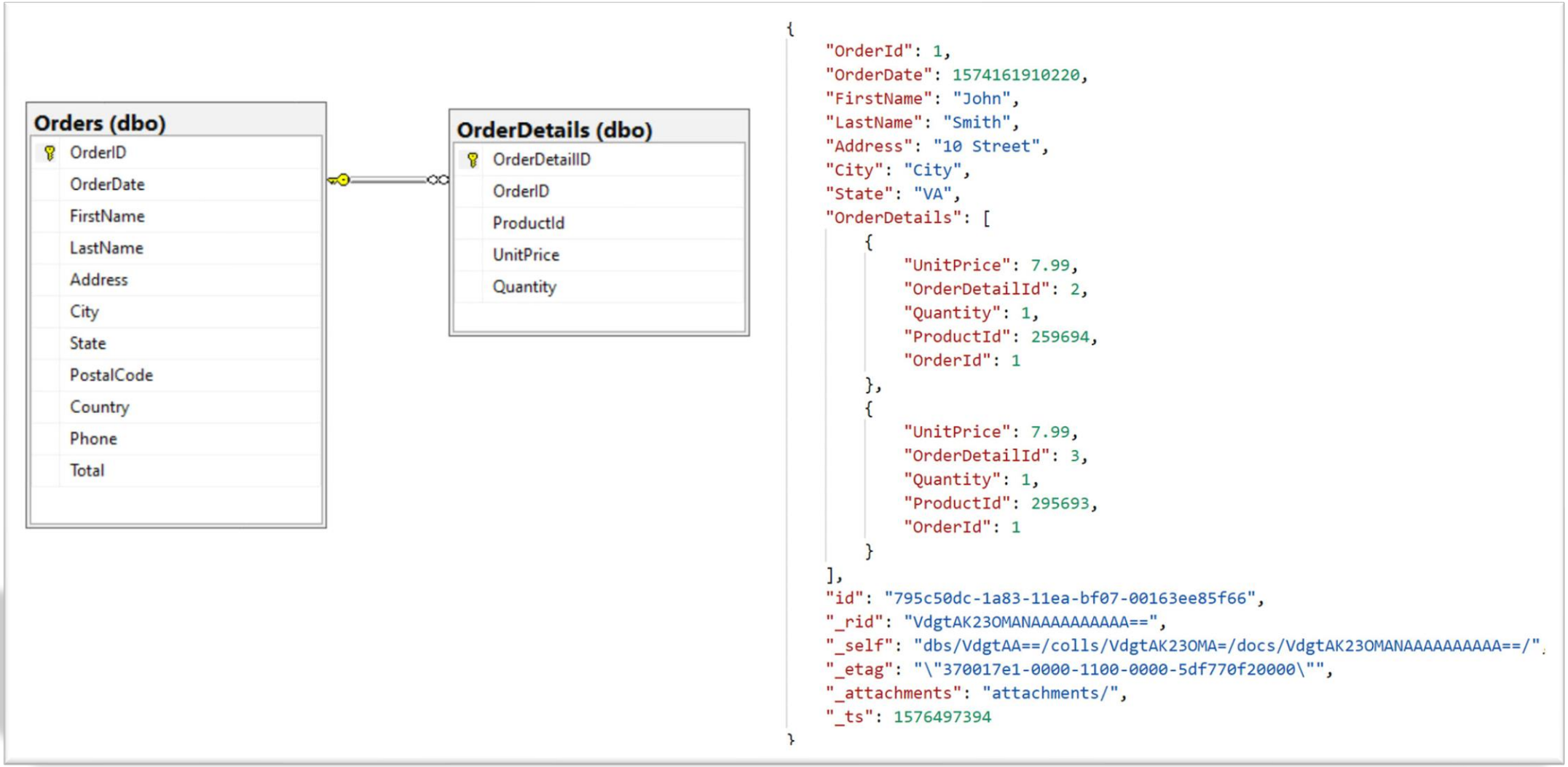
Flexibility

What is NoSQL



- Vertical scaling
  - Add more CPU, RAM, HDD in same system
- Horizontal Scaling
  - Add more commodity machines in system

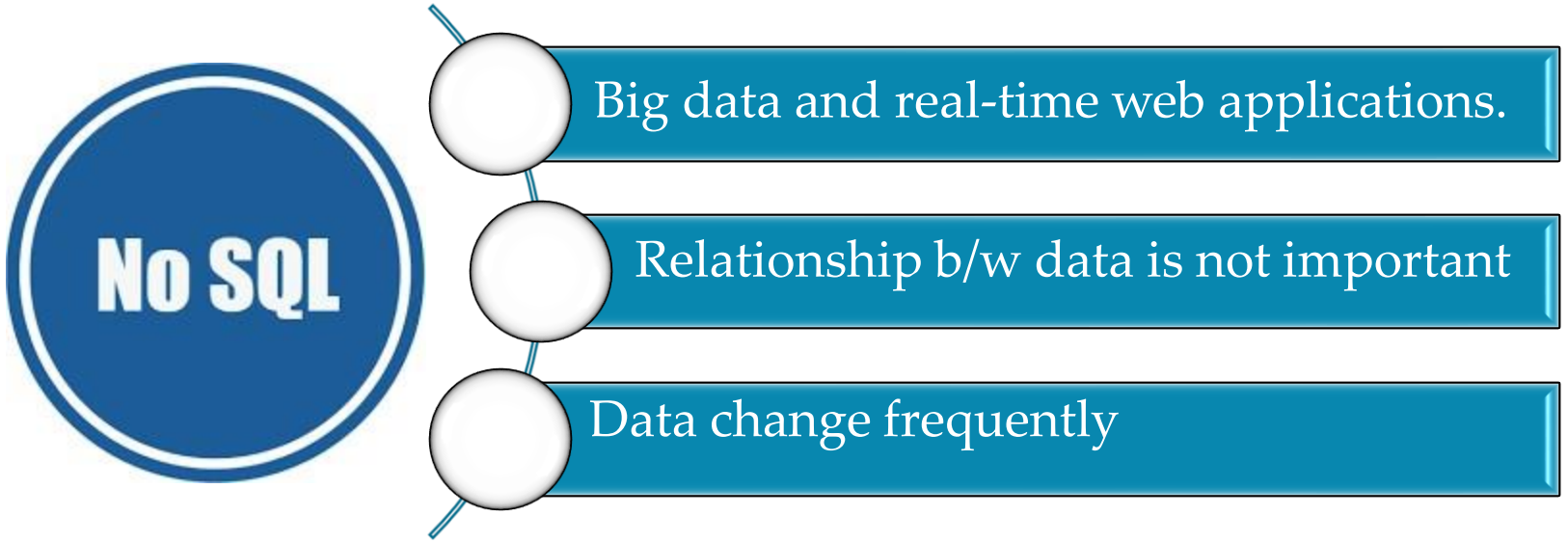




```
{
  "orderid": 12212,
  "orderdate": "12/4/2020",
  "customer":
      { "name": "Bob Smith", "email": "bobsmith@email.bob" },
  "status": "in process",
  "paymentmethod": "invoice",
  "products": [
      { "name": "Product 1", "quantity": 1 },
      { "name": "Product 2", "quantity": 1, status: 3 }
  ]
}
```


databag®

NoSQL Use Cases



NoSQL Limitations

- Schema-less data means inconsistent data
- Denormalized data means redundant data
- Redundant data means inaccuracies and conflicts
- Does not support many good features of Relational DB
  - SPs, Functions, Views, Row level security, Locks, etc.

databag®

SQL vs NoSQL

SQL

- Relational database
- Fixed schema
- Designed for complex queries
- SQL, MySql, Oracle, Postgres
- Vertical scaling
- Row Oriented
- Tables
- Limited for big data

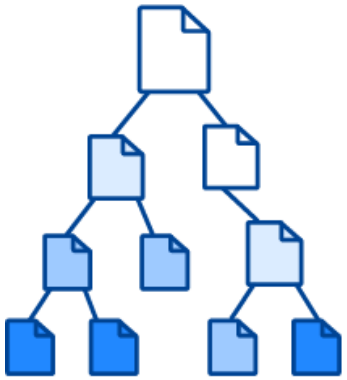
NoSQL

- Non-relational or distributed
- Dynamic
- Not for complex queries
- MongoDB, Redis, Hbase
- Horizontal scaling
- Multi-model oriented
- Collections
- Great for big data

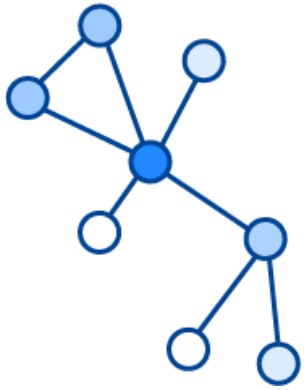
databag®

4 Types of NoSQL Databases

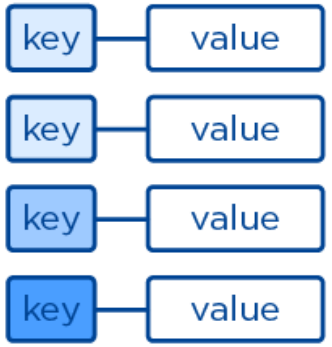
Document



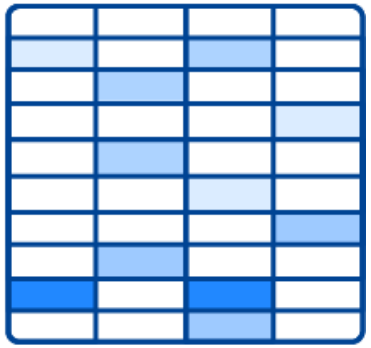
Graph



Key-Value



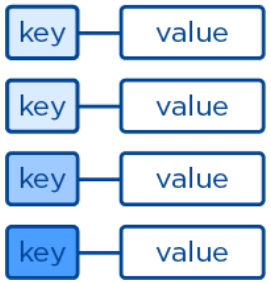
Wide-column



databag®

Key-value store

Key-Value



Phone Directory

Key	Value
Bob	(123) 456-7890
Jane	(234) 567-8901
Tara	(345) 678-9012
Tiara	(456) 789-0123

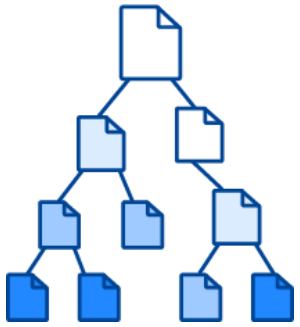
- Uses a simple key/value to store data
- Quick to query due to its simplicity
- Value can be JSON, BLOB, String etc.
- Use Cases:
  - User profiles and session info on a website, blog comments, telecom directories, IP forwarding tables, shopping cart contents on e- commerce sites, and more.
- Examples
  - Cosmos DB Table API, Redis, Table Storage, Oracle NoSQL Database,

    Voldemorte, Aerospike, Oracle Berkeley DB

databag®

Document store

Document

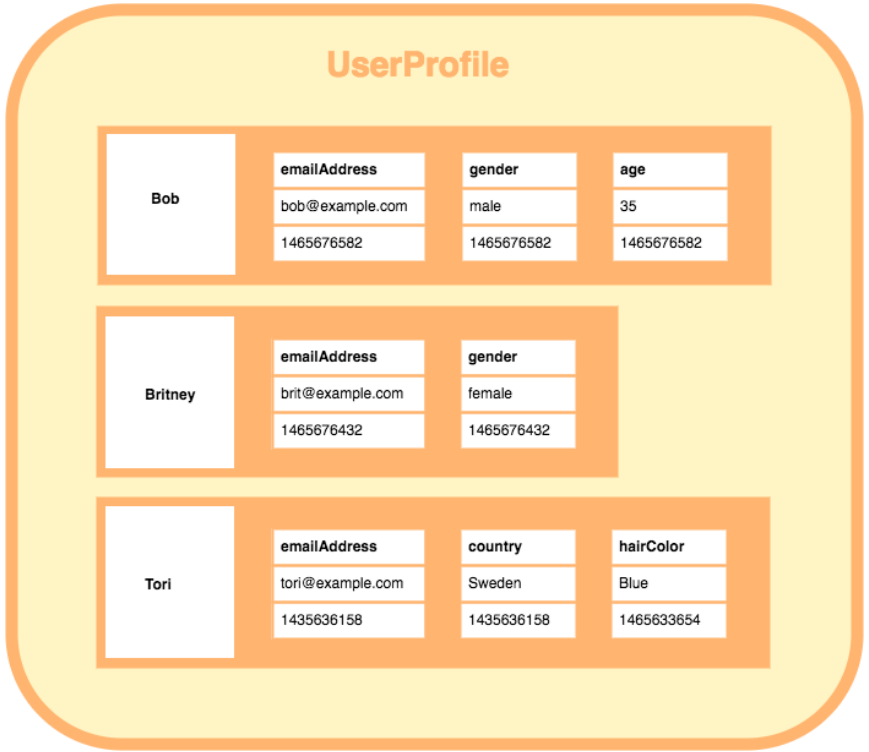


```
{
  "orderid": 12212,
  "orderdate": "12/4/2020",
  "customer":
      { "name": "Bob Smith", "email": "bobsmith@email.bob" },
  "status": "in process",
  "paymentmethod": "invoice",
  "products": [
      { "name": "Product 1", "quantity": 1 },
      { "name": "Product 2", "quantity": 1, status: 3 }
  ]
}
```

- Document-oriented model to store data
- Similar to key/value store, difference is that, the value in a document store database consists of semi-structured data.
- Each record and its associated data within a single document.
- Document stores are usually XML, JSON, BSON, YAML, etc.
- Use Cases:
  - Content management systems, blogging platforms, and other web applications, blog comments, chat sessions, tweets, ratings, etc.
- Examples
  - Cosmos DB, MongoDB, DocumentDB, CouchDB, MarkLogic, OrientDB

databag®

Column store



- Stores data using a column oriented model
- Columns in each row are contained within that row
- Each row can have different columns to the other rows.
- Extremely quick to load and query
- Use Cases:
  - Sensor Logs [Internet of Things (IOT)], User preferences, Geographic

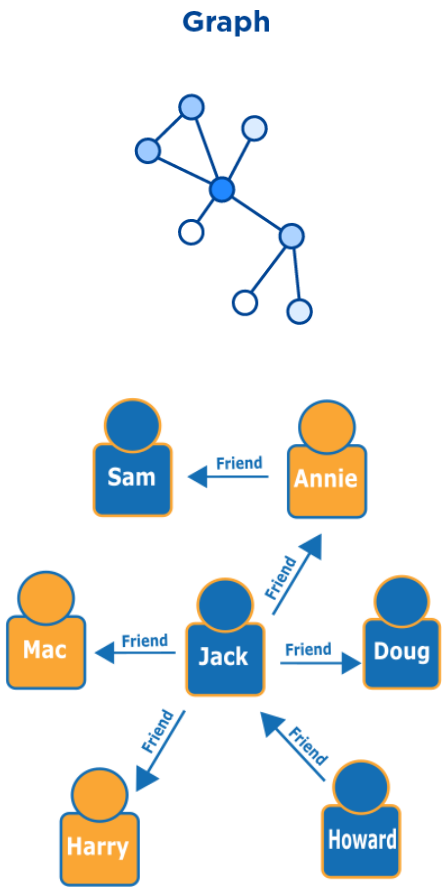
    information, Reporting systems, Time Series Data, Logging

    and other write heavy applications
- Examples
  - Cosmos DB, Bigtable, Cassandra, Hbase, Vertica, Druid, Accumulo,

    Hypertable

databag®

Graph store



- Focuses on how data relates to other data points.
- A node is a specific entity or piece of information
- Edge simply specifies the relationship between two nodes.
- Use Cases:
  - Social networks, realtime product recommendations, network diagrams, fraud detection, access management, and more.
- Examples
  - Cosmos DB Gremlin API, Neo4j, Blazegraph, and OrientDB.

databag®

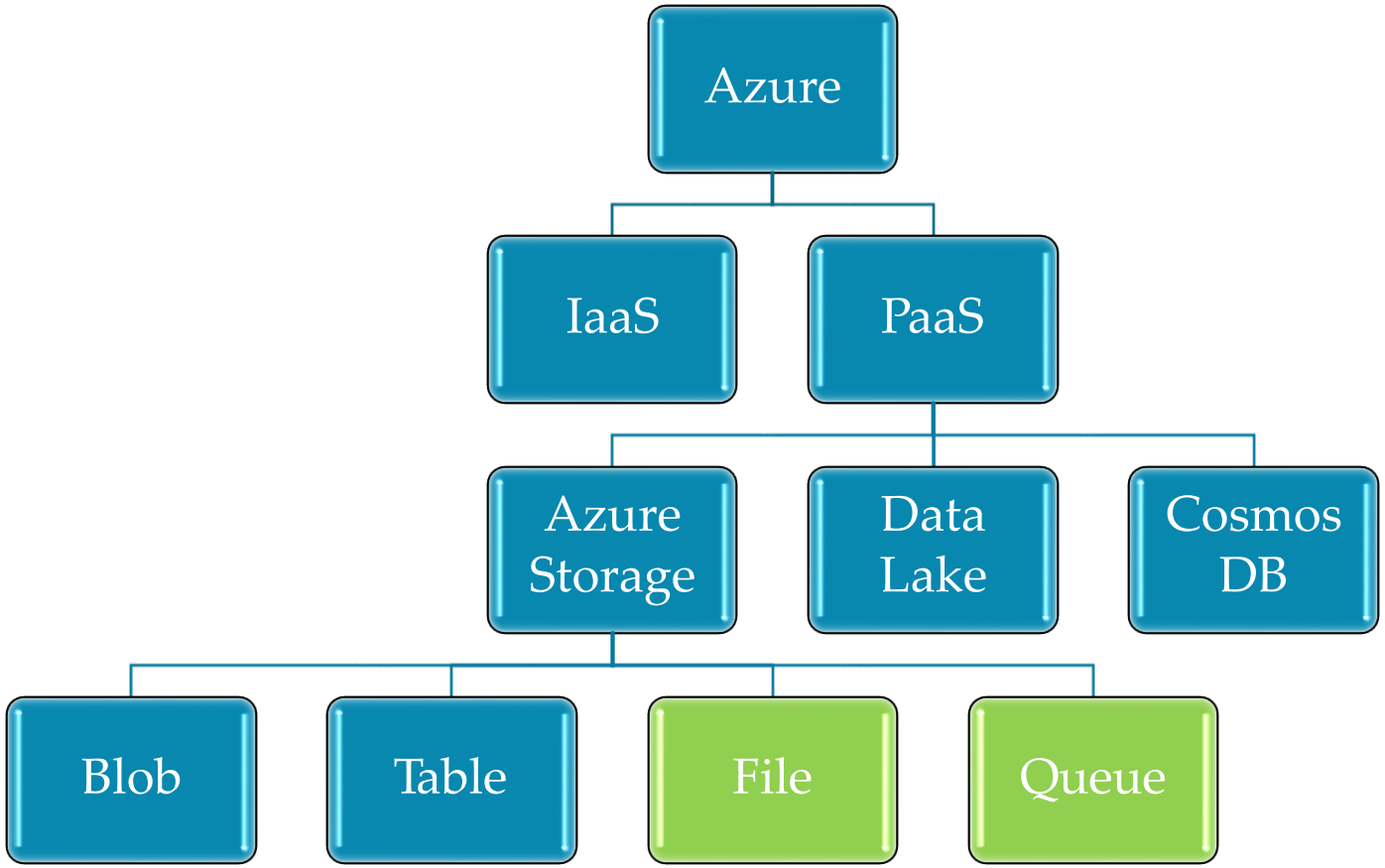
Multi-model

- Include features/characteristics of more than one data model.

- **Example:**
  - **OrientDB:** OrientDB combines a graph model with a document model.
  - **ArangoDB:** Uses key/value, document, and graph models.
  - **Virtuoso:** Combines relational, graph, and document models.

databag®

NoSQL Offerings by Microsoft Azure



databag®

Advantages of Blob storage

- Extremely cheap
- Simple to setup
- No configuration
- Doesn't require powerful computing to manage

Limitations of Blob storage

- No Indexes
- No Search Tools
- Not optimized for performance
- You are responsible for replication and synchronization
- Requires external compute to process

databag®

What is Cosmos DB?

Globally Distributed multi model database service for mission critical applications

Why Cosmos DB?

FULLY MANAGED

- Database as a service (DaaS)
- Serverless architecture
- No operational overhead
- No schema or Index management

CONSISTENCY CHOICES

- Azure Cosmos DB's support for consistency levels like strong, eventual, consistent prefix, session, and bounded-staleness.

SCALABLE

- Unlimited scale for both storage and throughput.

GLOBALLY DISTRIBUTED

- Turnkey global distribution

MULTIMODEL & MULTI-LANGUAGE

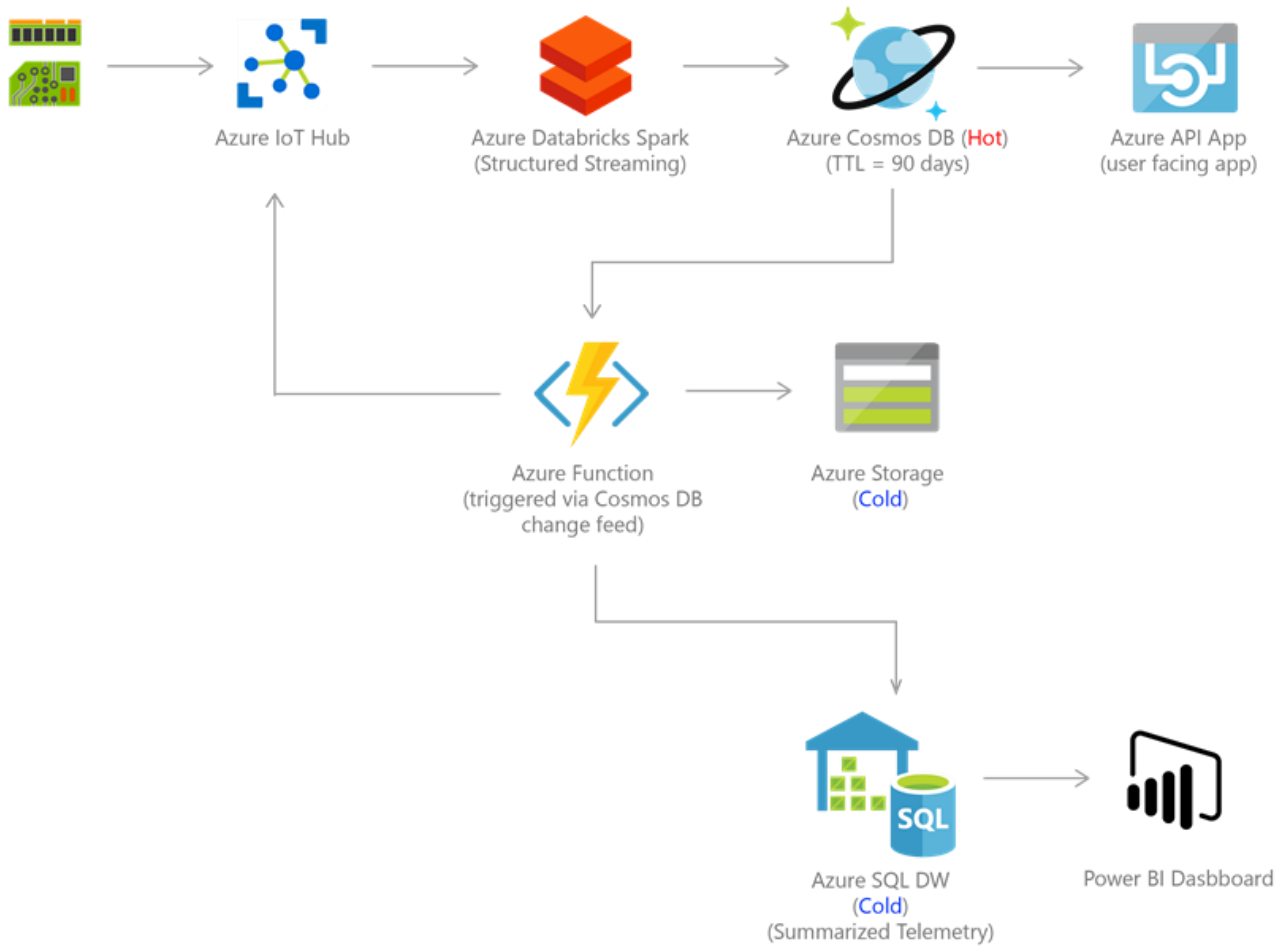
- Supports Json documents, table graph and columnar data models
- Java, .NET, Python, Node.js, JavaScript, etc.

HIGHLY AVAILABLE, RELIABLE & SECURE

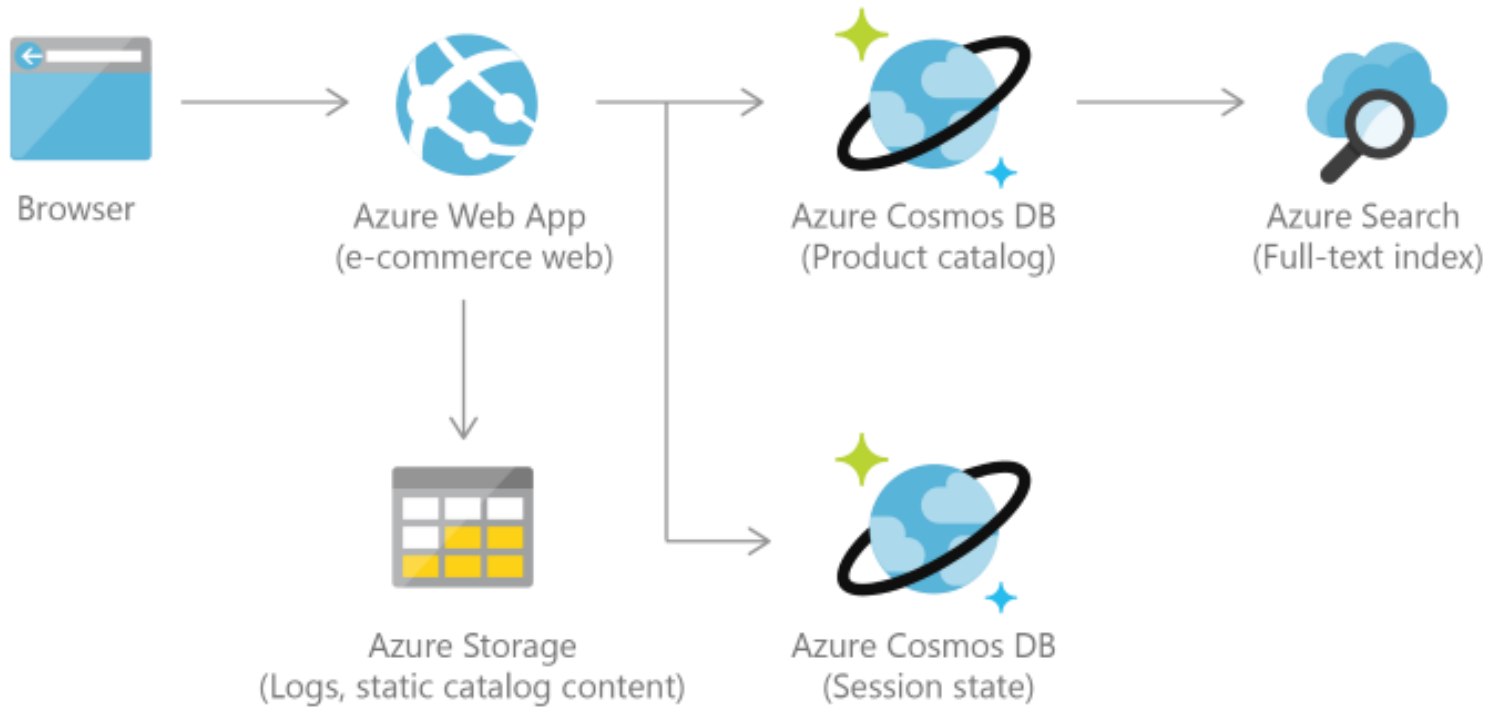
- Always on
- 99.999% SLA
- < 10ms latency

databag®

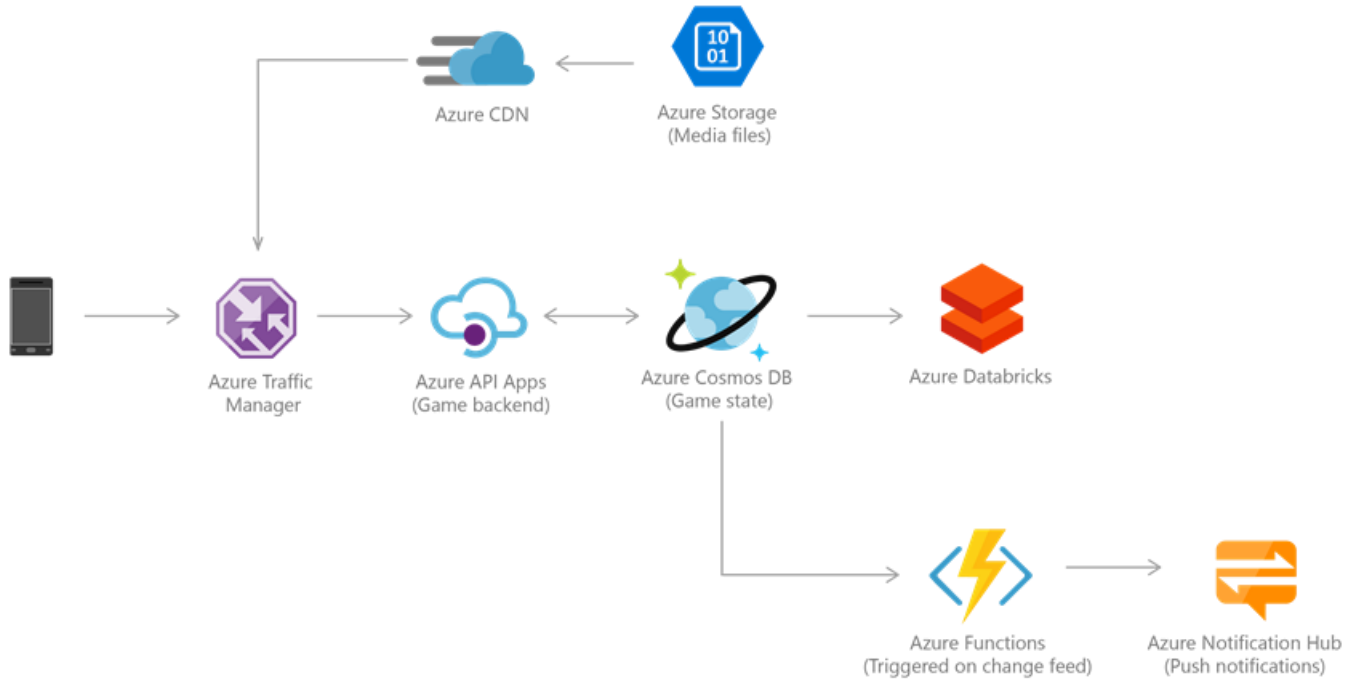
Use case - IOT



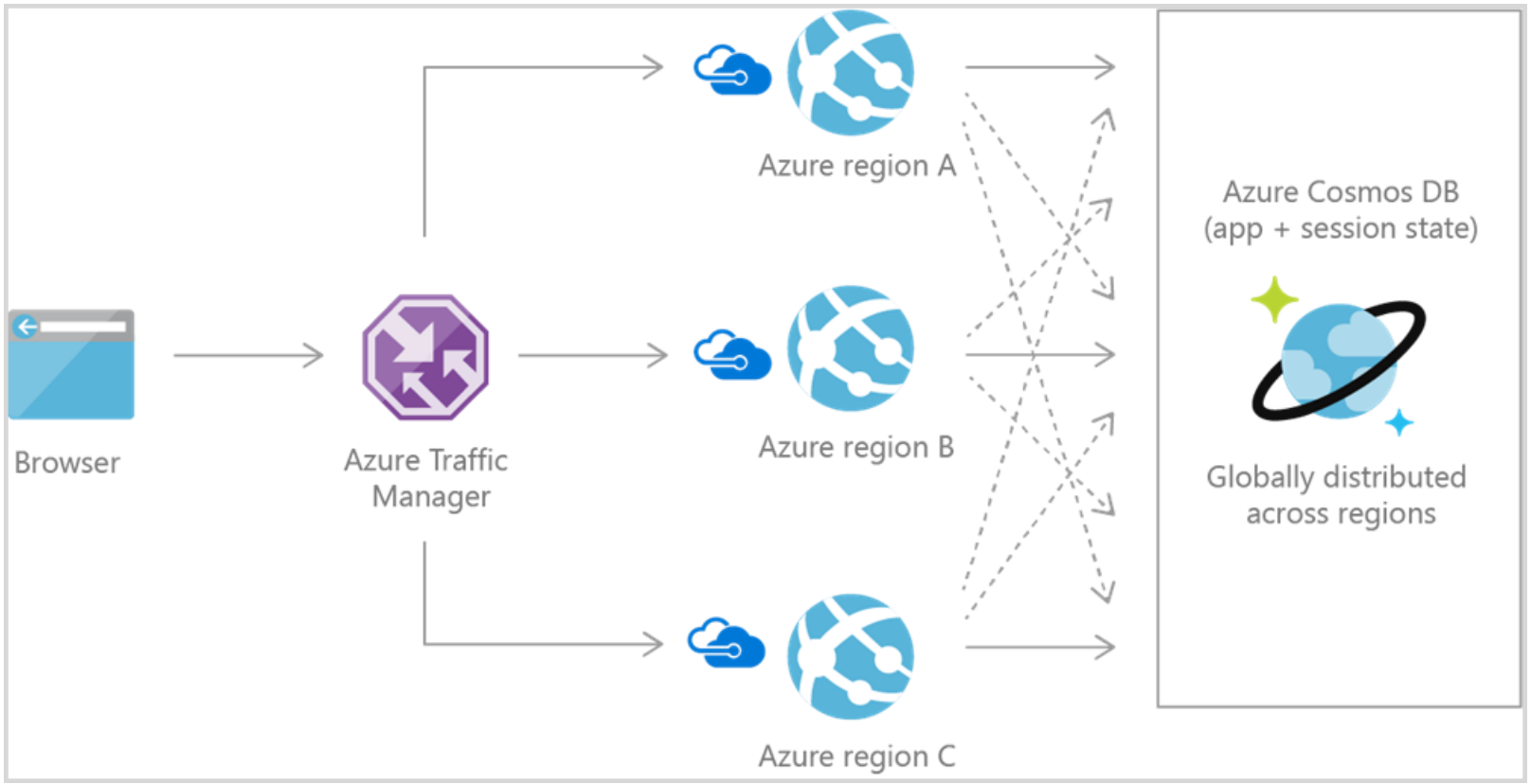
Use case – Retail and Marketing

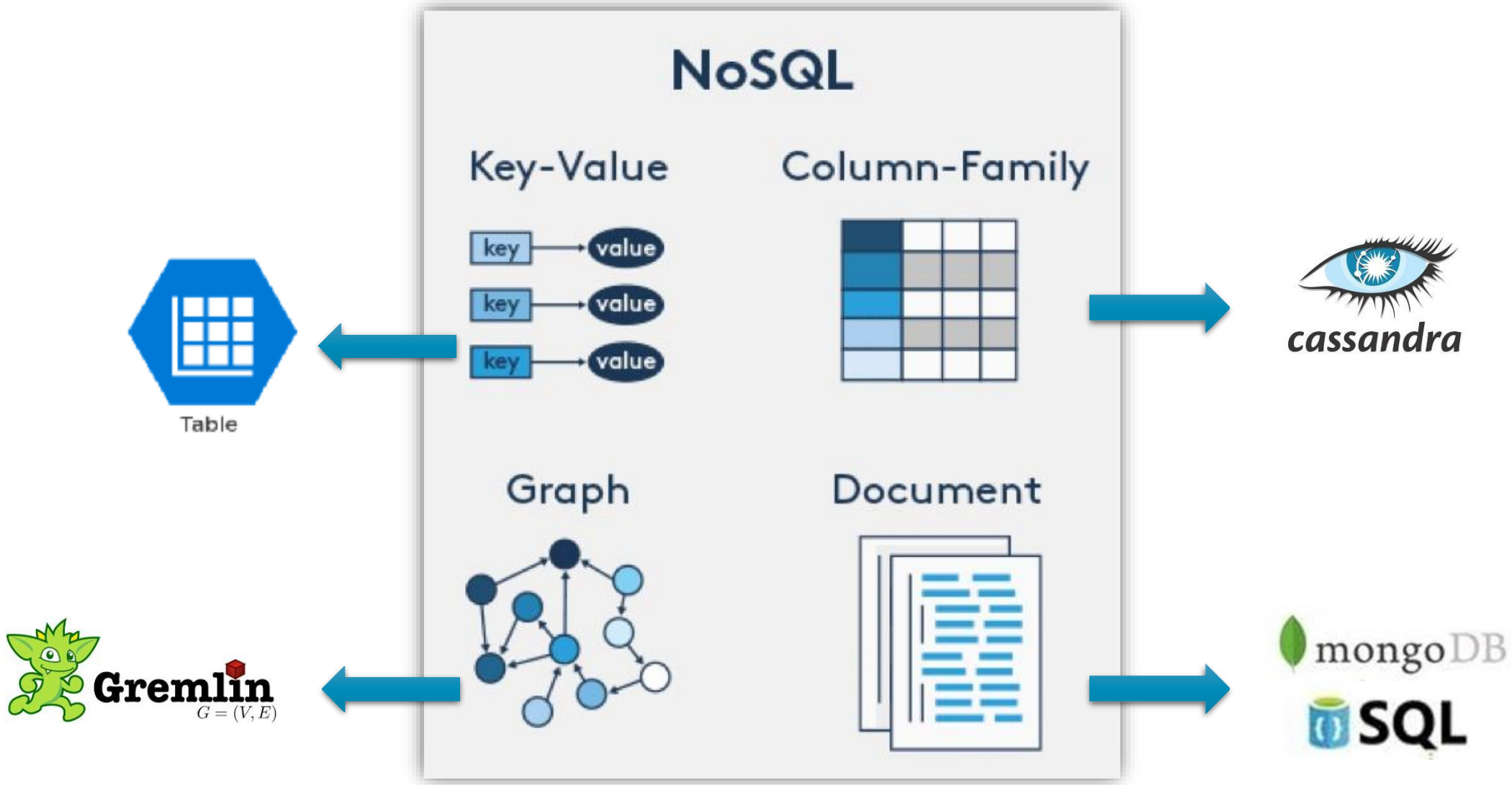


Use case – Gaming



Use case – Web and mobile





databag®

SQL API vs MongoDB API

SQL(CORE) API

JSON Documents

Microsoft original Document DB platform
Supports server side programming model

You can use SQL like language to query JSON documents.

MongoDB API

BSON Documents

Implement Wire protocol

Fully compatible with Mongo DB application code

Migrate existing Cosmos DB without much change of logic

Use SQL(CORE) API for new development

databag®

JSON File

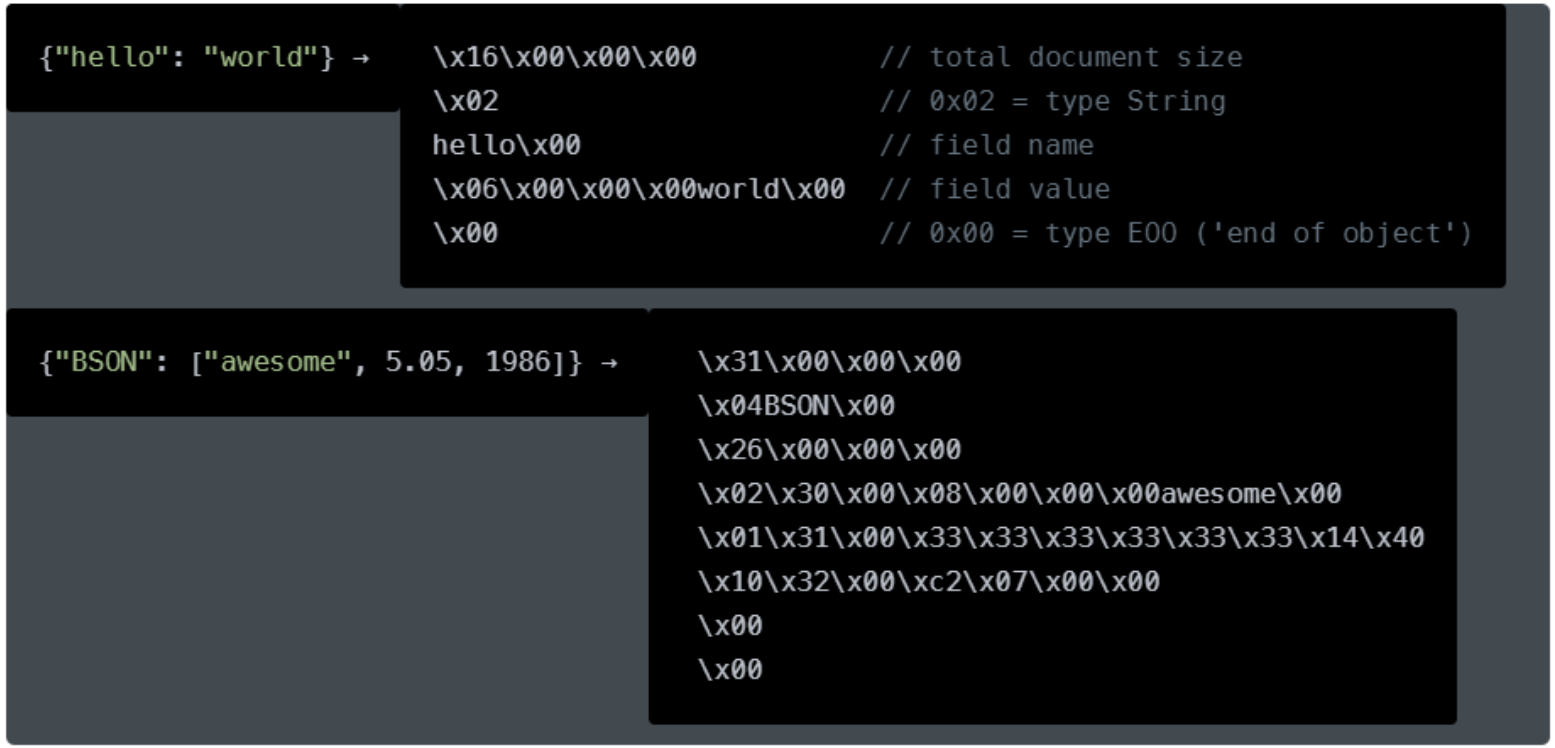
JavaScript objects are simple associative containers, wherein a string key is mapped to a value (which can be a number, string, function, or even another object)

```
{
  "_id": 1,
  "name" : { "first" : "John", "last" : "Backus" },
  "contribs" : [ "Fortran", "ALGOL", "Backus–Naur Form", "FP" ],
  "awards" : [
    {
      "award" : "W.W. McDowell Award",
      "year" : 1967,
      "by" : "IEEE Computer Society"
    }, {
      "award" : "Draper Prize",
      "year" : 1993,
      "by" : "National Academy of Engineering"
    }
  ]
}
```


databag®

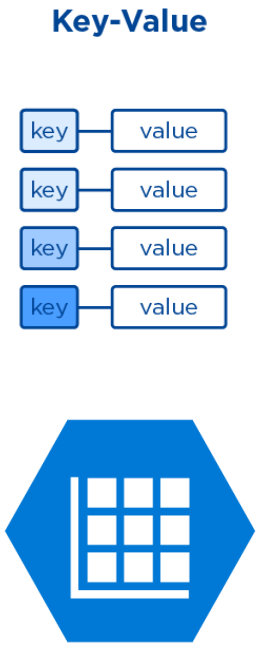
BSON File

BSON simply stands for “Binary JSON,” and that’s exactly what it was invented to be. BSON’s binary structure encodes type and length information, which allows it to be parsed much more quickly.



databag®

Cosmos DB Table API

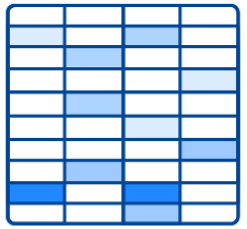


- Key-Value store
- Premium offering for Azure Table Storage
- Existing Table Storage customers will migrate to Cosmos DB Table API
- Row value can be simple like number or string
- Row cannot store object

databag®

Cosmos DB Cassandra API

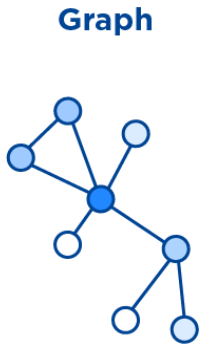
Wide-column



- Wide column No SQL Database
- Name and format of column can vary from row to row.
- Simple migrate your Cassandra application to Cosmos Cassandra API and change connection string.
- Interact
  - Cassandra based tools
  - Data Explorer
  - Programmatically, using SDK (CassandraCSharpdriver)

databag®

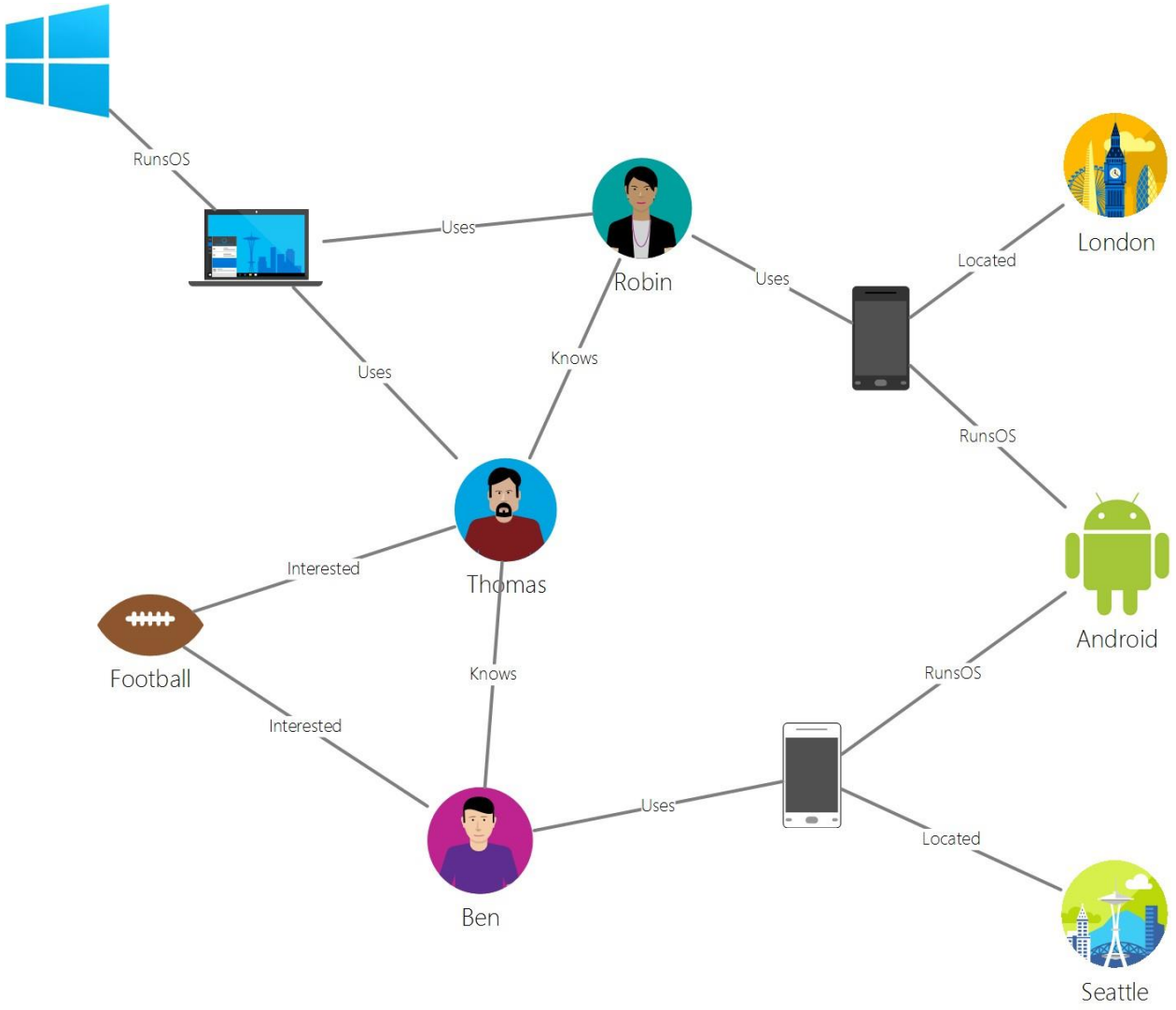
Cosmos DB Gremlin API



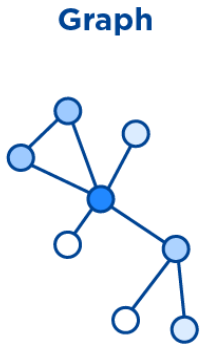
- Graph Data Model
- Real world data connected with each other
- Graph database can persist relationships in the storage layer

databag®

Graph Model



Cosmos DB Gremlin API



- Graph Data Model
- Real world data connected with each other
- Graph database can persist relationships in the storage layer
- Use cases
  - Social networks
  - Recommendation engines
  - Geospatial
  - Internet of things
- Migrate existing apps to Cosmos DB Gremlin API
- Graph traverse a language

databag®

Analyze the decision criteria

	Core (SQL)	MongoDB	Cassandra	Azure Table	Gremlin
New projects being created from scratch	✓				
Existing MongoDB, Cassandra, Azure Table, or Gremlin data		✓	✓	✓	✓
Analysis of the relationships between data					✓
All other scenarios	✓				

databag®

Azure Table storage vs Cosmos DB Table API

- Cosmos DB Table API is a prime version of Azure Table Storage

Azure Table Storage

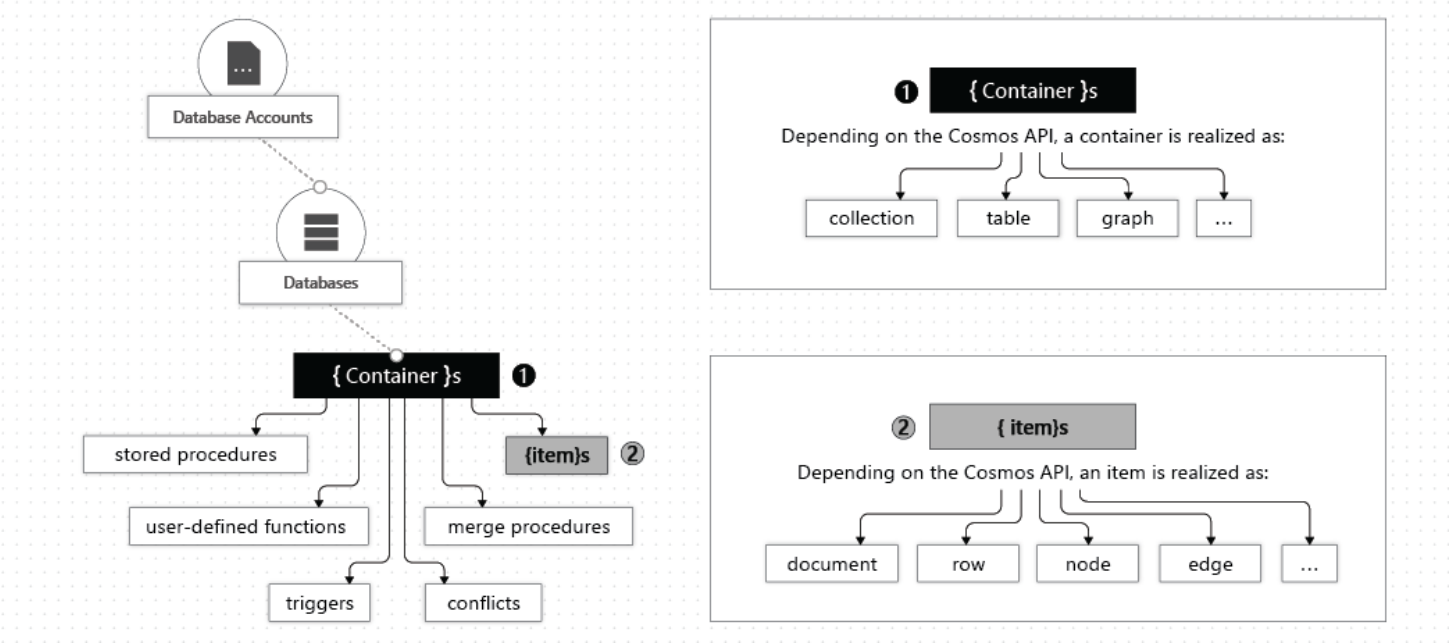
- Geo replication is restricted
  - Only 1 additional pair region
- Support for primary key lookups only
- Price optimized for cold storage
- Lower performance
  - Throughput is capped
  - Latency is higher
- No consistency options

Cosmos DB Table API

- Geo replication across your choice of any number of regions
- Secondary index support for lookups across multiple dimensions
- Better performance
  - Unlimited and predictable throughput
  - latency is lower
- 5 consistency options

databag®

Database Containers and Items



Azure Cosmos entity	SQL API	Cassandra API	MongoDB API	Gremlin API	Table API
Azure Cosmos database	Database	Keyspace	Database	Database	NA
Azure Cosmos container	Container	Table	Collection	Graph	Table
Azure Cosmos item	Document	Row	Document	Node or edge	Item

databag®