



DocumentDB customer  
welcome to

Azure Cosmos DB

ONE TO RULE THEM ALL |



# BUILD 2017

## Cosmos DB

Ing. Andreas Pollak

SpectoLogic® e.U.

[www.spectologic.com](http://www.spectologic.com)



# Agenda - CosmosDB

Overview

Resource Units

Gremlin Graph

Consistency

Global Distribution

Partitioning

SLA's & Emulator

# Cosmos DB

Global Distributed Database

Transparent multi-homing

99.99% high availability

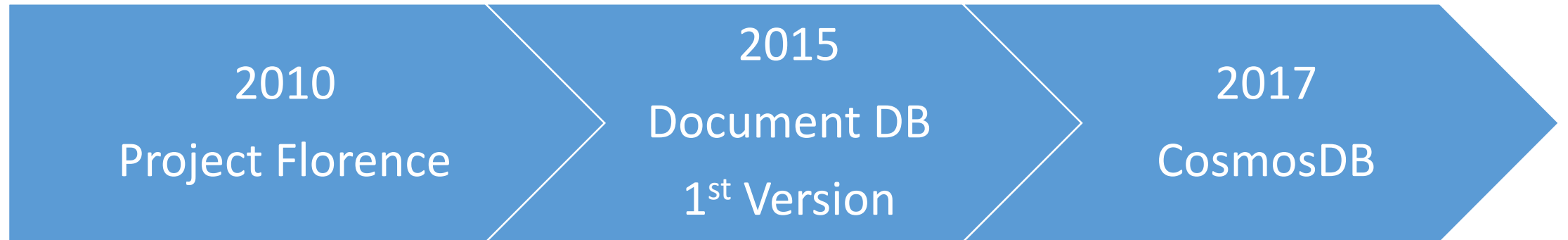
SLA backed

Schema agnostic

Automatically indexed



# History Lesson,...



# One base, multiple APIs

MONGO DB  
(Document)

SQL  
DocumentDB  
(Document)

Key/Value  
(Table Storage)

Gremlin  
Graph

Wide-Column  
(Table Storage)

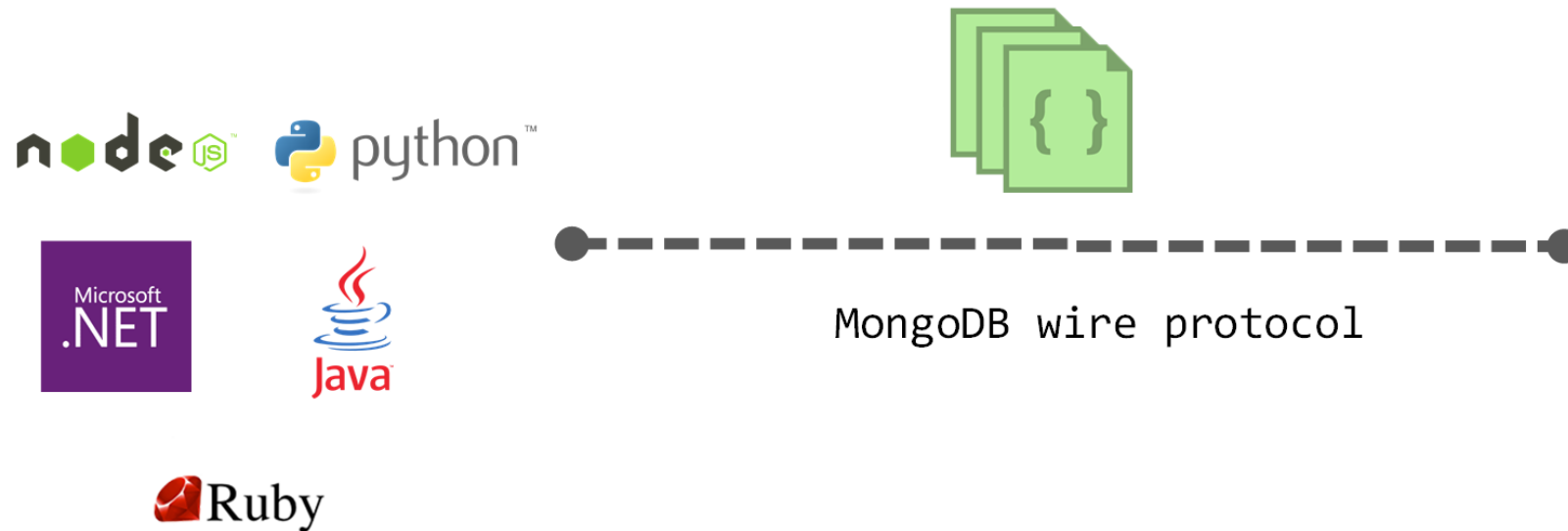
COSMOS DB – DISTRIBUTED DATABASE  
Consistency Levels, Indexing, Storage, Query Processor

# DocumentDB SQL



Azure Cosmos DB:  
DocumentDB API

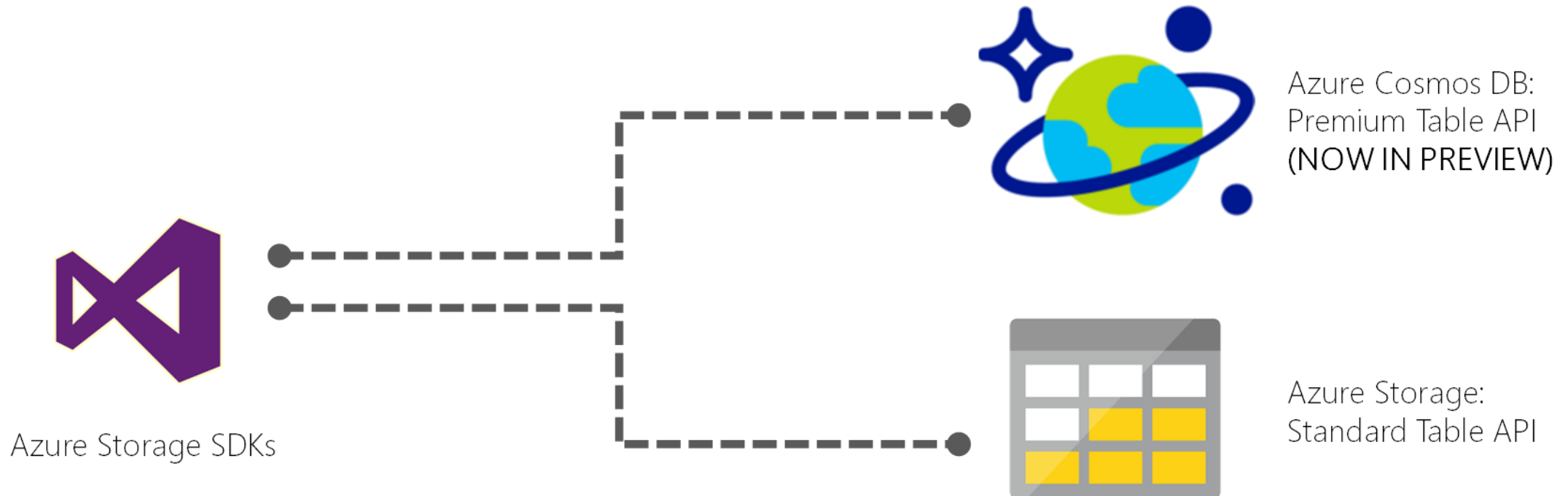
# MongoDB Native



Azure Cosmos DB:  
API for MongoDB

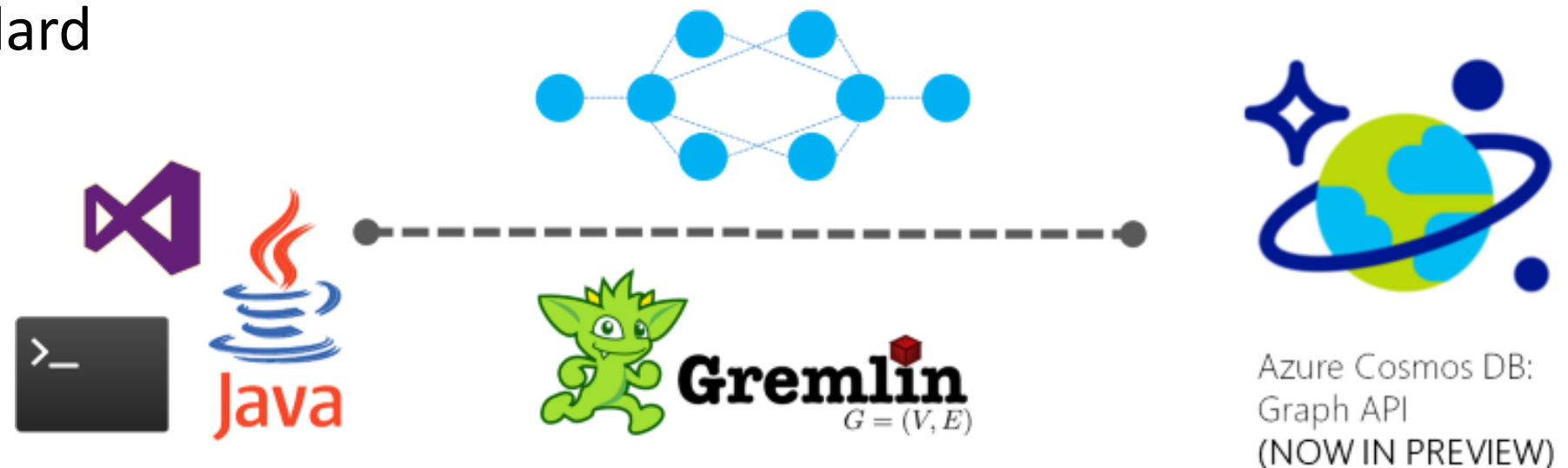


# Table Storage API

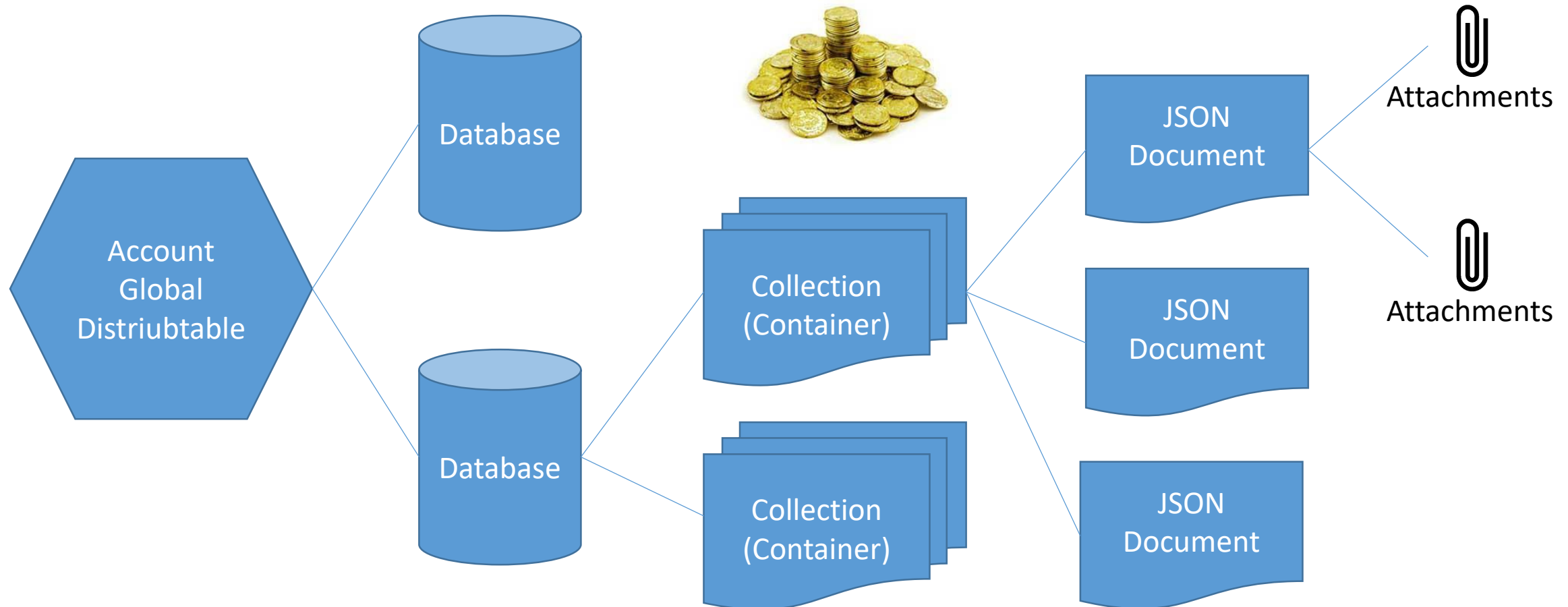


# Gremlin API

- Model the real world
- Relationships are first class citizens
- Optimized for graph storage and traversal
- Gremlin Standard



# Structure



# DEMO

Examining the storage structure

# Storage Capacity

**FIXED 10 GB (Small Tower)**

**UNLIMITED (Big Tower Bara-dûr)**



Actual used storage

# Request Unit aka Orc

- Computer, Memory and Routing Logic
- Security
- Bandwidth
- Network



# Request Units

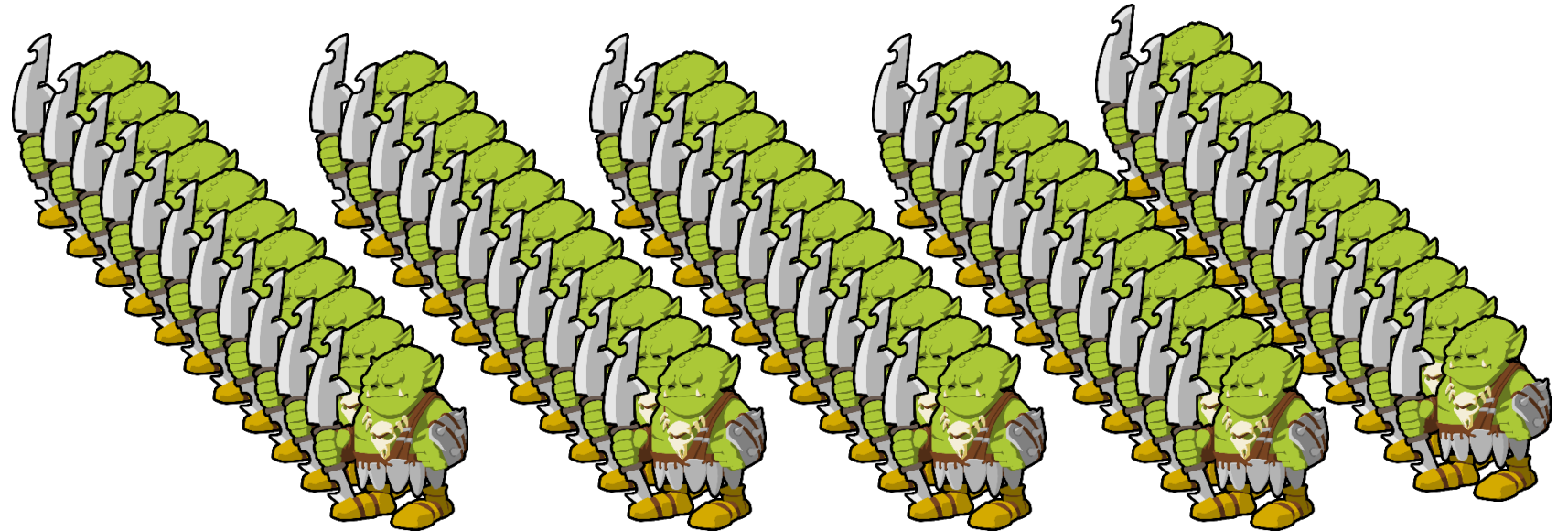
1 ORC can deliver 1 page parchment from Bara-dûr



# Request Units

How one paged parchments  
can be delivered by **50** Orcs?

100





# Request Units per Second

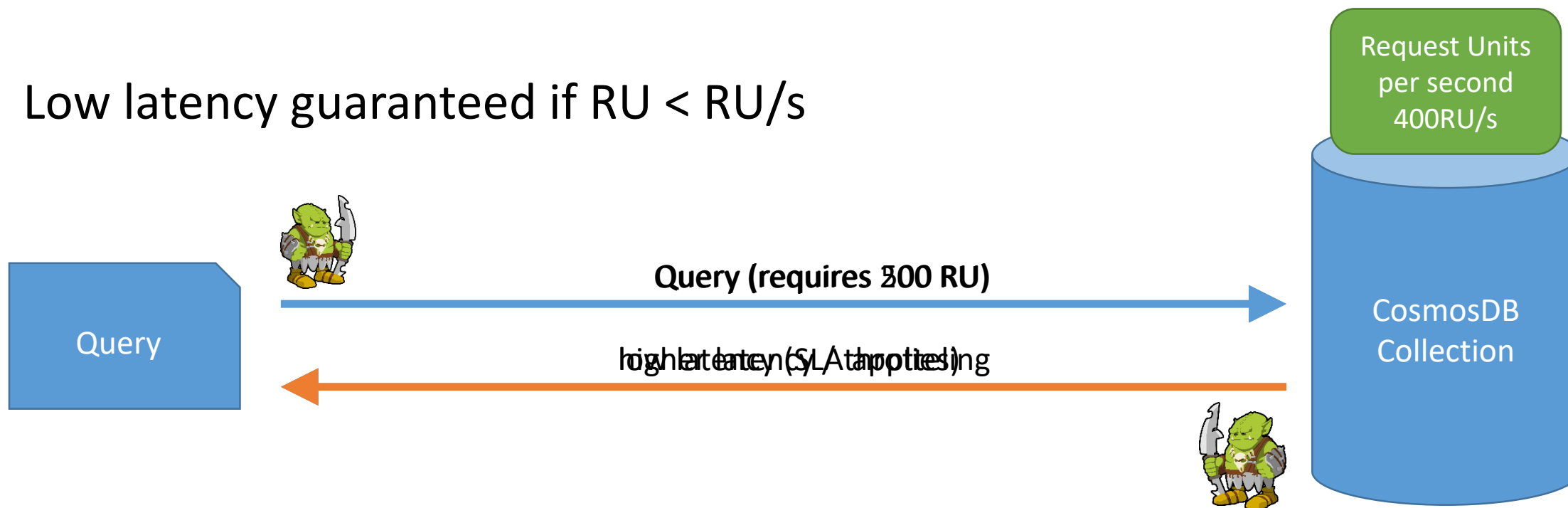
Guaranteed amount of Orcs per second at your disposal

Minimum 400 Orcs/s – Single partition

Minimum 2500 Orcs/s – Partitioned

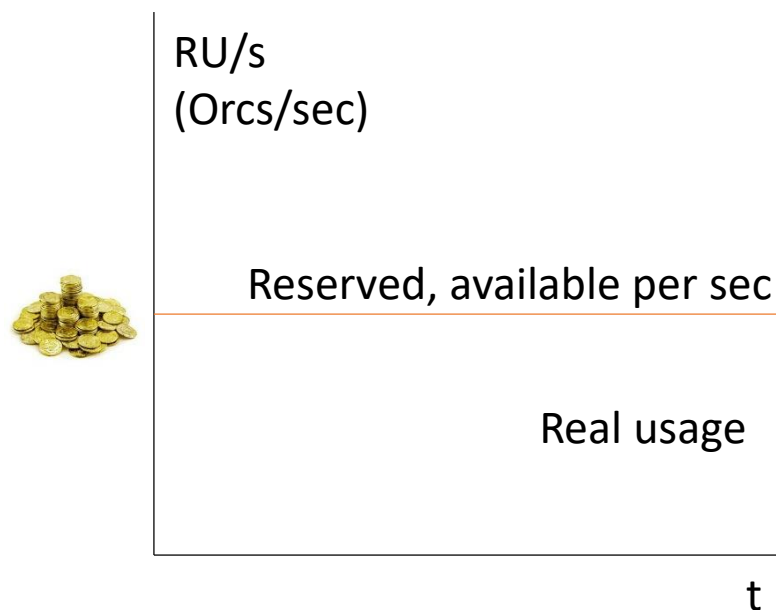
RU/s

Low latency guaranteed if  $RU < RU/s$

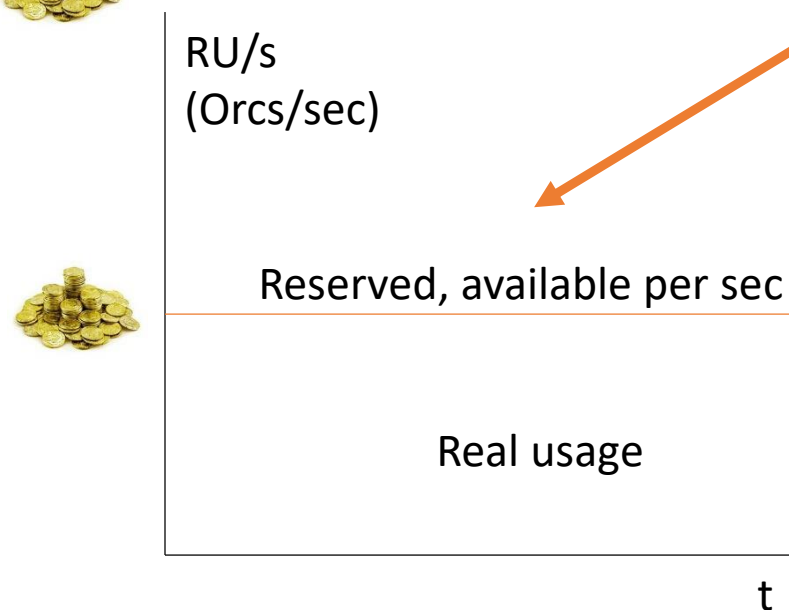


# Orcs/second cost gold coins,...

Constant load

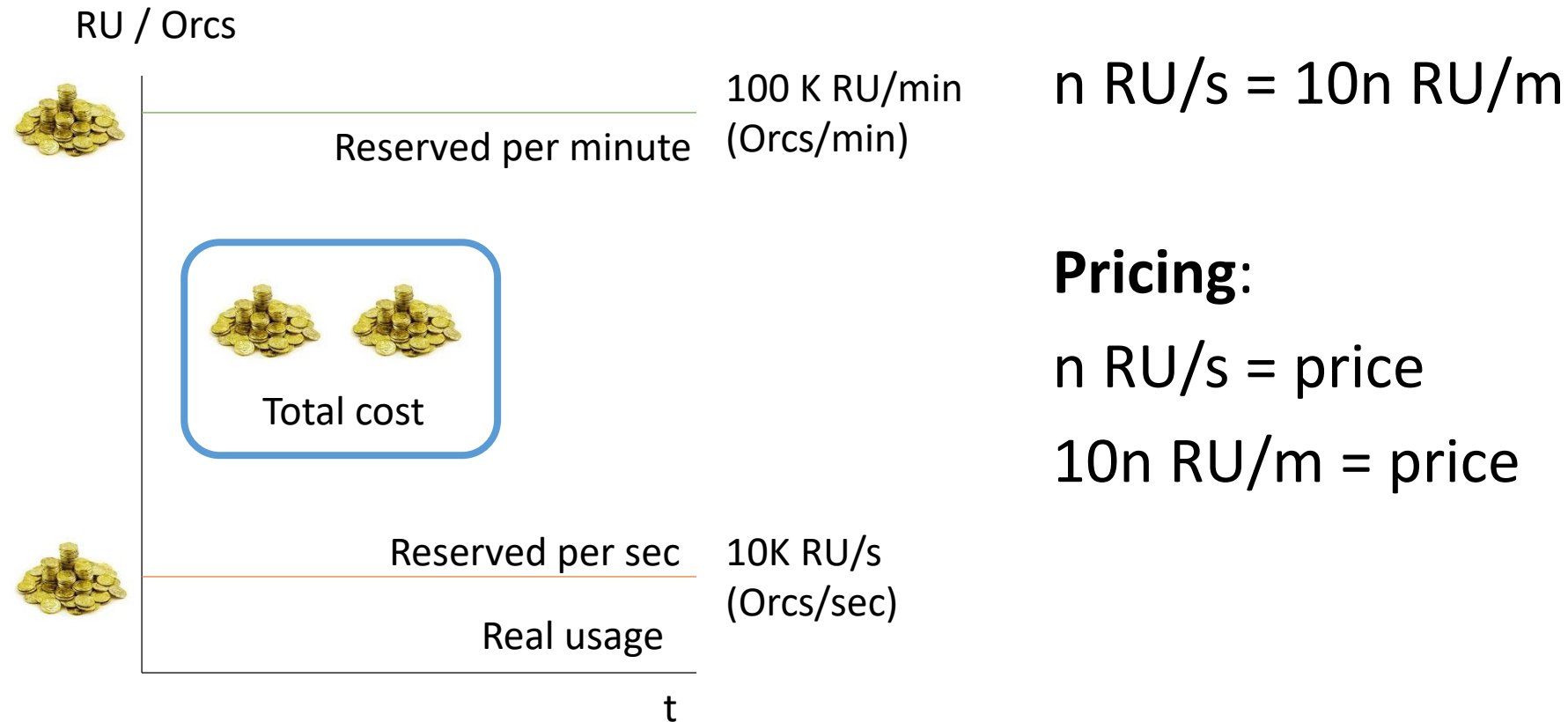


Spiky load



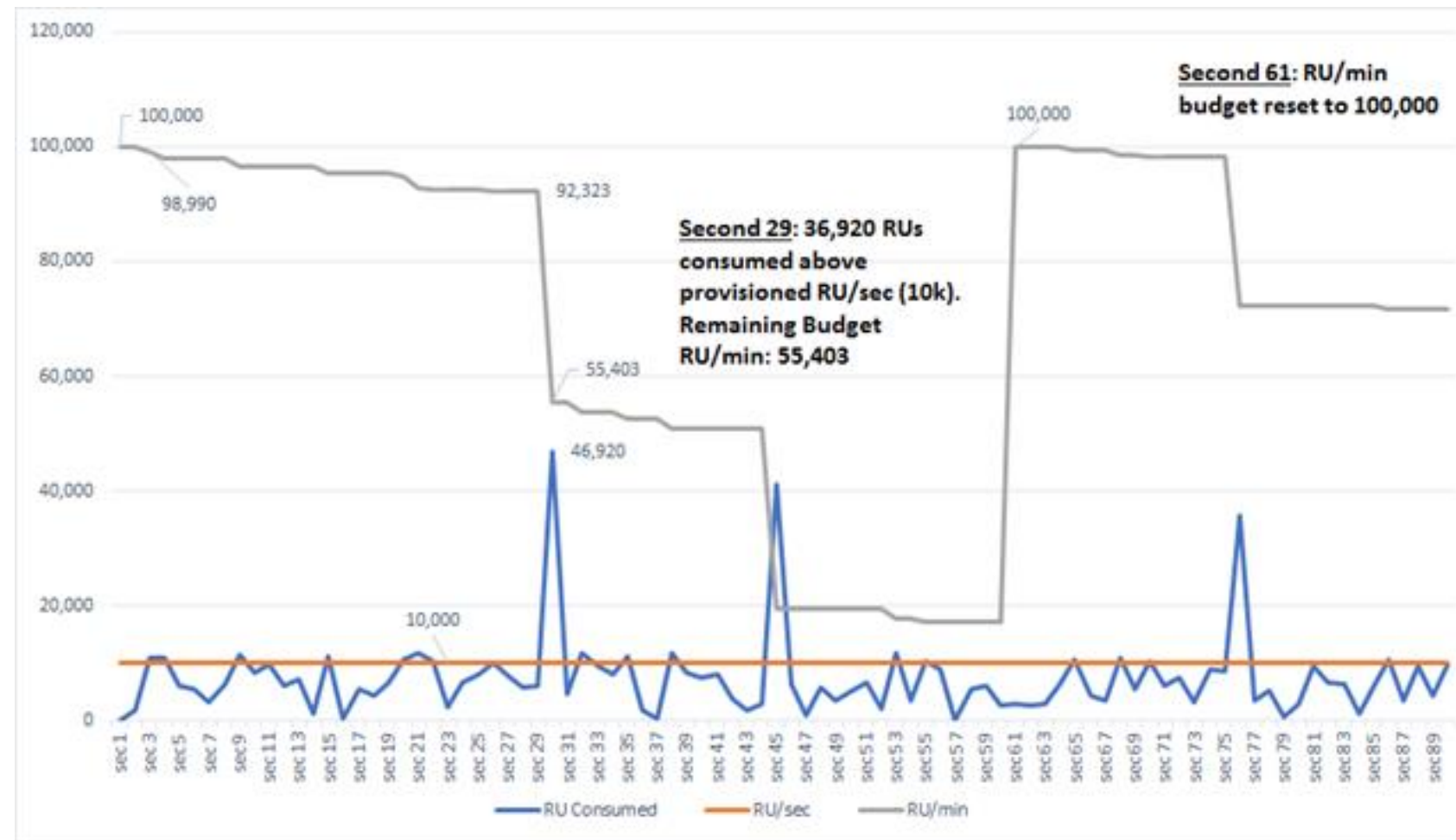
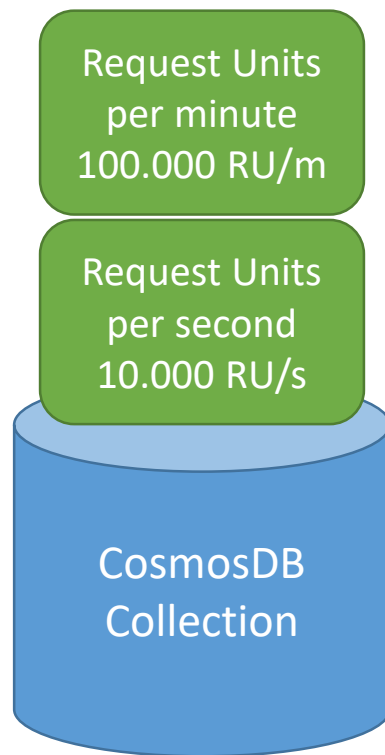
Latency  
Throttling

# Request Units || Orcs per Minute



50% discount on RU/m during preview

# How RU/m work,...



# Storage Capacity & guaranteed resources

**FIXED 10 GB (400-10.000 RU/s)**

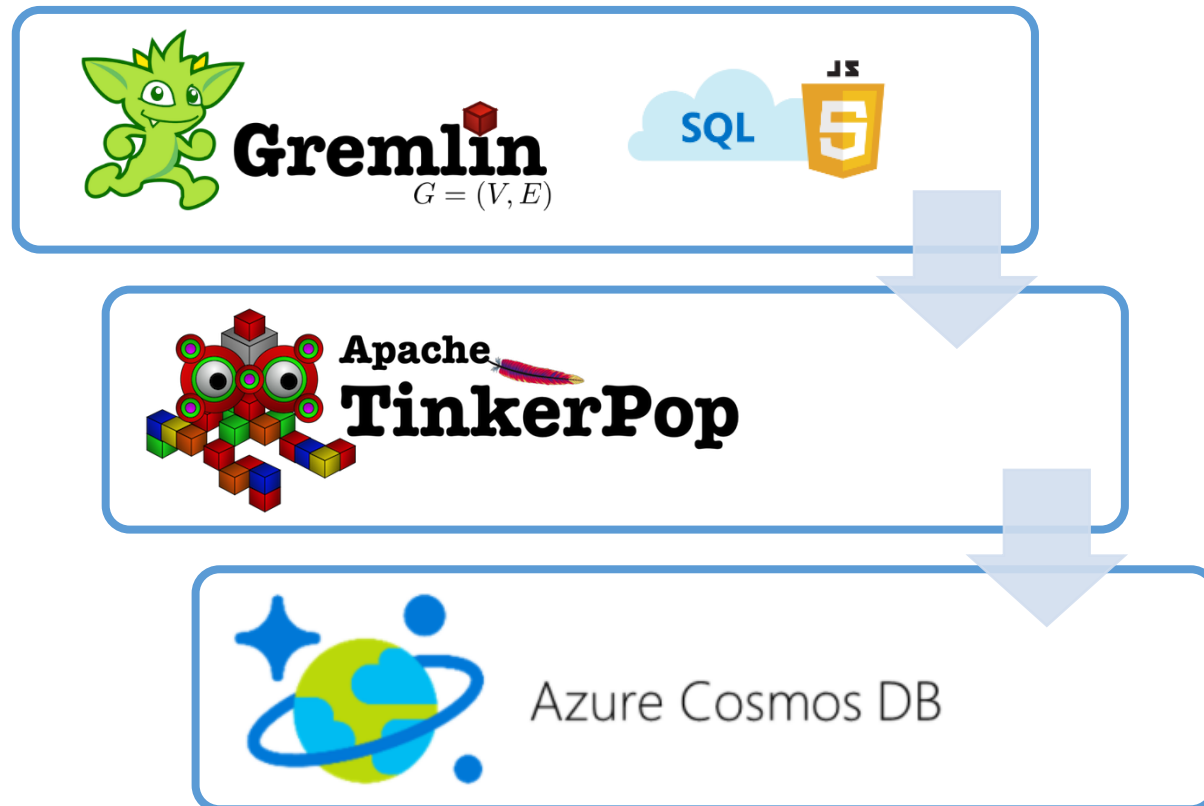


**UNLIMITED (2500-100.000 RU/s)**



The screenshot shows a web interface with a sidebar on the left and a main content area on the right. The sidebar contains the following items: 'Locks' (with a padlock icon), 'Automation script' (with a download icon), 'COLLECTIONS' (header), 'Browse' (with a magnifying glass icon), 'Scale' (with a scale icon, circled in red), and 'Query Explorer' (with a magnifying glass icon). The main content area on the right has the following settings: '\* THROUGHPUT (RU/s)' with a value of '1000' and a note 'Between 400 and 10000 RU/s'; 'RU/m' with a toggle switch set to 'ON'; and 'DEFAULT STORAGE CAPACITY' with a value of '10 GB'.

# Native Graph Processing



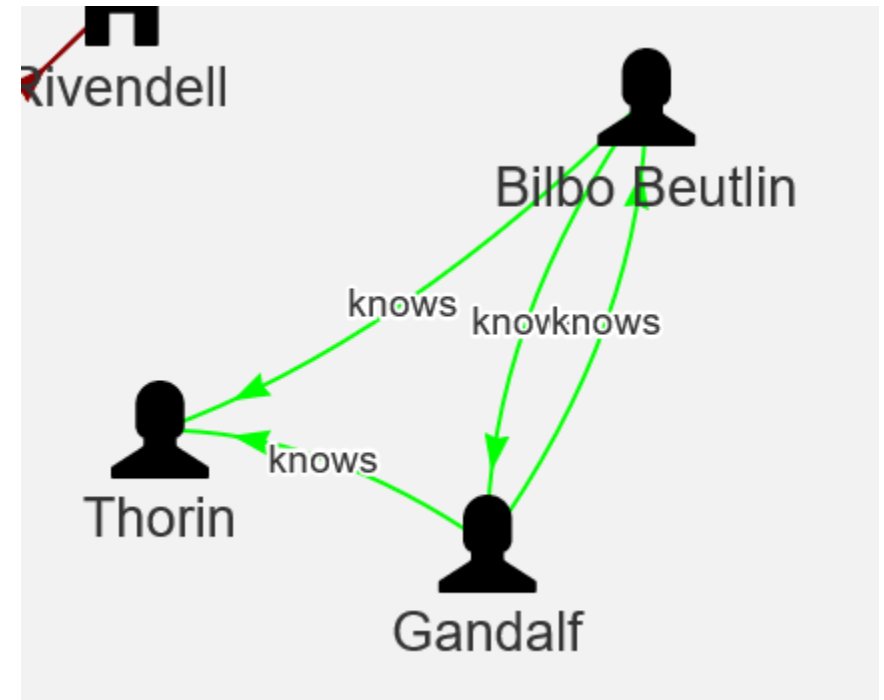
# Vertices & Edges

## Vertices

- Label, id
- Properties
- inE,outE

## Edges

- Label, id
- Properties
- inV,outV





# DEMO

Gremlin Graph & The Hobbit



# Data Consistency - Choose!

## **RED PILL**

Strong Consistency

High Latency



## **BLUE PILL**

Eventual Consistency

Low Latency



# New Consistency Level – Guess which?

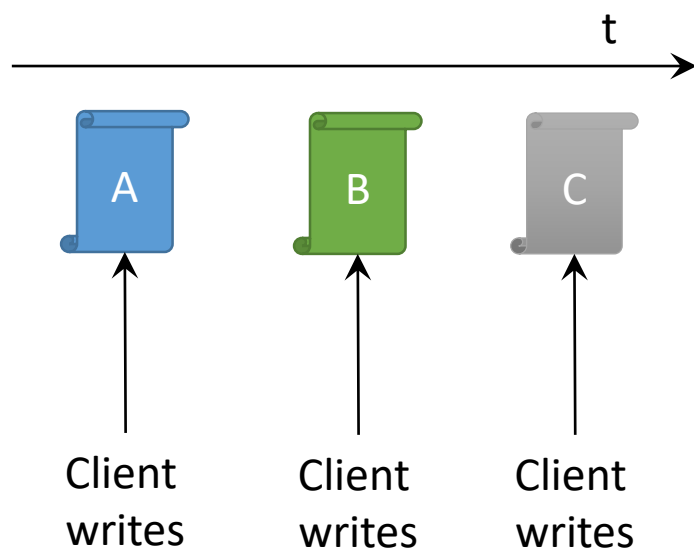


Overridable per request basis

# Consistent Prefix

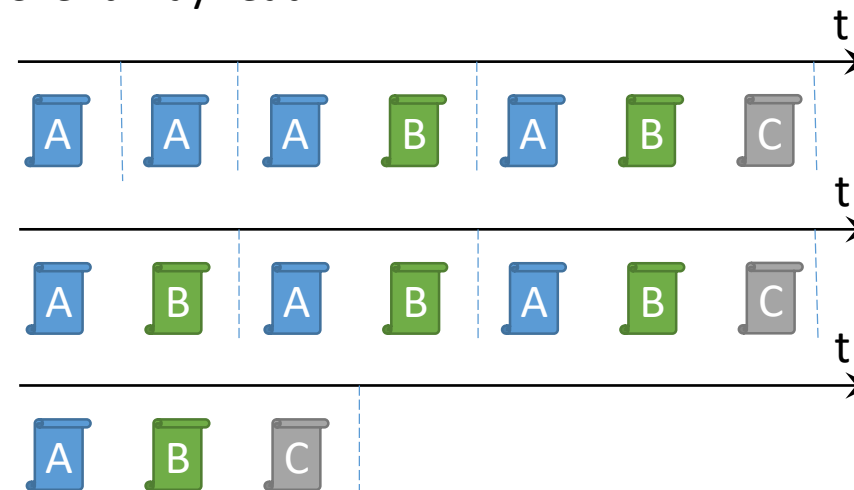
- Eventually converges
- Never sees out of order writes

# Consistent Prefix

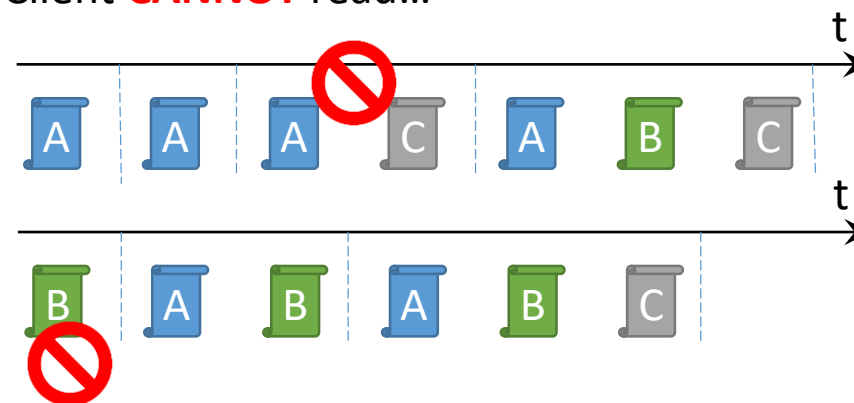


Three different documents!

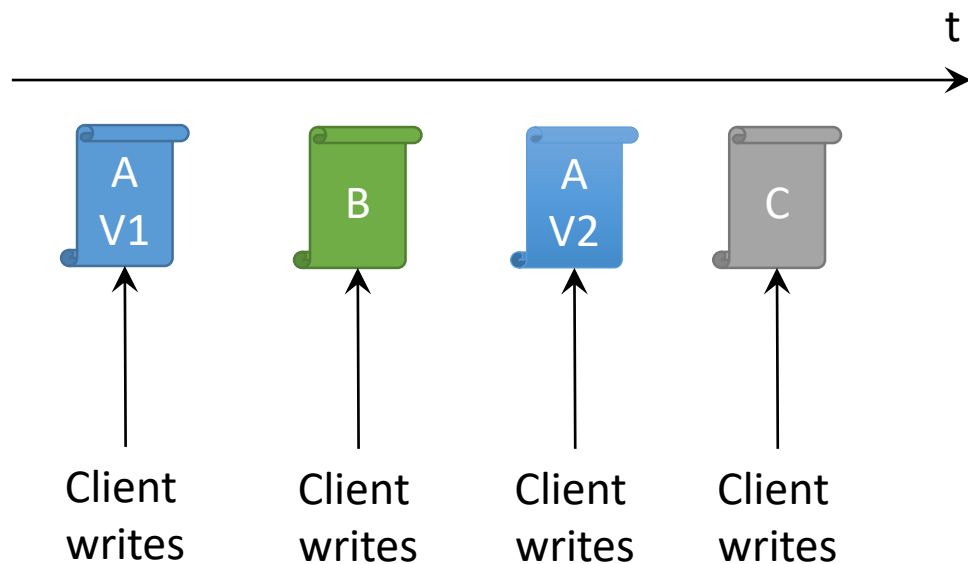
Client may read...



Client **CANNOT** read...

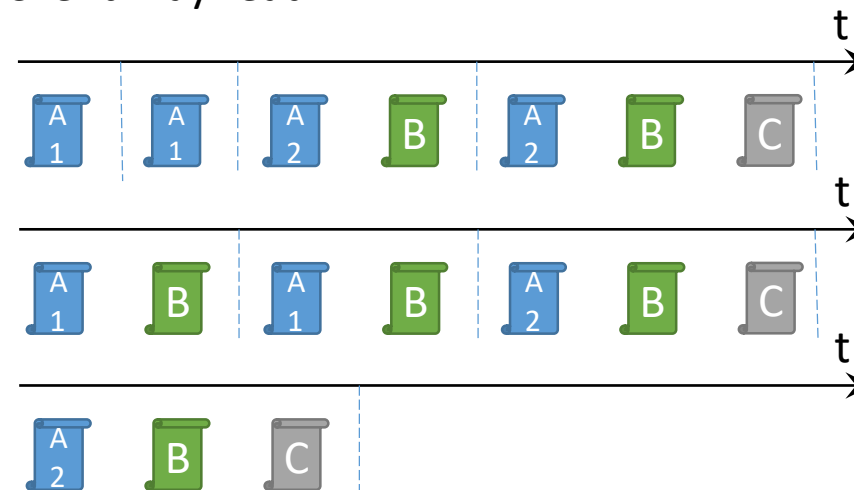


# Consistent Prefix

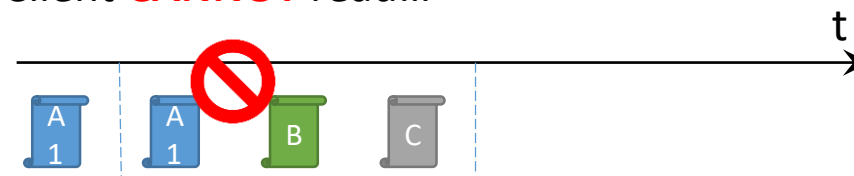


Three different documents!

Client may read...

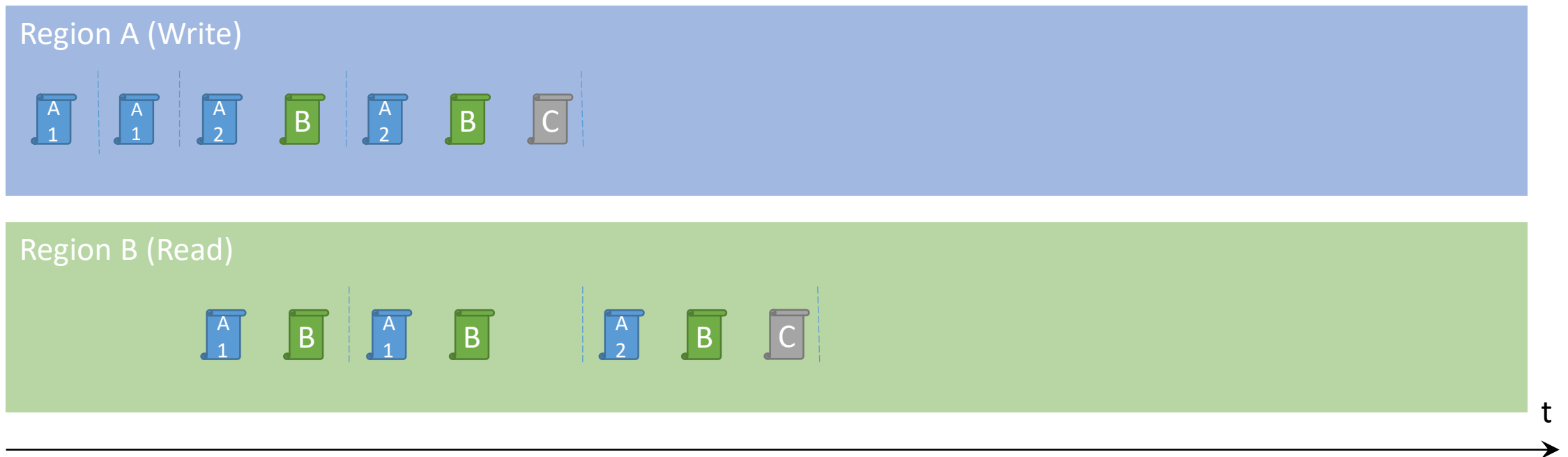


Client **CANNOT** read...



# Consistent Prefix

Two clients may read different versions in separate regions



# Session Consistency

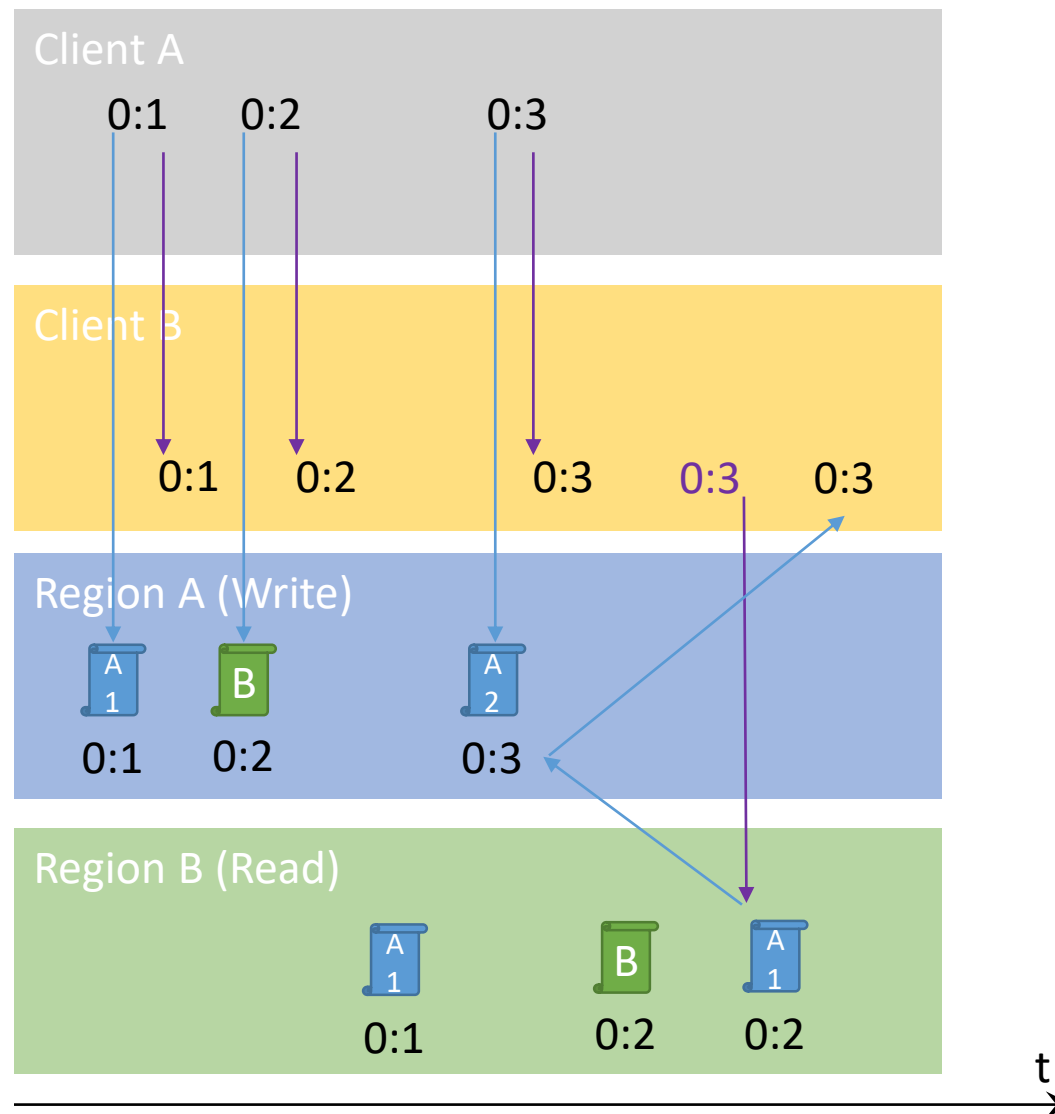
- Read own writes (within write region)
- Monotonic read
- Consistent prefix



# Independent Clients multiple regions



Clients sharing session token in multiple regions



# Bounded Staleness

- Read own writes (within write region)
- Monotonic read (within a region)
- Consistent prefix
- Staleness Bound  $< K, T$

Single region

$$K_{\min} = 10 \text{ op}$$

$$T_{\min} = 5\text{s}$$

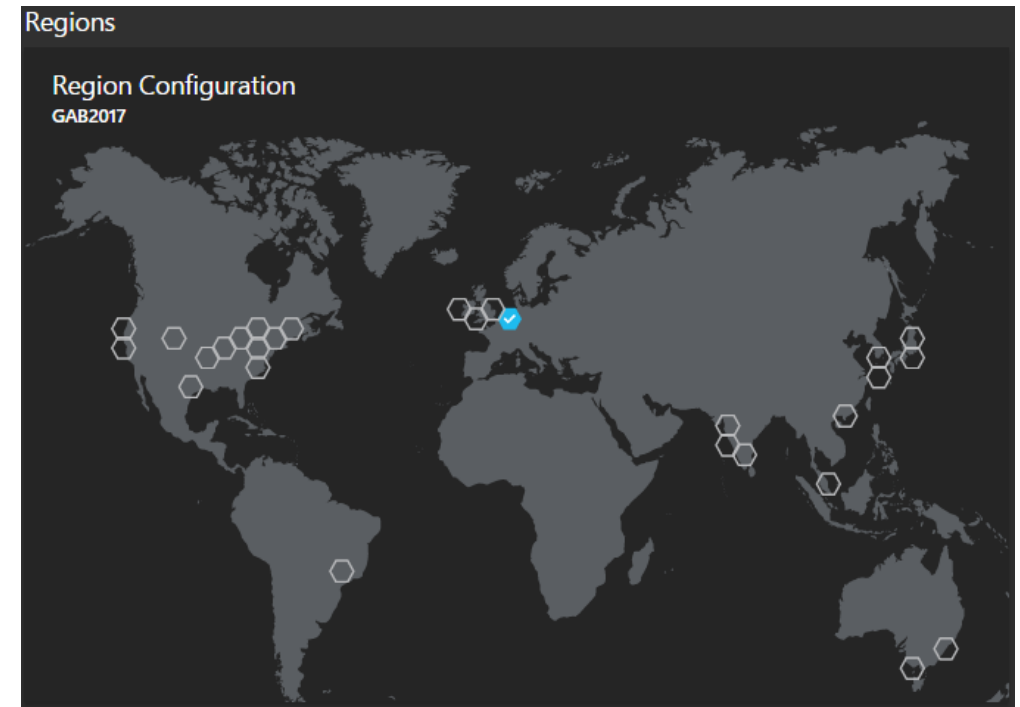
Multiple regions

$$K_{\min} = 100\text{K op}$$

$$T_{\min} = 5 \text{ min}$$

# Global Distribution

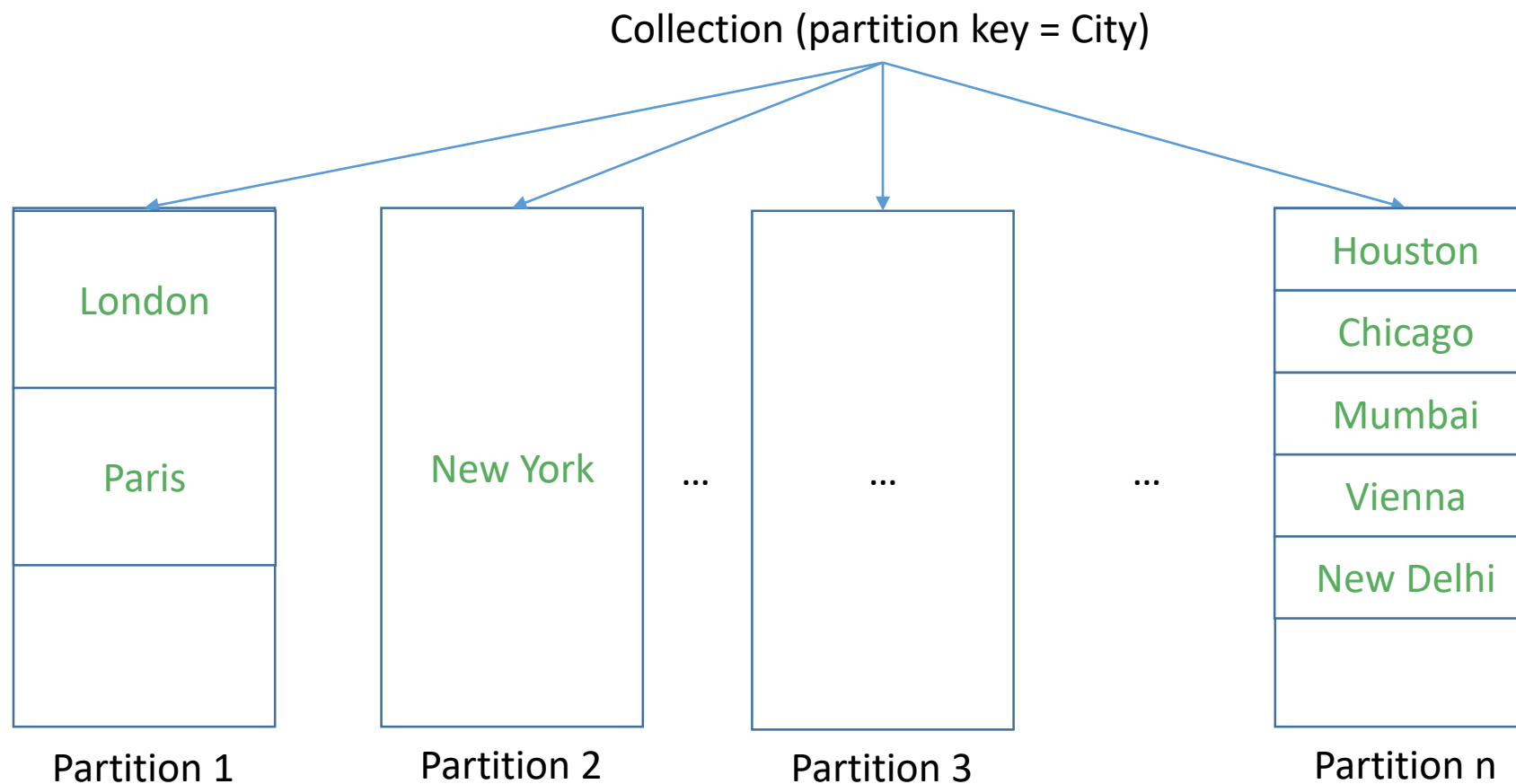
- Ring 0 - Service
- Automatic multi-region replication
- Multi-homing API's
- Manual and automatic failover



# DEMO

Global Distribution

# Partitioning



# Finding the right partition key

Scenario based

Wide range of values

Define as Json Path

Always indexed

**ALWAYS PROVIDE PARTITION KEY FOR PERFORMANCE**

/City  
/City/District  
/DeviceID

SLA - <https://aka.ms/acdbsla>

Operational

Availability (99.99%)

Throughput (99.99%)

Consistency (99.99%)

Latency (99.99 %, <=1KB, 99/100, <=10ms read, <=15ms write)

OPERATION	MAXIMUM UPPER BOUND ON PROCESSING LATENCY
All Database Account configuration operations	2 Minutes
Add a new Region	60 Minutes
Manual Failover	5 Minutes
Resource Operations	5 Sec
Media Operations	60 Sec

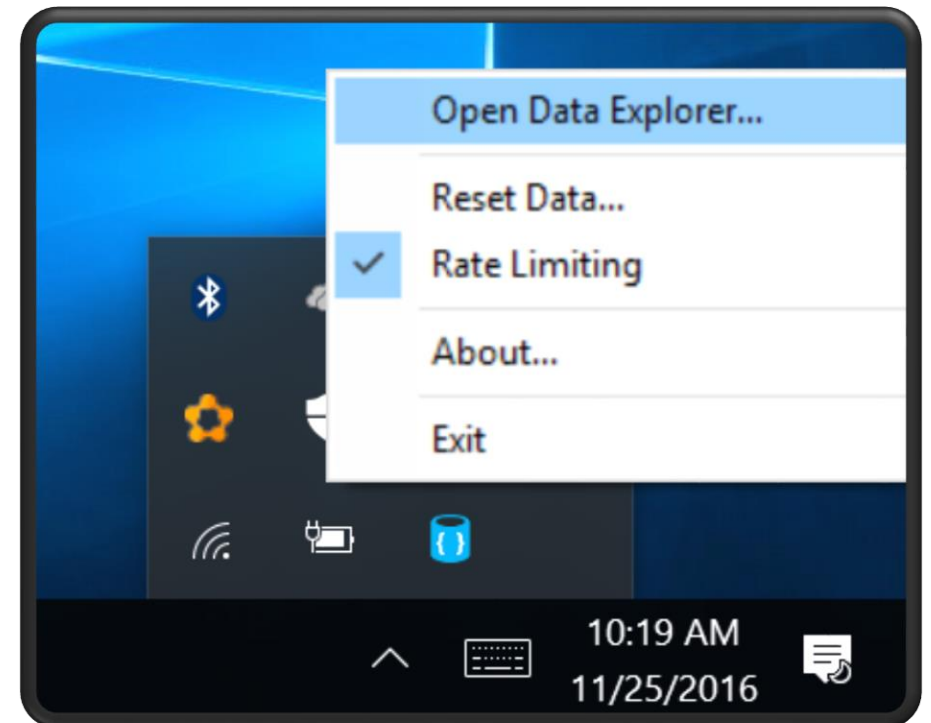


# CosmosDB Emulator

MSI or Docker for Windows Container

Only DocumentDB API

More an incomplete rebranding effort...



# Resources

[Overview Cosmos DB \(Documentation\)](#)

[Cosmos DB Session with Syam Nair \(Video\)](#)

[Azure Cosmos DB Build 2017 Talk Aravind Krishna R. \(Video\)](#)

[Modeling data for NoSQL Databases](#)

[Compare Chart \(CosmosDB / AWS DynamoDB / Google Cloud Spanner\)](#)

[Comparison Google Cloud Spanner vs. Microsoft Azure Cosmos DB](#)

# Resources

[Introduction to Graph Computing](#)

[CosmosDB Graph API Feedback](#)

[Gremlin Console - The first five minutes Tutorial](#)

[Traversal Recipies](#)

[.NET Sample - Gettings started with Gremlin \(GitHub\)](#)

[.NET Sample - More complex ASP.NET MVC SPA application with visualization](#)



**THANK YOU**

Specto  
logic<sup>®</sup>

---