Exam AZ-203

Developing Solutions for Microsoft Azure

Agenda

- Storage Account
- Blob Storage
- Content Delivery Network
- Table Storage

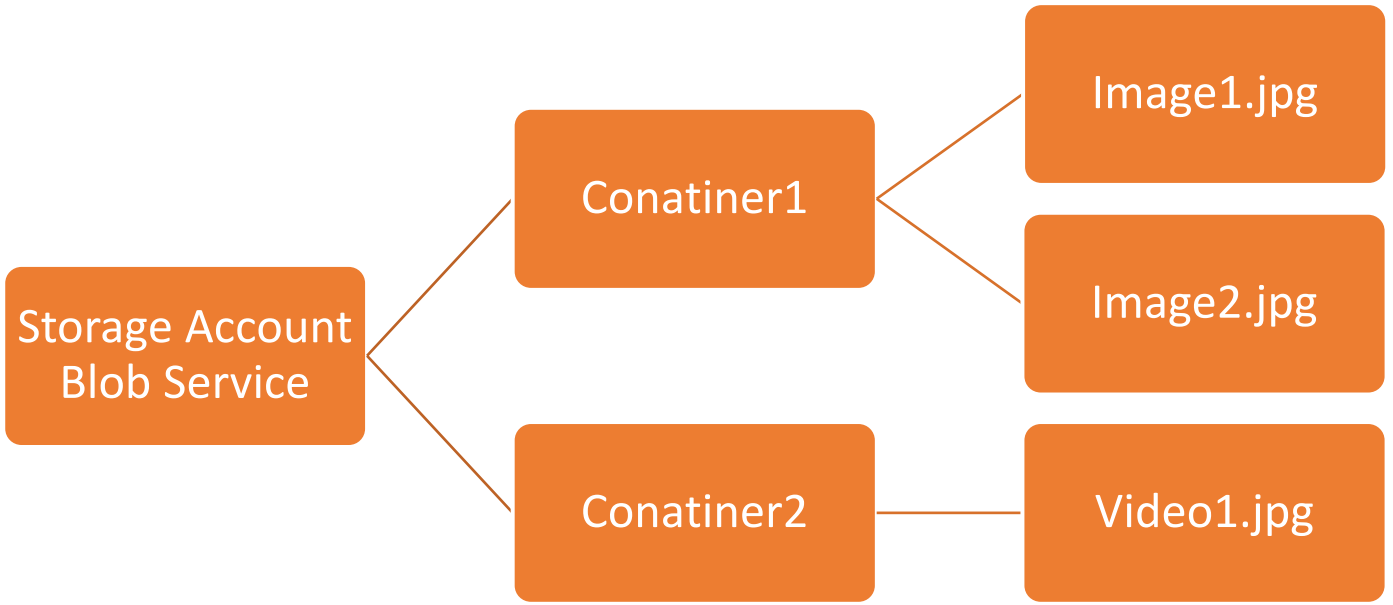
Blob Types

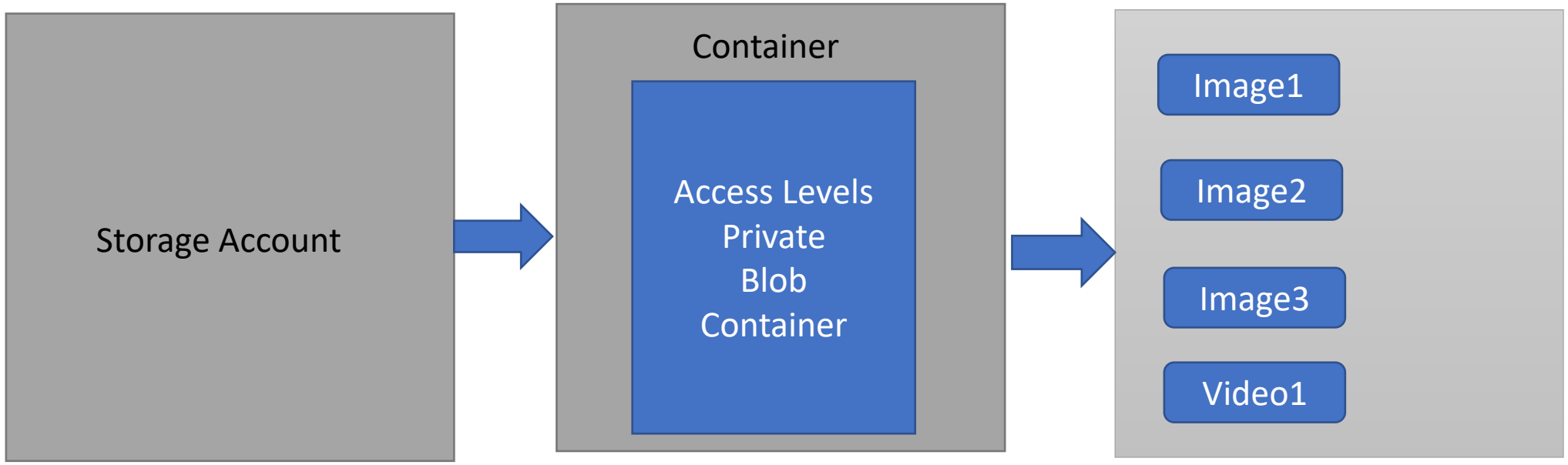
Block Blob

Append Blob

Page Blob

Blob Structure





http://mystorageaccount.blob.core.windows.net/container/Image1.jpg

http://mystorageaccount.blob.core.windows.net/container?comp=list

Access Levels

Private

Authenticated request

Blob

Anonymous read access for blobs but cannot list all the blobs inside container.

Container

All container and blob data can be read by anonymous request. Clients can enumerate blobs within the container by anonymous request but cannot enumerate containers within the storage account.

Security

Account key

SAAS token

Features

Handles concurrency

Soft Delete feature

Create Snapshots

Demo

- Create a container and Blobs
- Showcase access level
- Soft Delete
- .Net SDK

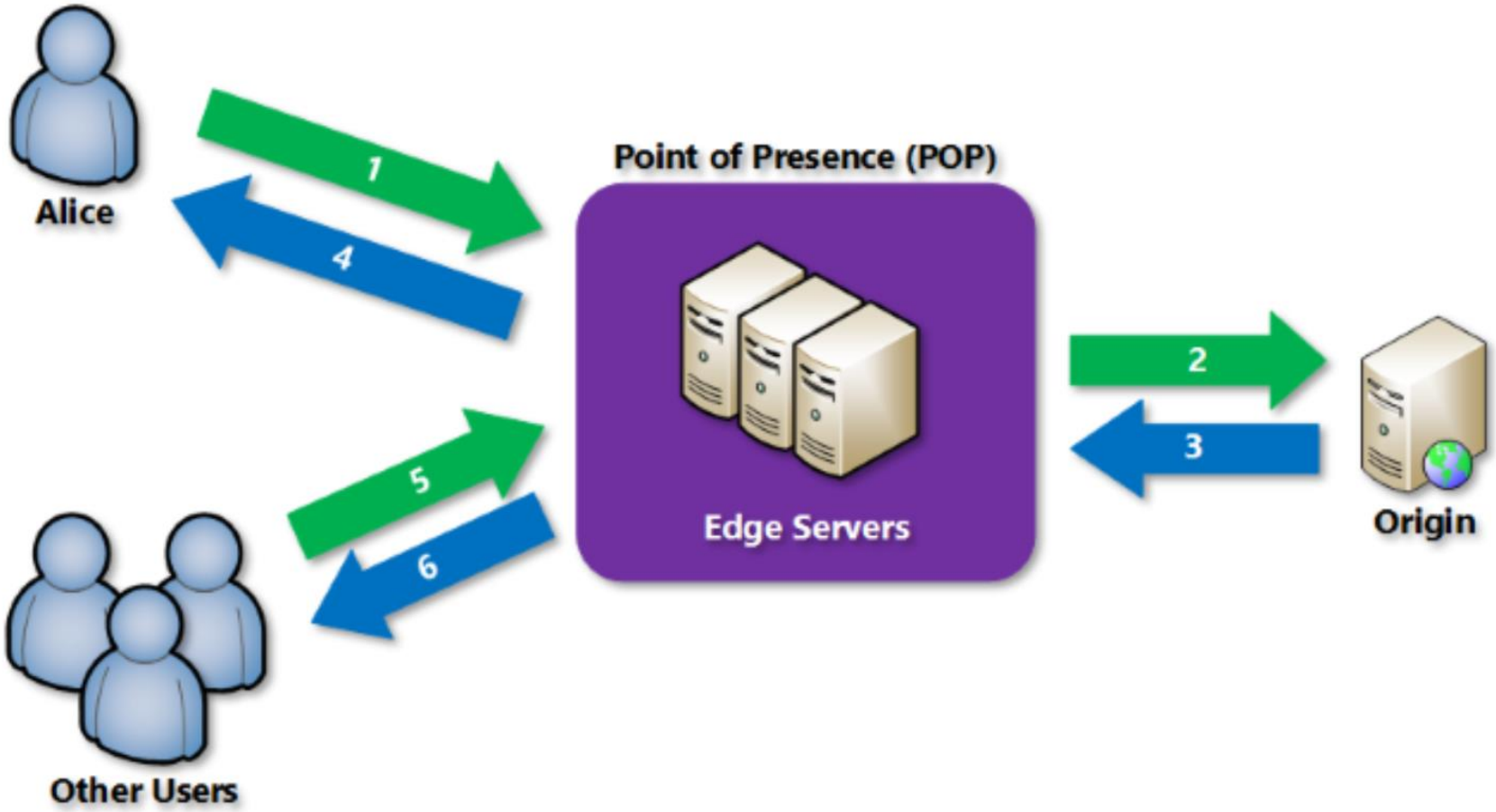
Concurrency

1. **Optimistic concurrency** – An application performing an update will as part of its update verify if the data has changed since the application last read that data. For example, if two users viewing a wiki page make an update to the same page then the wiki platform must ensure that the second update does not overwrite the first update – and that both users understand whether their update was successful or not. This strategy is most often used in web applications.

2. Pessimistic concurrency – An application looking to perform an update will take a lock on an object preventing other users from updating the data until the lock is released. For example, in a master/slave data replication scenario where only the master will perform updates the master will typically hold an exclusive lock for an extended period of time on the data to ensure no one else can update it.

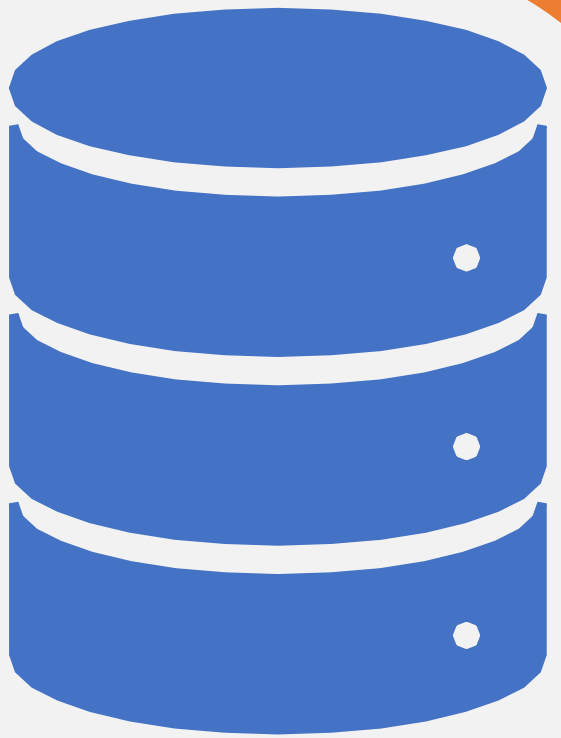
3. Last writer wins – An approach that allows any update operations to proceed without verifying if any other application has updated the data since the application first read the data. This strategy (or lack of a formal strategy) is usually used where data is partitioned in such a way that there is no likelihood that multiple users will access the same data

Azure CDN Overview



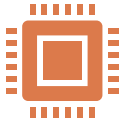
Cache Rules

- **Ignore query strings**: Default mode. In this mode, the CDN point-of-presence (POP) node passes the query strings from the requestor to the origin server on the first request and caches the asset. All subsequent requests for the asset that are served from the POP ignore the query strings until the cached asset expires.
- **Bypass caching for query strings**: In this mode, requests with query strings are not cached at the CDN POP node. The POP node retrieves the asset directly from the origin server and passes it to the requestor with each request.
- **Cache every unique URL**: In this mode, each request with a unique URL, including the query string, is treated as a unique asset with its own cache. For example, the response from the origin server for a request for example.ashx?q=test1 is cached at the POP node and returned for subsequent caches with the same query string. A request for example.ashx?q=test2 is cached as a separate asset with its own time-to-live setting.



Azure Table Storage

NoSQL



Uses JSON to serialize the data

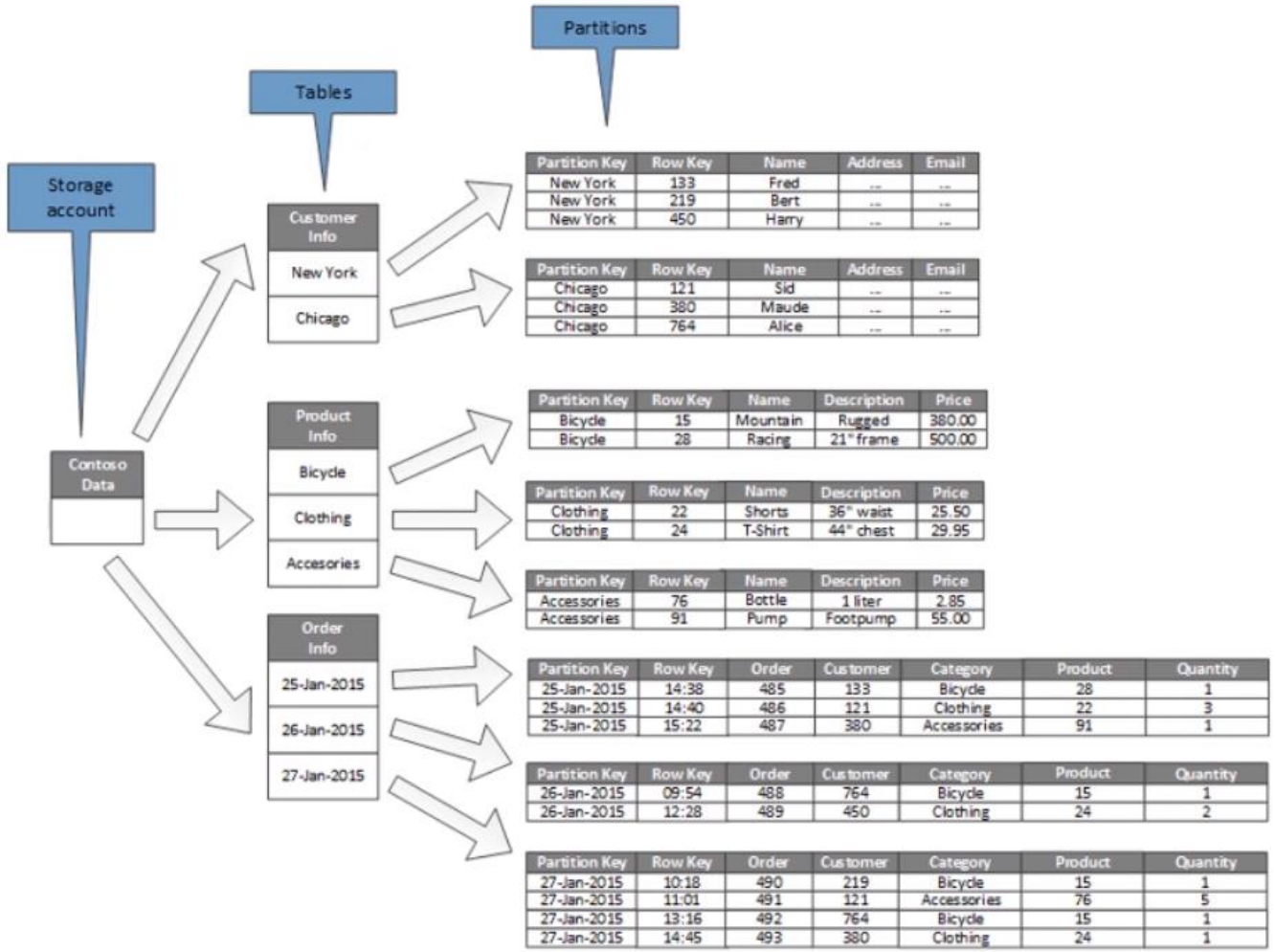
Flexible schema changes

Scalable and highly available

Single region with optional secondary

read-only region

Partitioning schemes



Partition Key	Row Key	Name	Address	Email
New York	133	Fred	...	...
New York	219	Bert	...	...
New York	450	Harry	...	...

Partition Key	Row Key	Name	Address	Email
Chicago	121	Sid	...	...
Chicago	380	Maude	...	...
Chicago	764	Alice	...	...

Partition Key	Row Key	Name	Description	Price
Bicycle	15	Mountain	Rugged	380.00
Bicycle	28	Racing	21" frame	500.00

Partition Key	Row Key	Name	Description	Price
Clothing	22	Shorts	36" waist	25.50
Clothing	24	T-Shirt	44" chest	29.95

Partition Key	Row Key	Name	Description	Price
Accessories	76	Bottle	1 liter	2.85
Accessories	91	Pump	Footpump	55.00

Partition Key	Row Key	Order	Customer	Category	Product	Quantity
25-Jan-2015	14:38	485	133	Bicycle	28	1
25-Jan-2015	14:40	486	121	Clothing	22	3
25-Jan-2015	15:22	487	380	Accessories	91	1

Partition Key	Row Key	Order	Customer	Category	Product	Quantity
26-Jan-2015	09:54	488	764	Bicycle	15	1
26-Jan-2015	12:28	489	450	Clothing	24	2

Partition Key	Row Key	Order	Customer	Category	Product	Quantity
27-Jan-2015	10:18	490	219	Bicycle	15	1
27-Jan-2015	11:01	491	121	Accessories	76	5
27-Jan-2015	13:16	492	764	Bicycle	15	1
27-Jan-2015	14:45	493	380	Clothing	24	1

Demo

- Table Storage Rest API