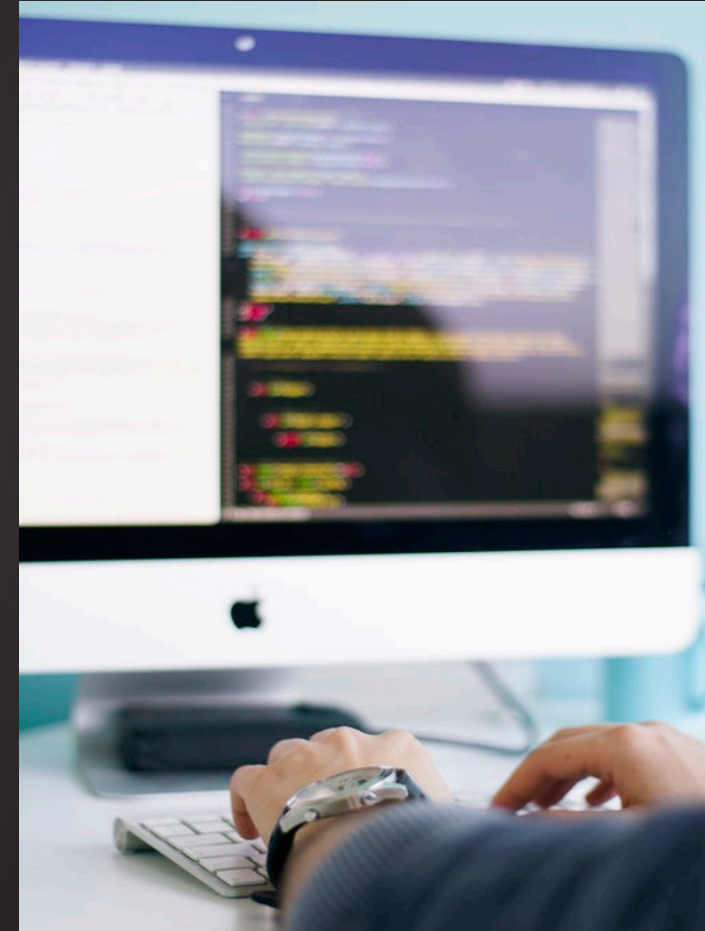


Session 03: Data Access Technologies in Azure

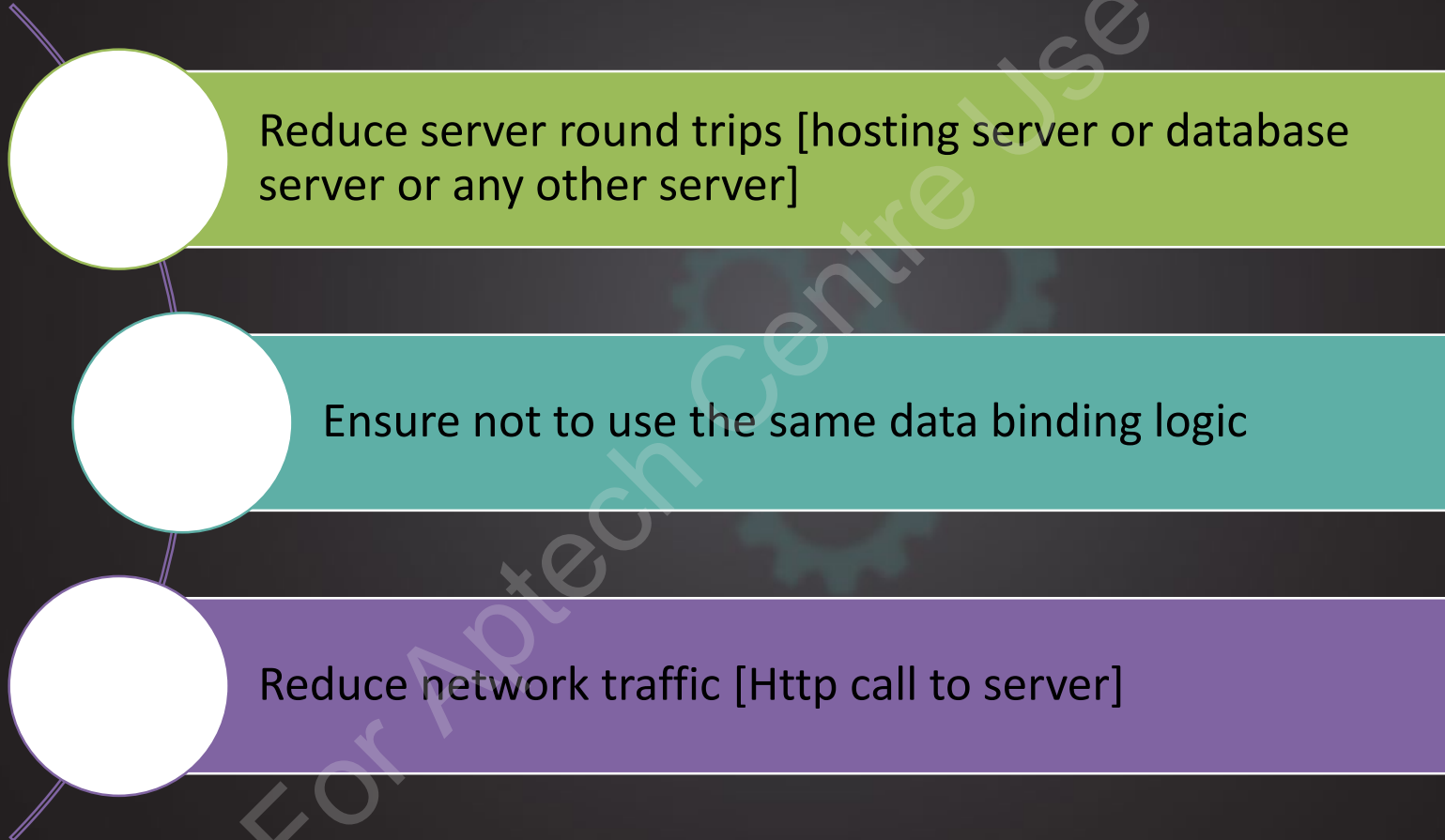
Objectives

- Explain the implementation of data caching services in Azure
- Describe how to use transactions
- Define distributed transactions in SQL Azure
- Describe data storage in Microsoft Azure
- Explain the role of Content Delivery Networks (CDNs)



Implementation of Data Caching Services

Caching means storing in memory. It helps in the following:



Cache Policies

An effective cache policy is determined by:

Client cache requirements

Server's content expiration requirements

Server's revalidation requirements

Including Policy Expirations

Two types of expiration policies available for cache expiration are:

AbsoluteExpiration

- Indicating whether sliding expiration is enabled.

Sliding Expiration

- Policy specifies when a cache entry should be evicted after a specified duration.

Using Cache Dependency

Following table displays the types of cache dependencies:

Type of Cache Dependency	Description
Cache dependency on cached items	Indicates dependence of cached items on other cached items. For example, removing item X from the cache allows removing item Y also from the cache.
Cache dependency on a file	Indicates cache being dependent on a file. Any update in the file contents leads to the expiration for the cache and refilling the content again.
Cache dependency on SQL	Indicates that until the time the table entries do not change, the data will be present in the cache. Any change to the content would expire the cache. SQL Cache dependency is enabled using the aspnet_regsql.exe command line tool.

Query Notifications in ADO.NET

ADO.NET helps to implement query notifications using following methods:

SqlDependency Class

- This class provides high-level implementation.
- In this, a high-level abstraction of notification functionality is provided between the source application and SQL Server.

SqlNotificationRequest Class

- This class provides low-level implementation.
- In this, the server-side functionality is exposed and a notification request with a command is executed.

SqlCacheDependency Helper Classes

- These classes are used by Web applications built using ASP.NET 2.0



Using Transactions

A transaction has the four standard properties, usually called by acronym ACID:

Atomicity	<ul style="list-style-type: none">Assures successful completion of work unit and the operations used. If the transaction fails then operations are rolled back to their previous state
Consistency	<ul style="list-style-type: none">Assures that when a transaction is successful, the state of the database changes
Isolation	<ul style="list-style-type: none">Assures independent and transparent transactions
Durability	<ul style="list-style-type: none">Assures that when a system fails, a committed transaction remains

Distributed Transactions

Some transactions include more than one data source. This is called a distributed or elastic transaction.

Implement distributed transactions:

Using Elastic Database Transactions with Azure SQL Database

Using System.Transactions Namespace

Transaction Isolation Level

- A locking mechanism isolates one transaction from another.
- Behavior of this locking policy depends on level of isolation set for each transaction.

Serializable	Repeatable Read	Read Committed	Read Uncommitted
<ul style="list-style-type: none">• This indicates the highest level of isolation.• During any read and write process, the data is locked exclusively.• To stop the creation of phantom rows, range locks are achieved.	<ul style="list-style-type: none">• This indicates the second highest level of isolation.• This is similar to serializable, the only difference being range locks are not achieved.	<ul style="list-style-type: none">• In this level, shared locks are allowed and reads only committed data.• It never reads any data that is being updated in between any transaction.	<ul style="list-style-type: none">• This lowest level of isolation allows dirty read.

Important Isolation Levels

Data Storage in Microsoft Azure

Behavior of Azure storage:

A scalable object store that is massive for data objects

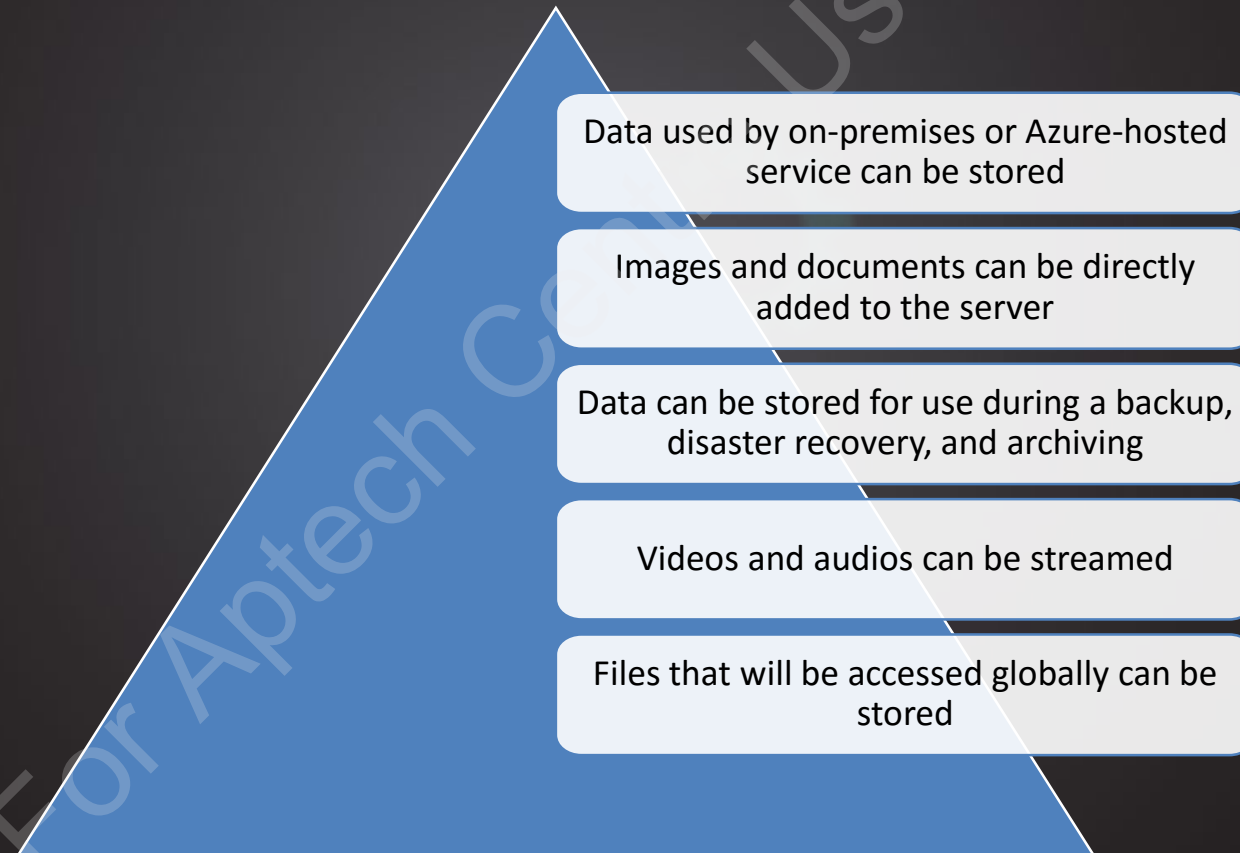
A file system service for the cloud

A messaging store for reliable messaging

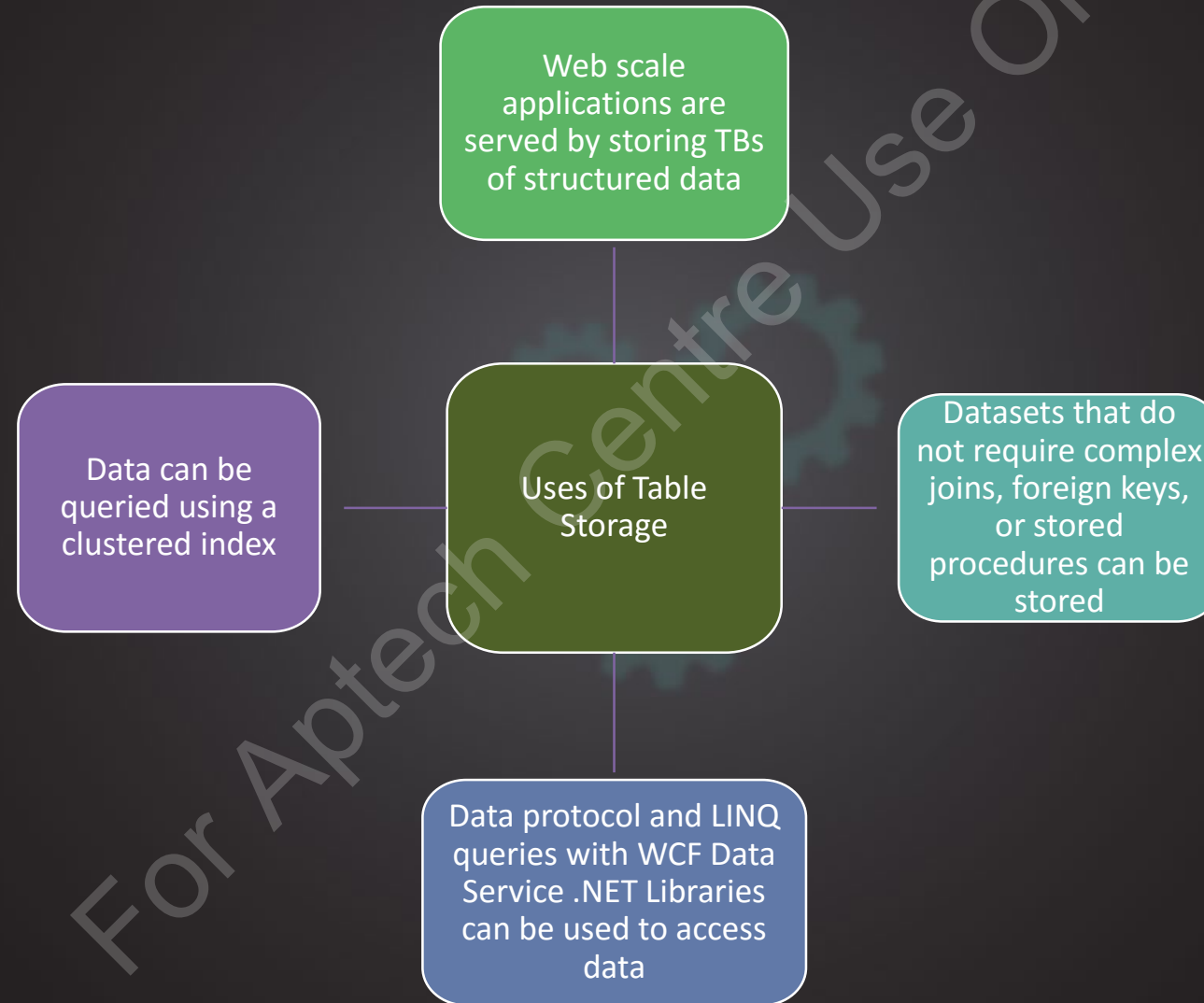
A NoSQL store

Microsoft Azure Blobs

Following are the uses of blob storage:



Microsoft Azure Tables



Microsoft Azure SQL



Features of Microsoft Azure SQL:

- Supports document, key value, relational, and graph models using one backend
- Supports ACID transactions
- Uses query language similar to SQL
- Does not rely on any schemas

Content Delivery Networks (CDNs)

Following are benefits of Azure CDN:

Massive scaling

- Helps in handling instantaneous high loads. For example, start of a product launch event.

Handling user request and content

- User requests are assigned and content is sent directly from edge servers. This prevents traffic from being formed at the origin server.

Good performance and easy to use

- Very useful for an application when various round-trips are required to add the content.

Summary

- Different caching techniques and policies can make applications respond faster.
- Using SQL Server for caching purpose improves the performance for data or information that are not changing very frequently.
- Transactions are important for data consistency and atomicity. Transactions can be achieved by using ADO.NET transactions, System.Transactions, and TransactionScope.
- There are many other data services provided by Azure such as Azure Cosmos DB, Azure Blob, and Azure Table. These help to solve various problems as per different requirements.
- Azure SQL database can be created and accessed from any location and facilitates a database as a service.