Microsoft

Develop solutions that use Microsoft Azure Blob storage

Kishore Chowdary

Microsoft

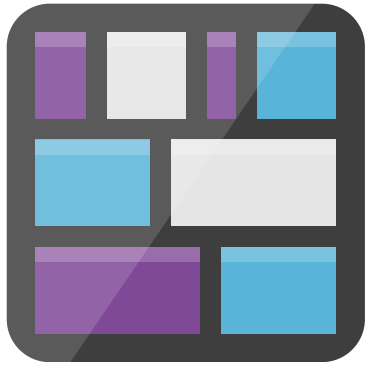
Topics

- Azure Blob storage
- Blob storage security
- Working with Azure Blob storage

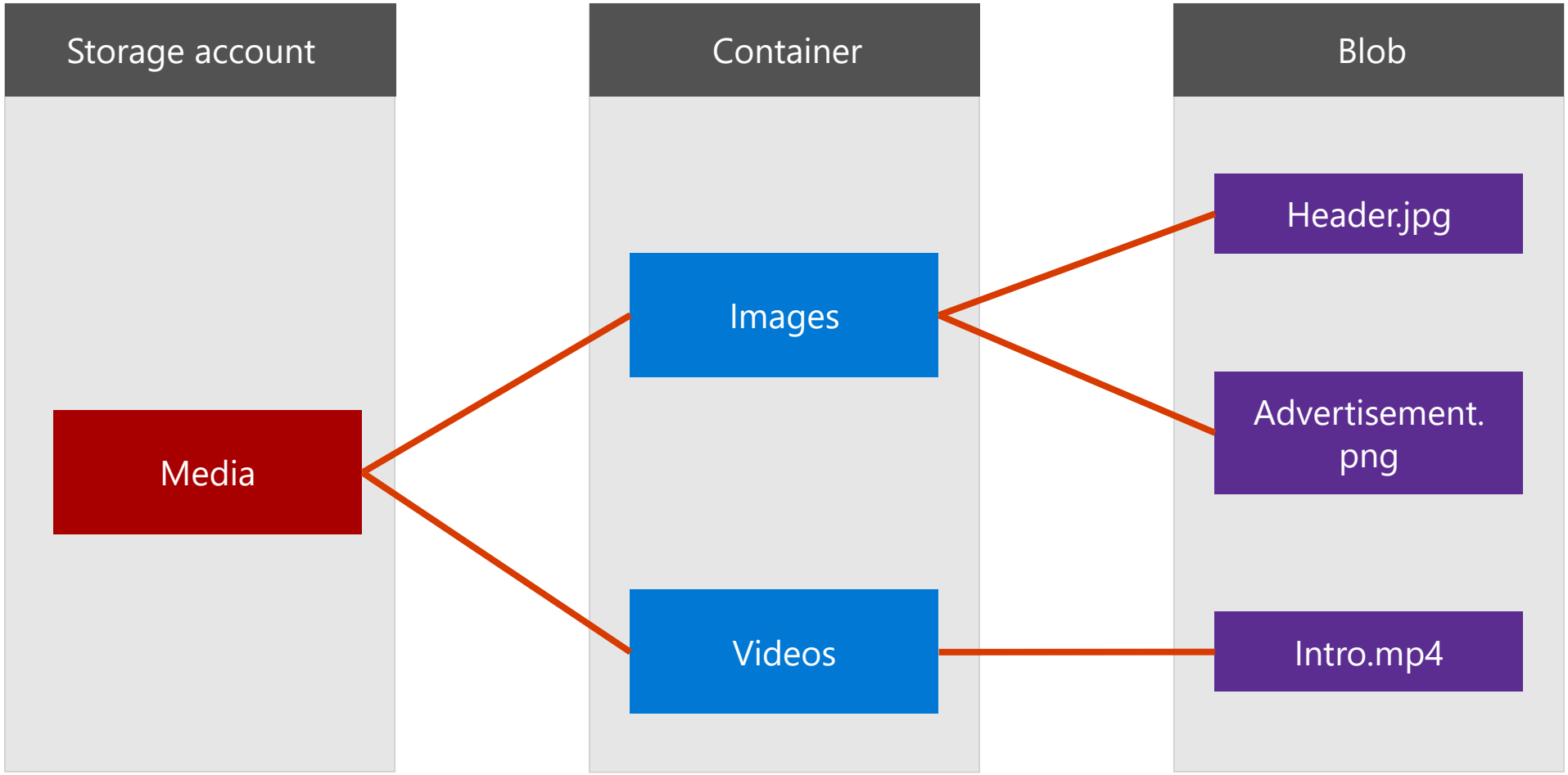
Lesson 01: Azure Blob storage

Azure Blob storage

- Object storage solution in the cloud
- Blob storage is designed for:
  - Serving images or documents directly to a browser
  - Storing files for distributed access
  - Streaming video and audio
  - Writing to log files
  - Storing data for backup and restore, disaster recovery, and archiving
  - Storing data for analysis by an on-premises or Azure-hosted service
- Accessible via a HTTP/HTTPS API



Azure Blob storage resource hierarchy



Storage tiers

- Storage tiers allow you to tune performance and cost to a ratio that is ideal for your solution
- There are three tiers:

Hot

- The most common tier
- Used for data that is accessed frequently

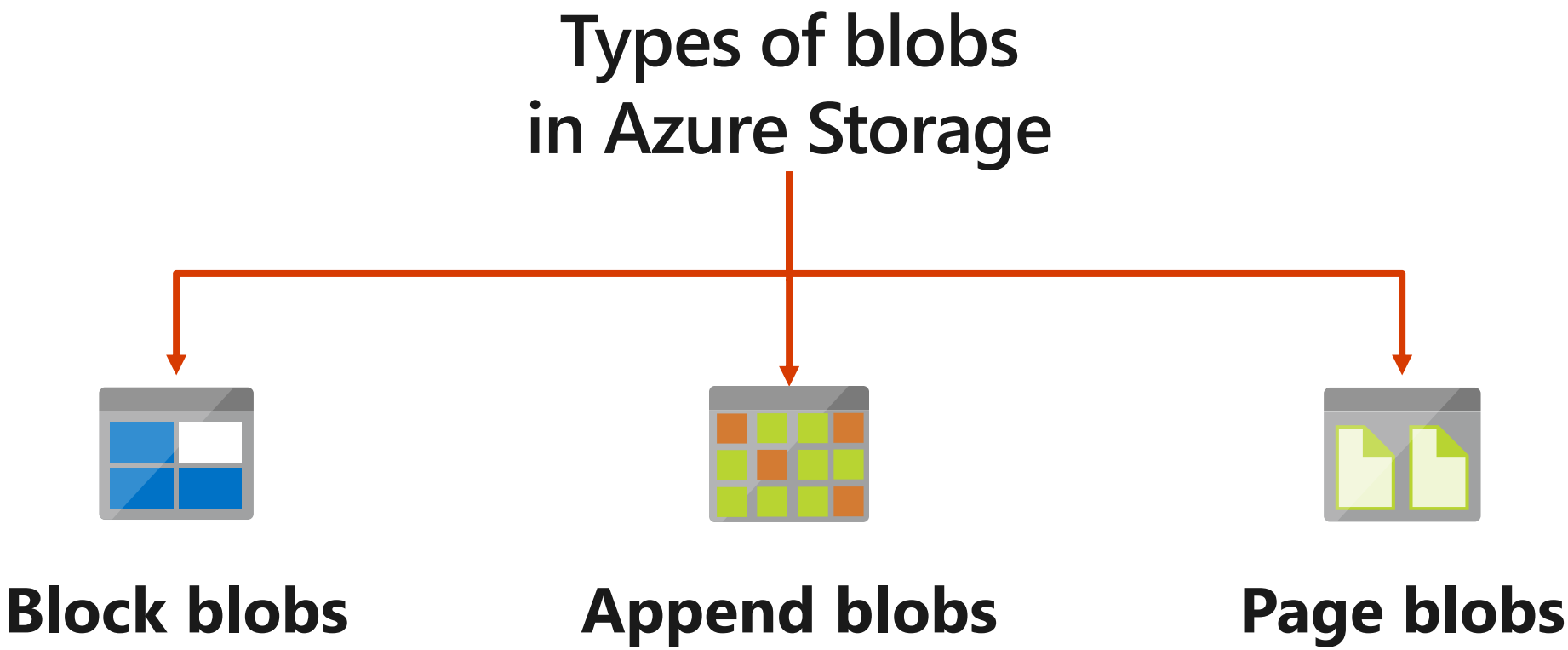
Cool

- Used for data that is infrequently accessed
- Ideal for data that is stored at least 30 days

Archive

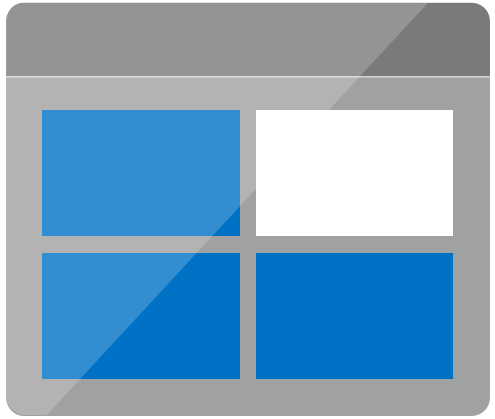
- Used for data that must be retained but is rarely accessed
- Data stored at this tier must be flexible for hours-based latency
- Ideal for data that is stored at least 180 days

Blob types


Types of blobs
in Azure Storage
Block blobs
Append blobs
Page blobs

Block blobs

- Comprise blocks of data
- Ideal for data that is stored in blocks—up to 100-MB chunks
- Simultaneous upload of large blobs with a single write operation
- A single block blob can include up to 50,000 blocks



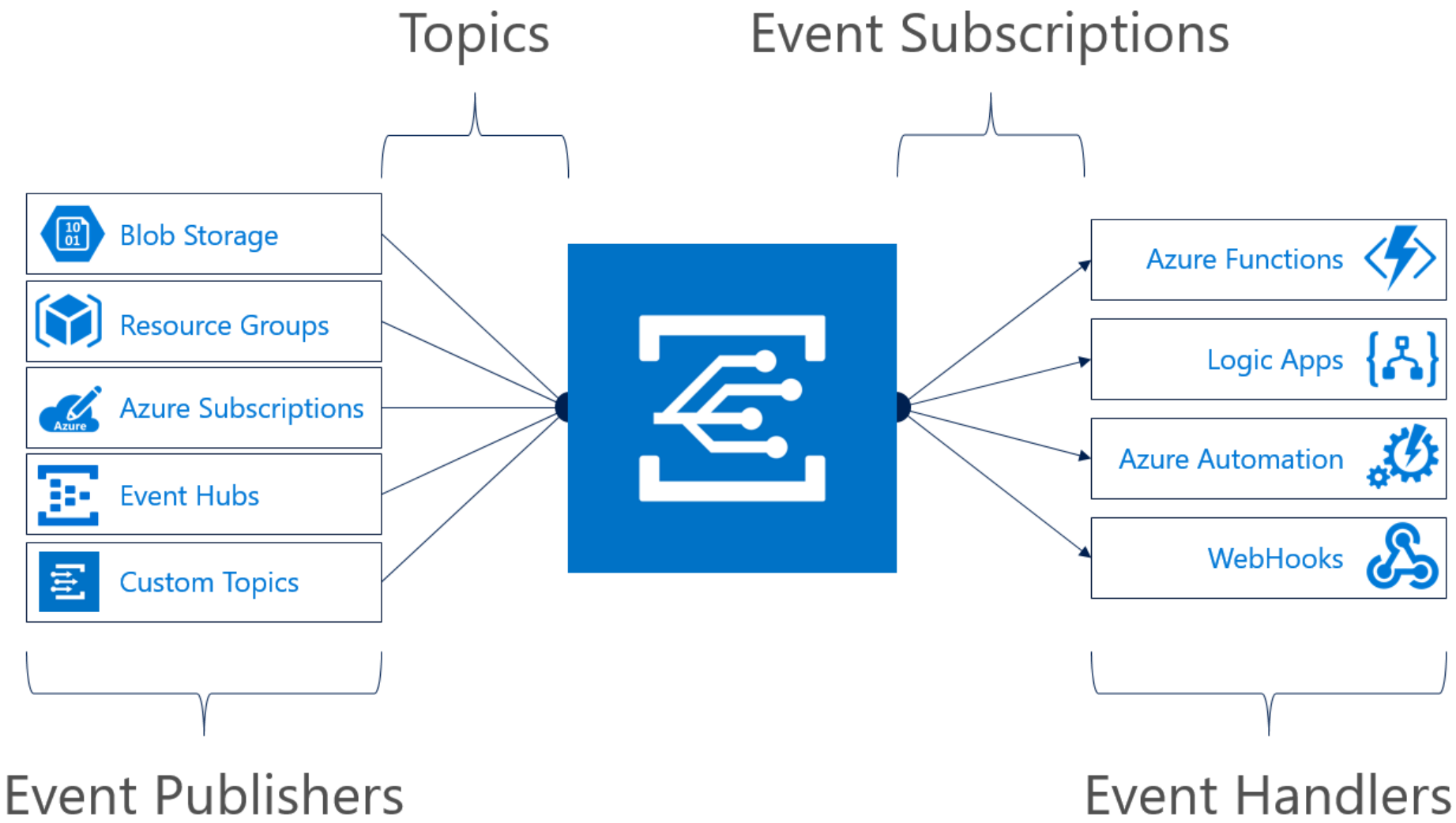
Append blobs

- Append blobs include the following characteristics:
  - They are composed of blocks
  - They are optimized for append operations
  - They are ideal for performant logging

Page blobs

- Composed of 512-byte pages
- Similar to hard disk storage
- Ideal for virtual hard disks
- Pages created by initializing the page blob and specifying the size
- Content to be added within 512-byte page boundaries
- Writes to page blobs commit immediately

Blob events


Topics
Event Subscriptions
Blob Storage
Resource Groups
Azure Subscriptions
Event Hubs
Custom Topics
Azure Functions
Logic Apps
Azure Automation
WebHooks
Event Publishers
Event Handlers

Lesson 02: Blob storage security

Shared Access Signatures (SASs)

https://myaccount.blob.core.windows.net/sascontainer/sasblob.txt?sv=2012-02-12&st=2013-04-29T22%3A18%3A26Z&se=2013-04-30T02%3A23%3A26Z&sr=b&sp=rw&sig=Z%2FRHIX5Xcg0Mq2rqI3OlWTjEg2tYkboXr1P9ZUXDtkk%3

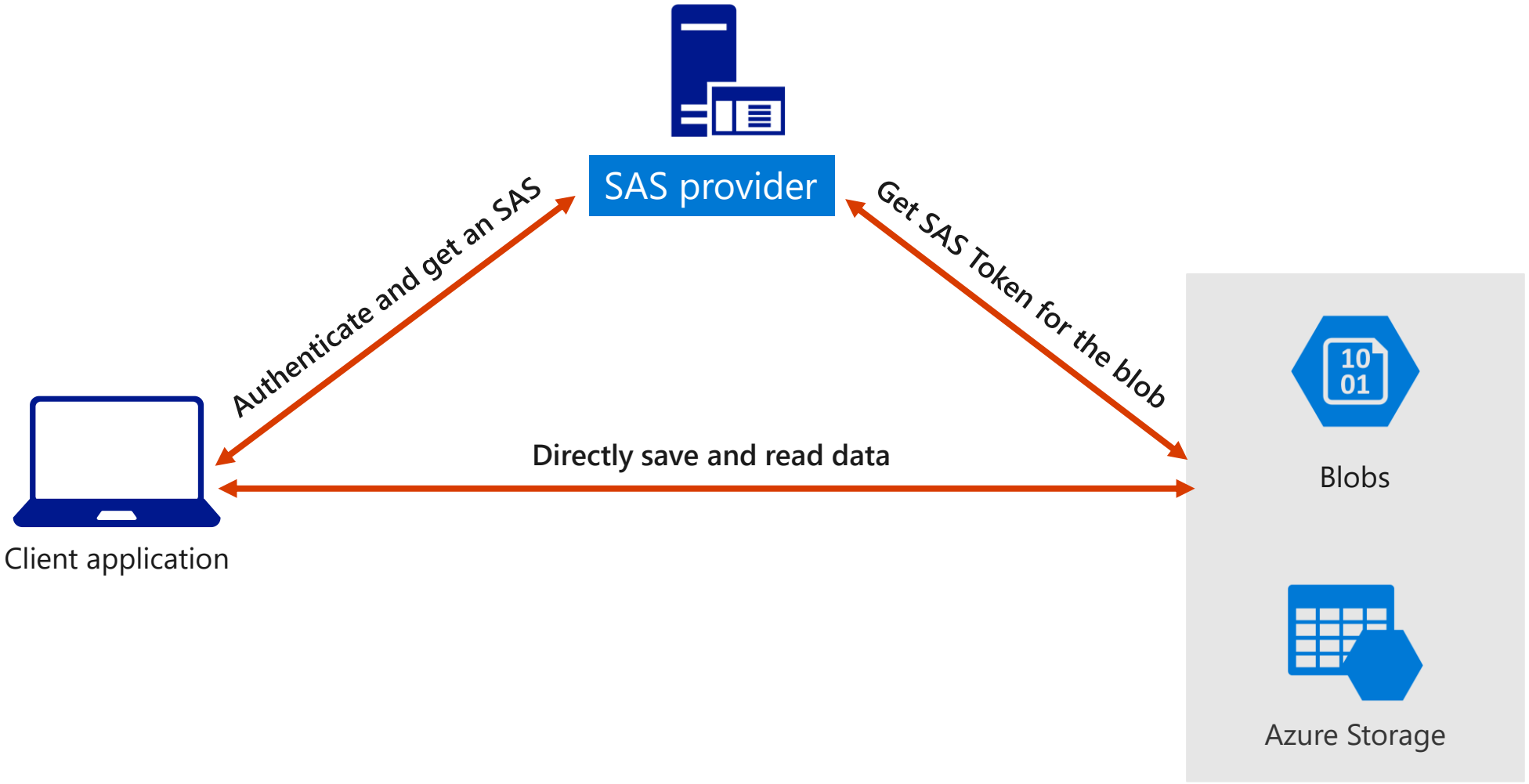
Component	Content	Description
Blob URI	https://myaccount.blob.core.windows.net/sascontainer/sasblob.txt	The address of the blob. Note that using HTTPS is highly recommended.
Storage services version	sv=2012-02-12	For Azure Storage services version 2012-02-12 and later, this parameter indicates the version to use.
Start time	st=2013-04-29T22%3A18%3A26Z	Specified in an International Organization for Standardization (ISO) 8061 format. If you want the SAS to be valid immediately, omit the start time.
Expiration time	se=2013-04-30T02%3A23%3A26Z	Specified in an ISO 8061 format.

Shared Access Signatures (SASs) (continued)

https://myaccount.blob.core.windows.net/sascontainer/sasblob.txt?sv=2012-02-12&st=2013-04-29T22%3A18%3A26Z&se=2013-04-30T02%3A23%3A26Z&sr=b&sp=rw&sig=Z%2FRHIX5Xcg0Mq2rqI3OlWTjEg2tYkboXr1P9ZUXDtkk%3

Component	Content	Description
Resource	sr=b	The resource is a blob.
Permissions	sp=rw	The permissions granted by the SAS include Read (r) and Write (w).
Signature	sig=Z%2FRHIX5Xcg0Mq2rqI3OlWTjEg2tYkboXr1P9ZUXDtkk%3D	Used to authenticate access to the blob. The signature is a Hash-based Message Authentication Code (HMAC) function computed over a string to sign and a key by using the SHA256 algorithm and then encoded by using Base64 encoding.

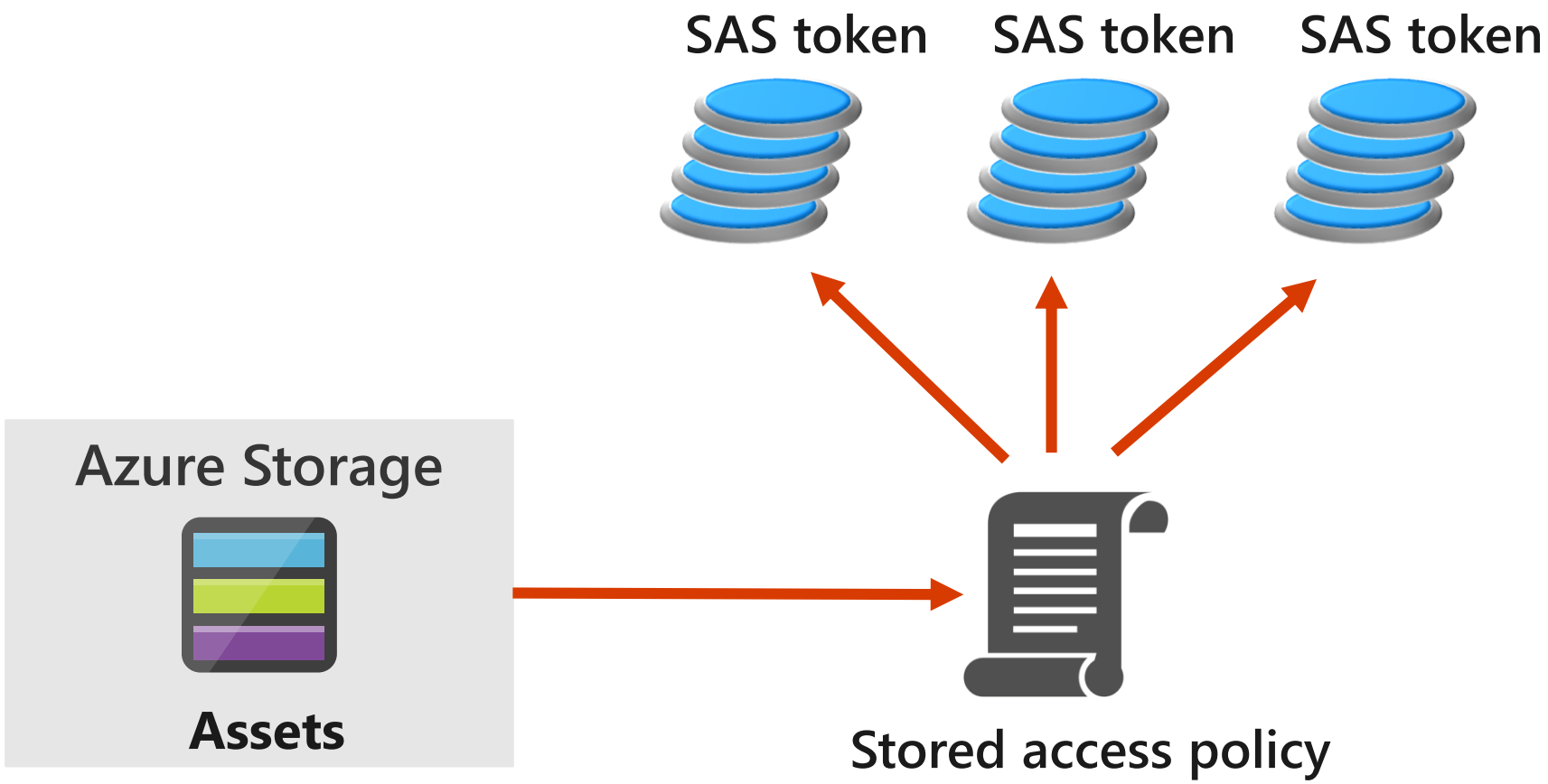
Valet key pattern by using Shared Access Signatures



Stored access policies

- A Shared Access Signature can take one of two forms:
  - Ad-hoc
    - When you create an ad hoc SAS, the start time, expiration time, and permissions for the SAS are all specified on the SAS URI (or implied in the case where the start time is omitted)
  - SAS generated from stored access policy
    - A stored access policy is defined on a resource container and can be used to manage constraints for one or more Shared Access Signatures
    - When you associate a SAS with a stored access policy, the SAS inherits the constraints defined for the stored access policy
- Anyone who obtains the SAS can use it

SAS token generation from a stored access policy



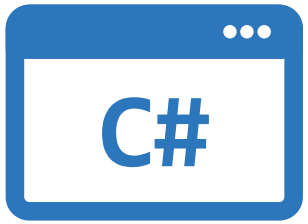
Lesson 03: Working with Azure Blob storage

Blob container properties

Property	Description
ETag	This is a standard HTTP header that gives a value that is unchanged unless a property of the container is changed. This value can be used to implement optimistic concurrency with the blob containers.
LastModified	This property indicates when the container was last modified.
PublicAccess	This property indicates the level of public access that is allowed on the container. Valid values include Blob, Container, Off, and Unknown.
HasImmutabilityPolicy	This property indicates whether the container has an immutability policy. An immutability policy will help ensure that blobs are stored for a minimum amount of retention time.
HasLegalHold	This property indicates whether the container has an active legal hold. A legal hold will help ensure that blobs remain unchanged until the hold is removed.

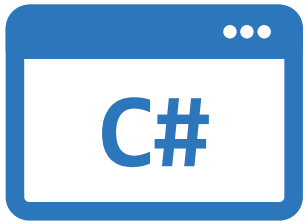
Manipulating blob container properties in .NET

```
CloudBlobClient client = storageAccount.CreateCloudBlobClient();

CloudBlobContainer container = client.GetContainerReference("images");

container.CreateIfNotExists();

await container.FetchAttributesAsync();

container.Properties.*
```

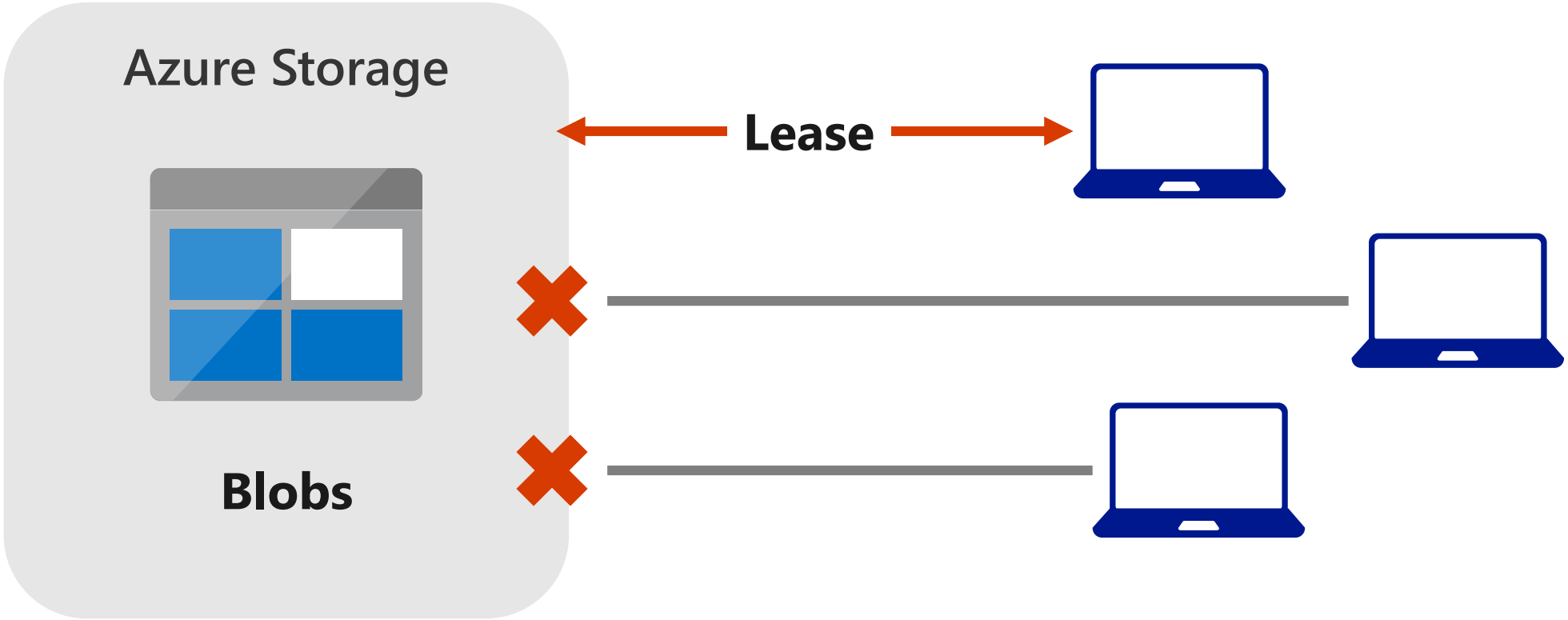


Manipulating blob container metadata in .NET

```
CloudBlobClient client = storageAccount.CreateCloudBlobClient();

CloudBlobContainer container = client.GetContainerReference("images");

container.CreateIfNotExists();

container.Metadata.Add("docType", "textDocuments");
container.Metadata["category"] = "guidance";

await container.SetMetadataAsync();
```



Exclusive access for modifying a blob



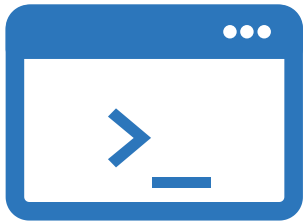
Lease Blob operation

- Establishes a lock on a blob for write and delete
  - Duration is typically 15 to 60 seconds
  - Optionally, you can establish an infinite lock
- Operation has five modes
  - Acquire
  - Renew
  - Change
  - Release
  - Break (end the lease but prevent other clients from acquiring a new lease)

AzCopy

Basic syntax:

```
AzCopy /Source:<source> /Dest:<destination> [Options]
```



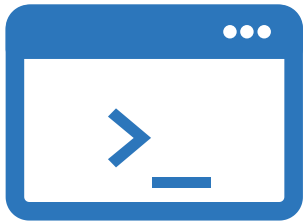
AzCopy – downloading blobs

Download blobs matching a specific pattern:

```
AzCopy
/Source:https://myaccount.blob.core.windows.net/container
/Dest:C:\myfolder /SourceKey:key /Pattern:"abc.txt"
```

Download all blobs in container:

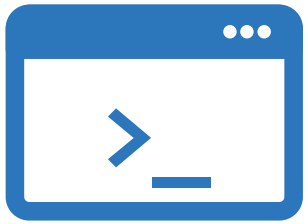
```
AzCopy
/Source:https://myaccount.blob.core.windows.net/container
/Dest:C:\myfolder /SourceKey:key /S
```



AzCopy – downloading blobs (continued)

Pattern match on a* prefix

```
AzCopy
/Source:https://myaccount.blob.core.windows.net/container
/Dest:C:\myfolder /SourceKey:key /Pattern:a /S
```



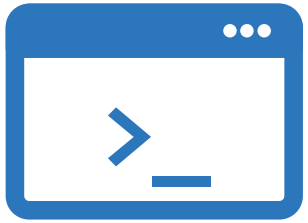
AzCopy – copying blobs between containers

Same account:

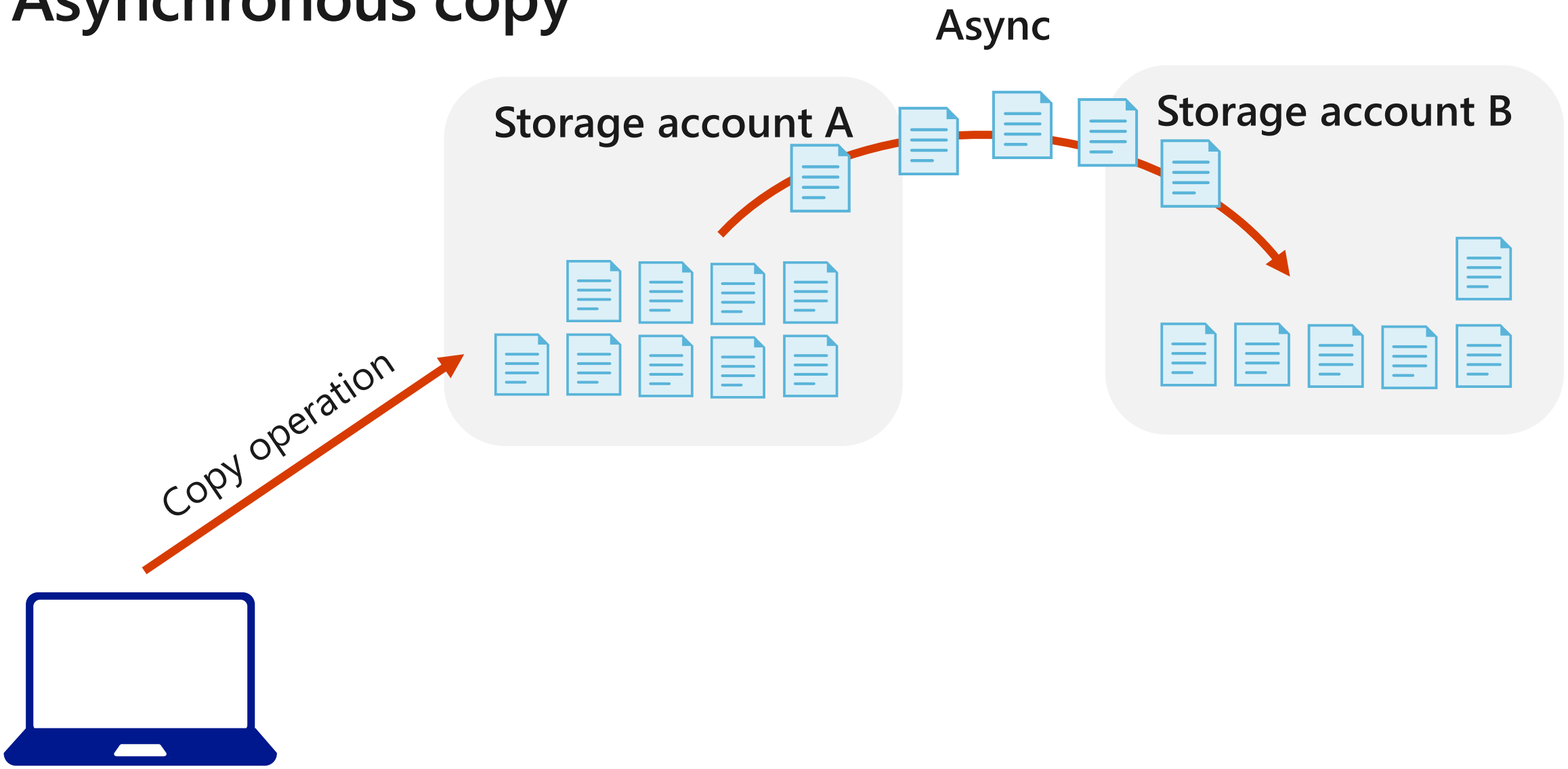
```
AzCopy
/Source:https://myaccount.blob.core.windows.net/mycontainer
/Dest:https://myaccount.blob.core.windows.net/theircontainer
/SourceKey:key /DestKey:key /Pattern:abc.txt
```

Different account:

```
AzCopy
/Source:https://myaccount.blob.core.windows.net/containera
/Dest:https://destaccount.blob.core.windows.net/containerb
/SourceKey:key1 /DestKey:key2 /Pattern:abc.txt
```



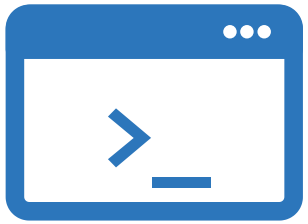
Asynchronous copy



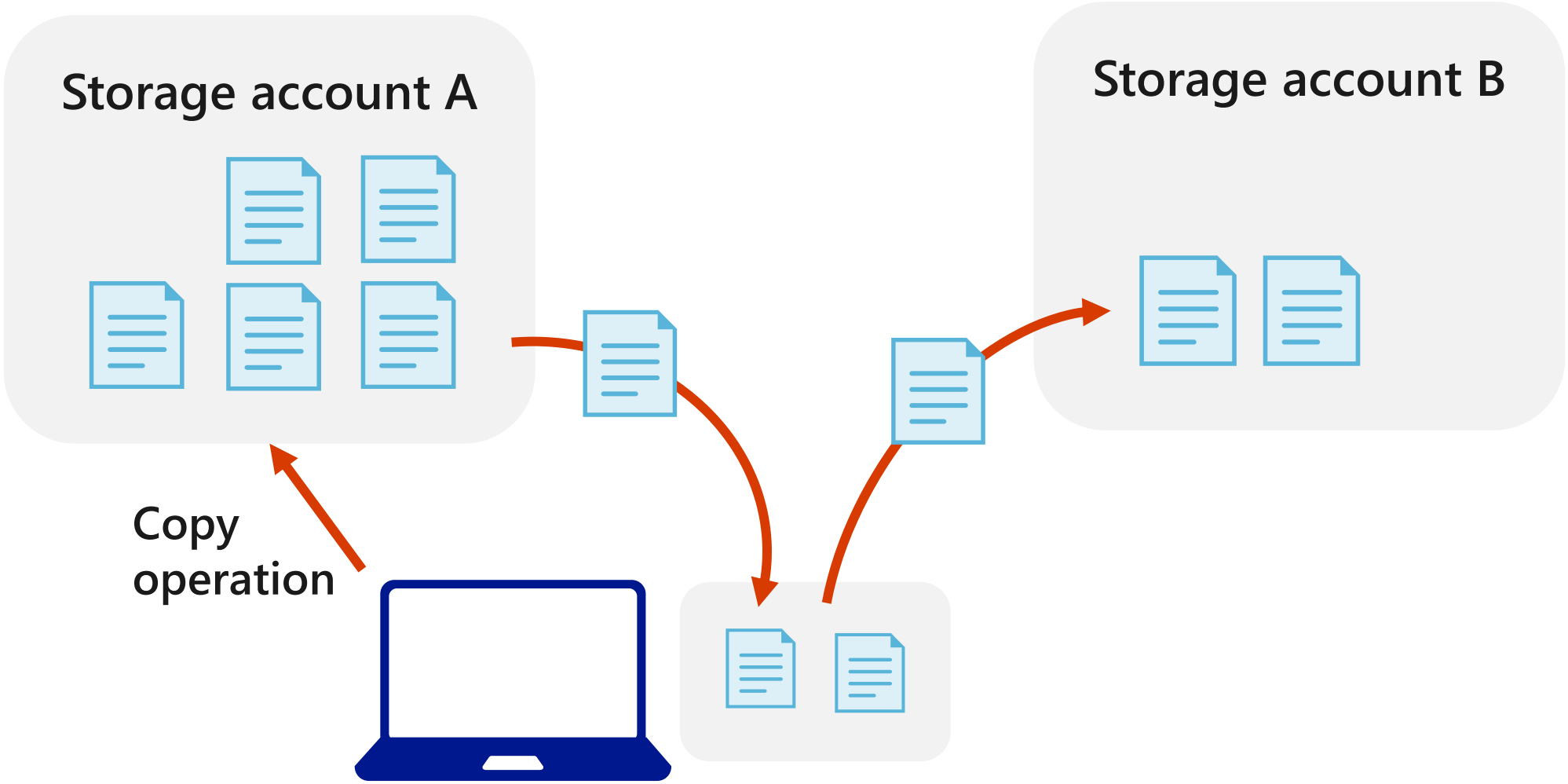
AzCopy – copying blobs between containers (continued)

Consistent speed by using /SyncCopy option:

```
AzCopy
/Source:https://myaccount.blob.core.windows.net/sourceContainer/
/Dest:https://myaccount2.blob.core.windows.net/destContainer/
/SourceKey:key1 /DestKey:key2 /Pattern:ab /SyncCopy
```



Synchronous copy



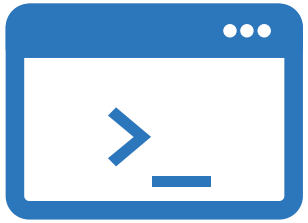
AzCopy – copying blobs to Azure Files shares

Azure Files share to blob:

```
AzCopy
/Source:https://myaccount1.file.core.windows.net/myfileshare/
/Dest:https://myaccount2.blob.core.windows.net/mycontainer/
/SourceKey:key1 /DestKey:key2 /S
```

Blob to Azure Files share:

```
AzCopy
/Source:https://myaccount1.blob.core.windows.net/mycontainer/
/Dest:https://myaccount2.file.core.windows.net/myfileshare/
/SourceKey:key1 /DestKey:key2 /S
```



Demo: Managing Azure Blob storage by using .NET

Microsoft

Review

- Azure Blob storage
- Blob storage security
- Working with Azure Blob storage