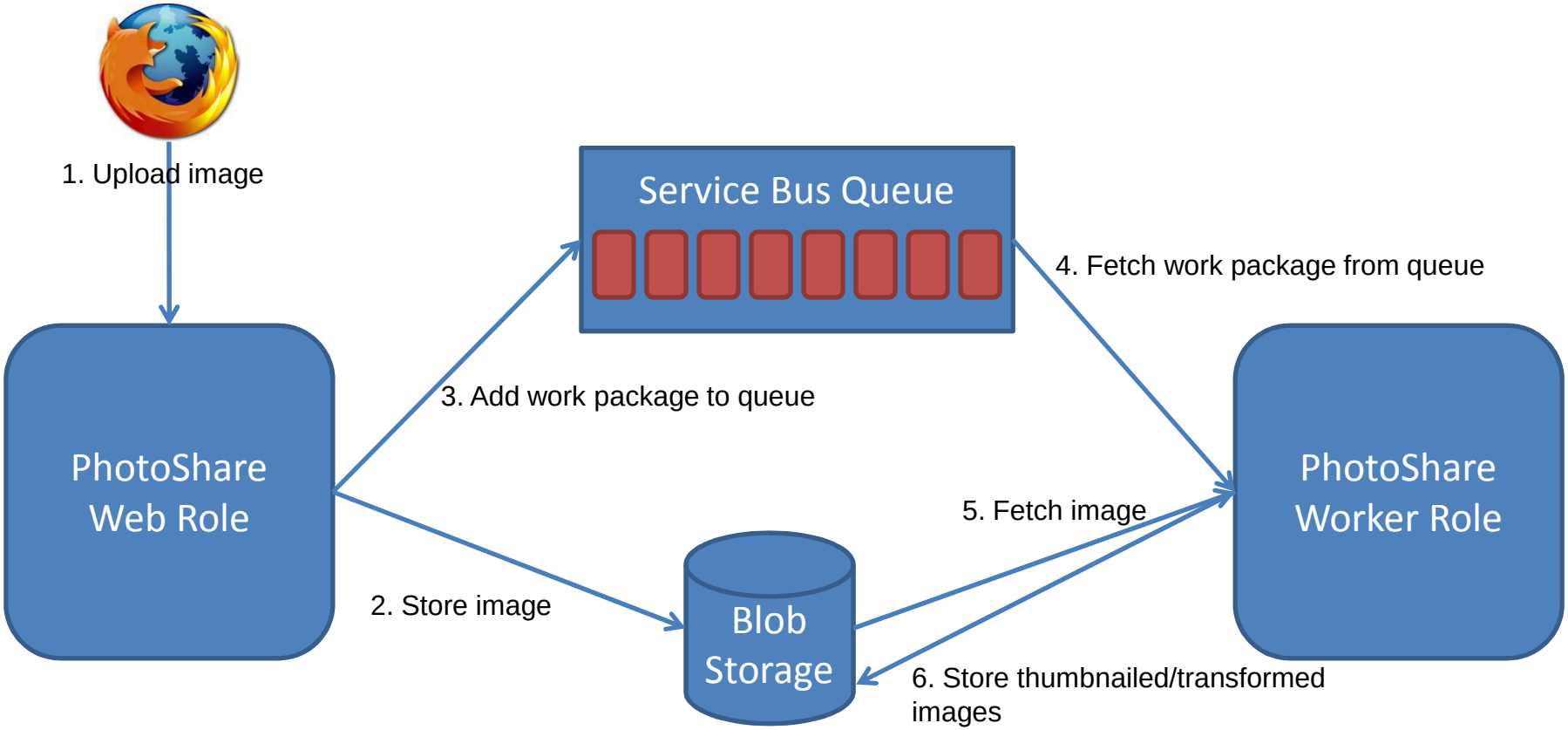
Cloud Computing
Übung 3

Björn Lohrmann
Complex and Distributed IT-Systems
Bjoern.lohrmann@tu-berlin.de

Project Assignment No 3

- Goal: Write a Multi-Tier PaaS application for Photo Sharing using Azure

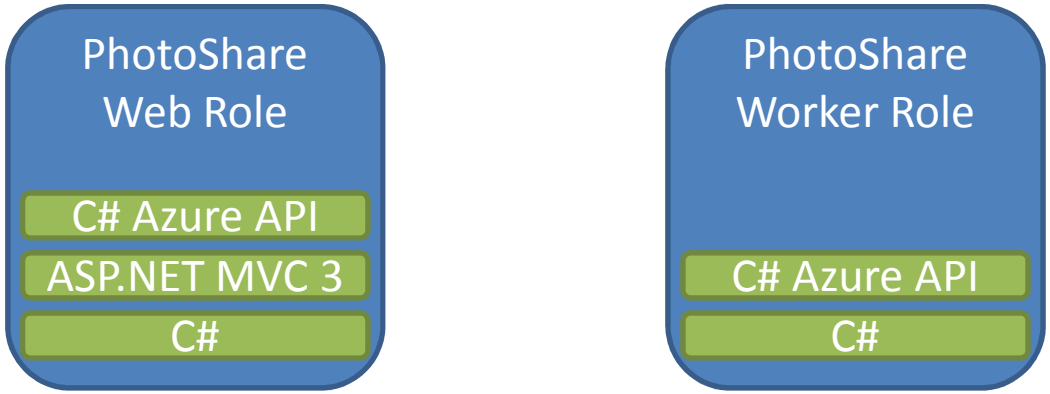


Windows Azure

- Microsoft‘s PaaS service
- Azure applications consist of roles (components)
  - Web roles (serve web requests)
  - Worker roles (do background work)
  - VM role (run „legacy“ software that cannot run in Azure otherwise)
- Azure applications use Azure services
  - Storage (blob, table, queue)
  - ServiceBus (queues, publish/subscribe)
  - Other services (SQL servers, etc)

Technologies

- Microsoft-based technology stack



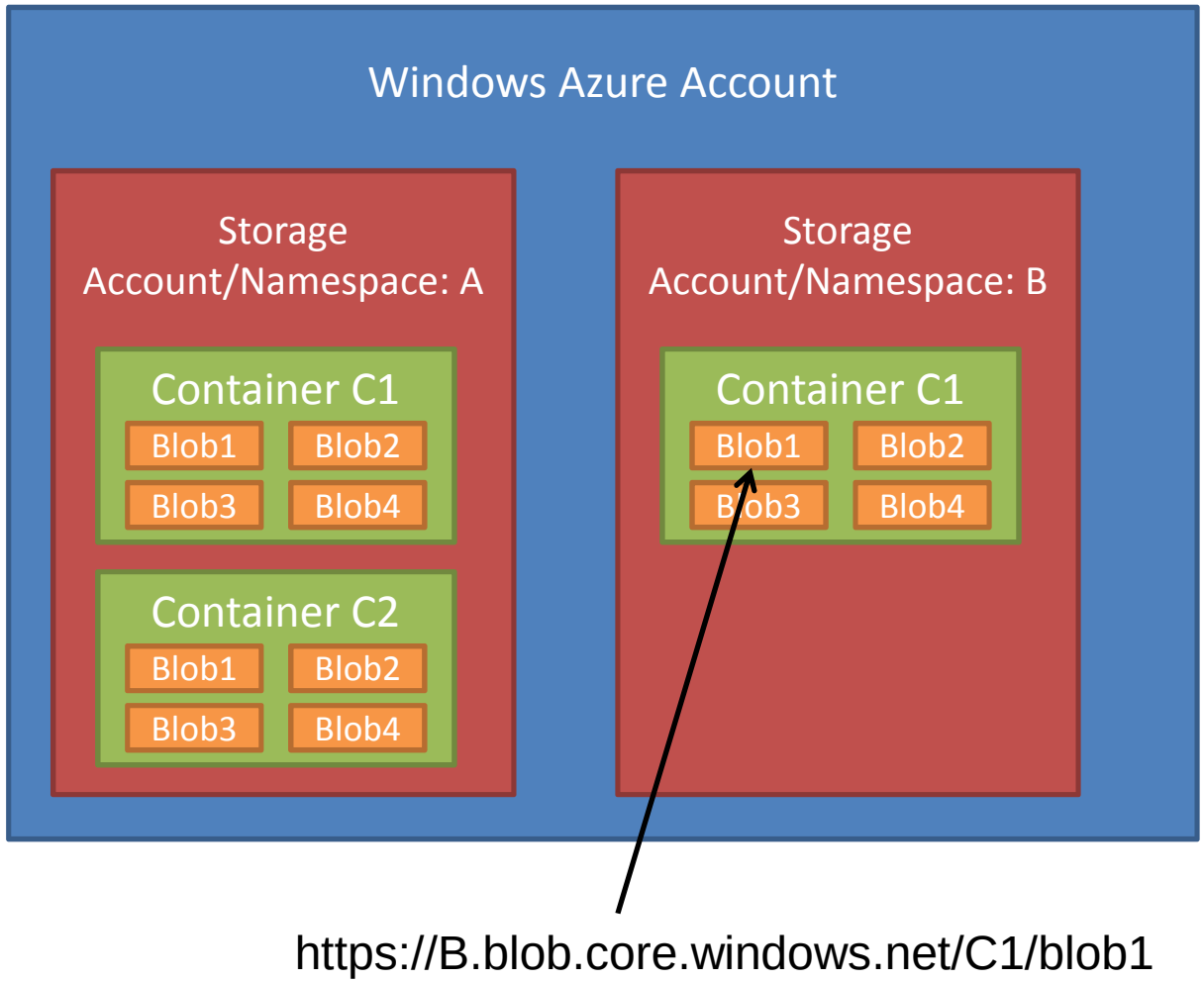
- Agenda today:
  - Azure blob storage and service bus queues
  - C# basics
  - Azure development with Visual Studio

Azure Blob Storage

- Blob = Binary Large Object

- Service that stores large amounts of unstructured/binary data
  - Blobs can have (almost) arbitrary size
  - More than 100TB of blobs allowed per account
  - Suitable for image, video, download etc delivery
  - Accessible directly via HTTP(S): https://namespace.blob.core.windows.net/container/blobname

Azure Blob Storage



- Storage Account Management via Management Portal: http://windows.azure.com
- Container/Blob Management via API
- Requires connection string:
  - `DefaultEndpointsProtocol=https; AccountName=[StorageAccountName]; AccountKey=[AccessKey]`

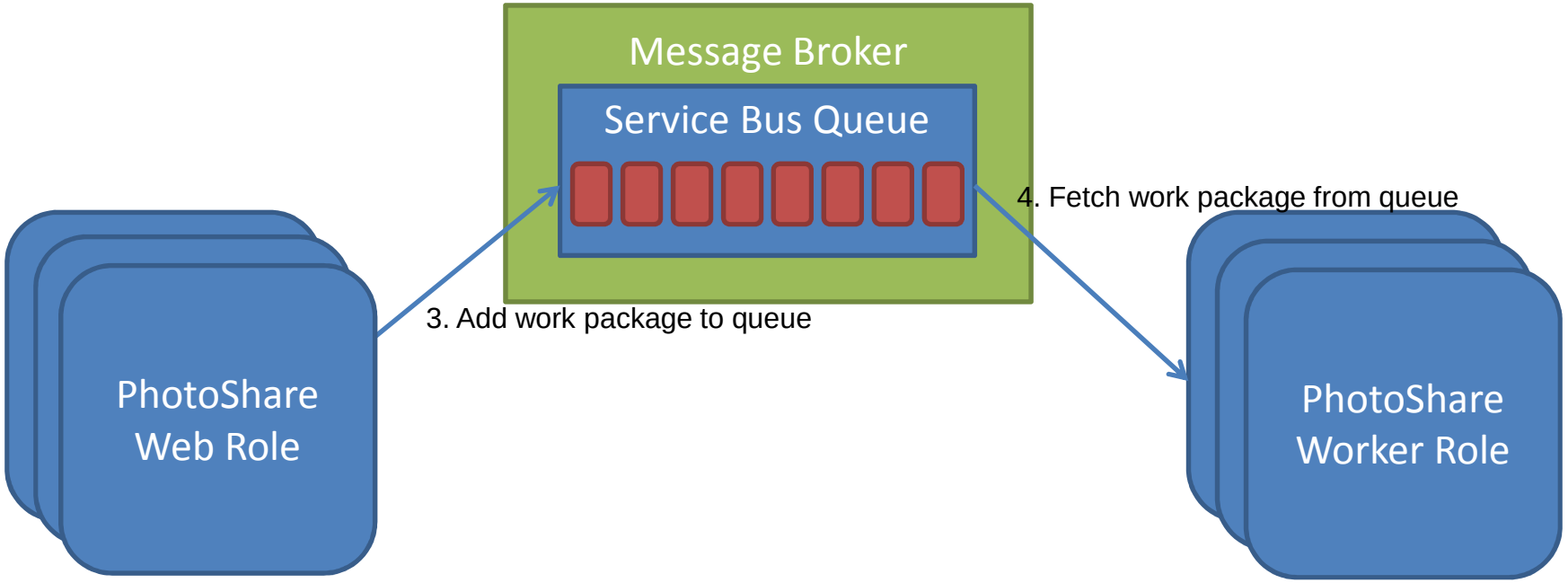
Azure Blob Storage

```
// retrieves connection string setting
CloudStorageAccount acc =
 CloudStorageAccount.FromConfigurationSetting("DataConnectionString");

// initializes connection to blob API
CloudBlobClient blobClient =
 CloudStorageAccountStorageClientExtensions.CreateCloudBlobClient(acc);

// retrieves container reference (container may not exist yet)
CloudBlobContainer thumbnailContainer =
 blobClient.GetContainerReference(„mycontainer");

// creates container if necessary
if (thumbnailContainer.CreateIfNotExist())
{
  // makes blobs in container available without authentication
  thumbnailContainer.SetPermissions(new BlobContainerPermissions { PublicAccess =
    BlobContainerPublicAccessType.Blob });
}

// lists blobs in container
IEnumerable<IListBlobItem> thumbnails = container.ListBlobs();


// uploads a new blob
CloudBlob blob = blobContainer.GetBlobReference(„myblob“);
blob.UploadFromStream(<some-input-stream>);
```

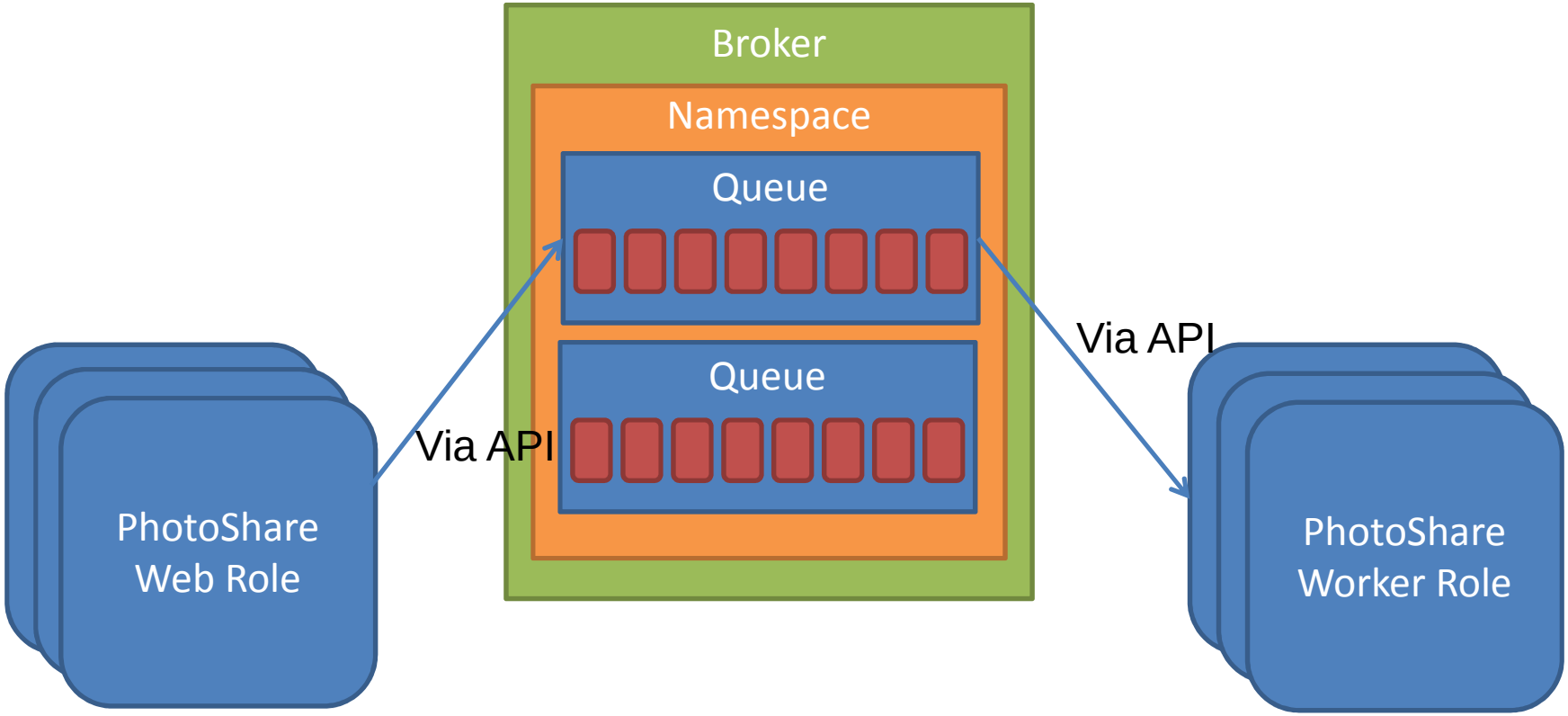
See also: https://www.windowsazure.com/en-us/develop/net/how-to-guides/blob-storage/

Service Bus Queues

- Enables components of distributed application to communicate via brokered queue (FIFO)
  - Reliable (guaranteed delivery)
  - N producers to N consumers
  - Messages can be arbitrary objects



Service Bus Queues



- API access to queue requires connection string:
  `Endpoint=sb://[namespace].servicebus.windows.net/;SharedSecretIssuer=[issuerName];SharedSecretValue=[yourDefaultKey]`
- Namespace Management via Management Portal, Queue management via API

Service Bus Queues

```
// construct URI and authentication token
Uri uri = ServiceBusEnvironment.CreateServiceUri("sb", „mynamespace“ , String.Empty);
TokenProvider tp = TokenProvider.CreateSharedSecretTokenProvider(„myIssuerName“, „myIssuerKey“);


// connect to namespace
NamespaceManager nm = new NamespaceManager(uri, tp);
if (!nm.QueueExists(„myqueue“))
{
    nm.CreateQueue(„myqueue“);
}

// Get a client to the queue
MessagingFactory messagingFactory = MessagingFactory.Create(nm.Address, nm.Settings.TokenProvider);
QueueClient requestQueue = messagingFactory.CreateQueueClient(„myqueue“);

// sending messages
requestQueue.Send(new BrokeredMessage(„My Message“));

// receiving messages
BrokeredMessage receivedMessage = clientQueue.Receive();
string msgString= receivedMessage.GetBody<string>();

// Remove it from the queue
receivedMessage.Complete();
```

C# Introduction

- see C# slides

Azure Development with Visual Studio

- Microsoft offers Azure SDK for Visual Studio
- Required installations
  - Visual Studio 2010 Ultimate Edition (for free with MSDNAA)
  - Visual Studio 2010 Service Pack 1 (update function of VS)
  - Azure SDK for .NET (www.windowsazure.com)
    - Azure .NET libraries
    - Visual Studio extensions
    - Compute and Storage emulators (debugging)
    - Various other tools
- Documentation: http://msdn.microsoft.com/en-us/library/windowsazure/