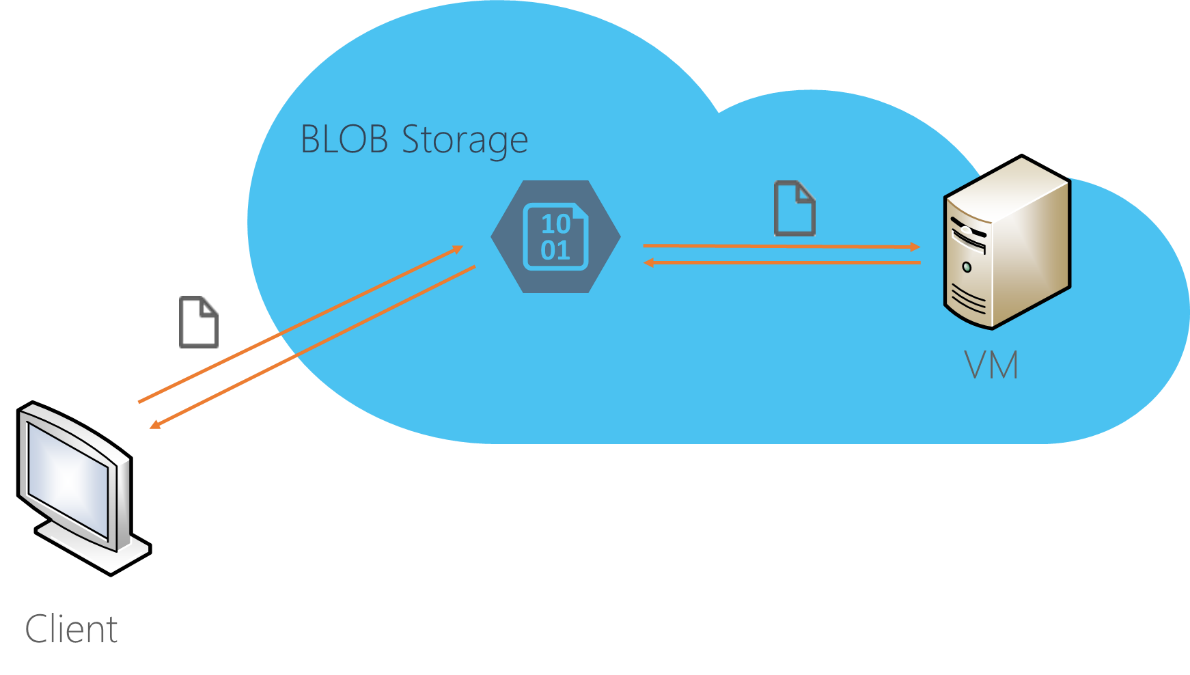
# bbv Academy - Labs

## Lab 1 a)

### Services

Azure BLOB Storage, Azure Virtual Maschines

### Aufbau



### Tasks

1. Erstellen Sie eine neue Azure VM im Management Portal. Sie werden Sie später benötigen.
2. Erstellen Sie einen neuen Storage Account im Azure Management Portal.
3. Verbinden Sie sich mit dem Azure Management Studio mit dem erstellten Storage Account.
4. Erstellen Sie in Visual Studio eine neue Konsolen-Applikation für den Client. Der Quellcode in der Klasse Lab1Client aus dem Flatterist-Projekt soll ihnen dabei helfen.
5. Erstellen Sie eine zweite Konsolen-Applikation für den Server. Der Quellcode in der Klasse Lab1Worker soll Ihnen dabei helfen.
6. Konfigurieren Sie Client- und Server-Applikation mit dem Connection-String des Storage-Accounts.
7. Testen Sie Ihr System lokal.
8. Deployen Sie die Server-Applikation auf die zuvor erstelle Azure VM und testen Sie Ihr System.
9. Beobachten Sie die Veränderungen auf dem Storage Account via Azure Management Studio.
10. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.
11. Löschen Sie allenfalls übriggebliebenen Job-Container im Storage Account.
12. Stoppen Sie die virtuelle Maschine.

### Diskussion

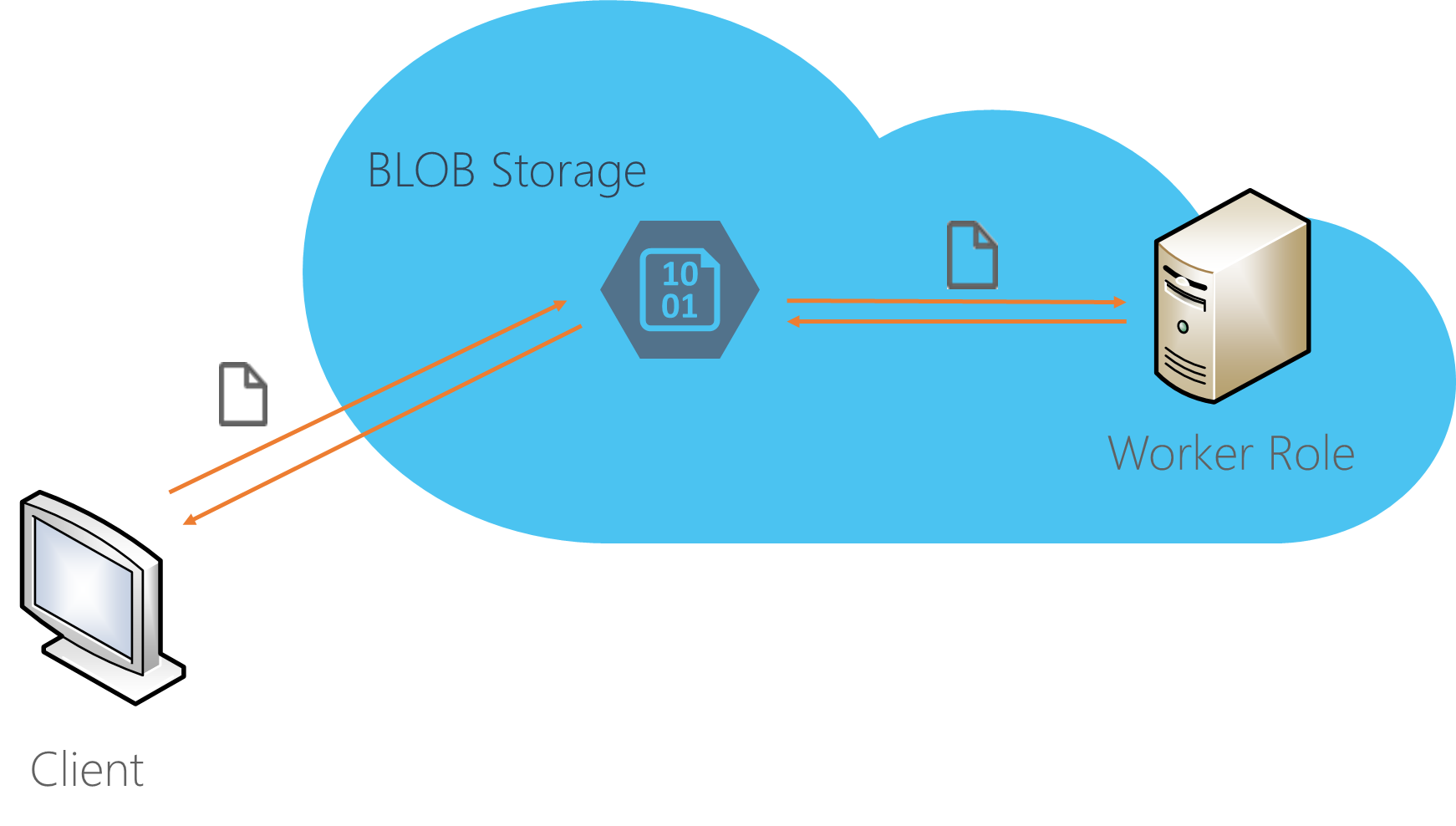
* Auf welches Laufwerk sollte die Server-App installiert werden?
* Auf welches Laufwerk sollten die grössere BLOBs temporär heruntergeladen werden?
* Wie hoch ist die Verfügbarkeit des Systems?

## Lab 1 b)

### Services

Azure BLOB Storage, Azure Worker Roles, Azure Table Storage

### Aufbau



### Tasks

1. Erstellen Sie einen neuen Cloud-Service im Azure Management Studio
2. Erstellen Sie eine neue Azure Worker Role in Visual Studio
3. Übernehmen Sie den serverseitigen Code aus der Übung 1a) in die erstellte Worker Role
4. Übernehmen Sie die Konfiguration aus Übung 1a) in die neue Worker Role
5. Konfigurieren Sie das Trace-Logging in den Azure Table Storage
6. Deployen Sie die Worker Role in den Cloud Service
7. Testen Sie Ihr System
8. Machen Sie sich mit den Log-Ausgaben im Azure Table Storage vertraut.
9. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.
10. Löschen Sie allenfalls übriggebliebene Job-Container im Storage Account.

### Diskussion

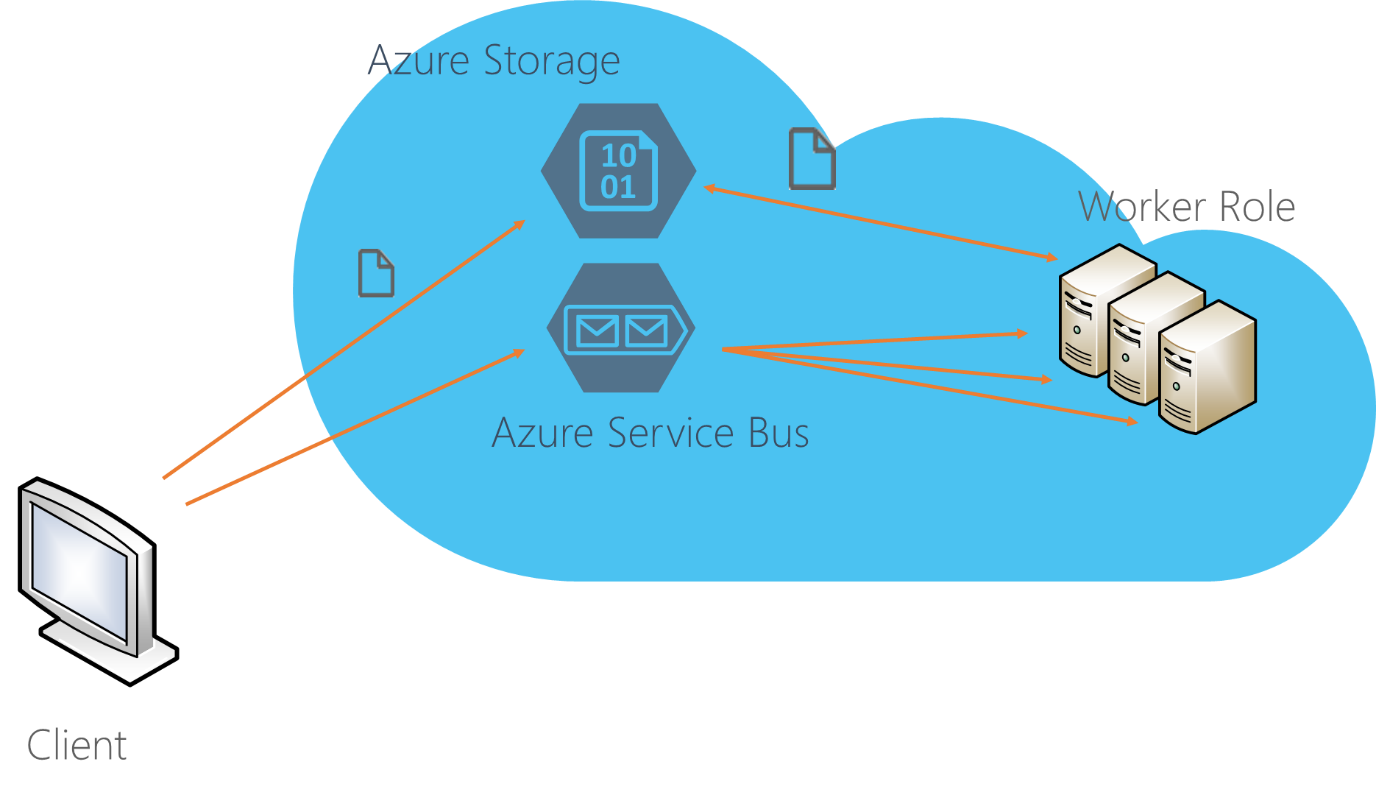
* Wie ändern wir Konfigurations-Einstellungen?
* Können wir via Remote-Desktop auf den Server zugreifen?
* Wie speichern wir serverseitig Daten, die dauerhaft persistent sein müssen?
* Wer führt Betriebssystem-Updates durch?
* Können wir die Worker Role vertikal skalieren (stärkere Maschine)?
* Können wir die Worker Role horizontal skalieren (mehr Maschinen parallel)?
* Was ist der Vorteil von horizontaler Skalierbarkeit gegenüber vertikaler Skalierbarkeit?

## Lab 2

### Services

Azure BLOB Storage, Azure Service Bus, Azure Worker Roles, Azure Table Storage

### Aufbau



### Tasks

Lab 2 setzt auf Lab 1 b) auf. Es wird vorausgesetzt, dass bereits ein Azure Storage Account existiert.

1. Erstellen Sie einen neuen Service Bus Account im Azure Management Portal
2. Verbinden Sie sich mit dem Service Bus Explorer zum erstellten Service Bus Account
3. Ändern Sie Client- und Server-Implementation so ab, dass sie dem obenstehenden Bild entsprechen. Der Quellcode in der Flatterist-Applikation unterstützt Sie dabei.
4. Testen Sie ihr System lokal. Öffnen Sie mehrere Client- und mehrere Server-Instanzen parallel. Beobachten Sie wie sich ihr System verhält (via Azure Management Studio und Service Bus Explorer).
5. Deployen Sie das System nach Azure und konfigurieren Sie das AutoScaling. Beobachten Sie wie sich ihr System verhält.
6. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.
7. Löschen Sie allenfalls übriggebliebene Job-Container im Storage Account.

### Diskussion

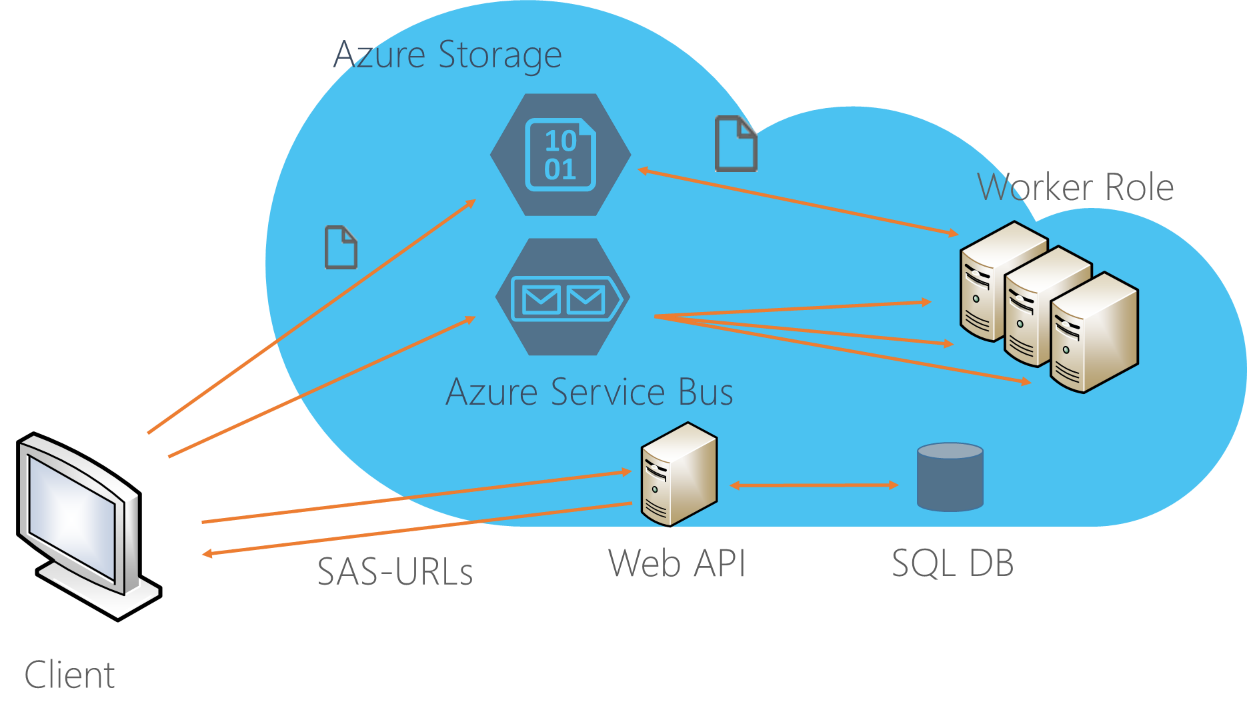
* Service Bus Messages müssen immer idempotent verarbeitet werden. Was heisst das und ist in der Flatterist-Applikation Idempotenz gewährleistet?
* Welche Nachteile hat diese Lösung in Bezug auf die Sicherheit?

## Lab 3 a)

### Services

Azure Websites, SQL Azure Databases, Azure BLOB Storage, Azure Service Bus, Azure Worker Roles, Azure Table Storage

### Aufbau



### Tasks

1. Erstellen Sie eine neue SQL Azure Datenbank im Management Portal. Konfigurieren Sie die Firewall Einstellungen ihres SQL Azure Servers, damit der Zugriff von ihrem lokalen Computer möglich ist.
2. Verbinden Sie sich mit dem Azure Management Studio mit der neu erstellen SQL Azure Datenbank.
3. Erstellen Sie eine neue Azure Website.
4. Erstellen Sie in Visual Studio eine neue Web-API Applikation. Fügen Sie einen API-Controller und den Datenbank Zugriff hinzu. Verwenden Sie dazu den Code aus der Flatterist Applikation.
5. Fügen Sie die Konfiguration in die Web.config Datei ihrer Web-API-Applikation hinzu.
6. Erstellen Sie das Datenbank-Schema mit Hilfe von EF Code First Migrations.
7. Erweitern Sie die Client- und Worker-Applikation aus dem vorangegangenen Lab. Der Quellcode aus der Flatterist-Applikation hilft ihnen dabei.
8. Testen Sie ihr System lokal.
9. Deployen Sie die Worker- und die Web-API-Applikationen nach Azure.
10. Testen Sie ihr System in der Cloud.
11. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.
12. Löschen Sie allenfalls übriggebliebene Job-Container im Storage Account.

### Diskussion

* Ist diese Architektur multi-tenant fähig? Welche Vorteile ergeben sich aus einer multi-tenant Architektur?
* Wie könnten wir grösseren Kunden einen Premium-Service mit garantierten Verarbeitungszeiten anbieten?
* Was fehlt noch zu einem fertigen Framework?

## Lab 3 b) (optional)

### Tasks

Verwenden Sie für das Deployment Ihrer Azure Website Git. Eine Anleitung dazu finden Sie unter:

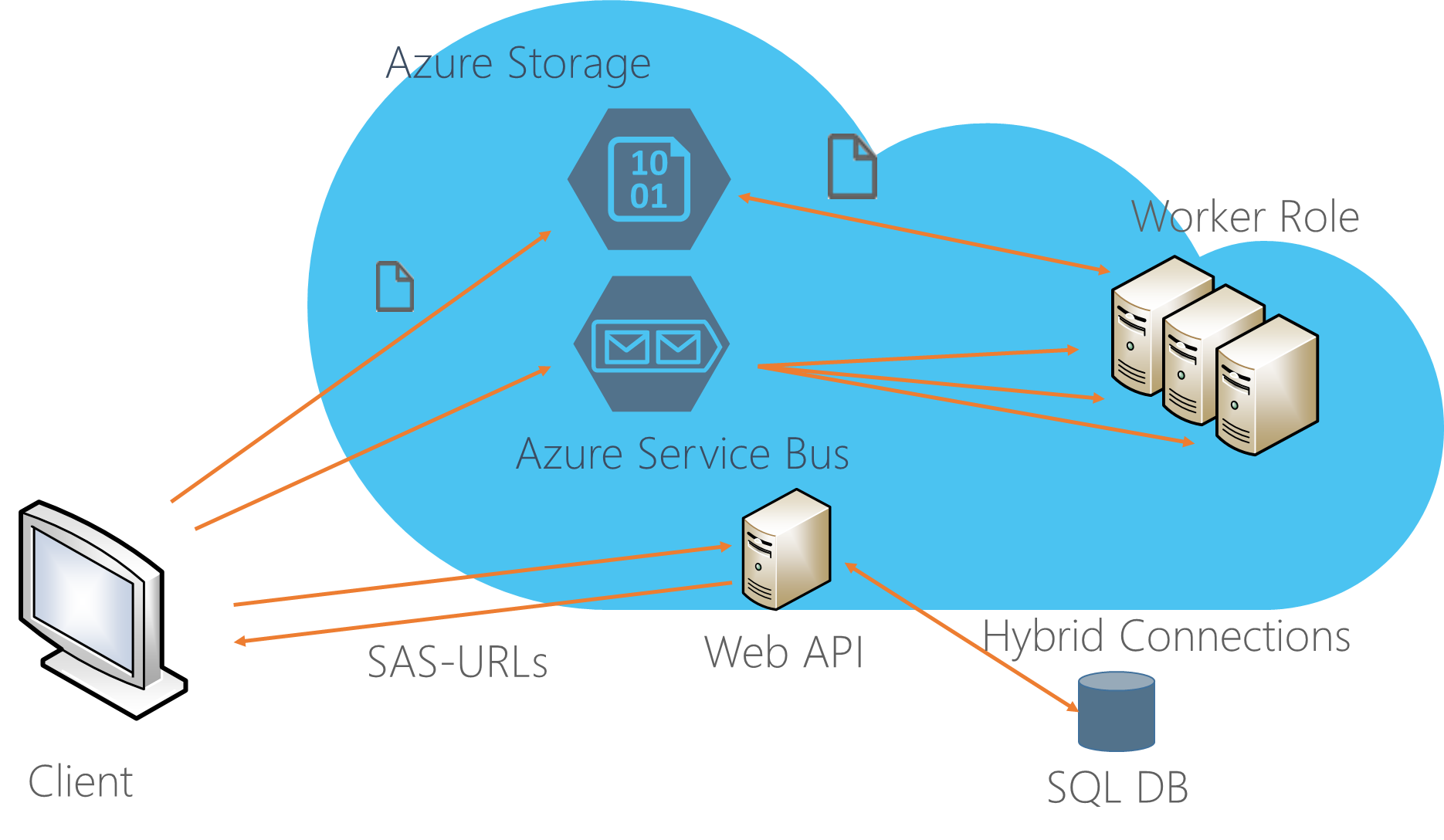
<http://azure.microsoft.com/en-us/documentation/articles/web-sites-publish-source-control>

### Diskussion

Sie wollen Continuous Deployment einrichten. Was müssen Sie tun?

## Lab 3 c) (optional)

### Aufbau



### Tasks

1. Implementieren Sie Lab 3 mit einer lokalen Datenbank (ohne SQL Azure). Verwenden Sie dazu Hybrid Connections von Azure Websites.
2. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.

### Diskussion

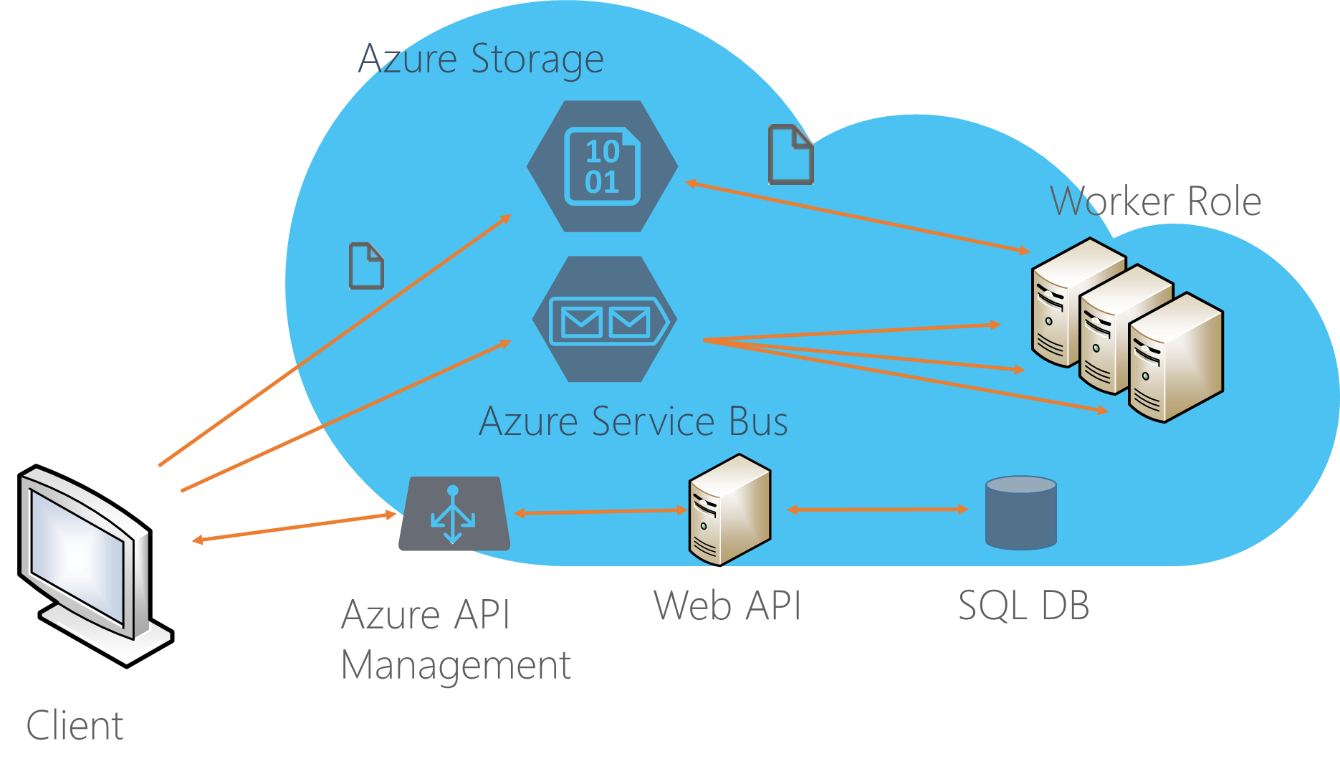
In welchen Szenarien sind Hybrid Connections sinnvoll (Business Case)?

## Lab 4

### Services

Azure API Management Service

### Aufbau



### Tasks

Bauen Sie ihre Lösung von Lab 3 so um, dass der Client nicht direkt mit der Web-API-Applikation kommuniziert sondern via Azure API Management Service.

1. Erstellen Sie einen Azure API Management Account.
2. Verbinden Sie sich mit dem API Management Portal und richten Sie das API ein.
3. Erstellen Sie ein Produkt und berechtigen Sie die „Developers“ Gruppe dem Produkt hinzu. Weisen Sie auch das zuvor erstellte API dem Produkt zu.
4. Fügen Sie einen Benutzer zur Developer Gruppe hinzu.
5. Loggen Sie sich mit dem eben erstellen Benutzer auf dem Developer Portal ein. Unter dem Menu-Punkt „APIs“ finden Sie Code-Beispiele in C# wie Sie die Client-Applikation anpassen müssen und eine Konsole.
6. Testen Sie das API mit der bereitgestellten Konsole.
7. Passen Sie die Client-Applikation an.
8. Testen Sie Ihr System.
9. Spielen Sie mit den Policies.
10. Studieren Sie die Informationen im Tab Analytics.
11. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.

### Diskussion

1. Die Flatterist-Applikationen bieten kein öffentliches API an. Ist das Azure API Management hier überhaupt von nutzen?
2. Wie sichern Sie Ihr ursprüngliches API vor unerlaubtem Zugriff?

## Lab 5 a)

### Services

Azure Automation, Azure Virtual Maschines

### Tasks

Ziel dieses Labs ist, eine laufende Virtuelle Maschine mittels PowerShell zu stoppen.

1. Stellen Sie sicher, dass sie eine laufende Virtuelle Maschine haben, die sie stoppen können (zum Beispiel sie VM aus Lab 1a).
2. Stellen Sie sicher, dass PowerShell auf Ihrem lokalen System installiert ist. <http://technet.microsoft.com/en-us/library/hh847837.aspx#BKMK_InstallingOnWindows8andWindowsServer2012>
3. Installieren Sie das Windows Azure Powershell Module <http://go.microsoft.com/?linkid=9811175&clcid=0x409>
4. Verbinden Sie sich mit ihrer Azure Subscription. Details dazu finden Sie im Dokument „Azure Automation Lab.docx“.
5. Stoppen sie die Virtuelle Maschine. Verwenden Sie dazu das die Azure Commandlets  
   Get-AzureVM bzw. StopAzureVM  
   <http://msdn.microsoft.com/en-us/library/dn495236.aspx>  
   <http://msdn.microsoft.com/en-us/library/dn495269.aspx>

### Diskussion

Wie verwalten Sie Ihre PowerShell Scripts?

## Lab 5 b)

### Tasks

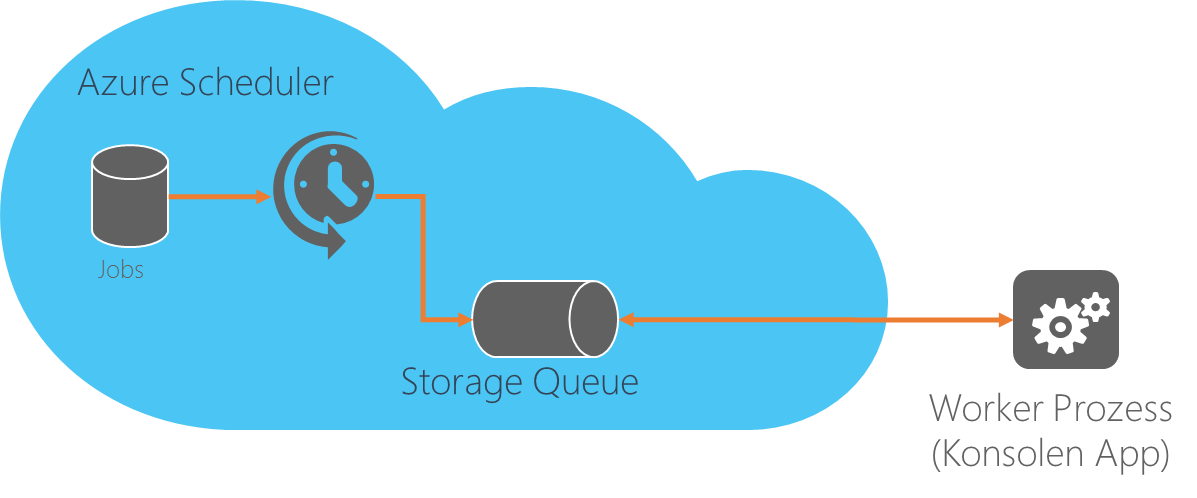
1. Automatisieren Sie den Vorgang in Lab 5a) mittels Azure Automation. Verwenden Sie dazu folgende Schritt-für-Schritt-Anleitung:  
   <http://blogs.technet.com/b/keithmayer/archive/2014/04/06/step-by-step-getting-started-with-windows-azure-automation.aspx>  
   Das PowerShell-Script am Ende des Tutorials müssen Sie leicht Ihren Gegebenheiten anpassen.
2. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.

## Lab 6

### Services

Azure Scheduler, Azure Storage Queues

Aufbau



Tasks

1. Erstellen Sie eine Azure Storage Queue.
2. Erstellen Sie eine Azure Scheduler Collection und ein Azure Scheduler Job, welcher in 1-Minuten-Abständen eine Nachricht in die Azure Storage Queue abfüllt.
3. Prüfen Sie mittels Azure Management Studio, ob die Nachrichten in die Queue abgefüllt werden.
4. Lesen Sie die Nachrichten mittels einer Konsolen-Applikation aus der Queue aus. Das Beispiel unter <http://azure.microsoft.com/en-us/documentation/articles/storage-dotnet-how-to-use-queues/#get-message> hilft Ihnen dabei.
5. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.

### Diskussion

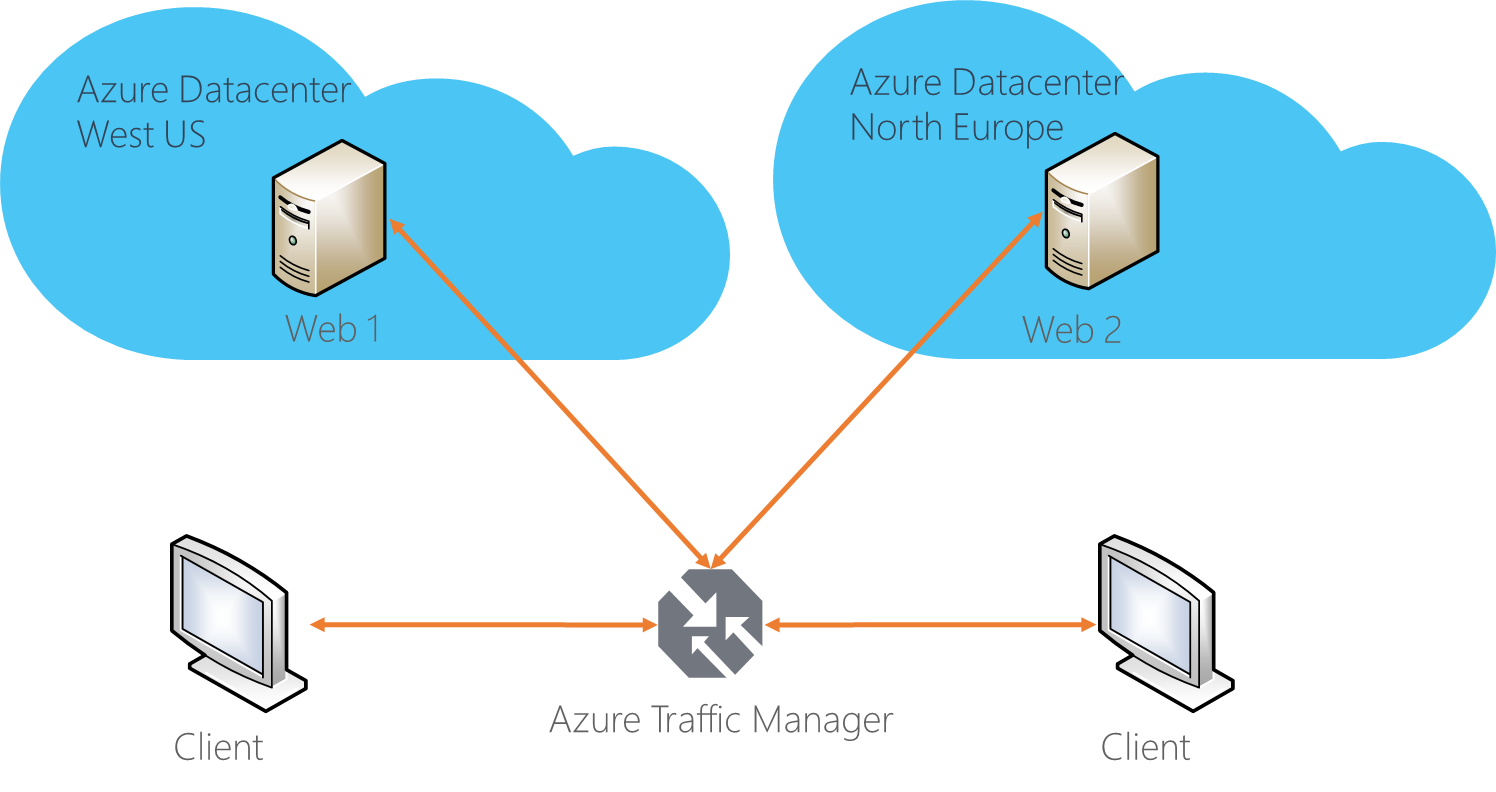
Was sind die Vorteile, wenn anstelle des direkten HTTP-Calls der Umweg über eine Storage Queue verwendet wird?

## Lab 7

### Services

Azure Traffic Manager, Azure Websites

### Aufbau



### Tasks

1. Erstellen Sie eine Web-Seite und deployen Sie diese Webseite in zwei unterschiedlichen Datencentern als Azure Website. Achten Sie darauf, dass sie die Deployments unterscheiden können.
2. Erstellen Sie einen Traffic Manager Endpoint für diese beiden Webseiten.
3. Erstellen Sie eine Virtuelle Maschine in der Region ihrer zweiten Webseite, verbinden Sie sich darauf und öffnen Sie den Traffic Manager Endpoint.
4. Spielen Sie mit den unterschiedlichen Verfahren (Performance, Round-Robin, Failover).
5. Identifizieren Sie anhand der Webseite <http://azure.microsoft.com/pricing> die Kostenkomponenten Ihres Systems.

### Diskussion

Wenn Sie zurück an die Flatterist-Applikation denken, wie würden Sie den Traffic-Manager einsetzen?