

Windows Azure Platform - Die Cloud-Lösung von Microsoft

Holger Reitz

Fakultät für Informatik
Hochschule Mannheim
Paul-Wittsack-Straße 10
68163 Mannheim

`holger.reitz1@stud.hs-mannheim.de`

Zusammenfassung Die Windows Azure Platform ist die aktuelle Cloud-Lösung von Microsoft. Die Plattform baut auf die bereits vorhandenen Cloud Services von Microsoft auf und erweitert diese Dienste um weitere Funktionalitäten. Windows Azure Platform besteht aus drei Komponenten, dem Betriebssystem Windows Azure, welches als Laufzeitumgebung für Cloud Anwendungen dient, dem Datenbankmanagementsystem SQL Azure, welches Datendienste zum Speichern und Synchronisieren der Daten anbietet, und Infrastrukturdiensten .NETServices, welche Rechteverwaltung und Zugriffsverwaltung auf lokale Anwendungen ermöglicht. Die Plattform ist nicht nur für Windows-Entwickler interessant, sondern auch für Software-Entwickler anderer Programmiersprachen, wie z.B. Java, da es vielfältige Nutzungsmöglichkeiten gibt. Das Dokument beschreibt den Aufbau und Verwendung der Windows Azure Platform und derer Komponenten.

1 Einleitung

Cloud Computing ist zurzeit das grosse Thema in der IT-Welt und wurde auch von Microsoft aufgegriffen. Mit der Windows Azure Platform bietet nun auch Microsoft eine Cloud-Plattform für Entwickler an. Allgemein kann man sagen, dass sich gewisse Eigenschaften, die sich aus zahlreichen Definitionen des Begriffs Cloud Computing hervorgegangen sind, herauskristallisiert haben:

Eine IT-Funktion wird, servicebasiert auf Basis einer hoch skalierbaren Infrastruktur über standardisierte Schnittstellen, angeboten. [3]

Bei einer IT-Funktion kann es sich dabei um eine virtualisierte Hardware, also Infrastructure-as-a-Service (IaaS), eine Infrastruktur- und Anwendungsservices, also Platform-as-a-Service (PaaS), und Softwaresysteme, also Software-as-a-Service (SaaS), handeln. Microsoft hat in all diesen eben genannten Bereichen in den letzten Jahren Fuss gefasst und ihr Produktfolio ausgebaut.

Als technisches Fundament für alle Cloud Services von Microsoft dienen weltweit aufgebaute Rechenzentren. In diesen werden IaaS-Dienste erbracht, die sogenannten “Global Foundation Services“. Im November 2008 veröffentlichte Microsoft die Windows Azure Platform, welche ein PaaS-Angebot darstellt. Aktuell ist die Plattform noch als Community Technology Preview (CTP) [4] für die Entwickler verfügbar. Für Februar 2010 ist dann die endgültige Einführung der kommerziellen Verfügbarkeit geplant, wobei die Plattform für Kunden ab Januar 2010 freigegeben wird. Dies hat Ray Ozzie, Chief Software Architect bei Microsoft, auf der im November stattgefundenen Microsofts Professional Developers Conference (PDC09) bekanntgegeben. Microsoft hat dabei folgende Vision:

three screens and a cloud. [7]

Mit “screens“ sind Endgeräte wie Mobiltelefone, Personal Computer und Fernseher gemeint, die alle über die Cloud verbunden werden. Zudem wird in Zukunft die Windows Azure Platform auch die Basis für Microsofts SaaS-Angebote “Microsoft Live“ und “Microsoft Online“ darstellen. Allgemein zusammengefasst stellt Azure Rechen- und Speicherdienste auf Basis der GFS zur Verfügung, und erlaubt es, den Entwicklern diese Dienste zu nutzen. Entwickler können eigene Anwendungen um Cloud Services erweitern, wie z.B. Datenaustausch, Speichern von Daten in der Cloud etc., oder eigene Anwendungen in der Cloud erstellen, überwachen und betreiben. Alle Cloud Services sind dabei über Standardschnittstellen (“http“, REST, SOAP, XML, etc.) erreichbar.

2 Windows Azure Platform

2.1 Einführung

Die Windows Azure Platform stellt eine Ansammlung an Cloud-Technologien zur Verfügung, um verschiedene Dienste für Anwendungsentwickler anzubieten. Wie in der Einleitung beschrieben, können dies Rechen- und Speicherdienste sein. Dabei können Anwendungen, die diese Dienste nutzen, entweder komplett in der Cloud betrieben werden oder auf lokalen Systemen bei Unternehmen laufen. Auch reine Desktop-Anwendungen und mobile Anwendungen können die Plattform und deren Dienste nutzen. [1] Für die optimale Nutzung der Dienste ist eine architektonisch gut aufgebaute Infrastruktur von Nöten. Abbildung 1 zeigt dabei eine Übersicht über den Aufbau der Plattform und das Zusammenspiel der Komponenten sowie Anwendungen.

Wie auf der Abbildung 1 zu sehen ist besteht die Windows Azure Platform aus drei Komponenten:

- Windows Azure: Diese Komponente bietet eine auf Windows basierte Laufzeitumgebung, um Anwendungen in der Cloud zu betreiben und deren Daten zu speichern. Physikalisch läuft Windows Azure auf Servern in Microsofts Rechenzentren.

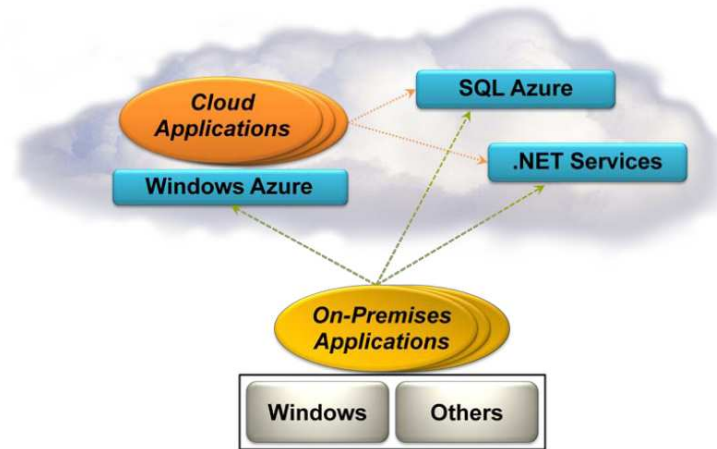


Abbildung 1. Windows Azure Plattform enthält Anwendungen, Daten und Infrastruktur in der Cloud. [1]

- SQL Azure: SQL Azure stellt Datendienste in der Cloud zur Verfügung und basiert auf Microsoft SQL Server.
- .NET Services: Die letzte Komponente bietet verteilte, infrastrukturelle Dienste für Cloud-basierte und lokale Anwendungen an.

Lokale Anwendungen, dies können Windows-Anwendungen sein aber auch andere Anwendungen, können die Rechenpower von Windows Azure nutzen, Datenbanken mit SQL Azure erstellen und nutzen und auf die .NET Services zurückgreifen. Das gleiche gilt auch für die Cloud Anwendungen mit dem kleinen Unterschied, dass diese Anwendungen direkt auf Windows Azure laufen. In den nachfolgenden Kapiteln werden die einzelnen Komponenten genauer beschrieben.

3 Windows Azure

3.1 Architektur

Windows Azure stellt eine Plattform für ausführbare Windows Anwendungen und für das Speichern von Daten in der Cloud dar. In der Abbildung 2 wird der Aufbau von Windows Azure dargestellt.

Als physikalische Basis an Ressourcen dient eine große Anzahl Rechner, die die Rechenpower für Windows Azure zur Verfügung stellen. Die Rechner befinden sich alle in Rechenzentren von Microsoft. Die sogenannte Windows Azure Fabric dient als "Controller" und "Manager" und bündelt die Vielzahl an Rechenpower

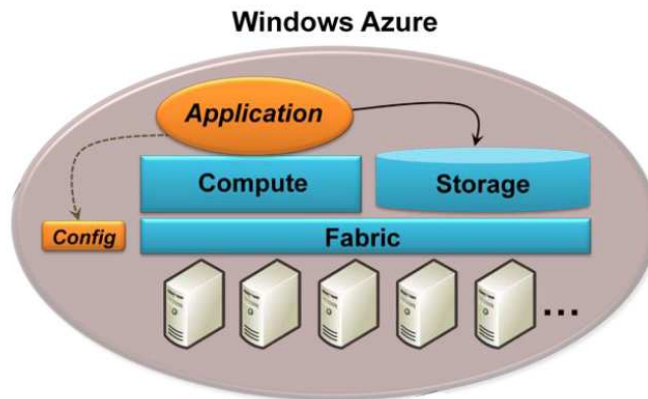


Abbildung 2. Windows Azure bietet Windows-basierte Rechen- und Datendienste für Anwendung an. [1]

zu einem großen Ganzen, so dass die Ressourcen an den Azure Compute Service je nach Bedarf verteilt werden können. Auf Azure Fabric baut dann der Azure Compute Service und Azure Storage Service auf. Die Anwendungen werden dann auf dem Azure Compute Service ausgeführt und haben Zugriff auf den Azure Storage Service, dieser kann mittels REST, SOAP oder auch XML erfolgen. Zu jeder Anwendung oder jedem Web Service gehört eine sogenannte Configuration File, über die Einstellungen zur Verwaltung der Anwendungen vorgenommen werden können. Windows Azure erfüllt hauptsächlich zwei Aufgabenbereiche, das Ausführen von Anwendungen und das Speichern von Daten. Wie diese Aufgaben bewältigt werden und welche Komponenten dabei eine Rolle spielen wird nachfolgend erläutert.

3.2 Azure Compute Service

Die Laufzeitumgebung basiert auf Windows Server und war zu Beginn der CTP nur für .NET-Anwendungen ausgelegt, was bedeutete, dass nur diese Anwendungen unter Windows Azure ausführbar waren. Im Laufe der Zeit wurde der Umfang um weitere Programmiersprachen erweitert, wodurch auch andere Anwendungen lauffähig sind. Die unterstützten Sprachen sind u.a. C#, C++, Visual Basic, Java etc. Auch Webanwendungen, die z.B. Technologien wie ASP.NET [13] und WCF (Windows Communication Foundation) [11] und andere benutzen, sind auf dem Azure Compute Service ausführbar. Damit die Anwendungen unter Windows Azure laufen können, benutzen diese Web Role Instanzen und Worker Role Instanzen. Dabei können Anwendungen mehrere Instanzen benutzen, je nachdem wie viel Rechenleistung benötigt wird, kann die Anzahl der Instanzen erhöht oder verringert werden. Jede Instanz läuft in ihrer

eigenen Virtual Machine (VM). Diese Virtual Machine ist ein 64-bit Windows Server 2008 Cluster und wird durch einen Hypervisor (Microsoft Hyper-V) [12] unterstützt. Durch das Konzept, mit der Zuweisung von Instanzen zu einer Anwendung, kann Windows Azure die Skalierbarkeit des Systems gewährleisten. Ein weiterer wichtiger Punkt hierfür ist, dass für jede Rolle die Anzahl der gleichzeitig laufenden Instanzen festgelegt werden kann. Je nach sich ändernder Last kann dann die Konfiguration geändert werden und Windows Azure sorgt automatisch für das Zu- oder Abschalten von Role-Instanzen. Eine Übersicht der Komponenten des Azure Compute Service stellt die Abbildung 3 dar.

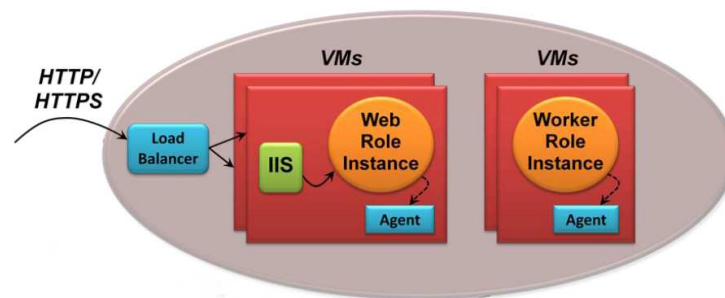


Abbildung 3. Windows Azure Anwendungen können aus Web Role und Worker Role Instanzen bestehen. [1]

3.3 Web Role

In erster Linie sind solche Web Role Instanzen dafür zuständig, um Aufruf-schnittstellen bereitzustellen, z.B. für Web Services oder Weboberflächen. Dabei werden die Anfragen akzeptiert, die über "http" oder "https" eingehen. Als erstes empfängt der Load Balancer diese Anfragen und verteilt diese dann auf die zur Verfügung stehenden Instanzen der Anwendung. In der Instanz selbst wird eine Anfrage dann über den Internet Information Service (IIS 7) von Microsoft [14] an die Web Role Instanz weitergeleitet, wo diese dann verarbeitet werden kann. Programmiertechnisch können Web Roles mit ASP.NET, WCF oder auch mit einer anderen Technologie, die mit IIS arbeiten kann, implementiert werden.

3.4 Worker Role

Im Gegensatz zu den Web Roles können die Worker Roles keine Anfragen von außerhalb, also per "http" oder "https" Anfrage, akzeptieren oder annehmen,

da keine eingehenden Netzwerkverbindungen erlaubt sind und auch kein IIS in der Instanz vorgesehen ist. Eine Worker Role ist aber durchaus in der Lage eine ausgehende Verbindung aufzubauen, genauso wie die Web Roles es auch sind. Die Worker Role Instanzen werden dazu verwendet, um Hintergrunddienste zu implementieren und ablaufen zu lassen. Die Arbeitsaufträge erhält eine Worker Role von Web Roles oder lokalen Anwendungen. Diese Aufträge werden von den Anwendungen in einer Queue im Azure Storage Service gespeichert. Auf die Queue-Einträge haben die Worker Roles Zugriff und können diese abarbeiten. Die Worker Roles können mit jeder beliebigen Windows Technologie implementiert werden, einzige Voraussetzung ist, dass eine `main()`-Methode vorhanden ist um die Role zu starten.

3.5 Windows Azure Agent

Eine wichtige Komponente, um die Funktionalität zu gewährleisten, fehlt allerdings noch. Diese Komponente ist der sogenannte Windows Azure Agent, welcher für die Kommunikation und die Interaktion mit der Azure Fabric verantwortlich ist. Zu dem stellt er die Windows Azure API zur Verfügung. Dies ist eine sehr wichtige Schnittstelle, da ohne die Kommunikation zwischen den Instanzen und der Fabric keine Zuteilung der physischen Rechenpower möglich wäre und somit die Anwendungen nicht lauffähig wären. Während der CTP Phase unterstützte Windows Azure eine "one-to-one" Beziehung zwischen einer Virtual Machine und einem physikalischen Prozessorkern. Dadurch hat jede Instanz ihren eigenen, zugeordneten Prozessorkern. Wenn es also nötig ist, die Performance der Anwendung zu steigern, kann man, wie schon erwähnt, einfach die Anzahl der Instanzen erhöhen, was zum Ergebnis hat, dass nun durch die Verwendung eines Prozessors pro Instanz mehr Rechenleistung für die Anwendung zur Verfügung steht. Die Azure Fabric verwaltet dann die Neuverteilung der Ressourcen und erkennt, wenn Instanzen fehlerhaft starten oder laufen und erstellt dann eine neue Instanz und weist sie der entsprechenden Anwendung zu.

3.6 Azure Storage Service

Wie schon oben erwähnt, können Anwendungen auch Daten unter Windows Azure speichern und auf diese zugreifen, dies soll die Anwendungen unterstützen, die auf Azure basieren. Die Datendienste erfolgen über den Azure Storage Service, welcher allerdings keine relationale Datenbank darstellt und auch kein SQL unterstützt und nur einfache, skalierbare Speicherarten zur Verfügung stellt. Dabei gibt es drei verschiedene Arten Daten im Azure Storage Service zu speichern:

1. Blobs (Binary large objects):
Diese Speicherart stellt die einfachste Art und Weise dar Daten, im Speziellen große Dateien, zu speichern. Ein Account kann dabei einen oder mehr sogenannte Container haben, wobei jeder Container wiederum einen oder

mehrere Blobs enthalten kann. Die Speichergröße der Binary large objects kann bis zu 50GB (Gigabyte) betragen. Da dies recht große Datenmengen darstellt und somit die Transferrate unter der Größe leidet, bietet der Azure Storage Service die Möglichkeit ein Blob in mehrere Blöcke aufzuteilen, so dass der Transfer der Daten schneller von statten gehen kann. Ein weiteres Feature, welches Blobs anbietet, ist es, dass Blobs Metadaten enthalten können (z.B. Titel, Interpret einer MP3-Datei oder Aufnahmeort- und Datum einer JPG-Datei, etc.).

2. Tabellen (Tables):

Um die Daten strukturierter zu speichern, kann man Tabellen (Tables) benutzen. Genutzt werden können diese Tabellen über eine einfache "query-language", stellen aber keine relationale Tabellen dar. Die zu speichernden Daten können in einem Set an Entitäten mit Objekten gespeichert werden. Die Objekte können dabei verschiedene Datentypen haben wie z.B. "string", "int", "bool", "dateTime", etc. Zugriff auf die Daten erhält man über ADO.NET [9] Datendienste oder LINQ (Language INtegrated Query) [10]. Die Tabellen können sehr viel Speicherplatz in Anspruch nehmen, es sind bis zu Terabytes an Daten möglich. Damit die Performanz nicht leiden muss, kann Azure Storage diese Datenmengen über mehrere Server verteilen, so dass "parallel" darauf zugegriffen werden kann.

3. Queues:

Als dritte Speicherart stehen die Queues zur Verfügung, welche allerdings nicht genutzt werden, um Daten zu speichern, sondern um eine Art der Kommunikation zwischen Komponenten von Windows Azure zu realisieren. Im Speziellen bietet diese Speicherart einen Weg an, damit Web Role Instanzen und Worker Role Instanzen miteinander kommunizieren können. Es werden z.B. Aufträge von den Web Roles oder Anwendungen gespeichert und diese werden dann von den Worker Roles ausgeführt, wie schon bei dem Kapitel über die Worker Roles beschrieben.

Abbildung 4 zeigt einen graphischen Überblick über die Azure Storage Speicherarten.

Um Fehlertoleranz zu gewährleisten, werden alle gespeicherten Daten dreifach redundant gespeichert, somit ist der Verlust einer Kopie nicht weiter tragisch, da noch zwei Sicherheitskopien bestehen. Damit einfach auf die Daten zugegriffen werden kann, benutzen alle drei Speicherarten die Konventionen von RESTful und können über Standard http-Operationen (z.B. "GET", "PUT", ...) erreicht werden.

Hierzu ein paar Beispiel Anfragen um die Daten in Azure Storage zu erreichen:

- Blob:
`http://<StorageAccount>.blob.core.windows.net/<Container>/<BlobName>`
- Tabelle:
`http://<StorageAccount>.table.core.windows.net/
<TableName>?$filter=<Query>`

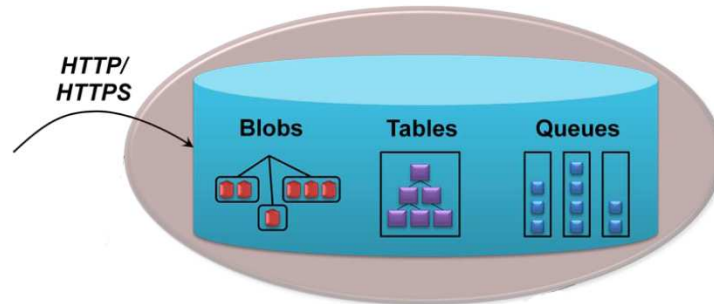


Abbildung 4. Windows Azure ermöglicht es Daten in Blobs, Tabellen und Queues zu speichern. [1]

– Queues:

`http://<StorageAccount>.queue.core.windows.net/<QueueName>`

<StorageAccount> ist eine Kennung, die beim Erstellen eines neuen Accounts generiert wird und dient zur eindeutigen Identifikation von Blobs, Tabellen und Queues. <Container> und <BlobName> sind lediglich die Namen des Blobs und Containers, auf welche durch die Anfrage zugegriffen werden soll. <TableName> spezifiziert die Tabelle, auf die eine Anfrage gestartet werden soll und *jQuery* enthält die Anfrage-Operation, die ausgeführt werden soll. <QueueName> ist der Name des Queue-Eintrages, der erreicht werden soll.

3.7 Configuration File

Ein weiterer wichtiger Bestandteil von Windows Azure sind die Configuration Files, diese sollen ein effektives Management ermöglichen, welches benötigt wird, um die Vorteile vom Cloud Computing zu nutzen. Jede Anwendung hat eine Configuration File und kann diese definieren, und Einstellungen damit deklarieren. Z.B. kann die Anzahl der zu verwendenden Instanzen festgelegt werden und es kann eingestellt werden, wie viele Instanzen einer Web Role und/oder Worker Role zugewiesen werden. Die Azure Fabric kontrolliert und verwaltet die Einstellungen auf richtige Ausführung und Zuordnung der Ressourcen zu den Instanzen etc.

4 SQL Azure

4.1 Übersicht

SQL Azure ist ein weiterer, wichtiger Bestandteil der Windows Azure Platform, wobei es in erster Linie um das Verwalten von Daten geht. Ziel ist es, mit SQL

Azure ein Set an Cloud-basierten Diensten anzubieten, die das Speichern und Arbeiten mit vielen Daten ermöglichen. Wie in Abbildung 5 zu sehen ist werden im Moment zwei Dienste angeboten, zum einen die SQL Azure Database und den “Huron“ Data Sync, dabei laufen bzw. interagieren beide mit den Datenbankservern in den Rechenzentren von Microsoft. In Zukunft sollen noch weitere Dienste folgen, diese können z.B. Dienste zur Datenanalyse sein.

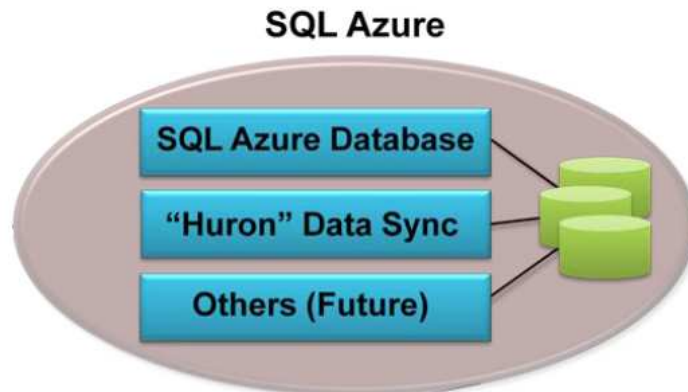


Abbildung 5. SQL Azure bietet datenorientierte Einrichtungen in der Cloud an. [1]

4.2 SQL Azure Database

Ursprünglich wurde der Dienst als SQL Data Service bezeichnet, bevor dieser den Namen SQL Azure Database erhielt. In erster Linie wird das Speichern von relationalen und anderen Datentypen ermöglicht, wobei Azure Database ein Datenbankmanagementsystem in der Cloud darstellt (DBMS). Dabei wird vom technischen Standpunkt aus auf Microsofts SQL Server aufgebaut. Zu Anfang des CTP stellte die Azure Database noch keine traditionelle, relationale Sicht auf die Daten dar. Dies wurde von Microsoft geändert, nachdem aufgrund von Feedback einiger Entwickler sich herausstellte, dass das Bedürfnis nach einer relationaler Datenbank definitiv bestand. Aktuell ist die SQL Azure Database so aufgebaut, dass sie auch relationale Daten unterstützt, in dem eine Microsoft SQL Server Umgebung (in abgespeckter Form) angeboten wird (Indizes, Sichten, Stored Procedures, Triggers, ...). Natürlich ist eine Datenbank nur dann sinnvoll, wenn auch auf die Daten zugegriffen werden kann, dies geschieht bei der SQL Azure Database über ein spezielles Protokoll, dem Tabular Data Stream (TDS) [1]. Dies ist das gleiche Protokoll, wie es auch beim Zugriff auf eine lokale SQL Server Datenbank genutzt wird. Um den Zugriff zu steuern, kann auf jede

SQL Server Client Bibliothek zurückgegriffen werden, wie ADO.NET oder andere, übliche "Data Access" Schnittstellen (ODBC). Microsoft übernimmt dabei die Verwaltung der Operationen und Datenablage, die genutzt wird, um mit der Datenbank zu arbeiten, somit bleibt dies dem Kunden erspart. Und dadurch werden auch keine physikalischen, administrativen Funktionen zur Verfügung gestellt. Ein wichtiger Punkt ist noch zu erwähnen, dass eine abgesetzte Abfrage ("query") nur für eine begrenzte Zeit aktiv ist und ein "query" kann auch nur eine vordefinierte Anzahl an Ressourcen nutzen, was bei Operationen mit großen Datenmengen berücksichtigt werden sollte. Eine interessante Frage ist es, welche Speichermengen die Datenbanken zur Verfügung stellen? Es gibt zwei unterschiedliche Größen, zum einen 1GB große Datenbanken, zum anderen 10GB große Datenbanken. Anwendungen, die mehr Speicherplatz benötigen, haben die Möglichkeit multiple Datenbanken zu nutzen, was allerdings logistische Herausforderungen der Datenaufteilung an die Anwendung stellt. Dies bedeutet, dass Anwendungen ihre Daten geschickt aufteilen müssen, und diese dann auf mehrere Datenbanken verteilen, da mit einem "query" nicht auf alle Datenbanken zugegriffen werden kann und somit auch nicht mehr alle Daten erreicht werden. Im Gegensatz dazu können die Abfragen wie gewohnt auf eine "single" Datenbank ausgeführt werden. Die Abbildung 6 veranschaulicht die Idee der "multiplen" Datenbanken.

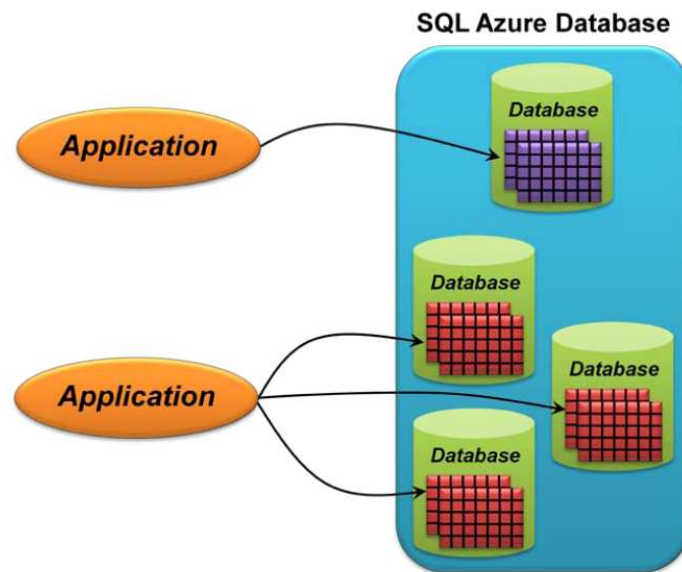


Abbildung 6. Eine Anwendung kann eine "Single" oder eine "Multiple" Datenbank benutzen. [1]

Wie man sehen kann, ist es der Anwendung möglich, seine Daten auf mehrere Datenbanken zu verteilen. Dies bedeutet, dass die Anwendung genau darauf achten muss, wie sie ihre Daten teilt, um diese zu speichern. Physikalisch werden die Daten genauso wie beim Azure Storage Service dreifach redundant gespeichert, um eben die Ausfallsicherheit zu gewähren.

4.3 “Huron“ Data Sync

In der IT-Landschaft sind Daten oftmals physikalisch über mehrere, verschiedene Datenbanken verteilt, genauso wie Sicherheitskopien von den Daten. Dadurch ist es notwendig, dass die Daten auf den Datenbanken synchronisiert werden müssen. Und genau diese Anforderung soll die “Huron“ Data Sync Technologie erfüllen, indem sie die Daten von Datenbanken (in der Regel relationale DB) synchronisiert. Dabei basiert dieser Dienst auf dem Microsoft Sync Framework [8] und der SQL Azure Database. Abbildung 7 zeigt eine sehr gute “High-Level-Sicht“ über die Verwendung von “Huron“ Data Sync. “Huron“ Data Sync dient als Synchronisationsschnittstelle zwischen den verschiedenen Datenbanken oder Datenbankmanagementsystemen, z.B. können SQL Azure Datenbanken, SQL Server Datenbanken oder andere Datenbanken miteinander verknüpft werden. Ein Arbeitsablauf geht nach einer bestimmten Arbeitsweise vor, als erstes werden die in einer Datenbank (SQL Server) geänderten Daten mit dem “Huron“ Data Sync auf der dahinter liegenden DB (SQL Azure Database) synchronisiert. Als nächstes werden dann die Daten mit den anderen am Data Sync angeschlossenen Datenbanken synchronisiert, so dass alle DBMS auf dem neuesten Stand sind. Um die Benutzerfreundlichkeit zu steigern, bietet der Datendienst eine grafische Schnittstelle an, mit der Einstellungen vorgenommen werden können, um zu definieren, welche Daten mit welchen DBMS synchronisiert werden sollen, und man kann festlegen, wie mit Konflikten bei der Synchronisation umgegangen werden soll.

5 .NET Services

5.1 Übersicht

Mit .NET Service will Microsoft Cloud-basierte Infrastruktur Dienste anbieten, die sowohl von Anwendungen in der Cloud als auch von lokalen Anwendungen verwendet werden können. Dazu werden aktuell zwei Komponenten bereitgestellt, den Access Control Service und den Service Bus. Access Control stellt eine Cloud-basierte Lösung dar, um Benutzerrechteverwaltung für Anwendungen zu realisieren, und unterstützt somit die Verwaltung von Benutzerrechte über verschiedene Unternehmen hinweg. Es wird auch ermöglicht, Rechte zu bearbeiten, neu zu verteilen etc. Der Dienst des Service Bus ermöglicht es den Anwendungen, ihre Dienste anzubieten für andere Anwendungen, ohne dass für die fremden Anwendungen der Zugriff auf die interne Netzwerkstruktur gewährt

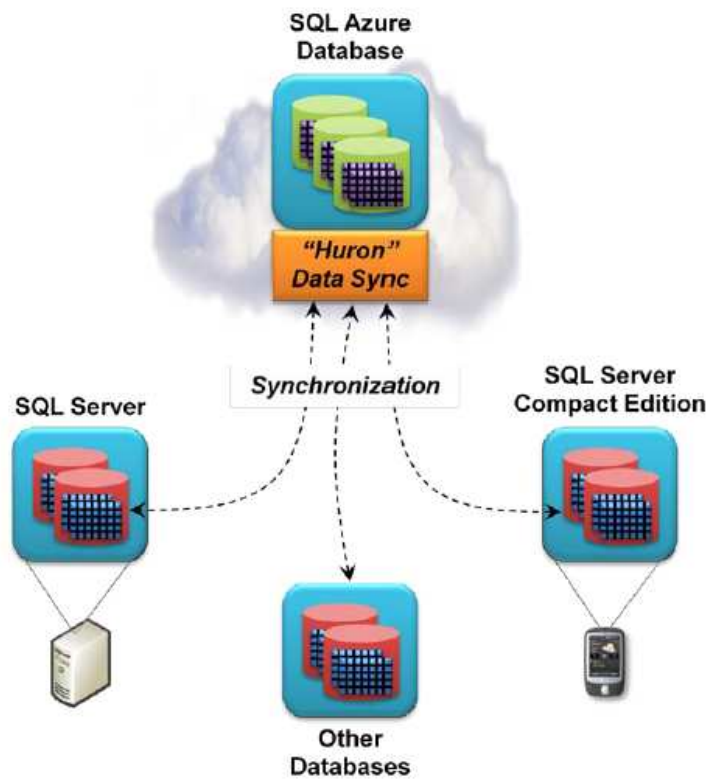


Abbildung 7. "Huron" Data Sync benutzt das MS Sync Framework, um Daten über Datenbanken hinweg zu synchronisieren. [1]

werden muss. Es wird den Anwendungen erlaubt, "Endpoints" zu definieren. Jeder "Endpoint" hat eine eigene, eindeutige URI, die von anderen Anwendungen gefunden und erreicht werden können, um den angebotenen Service zu nutzen. Um diese allgemeine Beschreibung der beiden .NET Services zu vertiefen, wird in den nächsten beiden Abschnitten die Verwendung und Arbeitsweise der Services ein wenig detaillierter beschrieben.

5.2 Access Control Service

Ein genauerer Blick auf den Access Control Service zeigt, dass dieser einen Security Token Service (STS) [1] anbietet in der Cloud. STS ist eine Software, die Token erstellen kann, und diese Tokens nutzen die Security Assertion Markup Language (SAML) [1]. Die Tokens enthalten sogenannte "Claims", in diesen sind Informationen über die Benutzer und deren Rechte gespeichert. Mit Hilfe der Tokens können die Anwendungen die Zugriffsrechte für die Benutzer deklarieren. Der Besitzer einer Anwendung definiert bestimmte Regeln, die festlegen, welche Dienste ein Benutzer der Anwendung nutzen darf oder nicht. Auf diesen Regeln basiert der STS des Access Control Service. Abbildung 8 zeigt ein Beispielszenario über die Arbeitsweise des Access Control Service.

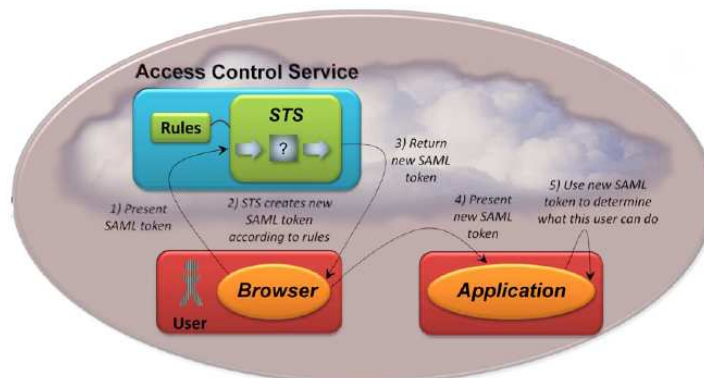


Abbildung 8. Access Control Service bietet regelbasierte Rechteverwaltung. [1]

Wie in der Abbildung 8 zu sehen ist, gibt es drei Akteure, einen "User" (Browser), die "Application" und der Access Control Service mit dem STS und "Rules". Im ersten Schritt schickt der Benutzer ein SAML Token zum Access Control Service, genauer an den STS, welcher die Unterschrift des Token validiert. Der Service erstellt und unterzeichnet ein neues SAML Token, welches die "Claims" mit den vom Besitzer der Anwendung vorgegebenen Regeln beinhaltet. Danach wird das neue Token vom STS an den Benutzer wieder zurückgeschickt. Dieser nutzt das neue Token, um sich bei der Anwendung anzumelden

und auf die Dienste zuzugreifen. Die Anwendung validiert die Unterschrift des Token und nutzt die mitgelieferten “Claims“ um festzulegen, welche Rechte der Benutzer erhält. Damit der Service reibungslos genutzt werden kann, bedarf es der Gewährleistung einer Kommunikation zwischen den Komponenten. Jegliche Kommunikation mit dem Access Control Service erfolgt dabei über Standard Protokolle wie z.B. WS-Trust und WS-Federation [1]. Dadurch wird der Service für jede Art von Anwendung oder Plattform, die die Protokolle verwenden, erreichbar und nutzbar. Um die Regeln zu definieren, wird eine Browser-basierte GUI zur Verfügung gestellt. Möchte man die Regeldefinition programmatisch erledigen, wird eine API bereitgestellt.

5.3 Service Bus

Mit dem Service Bus versucht Microsoft die Fragestellung “Wie können Anwendungen von Unternehmen übergreifend auf lokale Dienste zugreifen?“ zu lösen. Dabei gibt es ein paar Schwierigkeiten zu überwinden, die größten sind dabei das Erreichen der Anwendungen, da meistens Firewalls oder NAT’s den “Weg“ erschweren. In Abbildung 9 ist übersichtlich die Funktionsweise des Service Bus dargestellt. Die Arbeitsweise kann man wie folgt beschreiben. Zunächst registriert eine Anwendung einen oder auch mehrere “Endpoints“, für die Dienste, die angeboten werden sollen, beim Service Bus, diese werden dann in der “Registry“ des Service Bus gespeichert. Jedes Unternehmen bzw. Anwendung bekommt eine eindeutige URI zugewiesen, wodurch die Services spezifiziert adressiert werden können. Ein wichtiger Punkt ist, dass die lokale Anwendung eine Verbindung zum Service Bus herstellen muss, die dann vom Service Bus aufrechterhalten wird. Dadurch werden die Probleme mit einer Firewall oder NAT (Network Address Translation) umgangen.

Wie kann eine externe Anwendung nun auf einen Service oder Anwendung zugreifen, die am Service Bus angemeldet ist? Als erstes stellt die Anwendung eine Anfrage an die Service Bus “Registry“, diese verwendet das “Atom Publishing Protocol“ [1] und liefert ein “Atom Push Service Document“ [1] zurück mit den Referenzen auf die gespeicherten Service “Endpoints“. Nun kann die externe Anwendung die benötigten Services aufrufen und erhält Zugang zu diesen und kann mit diesen arbeiten. Service Bus erlaubt auch die Kommunikation über Queues des Azure Storage Services. Dadurch gibt es die Möglichkeit, Nachrichten zu senden auch wenn die angesprochene Anwendung nicht erreichbar ist. Die Nachricht kann bis zu einer Woche gespeichert werden, um auf eine Antwort der Anwendung zu warten. Auch die Sicherheit wird durch den Service Bus erhöht, da nur die IP-Adresse des Service Bus sichtbar ist, wo der “Endpoint“ einer Anwendung registriert ist und nicht die lokale IP der Anwendung. Die beiden Dienste, Access Control Service und Service Bus können auch sehr gut miteinander kombiniert werden, so dass der Service Bus nur Tokens akzeptiert, die durch den STS des Access Control Service erstellt wurden, was wiederum eine zusätzliche Sicherheit bedeutet.

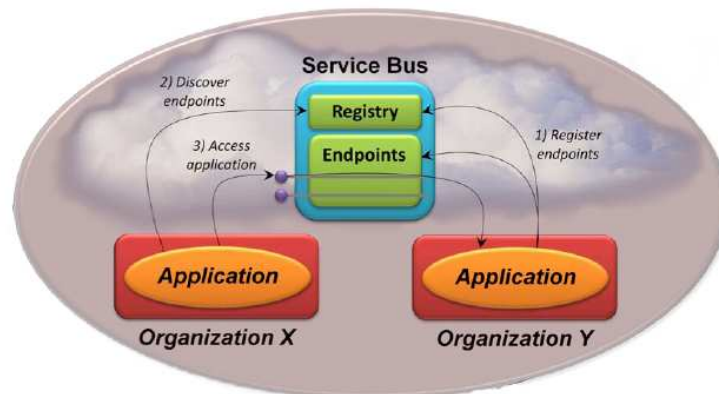


Abbildung 9. Service Bus erlaubt es Anwendungen “Endpoints“ zu registrieren, damit andere Anwendungen Zugriff auf die “Endpoints“ und damit verbundenen Dienste haben. [1]

6 Windows Azure für Java-Entwickler

Wie schon angedeutet ist die Windows Azure Plattform nicht nur für Windows Anwendungen gedacht, sondern es soll auch Interoperabilität gewährleistet werden können. Um dies zu erfüllen, können auch andere Technologien die Komponenten nutzen, sofern diese die bereitgestellten Webservice-Schnittstellen benutzen können. Dies trifft auf die Programmiersprache Java zu und es bieten sich drei verschiedene Möglichkeiten, vor allem Windows Azure zu nutzen:

- Aufruf von Azure Services aus einer Java-Anwendung heraus
- Integration von Java-Services über .NET Services aus Azure
- Betrieb von Java-Anwendungen auf Windows Azure

Die folgenden Abschnitte sollen einen kleinen Einblick geben, in die drei Arten Windows Azure mit Java zu nutzen.

6.1 Nutzung von Azure Services in Java-Anwendungen

Ein Wesensmerkmal der Serviceorientierung ist die Bereitstellung von Schnittstellen, über die diese Funktionalitäten verfügbar sind. Dabei sind die innere Funktionsweise und Details der Implementierung nicht genau bekannt, aber auch nicht von Interesse. Und genau nach diesem Prinzip arbeitet Windows Azure, welches über Webservice-Schnittstellen Zugriff auf Windows Azure und die Services gewährt. Am einfachsten kann man die Verwendung an einem kleinen Beispiel aufzeigen. Ein Java-Programm soll Nachrichten, die über Azure Storage

Queues verschickt werden, empfangen, verarbeiten und löschen können. Nachrichten werden dabei nicht sofort aus der Queue gelöscht, sondern erst nach erfolgreicher Bearbeitung der Anwendung. Um die Sicherheit zu gewährleisten, müssen die Anfragen bzw. Nachrichten verschlüsselt werden. Der dazu benötigte Schlüssel wird beim Erstellen eines Windows Azure Accounts generiert.

6.2 Java SDK für .NET Services

Die .NET Services schaffen mit dem Service Bus die Möglichkeit zu gesicherten Serviceaufrufen, wie schon im Kapitel über den Service Bus beschrieben. Als Ergebnis erhält man zwei Services, die über Outbound-Calls eine Kommunikation miteinander initiiert haben. Microsofts Partner Schakra Inc. bietet ein Java Software Development Kit (SDK) für die .NET Services an. Dieses setzt auf Metro (ein Open-Source-Webservice Stack basierend auf Glassfish von Sun Microsystems) [3] auf. Das SDK bietet Bibliotheken an, die es erlauben Java-basierte Webservices über .NET Services in der Cloud anzubieten bzw. aufrufbar zu machen. Eine weitere Nutzungsmöglichkeit ist es aus Java-Clients heraus entsprechende Cloud Services aufzurufen.

6.3 Betrieb von Java-Anwendungen auf Windows Azure

Im März 2009 wurde eine wichtige Funktion für Windows Azure hinzugefügt, so dass es auch möglich ist, nativen Code auf Windows Azure auszuführen. Dadurch sollte jeder beliebige Code ausführbar sein, der auch auf Windows Server laufen kann, und dazu gehört natürlich auch Java. Um z.B. Java Code auf Windows Azure laufen zu lassen, muss in der dazugehörigen Configuration File der Anwendung oder auch Web bzw. Worker Role die Ausführung von nativem Code aktiviert werden. Ein entsprechendes Code Listing sieht wie folgt aus:

```
... <WorkerRole name="WorkerRole" enableNativeCodeExecution="true" /> ...
```

Es gibt auch die Möglichkeit ein Java-Laufzeitsystem zu instanzieren um Java-Anwendungen auszuführen. Dazu muss von einer Worker Role aus die Java-Umgebung instanziiert werden und dadurch wird das Java-Laufzeitsystem als neuer Prozess gestartet. Die Ein- und Ausgabeströme werden auf Windows Azure umgeleitet, damit Azure die Verwaltung übernehmen kann. Im Laufzeitsystem muss dann nur noch eine Java-Klasse gestartet werden, dies kann beispielsweise die in Java geschriebene Worker-Klasse sein, in welcher dann die main()-Methode aufgerufen werden kann.

7 Preismodelle

Um die Windows Azure Platform und deren Komponenten kommerziell zu nutzen, bietet Microsoft drei Preismodelle an:

- Verbrauchsabhängige Abrechnung (“consumption-based“): Dieses Preismodell unterteilt sich in weitere drei Teilbereiche, die noch genauer beschrieben werden.
- Abonnenten-basierte Abrechnung (“subscription-based“): Mit dieser Abrechnungsvariante ist es möglich Abonnements zu beziehen, bei denen ein bestimmter Betrag in Rechnung gestellt wird und dann eine gewisse Anzahl an Diensten zur Verfügung steht über einen bestimmten Zeitraum.
- Abrechnung im Rahmen eines Volumenlizenzvertrages (“volume licensing“): Damit möchte Microsoft, genauso wie beispielsweise bei Betriebssystemen, dem Kunden eine Volumenlizenz anbieten über diese dann der Kunde die Windows Azure Platform verwenden kann.

7.1 “Pay as you go“ [6]

Wie schon erwähnt unterteilt sich die verbrauchsabhängige Abrechnung in drei Teile, die die Preise für Windows Azure, SQL Azure und die Windows Azure Services festlegen.

1. Windows Azure Preistabelle 1

Tabelle 1. Kosten fuer Windows Azure

Service	Kosten	Notiz
Compute	\$0.12/h	Berechnung wenn Anwendung gestartet
Storage	\$0.15/GB/Monat	Anteilige Berechnung
Storage transaction	\$0.01/10000 (requests)	
Data transfer	\$0.10in/\$0.15out/GB	

2. SQL Azure Preistabelle 2
Die SLA für SQL Azure liegt laut Microsoft bei einer monatlichen Verfügbarkeit von 99,9% während eines Monats.
3. Windows Azure Service Preistabelle 3
Abgerechnet wird hierbei in 100K-Blöcken bei den Messages plus Kosten für den Datentransfer.

8 Fazit/Schlußwort

Die Ausarbeitung soll eine einführende Beschreibung der Welt der Windows Azure Platform darstellen. Sie soll einen Einblick in die verschiedenen Dienste

Tabelle 2. Kosten fuer SQL Azure

Service	Kosten	Notiz
Web Edition	\$9.99/Monat	Bis zu 1GB relationaler DB
Business Edition	\$99.99/Monat	Bis zu 10GB relationaler DB
Data transfer	\$0.10in/\$0.15out/GB	

Tabelle 3. Kosten fuer SQL Azure

Service	Kosten	Notiz
Messages	\$0.15/100K (operations)	“messages“, “transactions“ und “operations“
Data transfer	\$0.10in/\$0.15out/GB	

und Komponenten der Plattform bieten und den Aufbau und Funktionsweise der einzelnen Komponenten erläutern. Dabei stellt die Plattform eine Reihe von Cloud Services zur Verfügung, die über Standardschnittstellen aufgerufen werden können, somit bietet die Plattform vielfältige Möglichkeiten, von unterschiedlichsten Anwendungen genutzt zu werden. Die einzelnen Komponenten decken dabei eine Angebotsvielfalt im Bereich des Cloud Computing ab. Windows Azure bietet eine Windows-basierte Rechen- und Speicherumgebung in der Cloud an. SQL Azure bietet in der Cloud ein Datenbankmanagementsystem mit der SQL Azure Database und Datensynchronisation mit dem “Huron“ Data Sync an. Aber noch mehr Cloud-basierte Datendienste sind geplant. .NET Services bietet eine Cloud-basierte Infrastruktur für Cloud-Anwendungen oder lokale Anwendungen. Wie bei vielen Cloud-Lösungen soll auch hier dem Kunden die Anschaffung und Verwaltung von kostspieligen Systemen erspart werden, indem die Kunden die Ressourcen der Windows Azure Plattform benutzen können. Der SLA für die Kunden verspricht, dass die Anwendungen immer laufen, Daten dauerhaft verfügbar sind und Skalierbarkeit gewährleistet wird. Allerdings kam es während des CTP auch zu einem Ausfall von Windows Azure. Die Windows Azure Plattform soll als Konkurrent zu Google, Yahoo, Salesforce.com und Amazon.com auf den Markt kommen, wobei Microsoft die Plattform auch als Basis für ihre SaaS-Angebote, wie beispielsweise “Microsoft Live“ und “Microsoft Online“, dienen soll. Wie schon in der Einleitung erwähnt, hat Ray Ozzie, “Microsofts Chief Software Architect“ die ehrgeizige Vision geäußert, Endgeräte wie Mobiltelefone, Computer und Fernseher über die Cloud zu verbinden. Inwieweit dies umsetzbar ist, wird sich noch zeigen, aber eines dürfte klar sein, dass Microsoft ihr Softwarevertrieb in Zukunft verstärkt über die Plattform abwickeln wird.

Literatur

1. Whitepaper: Introducing The Windows Azure Platform - David Chappell - August 2009
2. dot.NET Magazin - Ausgabe 10.2009 - Artikel: Wolkenloser Himmel - Entwicklung für Windows Azure

3. Java SPEKTRUM - Ausgabe 6 - Dezember'09/Januar'10 - Artikel: Microsoft Azure für Java-Entwickler
4. Internetauftritt der Microsoft Windows Azure Plattform
<http://www.microsoft.com/windowsazure>
5. Deutsche Informationsseite über die Windows Azure Plattform
<http://www.microsoft.com/germany/net/WindowsAzure>
6. Preisinformationen zu der Windows Azure Plattform
<http://www.microsoft.com/windowsazure/pricing>
7. Zeitungsartikel der Seattle Times über die Professional Developers Conference von Microsoft:
"Microsoft PDC09: Ray Ozzie puts the clouds in my coffee"
<http://seattletimes.nwsourc.com>
8. Microsoft Sync Framework
<http://msdn.microsoft.com/de-de/library/bb902854.aspx>
9. ADO.NET
<http://msdn.microsoft.com/de-de/library/bb979090.aspx>
10. LINQ (Language INtegrated Query)
<http://de.wikipedia.org/wiki/LINQ>
11. Windows Communication Foundation (WCF)
<http://msdn.microsoft.com/de-de/netframework/aa663324.aspx>
12. Microsofts Hyper-V
<http://de.wikipedia.org/wiki/Hyper-V>
13. ASP.NET
<http://www.asp.net>
14. Internet Information Service (IIS) von Microsoft
<http://www.iis.net/>
15. Wikipedia - Windows Azure Plattform
<http://www.wikipedia.de>