

10 Aktueller Stand der Plattform (Version 1.3)

Allgemeine Änderungen der Plattform	2
Aktueller Stand der einzelnen Dienste	6
Neue Dienste	23
Zusammenfassung	25

Der Betrieb von Software in der Cloud zeichnet sich unter anderem durch hohe Skalierbarkeit, hohe Verfügbarkeit und einem automatisierten Management der Plattformkomponenten aus. Zu Letzterem gehört die Möglichkeit Microsofts, in Abstimmung mit den Nutzern der Plattform laufend Plattformaktualisierungen und –patches einspielen zu können, ohne dass dies den laufenden Betrieb der Cloud Services beeinträchtigt. Die Windows Azure Plattform ist deshalb permanent Änderungen unterworfen, was es zwangsläufig erschwert, ein Buch zur Plattform auf dem aktuellen Stand zu halten.

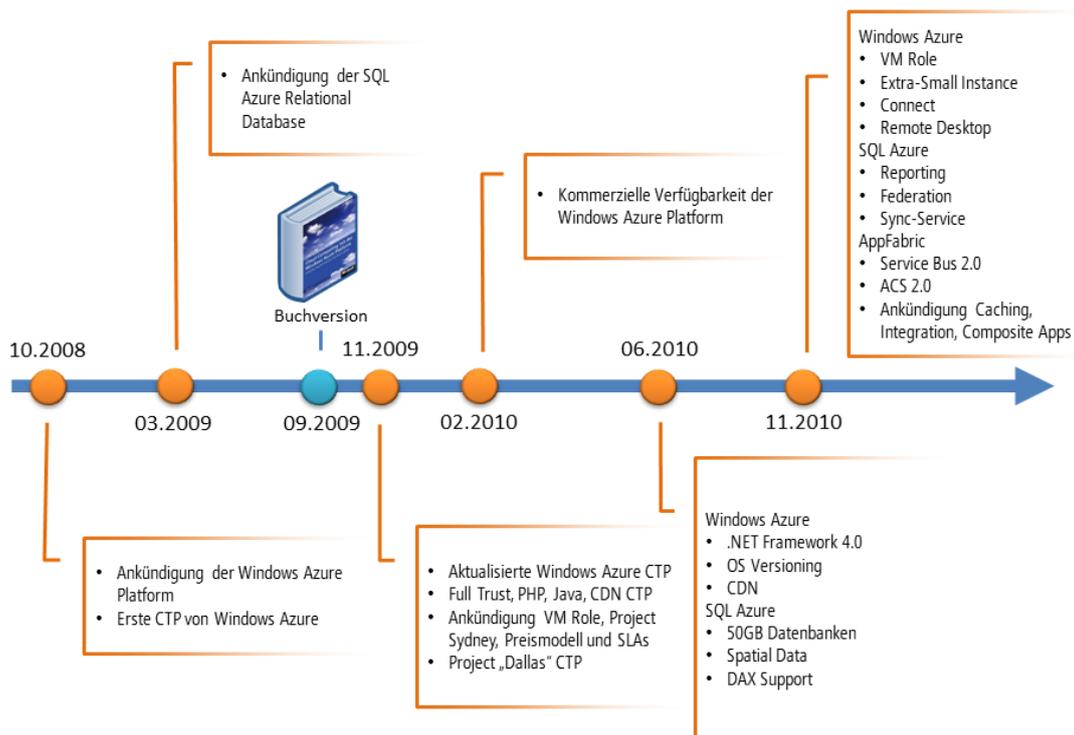


Abbildung 10.1: Weiterentwicklungen der Windows Azure Plattform

Seit der Fertigstellung des Buchmanuskriptes hat es eine Reihe von Änderungen und Erweiterungen an der Plattform gegeben, die in diesem Kapitel überblicksartig beschrieben werden. Zusammen mit der gedruckten Version des Buches bildet es ein Gesamtwerk, welches einen Überblick über die aktuelle Version 1.3 (Stand: März 2011) der Windows Azure Plattform gibt.

Abbildung 10.1 zeigt die wichtigsten Evolutionsschritte, die die Plattform seit Ankündigung im Oktober 2008 durchlaufen hat. Damit wird auch deutlich, welche Änderungen im Buch nicht mehr berücksichtigt werden konnten.

Allgemeine Änderungen der Plattform

Die Änderungen, die die Plattform erfahren hat, sind vielschichtig: Einige betreffen die Gesamtplattform, einige den Funktionsumfang von Windows Azure bzw. Azure Services. In **Abbildung 10.2** ist die Windows Azure Plattform im Versionsstand 1.3 im Überblick zu sehen.

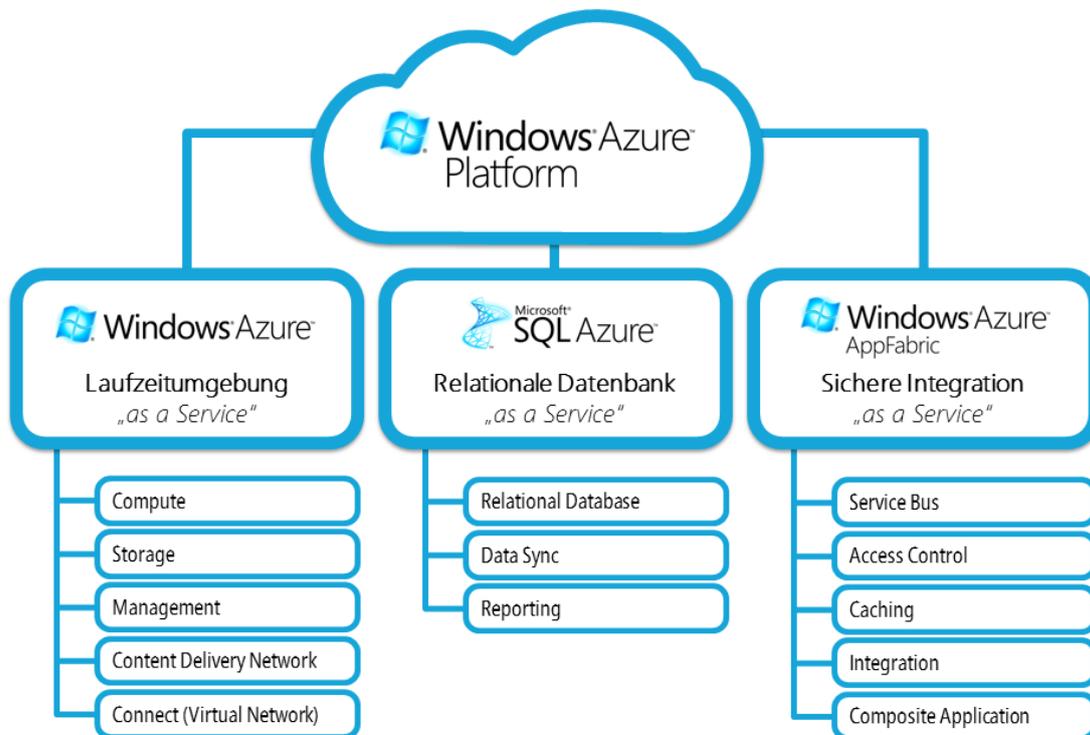


Abbildung 10.2: Die Windows Azure Plattform – Version 1.3

Hier fallen ein paar Änderungen gegenüber der im Buch behandelten CTP-Version unmittelbar auf:

- *.NET Services* sind nicht mehr zu sehen. Diese Dienstgruppe hat den Namen *Windows Azure AppFabric* erhalten.
- *Live Services* sind formal nicht mehr Bestandteil der Windows Azure Plattform. Zwar bestehen diese Dienste weiter und werden auch in den gleichen Rechenzentren wie Azure betrieben, sie werden aber nicht mehr unter der Marke *Azure* geführt, sondern als eigene Marke weiterentwickelt.
- *Dynamics CRM Services* und *SharePoint Services* sind formal ebenfalls nicht mehr Bestandteil der Windows Azure Plattform. Diese beiden Dienstgruppen werden im Rahmen der *Microsoft Online*

Services weiterentwickelt. Die beiden Produkte *Dynamics CRM Online* und *SharePoint Online* (verfügbar seit März 2009) werden neben der Möglichkeit der Anpassung über die Programmoberfläche bzw. SharePoint Designer erweiterbar sein. Damit erhalten diese Online Services die Funktionalität, die bisher über die beiden Azure Services geplant war.

Damit besteht die Windows Azure Plattform nun aus den drei Dienstgruppen Windows Azure, SQL Azure und Windows Azure AppFabric. In jeder der drei Gruppen wurden seit Produktstart Änderungen bekanntgegeben bzw. Erweiterungen angekündigt, die in den jeweiligen Folgeabschnitten beschrieben werden.

Preismodelle für die Bestandteile der Plattform

Mit der Produktivsetzung, bei der eine Abrechnung der genutzten Ressourcen erfolgen soll, sind jetzt auch entsprechende Preismodelle verfügbar. Diese orientieren sich grundsätzlich am Verbrauch der einzelnen Cloud Services, legen aber jeweils unterschiedliche Abrechnungseinheiten zugrunde. So wird die Rechenleistung von Windows Azure pro Bereithaltungsstunde, Windows Azure Storage nach Datenmenge und Speicheroperationen, SQL Azure nach Abrechnungsmonat und die AppFabric nach der Zahl der Transaktionen bzw. Verbindungen abgerechnet. **Tabelle 10.1** gibt einen Überblick über die Kosten der einzelnen Dienste.

Dienst	Abrechnungseinheit	Kosten	
Windows Azure Compute	Bereithaltungsstunde	Extra Small VM:	0,0852 €
		Small VM:	0,0355 €
		Medium VM:	0,0710 €
		Large VM:	0,1420 €
		XLarge VM:	0,2840 €
Windows Azure Storage	Datenmenge und Speichertransaktionen	GB pro Monat:	0,1064 €
		10.000 Transaktionen:	0,0071 €
Windows Azure CDN	Datenmenge und Speichertransaktionen	Transferierte GB:	0,1064 €
		10.000 Transaktionen:	0,0071 €
SQL Azure	Abrechnungsmonat	Web Edition (maximal 1 GB):	7,0850 €
		Business Edition (maximal 10 GB):	70,9130 €
AppFabric Access Control	Anzahl Transaktionen	100.000 Transaktionen:	1,4114 €
AppFabric Service Bus	Anzahl Verbindungen	Verbindung (im »pay-as-you-go«-Modell):	2,8298 €
		Paket von 5 Verbindungen:	7,0566 €

Tabelle 10.1: Kosten für die Nutzung der einzelnen Dienste der Windows Azure Plattform (Auszug; Stand 03/2011)

Zu beachten ist, dass bei Windows Azure Kosten anfallen, sobald eine Anwendung auf Windows Azure installiert ist und nicht erst, wenn auch tatsächlich Zugriffe auf die Anwendung erfolgen. Für die Betreiber von Anwendungen besteht somit ein Optimierungspotenzial aus der Steuerung der Anzahl aktiver Instanzen einer Anwendung. Die Anzahl sollte immer so klein wie möglich (um Kosten zu sparen) und so groß wie nötig (um Antwortzeiten für Anwender zu minimieren) gehalten werden. Ungenutzte Instanzen sollten (idealerweise automatisiert) deaktiviert werden. **Tabelle 10.5** gibt einen Überblick über Kosten und Leistungsparameter der einzelnen VM Größen.

Unabhängig vom genutzten Dienst kommen zu den in **Tabelle 10.1** aufgelisteten Kosten noch Beträge für den Datentransfer, der in Microsofts-Rechenzentren hinein bzw. aus diesen heraus geht. Diese sind in **Tabelle 10.2** aufgeführt.

Transferrichtung	Abrechnungseinheit	Kosten	
In ein Azure Rechenzentrum hinein	Datenmenge [pro GB]	Nord Amerika und Europa:	0,0710 €
		Asien und Pazifikregion:	0,0710 €
Aus einem Azure Rechenzentrum heraus	Datenmenge [pro GB]	Nord Amerika und Europa:	0,1064 €
		Asien und Pazifikregion:	0,1419 €

Tabelle 10.2: Kosten für die Datentransfers über die Grenzen von Microsofts Rechenzentren (Stand: 03/2011)

Der Datentransfer innerhalb der Rechenzentren ist kostenfrei. Kommuniziert eine auf Windows Azure betriebene Anwendung mit einer auf SQL Azure betriebenen Datenbank (und Anwendung und Datenbank liegen im gleichen Rechenzentrum), fallen für den Datenaustausch keine Kosten an.

Die Nutzung von Azure auf Basis der oben beschriebenen Preismodelle ist die flexibelste Variante. Es fallen genau dann Kosten an, wenn die entsprechende Ressource genutzt wird. Ohne Nutzung, keine Kosten. Bezogen auf die Abrechnungseinheit ist dies allerdings auch die teuerste Variante (die Flexibilität wird also über einen höheren Preis erkauft). Neben dieser sogenannten »pay as you go«-Variante gibt es weitere Bezugsmöglichkeiten, bei denen für eine vereinbarte Abnahmemenge Rabatte gewährt werden:

- *Windows Azure Platform Development Accelerator Core*
Befristet auf sechs Monate kann hier ein festgelegtes Kontingent von Windows Azure Compute, Windows Azure Storage, Datentransfer, Service-Bus-Verbindungen und Access-Control-Transaktionen mit 54% Kostenrabatt genutzt werden.
- *Windows Azure Platform Development Accelerator Extended*
Zusätzlich zu den Dienstkontingenten des Development Accelerator Core umfasst dieses Paket auch eine SQL Azure-Datenbank (Business Edition) und bietet 52% Kostenrabatt gegenüber dem »pay as you go«-Preis.
- *Windows Azure Platform Introductory Special*
In einem zeitlich sehr befristeten Rahmen können bei diesem Angebot einige Azure Services zu Test- und Evaluierungszwecken kostenfrei genutzt werden.

Weitere Informationen zu diesen und weiteren Angeboten können über den URL <http://www.microsoft.com/windowsazure/offers/> abgerufen werden.

Service Level der Plattform

Darüber hinaus erfolgt die Nutzung der Plattform unter verbindlichen Service Level Agreements (SLAs). Ein Ausschnitt aus diesen SLAs ist in **Tabelle 10.3** zu sehen.

Messgröße	Beschreibung	Service Level
Anbindung von Cloud Services	Anbindung von Diensten, die auf Windows Azure betrieben werden, ans Internet	>99,95%
Überwachung von Instanzen	Instanzen von Rollen werden laufend überwacht und bei Bedarf automatisch neu gestartet	>99,9%
Verfügbarkeit Speicher	Der Windows Azure Storage ist von außen zugreifbar und beantwortet Speicherzugriffe erfolgreich	>99,9%
Verfügbarkeit Datenbank	SQL Azure ist von außen zugreifbar. Alle Datenbanken werden laufend überwacht.	>99,9%

Verfügbarkeit Service Bus und Zugriffskontrolle	Endpunkte, die beim Service Bus bzw. Access Control Service registriert sind, sind von außen aus zugreifbar. Nachrichten können über diese Dienste ausgetauscht werden.	>99,9%
---	---	--------

Tabelle 10.3: Ausschnitt aus den Service Level Agreements der Windows Azure Platform (Angaben ohne Gewähr)

Eine Vereinbarung individueller Service Level (insbesondere höhere Service Level) ist grundsätzlich nicht möglich. Die Dienste werden in einer automatisierten Form angeboten, die es nicht zuließe, für einzelne Anwender Ausnahmeregelungen zu treffen. Werden andere als die angebotenen Service Level benötigt, verweist Microsoft an die große Zahl an Hosting-Partnern. Diese können auf Basis von Microsofts Server Plattform ähnliche Dienste bereitstellen und dabei in der Regel individuelle Vereinbarung von Service Leveln zulassen.

Auswirkungen auf die Buchkapitel im Überblick

Die Änderungen der Plattform haben zum Teil große Auswirkungen auf die entsprechenden Kapitel des Buches. **Tabelle 10.4** listet die wichtigsten Änderungen der Services auf.

Kapitel	Platfordienst	Wichtigste Änderungen
Kapitel 3	Windows Azure Compute	Verschiedene Größen virtueller Maschinen für die Service Rollen Full-Trust und Administratorzugang zu den virtuellen Maschinen Full-IIS in Web Roles Direkter Nachrichtenaustausch zwischen Rollen Möglichkeit der Konfiguration von Kommunikationsports Erweiterte Logging- und Diagnosemöglichkeiten Verwaltung von Diensten über eine Management API Erweiterte Interoperabilität (MySQL, Java, Ruby, Apache etc.) Versionierung des Gast-Betriebssystems Unterstützung von .NET Framework 4.0 VM Rollen (Hosting eigener virtueller Maschinen) Startup-Tasks Plugins
	Windows Azure Storage	Drives als virtuelle NTFS-Festplatten für Windows Azure Offizielle Storage Client Library Erweiterte Funktionalitäten der Storage Services
	Windows Azure CDN	Verfügbarkeit des Content Delivery Networks (CDN)
	Windows Azure Connect	Verfügbarkeit von Windows Azure Connect
Kapitel 4	Live Services	Live Services sind nicht mehr Bestandteil der Plattform
Kapitel 5	.NET Services	Umbenennung in <i>Windows Azure AppFabric</i>
	AppFabric Service Bus	Entfernung von Routern und Queues ¹ Möglichkeit zur Nachrichtenpufferung für asynchrone Kommunikation über Message Buffer

¹ Router und Queues sollen in zukünftigen Versionen der Plattform (ggf. mit erweitertem Funktionsumfang) wieder zur Verfügung stehen.

	AppFabric Access Control Service	Fokussierung auf die Absicherung REST-basierter Webservices Integrationsmöglichkeit mit ADFS v2 Unterstützung von WRAP und SWT Entfallen der Unterstützung der WS-* Standards ²
	AppFabric Caching Service	Verfügbarkeit des AppFabric Caching Service
	AppFabric Integration Service	Ankündigung des AppFabric Integration Service
	AppFabric Composite Application	Ankündigung des AppFabric Composite Application Service
Kapitel 6	SQL Azure Relational Database	Firewall zur Absicherung von SQL Azure Datenbanken Zwei Datenbankeditionen (Web und Business) mit verschiedenen Datenbankgrößen (bis zu 50 GB) Bulk-Inserts
	SQL Azure Reporting Service	Verfügbarkeit des SQL Azure Reporting Service
	SQL Azure Data Sync Service	Verfügbarkeit des SQL Azure Data Sync Service

Tabelle 10.4: Überblick über die wichtigsten Änderungen bezogen auf die Buchkapitel

Aktueller Stand der einzelnen Dienste

Die folgenden Abschnitte geben einen detaillierteren Überblick über die Änderungen und Erweiterungen der einzelnen Dienste der Windows Azure Plattform.

Überarbeitung des Developer Portals

Das Developer Portal hat seit seiner Einführung zum Produktstart der Windows Azure Plattform eine Runderneuerung erhalten. Die Oberfläche ist nun Silverlight-basiert und gibt Entwicklern deutlich mehr Möglichkeiten zur Administration ihrer Cloud Services. Dabei hat sich nicht nur die Oberfläche gewandelt, wie in **Abbildung 10.4** zu sehen. Es wurden auch die Workflows für bestimmte Aufgaben optimiert. So können beispielsweise nun beim Anlegen neuer Hosted Services direkt Deployment-Pakete (*.cspkg- und *.cscfg-Datei) installiert werden.

² Die Unterstützung der WS-* Standards soll in zukünftigen Versionen wieder zur Verfügung stehen.

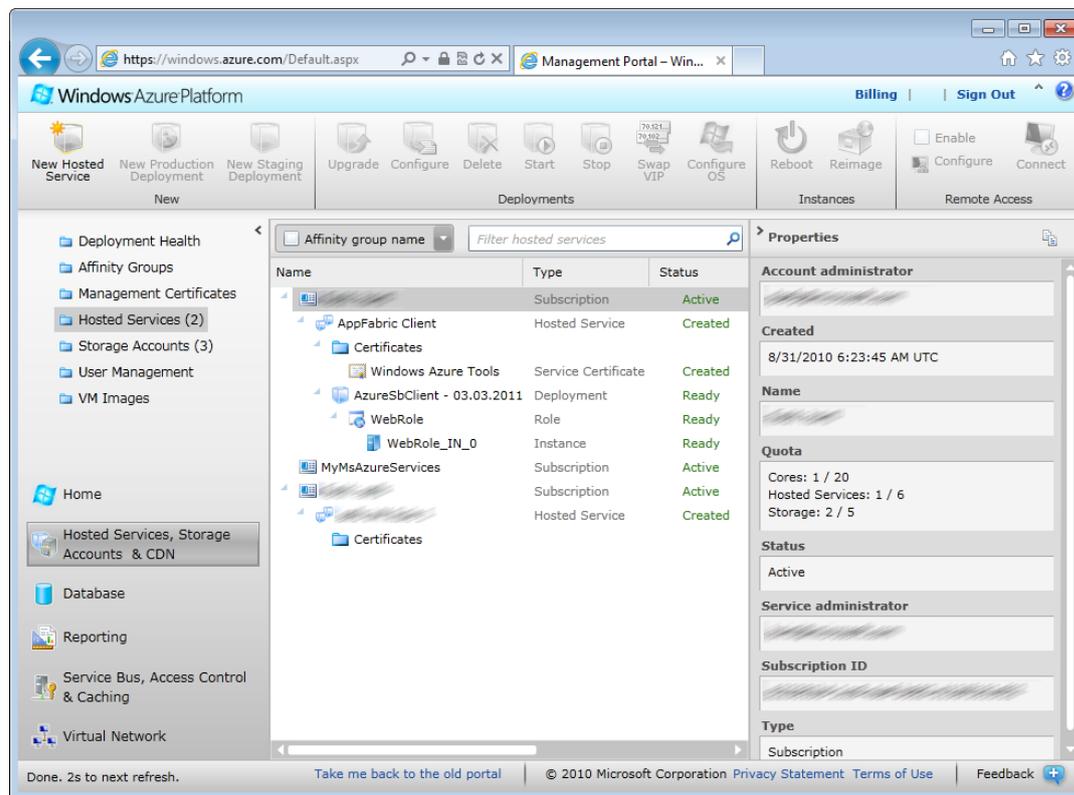


Abbildung 10.3: Das Silverlight-basierte Developer Portal der Windows Azure Plattform

Änderungen in Windows Azure

Nachdem es bereits in der CTP-Phase zahlreiche Veränderungen im Aufbau und Funktionsumfang von Windows Azure gegeben hat, steht mit der Produktivsetzung eine Windows Azure Plattform in einer ersten Version bereit. Der schematische Aufbau ist in **Abbildung 10.4** wiedergegeben.

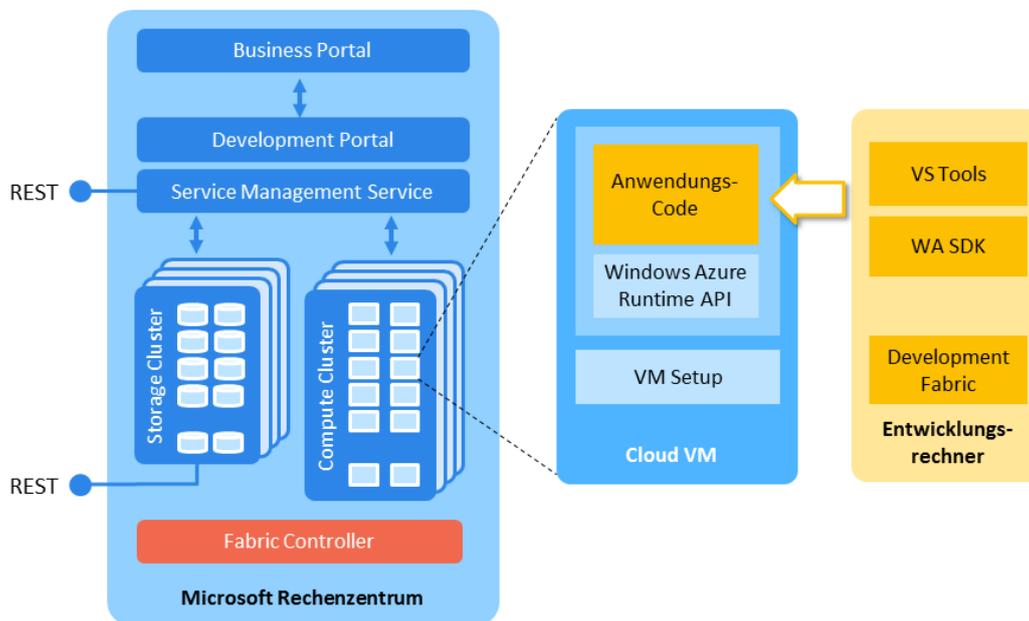


Abbildung 10.4: Aufbau von Windows Azure

Windows Azure umfasst damit folgende Bestandteile:

- **Business Portal**
Über dieses Portal erfolgt die Abrechnung der genutzten Dienste. Dabei teilt sich Windows Azure das Portal mit Microsofts SaaS-Angeboten (SharePoint Online, Exchange Online etc.). Somit haben Nutzer stets einen Überblick über die laufenden Kosten für die Nutzung aller Microsoft Cloud Services.
- **Developer Portal**
Entwickler und Administratoren nutzen dieses Portal, um Anwendungen auf Windows Azure zu installieren und zu verwalten. Hier können neue Projekte angelegt und Einstellungen vorgenommen werden.
- **Service Management Service**
Der Verwaltungsdienst führt die Management-Aufgaben, die ihm entweder über das Entwicklerportal oder die neue Management API (von der aus Aufgaben über Webservice-Aufrufe eingehen) übergeben werden, aus.
- **Storage Cluster**
In den Storage Clustern werden die Dienste für den Windows Azure Storage betrieben. Diese sind von den Rechendiensten getrennt. Ein Zugriff ist direkt von Windows Azure-Anwendungen und von außen über REST Webservices möglich.
- **Compute Cluster**
Die VMs zur Ausführung von Anwendungen werden in Compute Clustern betrieben. Diese sind von den Speicherdiensten getrennt.
- **Fabric Controller**
Das Herzstück von Windows Azure ist der Fabric Controller. Dieser hat alle Infrastrukturreourcen

unter seiner Kontrolle, d.h. er überwacht die Funktionsfähigkeit der Ressourcen, koordiniert Failover-Mechanismen bei Ressourcenausfall, teilt virtuellen Maschinen dynamisch Ressourcen zu, überwacht diese etc.

- *Entwicklungsumgebung*

Für Entwickler stehen zum einen die Tools für Visual Studio (die für Visual Studio 2010 nochmals erweitert werden), zum anderen ein überarbeitetes Software Development Kit (SDK) zur Verfügung. Mit der Development Fabric steht auf dem Entwicklungsrechner auch weiterhin ein lokal simulierter Fabric Controller zur Verfügung.

Mit der Produkteinführung wurden in den einzelnen Bestandteilen von Windows Azure Kenngrößen und funktionale Erweiterungen eingeführt bzw. angekündigt. Die folgenden Abschnitte geben einen Überblick über diese Änderungen.

Windows Azure Compute

Jede Instanz einer Rolle, die in Windows Azure zum Betrieb von Diensten angelegt wird, wird auf einer eigenen *virtuellen Maschine* (VM) ausgeführt. Während der CTP-Phase standen VMs in einer einzigen Größe zur Verfügung. Die VMs waren jeweils mit einem Prozessorkern mit 1,6 GHz und 1,75 Gigabyte Hauptspeicher ausgestattet.

Verschiedene Größen virtueller Maschinen

Mit Version 1.3 von Azure stehen nun fünf verschiedene Größen von VMs zur Verfügung, aus denen der Nutzer auswählen kann: *Extra Small*, *Small*, *Medium*, *Large* und *XLarge*. Diese sind in **Tabelle 10.5** aufgelistet.

	Extra Small	Small	Medium	Large	Extra Large
Prozessor	Shared	1 x 1,6 GHz	2 x 1,6 GHz	4 x 1,6 GHz	8 x 1,6 GHz
Speicher	768 MB	1,75 GB	3,5 GB	7,0 GB	14,0 GB
Lokale Festplatte	20	225	490	1000	2040
Bandbreite (Mbps)	5	100	200	400	800
Kosten pro Service-Stunde	0,0355 €	0,0852 €	0,1704 €	0,3408 €	0,6816 €

Tabelle 10.5: Verschiedene Größen virtueller Maschinen

Die Variante *Small* ist die zuvor im Rahmen der CTP verfügbare VM. Mit den weiteren VMs gehen bezüglich Zahl der Prozessorkerne und Speicherausstattung jeweils Verdoppelungen einher (*Medium* ist doppelt so groß wie *Small* usw.). *Extra Small* nimmt eine Sonderstellung ein, da hier nicht eine oder mehrere Rechenkerne exklusiv zur Verfügung stehen, sondern ein Rechenkern mit anderen Instanzen geteilt wird.

Direkter Nachrichtenaustausch zwischen Rollen

Die Kommunikation zwischen zwei Rollen erfolgte bislang immer über eine Vermittlerkomponente. Dies konnte eine Windows Azure Storage Queue, Blob-Storage etc. sein, über die Nachrichten zwischen Rollen ausgetauscht und hierüber eine Kommunikationsbeziehung unterhalten werden konnte. Neu ist, dass Web und Worker Roles direkt miteinander kommunizieren können. Die API stellt hierzu Funktionen bereit, mit denen einzelne Instanzen einer Rolle identifiziert und Nachrichten über direktem Weg (http oder tcp) ausgetauscht werden können. Die Identifikation einer Rolle ist insbesondere dann wichtig, wenn mehr als eine Instanz einer Rolle betrieben wird, die Kommunikation aber immer mit der jeweils gleichen Instanz erfolgen soll.

Änderungen hat es nicht nur in der Kommunikation zwischen Rollen, sondern auch im Bereich der Kommunikation mit der Außenwelt gegeben. Rollen haben nun gewissen Einfluss darauf, wie sie von außen über den Loadbalancer angesprochen werden können. Für Web und Worker Roles kann konfiguriert werden, über welchen Port sie Nachrichten vom Loadbalancer empfangen können. Dies erfolgt in der Servicedefinition (*.csdef-Datei). **Listing 10.1** zeigt die relevanten Stellen für den Fall einer Web Role. Es können der externe Port (<port>-Attribut), über den die Rolle von außen aus dem Internet ansprechbar ist, der interne Port (<localPort>-Attribut), auf dem der Rollenservice letztlich Nachrichten empfängt, sowie das Übertragungsprotokoll (<protocol>-Attribut) spezifiziert werden.

```
<ServiceDefinition ...>
  <WebRole name="<web-role-name>" vmSize="[ExtraSmall|Small|Medium|Large|ExtraLarge]" ... >
    <InputEndpoints >
      <InputEndpoint certificate="<certificate-name>"
        ignoreRoleInstanceStatus="[true|false]"
        name="<input-endpoint-name>"
        protocol="[http|https|tcp]"
        localPort="<port-number>"
        port="<port-number>" />
    </InputEndpoints>
    <Endpoints>
      <InputEndpoint certificate="<certificate-name>"
        ignoreRoleInstanceStatus="[true|false]"
        name="<input-endpoint-name>"
        protocol="[http|https|tcp]"
        localPort="<port-number>"
        port="<port-number>" />
      <InternalEndpoint name="<internal-endpoint-name>"
        protocol="[http|tcp]"
        port="<port-number>">
        <FixedPort port="<port-number>" />
        <FixedPortRange min="<minium-port-number>"
          max="<maximum-port-number>" />
      </InternalEndpoint>
    </Endpoints>
    ...
  </WebRole>
</ServiceDefinition>
```

Listing 10.1: Definition von Endpunkten zur Kommunikation mit und zwischen Rollen

Die Rolle selbst kann dabei über einen ähnlich dem in **Listing 10.2** angegebenen Code den eigenen Endpunkt ermitteln.

```
IPEndPoint endpoint =
  RoleEnvironment.Roles["WorkerRole1"].Instances[0].InstanceEndpoints["TestEndpoint"].IPEndPoint;
```

Listing 10.2: Ermittlung des IP-Endpunkts (Ports) über die Windows Azure Runtime API

Dieser Endpunkt kann dann verwendet werden, um Nachrichten zu empfangen, die von außen an den betreffenden Port gesendet wurden. Hierzu stehen wiederum Befehle der Windows Azure Runtime API zur Verfügung. Ein entsprechendes Codefragment ist in **Listing 10.3** zu sehen.

```
var client = new TcpClient();
client.Connect(endpoint);
var rd = new StreamReader(client.GetStream());
var test = rd.ReadToEnd();
client.Close();
```

Listing 10.3: Empfang von Daten über den zuvor ermittelten IP-Endpunkt

Logging- und Diagnose-System mit der Diagnostics API

Nachdem ein direktes Debuggen von Anwendungen, die auf Windows Azure betrieben werden, nicht möglich ist, kommt dem Logging- und Diagnose-System, welches mit Produktivsetzung der Plattform verfügbar ist, besondere Bedeutung zu. Dieses System zielt unter anderem auf folgende Einsatzszenarien ab:

- Messung der Performanz der Anwendung
- Ermittlung des Ressourcenverbrauchs
- Fehlersuche und -behebung
- Überwachung
- Prüfung der Metriken für die Dienstqualität der Anwendung
- Kapazitätsplanung (um die Zahl der Anwendungsinstanzen zu optimieren)
- Auslastungsanalyse (Anwender, Zugriffsstatistiken)
- Abrechnung und Auditierung der Anwendung

Um Tracing-Ausgaben erzeugen zu können, muss in der *web.config*-Datei einer Anwendung (diese wird von Visual Studio für Web Roles automatisch erzeugt) ein sogenannter Trace Provider registriert werden. Dies kann wie in **Listing 10.4** gezeigt erfolgen.

```
<system.diagnostics>
  <trace autoflush="false" indentsize="4">
    <listeners>
      <add name="AzureDiagnostics"
          type="Microsoft.WindowsAzure.Diagnostics.DiagnosticMonitorTraceListener,
              Microsoft.WindowsAzure.Diagnostics,
              Version=1.0.0.0,
              Culture=neutral,
              PublicKeyToken=31bf3856ad364e35" />
    </listeners>
  </trace>
</system.diagnostics>
```

Listing 10.4: Registrierung eines Trace Providers als Voraussetzung zur Erzeugung von Trace-Ausgaben

Mit Hilfe der bekannten Tracing API können damit dann Tracing-Ausgaben in Logs erfolgen, die gesammelt und dann zeitgesteuert oder auf Anforderung in Windows Azure Storage abgelegt werden. Beispielaufrufe der Tracing API sind in **Listing 10.5** zu sehen.

```
Trace.WriteLine("Hier ist ein KRITISCHER Log-Eintrag", "Critical");
Trace.WriteLine("Hier ist eine INFORMATION für das Log", "Information");
```

Listing 10.5: Beispielaufrufe der Tracing API

Bei der Übertragung in den Azure Storage, von wo aus die Logs dann auf dem üblichen Weg (Storage API oder Webservices) ausgelesen werden können, können die Logs nach Datentyp, Level und Zeitraum gefiltert werden. **Tabelle 10.6** gibt einen Überblick über die möglichen Informationsquellen für die Anwendungsüberwachung und Analyse.

Datenquelle	Standardkonfiguration	Konfiguration	Format
Trace Logs	aktiviert, lokal gespeichert	Diagnostics API, Trace Listener	Table
Performance Counters	deaktiviert	Diagnostics API	Table
Windows Event Logs	deaktiviert	Diagnostics API	Table
Infrastruktur-Logs	aktiviert, lokal gespeichert	Diagnostics API	Table

IIS Logs	aktiviert, lokal gespeichert	Diagnostics API, Web.config	Blob
IIS Failed Request Logs	deaktiviert	Diagnostics API, Web.config	Blob
Application Crash Dumps	deaktiviert	Diagnostics API, Crash API	Blob
Beliebige Logs & Dateien	deaktiviert	Diagnostics API	Blob

Table 10.6: Informationsquellen für die Anwendungsüberwachung und -analyse

Die Datenquellen erzeugen Informationen, die in den Windows Azure Storage (als Tables oder Blobs) übertragen werden können und dort für Analyseprozesse zur Verfügung stehen.

Programmgestützte Verwaltung über die Service Management API

Es steht nun eine Management API zur Verfügung, über die die meisten Aktionen, die über das Windows Azure Developer Portal zugänglich sind, aus einem Skript oder Programm (unabhängig davon, ob dieses auf Windows Azure oder außerhalb betrieben wird) heraus ausgeführt werden können. Über die Service Management API können Storage Accounts, Services, Service Deployments und Affinity Groups verwaltet werden. Die API selbst ist dabei eine REST API, deren Operationen über SSL erfolgen und über X.509 v3-Zertifikate abgesichert werden.

Um mit der Service Management API arbeiten zu können, ist zunächst das Anlegen eines Benutzerkontos über das Windows Azure Developer Portal erforderlich. Diesem ist eine eindeutige Subscription ID zugeordnet, die für API-Aufrufe benötigt wird und die auf dem Portal beim Konto angezeigt wird. Innerhalb eines Benutzerkontos können zwei Arten von Diensten angelegt werden: Storage Accounts und Hosted Services.

Im Falle eines Storage Accounts umfasst die API beispielsweise Operationen zum Auflisten von Storage Accounts innerhalb eines Benutzerkontos, zum Ermitteln von Eigenschaften des Storage Accounts, Auslesen oder Neugenerieren des primären oder sekundären Zugriffsschlüssels. Eine vollständige Liste der API-Operationen findet sich in der MSDN-Dokumentation.

Die API kann im Falle von Hosted Services zum Beispiel dazu verwendet werden, die Eigenschaften des Service auszulesen, Deployments zu aktualisieren und zu verwalten und Konfigurationen der Dienste zu ändern.

Betrieb eigener VMs

Neben den bereits während der CTP-Phase verfügbaren Web und Worker Roles wurde mit Version 1.3 der Plattform eine dritte Art von Rollen eingeführt: die Virtual Machine Role. Während bei Web und Worker Roles Software auf eine von Azure vorkonfigurierte VM aufgespielt wird, besteht bei VM Roles die Möglichkeit, das VM-Image selbst zu erstellen und dann auf Windows Azure auszuführen. Für die Konfiguration erhält der Entwickler deutlich mehr Flexibilität als bei Web und Worker Roles. Erkauft wird diese Flexibilität allerdings mit der Verantwortung, das Gast-Betriebssystem (aktuell Windows Server 2008 R2 Enterprise) selbst zu warten, d.h. Patches und Upgrades einzuspielen etc. **Abbildung 10.5** skizziert die Schritte zur Erstellung einer eigenen VM Role.

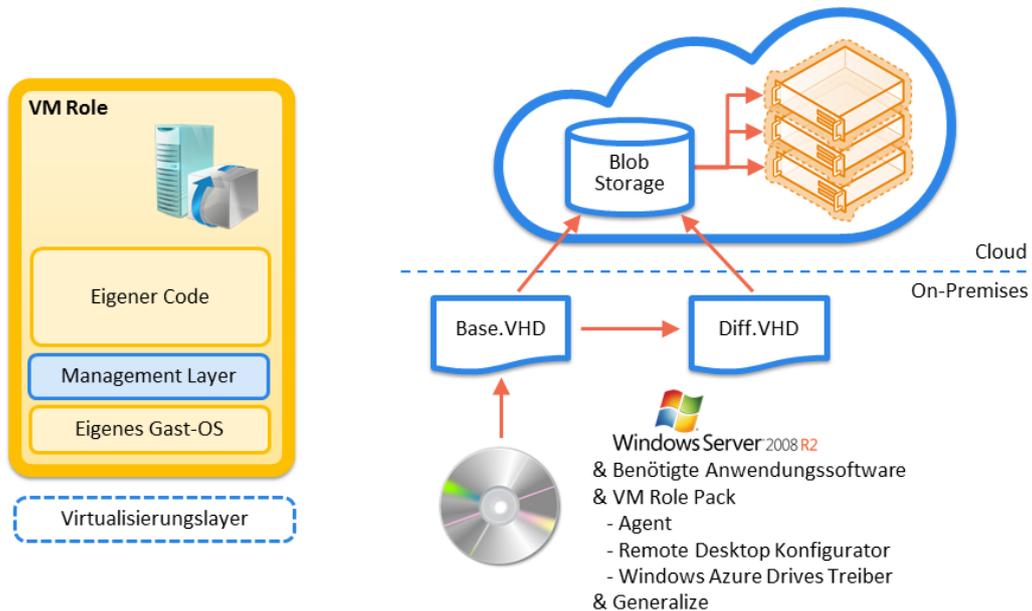


Abbildung 10.5: Erstellung einer VM Role

Auf Basis eines Installationsmediums (Windows Server 2008 R2) kann ein neues VM-Image in Form einer Base.VHD erstellt werden. In dieses Image müssen Windows Azure Integrationskomponenten, die zur Kommunikation der VM mit Windows Azure dienen, installiert werden. Danach können benötigte Anwendungen installiert, das Betriebssystem angepasst werden etc. Abschließend werden durch einen Generalize-Vorgang Abhängigkeiten zur Installationsmaschine entfernt. Eine Basisversion der VM kann dann als Base.VHD in Windows Azure Blob Storage geladen werden. Werden Variationen von der Base.VHD benötigt, können diese in Form von Diff.VHDs ebenfalls in Blob Storage gespeichert werden. Per Service Definition kann nun spezifiziert werden, aus welcher Base.VHD und (optional) Diff.VHD die VM Rolle erstellt und in einer oder mehreren Instanzen ausgeführt werden soll.

VM Roles eignen sich insbesondere dann, wenn aufwändigere Installationsroutinen für bestehende Anwendungen, die in der Cloud ausgeführt werden sollen, erforderlich sind. Beim Einsatz muss allerdings beachtet werden, dass VM Roles bei einem Re-Image-Vorgang (der beispielsweise bei Hardware-Ausfall vorgenommen wird) wieder in ihren Ausgangszustand, d.h. auf den Stand der Base.VHD/Diff.VHD-Kombination zurückgesetzt werden. Änderungen, die zur Laufzeit innerhalb der VM gespeichert wurden (z.B. nachträgliche Installation oder Konfiguration von Software), gehen dabei verloren. Eigene Anwendungen sollten Datenspeicherungen deshalb unbedingt außerhalb der VM, beispielsweise in Windows Azure Storage oder SQL Azure, vornehmen. Dort bleiben sie auch beim Zurücksetzen der VM erhalten.

Startup-Tasks

So naheliegend der Einsatz von VM Roles für die Migration bestehender Anwendungen in die Cloud auf den ersten Blick auch erscheint, so teuer wird die Flexibilität der VM Roles auch erkaufte: der Anwender ist für die Betreuung und Administration des Gast-Betriebssystems alleine verantwortlich. Security Patches, Betriebssystem-Upgrades müssen selbst eingespielt werden. Es ist daher eine gute Nachricht, dass viele Vorteile der VM Role durch neu eingeführte Startup-Tasks auch für Web und Worker Roles verfügbar sind. Startup-Task erlauben die Ausführung von Kommandozeilen-Skripten

und Anwendungen beim Hochfahren von Web und Worker Roles. In der Service-Definition werden hierzu die Aufrufkommandos und Einstellungen für den Ausführungskontext beschrieben.

```
<ServiceDefinition ...>
  <WebRole ...>
    ...
    <Startup priority="">
      <Task commandLine="<command-to=execute>"
        executionContext=" [limited|elevated] "
        taskType=" [simple|foreground|background] ">
        <Environment>
          <Variable name="<variable-name>" value="<variable-value>" />
        </Environment>
      </Task>
    </Startup>
  </WebRole>
</ServiceDefinition>
```

Listing 10.6: *Startup-Tasks in der Service-Definition einer Web Role*

Mit Startup-Tasks ist es also möglich, beim Hochfahren einer Web oder Worker Role Installationsroutinen auszuführen, was es in vielen Fällen unnötig macht, auf VM Roles zurück zu greifen. Beim Einsatz von Startup-Tasks bleibt das automatisierte Management der betreffenden Roles erhalten. Man gewinnt also Flexibilität in den Rollen ohne auf den Komfort des Rollenmanagements von Windows Azure verzichten zu müssen.

Plugins

Mit Plugins können vorgefertigte Import-Module für das Gast-Betriebssystem einer Web oder Worker Roles aktiviert werden. Diese werden in der Service-Definition der betreffenden Rolle im <Import>-Element spezifiziert. **Listing 10.7** zeigt den entsprechenden Abschnitt in der Service-Definition einer Web Role.

```
<ServiceDefinition ...>
  <WebRole ...>
    <Imports>
      <Import moduleName="<import-module>" />
    </Imports>
  </WebRole>
</ServiceDefinition>
```

Listing 10.7: *Plugins in der Service-Definition einer Web Role*

Derzeit sind vier Plugins verfügbar, die aktiviert werden können:

- RemoteAccess und RemoteForwarder – werden zur Konfiguration von Remote Desktop Verbindungen benötigt werden
- Diagnostics – tritt an Stelle der bisher in der OnStart()-Methode einer Rolle einzustellenden Azure Diagnostics Funktionalität
- Connect – wird zur Konfiguration einer geschützten IPsec Verbindung zwischen Azure Rollen und lokalen Ressourcen eines eigenen Rechenzentrums benötigt

Erweiterte Interoperabilität

Nachdem schon bei den Schnittstellen nach außen großer Wert auf Standards gelegt wurde – so bleibt es grundsätzlich möglich, auf Services, die auf Azure betrieben werden, über REST- und andere Schnittstellen zuzugreifen – können nun auch Dienste, die in PHP oder Java entwickelt wurden, auf Windows Azure betrieben werden. Allgemein formuliert können auf Windows Azure grundsätzlich alle Technologien betrieben werden, die auf Windows Server ausgeführt werden können. Entwickler haben

hierbei die Möglichkeit, ihre jeweils benötigte Laufzeitumgebung selbst zu bestimmen.

Mit der Möglichkeit Ports auf Windows Azure für den Nachrichtempfang von außen zu konfigurieren, können auf Azure auch Anwendungsserver wie Apache Webserver oder Tomcat betrieben werden und über die entsprechenden Ports Nachrichten empfangen. Darüber hinaus ist es auch möglich, MySQL auf Azure zu betreiben. Damit sind die Voraussetzungen geschaffen, auf Basis von Azure einen WAMP-Stack (Windows-Apache-MySQL-PHP) in der Cloud bereitzustellen. Auf der PDC wurde dies von Matt Mullenweg, Gründer von Automattic, eindrucksvoll belegt, indem er sein beliebtes, auf PHP, Apache und MySQL basierendes Content Management System WordPress auf Azure installiert vorführte.

Neben Visual Studio wird es auch für Eclipse Entwicklerwerkzeuge zur Programmierung von Windows Azure-basierten Anwendungen bzw. zum Zugriff auf SQL Azure oder die AppFabric geben.

Windows Azure Storage

Der Speicherdienst von Windows Azure umfasste bisher Queue, Table und Blob Storage. Diese wurden auf Basis zahlreichen Kunden- und Anwenderfeedbacks in ihren Funktionen deutlich erweitert. Viele der neuen Funktionen reduzieren den Umstellungsaufwand bei der Migration bestehender Anwendungen auf Windows Azure.

Drives als virtuelle NTFS-Festplatten

Blob Storage wurde beispielsweise um eine wichtigen Funktionalität erweitert: Drives. Bei Drives handelt es sich um Blob Storage, der als eine virtuelle NTFS-Festplatte (Virtual Hard Drive (VHD)) formatiert wird. Diese steht dann Windows Azure-Anwendungen zur dauerhaften Speicherung von Daten in einem virtuellen Dateisystem zur Verfügung. Drives zeichnen sich durch folgende Eigenschaften aus.

- Ein Drive kann eine Maximalgröße von 1 Terabyte haben
- Eine VM kann dynamisch bis zu acht Drives anbinden
- Ein Drive kann zu einem Zeitpunkt nur an eine einzige VM schreibend angebunden werden
- Zugriff von außen erfolgt über die Page Blob API
- Import und Export von Drives ist möglich. Ein VHD kann über die Blob-Schnittstelle hochgeladen bzw. darüber auch wieder heruntergeladen werden.

Mit Drives hat Microsoft nun eine Möglichkeit geschaffen, Anwendungen, die auf den Zugriff auf ein Dateisystem angewiesen sind, auf Windows Azure zu migrieren.

Offizielle Storage Client Library

Für den Zugriff auf den Windows Azure Storage ist eine offizielle Storage Client Library verfügbar. Dies hat zur Folge, dass Entwickler nicht mehr über Klassenbibliotheken aus dem *Samples*-Namespace, sondern über den Namespace *Microsoft.WindowsAzure.StorageClient* mit dem Windows Azure Storage interagieren. Anwendungen, die während der CTP-Phase erstellt wurden, müssen in diesem Bereich also geändert werden, um den neuen Namespace zu unterstützen.

Erweiterte Funktionalitäten der Speicherdienste

Die Speicherdienste werden noch um einige weitere Funktionalitäten erweitert. Stichpunktartig die wichtigsten Erweiterungen:

- Operationen auf Table Entitys können unter bestimmten Voraussetzungen zu Entity Group Transactions (bis zu 100 Anweisungen in einer Transaktion) zusammengefasst werden
- Mittels *CopyBlob* können Kopien eines Blobs innerhalb eines Storage Accounts angelegt werden

- Über *SnapshotBlob* können schreibgeschützte Backups und Versionen eines Blobs erzeugt werden. Kostenpflichtig sind dabei nur jeweils individuelle Versionen von Blöcken über alle Versionen eines Blobs.
- Für den Blob Storage kann ein Root Blob Container definiert werden. Dieser hat die Form *http://mystorage.blob.core.windows.net/\$root/blobdocument.xml*. Beim Zugriff kann dann die Angabe eines Containernamens entfallen.
- Für den Windows Azure Storage kann ein eigener Domainname vergeben werden

Windows Azure Content Delivery Network

Beim Zugriff auf große Dateien, die im Windows Azure Blob Storage abgelegt sind, kann die Netzwerklatenz zu einem Problem werden, wenn der Anwender in großer räumlicher Distanz zum Rechenzentrum ist, in dem der Blob abgelegt ist. Um diese Latenz zu minimieren, bietet Microsoft das sogenannte *Content Delivery Network (CDN)* an.

Das CDN umfasst 18 Standorte rund um den Globus und wird stetig ausgebaut. Das Windows Azure CDN legt Kopien von Blobs in einem Zwischenspeicher an den verschiedenen Standorten an, um einem Anwender Blobs vom jeweils nächsten Standort auszuliefern und dadurch die Übertragungszeit zu optimieren. Der Einsatz des CDN kann über das Windows Azure Developer Portal aktiviert werden.

Dabei wird für den Blob Storage eine CDN-URL der Form *http://<guid>.vo.msecnd.net/* vergeben. Dieser URL kann dazu verwendet werden, Blobs aus einem öffentlichen Blob-Speicher zu lesen. Der Zugriff über den bestehenden URL über den Storage Account ist dabei weiterhin möglich.

Ein Beispiel: Gegeben sei ein Storage Account namens *mystorageaccount*. In diesem befindet sich ein öffentlicher Container *images*. Sobald der CDN-Zugriff auf diesen Container eingerichtet ist, können Anwender den Container über zwei URLs auslesen:

- Windows Azure Blob Service URL: *http://mystorageaccount.blob.core.windows.net/images/*
- Windows Azure CDN URL: *http://<guid>.vo.msecnd.net/images/*

Beim Zugriff über den Service-URL erfolgt der Zugriff direkt auf den am ursprünglichen Ort gespeicherten Container. Beim Zugriff über den CDN-URL wird die Anfrage sofort an den Endpunkt weitergeleitet, der dem Anwender am nächsten ist.

Windows Azure Connect

Mit Windows Azure Connect (vormals Project »Sydney«) hat Microsoft die Möglichkeit geschaffen, auf Netzwerkebene eine sichere IPv6-Anbindung von Azure Ressourcen mit lokal im Unternehmensnetzwerk betriebenen Ressourcen herzustellen. Die Verbindung wird hierbei über das Service Modell gesteuert. Einzelne Rollen können hierbei einzeln für die Netzwerkanbindung konfiguriert werden. In Web und Worker Roles muss in der Service Definition (*.csdef) das Connect Plugin aktiviert werden, in VM Roles muss der Connect Agent über das Connect VM Installationspaket installiert werden.

Auf den betreffenden Rechnern des Unternehmensnetzwerks muss hierzu ein sogenannter »Connect Agent« installiert werden. Dieser ist für die Betriebssysteme Windows Server 2008, 2008 R2, Windows Vista und Windows 7 verfügbar. Die lokalen Ressourcen werden zu Gruppen zusammengefasst. Ein Rechner kann dabei nur einer Gruppe angehören. Rollen in Azure Services werden dann mit lokalen Gruppen verknüpft. **Abbildung 10.6** skizziert exemplarisch eine solche Konfiguration.

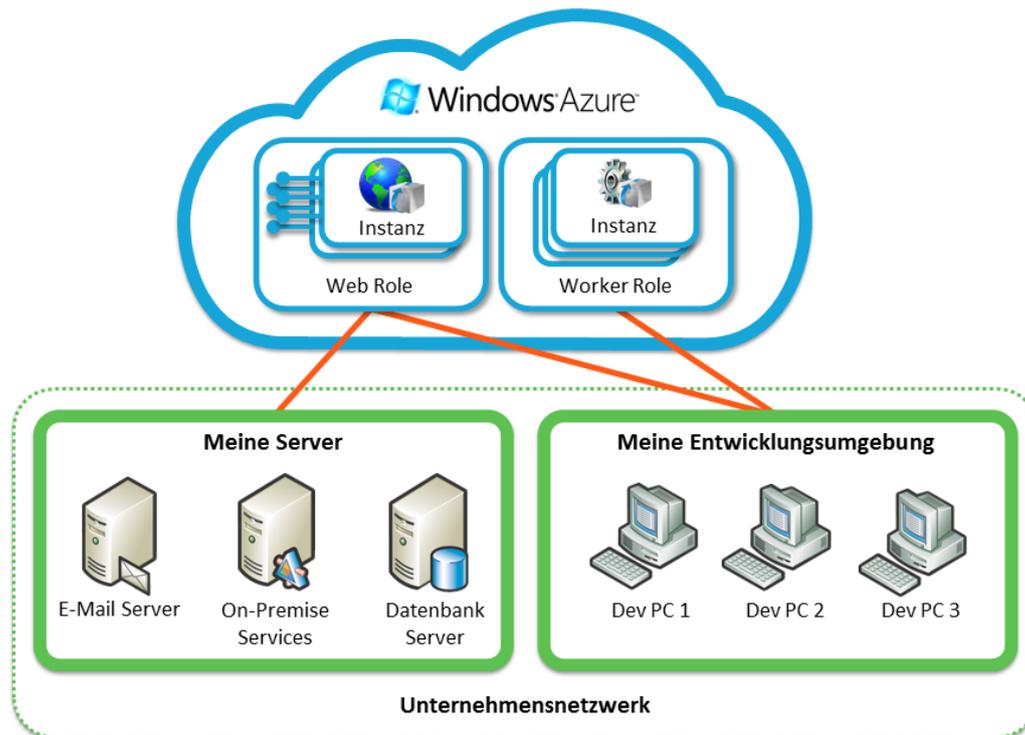


Abbildung 10.6: Windows Azure Connect

Im gezeigten Szenario sind die beiden Rollen über Windows Azure Connect mit dem Unternehmensnetzwerk verbunden. Die Web Role hat dabei Zugriff auf die lokalen Gruppen »Meine Server« und »Meine Entwicklungsumgebung«. Somit erhält sie Zugriff auf den lokalen E-Mail-Server, verschiedene lokale Web Services sowie einen lokalen Datenbank Server. Die Worker Role ist nur mit der lokalen Entwicklungsumgebung verbunden. Über die dort enthaltenen Entwicklerrechner kann (z.B. über Remote Powershell) auf Web und Worker Role zugegriffen werden.

Version 1.3 der Windows Azure Managed Library

Die Klassenbibliothek für die Interaktion mit Windows Azure, die im Windows Azure SDK enthalten ist, wurde im Rahmen der Produktivsetzung der Plattform überarbeitet. Sie umfasst nun die folgenden Namespaces:

- `Microsoft.WindowsAzure`: Über die Klassen dieses Namespaces können die Credentials für Windows Azure Storage Accounts verwaltet werden
- `Microsoft.WindowsAzure.ServiceRuntime`: Klassen in diesem Namespace erlauben die Interaktion mit der Windows Azure-Umgebung aus einer Rolle heraus
- `Microsoft.WindowsAzure.Diagnostics`: Die Klassen aus diesem Namespace werden für die Sammlung von Logging- und Diagnosemeldungen benötigt
- `Microsoft.WindowsAzure.Diagnostics.Management`: Klassen dieses Namespaces können dazu verwendet werden Logging- und Diagnoseinformationen auszulesen
- `Microsoft.WindowsAzure.StorageClient`: Dieser Namespace umfasst die Klassen für den Zugriff auf den Windows Azure Storage Service. Die betreffenden Klassen waren während der CTP-

Phase im Namespace `Samples` enthalten.

- `Microsoft.WindowsAzure.StorageClient.Protocol`: Klassen aus diesem Namespace stellen Wrapper für den Zugriff auf den Windows Azure Storage via REST-Protokoll zur Verfügung.

Für Anwendungen, die während der CTP-Phase erstellt wurden, ist hier also unter Umständen eine Überarbeitung erforderlich. Dies ist insbesondere dann der Fall, wenn Funktionalitäten des `StorageClients` genutzt wurden, der sich zur CTP-Phase im `Samples`-Namespace befand.

Änderungen in SQL Azure

SQL Azure ist im Kern unverändert, wurde allerdings um ein paar sehr interessante Funktionen erweitert. Die in [Kapitel 6](#) beschriebenen Aspekte und Beispiele sind weiterhin gültig.

SQL Azure Relational Database Service

Die Weiterentwicklungen der Datenbankengine von SQL Azure betrafen vor allem den Umgang mit Datenbeständen, die größer sind als die zur Produkteinführung verfügbare Maximalgröße von 10 GB. Hier wurden sich zum einen weitere Datenbankgrößen ermöglicht, und – für alle, für die auch die erweiterte Maximalgröße nicht ausreicht – mit SQL Azure Federation ein Konzept eingeführt, mit dem es möglich wird, Datenbestände sinnvoll auf mehrere Datenbanken zu verteilen.

Neue Editionen und Datenbankgrößen

Nutzer haben die Auswahl aus zwei Editionen und verschiedenen Datenbankgrößen. Diese sind in [Tabelle 10.7](#) gibt einen Überblick über die Möglichkeiten.

Edition	Maximalgrößen und Kosten pro Monat	
Web Edition	1 GB:	7,085 €
	5 GB:	35,425 €
Business Edition	10 GB:	70,913 €
	20 GB:	141,826 €
	30 GB:	212,739 €
	40 GB:	283,652 €
	50 GB:	354,565 €

Tabelle 10.7: Editionen, Maximalgrößen und monatliche Kosten bei SQL Azure

Beim Anlegen ist zunächst wichtig, sich für eine bestimmte Edition zu entscheiden. Innerhalb einer Edition kann die Maximalgröße zur Laufzeit geändert werden. Bis zur jeweils eingestellten Maximalgröße können Daten in die Datenbank geschrieben werden. Bei Erreichen des Limits, gibt SQL Azure eine Fehlermeldung zurück.

Die für SQL Azure anfallenden Kosten richten sich nach dem maximalen Füllstand die eine Datenbank an einem bestimmten Tag erreicht. Der Rechnungsbetrag ergibt sich innerhalb einer Edition aus dem Tarif für die kleinste Maximalgröße, die größer als der Füllstand ist. Werden beispielsweise in einer Datenbank der Business Edition 15 GB Daten gehalten, so fallen täglich Kosten für eine Datenbank der Maximalgröße 20 GB an (also 141,826 € / 30 Tage).

SQL Azure Federation

Für viele Anwender ist die maximale Datenbankgröße in SQL Azure von 50 GB für ihren zu speichernden Datenbestand nicht ausreichend. Mit SQL Azure Federation besteht die Möglichkeit, größere Datenbestände auf mehrere Datenbanken aufzuteilen. Hierbei kommen folgende Eigenschaften

zum Einsatz:

- *Federation* – die Menge der Daten, die auf mehrere Datenbanken verteilt wird
- *Federation Root* – Datenbank, die das Federation Verzeichnis enthält
- *Federation Key* – Schlüsselattribut, das die Zuordnung eines Records auf eine Datenbank bestimmt
- *Atomic Unit* – alle Records mit dem gleichen Wert im Federation Key
- *Federation Member* (auch bekannt als *Shard*) – Datenbank, die die Records einer Menge an Atomic Units enthält

Abfragen werden stets an die Federation Root gestellt. Diese leitet die Abfragen gegebenenfalls an den Federation Member, der die Atomic Unit des abgefragten Federation Keys weiter. **Abbildung 10.7** zeigt den Aufbau eines Datenbankverbunds, in dem über Federation ein Datenbestand gehalten wird.

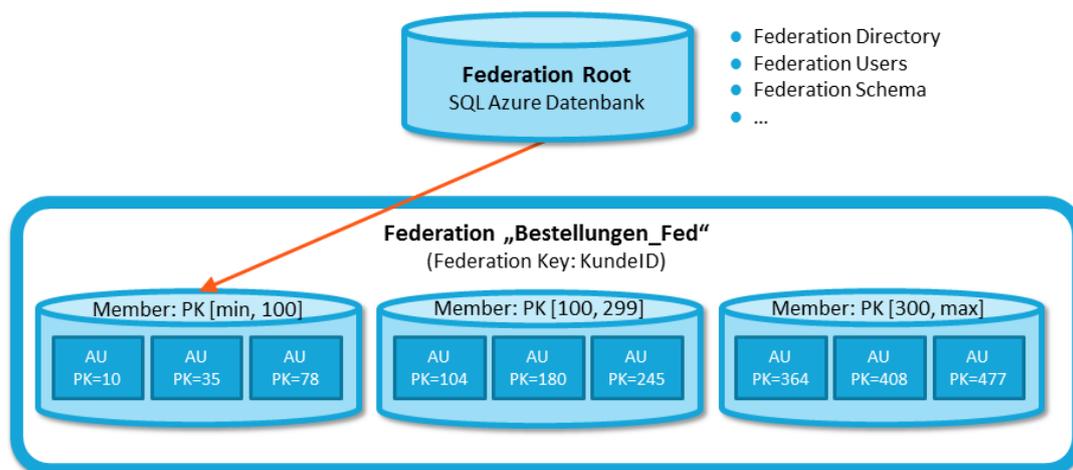


Abbildung 10.7: SQL Azure Federation

Im Beispiel ist die KundenID der Federation Key, d.h. anhand des Wertes der KundenID entscheidet das System, in welchem der drei Federation Member der betreffende Datenbank-Record gespeichert wird.

SQL Azure Reporting Service

Die Reporting Services von SQL Azure erlauben es, auf Basis von SQL Azure Datenbanken Reportings zu erzeugen und bei Bedarf auch in Windows Azure Anwendungen einzubetten. Hierbei können die gleichen Werkzeuge verwendet werden, die auch bei SQL Server Reporting zum Einsatz kommen.

Entwickler können Reports, die auf Daten aus SQL Azure basieren, mit Hilfe des Business Intelligence Development Studios erstellen. Reports können verschiedene Datenvisualisierungen wie Maps, Charts etc. enthalten und können in verschiedene Ausgabeformate (Excel, Word, XML, PDF) exportiert werden. Über URLs ist auch der direkte Zugriff auf Reports im SQL Azure Reporting Service möglich. Reports und User können dabei über das Azure Portal verwaltet werden.

Die Einbettung von Reports kann entwicklerseitig über SOAP Webservice APIs oder über ein Report Viewer Control im Connected Mode erfolgen.

SQL Azure Sync Service

Der SQL Azure Data Sync Service umfasst Bibliotheken und eine Laufzeitumgebung, die die Datensynchronisation mit SQL Azure unterstützt. Der Dienst basiert auf dem Sync Framework und

unterstützt zwei grundsätzliche Szenarien:

- Erweiterung einer bestehenden Vor-Ort-Infrastruktur in die Cloud
Daten können über die Synchronisation mit SQL Azure vielen Anwendern bereitgestellt werden, ohne dass diese notwendigerweise direkten Zugriff auf die Vor-Ort-Infrastruktur (die möglicherweise hinter einer Firewall liegt) haben müssen.
- Implementierung von Offline-fähigen Anwendungen
Anwendungen, die ihre lokalen Datenbanken mit SQL Azure synchronisieren, können beim Ausfall der Netzwerkverbindung mit ihrer lokalen Datenkopie arbeiten. Änderungen werden, sobald die Verbindung wieder steht, automatisch in die zentrale SQL Azure-Datenbank übertragen.

SQL Azure Migration Wizard

Als weiteres Entwicklungswerkzeug steht weiterhin der SQL Azure Migration Wizard, der über den URL <http://sqlazuremw.codeplex.com/> heruntergeladen werden kann, zur Verfügung. **Abbildung 10.8** zeigt dessen Hauptbildschirm, von dem aus die verschiedenen Funktionsbereiche aufgerufen werden können.

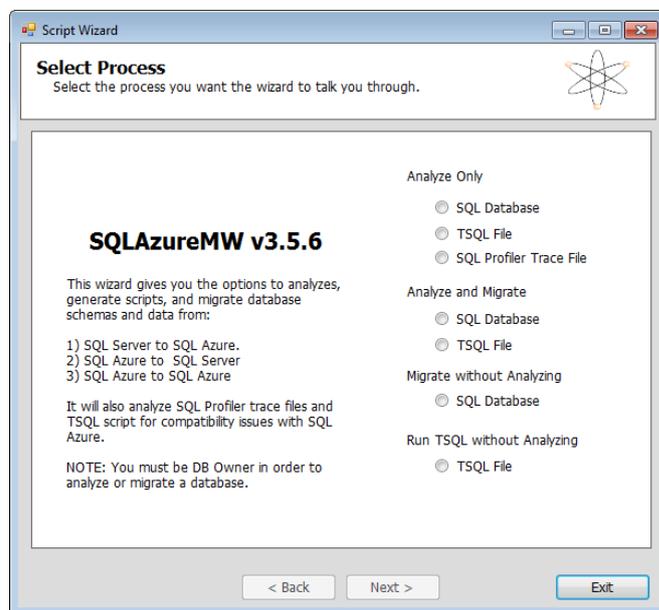


Abbildung 10.8: Hauptbildschirm des SQL Azure Migration Wizards

Der SQL Azure Migration Wizard bietet folgende Möglichkeiten:

- Analyse von SQL, TSQL und SQL Profiler-Skripten
- Migration einer Datenbank von SQL Server nach SQL Azure
- Migration einer Datenbank von SQL Azure nach SQL Server
- Migration einer Datenbank von SQL Azure nach SQL Azure

Die Analysefunktion prüft, ob ein bestehendes Datenbankskript kompatibel mit dem Funktionsumfang von SQL Azure ist.

Ausblick

Bezüglich SQL Azure wurden für zukünftige Versionen bereits einige vielversprechende Funktionen

angekündigt:

- *Database Cloning* zur Erstellung von Datenbankkopien
- Laufende Backups, mit deren Hilfe historische Daten wiederhergestellt werden können
- Upgrade und Downgrade zwischen Größenoptionen (Web Edition vs. Business Edition)
- Schreibgeschützte Datenbanken
- Dynamische Datenbanksplits und -zusammenführungen

Eine im Kontext von Unternehmensanwendungen äußerst interessante Erweiterung von SQL Azure wird derzeit unter der Bezeichnung *SQL Azure Codename »Vidalia«* geführt. Dabei handelt es sich um eine Technologie, mit der Unternehmen sehr feingranular Zugriffsrechte auf SQL Azure-Datenbanken erteilen können. Dies ist insbesondere für sensible Geschäftsdaten erforderlich, auf die beispielsweise nur vereinzelte Anwender und Auditoren, nicht aber der Systemadministrator Zugriff haben sollen. Mit Vidalia wird es möglich, genau festzulegen, wer auf welche Daten in welcher Detailliertheit zugreifen darf. Technisch liegen Vidalia folgende Konzepte zugrunde:

- Datenbezogene Sicherheit (mit Datenverschlüsselung basierend auf Policies)
- Federated Access (bei dem die Zugriffskontrolle vom Speicher getrennt ist)
- Audit as a Service (zur gesicherten Bereitstellung von Audit-Informationen)

Vidalia liegt als Zugriffsschicht über SQL Azure. Zugriffe auf die Datenbank werden über Vidalia abgewickelt.

Änderungen der AppFabric (ehem. .NET Services)

Die .NET Services haben mit der Produktivsetzung eine sofort ins Auge fallende Änderung erfahren. Sie wurden umbenannt. Die neue Bezeichnung für diese Dienstgruppe ist *Windows Azure AppFabric*. Die AppFabric umfasst die beiden Dienste

- *Windows Azure AppFabric Service Bus*
- *Windows Azure AppFabric Access Control Service*

Im Folgenden werden diese beiden Dienste kurz als *Service Bus* und *Access Control Service* bezeichnet. Der Workflow Service wird in zukünftigen Versionen der AppFabric enthalten sein.

An der Positionierung der AppFabric hat sich nichts geändert: Sie umfasst Cloud Services, mit deren Hilfe Entwickler Anwendungen und Dienste, die auf Windows Azure, Windows Server und einer Reihe von anderen Plattformen betrieben werden, verknüpfen. Die Kommunikation kann dabei über den Service Bus abgewickelt und – sofern es sich um REST-basierte Services handelt – Claims-basiert über den Access Control Service abgesichert werden.

Service Bus

Der Service Bus behält seine zentrale Rolle in der Vernetzung verteilter Anwendungskomponenten. Folgende primäre Anwendungsmuster werden unterstützt:

- *Eventing*, d.h. der Austausch von Benachrichtigungen zwischen Anwendungen über verschiedene Kommunikationsprotokolle
- *Service Remoting*, d.h. das Angebot von lokal beim Anwender (hinter einer Firewall) betriebenen Services in der Cloud
- *Tunneling*, d.h. die Kommunikation zwischen Anwendungen über Firewall-Grenzen hinweg

Gegenüber dem Juli-CTP wurde der Funktionsumfang des Service Bus leider in einem entscheidenden Bereich reduziert: Router und Queues wurden aus dem Service Bus entfernt (diese sollen in einem zukünftigen Release mit erweiterten Möglichkeiten wieder verfügbar werden). Entsprechende

Abschnitte in diesem Buch sind also zunächst hinfällig.

Im SDK der Produktivversion ist das Beispiel *LoadBalance* enthalten, das zeigt, wie mit den bestehenden Mitteln des Service Bus eine ähnliche Funktionalität wie die durch Router zuvor bereitgestellte erzielt werden kann.

Die Queues wurden vorübergehend durch *Message Buffers* ersetzt. Diese bieten einen großen Teil der zuvor von Queues bereitgestellten Funktionalitäten an. Message Buffers sind in den folgenden Bereichen gegenüber Queues eingeschränkt:

- Ein Message Buffer kann maximal 1 MB Daten enthalten
- Eine Dequeue-Operation liefert immer nur eine Nachricht zurück
- Die Overflow-Policy ist auf `reject` beschränkt
- Message Buffer sind ausschließlich über REST ansprechbar

Darüber hinaus sind die Konfigurationsmöglichkeiten gegenüber Queues eingeschränkt. Das grundsätzliche Verhalten (»best effort FIFO«) ist allerdings vergleichbar.

Access Control Service

Der Access Control Service (ACS) wurde funktional ebenfalls beschränkt und fokussiert nun auf die Claims-basierte Zugriffskontrolle für REST-basierte Webservices. Vorübergehend wurden also Funktionen für Single-Sign-On oder WS-* Unterstützung aus dem Dienst entfernt und für spätere Releases des ACS angekündigt.

Die Funktionalitäten des ACS umfassen unter anderem folgende Bereiche:

- Claims-basierte Zugriffskontrolle für REST-basierte Webservices
- Integrationsmöglichkeit mit ADFS v2
- Unterstützung von WRAP (= *Web Resource Authorization Protocol*) und SWT (= *Simple Web Token*)

Der Prozess zum zugriffsgesicherten Aufruf einer durch ACS geschützten Ressource durch einen Client erfolgt wie in **Abbildung 10.9** dargestellt.

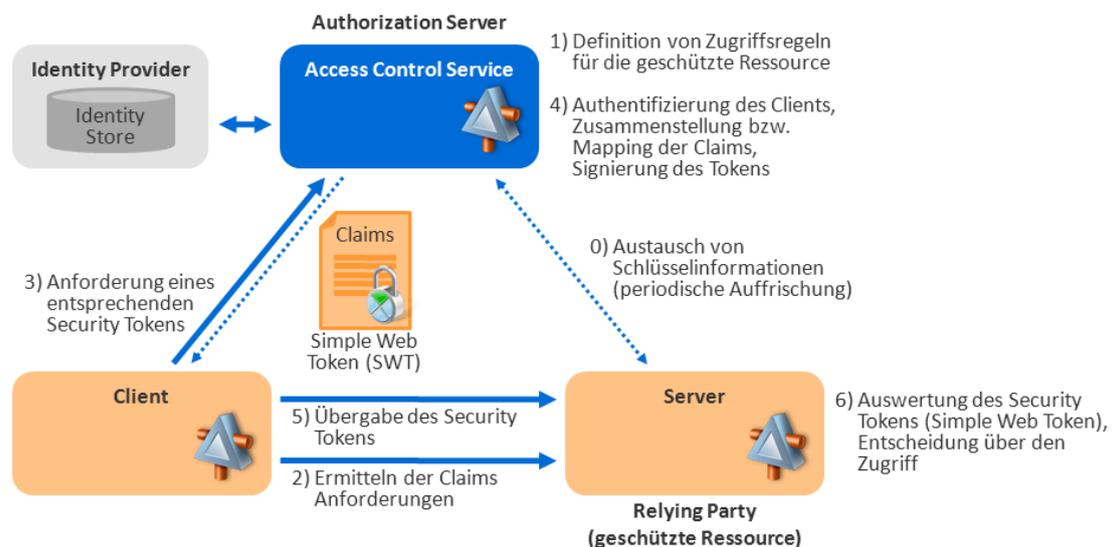


Abbildung 10.9: Claims-basierte Zugriffskontrolle in OAuth Terminologie

Der ACS stellt dem Client also Security Tokens als sogenannte *Simple Web Tokens* zur Verfügung, das der Client beim Zugriff auf die geschützte Ressource übergeben kann und das diese dann für die Entscheidung über den Zugriff auswerten kann.

Der Client kann drei Arten von Tokens anfordern:

- Plaintext (einfachste Variante, ohne Verschlüsselung)
- Signierte Token (HMAC SHA 256 erforderlich)
- Von AD FS v2 ausgegebene Token, die SAML enthalten (erlaubt Unternehmensintegration)

ACS liefert Tokens immer im SWT-Format zurück. Diese Token haben einen Aufbau ähnlich dem in **Listing 10.8** gezeigten Token.

```
role=Admin%2cUser&
customerName=Contoso%20Corporation&
Issuer=https%3a%2f%2fadatum.accesscontrol.windows.net%2fWRAPv0.8&
Audience=http%3a%2f%2fadatum%2fbillprint&
ExpiresOn=1255912922&
HMACSHA256=yuVO%2fwc58%2ftYP36%2fDM1mS%2fHr0hswpsGTWwgfVAbpL64%3d
```

Listing 10.8: Beispiel für das SWT-Format

Ausblick

Microsoft plant, Funktionen, die in der Produktivversion der AppFabric gegenüber früheren CTP-Versionen entfernt wurden, in zukünftigen Releases der Plattform wieder einzuführen. Hierzu gehört beim Service Bus insbesondere die Verfügbarkeit von Routern und Queues.

Für den Access Control Service sind unter anderem folgende Erweiterungen angekündigt:

- Web Single-Sign-On
- WS-* Unterstützung (WS-Trust und WS-Federation)
- Unterstützung verschiedener Web Identity Provider (Live ID, Facebook Connect, Google, Open ID, etc.)
- Erweiterte Unterstützung von Enterprise Identity Providern
- Integration von CardSpace

Änderungen der Live Services

Was die Live Services betrifft, so war die Änderung im Bezug auf die Windows Azure Plattform am einschneidendsten: Die Live Services wurden aus der Plattform komplett herausgelöst und werden in Zukunft unabhängig von Azure weiterentwickelt.

Neue Dienste

Neben Erweiterungen an den Kerndiensten der Windows Azure Plattform (Windows Azure, AppFabric und SQL Azure) hat Microsoft noch eine Reihe von ergänzenden Diensten veröffentlicht, die Anwender der Plattform dabei unterstützen, auf Windows Azure basierende Cloud Services zu vermarkten und zu integrieren.

Microsoft Pinpoint

Microsoft stellt mit Microsoft Pinpoint Kunden und Partnern einen Marktplatz für Cloud-Anwendungen, die sie auf Windows Azure betreiben, zur Verfügung. Partner haben die Möglichkeit, Ihre Services

(sowohl Anwendungsdienste als auch Dienstleistungen) über das Developer Portal in diesen Marktplatz einzustellen und mit Metainformationen (Dienstbeschreibung, Kosten etc.) zu versehen. Kunden können diese Anwendungen und Dienste in Pinpoint suchen, auswählen, und direkt aus den Microsoft-Rechenzentren beziehen. **Abbildung 10.10** zeigt die Hauptseite von Microsoft Pinpoint.



Abbildung 10.10: Microsoft Pinpoint – Marktplatz für Cloud Services

Über das Pinpoint Portal besteht die Möglichkeit, Einträge nach Bewertung, regionaler Verfügbarkeit und weiteren Kriterien zu filtern.

Microsoft DataMarket

Über Microsoft DataMarket können Entwickler und Anwender Datendienste über die Windows Azure Plattform bereitstellen, auffinden, konsumieren und abrechnen. DataMarket agiert dabei als technische und vermittelnde Datendrehscheibe, indem es eine Plattform zur technischen Bereitstellung der Datendienste, Abrechnungsfunktionalitäten und standardisierte Schnittstellen für den Zugriff zur Verfügung stellt.

DataMarket umfasst folgende Funktionsbereiche:

- Rich Web Services

Den Datendiensten liegt jeweils ein REST-basiertes Zugriffsmodell zugrunde. Darüber hinaus sind für die meisten Dienste auch ATOM 1.0 Feeds verfügbar. Die Dienstenutzung wird über einheitliche Verfahren erfasst und abgerechnet.

- Service Explorer

Mit diesem können C#-Proxy-Klassen generiert werden, die bei den Service-Aufrufen und der Darstellung der Daten unterstützen. Die generierten Klassen enthalten eine Kurzdokumentation des betreffenden Dienstes und Beispielwerte für die Aufrufe.

- Marktplatzintegration und Portal zur Verwaltung

Über den Marktplatz können Daten gefunden, Zugriffe eingerichtet, verwaltet und abgerechnet werden. Das Portal ermöglicht die Erstellung von Zugriffsstatistiken.

Über DataMarket können quasi beliebige Arten von Inhalten bereitgestellt werden: Blobs, strukturierte Daten, Webservices etc. **Abbildung 10.11** zeigt den schematischen Aufbau von Microsoft DataMarket aus Entwicklersicht.

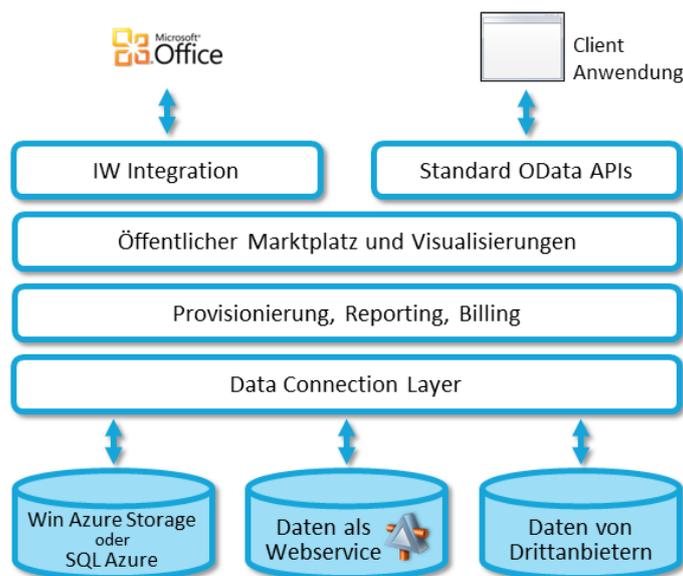


Abbildung 10.11: Struktur des Microsoft Data Market

Die Daten können aus verschiedenen Quellen stammen. Daten aus Windows Azure Storage und SQL Azure können unmittelbar integriert werden. Für Daten, die als Webservice bereitgestellt werden, stehen generische Webservice Konnektoren bereit. Daten von Drittanbietern können über eigene Konnektoren eingebunden werden. Der Datenkonsum kann schlussendlich über eine OData API erfolgen oder – im Fall von Office – über eine entsprechende IW Integrationsschicht.

Microsoft hat bereits einige Ergänzungen und Weiterentwicklungen angekündigt:

- Einheitliche EULA für die Dienstnutzung
- Vorschaufunktionen für Videos, Bilder, 3D-Modelle
- Integrationspunkte für Microsoft Office, SQL Server, SQL Azure-Datenbank und andere Systeme

Zusammenfassung

Die Windows Azure Plattform enthält in Version 1.3 Windows Azure, die Windows Azure AppFabric (ehemals .NET Services) und SQL Azure.

Mit Produktivsetzung hat Microsoft Änderungen und Erweiterungen an den einzelnen Diensten

vorgenommen sowie Preismodelle eingeführt. Zum Teil sind neue Funktionen hinzugekommen (z.B. Logging und Diagnose bei Windows Azure), zum Teil sind aber auch Funktionen entfallen, deren Wiedereinsetzung für zukünftige Versionen der Plattform angekündigt sind (z.B. Queues und Router beim Service Bus, WS-* Unterstützung beim Access Control Service).

Hinzu gekommen sind mit Microsoft Pinpoint und Microsoft DataMarket zwei Dienste, die Entwickler und Anwender bei der Vermarktung, Nutzung und Integration von Daten und Services unterstützen.