

## Kapitel 2

# Überblick über die Windows Azure Plattform

### **In diesem Kapitel:**

Die Windows Azure Plattform	48
Bestandteile der Plattform	51
Bedeutung der Plattform	60
Schrittweise Adaption von Azure	65
Softwareentwicklung mit Azure	66
Zusammenfassung	82

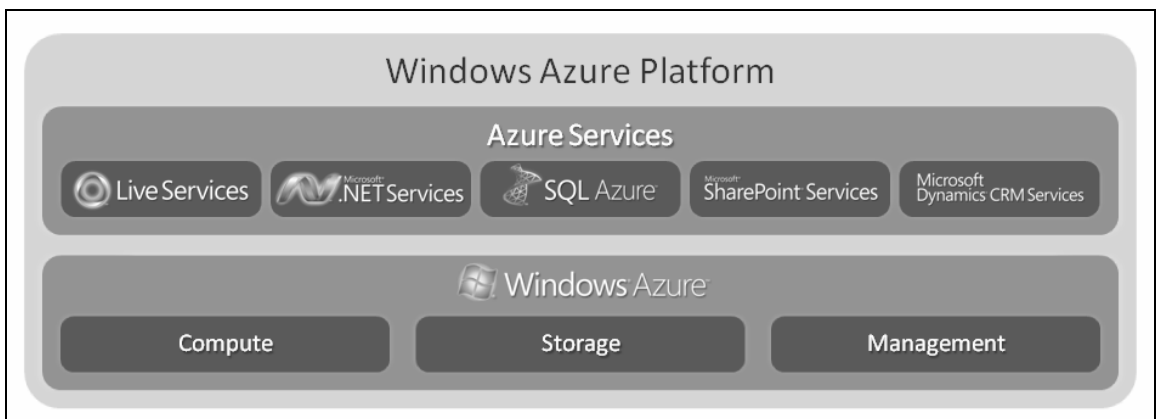
Anwender und Entwickler wünschen sich maximale Flexibilität hinsichtlich des Aufbaus und Betriebs von IT-Funktionen. Sie wollen sich individuell entscheiden können, welche Anwendungen bzw. Teile von Anwendungen vor Ort auf verschiedenen Servern und Endgeräten, bei Hostern oder in der Cloud betrieben werden. Darüber hinaus sollen Lösungen, die sich über verschiedene Betriebsarten erstrecken, einfach integriert werden können. Diesem Wunsch entspricht Microsoft, indem es eine Plattform anbietet, die für Anwendungen und Anwendungskomponenten eine Wahlfreiheit des Betriebsmodells bietet und aufgrund der Durchgängigkeit der Technologie und der konsequenten Unterstützung von Standardschnittstellen die Integrationen verteilter IT-Funktionen erleichtert. Azure ist der Teil der Gesamtplattform, der Cloud Computing auf Basis der Microsoft-Technologie ermöglicht. Dieses Kapitel beschreibt die Windows Azure Plattform genauer, gibt einen Überblick über die einzelnen Bestandteile und geht auf deren Relevanz und mögliche Einsatzszenarien für Anwender und Entwickler ein.

## Die Windows Azure Plattform

Die Windows Azure Plattform ist eine hoch skalierbare Cloud-Plattform, die in Microsofts Rechenzentren betrieben wird. Auf Basis eines speziellen Betriebssystems – Windows Azure – stellt sie Infrastruktur- und Anwendungsdienste bereit, die einzeln oder in Kombination verwendet werden können.

### Microsofts Plattform für die Cloud

In Abbildung 2.1 ist der grobe Aufbau der Windows Azure Plattform, kurz: Azure, zu sehen. Basis der Plattform ist Windows Azure, Microsofts Betriebssystem für die Cloud, das grundlegende Speicher-, Rechen- und Managementdienste zum Betrieb von Windows-Anwendungen bereitstellt. Auf Windows Azure setzen eine Reihe von Cloud Services auf, die sich in fünf Funktionsgruppen, die jeweils unterschiedliche Anwendungsaspekte abdecken, unterteilen. Auf diese Dienste können Entwickler zur Verwendung in eigener Software über Standardschnittstellen zugreifen.



**Abbildung 2.1** Die Windows Azure Plattform

Auf Windows Azure setzen Dienste folgender Gruppen auf:

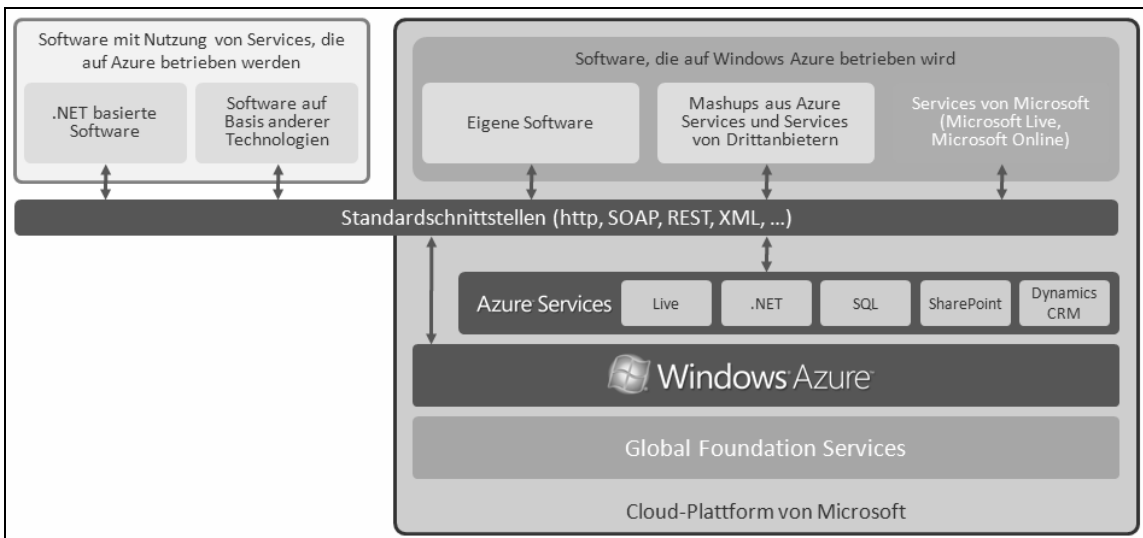
- Live Services – Anwendungsdienste, die sich für Software im Bereich des *Social Computing* oder *Enterprise 2.0* eignen, bei denen Informationen von vernetzten Anwendern zwischen verschiedenen Endgeräten, kollaborativen Anwendungen und Anwender-Communitys ausgetauscht und synchronisiert werden sollen.
- .NET Services – Infrastrukturdienste zur Integration verteilter Anwendungskomponenten, die in der Cloud oder auf lokal beim Anwender betriebenen Systemen bereitgestellt werden.
- SQL Azure – Datenbankdienste, die auf Basis von SQL Server Informationsspeicherung in der Cloud ermöglichen.
- SharePoint Services und Dynamics CRM Services – Anwendungsdienste für kollaborative Geschäftsanwendungen und Software für das Kundenbeziehungsmanagement.

**HINWEIS** Zum Zeitpunkt der Erstellung dieses Buchs waren die SharePoint Services und Dynamics CRM Services noch nicht verfügbar. Die Dienste der anderen Gruppen waren im Rahmen einer *Communiy Technology Preview (CTP)* einer eingeschränkten Entwicklergemeinde zugänglich.

Ein Zugriff auf diese Dienste kann sowohl von Anwendungen aus erfolgen, die selbst auf Azure betrieben werden, als auch von externen Anwendungen. Letztere können die Dienste dann über Standardschnittstellen nutzen.

## Möglichkeiten mit Azure

Als Microsofts Cloud-Plattform ermöglicht Azure den Einsatz einer hoch skalierbaren Infrastruktur zur Nutzung bestehender bzw. für das Angebot eigener Cloud Services. Für Entwickler ergibt sich somit zum einen die Möglichkeit, Anwendern die Cloud als dritte Betriebsalternative (neben dem Betrieb vor Ort und dem Betrieb bei einem Hosting Partner) anzubieten. Zum anderen steht mit Azure auch eine Reihe von Services zur Verfügung, die wiederum von Entwicklern zur Anreicherung ihrer eigenen Anwendungen genutzt werden können.



**Abbildung 2.2** Möglichkeiten der Windows Azure Platform für Entwickler

Die Kommunikation zwischen Software – unabhängig davon, ob sie vor Ort beim Anwender, bei einem Host oder auf Azure betrieben wird – und Services, die auf Azure bereitgestellt werden, erfolgt über Standardtechnologien, die sich im Bereich des Internets bewährt haben. Abbildung 2.2 skizziert die verschiedenen Möglichkeiten, die sich bei Einsatz von Azure für Entwickler bieten:

- Entwicklung von Software für Azure

Windows Azure stellt eine Ausführungsumgebung für den Betrieb eigener Software in der Cloud zur Verfügung. Für diesen Betrieb ist es erforderlich, die Anwendung in geeigneter Art und Weise serviceorientiert zu strukturieren und die Services lose miteinander zu koppeln, die es Windows Azure erlauben, die Anwendung autonom zu verwalten, zu überwachen und die für die Ausführung benötigten Speicher- und Rechenressourcen zuzuordnen.

Wenngleich Windows Azure für die Anwendungsarchitektur gewisse Vorgaben macht, ist die Entwicklung für Azure sehr ähnlich zur Entwicklung klassischer ASP.NET-basierter Webanwendungen. Bei der Konzeptionierung war die Möglichkeit der Nutzung bestehender Programmierkenntnisse zentrales Ziel. Entwickler, die Erfahrung in ASP.NET besitzen, benötigen deshalb nur minimalen Umstellungsaufwand, um Anwendungen für Windows Azure entwickeln zu können.

- Erweiterung von Anwendungen durch Nutzung von Azure Services

Nicht immer ist eine Neuentwicklung oder vollständige Migration einer Anwendung zum Betrieb auf Azure möglich oder gewünscht. Azure Services können aus beliebigen Anwendungen heraus genutzt werden. Aufrufe und Kommunikation mit den Services können über Standardschnittstellen wie SOAP oder REST erfolgen. Hierüber ist es möglich, punktuell von den Vorteilen des Cloud Computing zu profitieren, ohne dass notwendigerweise die gesamte Software in der Cloud betrieben wird. Dies kann beispielsweise dann der Fall sein, wenn für bestimmte Teile einer Anwendung hohe Skalierbarkeit – und damit kostengünstigste Bereitstellung – bei Speicher- und Rechenleistung erforderlich sind. So könnten beispielsweise bestehende Anwendungen zur kostengünstigen Speicherung großer Datenmengen anstelle einer lokalen Datenbank SQL Azure nutzen.

- Verknüpfung von Services in Azure

Je weiter das bestehende Angebot von Azure Services von Microsoft durch Services von Drittanbietern ergänzt wird, desto leichter wird es in Zukunft möglich sein, neue IT-Lösungen rein durch Neukombination vorhandener Services in Azure zu erzeugen, ohne eine Integrationskomponente außerhalb von Azure zu verwenden. Die Erstellung derartiger *Mashups* wird durch die strikte Serviceorientierung und die Einhaltung offener, standardisierter Programmierschnittstellen in Azure erleichtert.

- Interoperabilität über Standardschnittstellen

Durch die konsequente Bereitstellung von Funktionalitäten der Windows Azure Plattform über Schnittstellen wie SOAP, REST und Standard-XML öffnet Microsoft diese Plattform allen Technologien, die diese Schnittstellen unterstützen. So ist es möglich, Azure-Funktionalitäten aus einer Java-, PHP- oder Ruby-basierten Software zu nutzen. Tatsächlich bietet Microsoft sogar Software Development Kits (SDKs) für eben diese Sprachen an, um dort von den Möglichkeiten der Microsoft-Cloud zu profitieren.

Diese verschiedenen Möglichkeiten bringen Entwicklern ein hohes Maß an Flexibilität, ganze Anwendungen oder auch nur Teile von Anwendungen für den Betrieb auf Azure vorzusehen, um für die Gesamtanwendung oder punktuell in einzelnen Bereichen von den speziellen Vorteilen des Cloud Computing, wie hohe Skalierbarkeit, weltweite Verfügbarkeit, kostengünstiger Betrieb etc., zu profitieren. Diese Wahlfreiheit, sich je nach Anforderungen für einzelne Anwendungsteile zwischen einem klassischen Betrieb vor Ort oder einem Betrieb in der Cloud entscheiden zu können, ist bezeichnend für Microsofts Software plus Services-Strategie.

## Entwicklungsstand der Plattform

Zum Zeitpunkt der Erstellung dieses Buchs ist die Windows Azure Plattform im Rahmen einer Community Technology Preview (CTP) kostenlos einer eingeschränkten Entwicklergemeinde zugänglich. Hauptziel dieser Phase ist die Vorbereitung Interessierter auf diese neue Plattform und die Berücksichtigung der Rückmeldungen dieser Interessierten für die Weiterentwicklung von Azure. Die CTP-Version ist in einigen Bereichen mit Einschränkungen belegt:

- Noch nicht alle Funktionen und Services sind verfügbar und die vorhandenen in ihrer Funktionalität teilweise eingeschränkt.
- Für das Preismodell sind nur grobe Aussagen bekannt. Die Nutzung von Services soll über verschiedene Lizenzmodelle, nutzungsabhängige Abrechnung bzw. werbebasierte Modelle bezahlt werden können.
- Ähnliche Unsicherheiten bestehen bezüglich der späteren Service Level wie konkreter Verfügbarkeit, Kompensationszahlungen im Fehlerfall etc.

Wenngleich diese Unwägbarkeiten konkrete Projektplanungen stark erschweren, kann diese Testphase bereits genutzt werden, um sich mit den grundsätzlichen technischen Möglichkeiten der Windows Azure Plattform auseinanderzusetzen und schon heute bei den Vorbereitungen für zukünftige Entwicklungsarbeiten Berücksichtigung zu finden.

## Bestandteile der Plattform

Die folgenden Abschnitte geben einen Überblick über die Bestandteile der Windows Azure Plattform, gehen auf die Funktionalitäten der einzelnen Bestandteile ein und zeigen beispielhaft auf, in welchen möglichen Einsatzszenarien die beschriebenen Services sinnvoll eingesetzt werden können, um einen Eindruck über den Nutzen der Plattform zu vermitteln.

### Global Foundation Services

Das technische Fundament, auf die die Windows Azure Plattform aufsetzt, bezeichnet Microsoft als *Global Foundation Services*. Es handelt sich dabei um die technische Infrastruktur, die in Microsofts Rechenzentren vorgehalten wird und Rechen- und Speicherleistung zur Verfügung stellt, die zur Ausführung der Azure Services sowie der auf Azure betriebenen Anwendungen benötigt wird. Dabei wird eine große Anzahl standardisierter Server eingesetzt, auf denen unter anderem eine spezielle Version von Windows Server Hyper-V sowie eine speziell optimierte Version von SQL Server Datacenter Edition betrieben werden, die jeweils für eine homogene Server-Infrastruktur, wie sie in den Rechenzentren von Microsoft besteht, optimiert sind. All diese Bestandteile sind auf horizontale Skalierbarkeit (Scale-out) ausgelegt.

Die gesamte Infrastruktur wird zu einer so genannten *Fabric*, d.h. zu einem großen Ganzen, zusammengefasst, deren innere Komplexität durch eine Virtualisierungsschicht für Anwender und Entwickler verborgen bleibt. Tatsächlich erhalten Nutzer der Windows Azure Plattform jeweils ihre eigenen virtuellen Umgebungen, auf die sie nur über eine webbasierte Managementkonsole bzw. Services-Schnittstellen zugreifen können. Ein Zugriff auf interne Mechanismen der Plattform bleibt Anwendern verwehrt. Somit entsprechen auch die *Global Foundation Services* dem Dienstleistungsgedanken. Die Funktionen der Dienste werden nur über Service Level und die Zugriffsschnittstellen beschrieben, die innere Arbeitsweise zur Dienstbringung ist für den Dienstanutzer, d.h. Anwender und Entwickler, irrelevant.

## Windows Azure

Auf die Infrastruktur setzt Windows Azure auf. Der Name ist nicht zufällig so gewählt. Windows Azure übernimmt in der Microsoft-Cloud die Funktionen, die in klassischen Betriebsmodellen (Betrieb vor Ort beim Anwender bzw. bei einem Host) von Windows Client bzw. Windows Server übernommen werden. Windows Azure ist das Betriebssystem für die Microsoft-Cloud und bietet neben Speicher und Rechen-diensten eine Ausführungs- und Managementumgebung für Anwendungen und Dienste in Microsofts Rechenzentren.

Windows Azure erlaubt sowohl die Ausführung von .NET Managed Code, d.h. auf Basis des .NET Framework können Anwendungen für Azure entwickelt werden, als auch von Unmanaged Code. So können beispielsweise sowohl ASP.NET-Anwendungen und Windows Communication Foundation (WCF) Services für den Betrieb auf Azure erstellt werden als auch Dienste, die auf anderen Technologien basieren, aber auf einer Windows-Umgebung ausführbar sind.

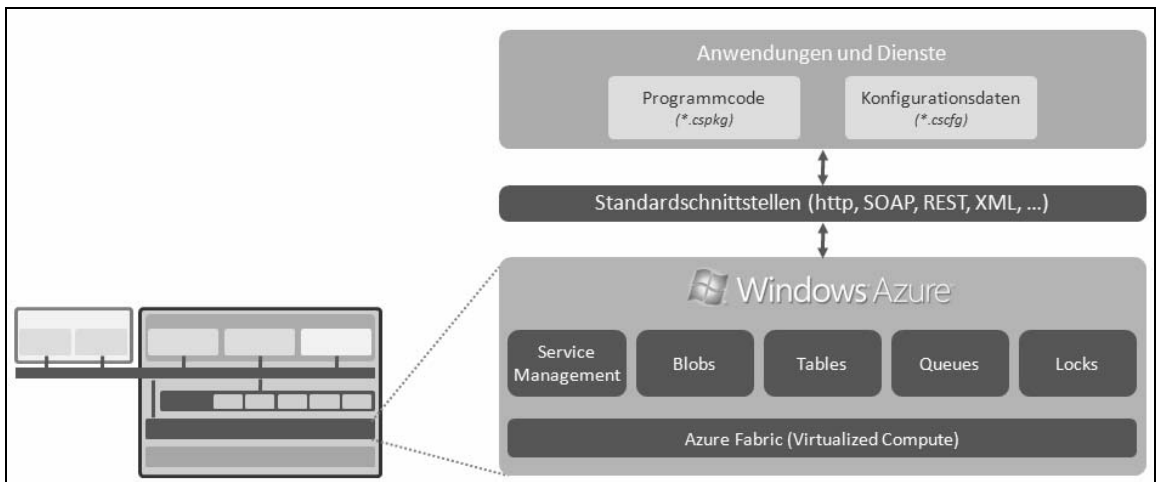


Abbildung 2.3 Windows Azure

Anwendungen, die auf Azure betrieben werden sollen, können unter Verwendung der in den jeweiligen Software Development Kits (SDKs) enthaltenen Projektvorlagen entwickelt werden. Diese helfen dabei, den für Azure erforderlichen Anwendungsaufbau einzuhalten. Wenngleich die Entwicklung für Azure Services sehr ähnlich zur Entwicklung klassischer .NET-basierter Anwendungen ist, müssen doch gewisse Regeln eingehalten werden, auf die in diesem Buch an späterer Stelle eingegangen wird. Es ist nicht möglich, eine bestehende lokal ausführbare Datei (\*.exe) ohne Änderung auf Azure zu installieren.

Zur Entwicklung von Azure Services kann Visual Studio in der kommerziellen oder in den kostenfrei erhältlichen Express Edition verwendet werden. Entwickler können also eine vertraute Entwicklungsumgebung einsetzen und benötigen nur geringen Einarbeitungsaufwand bei der Verwendung der Projektvorlagen der Azure-SDKs. Mit diesen können insbesondere Webanwendungen und zugehörige Hintergrundprozesse für Azure implementiert werden.

## Funktionalitäten

Windows Azure stellt eine Reihe von grundlegenden Infrastrukturdiensten zur Verfügung, die in Abbildung 2.3 aufgeführt sind:

### ■ *Azure Fabric*

Diese Technologie dient als Abstraktionsschicht, um die zugrunde liegende physische Infrastruktur zu virtualisieren und dem Anwender jeweils eigene virtuelle Umgebungen bereitzustellen, um dort isoliert von anderen Anwendern Services zu betreiben, zu verwalten und zu nutzen. Die Azure Fabric verwaltet und überwacht installierte Services und stellt sicher, dass sich diese entsprechend der vom Entwickler bereitgestellten Konfiguration verhalten. Weist dieses Verhalten Unregelmäßigkeiten auf oder antwortet ein Service nicht auf Überwachungsanfragen, kann der Service automatisch neu gestartet werden. Dies erfolgt vollautomatisch.

### ■ *Service Management*

Über eine Managementkonsole können Services, die auf Azure betrieben werden, verwaltet, d.h. installiert, konfiguriert, gestartet und beendet werden. Dabei kann das Verhalten des Service spezifiziert werden, das von der Fabric überwacht wird.

### ■ *Blobs*

Windows Azure bietet die Möglichkeit, große Datenobjekte (*binary large objects* = BLOBs) zu speichern.

### ■ *Tables*

Strukturierte Daten können in Tabellenstrukturen abgelegt werden. Dabei entsprechen die Speichermöglichkeiten von Windows Azure eher einem klassischen Dateisystem als einer Datenbank im engeren Sinn. Datenbankfunktionalitäten werden auf Azure über SQL Azure angeboten.

### ■ *Queues*

Warteschlangen in Windows Azure können dazu verwendet werden, asynchrone Kommunikation zwischen Services umzusetzen und darüber eine Entkoppelung von Diensten zu erreichen.

### ■ *Locks*

Schließlich besteht die Möglichkeit, Objekte in Windows Azure für die Verwendung zu sperren. Dies ist insbesondere dann sinnvoll, wenn parallel laufende Services auf gemeinsame Objekte zugreifen.

Die einzelnen Dienste sind über Standardschnittstellen (http, SOAP, REST, ...) zugreifbar. Die Managementkonsole zur Verwaltung von Anwendungen, die auf Windows Azure betrieben werden, kann über einen normalen Browser mittels HTTP aufgerufen werden.

## Mögliche Einsatzszenarien

Die Rechen- und Speicherdienste von Windows Azure bieten eine Reihe von Möglichkeiten, von den Vorteilen einer hoch skalierbaren Cloud-Infrastruktur, wie Azure sie bietet, zu profitieren.

### ■ Betrieb eigener Webanwendungen und Dienste in der Cloud

Auf Windows Azure können eigene Dienste installiert werden. Dies können spezielle ASP.NET-basierte Webanwendungen oder WCF Services sein. Diese können mithilfe von Azure-SDKs entwickelt werden. Ein Betrieb auf Windows Azure bietet den Vorteil, dass damit der Anwendung bzw. dem Dienst eine Infrastruktur zugrunde liegt, die weltweite Verfügbarkeit ermöglicht und theoretisch höchste Zugriffs-

zahlen bedienen kann. Ein Betrieb auf Azure kann auch dann sinnvoll sein, wenn hohe Last nur zu bestimmten Zeitpunkten zu erwarten ist, für die dauerhaft keine entsprechende Umgebung vorgehalten werden soll.

- Nutzung von Queues für den asynchronen Daten- und Informationsaustausch  
Die Warteschlangen-Infrastruktur von Azure kann dazu verwendet werden, die asynchrone Kommunikation zwischen Anwendungen zu implementieren.

Wenngleich diese Einsatzszenarien auch mit Softwaresystemen mit klassischen Betriebsformen (z.B. mit einem von einem Hostler betriebenen Webserver) umgesetzt werden können, bietet der Einsatz von Windows Azure die Möglichkeit, die speziellen Fähigkeiten der Azure Cloud-Infrastruktur wie flexible Skalierbarkeit zu nutzen.

## Azure Services

Mit den Azure Services stellt Microsoft selbst eine Reihe von Infrastruktur- und Anwendungsdiensten zur Nutzung durch Entwickler auf Basis von Windows Azure zur Verfügung. Durch Bereitstellung der Funktionalitäten über Standardschnittstellen (HTTP, SOAP, REST, XML) können diese Dienste von beliebigen Technologien, die diese Standards unterstützen, genutzt werden. Wie in Abbildung 2.1 zu sehen, setzen auf Windows Azure fünf Gruppen von Services auf:

- Live Services
- .NET Services
- SQL Azure
- SharePoint Services
- Dynamics CRM Service

Diese werden in den folgenden Abschnitten näher beschrieben. Zusammen mit Windows Azure bilden sie die Windows Azure Plattform, die wiederum auf die Global Foundation Services aufsetzt.

### Live Services

Microsoft verwendet die Marke *Live* für verschiedene Dienste, die sich an Endkunden bzw. Privatkunden richten. Diese Zielgruppe zeichnet sich durch hohe Technikaffinität aus, die sich unter anderem im Besitz verschiedenartigster IT-Endgeräte (PC, Laptop, Mobiltelefon, Spielekonsole, MP3-Player etc.) und der Bereitschaft, sich an so genannten *Social Communitys* zu beteiligen, niederschlägt.

Die Live Services bieten Anwendungsdienste, die diesem Trend Rechnung tragen, indem sie die Erstellung, Verarbeitung und Synchronisation von anwendungs-, geräte- und benutzerbezogenen Informationen ermöglichen. Abbildung 2.4 skizziert den Aufbau der Live Services. Herzstück der Live Services ist das so genannte *Live Operating Environment* (LOE). Diese spezielle Umgebung läuft zum einen auf Azure in der Cloud, zum anderen kann es auf Umgebungen, d.h. diversen Endgeräten von Anwendern, außerhalb von Azure – derzeit werden die externen Betriebssysteme Windows XP, Windows Vista, Windows 7, Windows Mobile 6 sowie Mac OS X unterstützt – installiert werden. Auf Daten, die über die Live Services verwaltet werden, kann über das LOE, HTTP sowie ATOM- oder RSS-Feeds zugegriffen werden.



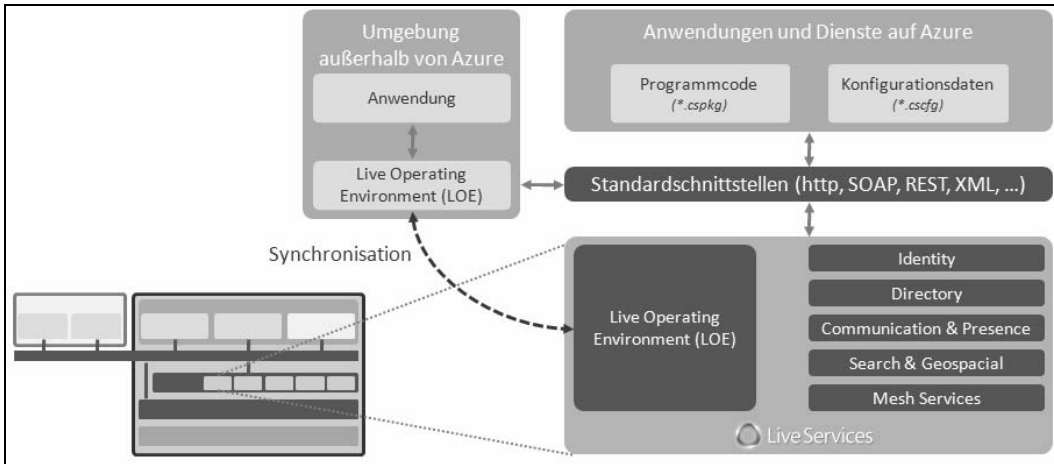


Abbildung 2.4 Live Services

Durch Installation des LOE können Anwender ihre unterstützten Endgeräte zu einem persönlichen Geräteverbund, dem so genannten *Mesh* verbinden. Die verschiedenen LOE-Instanzen ermöglichen eine Daten- und Informationssynchronisation über die verschiedenen Geräte hinweg, wobei die Anwender festlegen können, welche Daten zwischen den einzelnen Geräten synchronisiert werden sollen und wo jeweils lokale Replikate der Daten angelegt werden sollen, um im Falle fehlender Netzverbindung zu Azure Zugriff auf die lokal gespeicherten Daten zu haben. Alles Weitere wird durch die LOE-Instanzen geregelt. Es besteht auch die Möglichkeit, Daten anderen Anwendern zur Verfügung zu stellen, die diese über entsprechende LOE-Installationen in ihr eigenes Mesh aufnehmen können.

## Funktionalitäten

Die Live Services umfassen eine Reihe von Diensten, die in Anwendungen, die Funktionen zur Speicherung, Verarbeitung und Synchronisation von anwenderbezogenen Daten über verschiedene Geräte hinweg benötigen, eingesetzt werden können:

- *Identity (Live ID)*  
Hierüber kann die Authentifizierung von Anwendern erfolgen. Basis ist die bekannte Windows Live ID, die bereits auf vielen Webseiten als Authentifizierungsmechanismus dient.
- *Directory*  
Dieser Dienst stellt weitere, detailliertere Informationen zu Anwendern bereit, die personalisierte Anwendungslogik ermöglicht.
- *Communication & Presence (Live Messenger)*  
Über diesen Dienst kann auf Funktionalitäten des Live Messengers zugegriffen werden. So kann beispielsweise die Kommunikation mit anderen Anwendern erfolgen oder der Präsenzstatus in Anwendungen angezeigt werden.
- *Search (Live Search)*  
Suchergebnisse können über Live Search ermittelt und verarbeitet werden. Dabei kommt Microsofts Suchtechnologie zum Einsatz.

- *Geospatial (Live Maps)*

Microsofts Geodaten-Technologie Virtual Earth kann über den Live Maps Service eingesetzt werden. Hierüber können Karteninformationen (Positionen, Wegbeschreibungen etc.) genutzt werden.

- *Mesh Services*

Um spezielle Funktionen zur Daten- und Informationssynchronisation über Anwendungen und Geräte eines Anwender-Meshs zu nutzen, stehen die Mesh Services zur Verfügung.

Das Live Operating Environment bietet einen einheitlichen Zugriff auf die Live Services. Zum Teil können die Services aber auch über alternative Wege, die in Kapitel 4 beschrieben werden, aufgerufen werden.

## Mögliche Einsatzszenarien

Live Services bieten eine Vielzahl von Möglichkeiten zur Erstellung von Anwendungen, für die anwender- und gerätebezogene Daten verarbeitet werden müssen.

- Datensynchronisation von Informationen über Geräte eines Anwenders hinweg

Das Live Operating Environment bietet im Zusammenspiel mit den Mesh Services eine ideale Grundlage zur Entwicklung von Anwendungen, bei denen Anwender Informationen und Daten zwischen ihren Endgeräten synchronisieren und hierüber auch *offline* mit ihren Daten arbeiten können.

- Nutzung von Services aus (Web-) Anwendungen heraus

Live Services können, wie die anderen Services auch, aus eigenen Anwendungen heraus verwendet werden. Webanwendungen können beispielsweise über den Verzeichnisdienst personalisiert und um Benutzerinformationen (z.B. Präsenzstatus) angereichert werden.

- Windows Live ID als Authentifizierungsmedium

Hinter der Windows Live ID steckt eine ganze Sicherheitsinfrastruktur für Millionen von Windows Live Nutzern. Diese Infrastruktur kann über Live Services in eigenen Anwendungen zur Authentifizierung von Anwendern genutzt werden.

Durch die Unterstützung diverser standardisierter Kommunikationsschnittstellen (HTTP, SOAP, REST, ATOM, RSS, ...), können die Dienste aus verschiedensten Technologien heraus, die die Standards unterstützen, eingesetzt werden.

## .NET Services

Die Entwicklung verteilter Anwendungen, bei denen Teile auf Azure und andere Teile außerhalb (z.B. vor Ort oder bei einem Hoster) betrieben werden können, bringt eine Reihe von Herausforderungen mit sich:

- Einzelne Anwendungsteile können jeweils unterschiedliche Identitäts- und Verzeichnisdienste zur Verwaltung von Berechtigungen nutzen.
- Mit steigender Zahl verteilter Dienste steigt die Komplexität von Punkt-zu-Punkt-Verbindungen, d.h. jeder Dienst kann theoretisch mit jedem verbunden werden.
- Geschützt durch Firewalls und andere Sicherheitsmaßnahmen ist ein geregelter Aufruf von vor Ort betriebenen Services von außerhalb der Firewall-Grenzen nur schwer realisierbar, ohne Sicherheitsanforderungen zu kompromittieren.

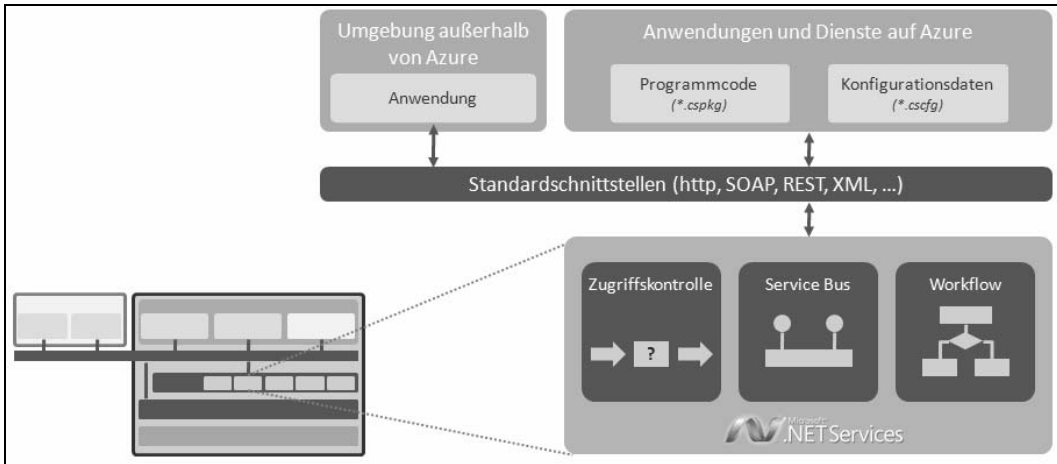


Abbildung 2.5 .NET Services

Die Vernetzung, Integration und Orchestrierung von Services ist die Domäne der .NET Services. Diese stellen Infrastrukturdienste bereit, die zur Kommunikation zwischen verteilten Anwendungskomponenten verwendet werden können.

## Funktionalitäten

In Abbildung 2.5 sind die einzelnen Dienste der .NET Services skizziert, die – wie die anderen Azure Services auch – über Standardschnittstellen zugreifbar sind, und damit aus allen Technologien heraus, die diese Standards unterstützen, verwendet werden können.

### ■ Zugriffskontrolle (*Access Control*)

Sehr vereinfacht dargestellt funktioniert die Zugriffskontrolle im Allgemeinen immer nach dem gleichen Schema: Der Aufrufer eines Service präsentiert diesem eine bestimmte Behauptung (engl. *Claim*) zu seiner Identität in Form eines *Tokens*. Für den aufgerufenen Service stellt sich nun zum einen die Frage, ob er dieser Behauptung Glauben schenkt, d.h. ihr vertraut, und ob er einem Aufrufer mit der betreffenden Identität Zugriff gewährt. Eine Herausforderung stellt sich nun dadurch, dass in IT-Systemen, deren Teile über das Internet verteilt sein können, Token in der Regel von Organisationen ausgegeben werden können, die dem aufgerufenen Service unbekannt sind. Eine Möglichkeit zur Verifikation des Tokens wäre es nun, bei jedem Aufruf ein dediziertes Trust Center aufzurufen, um die Authentizität zu bestätigen. Dieser Aufwand kann durch den Einsatz der durch die .NET Services bereitgestellten Zugriffskontrolle vermieden werden. Diese erlaubt so genanntes *Claims Mapping*. Dabei können eingehende Claims in einheitliche Ausgangs-Claims konvertiert werden, sodass der aufgerufene Service von einem bestimmten Claims-Format ausgehen kann.

### ■ Service Bus

Die Verteilung von Services bringt einige Herausforderungen mit sich: Zum einen entstehen bei direkten Aufrufen leicht komplexe, unübersichtliche Aufrufgeflechte, zum anderen ist der Aufruf eines hinter einer Firewall betriebenen Services nicht ohne Weiteres möglich. Die .NET Services bieten mit dem Service Bus einen Mechanismus, der in diesen Fällen Komplexität zu reduzieren und Aufrufe vor Ort hinter den Sicherheitsmechanismen betriebener Services ermöglicht. Services können sich beim Service Bus registrieren, sind dadurch für Aufrufer auffindbar und können auf diese Weise miteinander verknüpft werden.

### ■ *Workflow*

Neben Zugriffskontrolle und der Integration von Services bieten die .NET Services zukünftig auch die Möglichkeit, Aufrufsequenzen von Services in Form von Workflows zu definieren und diese wiederum selbst als Dienst anzubieten. Grundlage dieser Funktionalität ist die bereits von klassischen Windows-Anwendungen bekannte, im .NET Framework enthaltene Windows Workflow Foundation (WF). Tatsächlich werden Workflows in Azure auch über die XML-basierte Beschreibungssprache XOML (Extensible Object Markup Language) definiert. Der Workflow Service wird in zukünftigen Versionen der .NET Services enthalten sein.

## Mögliche Einsatzszenarien

Die Infrastrukturdienste der .NET Services bieten für die Entwicklung von Anwendungen, die aus verteilten Diensten bestehen, eine Reihe von Einsatzmöglichkeiten.

### ■ Outsourcing von *Identity Federation*

Der Einsatz der Zugriffskontrolle macht die Komplexität verteilter Identitäts- und Verzeichnisdienste beherrschbar. Durch das bereitgestellte *Claims Mapping* kann die Verarbeitung der unterschiedlichen Sicherheitsinformationen zusammengeführt und für einen über .NET Services aufgerufenen Dienst vereinheitlicht werden, sodass dieser nur noch genau festgelegte sicherheitsrelevante Informationen prüfen muss.

### ■ Angebot von Services, die vor Ort betrieben und in der Cloud angeboten werden sollen

Werden Services im Service Bus der .NET Services registriert, können Aufrufe über den Service Bus erfolgen. Dadurch, dass die Kommunikation bei diesem Vorgehen zu Beginn über den aufgerufenen Service initiiert wurde, also eine ausgehende Kommunikation erfolgte, wird die Problematik, dass eingehende Kommunikation von Firewalls in der Regel blockiert wird, gelöst. Somit müssen in der Firewall keine zusätzlichen Ports geöffnet werden.

### ■ Zugriff einer Webanwendung auf unternehmensinterne Ressourcen (z. B. SAP-Backend)

.NET Services ermöglichen es, interne Ressourcen abgesichert Externen zur Verfügung zu stellen. Dies kann auch eine auf Windows Azure betriebene Webanwendung sein, die beispielsweise SAP-Daten anzeigen möchte, die innerhalb der Firewallgrenzen des Unternehmens liegen.

## SQL Azure

Wie der Name bereits erahnen lässt, stellt SQL Azure Datenbankfunktionalität in der Microsoft-Cloud zur Verfügung. Verglichen mit klassischen Betriebssystemen entspricht der Windows Azure Storage einem Dateisystem, SQL Azure einem relationalen Datenbanksystem. Abbildung 2.6 skizziert den Aufbau von SQL Azure.

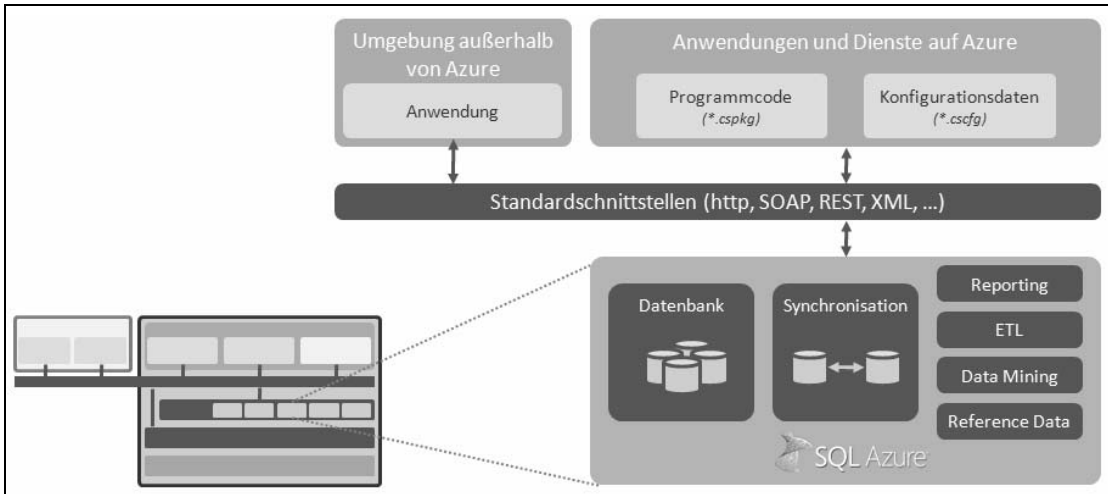


Abbildung 2.6 SQL Azure

Zum Zugriff auf SQL Azure können im Wesentlichen die vom klassischen SQL Server bekannten Aufruftechnologien wie ADO.NET, ADO.NET Data Services (in Kombination mit LINQ) nahezu unverändert verwendet werden.

## Funktionalitäten

SQL Azure stellt eine Teilmenge der Funktionen eines SQL Server, der SQL Azure als Technologiebasis zugrunde liegt, zur Verfügung. Diese ist für die hohen Anforderungen eines Betriebs in der Cloud, wie Skalierbarkeit und Verfügbarkeit, optimiert.

### ■ Datenbank (SQL Azure Database)

SQL Azure ist Microsofts Datenbank für die Cloud, die die Funktionen eines relationalen Datenbanksystems bereitstellt. Im Wesentlichen kommen hier die gleichen Konzepte wie Tabellen, Views, Joins, Stored Procedures etc. zum Einsatz. Tatsächlich liegt SQL Azure ein SQL Server zugrunde, was auch erklärt, dass für die Zugriffe auf die Datenbank die gleichen Mechanismen zum Einsatz kommen können.

### ■ Synchronisation

Über den Synchronisationsdienst können über SQL Azure gespeicherte Daten mit anderen Datenbeständen (z.B. auf mobilen Endgeräten) abgeglichen werden.

### ■ Reporting, ETL, Data Mining, Reference Data

Diese Funktionen sind bereits für spätere Ausbaustufen von SQL Azure vorgesehen und führen den Funktionsumfang von SQL Azure an den eines SQL Server heran.

## Mögliche Einsatzszenarien

Die Alternative, mit einer Cloud-Datenbank kostengünstig und dennoch hoch skalierbar, d.h. in großen Mengen, Daten ablegen zu können, eröffnet eine Reihe interessanter Möglichkeiten.

- **Regelmäßiges Anlegen von Sicherungskopien in der Cloud**  
Entwickler können beispielsweise vorsehen, dass mit jeder lokalen Speicherung von Daten eine Kopie in SQL Azure abgelegt wird. Damit überstehen die Daten auch Beschädigungen lokaler Speichermedien.
- **Speicherung zu archivierender Daten**  
Archivierung ist ein häufig unterschätztes Thema. Aufgrund vermeintlicher Kosten für Systeme, die nur in den seltensten Fällen (eben nur, wenn Zugriff auf »alte« Daten benötigt wird) tatsächlich einen Mehrwert bieten, scheuen viele Anwender vor Archivierungslösungen zurück. Mit SQL Azure steht nun eine Alternative zur Verfügung. Sehr kostengünstig können hier massenhaft Daten zuverlässig und über längere Zeiträume archiviert werden.

### Zukünftige Weiterentwicklungen

Neben den genannten Services sind weitere Anwendungsdienste in Planung, die bereits durch ihre Namensgebung Hinweise auf die bereitgestellten Funktionalitäten geben.

- *SharePoint Services*  
Diese Dienste werden ähnliche Funktionalitäten, wie sie bereits von den in Windows Server integrierten gleichnamigen Services bekannt sind, bereitstellen. Die SharePoint Services können sinnvoll beim Aufbau von Anwendungen, die die Zusammenarbeit zwischen Anwendern unterstützen, eingesetzt werden.
- *Dynamics CRM Services*  
Auch hier ist die Bezeichnung der Dienste nicht zufällig gewählt. Mithilfe der Dynamics CRM Services können Anwendungen, die Funktionen für das Management von Kundenbeziehungen benötigen, entwickelt bzw. ergänzt werden.

Während Windows Azure, die .NET Services sowie die SQL Azure infrastrukturnahe Funktionalitäten bereitstellen und die Live Services auf Funktionen für Endkunden abzielen, bieten die SharePoint bzw. Dynamics CRM Services also Hilfestellungen bei der Entwicklung von Anwendungen im Kontext von Unternehmensanwendungen.

## Bedeutung der Plattform

Das Angebot der Windows Azure Plattform richtet sich an viele verschiedene Zielgruppen, die von den Möglichkeiten einer Cloud-Plattform auf Basis der Microsoft-Technologie profitieren möchten. Das Spektrum reicht hier von Großunternehmen bis hin zu Endanwendern und von Entwicklern über Systemintegratoren bis hin zu Unternehmensgründern.

## Azure für Unternehmen

Unternehmen können von Azure zum einen als weitere Alternative für den Betrieb einzelner IT-Funktionen, d.h. ganzer Anwendungen aber auch Anwendungsteile, profitieren. Im Einzelnen ergibt sich hieraus eine Vielzahl von Vorteilen.

- Vereinfachte Kapazitätsplanung für IT-Funktionen  
Bei einem Softwarebetrieb auf Windows Azure entfallen wesentliche Schritte zur Kapazitätsplanung. Der Aufbau einer detailliert geplanten Infrastruktur entfällt. Diese wird von Microsoft bezogen und skaliert automatisch entsprechend den Anforderungen. Insbesondere kann die Infrastruktur dynamisch skaliert werden, d.h. in der Anfangsphase kann mit einer einzigen bzw. wenigen Instanzen gestartet werden und bei steigenden Zugriffszahlen zusätzliche Instanzen hinzugefügt werden. Dies ist ein entscheidender Vorteil gegenüber Planungen für einen Vor-Ort-Betrieb, bei dem typischerweise für eine zukünftige Maximalauslastung geplant wird. Die immer noch vorhandenen Kosten können besser an der tatsächlichen, aktuellen Nutzung der Plattform ausgerichtet werden und wandeln sich somit von Fixkosten (für eine feste Infrastruktur) zu variablen Kosten.
- Übertragung der Verantwortung für die Einhaltung der Service Level an Microsoft  
Eine Verlagerung des Anwendungsbetriebs zu Azure geht einher mit der Verlagerung von Verantwortung für die Einhaltung der von den Anwendern geforderten Service Level an Microsoft. Microsoft garantiert eine bestimmte Dienstqualität für den Betrieb. Wird diese nicht erreicht, wird das nutzende Unternehmen über zuvor vereinbarte Kompensationsleistungen entschädigt.
- Nutzungsabhängige Abrechnung von IT-Leistungen  
Neben dem Wegfall des Aufbaus einer kostenintensiven Infrastruktur für den Betrieb einer Anwendung können bei einer Verlagerung des Anwendungsbetriebs auf eine Cloud-Plattform die anfallenden Kosten stärker mit der Nutzung der betreffenden Anwendung korreliert werden. In den Gesamtkosten für eine Anwendung steigt der Anteil der nutzungsabhängigen, variablen Kosten gegenüber den Fixkosten (die sonst für die Bereitstellung der Infrastruktur anfallen). Dies ist insbesondere dann von Vorteil, wenn die Nutzungsintensität starken Schwankungen unterliegt. Beim Betrieb in der Cloud passen sich die Kosten an.
- Möglichkeit der Fokussierung auf höherwertige IT-Leistungen  
Die Verlagerung des Betriebs in die Cloud ist insbesondere dann sinnvoll, wenn die Anforderungen an den Betrieb sehr generisch sind. Azure bietet Standarddienste sehr kostengünstig an. Dies erlaubt es dem IT-Betrieb eines Unternehmens, für derartige Anwendungen Azure als Betriebsplattform zu nutzen und die eigenen Ressourcen für Spezialanforderungen, d.h. höherwertige und beratungsintensive Leistungen, zu nutzen.
- Beschleunigte Bereitstellung von IT-Lösungen  
Azure Services stehen im Bedarfsfall sofort zur Verfügung, d.h. nach Anmeldung in der Azure-Managementkonsole können neue Projekte angelegt und darüber neue virtuelle Umgebungen eingerichtet und zum Betrieb von Anwendungen genutzt werden. Die Bereitstellung der hierfür benötigten Infrastruktur wird von Microsoft sichergestellt.
- Vergrößerte Reichweite durch weltweite Verfügbarkeit  
Die Microsoft-Infrastruktur, auf die Azure aufsetzt, ist darauf ausgelegt, weltweite Verfügbarkeit rund um die Uhr sicherzustellen. Speziell für Unternehmen mit (weltweit) verteilten Standorten ist dies von Vorteil, da insbesondere auch kleinere Standorte mit hoher Servicequalität angebunden werden können. Beim Anlegen von Projekten kann die Region gewählt werden, in der Anwendungskomponenten in einem dort betriebenen Rechenzentrum ausgeführt werden. Nutzer haben somit die Möglichkeit, Antwortzeiten für Anwender zu optimieren, indem sie für IT-Funktionen den Standort wählen, der möglichst nah am Anwender liegt.

- Verbesserte Integration (Standardschnittstellen)

Das Angebot und die Nutzung von Services, die auf Azure betrieben werden, und die Möglichkeit des Zugriffs über Standardschnittstellen erleichtern eine Integration in bestehende Systemlandschaften. Wenngleich eine solche Serviceorientierung natürlich auch auf anderem Weg erreichbar ist, erzwingt Azure diesen Architekturansatz und trägt hierdurch zu einer guten Integrierbarkeit der Anwendungen bei.

## Azure für Endanwender

Endanwender werden Azure indirekt über Anwendungen nutzen, die auf Azure aufsetzen bzw. auf Azure Services zugreifen. Mögliche Einsatzszenarien sind:

- Vernetzung aller Endgeräte und Datensynchronisation über die Mesh Services

Mit *Live Mesh* steht bereits eine Anwendung bereit, die auf Basis der Mesh Services eine Vernetzung der verschiedenen Endgeräte eines Anwenders ermöglicht. Die Windows Azure Plattform ist das Rückgrat dieser Anwendung, indem es Speicherdienste zur Ablage von Dokumenten in der Cloud, Synchronisationsdienste zur Weiterleitung von Daten an die einzelnen Endgeräte und eine Weboberfläche zum direkten Zugriff auf die Mesh-Inhalte ermöglicht.

- Durchführung von Backups lokaler Speichermedien

Auch für Endkunden ist eine Speicherung von Sicherungskopien auf einem kostengünstigen Speichersystem, wie es durch SQL Azure bzw. Windows Azure Storage bereitgestellt wird, eine gute Alternative für die Sicherung wichtiger Daten.

- Teilnahme an Communitys

Die Ausbildung so genannter *Social Communitys* lebt von der möglichst weltweiten Verfügbarkeit entsprechender Anwendungen, einfachen Zugangsmöglichkeiten und der personalisierten Darstellung von Inhalten. Genau diese Anforderungen können von den Azure Services, insbesondere den Live Services abgedeckt werden. Die Plattform selbst garantiert weltweite, unterbrechungsfreie Zugriffsmöglichkeiten, Standardschnittstellen ermöglichen den Zugriff aus verschiedenen Geräten und Anwendungen heraus und die Live Services schaffen mit ihren nutzerbezogenen Diensten die Voraussetzung für personalisierte Anwendungsoberflächen.

## Azure für Softwareentwickler

Die Möglichkeiten von Azure für Softwareentwickler sind komplementär zu den Möglichkeiten der Unternehmens- oder Privatanwender. Was Letztere nachfragen, kann von den Entwicklern im Rahmen von Software as a Service (SaaS)-Angeboten bereitgestellt werden.

- Verkleinerung des Hard- und Software-Footprints einer Anwendungslösung

Die Option, eine Anwendung auch als Dienst nutzen zu können, senkt die Hürden für einen Einsatz der Anwendung beträchtlich. Anwender können die Anwendung *auf Anforderung* nutzen. Der Einfluss auf die bestehende Hard- und Softwarelandschaft des Anwenders beschränkt sich auf die Integration von Services. Softwareentwickler können durch das Angebot ihrer Lösung als einen auf Azure gehosteten Dienst den potenziellen Nutzerkreis deutlich vergrößern.



- Vergrößerung des adressierbaren Marktes durch Bereitstellung einer Anwendung *online*  
Die Bereitstellung einer Anwendungslösung bedeutet automatisch, dass diese weltweit verfügbar ist. Allein dadurch ergibt sich eine Ausweitung des Kreises möglicher Kunden.
- Schnelle Bereitstellung von Installationen  
Nachdem beim Kunden der zeit- und kostenintensive Aufbau der erforderlichen Infrastruktur wegfällt, da diese von Microsoft im Bedarfsfall sofort verfügbar ist, können ohne nennenswerte Zeitverzögerung Installationen für Anwender durchgeführt werden. Da diese ohne großen Aufwand wieder deaktiviert werden können, besteht die Möglichkeit, Kunden eine Testinstallation zur Verfügung zu stellen, die bei positiver Testphase in den Produktiveinsatz münden kann.
- Verbesserte Integration von Lösungen durch Unterstützung von Standardschnittstellen  
Die strikte Serviceorientierung von Azure, bei der Funktionalitäten über Standardschnittstellen verfügbar sind, erleichtert es, eigene Lösungen in bestehende Systemlandschaften beim Kunden zu integrieren.
- Nutzung bestehenden Entwickler-Know-Hows  
Microsoft hat bei der Planung von Azure großen Wert darauf gelegt, Softwareentwicklern, die bereits Erfahrung im Umgang mit Microsoft-Entwicklertechnologien haben, den Umstieg auf Azure so einfach wie möglich zu machen. Tatsächlich können .NET-Entwickler in ihrer gewohnten Entwicklungsumgebung und mit bekannten Technologien Software für den Betrieb auf Azure bzw. Software, die Azure Services nutzt, schreiben.

## Azure für Webentwickler und Webagenturen

Webentwicklern bieten sich, insbesondere im Rahmen der Nutzung der Live Services, einige spezielle Möglichkeiten beim Einsatz der Windows Azure Plattform.

- Bereitstellung und Betrieb hoch verfügbarer Websites  
Webentwickler erhalten mit Windows Azure eine Hosting-Plattform, die es erlaubt Websites zuverlässig, ausfallsicher und global zu betreiben.
- Zugang zu einer sehr großen Nutzergemeinde  
Bei Nutzung der Windows Live ID als Authentifizierungsmechanismus steht ein Anwenderkreis von ca. 450 Millionen aktiver Windows Live Nutzer weltweit zur Verfügung. Deren Daten können, sofern freigegeben, zur Personalisierung von Webinhalten verwendet werden.
- Nutzung von Live Services zur Anreicherung von Anwendungen  
Webanwendungen können bei Integration verschiedener Live Services wie Kontakte, Messenger, Search, Maps, Feeds, Alerts etc. ohne großen Aufwand um interessante Funktionen, die auf eine weltweite, vernetzte Plattform aufsetzen, erweitert werden. So kann Anwendern der Präsenzstatus von anderen Nutzern angezeigt werden, Karteninformationen zu bestimmten Inhalten dargestellt werden etc.
- Entwicklung *offline*-fähiger Webanwendungen  
Das Live Operating Environment, welches Bestandteil der Live Services ist, ermöglicht den Betrieb spezieller Webanwendungen auf allen Geräten, auf denen diese Umgebung installiert ist. Dieser Betrieb ist insbesondere auch ohne Netzverbindung möglich (sofern die Anwendung über den Synchronisationsdienst auf das betreffende Gerät geladen wurde).

- Kurzfristige Bereitstellung zeitlich befristeter Websites für Internetkampagnen  
Azure bietet die Möglichkeit, auch kurzfristig große Infrastrukturen bereitzustellen, die nach einer zeitlichen Frist auch wieder deaktiviert werden können. Diese Eigenschaft macht Azure insbesondere für Webagenturen interessant, die zeitlich befristet Web-Kampagnen durchführen, für die für einen kurzen Zeitraum hohe Last möglich ist, für die aber nicht dauerhaft eine entsprechende IT-Infrastruktur vorgehalten werden soll.

## Azure für Systemintegratoren

Bei Systemintegratoren liegt der Schwerpunkt der Nutzung von Azure in der Ergänzung des eigenen Produktportfolios um Betriebsdienstleistungen, die dann tatsächlich von Microsoft erbracht werden. Mit der Erweiterung des Angebots um die Cloud als weitere Betriebsalternative ergeben sich einige neue Möglichkeiten.

- Vereinfachte Kapazitätsplanung  
Analog zu Anwenderunternehmen vereinfachen sich auch für Systemintegratoren Fragestellungen hinsichtlich der Kapazitätsplanung für Anwendungen. Durch die Flexibilität von Azure kann zunächst eine kleine Infrastruktur eingesetzt werden, die mit dem Erfolg der Anwendung dynamisch skaliert werden kann. Lastspitzen stellen ebenfalls kein Problem dar. Wo im klassischen Softwarebetrieb vor Ort die Infrastruktur theoretisch für eben diese Spitzenzeiten ausgelegt werden muss (und zu anderen Zeiten ggf. brachliegt), kann dies durch Azure flexibel aufgefangen werden, indem zu Zeiten hoher Last dynamisch weitere Instanzen zugeschaltet werden.
- Fokussierung auf die Erstellung von Geschäftslogik  
Nachdem sich die Verantwortung für die Infrastruktur und den Anwendungsbetrieb zu Microsoft verlagern, können sich Systemintegratoren verstärkt auf die Erstellung der Geschäftslogik, die für den Anwender den eigentlichen Mehrwert darstellt, fokussieren.
- Zugang zu neuen Kundengruppen durch vereinfachte Bereitstellung von IT-Ressourcen  
Für Kunden von Systemintegratoren gestaltet sich der Aufbau von Anwendungen deutlich einfacher dadurch, dass die Infrastruktur durch die Azure Cloud bereitgestellt wird. Durch diese Vereinfachung sinken Hürden für den Einsatz von Anwendungen, was Systemintegratoren wiederum neue Kundengruppen, d.h. kleinere Unternehmen aus dem Mittelstand bzw. Fachabteilungen erschließt.
- Beratungsleistungen im Bezug auf Cloud Computing  
Systemintegratoren sind für Anwender wichtige Ansprechpartner bei der Planung des IT-Portfolios. Mit der Verfügbarkeit von Azure wächst hier der Beratungsbedarf hinsichtlich des Cloud Computing. Anwender benötigen Antworten auf die Fragen, wie Anwendungen Cloud-kompatibel gemacht oder ganz migriert werden können; wie bestehende, vor Ort betriebene Anwendungen mit Azure kombiniert werden können usw.
- Vereinfachte Integration über Standardschnittstellen  
Die Unterstützung von Standards bei den Zugriffsschnittstellen erleichtert die Implementierung von Hybridlösungen, bestehend aus vor Ort betriebenen Anwendungen und Azure Services. Für den Zugriff auf Azure können beliebige Technologien (auch außerhalb des Microsoft Produktspektrums) zum Einsatz kommen, solange sie die Standards unterstützen. Auch diese Interoperabilität vereinfacht die Integration von Azure in bestehende Systemlandschaften.

## Azure für Unternehmensgründer

Für Unternehmensgründer ist die Flexibilität der Azure Cloud ein entscheidender Vorteil bei der Wahl eines Betriebsmodells für die Anwendungen. In einer Unternehmensphase, in der die Kapazitätsplanung durch die unsichere Erfolgsprognose erschwert wird, bringt eine Infrastruktur, die nach dem Motto »klein anfangen und im Erfolgsfall schnell wachsen« flexibel skalierbar und im Bedarfsfall weltweit verfügbar ist, deutliche Vorteile gegenüber anderen Betriebsmodellen. Nachdem der Betrieb nutzungsabhängig abgerechnet wird, entlastet Cloud Computing mit Azure die Bilanz des Gründers insofern, als dass dieser damit keine eigene Hardware-Infrastruktur aufbauen muss.

## Schrittweise Adaption von Azure

Die Windows Azure Platform ist als Ergänzung der klassischen Microsoft-Plattform zu verstehen. Azure ist also weder als Nachfolger noch als Ersatz für bestehende Microsoft-Technologien zu sehen. Analog bedeutet die Verfügbarkeit von Azure eine Erweiterung von Möglichkeiten für die Anwendungsentwicklung, ohne dass bestehende Anwendungen notwendigerweise komplett migriert werden müssten. In vielen Fällen ist dies nicht gewünscht oder gar nicht möglich. Dennoch ist es sinnvoll, sich der neuen Möglichkeiten und der Herausforderungen bewusst zu werden, um sich schrittweise auf Azure vorzubereiten und letztlich Azure als einen Bestandteil zukünftiger Entscheidungen zur Architektur von Softwaresystemen zu etablieren. Das im Folgenden beschriebene schrittweise Vorgehen macht die Komplexität der Änderung in der Anwendungsentwicklung beherrschbar.

### 1. Erweiterung eigener Software um Cloud Services

In einem ersten Schritt kann untersucht werden, ob möglicherweise ein punktueller Einsatz von Azure Services aus bestehenden Anwendungen heraus sinnvoll ist. Beispielsweise können Backup-Mechanismen über SQL Azure eingeführt werden, bei denen Sicherungskopien in Azure gespeichert werden.

### 2. Vorbereitung der Software zum Betrieb auf Azure

Azure fordert von Anwendungen, die in der Cloud betrieben werden sollen, eine strikte Serviceorientierung. Komponenten dieser Anwendungen müssen über klar definierte Schnittstellen miteinander kommunizieren. Grundsätzlich kommen beim Entwurf derartiger Systeme die gleichen Überlegungen zum Tragen, wie sie bereits im Rahmen serviceorientierter Architekturen (SOA) angestellt werden. Azure erweitert die Möglichkeiten einer SOA in die Cloud. Umgekehrt erleichtert eine SOA die Integration von Azure Services.

### 3. Migration von Anwendungen auf Azure

In der nächsten Phase kann eine bestehende Anwendung so umgeschrieben werden, dass sie für einen einzelnen Mandanten auf einer Instanz in Azure betrieben werden kann (*single tenant*). In diesem Schritt müssen unter anderem folgende Dinge beachtet werden:

- Die Architekturvorgaben von Azure (Serviceorientierung, lose Koppelung der Services etc.) müssen eingehalten werden.
- Logging-Ausgaben sollten ausführlich genug sein, damit Fehlerfälle im Nachhinein auf der lokalen Entwicklungsumgebung nachgestellt werden können.

- Da kein direkter Zugriff auf die Infrastruktur der Cloud möglich ist, sollte die Anwendung robust genug sein, Hardware-Ausfälle zu tolerieren, d.h. bei Hardware-Ersatz ohne Probleme neu gestartet werden können.

Für weitere Mandanten würden jeweils eigene Umgebungen auf Azure eingerichtet.

#### 4. Optimierung von Anwendungen für den Betrieb auf Azure

Um die bei Azure vorgehaltene Infrastruktur und die darauf betriebenen virtuellen Umgebungen optimal auszulasten, kann im nächsten Schritt eine Weiterentwicklung der Anwendung stattfinden, die den Multimandantenbetrieb (*multi-tenant*) auf einer Instanz erlaubt. Dies ist der wahrscheinlich aufwändigste Schritt, da mit diesem einige nur schwer zu beantwortende Fragestellungen einhergehen (z.B. Werden für einzelne Mandanten unterschiedliche Versionen der Anwendung unterstützt? Wie kann ein Daten-Rollback für einzelne Mandanten erfolgen? Wie werden Mandanten voneinander separiert?).

#### 5. Software plus Services als Strategie für die Entwicklung

Im letzten Schritt wird die Möglichkeit des Betriebs einzelner Anwendungskomponenten auf Azure integraler Bestandteil der Architekturplanung zukünftiger Softwaresysteme. Mit Azure eröffnet sich neben dem Betrieb bei einem Host und dem klassischen Betrieb vor Ort beim Anwender eine dritte Möglichkeit. Jede dieser Alternativen hat spezifische Vor- und Nachteile, und es liegt in der Verantwortung des jeweiligen Softwarearchitekten, die für eine Anwendungskomponente jeweils bestgeeignete auszuwählen, um – entsprechend Microsofts Software plus Services-Strategie – die Vorteile vor Ort betriebener Software mit Cloud Services zu kombinieren.

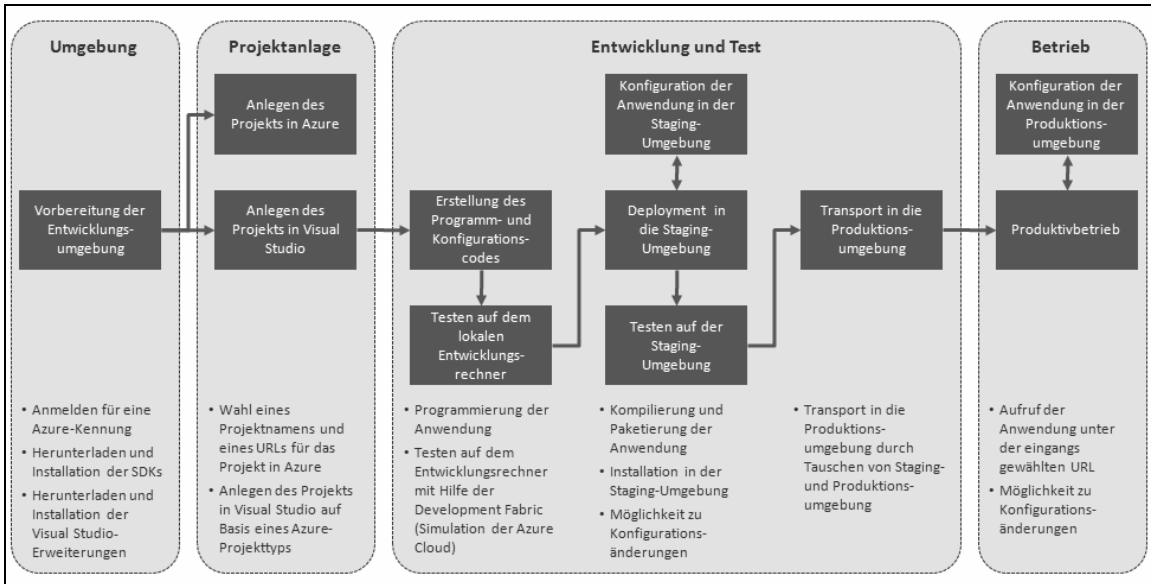
## Softwareentwicklung mit Azure

Beim Entwurf von Azure legte Microsoft sehr großen Wert darauf, den Umstellungsaufwand für Anwendungen, die zum Betrieb auf Azure migriert werden sollen, möglichst gering zu halten. Gleiches sollte auch für die Entwickler selbst gelten. Die Programmierung von Azure-Anwendungen selbst ist weitestgehend identisch zur Entwicklung klassischer Windows-Anwendungen auf Basis des .NET Framework. Tatsächlich enthalten Azure-basierte Webanwendungen im Kern klassischen ASP.NET-Programmcode, wie später noch gezeigt wird.

Grundsätzlich erfolgt die Entwicklung von Anwendungen, die auf Azure betrieben werden sollen, in vier Schritten:

1. Vorbereitung der Entwicklungsumgebung
2. Anlegen der benötigten Anwendungsprojekte in Azure und in Visual Studio
3. Entwicklung und Test des Programmcodes und der Konfigurationsdaten
4. Inbetriebnahme auf Azure

Diese Schritte sind in Abbildung 2.7 skizziert und werden in den folgenden Abschnitten – zum Teil anhand einer kleinen Beispielanwendung – genauer erläutert.



**Abbildung 2.7** Schritte zur Entwicklung eines Azure Web Cloud Services

## Vorbereitung der Entwicklungsumgebung

Zunächst müssen auf dem lokalen Entwicklungsrechner einmalig die Voraussetzungen zur Entwicklung von Azure Services geschaffen werden. Die zur Entwicklung von Azure-Anwendungen erforderlichen Programmierwerkzeuge können kostenfrei von der Microsoft-Website heruntergeladen werden. Für die Entwicklung der Beispiele in diesem Buch werden ausschließlich die folgenden, frei verfügbaren Werkzeuge eingesetzt (Ausnahmen werden an entsprechender Stelle explizit gekennzeichnet):

- Visual Web Developer 2008 Express Edition SP1
- SQL Server Express Edition 2005 oder 2008 (sofern eine lokale Testumgebung benötigt wird)
- Windows Azure SDK
- Azure Tools for Visual Studio
- .NET Services SDK
- SQL Azure SDK
- Live Framework SDK

Als Betriebssystem können Windows Vista SP1, Windows 7 oder Windows Server 2008 eingesetzt werden. Diese sind selbstverständlich lizenzpflichtig.

**HINWEIS** Im Rahmen der Community Technology Preview (CTP) wird zum Anlegen eines Projekts in Azure und dortigem Deployment ein Token zur Freischaltung eines Testaccounts benötigt. Dieser kann unter der Microsoft-Connect-Website (<http://connect.microsoft.com/>) beantragt werden und wird dem Antragsteller innerhalb weniger Tage per E-Mail zugestellt.

Nach der Installation der Software Development Kits (SDKs) steht in Visual Studio eine Reihe neuer Projekttypen zur Verfügung, die Programmierung, Test, Deployment und Verwendung von Azure Services deutlich vereinfachen. So legt der Projekttyp für Web Cloud Services bereits Projekte und Vorlagen für Programmcode- und Konfigurationsdateien an, die vom Entwickler an den entsprechenden Stellen ausgefüllt werden müssen.

## Azure Development Fabric

Bei der Installation des Azure SDK wird eine Testumgebung auf dem lokalen Entwicklerrechner installiert, die es in sich hat, und die einzigartig für die verschiedenen im Markt befindlichen Cloud Computing-Angebote ist: die Azure Development Fabric. Diese Umgebung bildet die Infrastruktur der Azure-Plattform auf dem Entwicklungsrechner in Form einer *simulierten Cloud* nach. Entwickler haben hierüber die Möglichkeit, vor der Installation einer Anwendung auf Azure diese lokal zu testen. Wenngleich die Simulation durch einige Restriktionen des Entwicklerrechners (z.B. beschränkte, nicht veränderbare Anzahl an Rechenkernen, begrenzter Speicher usw.) an Grenzen stößt, können damit das Laufzeitverhalten, mehrere parallele Instanzen etc. der *echten Cloud* nachgestellt werden. In der Azure Development Fabric stehen, wie auf Azure selbst, eine Ablaufumgebung, Azure Storage, Azure Queues etc. zur Verfügung.

## »Hello World« mit Azure

Es ist in der IT-Welt bereits Tradition, eine neue Entwicklungstechnologie anhand eines kleinen »Hello World«-Beispiels vorzustellen und darüber einen ersten Eindruck über die Funktionsweise und die Möglichkeiten der Technologie zu vermitteln. Die folgenden Abschnitte führen Schritt für Schritt durch den Entwicklungsprozess, der zum Anlegen eines kleinen Azure Services durchlaufen werden muss. Es handelt sich dabei um eine kleine Webanwendung, die ein Eingabefeld, eine Schaltfläche und ein Textlabel enthält. In das Eingabefeld kann der Anwender einen Namen eintragen. Nach Klicken auf die Schaltfläche wird die Person dieses Namens mit einem kleinen Willkommensgruß beglückt. Um das Beispiel einfach, nachvollziehbar und den Programmcode verständlich zu halten, wird auf eine ausgefeilte Fehlerbehandlung verzichtet.

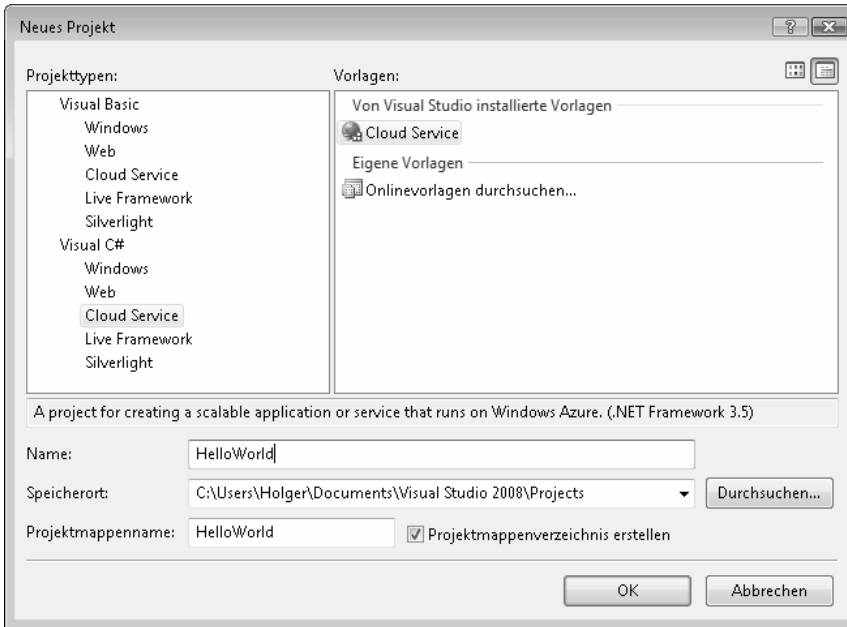
### Anlegen einer Web Role

Nach der Installation der Azure Tools for Visual Studio stehen in Visual Web Developer eine Reihe neuer Projekttypen zur Verfügung. Zum Anlegen einer Projektmappe (engl. Solution) für das Beispielprojekt wählen Sie im Menü *Datei* den Befehl *Neues Projekt*. Dieser öffnet das in Abbildung 2.8 abgebildete Dialogfeld.

---

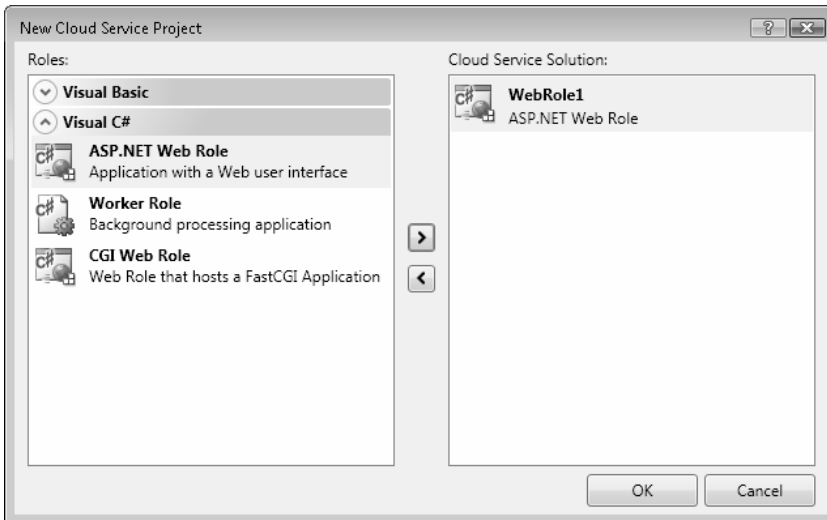
**HINWEIS** Um in einem späteren Schritt in der Azure Development Fabric auf dem lokalen Entwicklerrechner testen zu können, muss Visual Studio bzw. Visual Web Developer mit Administratorrechten gestartet werden.

---



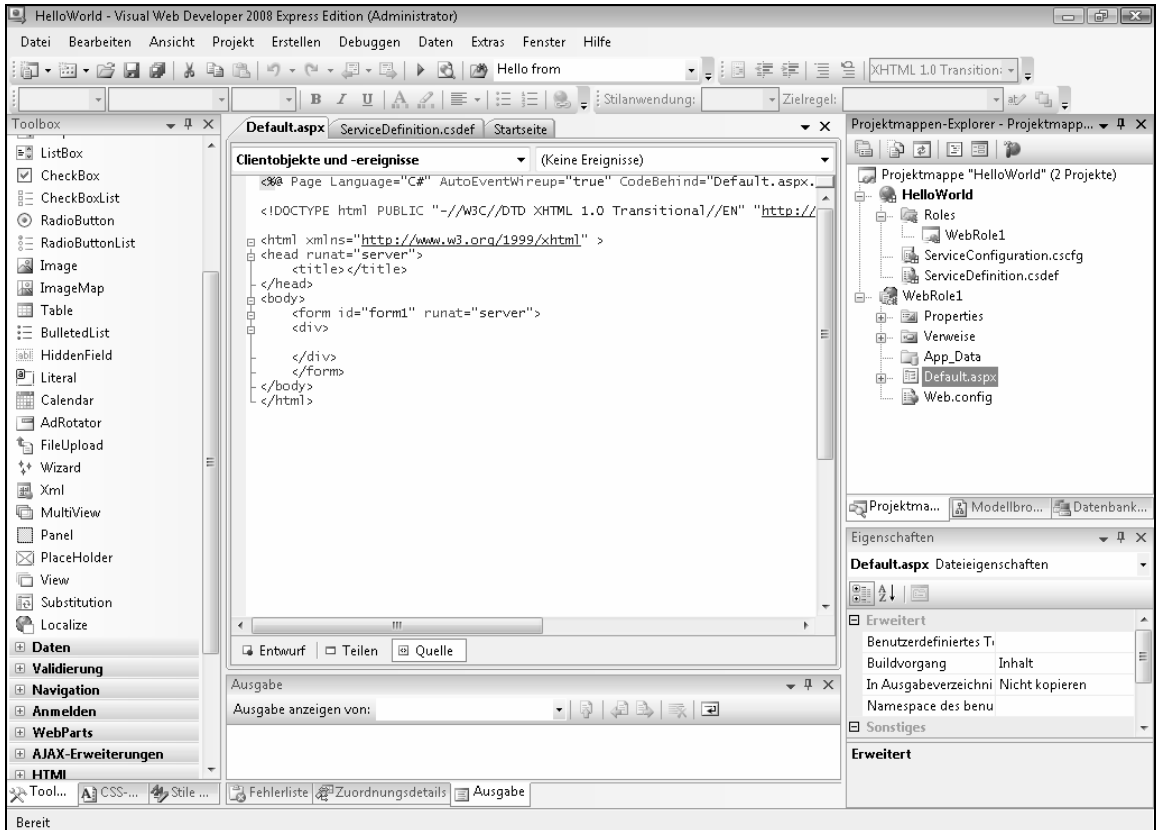
**Abbildung 2.8** Neue Projekttypen im Azure SDK

In dem Dialogfeld ist die neue Projektvorlage für Azure unter dem Projekttyp *Cloud Service* zu sehen. Für das Beispielprojekt markieren Sie die Projektvorlage *Cloud Service* und wählen als Projektnamen HelloWorld. Bestätigen Sie Ihre Auswahl mit *OK*. Eine Eingabemaske, die in Abbildung 2.9 zu sehen ist, öffnet sich.



**Abbildung 2.9** Einfügen von Web und Worker Roles in das Projekt

Zur Auswahl stehen dort Vorlagen für eine ASP.NET Web Role, für eine CGI Web Role und für eine Worker Role. Web Roles sind Services, die die Interaktion mit Anwendern bzw. Anwendungen von außen regeln. Sie sind von außen über Standardschnittstellen zugreifbar und für die Darstellung der Oberfläche einer Webanwendung verantwortlich. Worker Roles implementieren Hintergrundprozesse, d.h. länger laufende Prozesse, die von Web Roles entkoppelt werden sollen. Worker Roles sind nur indirekt von außen – in der Regel über Web Roles, die die Worker Roles über Queue-Einträge benachrichtigen – aufrufbar. Visual Studio erzeugt daraufhin automatisch eine Projektmappe, die zwei Projekte enthält. Abbildung 2.10 zeigt das Fenster von Visual Web Developer, wie es sich dem Entwickler präsentiert.



**Abbildung 2.10** Die über den Projekttyp *Cloud Service* angelegte Projektmappe mit zwei Projekten

Auf der linken Seite sind im Projektmappen-Explorer zwei Projekte zu sehen:

- *HelloWorld*

Dieses Projekt enthält die Konfigurationsinformationen, die später für die Installation auf Azure benötigt werden.

- *WebRole1*

In diesem Projekt befindet sich das Gerüst für das eigentliche Programm, welches später auf Azure installiert wird. Im Aufbau entspricht es einer klassischen ASP.NET-Anwendung.



Im Programmcode-Fenster in der Mitte ist bereits die Datei *Default.aspx*, die beim späteren Start der Anwendung als erste aufgerufen wird, geöffnet. Auch sie entspricht einer klassischen ASP.NET-Seite. Die Programmierung kann, wie bei ASP.NET gewohnt, über eine Quellcode-, eine Entwurfs- oder eine kombinierte Ansicht erfolgen. Die Toolbox auf der linken Seite enthält die verschiedenen zur Entwicklung von ASP.NET-Seiten benötigten Elemente.

## Programm- und Konfigurationscode

Nachdem in den beiden Projekten Gerüste für den Programm- und Konfigurationscode erstellt wurden, kann die eigentliche Entwicklung der Beispielanwendung beginnen. Rufen Sie durch Doppelklick die Datei *ServiceDefinition.csdef* auf. Diese Datei legt fest, welche Konfigurationseinstellungen in unserem Beispielprojekt grundsätzlich möglich sind. Nachdem die Begrüßungsformel zur Laufzeit konfigurierbar sein soll, erweitern Sie diese Datei um die in Listing 2.1 enthaltenen Einträge für den Konfigurationsparameter Begrueßung.

```
<?xml version="1.0" encoding="utf-8"?>
<ServiceDefinition name="HelloWorld"
  xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceDefinition">
  <WebRole name="WebRole">
    <InputEndpoints>
      <!-- Must use port 80 for http and port 443 for https when running in the cloud -->
      <InputEndpoint name="HttpIn" protocol="http" port="80" />
    </InputEndpoints>
    <ConfigurationSettings>
      <Setting name="Begrueßung" />
    </ConfigurationSettings>
  </WebRole>
</ServiceDefinition>
```

**Listing 2.1** *ServiceDefinition.csdef*-Datei mit den Erweiterungen für das »Hello World«-Beispiel

Der Wert des hierüber definierten Konfigurationsparameters Begrueßung kann nun in der Datei *ServiceDefinition.cscfg* festgelegt werden. Rufen Sie hierzu die Konfigurationsdatei auf und ergänzen Sie den Code um den in Listing 2.2 gezeigten Eintrag.

```
<?xml version="1.0"?>
<ServiceConfiguration serviceName="HelloWorld"
  xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration">
  <Role name="WebRole">
    <Instances count="1"/>
    <ConfigurationSettings>
      <Setting name="Begrueßung" value="Hello, {0}! How are you?" />
    </ConfigurationSettings>
  </Role>
</ServiceConfiguration>
```

**Listing 2.2** Die Datei *ServiceDefinition.cscfg* mit den Erweiterungen für das »Hello World«-Beispiel

Diese Begrüßungsformel wird später vom Programm aus der Konfiguration gelesen und zur Begrüßung des Anwenders verwendet. Nun kommen wir zum eigentlichen Programmcode. Ergänzen Sie das Formular in der Datei *Default.aspx.cs* um ein Textfeld, eine Schaltfläche und ein Beschriftungsfeld (engl. label). Wie von ASP.NET gewohnt, können Sie diese Erweiterung entweder über das Eintippen des entsprechenden Codes in der Datei direkt vornehmen oder über die Toolbox die entsprechenden Elemente per Drag-and-Drop in den Code einfügen. Wählen Sie für die Benennung der Elemente bitte die Namen, die in Listing 2.3 zu sehen sind (z.B. txtName als ID für das Textfeld).

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="HelloWorld.WebRole.Default" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Untitled Page</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:TextBox ID="txtName" runat="server" />&nbsp;
      <asp:Button ID="btnSubmit" runat="server" Text="Submit" />
      <br />
      <br />
      <asp:Label ID="lblBegrueussung" runat="server" Text="" />
    </div>
  </form>
</body>
</html>
```

**Listing 2.3** Die Datei *Default.aspx* mit den Erweiterungen für das »Hello World«-Beispiel

Nun fehlt als letzter Schritt die Programmierung der eigentlichen Anwendungslogik in der zugehörigen Code-behind-Datei *Default.aspx.cs*. Nach Klick auf die Schaltfläche soll das Programm den in das Textfeld eingegebenen Namen auslesen und integriert in die Grußformel über das Textlabel als Gruß an die betreffende Person ausgeben. Hierzu ist es zunächst notwendig, die Grußformel aus der Konfigurationsdatei auszulesen. Hierbei kann die Klasse *RoleManager* verwendet werden. Diese enthält die Methode *GetConfigurationSetting*, die unter Angabe des betreffenden Konfigurationseintrags den jeweils eingestellten Wert zurückliefert. Anschließend wird der Wert des Textfelds ausgelesen und mit der Grußformel verknüpft und das Ergebnis dem Text-Attribut des Beschriftungsfelds übergeben. Zur Implementierung ergänzen Sie in der Datei *Default.aspx.cs* die Ereignisbehandlung der Schaltfläche um die in Listing 2.4 markierten Zeilen.

```
using System;
using Microsoft.ServiceHosting.ServiceRuntime;

namespace WebRole1
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            string strFormel =
                RoleManager.GetConfigurationSetting("Begrueussung");

            string strBegrueussung = String.Format(strFormel, txtName.Text);

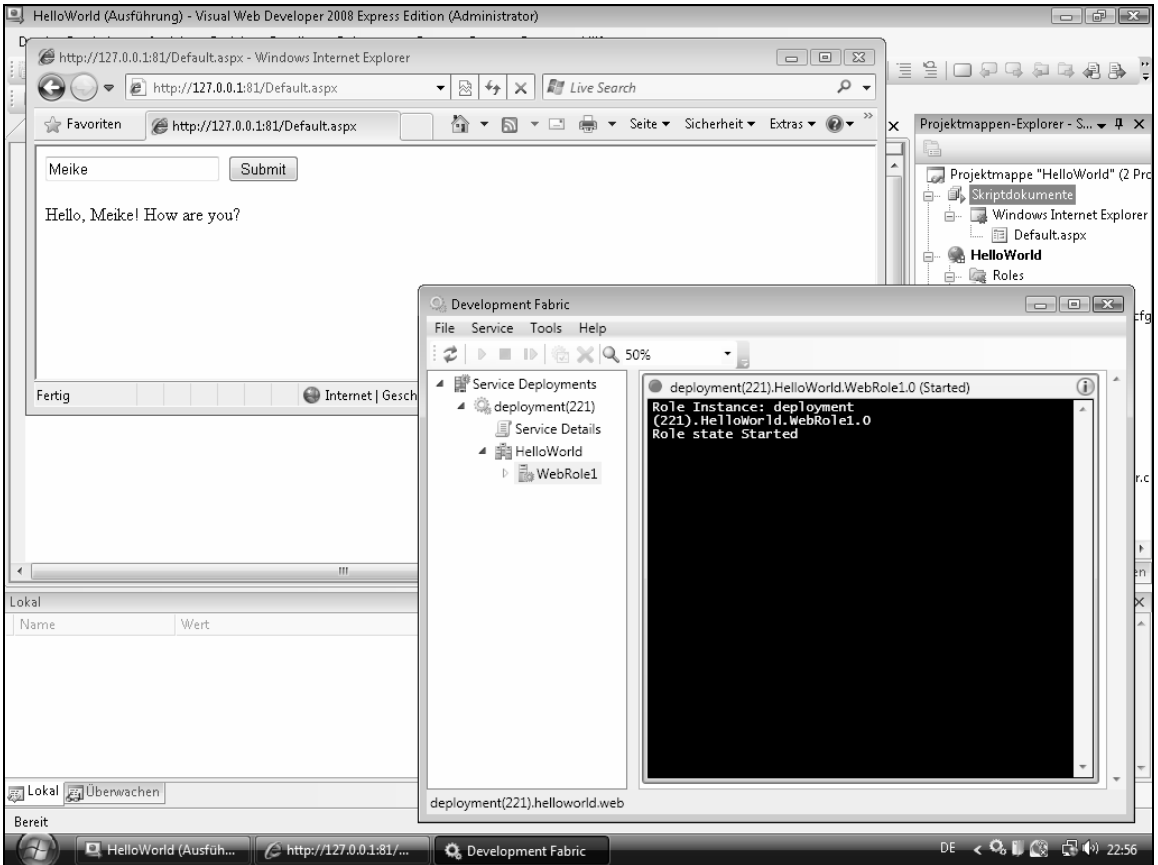
            lblBegrueussung.Text = strBegrueussung;
        }
    }
}
```

**Listing 2.4** Die Datei *Default.aspx.cs* mit den Erweiterungen für das »Hello World«-Beispiel

Damit ist die Programmierung des Beispiels abgeschlossen. Als nächster Schritt folgt das Testen der Anwendung in der lokalen Testumgebung.

## Lokales Testen

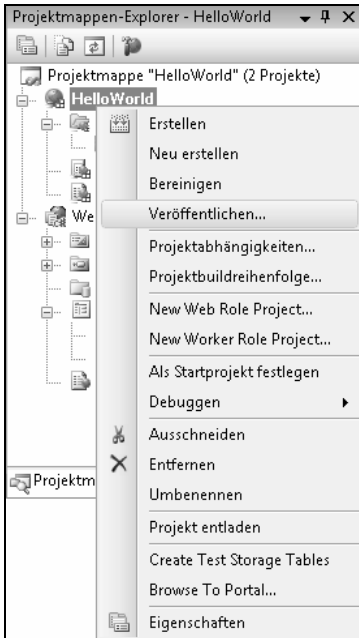
Die Möglichkeit, einen Cloud Service noch vor einem Deployment auf Azure auf dem lokalen Entwicklerrechner in einer simulierten Cloud-Umgebung testen zu können, gehört zu den bemerkenswertesten Eigenschaften von Microsofts Cloud-Technologie. Kompilierung und Test können durch Betätigen der Taste **F5** oder über den Menüpunkt *Debuggen/Debugging starten* ausgelöst werden. Visual Web Developer übersetzt und paketierte daraufhin die Anwendung, startet die Azure Development Fabric, installiert die Anwendung in der Testumgebung, öffnet ein Browserfenster und ruft dort die Datei *Default.aspx* auf. Das Ergebnis ist in Abbildung 2.11 zu sehen.



**Abbildung 2.11** Lokaler Test der »Hello World«-Anwendung

Im Browserfenster wird die aus der lokalen Umgebung heraus aufgerufene Anwendungsseite *Default.aspx* angezeigt. Durch Eingeben eines Namens und Klicken auf die Schaltfläche kann der Begrüßungstext aufgerufen werden. In der Windows-Taskbar sind unten rechts zwei kleine Icons zu sehen, die den Start der Development Fabric und des Development Storage anzeigen. In Abbildung 2.11 ist im Vordergrund die Oberfläche der Development Fabric zu sehen. Links im Fenster ist die simulierte Infrastruktur mit dem Deployment einer Web Role – unserer Beispielanwendung – zu sehen. Rechts die Konsolenausgabe dieser Anwendungsinstanz, die über den Status der Web Role informiert.

Nach erfolgreichem Test kann die Anwendung paketiert und auf Windows Azure veröffentlicht werden. Klicken Sie hierzu mit der rechten Maustaste auf eines der beiden Projekte im Projektmappen-Explorer und wählen Sie im sich öffnenden Kontextmenü den Punkt *Veröffentlichen*. Abbildung 2.12 zeigt das Kontextmenü.



**Abbildung 2.12** Veröffentlichen des Projekts

Hierdurch werden in Visual Web Developer zwei Aktionen ausgelöst:

- **Paketierung der Anwendung**

Die Anwendung wird kompiliert und in ein Paket namens *HelloWorld.cspkg* verpackt. Visual Web Developer öffnet das Verzeichnis, in dem sich das Anwendungspaket befindet, in einem Explorer-Fenster. Dort befindet sich, außerhalb des Anwendungspakets, auch die Konfigurationsdatei mit der Begrüßungsformel.

- **Öffnen eines Browserfensters mit dem Azure Services Developer Portal**

Neben dem Explorer-Fenster wird auch ein Browserfenster geöffnet, in dem die Management-Konsole von Azure unter der URL

*http://lx.azure.microsoft.com/*

aufgerufen wird. Die Konsole erfordert zunächst die Eingabe des zur Windows Live ID des Entwicklers gehörenden Passworts.

Abbildung 2.13 zeigt das Explorer-Fenster mit dem kompilierten Anwendungsprojekt sowie das Anmeldefenster der Azure-Management-Konsole.

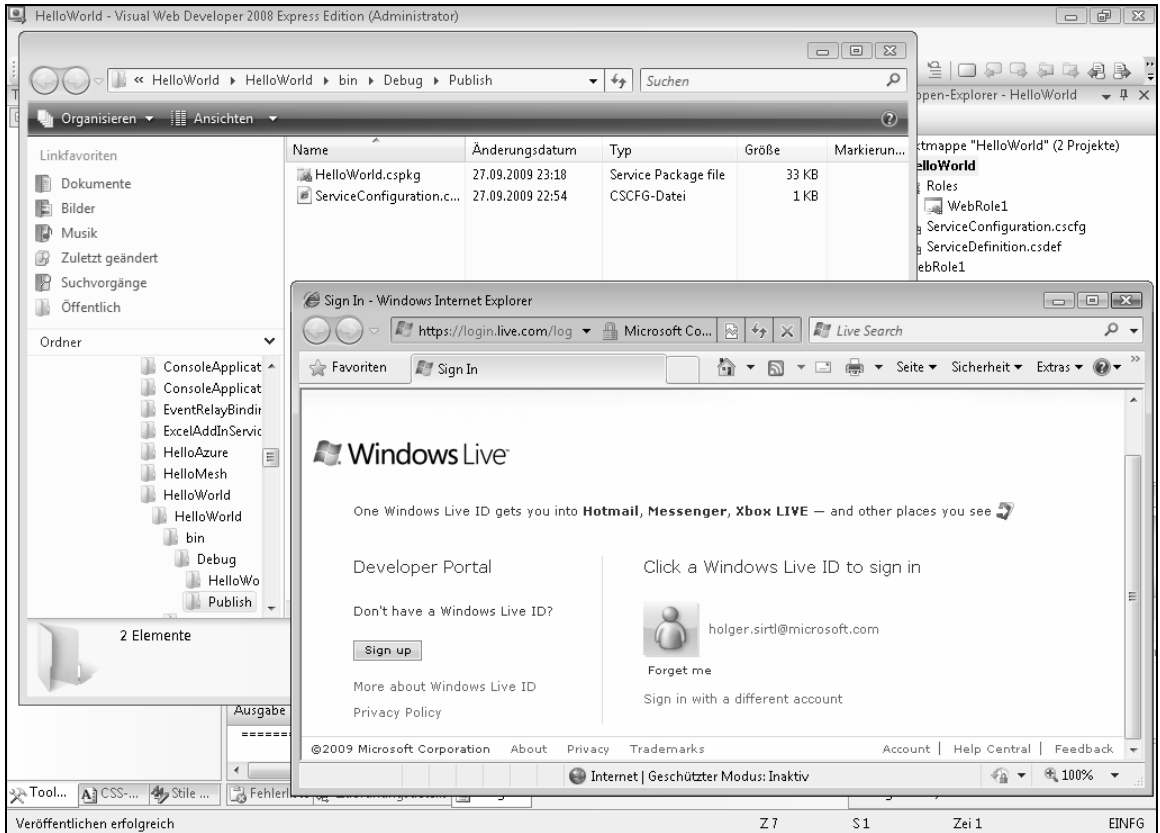


Abbildung 2.13 Paketierung des Projekts und Anmeldung an der Azure-Management-Konsole

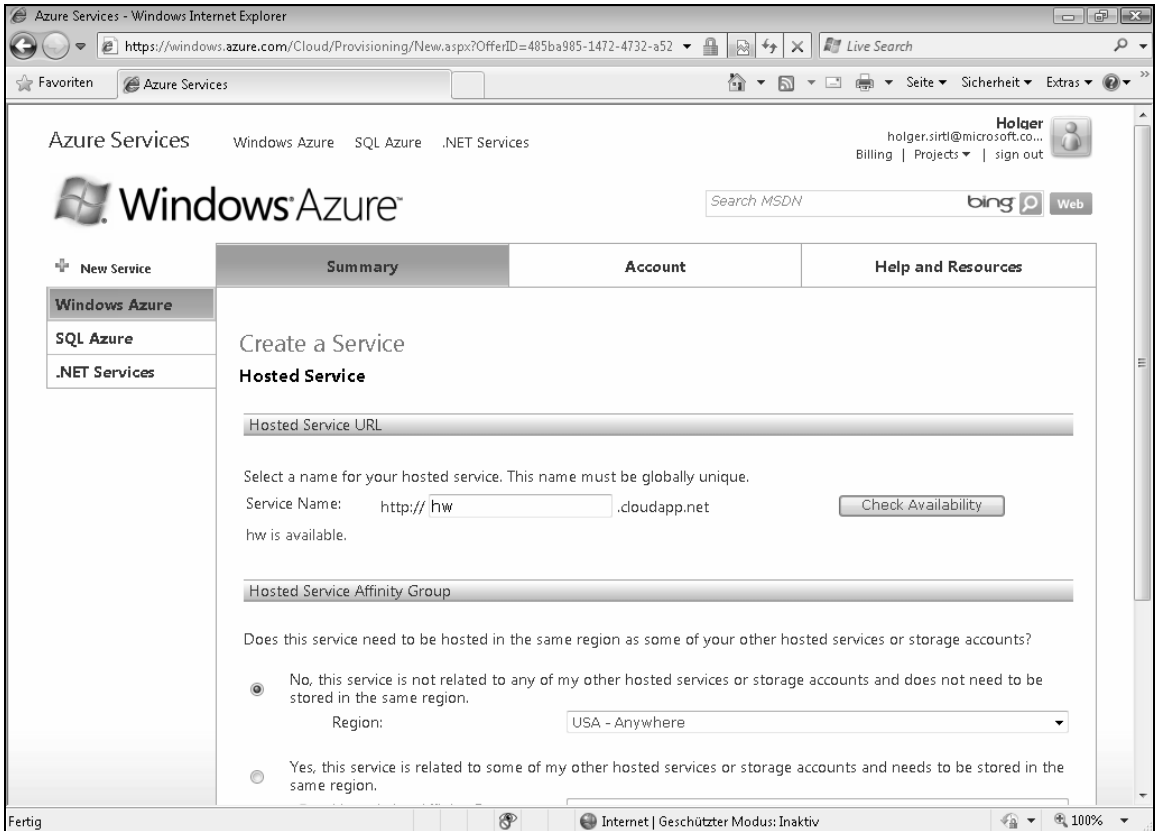
Nach erfolgter Anmeldung an Azure öffnet sich die Startseite der Management-Konsole. Hier können über den Menüpunkt *New Project* verschiedene Azure-Projekte angelegt werden:

- *Storage Account* – zur Nutzung des Windows Azure Storage
- *Hosted Services* – zum Betrieb eines eigenen Service auf Windows Azure
- *Live Services* – zur Nutzung bzw. zum Betrieb eigener Anwendungen auf Basis der Live Services

Für unsere Beispielanwendung wird ein *Hosted Services*-Projekt benötigt.

## Anlegen eines Projekts in Azure

Bei erstmaligem Deployment der Beispielanwendung muss in Azure ein Projekt angelegt werden. Wählen Sie hierzu unter dem Menüpunkt *New Project* den Eintrag *Hosted Services*. Sie können nun einen beliebigen Projektnamen (z.B. *Meine erste Cloudapp*) vergeben. Durch Anklicken der Schaltfläche *Next* kommen Sie auf die nächste Seite, in der Sie einen eindeutigen URL für das Projekt wählen können. Dieser URL darf noch nicht vergeben sein. Um dies zu prüfen, können Sie auf die Schaltfläche *Check Availability* klicken, um Auskunft darüber zu erhalten, ob der Wunsch-URL noch verfügbar ist. Abbildung 2.14 zeigt die zweite Seite zur Informationserhebung für die Projektanlage, die über die Schaltfläche *Create* angestoßen werden kann.



**Abbildung 2.14** Anlegen eines neuen Azure Projekts und Auswahl des URLs für das »Hello World«-Beispiel

Auf Azure werden daraufhin zwei identische, virtuelle Umgebungen angelegt: eine so genannte *Staging*-Umgebung und eine *Production*-Umgebung. Die Staging-Umgebung ist die Umgebung, in der Installationen eines Service erfolgen. Dort kann der betreffende Service erneut unter realen Bedingungen getestet werden. Nach erfolgreichem Test kann der Service dann in die Production-Umgebung transportiert werden. Abbildung 2.15 zeigt die Managementseite des Projekts. Dort sind diverse Informationen über das Projekt einsehbar:

- *Application ID*  
Diese ID kann in Visual Web Developer in die Projekteigenschaften kopiert werden. Dies hat zur Folge, dass bei späteren Deployments automatisch die korrekte Projektseite in Azure geöffnet wird.
- *Domain(s)*  
Dies ist der bei der Anlage des Azure-Projekts gewählte URL.
- *Return URL*  
Formulare der installierten Webanwendung werden an diesen URL gesendet.
- *Secret Key*  
Ein 256-Bit-Schlüssel für spätere Zugriffe auf das Projekt bzw. den Service.



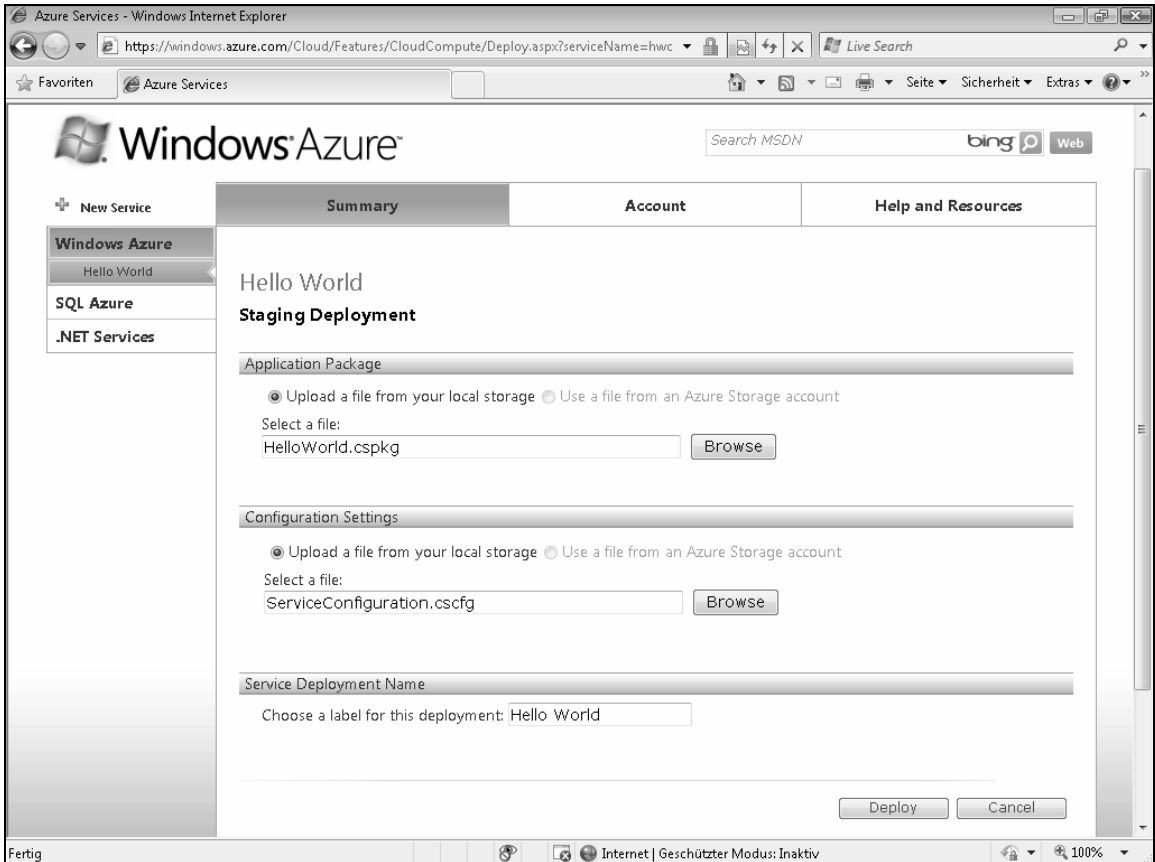
Abbildung 2.15 Zwei virtuelle Umgebungen für das »Hello World«-Beispiel

Noch sind beide Umgebungen leer (und deshalb grau gekennzeichnet). Im nächsten Schritt werden wir die Anwendung in der Staging-Umgebung installieren.

### Installation und Test auf Azure

Zur Installation der »Hello World«-Anwendung klicken Sie auf die Schaltfläche *Deploy*. Diese öffnet das in Abbildung 2.16 gezeigte Eingabeformular, über welches das zuvor von der Entwicklungsumgebung generierte Anwendungspaket und die Konfigurationsdatei auf Azure geladen werden kann. Für das Deployment kann ein sprechender Name vergeben werden. Dieser Name kann beliebig gewählt werden, sollte aber Rückschlüsse auf das Deployment (z.B. Version des Deployments oder neue Funktionalitäten der Anwendung) enthalten. Zur Installation klicken Sie auf die Schaltfläche *Deploy*.





**Abbildung 2.16** Hochladen des Projektpakets und der Konfigurationsdatei

Im Hintergrund wird die Anwendung jetzt auf der Staging-Umgebung installiert. Der Zeitbedarf hierfür kann im Minutenbereich liegen. Die Management-Konsole meldet sich danach zurück. Die Staging-Umgebung ist mit der Anwendung bestückt (erkennbar an der blauen Einfärbung des Würfels und der Statusmeldung *Allocated*). Klicken Sie nun auf die Schaltfläche *Run*, um die Anwendung zu starten.

Nach einer Weile wechselt die Statusmeldung auf *Started* (siehe Abbildung 2.17). Die Anwendung ist jetzt ausführbar. Beachten Sie, dass unterhalb der Statusmeldung ein zukünftiger URL ausgegeben wird. Über diesen URL ist die Anwendung jetzt ausführbar. Klicken Sie auf den URL, woraufhin sich ein weiteres Browserfenster öffnet, in dem die Anwendung mit dem Textfeld und der Schaltfläche angezeigt wird. Auch hier können Sie durch Eingabe eines Namens und Anklicken der Schaltfläche einen Gruß an die betreffende Person ausgeben lassen.

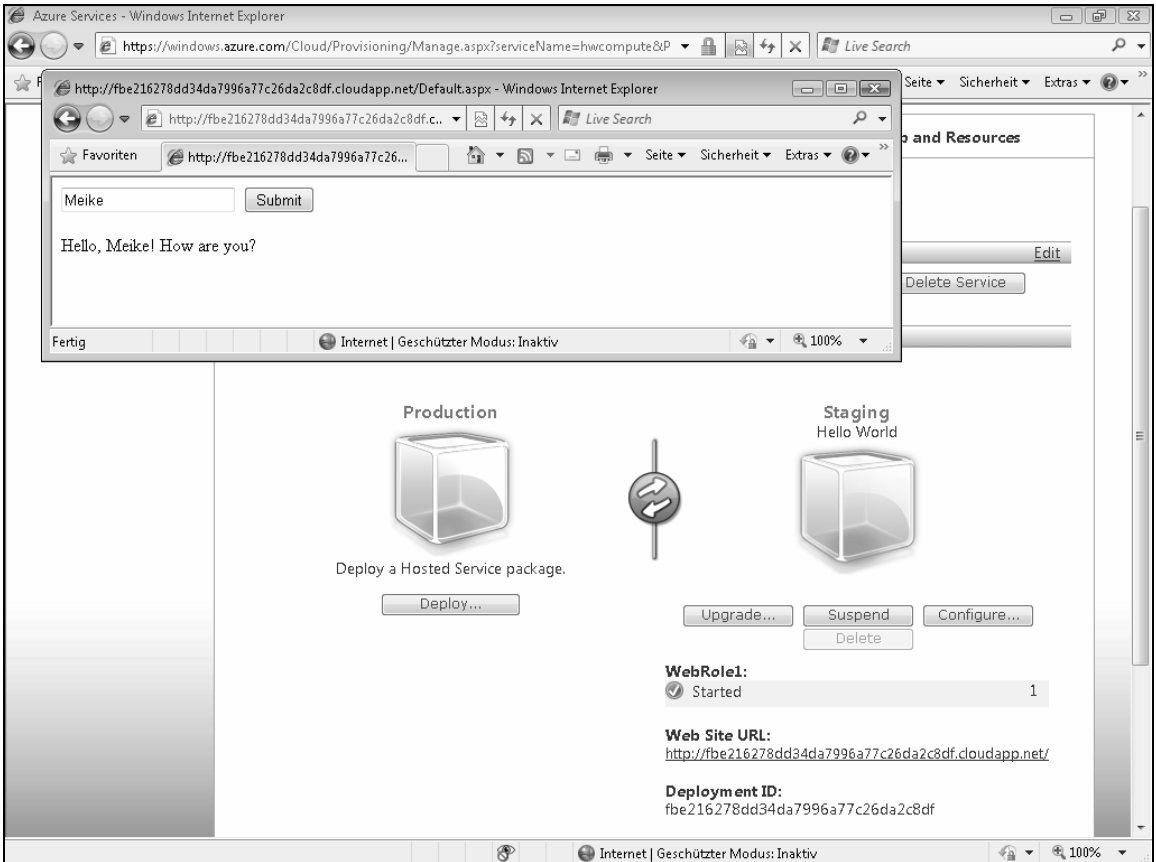


Abbildung 2.17 Test der »Hello World«-Anwendung in der Staging-Umgebung

Nachdem wir die Anwendung nun nach der lokalen Development Fabric auch auf der Staging-Umgebung in Azure erfolgreich getestet haben, fehlt nur noch der Transport in die Produktivumgebung, um die Anwendung unter dem zuvor gewählten URL verfügbar zu machen.

## Produktivsetzung auf Azure

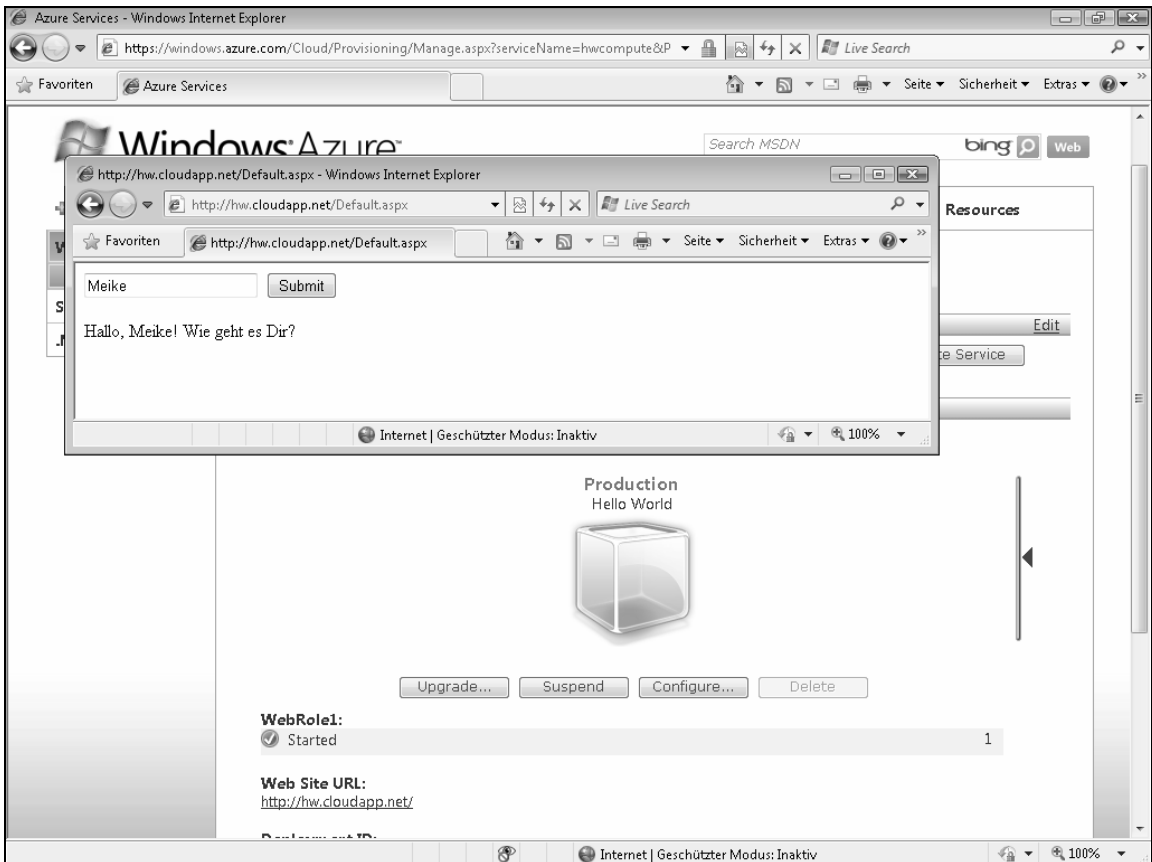
Zur Produktivsetzung klicken Sie auf den blauen Pfeil zwischen den Würfelsymbolen für die beiden Umgebungen. Bestätigen Sie die Rückfrage mit OK. Einen Augenblick später ist die Anwendung auf der Produktionsumgebung unter dem gewählten URL verfügbar. Die Geschwindigkeit des Transports erklärt sich dadurch, dass nicht wirklich ein physischer Transport der Anwendung erfolgt, sondern lediglich im Load-balancer die Zugriffweiterleitungen zu Staging- und Produktionsumgebung vertauscht werden.

Die Anwendung ist nun also produktiv gesetzt. Als letzter Schritt soll nun noch die Möglichkeit vorgestellt werden, im laufenden Betrieb die Konfiguration der Anwendung zu ändern. Klicken Sie hierzu auf die Schaltfläche *Configure*. Dies öffnet eine Eingabemaske, in der die zuvor hochgeladene Konfigurationsdatei eingetragen ist. Ändern Sie in dieser Konfigurationsdatei den Eintrag für die Grußformel wie folgt:

```
<ConfigurationSettings>
  <Setting name="Begrueßung" value="Hallo, {0}! Wie geht es Dir?" />
</ConfigurationSettings>
```

Ohne die Anwendung selbst zu ändern, können wir hierüber also die Grußformel ins Deutsche übersetzen. Die Anwendung erhält jeweils die aktuellste Einstellung. Diese Möglichkeit kann in anderen Szenarien beispielsweise dazu genutzt werden, um Logging-Einstellungen (z.B. Detailgrad) vorzunehmen.

Neben einem direkten Eintrag in die Konfigurationsdatei über das Eingabeformular könnte auch eine neue Konfigurationsdatei hochgeladen werden. Klicken Sie auf *Save*, um die geänderten Einstellungen abzuspeichern. Die Management-Konsole zeigt daraufhin wieder die in Abbildung 2.18 gezeigte Startseite des Projekts an.



**Abbildung 2.18** Ausführung der »Hello World«-Beispielanwendung

Dort kann die Anwendung nun über den eingangs gewählten URL aufgerufen werden. Klicken Sie also auf den Hyperlink unter dem Punkt *Web Site URL*, um ein Browserfenster mit der Anwendung zu öffnen. Dort können Sie, wie erwartet, ins Eingabefeld einen Namen eintragen und über Anklicken der Schaltfläche einen Gruß ausgeben lassen. Durch die Umkonfiguration im laufenden Betrieb erscheint dieser Gruß nun in deutscher Sprache.

## Zusammenfassung

Die Windows Azure Plattform ist Microsofts Plattform für die Cloud. Auf dem Fundament von Microsofts weltweit verteilten Rechenzentren, in denen die Global Foundation Services bereitgestellt werden, bietet Windows Azure Betriebssystemfunktionalitäten für Anwendungen, die in der Cloud betrieben werden sollen. Entwickler haben die Möglichkeit, eigene Anwendungen auf Windows Azure zu betreiben und verschiedene Services, die Azure Services, die von der Plattform angeboten werden, zu nutzen. Nachdem die Azure Services über Standardschnittstellen ansprechbar sind, können die Funktionalitäten der Plattform von beliebigen Technologien aus genutzt werden, die diese Standards unterstützen.

Bei der Entwicklung der Windows Azure Plattform hat Microsoft großen Wert darauf gelegt, Programmiermodelle weitestgehend an die Entwicklung klassischer .NET-Anwendungen anzupassen, um den Umstellungsaufwand für Entwickler zu minimieren. Durch Azure wird ihnen ein neues Betriebsmodell, das des Anwendungsbetriebs in der Cloud, zur Verfügung gestellt. Hierdurch werden Anwendungssysteme möglich, in denen für einzelne Anwendungskomponenten das jeweils bestmögliche Betriebsmodell gewählt werden kann.

Damit lassen sich Anwendungssysteme entwickeln, die – entsprechend Microsofts Software plus Services-Strategie – die Vorteile vor Ort betriebener Software mit denen von Cloud Services verknüpfen.