



**DavidChappell**  
& Associates

# WINDOWS AZURE PLATFORM

## の概要

本ホワイトペーパーは「Introducing the Windows Azure Platform」David Chappell 著の翻訳  
です。原文はこちらをご参照ください：

(<http://go.microsoft.com/?linkid=9752185>)

DAVID CHAPPELL

2010 年 10 月

SPONSORED BY MICROSOFT CORPORATION

## 目次

<b>Windows Azure Platform の概要</b> .....	<b>3</b>
Windows Azure .....	4
SQL Azure.....	6
Windows Azure AppFabric.....	8
Windows Azure Marketplace .....	10
<b>各テクノロジーの詳細</b> .....	<b>11</b>
Windows Azure .....	11
コンピューティング.....	12
ストレージ.....	13
ファブリック コントローラー .....	15
コンテンツ配信ネットワーク.....	16
Connect.....	16
SQL Azure.....	16
データベース.....	17
レポート.....	18
Data Sync.....	19
Windows Azure AppFabric.....	20
サービス バス .....	20
アクセス コントロール .....	22
キャッシュ .....	24
Windows Azure Marketplace .....	26
<b>将来的な展望</b> .....	<b>28</b>
<b>まとめ</b> .....	<b>28</b>
<b>執筆者について</b> .....	<b>29</b>

## WINDOWS AZURE PLATFORM の概要

クラウド内のコンピューターの利用には多くのメリットがあります。現在、インターネット経由でアクセスできる大量のサーバーが提供されていますが、自社でマシンを購入したり管理したりする代わりに、それらを活用してみたいかがでしょうか。アプリケーションによっては、コードとデータの両方をクラウド内で運用でき、アプリケーションが使用するシステムの管理と保守を第三者に任せることができます。代わりに、組織内で実行するアプリケーション、すなわちオンプレミスのアプリケーションのデータをクラウド内に保存したり、他のクラウド インフラストラクチャ サービスを利用したりすることも可能です。方法にかかわらず、クラウドの各種機能の活用は、利便性向上につながる可能性があります。

ただし、アプリケーションをクラウド内で運用する場合やクラウドで提供されるサービスを利用する場合、またはその両方を行う場合、ある種のアプリケーション プラットフォームが必要となります。広義に解釈すると、アプリケーション プラットフォームとは、開発者にとってアクセス可能なサービスを提供することによって、アプリケーションの作成やデータの格納が行えるようにするものすべて、と考えることができます。オンプレミスの Windows 環境には Windows Server や SQL Server といったテクノロジーが含まれます。各種アプリケーションでクラウドを利用するには、クラウド アプリケーション プラットフォームも用意する必要があります。

それこそが Microsoft Windows Azure Platform を提供する目的です。このプラットフォームは複数のクラウド テクノロジーの集合体であり、アプリケーション開発者向けに、それぞれが特定のサービスのセットを提供します。図 1 に、このプラットフォームのコンポーネントを示します。

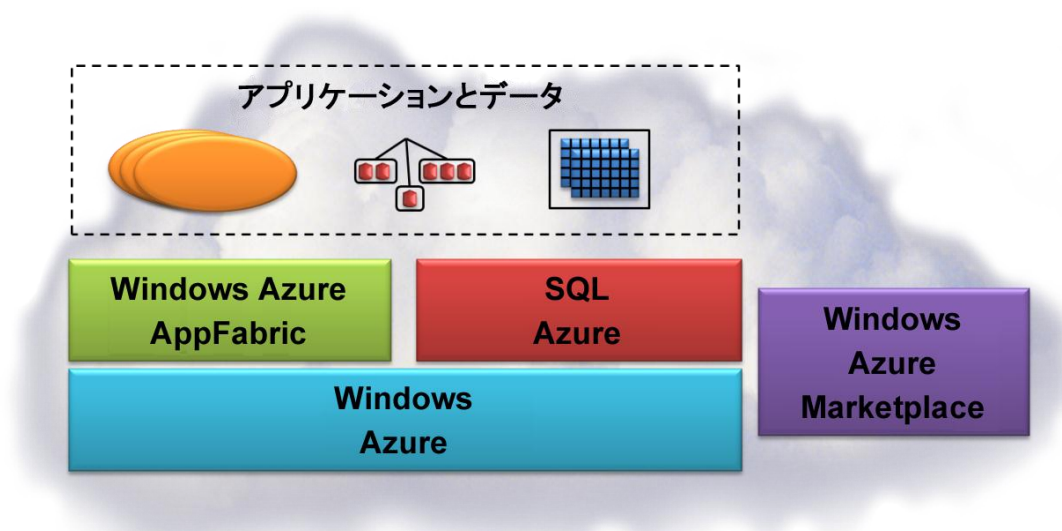


図 1: クラウド内の各種アプリケーションやデータ、インフラストラクチャ、クラウド向けのオンライン ストアをサポートする Windows Azure Platform

現在、Windows Azure Platform は 4 つのコンポーネントで構成されています。

- Windows Azure: マイクロソフト データ センター内のコンピューター上で、アプリケーションの実行とデータの保存を行う Windows 環境。
- SQL Azure: SQL Server をベースとした、クラウド用のリレーショナル データ サービス。

- Windows Azure AppFabric: クラウド内またはオンプレミスで実行されるアプリケーションを対象とした、クラウドベースのインフラストラクチャ サービス。
- Windows Azure Marketplace: クラウドベースのデータおよびアプリケーションを購入できるオンライン サービス。

これらの 4 つのコンポーネントはすべて、世界各地にあるマイクロソフトのデータ センター（北アメリカに 2 拠点、ヨーロッパに 2 拠点、およびアジアに 2 拠点）で実行されています。このプラットフォームを使用する開発者は、どのデータ センターを自身のアプリケーションの実行元にするか、どのデータ センターをデータの保存先にするかを制御でき、それらのデータ センターがユーザーに近い場所となるように設定できます。

Windows Azure Platform の各要素には、それぞれが果たすべき役割があります。本概要では 4 つの要素すべてについて、まず全体的な構造を紹介し、より詳細な内容へと説明を進めていきます。このクラウドプラットフォームの概要を包括的に理解していただくことを最終的な目標としています。

## WINDOWS AZURE

Windows Azure は、全体的にシンプルでわかりやすい構造をしており、Windows アプリケーションの実行機能とデータの保存機能をクラウド内で提供します。図 2 に、Windows Azure のコンポーネントを示します。

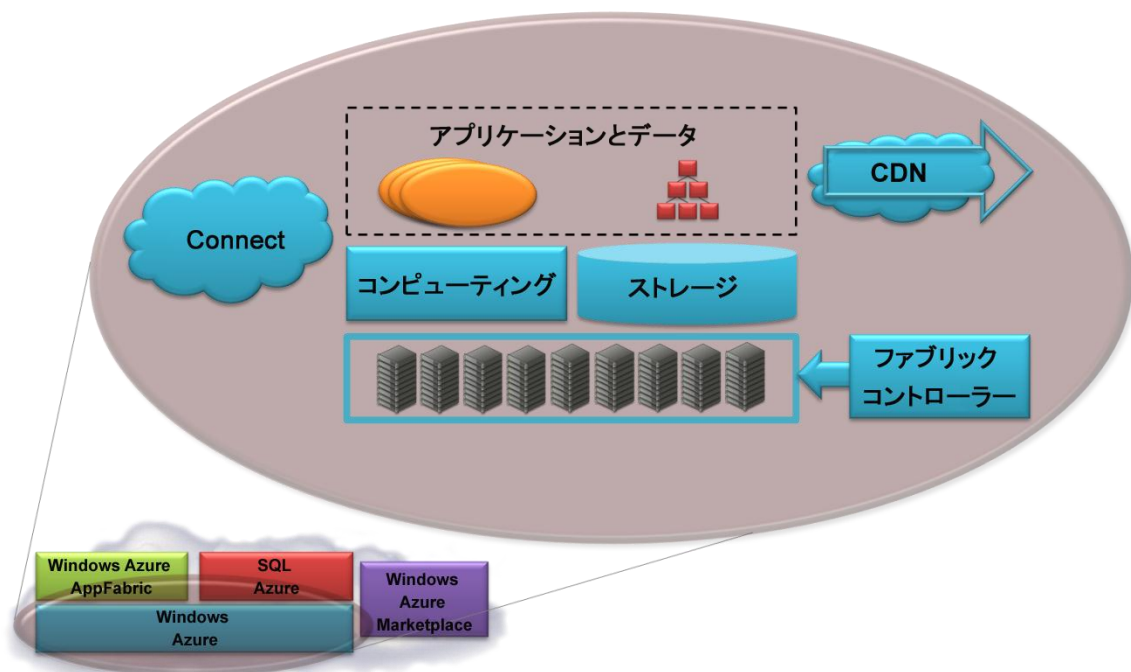


図 2: コンピューティング サービスとストレージ サービスをクラウド内で提供する Windows Azure

Windows Azure は現在、次の 5 つのコンポーネントで構成されています。

- コンピューティング: Windows Azure コンピューティング サービスは、Windows Server を基盤として各種のアプリケーションを実行します。こうしたアプリケーションは、.NET Framework を使用して C# や Visual Basic などの言語で作成できるほか、.NET を使用せずに C++ や Java などの言語で作成することもできます。開発者は Visual Studio を始めとする各種開発ツールを使用し、

ASP.NET、Windows Communication Foundation (WCF)、および PHP といったテクノロジーを自由に使用できます。

- ストレージ: このサービスでは、バイナリ ラージ オブジェクト (BLOB) にデータを保存し、キューを使用して Windows Azure アプリケーションのコンポーネント間で通信できるほか、シンプルなクエリ言語によるテーブルのフォームも利用できます (従来のリレーショナル ストレージを必要とする Windows Azure アプリケーションでは、SQL Azure を使用することもできます)。Windows Azure アプリケーションとオンプレミス アプリケーションは、いずれも Windows Azure ストレージ サービスにアクセスできます。アクセスには、両方とも REST ベースのアプローチを使用します。
- ファブリック コントローラー: 図が示すように、Windows Azure は多数のマシン上で実行されます。ファブリック コントローラーの役割は、1 つの Windows Azure データ センター内にある複数のマシンを連携させて一体化することです。この処理能力のプールを基盤とすることで、Windows Azure コンピューティング サービスおよびストレージ サービスを提供できるようになります。
- コンテンツ配信ネットワーク (CDN): 頻繁にアクセスされるデータをユーザーに近い場所にキャッシングし、対象のデータへのアクセス時間を短縮します。Windows Azure CDN ではこれを BLOB に対して行い、世界各地のサービス拠点にキャッシュ コピーを維持しています。
- Connect: このサービスは多くの場合、企業が自社のファイアウォール内と同様にクラウド アプリケーションと通信できるようにするために利用されます。Windows Azure Connect ではこれを、たとえば Windows Azure アプリケーションからオンプレミスのデータベースに簡単に接続できるようにすることで実現します。

クラウド内でのアプリケーションの実行とデータの保存は、明確なメリットをもたらします。たとえば、自社のシステムを購入、設置、および運用する代わりに、企業はクラウド プロバイダーを利用し、同じサービスを受けることができます。また、利用者はコンピューティング機能とストレージを使用した分だけ料金を支払えば済み、ピーク時の負荷のためだけに多数のサーバーを維持する必要はありません。さらに、Windows Azure 用に記述されたアプリケーションは、従来の Windows Server プログラミング モデルを使用して記述された場合よりもスケラブルで、信頼性が高く、管理の必要性も減少します。

Windows Azure の利用者は、アプリケーションの作成、設定、および監視を、ブラウザー経由でアクセスできるポータルから実行できます。利用者は Windows Live ID を使用してログインした後、アプリケーションを実行するための "ホスティング" アカウントか、データを保存するための "ストレージ" アカウントの、いずれかまたは両方を作成するよう選択します。マイクロソフトは、コンピューティング機能の利用時間、ストレージの利用量、および帯域幅の利用量に基づいて各利用者に課金します。アプリケーション自体の利用顧客に対する課金方法 (課金を行う場合) は、そのアプリケーションの作成者が自由に決定できます。

Windows Azure は多様なシナリオで使用できる汎用プラットフォームです。以下に、その一例を示します。

- 独立系ソフトウェア ベンダー (ISV) が、既存のオンプレミスの Windows アプリケーションをサービスとしてのソフトウェア (SaaS) に移植する場合は、Windows Azure 上でそのソフトウェアを構築できます。Windows Azure はほぼ標準の Windows 環境を提供するため、対象のアプリケーションのビジネス ロジックをこのクラウド プラットフォームへと移行しても、一般的にあまり問題は起こりません。さらにこの場合も、既存のプラットフォームを基盤とすることで、ISV は直接的に収益をもたらすビジネス ロジックに集中でき、インフラストラクチャに時間を割く必要はありません。
- 企業が自社の顧客や社員向けのアプリケーションを作成する場合、構築基盤として Windows Azure を選択することも可能です。Windows Azure は .NET をサポートするため、適切なスキルを

持つ開発者であれば、これがひどく高価な選択ではないことがすぐにわかるでしょう。マイクロソフト データセンターでアプリケーションを実行すると、社内サーバーの管理に伴う負担とコストが解消されるため、資本コストが運用コストに転換されます。特に、一時的に利用量が急増するようなアプリケーションの場合（たとえば、オンラインの生花店が母の日の注文ラッシュに対応しなければならない場合）は、この処理に必要な大規模なサーバー環境がマイクロソフトによって維持されるため、コスト削減に役立ちます。

- 起業したばかりの会社が、新しい Web サイト（たとえば、第二の Facebook など）を作成する場合は、Windows Azure 上でそのアプリケーションを構築できます。このプラットフォームは Web 向けサービスとバックグラウンド プロセスの両方をサポートしているため、アプリケーションではインタラクティブなユーザー インターフェイスを提供すると共に、ユーザーによる要求を非同期的に処理できます。これにより、起業直後の会社は、インフラストラクチャの準備に時間やコストをかける代わりに、自身の顧客と出資者に利益をもたらすコードの作成だけに集中できます。また、小規模から事業を開始し、アプリケーションのユーザー数が少ない間はコストを抑えることができます。アプリケーションの人气が高まってユーザーが増えた場合でも、Windows Azure なら必要に応じてアプリケーションを拡張できます。

この 3 つの事例は、企業が Windows Azure をどのように利用できるかを具体的に示していますが、用途はこれだけではありません。クラウド コンピューティングへの関心が高まるにつれ、このクラウド プラットフォーム用にさまざまなアプリケーションが作成されると予想されます。

## SQL AZURE

アプリケーションの実行と並ぶ、クラウドの魅力的な用途にデータの格納があります。SQL Azure はこの領域に対応し、クラウドベースのリレーショナル データ サービスを提供します。図 3 で示すように、SQL Azure は現在、3 つのコンポーネントで構成されています。

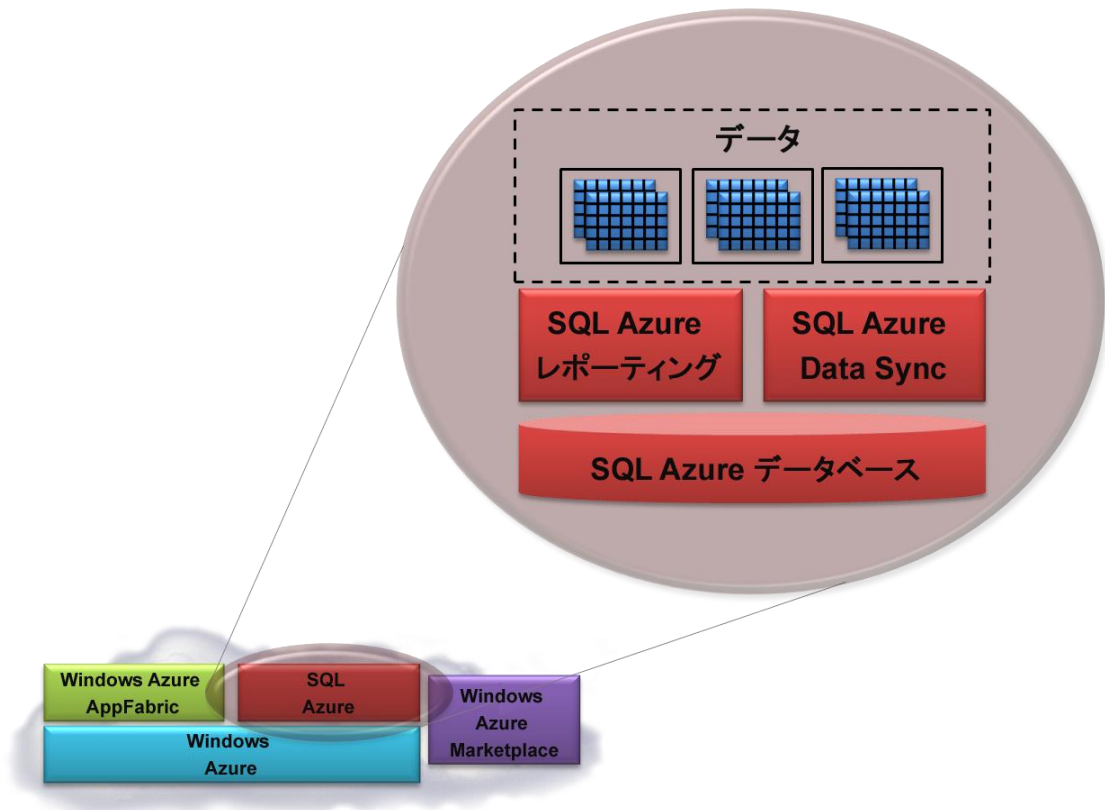


図 3: リレーショナル データベース サービスをクラウドで提供する SQL Azure

現在の SQL Azure のコンポーネントは次のとおりです。

- SQL Azure データベースは、クラウドベースのデータベース管理システム (DBMS) を提供します。このテクノロジーを使用すると、オンプレミスのアプリケーションおよびクラウド アプリケーションから、マイクロソフトのデータ センター内に配置されたマイクロソフト サーバー上にリレーショナル データを格納できます。他のクラウド テクノロジーと同様に、企業は使用量に応じて料金を支払えばよく、自社のニーズの変化に応じて使用量 (およびコスト) を増減できます。クラウド データベースの使用には、ディスクや DBMS ソフトウェアへの投資といった資本コストを、運用コストに転換する効果もあります。
- SQL Azure レポーティングは SQL Server Reporting Services (SSRS) のバージョンの 1 つで、クラウド内で動作します。主に SQL Azure データベースと併用することを目的としており、クラウド データに関する標準的な SSRS レポートの作成および発行機能を提供します。
- SQL Azure Data Sync を使用すると、SQL Azure データベースとオンプレミスの SQL Server データベースとの間でデータを同期できます。また、複数のマイクロソフト データ センター内の、さまざまな SQL Azure データベース間でデータを同期することも可能です。

SQL Azure は Microsoft SQL Server をベースに構築されています。SQL Server と同様に、開発者はインデックスとビューの作成、ストアド プロシージャの使用、トリガーの定義などを実行できます。アプリケーションから SQL Azure のデータにアクセスするには、Entity Framework、ADO.NET、およびその他各種の Windows データ アクセス インターフェイスを使用できます。実際に、現在ローカルで SQL Server にアクセスしているアプリケーションは、大きな変更を必要とせずに SQL Azure 内のデータを利用できます。利



用者は、SQL Server Analysis Services のようなオンプレミスのソフトウェアを使用して、クラウドベースのデータと連動させることも可能です。

各アプリケーションでは、SQL Azure をローカルの DBMS と同様に使用できると同時に、管理要件が大幅に削減されます。細かい運用はマイクロソフトが行うため、SQL Azure の利用者は、ディスク使用量の監視やログ ファイルの提供など、技術的な心配をする代わりに、自身のデータだけに集中できます。また、このクラウド プラットフォームの他のコンポーネントと同様に、利用者は Windows Azure Platform の共通ポータルからサービスにアクセスできます。

各アプリケーションでは、SQL Azure をさまざまな方法で利用できます。以下はその一例です。

- SQL Azure 内に Windows Azure アプリケーションのデータを保存できます。Windows Azure 自体にもストレージはありますが、提供されるオプションの中にリレーショナル テーブルは含まれません。多くの既存のアプリケーションにはリレーショナル ストレージが使用されており、多くの開発者がその活用方法を知っています。そのため、相当数の Windows Azure アプリケーションが SQL Azure を使用し、この馴染みのある方法でデータの処理を行っています。たとえば、Windows Azure 上に構築された SaaS アプリケーションで、利用者ごとに独立した SQL Azure データベースを作成するようにすれば、マルチテナント本来の設計が実現されます。また、利用者はパフォーマンス向上のため、特定の Windows Azure アプリケーションを実行するデータ センターが、必ず SQL Azure データベースによる対象のアプリケーションの情報格納先と同じとなるように指定できます。
- 小規模な企業や、より大きな組織の一部門で使用されるアプリケーションでは、SQL Azure の利用が効果的です。だれかのデスクの下に置かれたコンピューターで SQL Server や Access のデータベースを実行し、データを保存する代わりに、クラウド ストレージを利用することで、こうしたアプリケーションの信頼性と可用性が実現されます。また、このデータに関し、SQL Azure レポーティングかオンプレミスの SSRS を使用してレポートを作成することもできます。対象の組織内でもデータのコピーを維持する場合は、SQL Azure Data Sync を使用すれば、クラウドのデータベースとオンプレミスのデータベースを同期させることができます。
- あるメーカーが製品情報を自身の販売店ネットワークに提供し、さらに顧客にも直接提供しようと考えているとします。このデータを SQL Azure 環境に保存すれば、販売店側で実行されるアプリケーションや、メーカー自体が運用する顧客向けの Web アプリケーションからアクセスできます。

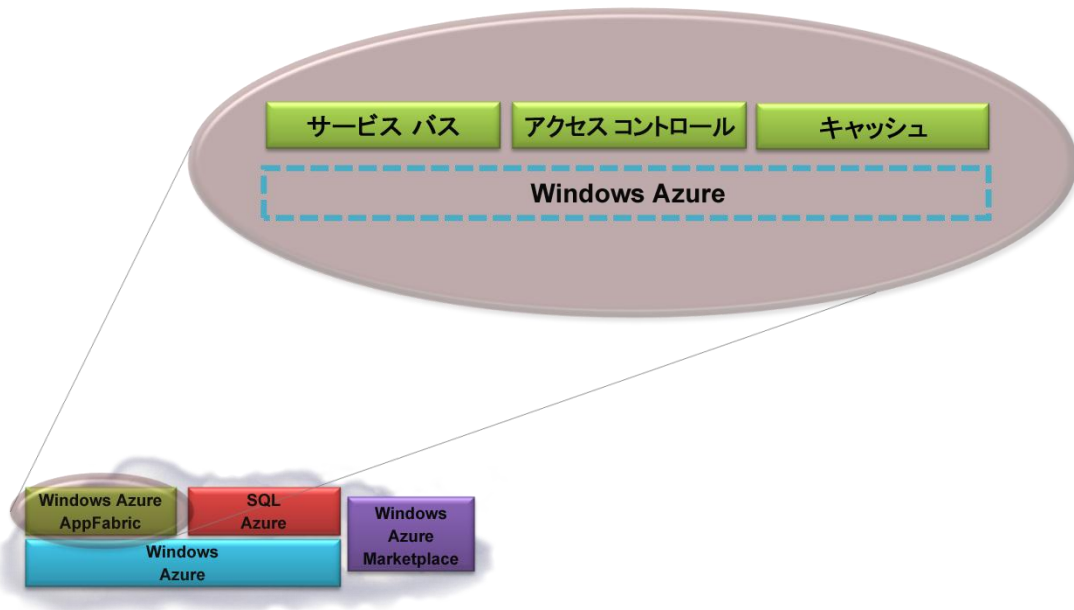
その目的が Windows Azure アプリケーションのサポート、データのアクセス性向上、またはそれ以外のいずれであるにかかわらず、クラウド内のデータ サービスには優れた利用価値があります。SQL Azure は、クラウド アプリケーションとオンプレミスのアプリケーションの両方が利用できる、馴染みのある方法でデータ サービスを提供することを目標としています。

## WINDOWS AZURE APPFABRIC

クラウド内でのアプリケーションの実行およびデータの保存は、いずれもクラウド コンピューティングの重要な側面ですが、クラウドにはほかにも多くの機能があります。クラウドをベースにすると、インフラストラクチャ サービスを提供することも可能です。このニーズを満たすのが Windows Azure AppFabric の目標です。

現在 AppFabric によって提供されている諸機能を利用すると、分散アプリケーションを構築する際の一般的な課題が解決されます。図 4 に、AppFabric のコンポーネントを示します。





**図 4: クラウド アプリケーションとオンプレミスのアプリケーションの両方にとって利用可能な Windows Azure ベースのインフラストラクチャ サービスを提供する Windows Azure AppFabric**

図に示すように、Windows Azure AppFabric のすべてのコンポーネントは Windows Azure を基盤としています (ただし、必ずしも Windows Azure アプリケーションだけにサービスを提供するわけではありません)。各コンポーネントの概要は次のとおりです。

- サービス バス: インターネット上でのアプリケーション サービスの提供は、口で言うほど簡単ではありません。サービス バスは、こうした作業を簡略化するために、オンプレミスやクラウド内で運用される他のアプリケーションからアクセスできるよう、アプリケーションがクラウド内でエンドポイントを公開する機能を提供します。公開された各エンドポイントには URI が割り当てられ、クライアントはその URI を使用することでサービスの場所を特定してアクセスできます。また、サービス バスはネットワーク アドレス変換の処理に伴う課題に対処し、公開されたアプリケーションに対して新たなポートをオープンすることなく、ファイアウォールを通過できるようにします。
- アクセス コントロール: 現在、ユーザーがデジタル ID を取得する方法は数多くあります。具体的には、Active Directory、Windows Live ID、Google アカウント、Facebook などが挙げられます。これらのどの ID を使用してもユーザーがアプリケーションにログインできるようにするには、上記のような数多くのアプローチをサポートする必要があり、アプリケーションの作成者にとってはきわめて困難な作業です。アクセス コントロールには、こうしたアプローチのすべて (およびそれ以外) のサポート機能が組み込まれているため、この作業を簡単に実行できます。また、各ユーザーのアクセス許可対象を制御するため、一元的にルールを定義する場所としての役割もあります。
- キャッシュ: アプリケーションは、同じデータに繰り返しアクセスするのが一般的です。こうした種類のアプリケーションを高速化する 1 つの方法が、頻繁にアクセスされる情報をキャッシングし、アプリケーションがデータベースを照会しなければならない回数を減らすというものです。キャッシュ サービスはこうした機能を Windows Azure アプリケーションに提供し、パフォーマンスを向上します。

マイクロソフトは、Windows Azure AppFabric にさらにサービスを追加する計画を発表しているため、近い将来にサービスのラインナップが拡充されると考えてください。

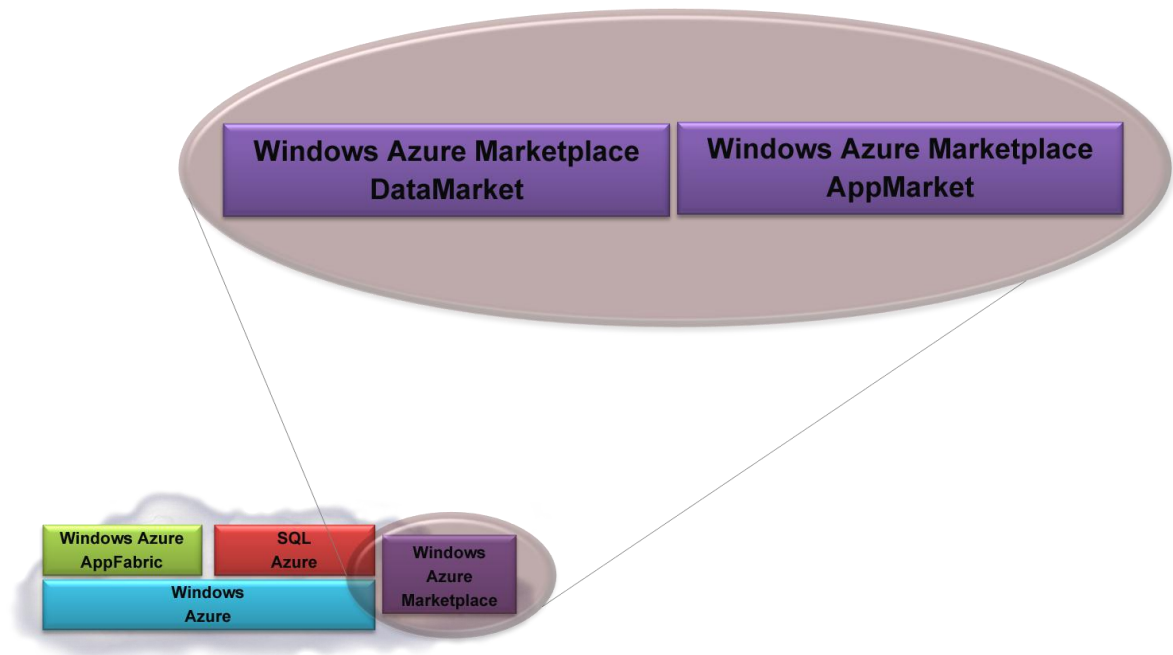
Windows Azure および SQL Azure と同様に、AppFabric にはブラウザー経由でアクセスできるポータルを使用してサインアップします。サインアップを完了すると、これらのサービスをさまざまな方法で使用できます。次に、その一例を示します。

- ある企業が自社のアプリケーションに、取引先で実行されているソフトウェアからアクセスできるようにしたいと考えているとします。この場合は、このアプリケーションの機能を、WCF を使用して作成された SOAP ベースまたは REST ベースのサービスを通じて公開した後、それらのサービス エンドポイントをサービス バスに登録します。これにより、取引先各社では、サービス バスを使用してこれらのエンドポイントを検索し、目的のサービスにアクセスできるようになります。
- 上記と同じアプリケーションで、取引先が複数の異なる ID を使用してログインできるようにする必要があります。対象の ID のサポート機能を自社で実装する代わりに、アクセス コントロール サービスを使用すれば、複雑な作業を行わずに済みます。
- ASP.NET を使用して作成された Windows Azure アプリケーションでは、Session オブジェクトを使用してクライアントごとのセッション情報を保存する場合があります。こうしたアプリケーションでは、構成設定さえ変更すれば、Windows Azure ストレージのテーブルなどではなく、対象のデータをキャッシュ サービスで保持するようにできます。これにより、アプリケーションの処理速度とスケーラビリティが向上する可能性があります。

Windows Azure AppFabric のクラウドベースの各種サービスに加えて、マイクロソフトは Windows Server AppFabric として知られる類似のテクノロジも提供しています。名前が示すように、このテクノロジが提供するサービスはオンプレミスのアプリケーションをサポートし、クラウド内ではなく Windows Server 上で動作します。現在、これらのオンプレミスのサービスは、Windows Azure AppFabric で提供されるものとすべて同じというわけではありません（ただし、マイクロソフトは 2 つのサービス内容を同一にする計画を発表しています）。本文書内で "AppFabric" という場合は、クラウドベースのサービスを指していますので、混同しないようにしてください。また、Windows Azure AppFabric と Windows Azure ファブリック コントローラーも混同しないように注意してください。両方とも "ファブリック" という言葉が含まれますが、これらは完全に独立したテクノロジであり、まったく異なる役割を持っています。

## WINDOWS AZURE MARKETPLACE

オンプレミス環境では、多くのアプリケーションが、独自に構築されるのではなく、購入されます。また、多くの企業は、商用プロバイダーが提供するデータセットを利用しています。クラウドの登場に伴い、クラウド アプリケーションを検索および購入する手段や、クラウド経由でアクセスできるデータが当然必要となるはずです。それを実現するのが Windows Azure Marketplace の目標です。図 5 に、Windows Azure Marketplace の 2 つのコンポーネントを示します。



**図 5: クラウド アプリケーションおよびデータのオンライン販売を行う Windows Azure Marketplace**

Windows Azure Marketplace は次の 2 つの要素で構成されます。

- DataMarket (当初のコードネームは "Dallas") は、コンテンツ プロバイダーがデータセットを提供する手段として使用されます。購入者は提供されるデータセットを閲覧し、利用できそうなものを自由に購入できます。購入したデータには、REST ベースの要求や OData プロトコルを通じて、カスタム アプリケーションまたは (Microsoft Excel のような) 市販のアプリケーションの両方からアクセスできます。
- AppMarket では、クラウド アプリケーションの作成者が、潜在的な顧客に対して自身のアプリケーションを公開できます。AppMarket は DataMarket の提供後、しばらくしてから利用可能となる予定です。

データおよびアプリケーションの検索、評価、および購入にクラウドを利用することは理にかなっています。Windows Azure Marketplace の役割は、これらをより簡単に行えるようにすることです。

## 各テクノロジーの詳細

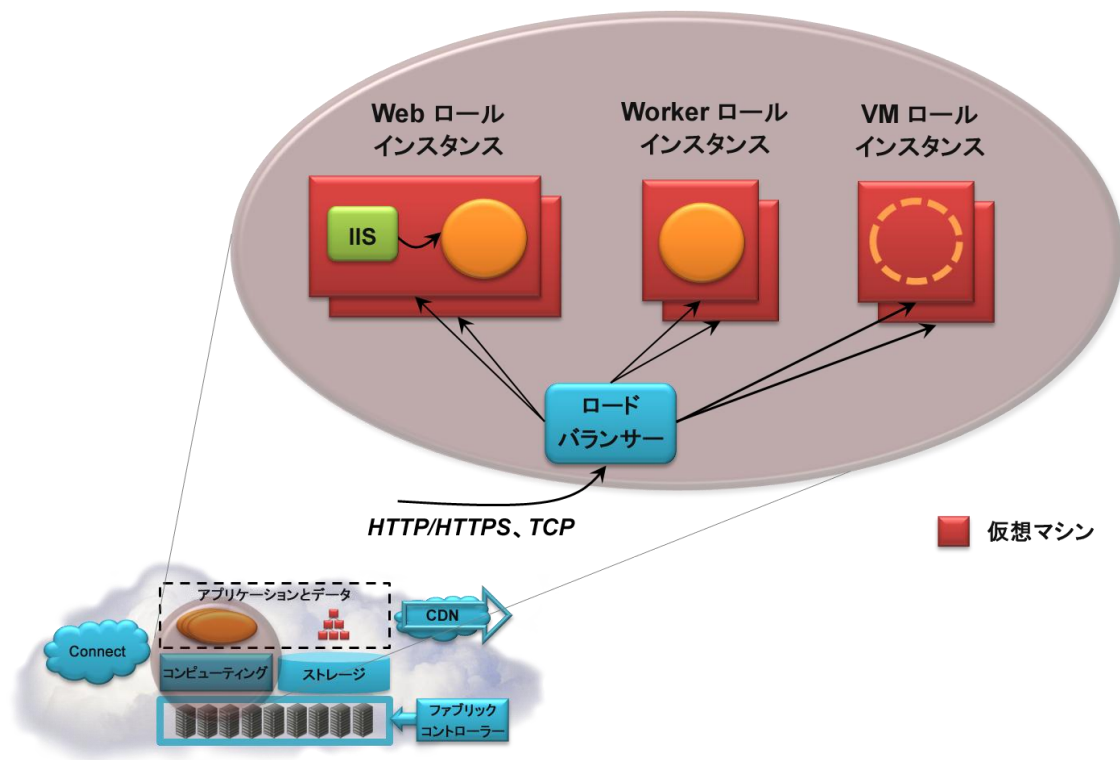
Windows Azure Platform について広範な知識を得ておくことは、重要な第一歩です。ただし、適切な意思決定を下すには、こうしたテクノロジーに関するより深い理解が必要となります。このセクションでは、ファミリ内の各サービスに関して、少し詳しく紹介していきます。

## WINDOWS AZURE

Windows Azure が提供するのは、一般に "サービスとしてのプラットフォーム (PaaS)" と呼ばれるサービスです。Windows Azure が提供するのは、オンプレミス環境のクラウド版ではなく、開発者と管理者の両方の業務を簡略化するような、より高次のサービスです。以降では、このサービスを全体で実現している 5 つのコンポーネントについて詳しく説明します。

## コンピューティング

Windows Azure コンピューティング サービス上に構築されたアプリケーションは、1 つ以上の "ロール" として構成されます。アプリケーションを実行すると、通常は各ロールの "インスタンス" が 2 つ以上実行され、それぞれのインスタンスは独立した仮想マシン (VM) として動作します。図 6 は、この様子を図で表したものです。



**図 6: 複数の Web ロール インスタンス、Worker ロール インスタンス、および VM ロール インスタンスの組み合わせからなる Windows Azure アプリケーションの実行**

Windows Azure アプリケーションは現在、次の 3 種類のロールを使用して作成できます。

- Web ロールは主に、Web ベースのアプリケーションの実行を目的としています。各 Web ロール インスタンスには、その内部で実行されるようあらかじめ構成されたインターネット インフォメーション サービス (IIS) 7 が含まれるため、ASP.NET や WCF などの各種 Web テクノロジを使用したアプリケーションを円滑に作成できます。PHP や Java など、マイクロソフト以外のテクノロジを使用したアプリケーションを作成することも可能です。
- Worker ロールは、さまざまなコードを実行するために設計されています。Worker ロールは、たとえばシミュレーションを実行する場合やビデオの処理を実行する場合など、ほぼすべての用途に使用できます。ユーザーが Web ロールを通じてアプリケーションとのやり取りを行った後、Worker ロールにタスクを渡して処理するのが一般的です。
- VM ロールは、ユーザーが指定した Windows Server 2008 R2 イメージを実行します。VM ロールの選択は、オンプレミスのいずれかの Windows Server アプリケーションを Windows Azure に移行する場合などに適しています。

実行するアプリケーションを Windows Azure にアップロードする際、開発者は同時に構成情報も提供します。この情報には特に、実行する各ロールのインスタンス数をプラットフォームに伝える役割があります。その情報に基づいて、Windows Azure ファブリック コントローラーがそれぞれのインスタンスに対応する VM を作成し、各 VM 内で適切なロールのコードが実行されます。図 6 に示すように、外部環境からの要求はロールの全インスタンスに負荷分散されます。

このことは、Windows Azure の各ロール インスタンス自体が、スケーラビリティ維持のため、要求間で自身のセッション状態を維持できない、という重要な意味を持っています。ロード バランサーが特定のロール インスタンスとアフィニティを構築することは許可されていない（スティッキー セッションがサポートされない）ため、同一のユーザーによる複数の要求は、必ずしも同じインスタンスに送られるとは限りません。代わりに、クライアント固有のセッション情報はすべて、Windows Azure ストレージに書き込まれるか、SQL Azure データベースに格納されるか、他のいずれかの方法によって外部で維持されます。

ロール インスタンスについて、Windows Azure では開発者が数種類の VM サイズを選択でき、サイズごとに固有のプロセッサ数とメモリ容量が決められています。各インスタンスに 1 つ以上のコアを割り当てることができるため、アプリケーションのパフォーマンスは予測可能です。また、アプリケーションの処理負荷を増やすために、アプリケーションの所有者またはアプリケーション自体が、1 つ以上のロールに対して実行インスタンス数を増やすように要求できます。すると、Windows Azure ファブリック コントローラーが要求されたインスタンスに対する新しい VM を作成し、それらの実行を開始します。ロールのインスタンス数を明示的に減らすことも可能です。負荷の変化に対応するため、必要に応じてアプリケーションを拡大したり、縮小したりできるようになっています。

開発者にとって、Windows Azure アプリケーションの構築は、従来の Windows アプリケーションの構築とほぼ同じです。開発者による Windows Azure アプリケーションの作成を支援するため、マイクロソフトは Visual Studio のプロジェクト テンプレートなどを提供しています。また、Windows Azure ソフトウェア開発キットには、開発者のマシン上で動作する Windows Azure 環境の一種が含まれています。これは Windows Azure "開発ファブリック" と呼ばれ、Windows Azure のコンピューティング サービスおよびストレージ サービスの、ローカル版レプリカを実装します。開発者は、このローカルのシミュレーション環境を利用してアプリケーションの作成とデバッグを行い、準備が整ってからクラウド内の Windows Azure に展開できます。Windows Azure はそのほかにも開発者向けのサービスを提供しており、実行中のアプリケーションによる CPU 消費量、受信側および送信側の帯域幅、ストレージ使用量などに関する情報を利用できます。

## ストレージ

---

アプリケーションでは多種多様な方法でデータが処理されます。こうした処理では、情報の格納にシンプルな BLOB を使用すれば十分な場合もあれば、より構造化レベルの高い方法が必要となる状況もあります。また、アプリケーションの異なる要素間でデータを交換する方法さえあれば十分な場合もあります。図 7 に示すように、Windows Azure ストレージを使用すれば、これらの 3 つの要件にすべて対応できます。



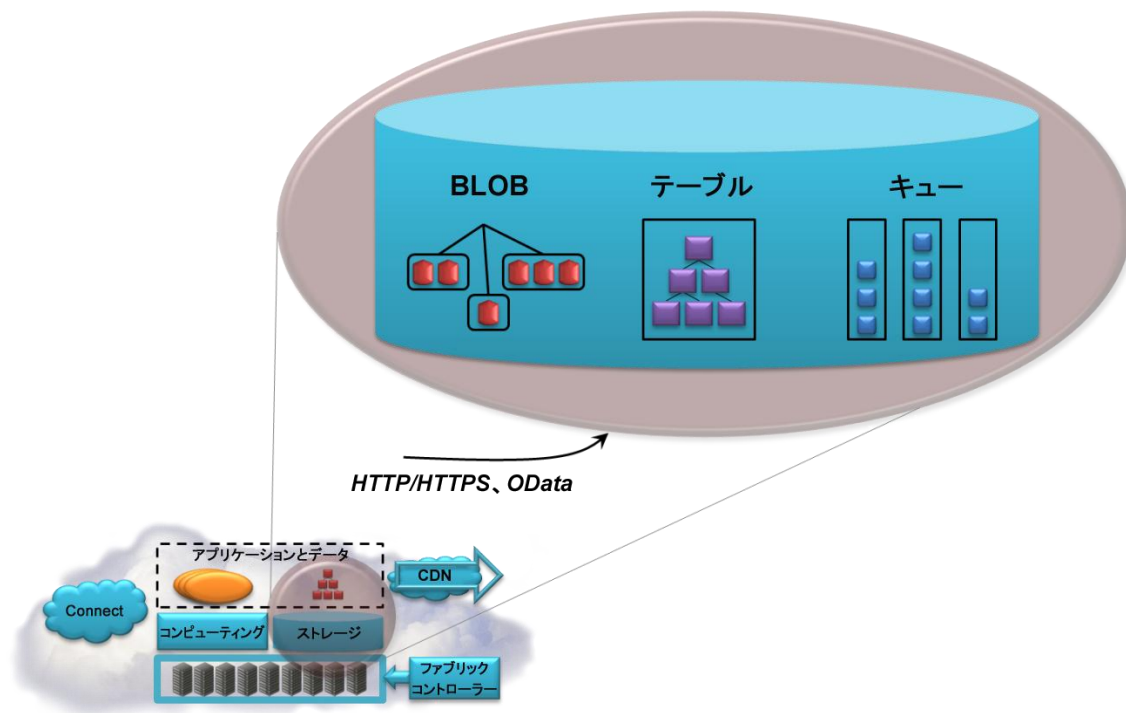


図 7: Windows Azure によって提供され、HTTP、HTTPS、または OData を介して REST ベースでアクセスできる BLOB、テーブル、およびキュー

Windows Azure ストレージにデータを保存する最も簡単な方法は、BLOB を使用することです。図 7 で示すように、BLOB はシンプルで階層構造になっています。ストレージ アカウントでは 1 つ以上のコンテナを保持し、それぞれのコンテナに 1 つ以上の BLOB を収容できます。各 BLOB は最大 1 テラバイトの大容量をサポートし、大規模な BLOB をより効率的に転送できるよう、複数のブロックに分割できる性質を持っています。障害が発生した場合は、BLOB 全体を送信し直さなくても、最後のブロックを再転送することで移動を再開できます。BLOB には、JPEG 形式の写真の撮影場所や MP3 ファイルの作曲者情報などをメタデータとして関連付けることもできます。

そのほか、ローカル インスタンスによってマウントできる Windows Azure ドライブを通じて BLOB を使用する場合もあります。ドライブの基盤となるストレージとして BLOB が使用されるため、ドライブのマウント後は、BLOB 内に永続的に格納されるファイル システム データを通じて、インスタンスが読み取りと書き込みを実行できます。

BLOB が最適な種類のデータもありますが、構造がシンプルすぎるため、適さない状況も多くあります。Windows Azure ストレージでは、より詳細なレベルでのアプリケーションのデータ処理を可能にするため、テーブルを提供しています。名前で誤解されやすいのですが、これはリレーショナル テーブルではありません。事実、名前こそ "テーブル" ですが、含まれるデータは、実際にはプロパティを持つ一連のエンティティ内に格納されています。テーブルには定義済みのスキーマがなく、代わりに、int、string、Bool、DateTime といった、さまざまな型がプロパティに適用されます。アプリケーションがテーブルのデータにアクセスする際は、SQL ではなく、OData によって定義されたシンプルでクエリ言語を使用します。1 つのテーブルはきわめて大きな容量をサポートし、何十億というエンティティで構成して数テラバイトのデータを保持することも可能です。Windows Azure ストレージでは必要に応じ、そのデータを分割して多数のサーバーに分散させ、パフォーマンス向上を図ることができます。

BLOB とテーブルはいずれも、データの保存に重点を置いています。Windows Azure ストレージの 3 つ目のオプションであるキューは、それとは目的が少し異なります。キューの主な役割は、Web ロール インスタンスと Worker ロール インスタンスとの間の通信を仲介することです。たとえば、ユーザーが、多くのコンピューティング処理を要するタスクの実行を、Windows Azure の Web ロールによって実装された Web ページ経由で要求したとします。この要求を受信した Web ロール インスタンスは、実行すべき作業を記述したメッセージをキューに書き込むことができます。すると、このキューを待機している Worker ロール インスタンスによってメッセージが読み込まれ、指定したタスクが実行されます。結果は他のキューから返される場合や、別のいずれかの方法で処理される場合があります。

格納方法 (BLOB、テーブル、またはキュー) にかかわらず、Windows Azure ストレージで保持されるすべての情報は 3 回複製されます。このレプリケーションによってフォールト トレランスが実現され、1 つのコピーが失われても致命的な結果にはなりません。このシステムでは一貫性が保証されるため、自身が書き込んだばかりのデータをアプリケーションが読み込んだ場合にも、目的のデータを取得できます。

Windows Azure ストレージには、Windows Azure アプリケーションと、他の場所で実行されているアプリケーションのどちらからもアクセスできます。いずれの場合も、Windows Azure ストレージの 3 つの形態ではすべて、REST の規則 (テーブルでは OData プロトコルも可) を使用してデータの特定と提供が行われます。すべての要素には URI を使用して名前が付けられ、標準の HTTP 処理を通じてアクセスできるため、クライアントの作成には .NET や Java のほか、各種の使い慣れたテクノロジーを使用できます。

Windows Azure プラットフォームでは、コンピューティング リソースとストレージ リソースへの課金が独立して行われます。すなわち、オンプレミスのアプリケーションから Windows Azure ストレージだけを使用することが可能であり、先ほど説明した REST ベースの方法でアプリケーション データにアクセスできます。たとえば、企業のデータ センター内で実行される Windows Server アプリケーションについて、Windows ストレージの BLOB にバックアップを保存することも可能です。

## ファブリック コントローラー

---

Windows Azure アプリケーションは VM 内で実行されますが、こうした VM は開発者が明示的に作成および管理するわけではありません。代わりに、前述の方法で、開発者はそのアプリケーションが必要とするインスタンス数をプラットフォームに伝えさえすれば、Windows Azure が自動的に必要な VM を作成し、対象のアプリケーションを実行します。この魔法のような操作は、Windows Azure の基本構成要素であるファブリック コントローラーによって実行されています。

ファブリック コントローラーは、VM を作成し、VM 内で実行する各種アプリケーションを起動するほかに、実行中のすべてのインスタンスの監視も行います。ロール インスタンスは、コードによって例外がスローされた場合や、VM のクラッシュのほか、インスタンスを実行している物理サーバーがダウンした場合など、多くの理由から障害を起こす可能性があります。理由にかかわらず、ファブリック コントローラーは新しいインスタンスを起動して、このアプリケーションで必要とされるインスタンスの合計数を維持します。

また、ファブリック コントローラーは、オペレーティング システムおよびその他のシステム ソフトウェアへのパッチの適用といったハウスキーピング タスクも実行するため、管理者の手間が最小限に抑えられます。これらはすべて、Windows Azure が PaaS と解釈される理由の一例です。Windows Azure は、利用者が、実行環境の管理に気を配ることなく、アプリケーションを実行できるプラットフォームを提供します。



## コンテンツ配信ネットワーク

---

BLOB の一般的な用途の 1 つに、さまざまな場所からアクセスできる情報の格納が挙げられます。世界中の Flash、Silverlight、または HTML 5 のクライアントにビデオを提供するアプリケーションがその例です。こうしたシナリオでのパフォーマンス向上を目的として、Windows Azure はコンテンツ配信ネットワーク (CDN) を提供しています。CDN では、BLOB のデータを使用するクライアントの近くのサービス拠点に、BLOB のコピーが保持されます。これにより、頻繁にアクセスされるコンテンツの配信時間が短縮され、この情報のユーザーのパフォーマンスが向上します。

## Connect

---

Windows Azure アプリケーションは、HTTP、HTTPS、または TCP を介して、外部環境と相互に通信できます。一方、アプリケーション内のロールを、クラウドの外部にあるマシンと IP レベルで接続する場合を考えてみます。たとえば、ある組織が、既存の ASP.NET アプリケーションを Windows Azure に移行することを決定したとします。ただし、アプリケーションのデータは引き続き、オンプレミスの SQL Server データベースで維持することを希望しています。この場合に役立つのが Windows Azure Connect です。

このオプションを使用するには、SQL Server がインストールされた社内マシン上で Windows Azure Connect ソフトウェアを実行する必要があるほか、簡単な設定が多少必要となります (この設定は、ネットワーク管理者の手を借りる必要はなく、開発者が自分で実行できます)。この設定が終わると、Web ロール内の全インスタンスを、SQL Server マシンと同じ IP ネットワーク上にあるマシンと同様に操作できます。実際、これらのインスタンスでは、アプリケーションをオンプレミスで実行する場合と同じ SQL Server 接続文字列を使用できるため、違和感を覚えることはありません。

これが本格的な仮想プライベート ネットワーク (VPN) ではない点に注意してください。マイクロソフトは、Windows Azure に VPN 経由でアクセスできるようにする計画を発表していますが、現在はサポートされていません。ただし、Windows Azure Connect を使用すると、Windows Azure アプリケーションをオンプレミス環境の Active Directory ドメインに参加させることができます。これにより、オンプレミス環境のユーザーによる、クラウド アプリケーションへのシングル サインオンが可能になります。また、このアプリケーションでは、既存の Active Directory のアカウントとグループを使用してアクセス コントロールを実施できます。

アプリケーション プラットフォームの目標とは、アプリケーションとデータを、オンプレミスかクラウドかにかかわらずサポートすることです。Windows Azure はこれらの両方をサポートしています。将来的に、従来はオンプレミスの Windows アプリケーションとして提供されていた機能のかなりの部分が、この新しいクラウド プラットフォーム上で実行されるようになる予定です。

## SQL AZURE

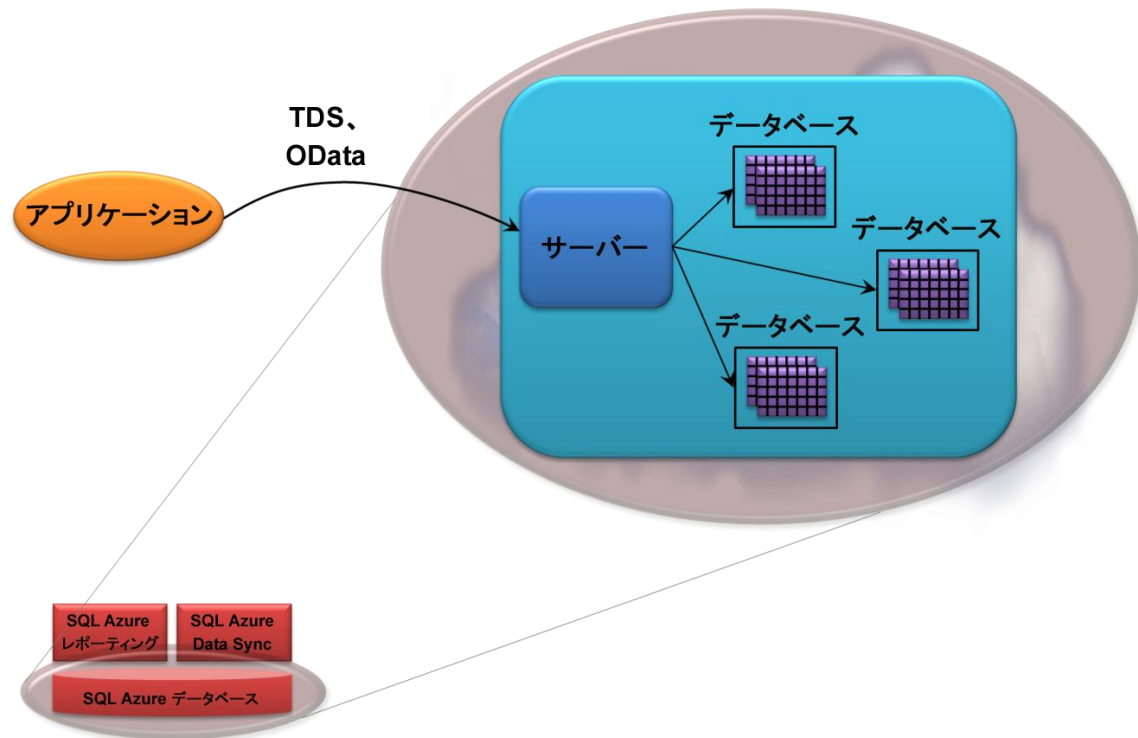
---

クラウド内の DBMS は、多くの理由から利用価値に優れています。専門のサービス プロバイダーに信頼性の確保と基本的な管理業務の実施を依頼することは、特にそうした業務を自社で行うことが難しい組織にとっては、理にかなった対応といえます。また、クラウドにデータを保存すれば、モバイル デバイスを含む、あらゆる場所で実行されるアプリケーションからアクセスできます。サービス プロバイダーがもたらすスケール メリットを考えれば、オンプレミス データベースよりもクラウド データベースを利用する方が、コストを抑えられる可能性があります。

SQL Azure はこうしたメリットに加えて、クラウドベースのレポート サービスとデータ同期サービスを提供します。こうした機能はいずれも重要ですが、まずは基盤となる SQL Azure データベースについて説明します。

## データベース

SQL Azure データベースの機能を理解するのは簡単です。アプリケーションから見た場合、SQL Azure データベースは、SQL Server の主要データベース機能をクラウド サービスとして提供します。図 8 に、このテクノロジーの基本構造を示します。



**図 8: マイクロソフトの TDS プロトコルまたは OData を通じて SQL Azure データベース内のデータにアクセスするアプリケーション**

SQL Azure データベースを使用するアプリケーションは、Windows Azure 上、企業のデータ センター内、モバイル デバイス上などで実行できます。実行場所にかかわらず、こうしたアプリケーションは通常、データへのアクセスに表形式データ ストリーム (TDS) と呼ばれるプロトコルを使用します。これはローカルの SQL Server データベースへのアクセスに使用されるのと同じプロトコルであるため、SQL Azure データベース アプリケーションは、任意の既存の SQL Server クライアントを使用できます。これには、Entity Framework、ADO.NET、ODBC、PHP などが含まれます。また、SQL Azure データベースは通常の SQL Server システムと類似しているため、SQL Server Management Studio、SQL Server Integration Services、およびデータの一括コピーを行う BCP といった、各種の標準ツールも利用できます。図 8 に示すように、アプリケーションから SQL Azure データベースにアクセスする際、任意で OData を使用することもできます。

各 SQL Azure アカウントでは、1 つ以上の論理サーバーを利用できます (これはアカウントのデータおよび課金を整理するための機能であり、SQL Server の実際のインスタンスではありません)。さらに、サーバーごとに、50 ギガビットまでのサイズのデータベースを複数保持できます。ユーザーは、パフォーマンス向上のためにデータを分散させるなどの理由で、必要に応じて自由に複数のデータベースを使用できます。1 つの SQL Azure データベースのスナップショットを他のデータベースに保存し、シンプルなバックアップ メカニズムとして使用することも可能です。

SQL Azure データベースを使用するアプリケーションは、馴染みのある SQL Server 環境とほぼ同じと考えることができます。ただし、SQL 共通言語ランタイム (CLR) や、全文検索のサポートなど、このテクノロジーの現在のリリースにはいくつかの機能が含まれていません (マイクロソフトは、将来のバージョンではこれらを提供すると発表しています)。また、管理作業はマイクロソフトが行うため、このサービスでは物理的な管理機能が提供されません。たとえば、利用者はシステムをシャットダウンしたり、システムを実行しているハードウェアと直接通信したりすることはできません。また、共有環境では一般的ですが、クエリの実行時間が制限され、単一の要求では、あらかじめ定義された量を超えるリソースは使用できません。

この環境はきわめて標準的に見えますが、しかしアプリケーションに提供されるサービスは、単一の SQL Server のインスタンスの場合よりも安定性に優れています。Windows Azure ストレージと同様に、SQL Azure データベースに格納されたデータはすべて、高可用性を維持するために 3 回複製されます。また、Windows Azure ストレージと同様に、このサービスは強力な一貫性を備えています。書き込み処理結果が返された時点で、対象のデータは永続化されています。このサービスの目的は、システム障害やネットワーク障害が発生した場合でも、信頼できるデータ ストレージを提供することです。

アプリケーションが必要とするデータベースの数が複数であるか 1 つだけであるにかかわらず、開発者は SQL Azure データベースを広範なシナリオで利用できます。解決する問題の内容を問わず、このテクノロジーの基本的な目標が、馴染みのある、信頼性に優れた低コストのクラウド データベースを多様なアプリケーションに提供することである点に変わりはありません。

## レポート

---

SQL Azure データベースでのデータの保存は便利です。ただし、データが存在するからには、そのデータに基づいたレポートを求められるのが一般的です。こうした要求に応えるのが SQL Azure レポーティングの目的です。このテクノロジーは、SQL Server Reporting Services (SSRS) に基づいて、クラウドベースでレポートを作成する手段を提供します。

アプリケーションから見た SQL Azure データベースは SQL Server と類似しているため、オンプレミスで運用する SSRS は、いつでも SQL Azure データベース内のデータに使用できます。では、わざわざクラウド レポート サービスを提供するのはなぜでしょうか。SQL Azure レポーティングは現在、主に 2 つのシナリオをターゲットとしています。

- SQL Azure レポーティングを使用して作成したレポートは、SQL Azure レポーティングのポータルに発行でき、ユーザーはポータル経由でレポートにアクセスするか、URL を指定して直接アクセスできます。
- ISV は発行されたレポートを、Windows Azure アプリケーションなど、任意のアプリケーション内の SQL Azure レポーティング ポータルに埋め込むことができます。これにより、ユーザーが使用中のアプリケーション内からレポートにアクセスできるようになります。これを実現するため、ISV は Visual Studio に標準で含まれる ReportViewer コントロールを利用することがあります。方法は、社内データのレポートをアプリケーションに埋め込む場合と変わりません。

SQL Azure レポーティングは、SQL Azure データベースに格納されているデータと連携するように設計されています。ただし、SQL Azure レポーティングで使用されるレポートは、Business Intelligence Developer Studio を使用して社内で作成されます。これは Visual Studio によって提供されるツールで、SSRS レポートの作成に使用されます。実際に、SQL Azure レポーティングは、レポート定義言語 (RDL) で記述された、SSRS で使用されるのと同じレポート形式を使用しています。

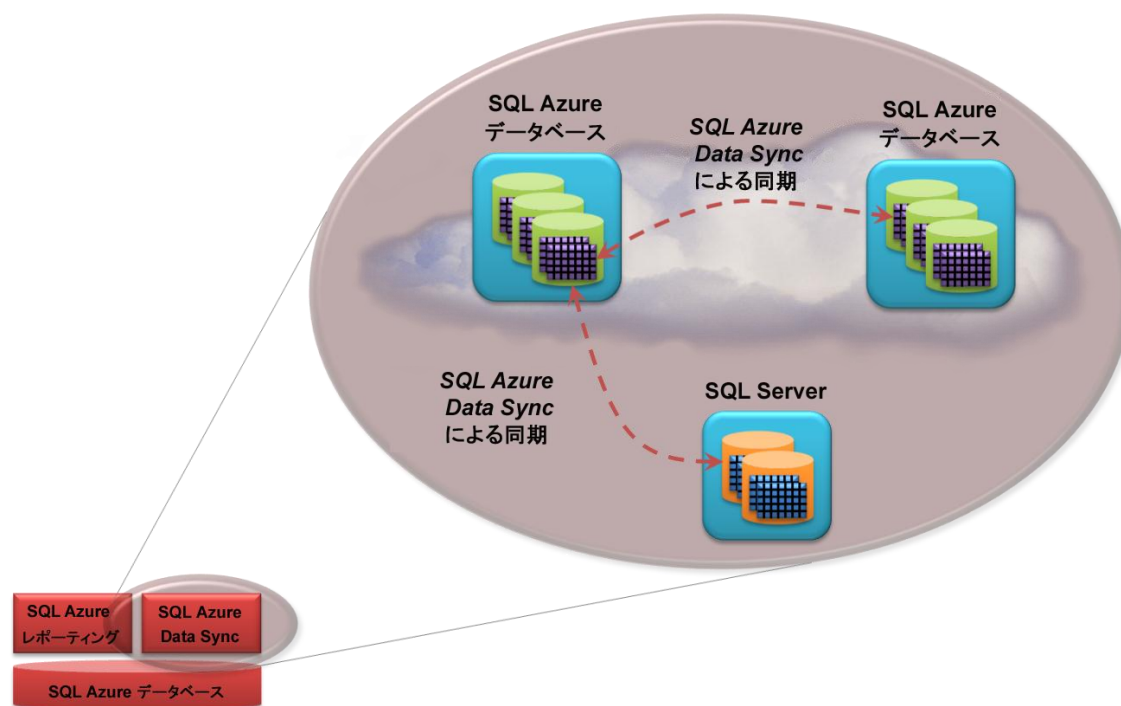
SQL Azure の他の要素と同様に、SQL Azure レポーティングでは、オンプレミスで運用する SSRS のすべての機能が提供されるわけではありません。たとえば現在は、レポートを定期的なペースで実行および配信

するための、スケジュールとサブスクリプションの機能がサポートされていません。その点を差し引いてもクラウドベースのレポート サービスには利用価値があり、SQL Azure テクノロジ ファミリの重要な部分を占めています。

## Data Sync

SQL Azure データベースにデータを格納すると、インターネットに接続可能なすべてのアプリケーションからそのデータにアクセスできるようになります。ただし、このクラウドベースのデータをコピーして、他のいずれかの場所で維持するのが適切な状況も少なくありません。たとえば、パフォーマンス上の理由や、ネットワーク障害の際にアクセスを確保する目的で、組織が同じ情報の社内コピーを必要としている場合を考えてみましょう。こうしたシナリオでは、SQL Azure データベース内のデータと他のデータ ストアを同期できると便利です。

Microsoft Sync Framework を使用すれば、いつでも自分で同期用のコードを記述できます。この作業を簡単にするために、マイクロソフトは SQL Azure Data Sync を提供しています。ユーザーがコードを記述する代わりに、このテクノロジーは完全に構成ベースで機能します（ただしご想像のとおり、このテクノロジーは Microsoft Sync Framework をベースとしています）。図 9 が示すように、SQL Azure Data Sync は現在、2 つのオプションをサポートしています。



**図 9: SQL Azure データベース間、または SQL Azure データベースとオンプレミスの SQL Server データベースとの間でデータを同期できる SQL Azure Data Sync**

この 2 つの同期オプションは次のとおりです。

- SQL Azure データベースとオンプレミスの SQL Server データベースとの間で行うデータの同期。クラウド上のデータをオンプレミスで複製しておく、さまざまな理由からメリットがあります。たとえば、ネットワーク障害の発生時にも、組織がこのデータに常にアクセスできるようにしておきたい場合もあれば、規制に準拠するため、常に国境内で対象のデータのコピーを利用できるようにしておく必要がある場合もあります。また、SQL Azure にはハードウェア障害に備え、データを複製して保護する機能が組み込まれていますが、データ所有者の裁量によりオンプレミスでバックアップ コピーを保持すれば、テーブルを誤って削除した場合など、管理作業のミスに備えることができます。



- さまざまなマイクロソフト データ センター内の SQL Azure データベース間で行うデータの同期。たとえば、ISV やグローバルな企業で、世界中のユーザーが使用するアプリケーションを作成した場合を考えます。ユーザーすべての良好なパフォーマンスを実現するため、作成者がこのアプリケーションを、3 か所の Windows Azure データ センター（北米、ヨーロッパ、およびアジアに 1 か所ずつ）で実行するとします。このアプリケーションのデータを SQL Azure データベース内に格納しておけば、SQL Azure Data Sync を使用して、これらの 3 つのデータ センター間で常に情報を同期させることができます。

SQL Azure Data Sync はハブアンドスポーク モデルを採用しています。すべての変更は最初に SQL Azure データベースのハブにコピーされ、その後にはスポーク ("メンバー" と呼ばれる) へとコピーされます。メンバーは、他の SQL Azure データベースでもオンプレミスの SQL Server データベースでもかまいません。どちらの場合も、このテクノロジーではデータベースの全体または特定のテーブルのみを同期でき、その他の場所に作成されたコピーにもすべて変更が反映されます。また、同期は手動で開始することもできますが、SQL Azure Data Sync にはスケジュール サービスも含まれています。この機能を使用すると、たとえば 1 時間ごとのように、2 つのデータベース間で同期スケジュールを設定できます。使用方法にかかわらず、目的は常に同じです。Data Sync は、マイクロソフト データ センター内の SQL Azure データベースとそれ以外の場所にある 1 つ以上のデータベースとの間でデータを同期する簡潔な手段を提供します。

## WINDOWS AZURE APPFABRIC

アプリケーションが存在するのは、そのビジネス ロジックに価値があるためです。ただし、アプリケーションには、そのロジックの基盤を提供するインフラストラクチャが不可欠です。優れたアプリケーション プラットフォームではこうしたインフラストラクチャが提供されるため、アプリケーション開発者が自らインフラストラクチャを記述する必要はありません。

Windows Azure AppFabric は、アプリケーションにインフラストラクチャを提供します。アプリケーションの作成者に応じて、利用価値のあるインフラストラクチャの種類はさまざまです。そのため、AppFabric は多彩な要素で構成されています。このセクションでは、サービス バス、アクセス コントロール、およびキャッシュという、現在 Windows Azure AppFabric に含まれている 3 つのコンポーネントについて詳しく見ていきます。

### サービス バス

Windows Communication Foundation (WCF) を使用して構築された Web サービスを提供するアプリケーションを、オンプレミスで実行している場合を考えます。さらに、インターネットを通じ、このサービスをオンプレミス環境の外部で実行されるソフトウェアと接続するとします。このクライアント ソフトウェアは、Windows Azure などのクラウド プラットフォーム上で実行されていても、他の組織内で実行されていてもかまいません。

一見すると、これはなんということのない問題に思われます。既存のアプリケーションでは Web サービス (REST ベースまたは SOAP ベース) を通じて機能を提供しているため、単にその Web サービスを外部環境から見えるようにすればよいのです。ただし、実際にこれを実現しようとすると、いくつかの問題が発生します。

まず、他の組織内のクライアントから、どうすれば対象のサービスに接続するためのエンドポイントを見つけられるでしょうか。外部からアプリケーションの場所を特定できるよう、ある種のレジストリがあると便利です。また、アプリケーションの場所がわかっても、他の組織内にあるソフトウェアの要求を、自社のサービスで受け取るにはどうすればよいでしょうか。ネットワーク アドレス変換 (NAT) がきわめて一般的に使用されているため、多くの場合、アプリケーションには外部に公開すべき固定 IP アドレスが割り当てられていません。さらに、たとえ NAT が使用されていなくても、どうすれば要求がファイアウォールを通過できるでしょうか。対象のアプリケーシ

オンへのアクセスを許可するため、ファイアウォール ポートをオープンにすることは可能ですが、ネットワーク管理者は賛成しないでしょう。

サービス バスはこれらの課題を解決します。図 10 に、その方法を示します。

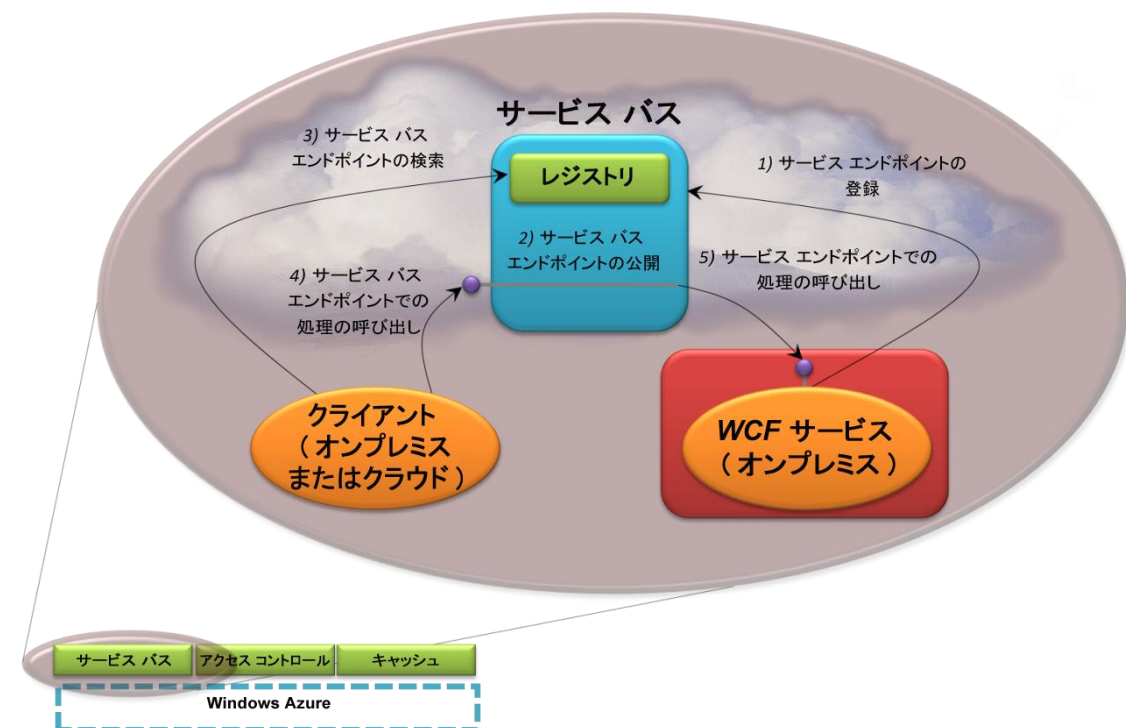


図 10: クライアントは、WCF サービスがサービス バスに登録したエンドポイントを検索し、そのエンドポイントを使用してサービスにアクセス

最初に、社内の WCF サービスから、1 つ以上のエンドポイントをサービス バスに登録します (ステップ 1)。登録された各エンドポイントについて、対応する専用のエンドポイントをサービス バスが公開します (ステップ 2)。サービス バスによって、利用者の組織に URI ルートが割り当てられるので、利用者は任意の名前を使用して階層を作成できます。これにより、使用するエンドポイントに対し、検索可能な固有の URI を割り当てることができます。

クラウド内または他の組織の社内で実行されているクライアントから利用者のサービスにアクセスするには、クライアントからサービス バスのレジストリにアクセスし (ステップ 3)、目的のエンドポイントの URI を入力してそのエンドポイントを検索します。この要求には Atom Publishing Protocol が使用されます。そして、サービス バスが利用者のアプリケーションに代わって公開するエンドポイントの参照先を記載した、AtomPub サービス ドキュメントが返されます。このドキュメントを受け取ると、こうしたエンドポイントを通じて公開されるサービスで、クライアントが処理を呼び出せるようになります (ステップ 4)。サービス バスは要求を受信するたびに、利用者の WCF サービスによって公開されるエンドポイント内で、対応する処理を呼び出します (ステップ 5) (図には示されていませんが、サービス バスはアプリケーションとクライアントとの間でできる限り直接接続を確立し、通信の効率化を図ります)。

ここで、ステップ 5 は具体的にどのように機能しているのか、とだれもが疑問に思うでしょう。サービス バスから利用者のサービスに返される要求では、NAT およびファイアウォールの課題にどのように対処しているのでしょうか。答えは、ステップ 1 で、利用者のサービスが、この公開されたエンドポイントに対してサービス バスとの TCP 接続をオープンした、ということになります。サービス バスがこの接続をオープンのまま維持することで、2

つの問題が解決されます。1 つ目に、NAT の問題が解消されます。これは、サービス バスとの間でオープンされた接続上のトラフィックは、常に利用者のアプリケーションへとルーティングされるためです。2 つ目に、この接続はファイアウォールの内部から開始されているため、この接続を介して情報をアプリケーションへと渡しても問題は起こりません。ファイアウォールがトラフィックをブロックすることはありません。

通信の利便性を高めると同時に、サービス バスはセキュリティ向上にも役立ちます。各クライアントから見えるのはサービス バスが提供する IP アドレスだけであるため、利用者の組織内の IP アドレスは一切公開する必要がありません。これにより、外部環境からは対象のアプリケーションが見えなくなるため、効果的に匿名性を実現できます。サービス バスは外部の DMZ として機能し、攻撃を阻止するための間接レイヤーを提供します。

サービス バスを介してサービスを公開するアプリケーションは、WCF を使用して実装されるのが一般的ですが、クライアントは WCF のほかに、Java のような他のテクノロジーを使用して構築できます。作成方法にかかわらず、こうしたクライアントは、TCP、HTTP、または HTTPS 経由で要求を実行できます。各アプリケーションでは、任意で独自のセキュリティ メカニズム（暗号化など）を使用することもでき、自身の接続を攻撃から防御できます。

次に、サービス バスのその他の便利な機能を紹介します。

- シンプルなキューと同様に動作するメッセージ バッファのサポート。クライアントはサービスを直接呼び出す代わりに、最大 256 キロバイトのメッセージをメッセージ バッファに書き込むことができます。このメッセージはディスクに永続化される（消去されない）ため、後からサービスでこのメッセージを読み込むことができます。また、Windows Azure Platform での通常の処理として、永続化されたメッセージは障害対策のために複製されます。
- 複数の WCF サービスで同じ URI をリッスンする機能。クライアントからの要求は、サービス バスによって、リッスン中の全サービスへとランダムに分散されます。これには、WCF サービスの負荷分散とフォールトトレランスの両方を達成する目的があります。

外部環境へのアプリケーションの公開は、口で言うほど簡単ではありません。サービス バスの目的は、こうした対話機能をできるだけ簡単に実装できるようにすることです。

## アクセス コントロール

---

ほとんどの分散アプリケーションにとって、ID の処理は基本となる機能です。これに対処する最近のアプローチである "クレームベース ID" を使用すると、ユーザーは ID 情報を含む複数の "クレーム" を集約した "トークン" を送信できます。クレームによっては、ユーザー名が含まれている場合もあれば、ユーザーの年齢や所属するグループが含まれている場合もあります。アプリケーションは、トークンに含まれるこうしたクレームか他の方法を使用して、対象のユーザーに許可される操作を判断します。

クレームベースの環境では、"ID プロバイダー (IdP)" によってトークンが発行されます。Active Directory フェデレーション サービス (AD FS) 2.0 のような一部の IdP は自社内に存在します。その他の、Windows Live ID や Google アカウントなどには、インターネット経由でだれもがアクセスできます。どの IdP を信頼するか、すなわちどのトークンを受け入れるかは、アプリケーションごとに決定できます。たとえば、オンプレミスで実行されるアプリケーションではオンプレミスの AD FS サーバーによって発行されたトークンだけを受け入れ、インターネット上で実行されるアプリケーションでは、Google や Facebook などによって発行されたトークンを受け入れることも可能です。



ただし、IdP が違えば使用されるトークンの形式も違い、クレームの表記法も異なります。Google、Facebook、Yahoo などの ID を直接受け入れるアプリケーションは、こうした相違点にそれ自体で対応する必要があります。しかし、そこまでする必要はあるでしょうか。代わりに、共通のクレーム表記法による、単一のトークン形式を生成できる中間機能を作成してはどうでしょうか。こうすると、1 種類のトークンだけに対応すればよくなるため、アプリケーションの作成担当者の業務が簡略化される可能性があります。

アクセス コントロール サービスの役割はまさにこれであり、クラウド内でクレームベース ID を処理するための中間機能を提供します。図 11 に、このしくみを示します。

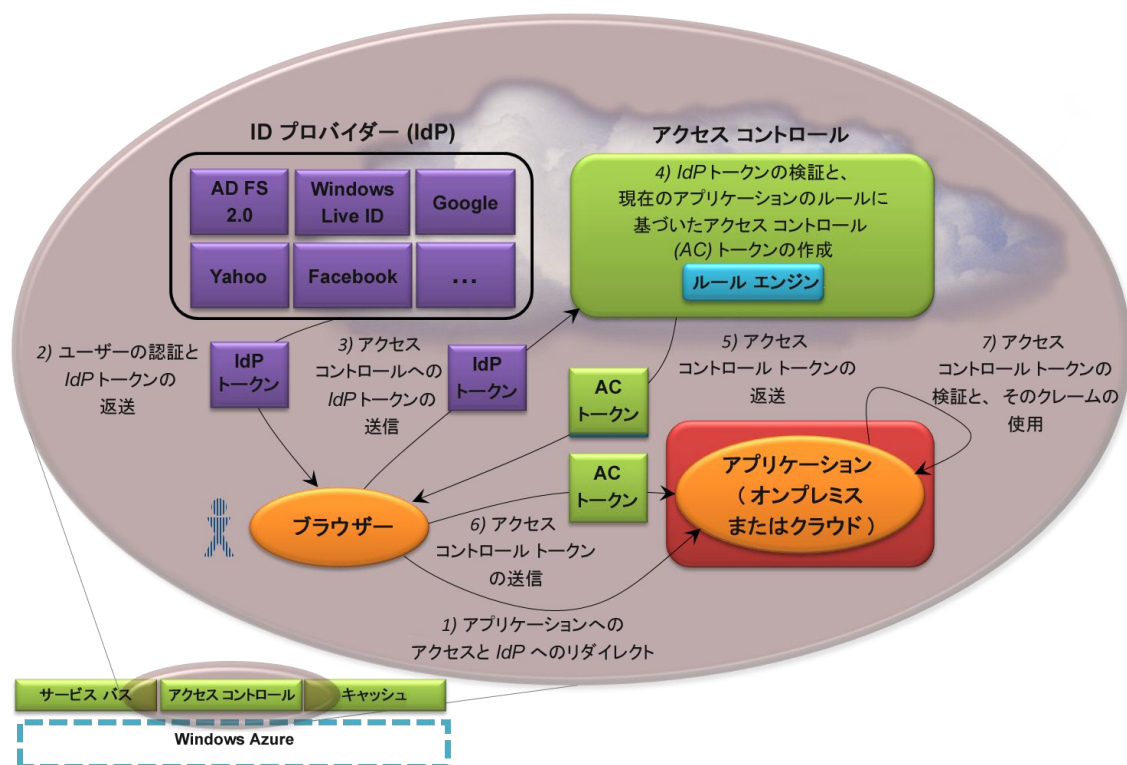


図 11: さまざまな ID プロバイダーによって発行された ID 情報の、アプリケーションによる受け入れを容易にするアクセスコントロール サービス

図で示すように、アクセス コントロールを利用するアプリケーションは、オンプレミスまたはクラウド内で実行できます。いずれの場合も、ユーザーがブラウザ経由でこのアプリケーションにアクセスしようとした時点でプロセスが開始されます (ステップ 1)。アプリケーションにより、このアプリケーションが受け入れるトークンの発行元である IdP にブラウザがリダイレクトされます。ユーザー名とパスワードを入力するなどして、ユーザーがこの IdP で認証されると、IdP がそのユーザーに関するクレームを含んだトークンを返します (ステップ 2)。

次に、ユーザーのブラウザから、その IdP のトークンがアクセス コントロールへと送信されます (ステップ 3)。アクセス コントロールはそのトークンを検証し、それが間違いなく対象の IdP によって発行されたものであることを確認した後、このアプリケーションに対して定義されている任意のルールに従って新しいトークンを作成します (ステップ 4)。アクセス コントロールにはルール エンジンが含まれており、さまざまな IdP から発行されたトークンをアクセス コントロール トークンへと変換するための方法を、各アプリケーションの管理者が定義できます。たとえば、さまざまな IdP がユーザー名の表記法に異なる形式を使用している場合、アクセス コントロール ルールによって、これらをすべて共通のユーザー名文字列に変換できます。続いて、アクセス コントロールはこの新しいトークンをブラウザに返し (ステップ 5)、ブラウザがそれをアプリケーションへと送信します (ステッ

ブ 6)。アプリケーションはこのアクセス コントロール トークンを受け取ると、それが間違いなくアクセス コントロールによって発行されたものであることを確認した後、トークンに含まれるクレームを適用します (ステップ 7)。

このプロセスは少し複雑に思えるかもしれませんが、実はアプリケーションの作成者の業務を大幅に効率化することができます。アプリケーションでは、さまざまなクレームを含む多種多様なトークンを扱う代わりに、馴染みのあるクレーム表記法を使用したトークンを 1 つだけ受信することにより、複数の ID プロバイダーから発行された ID を受け入れることができます。また、信頼するさまざまな IdP をアプリケーションごとに設定しなくても、こうした信頼関係がアクセス コントロールによって管理され、アプリケーションはそれを信頼すれば済みます。

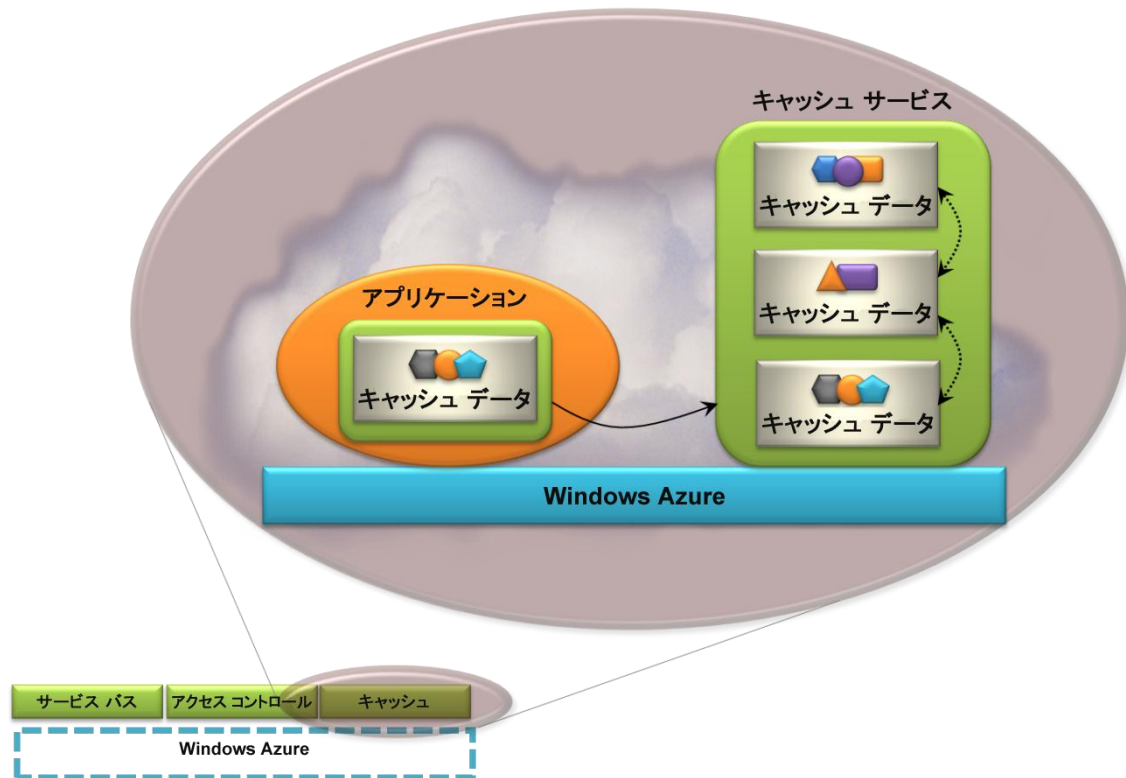
図 11 が示すように、アクセス コントロールには、AD FS 2.0、Windows Live ID、Google、Yahoo、および Facebook といった複数の ID プロバイダーのサポート機能が組み込まれています。また、アクセス コントロールは、OpenID をサポートする任意の IdP にも対応します。ブラウザーおよびその他のクライアントは、OAuth 2 または WS-Trust を使用してアクセス コントロール トークンを要求でき、これらのトークンには、SAML 1.1、SAML 2.0、および Simple Web Token (SWT) といったさまざまな形式を使用できます。また、Windows Identity Foundation (WIF) を使用すると、Windows 開発者はアクセス コントロール トークンを受け入れるアプリケーションを作成できます (ただし、アクセス コントロールは決して、Windows に関連付けられているわけではないことに留意してください。Google および Facebook の ID だけを受け入れる Linux アプリケーションでも同じように利用できます)。

ID の処理は、ほぼすべての分散アプリケーションにとって重要な機能です。アクセス コントロールの目的は、多様なプロバイダーによって発行された ID を受け入れる、セキュアなアプリケーションを開発者がより簡単に作成できるようにすることです。マイクロソフトはこのサービスをクラウドに実装することで、あらゆるプラットフォーム上で実行される任意のアプリケーションから、このサービスを利用できるようにしています。

## キャッシュ

---

多くのアプリケーションにとって、パフォーマンス向上に最も効果的な方法の 1 つは、頻繁にアクセスされるデータをキャッシングすることです。アプリケーションは同じ情報を繰り返し使用する傾向があるため、より短時間でこうした情報にアクセスできるようにしておけば、アプリケーションの速度を向上できます。これに対応するのが Windows Azure AppFabric キャッシュの目的です。図 12 に、この概念図を示します。



**図 12: Windows Azure アプリケーションの頻繁にアクセスされるデータへのアクセスを高速化する Windows Azure AppFabric キャッシュ**

このキャッシュ サービスは、Windows Azure アプリケーションの分散キャッシュ、およびキャッシュにアクセスするためのライブラリを提供します。図 12 が示すように、このサービスにはローカル キャッシュが含まれ、最近アクセスされたデータ アイテムのコピーをアプリケーションの各ロール インスタンス内で維持できます。アプリケーションが必要とするデータ アイテムがローカル キャッシュ内で見つからない場合、キャッシュ ライブラリは、キャッシュ サービスによって提供される共有キャッシュに自動的にアクセスします。図が示すように、このキャッシュは多数の Windows Azure インスタンスに分散されており、それぞれが別のキャッシュ データを保持しています。ただし、データが分散されていることは、キャッシュを使用するアプリケーションからはわかりません。アプリケーションはただデータ アイテムを要求すれば、キャッシュがそれを見つけ（キャッシュ内にある場合）、そのデータを含むいずれかのインスタンスから目的のデータが返されます。

ただし、最近アクセスされたデータは自動的にキャッシュされるわけではありません。代わりに、アプリケーションが、Caching API を使用するなどしてデータ アイテムを明示的にキャッシュ内に挿入します。また、Windows Azure 上で実行される ASP.NET アプリケーションで、Session オブジェクト データをキャッシュ サービス内に格納するように設定すれば、コードを一切変更することなく、短時間でキャッシュを保存できます。

キャッシュ サービスは、Windows Azure AppFabric のオンプレミス版である Windows Server AppFabric でも提供されます。実際、この 2 つはとてもよく似ています。最も大きな相違点は、オンプレミス版とは異なり、Windows Azure AppFabric キャッシュはサービスとして提供され、サーバーの構成やキャッシュの管理が必要ないことです。代わりに、こうした作業はすべて、サービス自体で自動的に対処されます。また、クラウド版のキャッシュ サービスはマルチテナントを前提としており、サービスを使用するアプリケーションごとに専用のインスタンスが割り当てられます。このインスタンスにアクセスするにはアプリケーションの認証が必要となるため、キャッシュ サービス内で保持されるデータに、他のアプリケーションからはアクセスできません。

キャッシュを使用すると、開発者はほとんど手間をかけることなくアプリケーションを高速化し、スケーラビリティを向上させることができるため、利用価値が高いといえます。Windows Azure AppFabric キャッシュではキャッシングをサービスとして提供することで、より簡単にこれを実現できます。

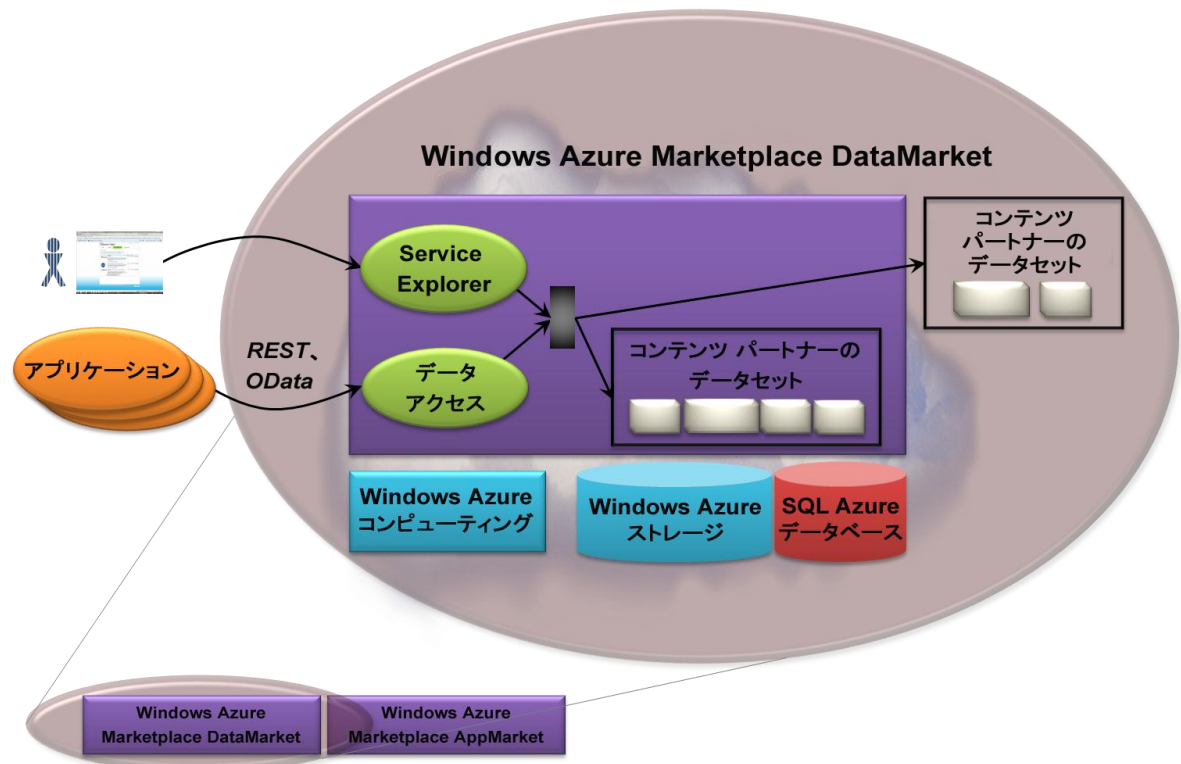
## WINDOWS AZURE MARKETPLACE

クラウド プラットフォームは便利ではありますが、目標を達成するための手段にすぎません。本当の目標は、価値あるアプリケーションおよびデータをユーザーに提供することです。Windows Azure プラットフォームはこれら両方の基盤を提供しているため、必要なアプリケーションおよびデータをユーザーが探すための手段も含んでいるのは当然のことです。

Windows Azure Marketplace は、このニーズを満たすために設計されており、AppMarket と DataMarket という 2 つのコンポーネントのそれぞれを通じ、ユーザーがアプリケーションまたはデータを検索、試用、および購入できます。これらはどちらも重要なコンポーネントですが、DataMarket が最初に一般提供されるため、以降では Marketplace のこの要素を中心に説明します。

現在、アプリケーションの購入は一般的なことであり、ほぼすべての組織が行っています。データの購入はそれほど一般的ではないものの、重要なことには変わりはありません。人口統計情報、財務情報、法務情報など、多くの会社によって、さまざまな種類のデータが数多く販売されています。ただし、購入したデータを使用するには通常、必要なデータがあるかどうかを見極め、そのデータを提供する会社を探してから、提供されるデータが自社のニーズを満たすかどうかを判断する必要があります。

これらすべてを行うのは、必要以上に困難な作業です。なぜ、あらゆる種類のコンテンツ プロバイダーが提供する、あらゆる種類のデータを 1 か所で検索できるようにならないのでしょうか。なぜ、データを検証し、自身のニーズを満たしていることを確認してから、そのときその場所で必要とするデータを購入できないのでしょうか。Windows Azure Marketplace の DataMarket コンポーネントは、これらの点を考慮して作成されています。図 13 に、Windows Azure Marketplace の主要コンポーネントを示します。



**図 13: Windows Azure 内、SQL Azure 内、または外部で保持する、コンテンツ パートナーのデータセットへのアクセスが、Windows Azure をベースとする DataMarket により実現**

図が示すように、ユーザーおよびアプリケーションの両方が DataMarket を通じて情報にアクセスできます。ユーザーは Service Explorer と呼ばれる Windows Azure ベースのアプリケーションを使用すると、どのようなデータセットが利用可能であるかを確認し、必要なものを購入できます。購入後は、各種アプリケーションから、REST ベースの要求または OData による要求を使用して対象のデータにアクセスできます。DataMarket を通じて販売されるデータセットは、Windows Azure ストレージまたは SQL Azure データベースを使用して、Windows Azure Platform 自体に保存できます。コンテンツ プロバイダーが所有するデータ センター内など、外部でデータセットを保持することも可能です。すべてをこのクラウド環境で維持する必要はありません。

購入者に対し、DataMarket はさまざまな商用データセットの検索、購入、およびアクセスが可能な場所を提供します。そうしたデータセットの所有者であるコンテンツ プロバイダーに対し、DataMarket はマイクロソフトのクラウド プラットフォームを通じて、より多くの顧客に自社の商品を公開する機会を提供します。データの価格は所有者であるコンテンツ プロバイダーが設定しますが、DataMarket が課金サービスを提供するため、プロバイダーが直接顧客とやり取りする必要はありません。また、マイクロソフトはコンテンツ プロバイダーの質を詳しく調査し、特定の業界で上位 5 社に入ったプロバイダーだけで DataMarket の提供を開始します。

各種アプリケーションでは、コンテンツ プロバイダーのライセンスで許可されている限り、購入したデータを任意の方法で自由に使用できます。たとえば、Microsoft Excel 2010 のユーザーは、DataMarket の情報に Excel アドインを使用して直接アクセスするほか、PowerPivot for Excel の OData のサポート機能を使用して DataMarket から情報にアクセスし、データを分析できます。DataMarket から購入したデータを自社のデータと統合し、SQL Server Reporting Services を使用して作成されたレポートに利用することも可能です。



情報テクノロジーからビジネス上の価値を実現するには、適切なアプリケーションが必要ですが、適切なデータも必要となります。DataMarket では市販のデータセットの検索、評価、および購入を容易にすることで、企業が必要とする情報をよりシンプルに検索および使用するための手段の提供を目指しています。

## 将来的な展望

マイクロソフトは、近いうちに Windows Azure Platform に多くの新機能を追加する計画であると発表しています。これには、次のような機能が含まれます。

- Windows Azure Platform Appliance: マイクロソフトは、ホスティング サービス会社および企業がオンプレミスのデータ センター内で Windows Azure を運用できるようにするため、Windows Azure Platform Appliance を提供する計画を発表しました。Windows Azure にはソフトウェアだけでなく、特定のハードウェア構成が必要となるため、このアプライアンスには Windows Azure の実行に必要なサーバー、ネットワーク接続、およびストレージがすべて含まれる予定です。また、このアプライアンスの当初のターゲットは大規模な組織のみになりますが、マイクロソフトは最終的に、小規模な組織でも Windows Azure Platform Appliance が利用できるようになることを目指しています。
- CDN ダイナミック コンテンツ キャッシュ: 現在、Windows Azure CDN で扱うことができるのは BLOB データだけです。新たに追加されるこの機能を使用すると、Windows Azure アプリケーションによって動的に作成されたコンテンツも CDN でキャッシュできるようになります。
- VM ロール スナップショット: Windows Azure VM ロールの最初のリリースでは、実行中に行われた OS ボリュームへの変更を一切保存できません。スナップショットによってこれが改善され、OS ボリュームの状態を永続的ストレージへと定期的に保存できるようになります。
- Java のサポートの向上: Windows Azure では現在も Java アプリケーションを実行できますが、マイクロソフトはさらなる機能向上を計画しています。予定されている改良点には、Java のパフォーマンス向上、Eclipse ベースのツールに対するサポート強化、および Windows Azure 用 Java ライブラリの拡充などがあります。
- コンポジット アプリケーションのサポート: 既存のコンポーネントおよびサービスを組み合わせることで Windows Azure アプリケーションをより簡単に構築できるようにするため、Windows Azure AppFabric に複合モデルを取り入れます。各種のランタイム サービスおよび Visual Studio ベースのデザイナーが付属したこのモデルを使用すると、開発者はアプリケーションのさまざまな部分を、そのライフサイクルを通じて単一の論理エンティティとして扱うことができます。

これらの変更点はすべて、より広範なシナリオで Windows Azure Platform を利用できるようにするという同じ目的に基づいています。

## まとめ

クラウド コンピューティングの時代が到来したのは明白な事実です。開発者にとって、クラウドの活用とは、クラウド プラットフォームを使用することを意味しています。Windows Azure Platform により、マイクロソフトはさまざまなニーズに対応する広範なオプションを提供します。

- Windows Azure は、クラウド内でコンピューティングとストレージの環境を提供します。
- SQL Azure はクラウド内で、リレーショナル DBMS に加えてレポート サービスとデータ同期サービスを提供します。

- Windows Azure AppFabric は、クラウド アプリケーションとオンプレミスのアプリケーションの両方をサポートする、クラウドベースのインフラストラクチャを提供します。
- Windows Azure Marketplace は、コンテンツ プロバイダーが提供するデータセットを検索および購入するためのオンライン ストアです。今後はクラウド アプリケーションの販売も開始される予定です。

これらのアプローチはさまざまな要件に対応しますが、すべての開発者がすべての機能を使用するとは限りません。ただし、皆さんが ISV、システム インテグレーター、または企業のいずれで働いているかにかかわらず、所属する組織で作成するアプリケーションにとって、一定のクラウド プラットフォーム サービスの利用が役に立つ可能性があります。新たな時代の幕開けに備え、時流に乗り遅れないように準備を整えましょう。

## 執筆者について

David Chappell は、カリフォルニア州サンフランシスコに拠点を置く Chappell & Associates ([www.davidchappell.com](http://www.davidchappell.com)) の代表を務めており、講演、執筆、コンサルティングを通じて、世界中の人々が最新テクノロジーを理解して使用し、適切な意思決定を行えるよう支援しています。