This article describes how to use Windows Azure together with on-premises resources.

The Benefits of Hybrid

The cloud offers many compelling benefits to businesses and government agencies. These benefits include:

- Cost savings by not requiring on-premises compute and data storage resources.
- Reliability through hardware failure detection and automated repair.
- Insurance against data loss through seamless and transparent data replication.
- Massive up-and-down scalability of data and computational services on-demand.
- Federated single-sign-on across business partner boundaries.

When people first contemplate moving their applications into the cloud, they naturally think about it in "all-or-nothing" terms. It turns out that many applications lend themselves better to a "hybrid" approach, where parts of the application live in the cloud, and other parts live on-premises.

Windows Azure is comprised of a suite of cloud service technologies that are sold a la carte on a pay-as-you-go basis. These services can all work together to support applications that reside completely in the cloud, or they can be used independently. This model provides organizations with the flexibility to select and combine cloud storage and services with option to offload some of your on-premises storage and services workloads to the cloud.

The hybrid approach to application architecture can be utilized to meet many of today's IT challenges. It is not uncommon in today's application landscape for resources used by an application to be stored in multiple locations and deployed on multiple platforms. Some resources may be stored in a corporation's own data center, while other resources may be located in the cloud or in the data centers of your business partners.

There are many reasons for requiring a hybrid approach to cloud applications:

- Ownership of proprietary data or protection of proprietary algorithms can lead a business to store its data in its own on-premises facilities.
- Contractual considerations between business partners may also create constraints.
- Certain types of data and government agencies come under regulations which explicitly forbid data from being stored in another country.
- Some data may be sensitive or subject to privacy concerns, even if it is not government regulated.

Businesses may have legacy applications which are in a “maintenance-only” phase of their life-cycle and cannot be easily migrated to the cloud.

Data may need to be shared locally with on --premise applications and services.

With a hybrid application solution that uses both on-premises and cloud-based resources, application code and data can be stored in an appropriate on-premises location according to regulations, privacy concerns, and a measurement of acceptable risk, while solution components requiring the features and pricing model of cloud computing can be migrated to the cloud. This paper will explore several Windows Azure services that work well in the creation of hybrid applications.

Service Bus Relay

The Service Bus in Windows Azure, pictured in Figure 1, provides secure messaging and connectivity capabilities to facilitate the development of building highly-scalable, loosely-coupled applications in the cloud or in hybrid configurations between on-premises and cloud applications and data sources. Service Bus Relay supports an extensive assortment of communications and messaging protocols and patterns thus enabling a wide variety of business application connectivity scenarios without the usual level of friction associated with setting up security and on-premises endpoints.

Service Bus Relay is useful when you require a low-latency synchronous connection or bi-directional message exchange. For example, a cloud-based website that needs to pull data from an on-premise CRM system and show it to the user immediately might use the Service Bus Relay to accomplish this objective. The Service Bus Relay provides direct and low-latency service access.

Windows Azure Connect is another technology which could be used to accomplish the objective of exposing an on-premises service endpoint to the cloud-deployed web application. The primary differentiator between these two technologies is that the Service Bus Relay is a messaging technology operating at the application layer and exposing specific application endpoints, while Windows Azure Connect is a virtualized network service operating at the network layer connecting machines and VMs into a single trust domain with shared access to all endpoints and services running in that domain. Thus, the choice is along the lines of whether applications want to integrate individual endpoints as it is common with service-oriented architectures, or whether workloads need broader network-level integration that’s not easily scoped to application endpoints.

Modern business networks and security policies are complicated. Businesses routinely setup special “DMZ” network zones to shield their intranets from the wilds of the Internet. Networks are often configured with non-routable and randomly assigned IP addresses which sit behind port-mapping firewalls using Network Address Translation. To make matters even more complicated, applications are often configured to operate on non-standard, arbitrarily assigned IP ports, and certain kinds of message traffic may be further restricted by security policies.

The day-to-day maintenance of identities between business partners is also very complicated and subject to oversights and procedural errors. These mistakes might subject a business to many negative consequences and potential legal penalties when unauthorized access or malicious operations are undertaken by unauthorized personnel. The procedures for securely onboarding another business

partner, service, or application can also be very daunting. Configuring the security for a new application can take longer to implement than it did to write.

Service Bus Relay cuts through many of the physical and political difficulties of connecting enterprise applications across departmental and corporate boundaries, and it does so in a rather elegant way. When applications are registered for use with Service Bus Relay, the administrator selects a unique endpoint address in the Windows Azure Access Control namespace by prefixing the accesscontrol.windows.net suffix with an exclusive (e.g. https://*contoso*.accesscontrol.windows.net/). Rather than waiting for incoming traffic to arrive at the corporate firewall or DMZ for the application, the Service Bus Relay model allows the application to open up an outbound communication channel with the Service Bus Relay service in Windows Azure as depicted in Figure 1. The application then keeps its connection with the cloud open in perpetuity (or until the application is shut down or recycled) by sending tiny outbound "ping" heartbeat message every few seconds. Service Bus Relay then relays any traffic sent to endpoint address in the cloud to the on-premises service application where the messages are processed.

Figure 1-Windows Azure Service Bus Relay

Coding this orchestration by hand would undoubtedly be challenging, but fortunately the complexity is encapsulated in a set of convenient and easy to use Windows Communication Foundation (WCF) bindings (e.g. BasicHttpRelayBinding, WebHttpRelayBinding, WS2007HttpRelayBinding, NetTcpRelayBinding, NetOneWayRelayBinding, and NetEventRelayBinding), which do all of the "heaving lifting" of the outbound heartbeats and the instancing of service objects (including the deserialization of the message and its dispatch to the appropriate service operation).

Service Bus Relay allows up to 25 simultaneous listeners to be registered to a single service endpoint. In this configuration, Service Bus Relay randomly selects a listener to receive an incoming HTTP request or TCP session, which is generally adequate enough to ensure reasonable load balancing of messages across all listeners. Note: There isn't an SLA guarantee of equal distribution of incoming message; it is stated only as a "best effort." For more information on the Service Bus load balancing feature, see Windows Azure Service Bus Relay Load balancing (Scale & Availability Feature).

Service Bus Queues and Topics

The previous section outlined the benefit of using Service Bus Relay to solve application scenarios requiring immediate low-latency direct connectivity to services. Of course there are applications where direct connectivity to the service is not required. Many applications require independence from the service's availability, or they may need guaranteed message delivery. Service Bus Queues and Topics provide these capabilities.

For example, an application posting work items to a business partner's services would benefit by adding queues to the architecture. Service Bus Queues and Topics allow the sending application to continue unabated when their business partner's services are down for maintenance or experiencing outages, and for processing of those messages to resume when the business partner's services are back online.

Service Bus Queues and Topics support a brokered messaging communication model. When using these brokered messaging entities, components of a distributed application do not communicate directly with each other, they instead exchange messages via a message broker, which acts as an intermediary.

A Queue is an entity managed by the broker that manages a sequence of messages with a single retrieval point - the Queue's "head"; if there are multiple receivers only one of them will receive any particular message. A Topic is another entity that is much like a queue, but it can have multiple such retrieval points, called Subscriptions. Each Subscription gets to see a copy of each added message and a filter can determine whether the message is eligible for retrieval through the Subscription. If it is, receivers can receive the message just like from the Queue. Thus, Topics are a mechanism very similar to Queues and differ primarily in allowing distribution of message copies to different Subscriptions – and with that key capability they are often a more flexible choice compared to Queues.

A message producer (sender) hands off a message to the Queue (or Topic) and then continues its processing. Asynchronously, a message consumer (receiver) pulls the message from the Queue (or Subscription) and processes it. The producer does not have to wait for a reply from the consumer in order to continue to process and send further messages. Queues and Topics with their Subscriptions offer First In, First Out (FIFO) message delivery to one or more competing consumers. That is, messages are typically received and processed by the receivers in the order in which they were added to the queue, and each message is received and processed by only one message consumer.

Service Bus Queues and Topics may be used to achieve some of the following business objectives:

Temporal decoupling allows message senders and receivers to work on independent schedules. The producers and consumers are no longer required to be online at the same time. That means the service can be taken down without impacting service consumers as work will simply be stored in the queue until the service comes back online.

Load leveling allows work to be spread over time generally saving money because the message consumer only needs to be capable of handling an average message load instead of the peak message load.

Figure 2-Load Leveling

Load balancing allows additional message consumers to be added during peak message volume to keep throughput high but without breaking the bank by having idle capacity during non-peak loads.

Figure 3-Load Balancing

Loose coupling allows message producers and message consumers to operate with complete independence of each another, which is consistent with one of the primary tenants of Service-Oriented Architecture (SOA) that states that services share schema and contract, not class. Only adherence to the message schema by the service and its consumer is important.

SQL Data Sync

The SQL Data Sync service can be used to keep on-premises SQL Server database synchronized with one or more SQL Database instances in the cloud. There are a myriad of application scenarios where this capability proves itself useful.

For example, this service can be used to keep two or more SQL Database instances synchronized when you want to improve performance by geographically collocating application data with your customers. You can use the SQL Data Sync service to accomplish this objective. Synchronization can also be useful when you want to assure that your data is being redundantly stored in different geographical locations in the unlikely event of a cataclysmic disaster striking a regional data center. In hybrid application scenarios, data may need to be used by on-premises applications as well as cloud applications, or you may simply want to replicate data based on proximity in order to improve performance.

Unidirectional and bidirectional synchronization are supported, giving you the flexibility of allowing changes only to a replication master or having changes to the data flow in both directions. Individual tables and even columns to be synchronized can be specified and filters can be established to apply synchronization only to a subset of rows. Conflict resolution for two-way synchronizations and synchronization frequency are also tunable.

Figure 4-SQL Data Sync

For more information on SQL Data Sync, see:

SQL Database Data (MSDN Library)

Access Control Service (ACS) paired with Active Directory Federation Services (AD FS)

Pairing the Windows Azure Access Control Service (ACS) in the cloud with on-premises Active Directory Federation Services (AD FS) makes it easy to extend corporate identities, such as those in your own or partners' Active Directory services, beyond the confines of the firewall. Extending Active Directory beyond the enterprise firewall provides secure identities that can be used to make authentication and authorization decisions for Internet applications, as well as to give single-sign-on even between business partners across corporate boundaries.

Windows Azure Access Control Service (ACS)

ACS brings federated identity and claims-based security to your applications and services. Claims-based security allows developers to isolate themselves from the complexities of security code by decoupling application logic from security decisions. The application is allowed to make assertions on claims which must be present for a given operation to be executed. The validation of an identity and the issuance of claims by a trusted claims issuer are handled by separate services and processes, rather than by the application. Another way of looking at ACS is as a single-sign-on security service in the cloud. Single-sign-on in the cloud is often referred to as "Federated Identity" because identities can be used in a universal manner across business domain boundaries, whether or not the applications are located on-premises, deployed to the cloud, or a hybrid of both.

ACS operates by taking authenticated identities and transforming them into one of two kinds of specific security tokens that are used by your application to make authorization decisions. The specific choice of token format is not important to this article, but both Security Access Markup Language (SAML) and Simple Web Token (SWT) formats are supported by ACS. You can find additional links to resources that will further explain token formats referenced below.

The authenticated identities come from identity providers, and generally speaking the identity provider is responsible for verifying the authenticity of an identity. The most popular form of identity verification in today's world is a simple username and password, but more sophisticated verifications are possible too, such as the use of digital certificates and biometric devices.

Identities are selected according to your application's requirements for trustworthiness. Self-verified identities have a relatively low level of trust, and are generally suitable for social websites, blogs, and many publicly accessible commercial applications where a person does not have to prove who they are, but can simply make their own unverified statements about their identity (called "claims" in security lingo). Identities verified by a third party are generally more suitable for enterprise and business-to-business applications and services.

ACS supports self-verified identities issued by Windows Live ID, Google, Yahoo!, and Facebook. When used in conjunction with a credit card and an email address, the level of trust of self-verified identities can be enhanced dramatically, but they are generally inadequate for most business applications that must restrict application access to identities that are tightly controlled. In the corporate environment, Active Directory domain controller provides identity services related to an organization. Identities in Active Directory Domain Services (AD FS) carry a higher degree of trust, given that an employee typically has to show government-issued proof of identity before an organization will hire them, and corporate IT professionals administrate these identities in accordance with strict and verifiable corporate policies.

Active Directory Federation Services (AD FS)

Using Active Directory Federation Services v2.0 (AD FS), corporate identities stored in AD FS can be put into use beyond the organization's firewall for use in securing Internet applications. AD FS adheres to the WS-Federation specification, an industry standard with widespread adoption, which was jointly developed by BEA Systems, BMC Software, CA Inc., IBM, Layer 7 Technologies, Microsoft, Novell, Ping Identity, and VeriSign for brokering trust of identities between participating applications and services across heterogeneous machines and platforms. A custom Security Token Services (STS) written using Microsoft Windows Identity Foundation or an AD FS installation are two examples of WS-Federation compliant identity sources.

ACS Paired with AD FS

Figure 5-ACS Paired With AD FS

Many organizations use AD FS for managing their corporate identities. AD FS v2.0 is a service that ships with Windows Server 2008 R2 and can be used by itself or in conjunction with ACS to extend corporate identities beyond the boundaries of the enterprise's Intranet. This offers organizations identities that

can be seamlessly used in the cloud and on-premises to make application authorization decisions, and can cross the boundaries of corporate firewalls without any additional maintenance burdens. AD FS may be used directly without ACS to support extension of your enterprise identities to the cloud, so the next question that most people logically ask is “why use them together?” To answer this question requires a little more information about the features offered by ACS.

Each of the aforementioned identity sources has its own API, and there isn’t a lot of overlap between them. Security is a complex topic to begin with and learning all of the different API’s can be a daunting task for an application developer. By learning only the API of the ACS, the application developer can use any of these identity providers individually or in concert with one other without the need to learn many disparate identity technologies.

ACS also has its own built-in identity store, which can be very useful for solving a variety of applications. Use of ACS allows your applications to trust security tokens that it issues, thereby loosely coupling your choices of identity providers from your application and allowing them to change without changing your application. In addition, the ACS places another layer of security between your enterprise and those who might cause it harm.

AD FS emits identities as Security Access Markup Language (SAML) tokens, so you could configure your applications to utilize the AD FS directly. However, ACS provides two valuable advantages over using ADFS directly:

The first advantage is that it allows your AD FS instance to only trust ACS queries, thereby providing a level of isolation from direct attacks.

Secondly it provides loose coupling of the identity stores, allowing new identity stores to be added or changed in the future without impacting your applications.

In federated security parlance, your application is referred to as a relying party, because it relies upon ACS for its security. In order for that security to be trustworthy, ACS must know information about the relying party application so that it can issue security tokens only for use by that application. Likewise, your application needs to know that it can trust the claims that are being made inside of the security token, and it does this by inspecting the signature on the token with its own list of trusted identity providers.

For a deeper and more thorough understanding of identities and claims-based security systems, see:

Access Control Service (MSDN library)

Getting Started with Security and Claims-Based Identity Model

Web Service Federation Language

WS-Federation Identity Providers

Windows Server 2008 R2 Active Directory Overview

Windows Azure Connect

The basic theme of this article on hybrid Windows Azure solutions is that it is impractical for organizations to store all of their resources and run all of their applications in the cloud. There are many everyday business reasons why this statement holds true, and why it will likely always be true for at least the foreseeable future. It would seem that concerns over data privacy, ownership, governmental regulation, legal controls, and basic paranoia about where data resides and who has control are going to be present long into the foreseeable future.

With Windows Azure Connect, a worker or web role in the cloud can access on-premises resources, such as a SQL Server database or an on premise web service, directly because Windows Azure Connect provides the illusion that the on-premises and cloud networks are actually one by routing the traffic through an IPsec tunnel between the two networks. In fact, in such a configuration, the application's connection string doesn't even have to change as the on-premises SQL Server just looks to be part of the same network as far as the Windows Azure cloud instances are concerned.

To accomplish these objectives, Windows Azure Connect creates a secured communications tunnel between your on-premises applications and resources and your cloud-based applications and resources. This allows your Windows Azure role instances to become a logical extension of your corporate network. The instances in the cloud become additional machines on your corporate network as shown in Figure 6.

Figure 6-Windows Azure Connect

Another popular scenario is business network augmentation. Some businesses may require additional computation or storage capacities during peak demand than they have in their data centers. The resources in the cloud can be configured to become an extension of your on-premises computing resources thereby allowing a very rapid expansion of resources during business peaks and a very quick downsizing when that need has passed. When using the Windows Azure Connect feature this way, the role instances in the Windows Azure cloud can even be joined to your domain thus allowing security to be regulated by your existing Active Directory Domain Controller, and for users of the network to access these machines as they do any other machines within their corporate networks.

With so many choices available, it can be difficult to know which solutions represent "best practices." The following approaches might help;

Consider using Service Bus Queues (discussed earlier in this article) when interoperating with resources that are accessible only through the Internet.

Consider using Windows Azure Connect for accessing resources and services that are an extension of your corporate intranet, or when no external access is required.

For more information on Windows Azure Connect, see the following resources:

Overview of Windows Azure Connect (MSDN Library)

Windows Azure Connect Team Blog

Windows Azure Connect Team Blog - Domain Joining Windows Azure roles

Guide to Windows Azure Connect

Using Windows Azure Connect to Integrate On-Premises Web Services

Conclusion

It is impractical for organizations to store and run all of their resources and applications in the cloud. There are many practical business reasons why this statement is true, and why it will likely always be true. Concerns over data privacy, ownership, governmental regulation, legal controls, and basic paranoia about where data resides and who has control are always going to be present, resulting in many cloud applications being deployed in “hybrid” configurations, where portions of the application or its data remains on-premises.

Windows Azure Service Bus Relay and Queue services can be used in a heterogeneous computing environments consisting of both cloud and on-premises computing and storage assets to provide extraordinary scalability and security to your applications. Service Bus Queues additionally provide your existing applications with load leveling and balancing and offer resilience to failures and downtime for maintenance and software upgrades.

SQL Data Sync in the cloud can be used to synchronize your on-premises and cloud databases, enabling on-premises and cloud applications to share data, or allowing data to be geo-located via replication to better service customers.

Pairing the ACS in the cloud with your on-premises AD FS or AD FS can extend your corporate security beyond the confines of the firewall where they can be used to make authentication and authorization decisions for Internet applications (which have traditionally used Form-based authentication). Extending Active Directory beyond the enterprise firewall also provides secure identities that can cross corporate boundaries giving single sign-on even between business partners.

Windows Azure Connect dynamically extends your on-premises network with your cloud network providing full and flexible access to the resources for use by your applications with minimal or no code changes.

Because of the sophistication of modern applications with distributed services, heterogeneous platforms and operating systems, and security, talking “cloud computing” often involves hybrid solutions. The ability to augment on-premises network applications with cloud resources to deliver new and innovative business solutions is a core value of Windows Azure.