# Data access strategy in Azure hybrid environment

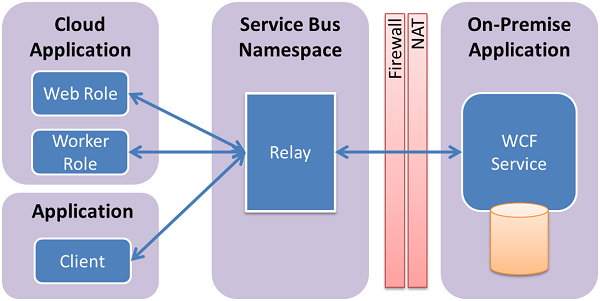
Azure cloud has provided lots of exciting IaaS, PaaS and SaaS and more are coming. But realistically, a company can’t move all data to cloud because of legislation.

Azure provides several services for different on-premises data access business cases. Let’s take a look at these scenarios and corresponding azure services.

Business Case 1:

Your company has hosted several WCF Services behind your company’s firewall for resource access. You want to expose this WCF service to applications running in Azure or extranet partner’s applications but don’t want to change incoming firewall rules and current corporate network infrastructure.

Azure Service Bus Relay could be a candidate.



(From [MSDN](https://azure.microsoft.com/en-us/documentation/articles/service-bus-dotnet-how-to-use-relay/) )

Using Service Bus Relay involves these steps:

1. Creating a Service Bus namespace for the Relay service. There are two types of namespace: Messaging or notification hub. For service Bus Relay messaging, messaging type is the right one.
2. Creating shared access polices for management operations. When the Service Bus namespace is provisioned, a default shared access rule with name “SharedAccessAuthorizationRule” is created with KeyName set toRootManageSharedAccessKey. In production environment, you should create separate keys for the sender and receiver. You may create multiple keys according to different groups of senders and receivers. It gives you more granularly control on the access.
3. Exposing WCF SOAP web service for external consumption. We must change the WCF service bindings and addresses. Here is the sample of App.config of WCF service. It has two endpoints. One for local and the other one is for Service Bus Replay. It uses hetTcpRealyBinding transport protocol although Service Bus Relay supports various protocols and web services standards. Here is the list of bindings:
   1. basicHttpRelayBinding
   2. WS2007HttpRelayBinding
   3. WebHttpRelayBinding
   4. NetTcpRelayBinding
   5. NetOneWayRelayBinding
   6. NetEventRelayBinding

<services>

<service name="BGISService.VendorAccess">

<endpoint contract="BGISService.IVendorAccess"

binding="netTcpBinding"

address="net.tcp://localhost:9358/vendor"/>

<endpoint contract="BGISService.IVendorAccess"

binding="netTcpRelayBinding"

address="sb://BGISNS.servicebus.windows.net/vendor"

behaviorConfiguration="sbTokenProvider"/>

</service>

</services>

<behaviors>

<endpointBehaviors>

<behavior name="sbTokenProvider">

<transportClientEndpointBehavior>

<tokenProvider>

<sharedAccessSignature keyName="RootManageSharedAccessKey" key="XXXXXXXX" />

</tokenProvider>

</transportClientEndpointBehavior>

</behavior>

</endpointBehaviors>

</behaviors>

1. Consume the service from the Client. We also just need to config the App.config file.

<client>

<endpoint name="Vendor" contract="BGISService.IVendorAccess"

binding="netTcpRelayBinding"

address="sb:// BGISNS.servicebus.windows.net/Vendor"

behaviorConfiguration="sbTokenProvider"/>

</client>

<behaviors>

<endpointBehaviors>

<behavior name="sbTokenProvider">

<transportClientEndpointBehavior>

<tokenProvider>

<sharedAccessSignature keyName="RootManageSharedAccessKey" key="xxxxxx" />

</tokenProvider>

</transportClientEndpointBehavior>

</behavior>

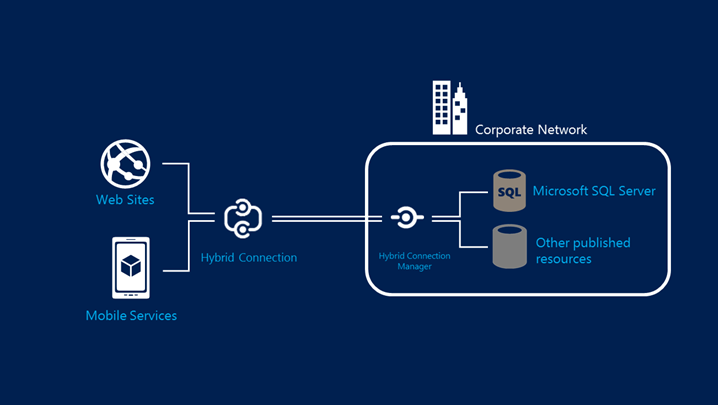
</endpointBehaviors>

</behaviors>

Business Case 2:

You want to move some of your on-premises .net web or mobile applications to Azure but you don’t want to make any code changes. You want these Web Apps and mobile Apps access on-premises resources in exactly the same way as if the Web or Mobile App is located on your local network. You don’t mind allow some outbound TCP connectivity from your private network. You also don’t mind installing some software on one or more windows 2008 R2+ servers. All these on-premises resources are using static TCP ports.

In this case, Azure App Service BizTalk API Apps Hybrid Connections provides all features you want.



(From [MSDN](https://azure.microsoft.com/en-us/documentation/articles/integration-hybrid-connection-overview/))

To use the BizTalk API Apps Hybrid Connections, follow these steps:

1. Create a new Hybrid Connection in BizTalk API Apps.

Basically, you need to create one Hybrid connection for one resource. For example, your application needs two on-premises resources, a SQL Server and a Web service. Then you need to create two connections. One for SQL server and the other one for Web Service.

Although multiple connections can be created, there are connection limitations by edition. [Here is the Feature charts](https://azure.microsoft.com/en-us/documentation/articles/biztalk-editions-feature-chart/)

1. Link your Azure Web Apps or Azure Mobile Apps
2. Install the Hybrid Connection Manager on-premises

The download could be found in Azure Management Portal->BizTalk Services->Hybrid Connections->On-Premises Setup. You don’t need to install the manager on the resource server. It could be any server and you could install multiple instances on multiple servers for scale out. But you have to configure the on-premises listener to use the same address as the first on-premises listener.

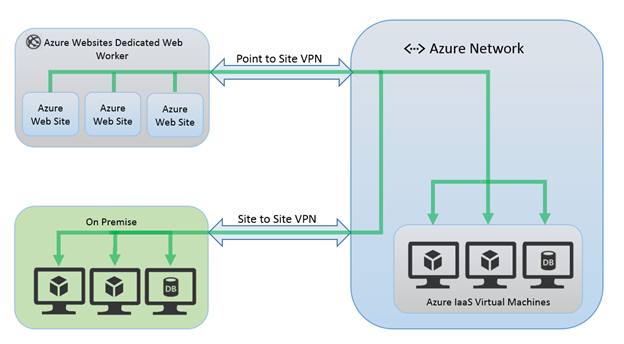
Hybrid Connections support the following framework and application combinations:

1. .NET framework access to SQL Server
2. .NET framework access to HTTP/HTTPS services with WebClient
3. PHP access to SQL Server, MySQL
4. Java access to SQL Server, MySQL and Oracle
5. Java access to HTTP/HTTPS services

Business Case 3:

You have moved your web site/application to Azure as a Web, mobile or API App but you want that App access resources like web services or databases running on your Azure Virtual Machines. And you may also want that Web App can access these types of resources in your own on-premises datacenter. Basically you want the web app can access everything or connect to it. You have App Server plan in Standard+ tier.

Azure virtual network integration preview feature is the answer.



(From [MSDN](https://azure.microsoft.com/en-us/documentation/articles/web-sites-integrate-with-vnet/))

In the Web App portal, you can connect to an existing virtual network or create a new one to connect to. The Point to Site Virtual network must be enabled with a Dynamic routing gateway. You can configure up to 5 VPN connections in the web service plan but only one can be connected to the Web App at a time.

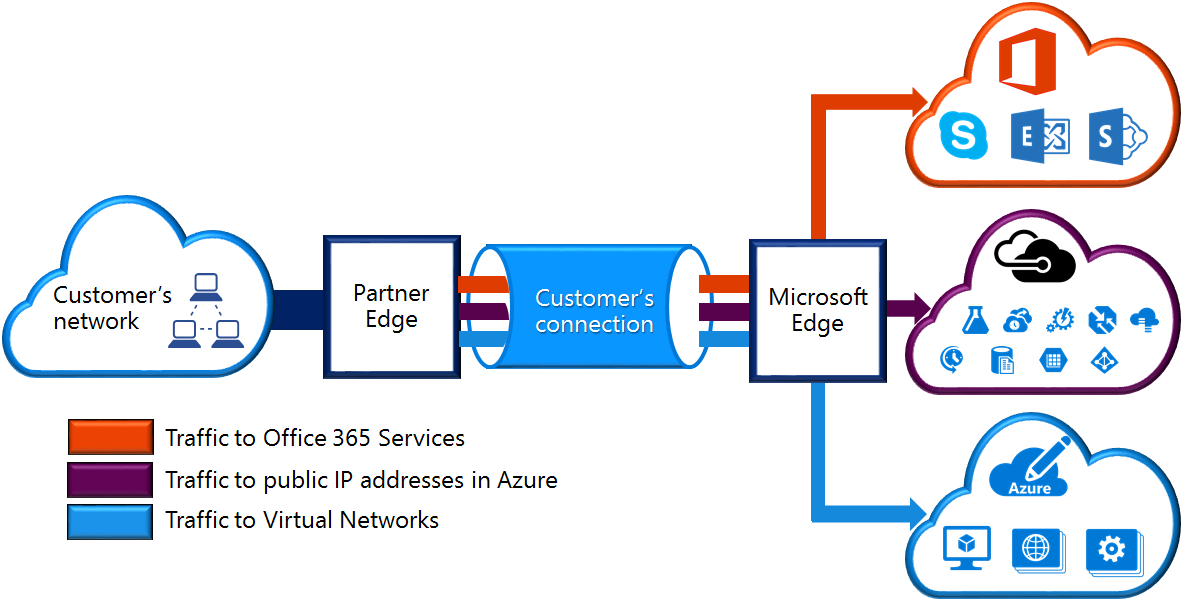
The web app accesses resources in the virtual network over TCP or UDP. In order to access resources in your on-premises system, you need to add routes on your own corporate network to allow traffic to go from your network to the Point-to-Site addresses configured in your virtual network after Site to Site VPN is configured.

Business Case 4:

You want a high secure, faster speed, lower latencies and more reliability network connections than typical connections over internet among azure and your on-premises data centers.

ExpressRoute is the easiest way to connect your on-premises resources to Azure resources together. The expressRoute connections can access Azure services, office 365 services and Microsoft Online services. It provide Layer 3 connectivity and consists of two connections to two Microsoft Enterprise edge routers from the connectivity provider. You can dynamic scale the bandwidth without tearing down connections. The ecosystem of connectivity providers is growing to provide expressRoute service in more locations.

But you pay what you get. It’s the solution with the least limitation but also the most expensive one.



(From [MSDN](https://azure.microsoft.com/en-us/documentation/articles/expressroute-introduction/))