

Blockchain beyond cryptocurrency

A white paper on Azure Blockchain Workbench

By Microsoft MVP - Steve Buchanan (www.buchatech.com)

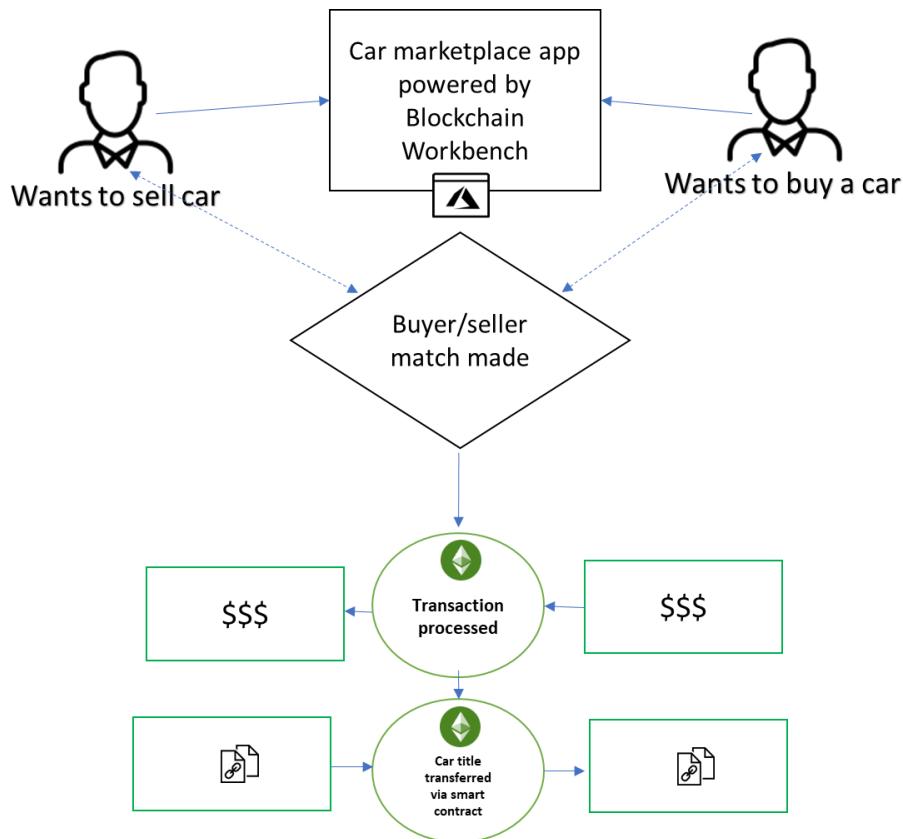
Contents

Blockchain beyond cryptocurrency.....	2
Overview of Ethereum and smart contracts.....	3
When and what to use blockchain for?	4
Blockchain Workbench architecture.....	5
Deploying Azure Blockchain Workbench.....	10
Prerequisites	10
Configuring Blockchain Workbench administrators	15
Deploy Blockchain Workbench	16
Deploy a blockchain application	19
Enterprise ready.....	23

Blockchain beyond cryptocurrency

Imagine that you want to sell a car. The buyer comes to take a test drive, likes the car, and pays for the car electronically. After that transaction is processed and the money is in your account, the title of ownership is automatically transferred from you to the buyer and registered with your local Department of Motor Vehicles (DMV). There's no hassle of going to the DMV and the fees are greatly reduced, costing only a few dollars at most.

The scenario just described is an idea for a future blockchain application to get you started thinking about the possibilities available with Blockchain. Here is a breakdown of what this would look like and how the smart contract would work.



This is the future of transactions of all kinds—a future that is not far off. In fact, the technology to power these transactions has arrived! It is blockchain smart contracts.

When people think of blockchain or you hear about it in the news, most of the time the focus is on cryptocurrency such as Bitcoin, Ethereum, Litecoin, and Ripple. But cryptocurrency is just one-use case of blockchain technology, and though it is exciting and revolutionary, blockchain smart contracts promise to revolutionize industries and society even more. Smart contracts are self-executing digital

agreements directly between two or more parties. The terms of a smart contract are contained within code and exist on a blockchain network. Smart contracts do not require a third-party entity to enforce them as they are automatically enforced per the terms of the contract being met. Some of the many ways that smart contracts could impact our daily lives are replacing title companies when buying and selling homes, becoming the platform for electronic medical records (EMR) transactions, powering marketplaces where you buy and sell items, authenticating high-end fashion items (e.g., Gucci, Coach purses), verifying goods along supply chains, and securing music and movies.

In my opinion, this is as exciting as living in the early days of the internet, when no one could have imagined how intertwined the “innerwebs” would become in our daily lives. We are in the early days of blockchain and one can only imagine what it will become—and what technologists will do with it.

Overview of Ethereum and smart contracts

Ethereum is an open-source blockchain distributed platform. It runs smart contract applications that protect against censorship, downtime, and third-party fraud. Blockchain technologies have different protocols for coming to a consensus on transactions, including Proof of Stake (PoS), Proof of Work (PoW), and Proof of Authority (PoA). The popular Bitcoin is based on Proof of Stake. With PoS, you take blockchain transactions and distribute them across multiple computers or nodes on the blockchain network. This distribution of the transactions is the distributed ledger. The nodes in a PoS act as the administrative body of the blockchain. These nodes are there to verify the legitimacy of each transaction in the blockchain ledger.

Ethereum is commonly based on Proof of Work. In the future, however, Ethereum may be switching to Proof of Stake. With PoW, miners compete against each other for getting and completing transactions on the blockchain network. PoW and PoS are similar in the way they work, but there are some key differences. Both protocols need to come to a consensus on the transactions in the blockchain and utilize miners to process/validate the transactions.

In PoW, miners validate transactions by solving mathematical problems on the distributed network and getting rewarded. The miners show the complete proof of how they arrived at the answer, that is, the steps taken to complete the complex algorithm of the transaction. This ensures the accuracy of the transaction. PoW requires a great deal of compute power because it must show the steps taken to solve the mathematical puzzle. With PoW, transactions are bundled into blocks. These blocks are verified by miners to validate if the transactions within each block are legitimate. The verification process is when the miners solve the mathematical puzzle known as proof of work. The puzzle is taking a transaction as the input and running it through a hashing algorithm to get an output. Each miner should have this proof of work, and it should be the same result across all of them every time it is processed, as long as the input is the same. Producing each proof of work is costly and time-consuming, eating up compute power.

In PoS transactions, miners do not get a reward. Instead, transactions have an amount of wealth, known as the stake. The miners get the stake (transaction fee) for forging the transaction (block) to the chain. PoS does not have to show the calculations taken to validate the transaction.

The Ethereum protocol used in [Azure Blockchain Workbench](#) is based on Proof of Authority. PoA is about smart contracts and committing changes to a blockchain network. This requires multi-member

validator nodes that belong to a consortium to validate the changes so that the transaction is added to the block. Validator nodes have unique Ethereum identities. These nodes are members of a smart-contract permission list and that is when they become a part of a consortium.

In PoA, changes are proposed to the nodes in a consortium and the nodes vote on whether the change is valid. Once a change reaches a majority vote by the nodes, the change is then enacted and committed to the blockchain.

Distributed applications running on Blockchain Workbench are a private/consortium blockchain network. The underlying ether in Ethereum's cryptocurrency does not hold value in a private blockchain network, which means that the Proof of Work protocol is not a good option.

Proof of Authority is a better option in private blockchain networks because all the consensus validators are reputable. Within PoA, each validating node has its own Ethereum identity and is fault tolerant. The fault tolerance is based on having multiple load-balanced nodes (Azure IaaS VMs) that are participants in the network. Each consensus node can delegate other nodes to run on its behalf, ensuring there will always be a participant in the event of an IaaS VM failure.

When and what to use blockchain for?

When considering the use of blockchain technology for your organization, it is important to start with a real business case and a measurable outcome for using it. Blockchain typically fits when organizations in different industries have shared business processes or need to share and validate information in immutable form.

Here are three questions to answer when deciding whether blockchain fits your scenario:

- Can third parties or intermediaries be removed to reduce complexity and cost from a business process?
- Do multiple organizations share or update data for supply chain, medical, legal, or other records?
- Is there a requirement for validation or audit?

If the answer is yes to these three questions, your organization may benefit from taking advantage of blockchain technology. Here is a list of scenarios where blockchain smart-contract technology could be put to good use:

- Tracking ownership, relationships, and lineage of assets (e.g., car titles, house titles, business ownership)
- Supply chain (e.g., logistics, food safety)
- Health data exchange
- Electronic medical records exchange
- Trade Information Warehouse
- Post-trade reconciliation or settlement
- Private equity fund management
- Syndicated loans
- Tracking and protecting valuables (e.g., diamonds, collectibles)
- Cross-border payments

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

- Buying and selling IP (e.g., music, movies, software)
- Processing royalty payments
- Payments for unbanked populations

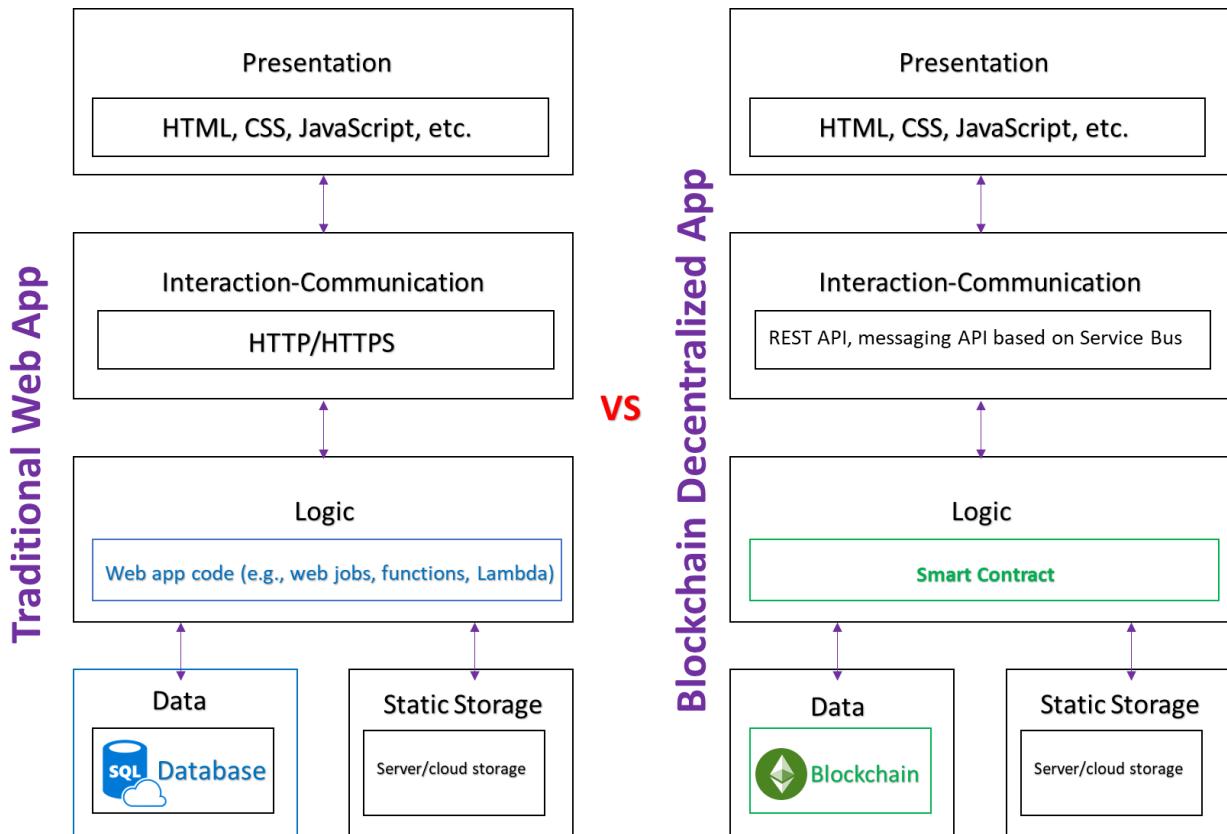
Here is a list of benefits that blockchain technology offers to the enterprise:

- **Reduced fraud**, because if anyone tampers with a contract, validators on the blockchain will invalidate the contract
- **Transparency** through direct relationships with customers via smart contracts
- **Cost-efficiency** by eliminating intermediaries and reducing traditional fees
- **More trust**, because agreements are automatically executed and enforced between enterprises
- **Rock-solid record keeping**, as all executed contracts are immutable and stored on the blockchain
- **A permanent audit trail**, because all contract steps are stored on the blockchain

Given these numerous blockchain scenarios and benefits, it is clear that there are many useful ways that enterprises can take advantage of blockchain technology to enhance their business processes.

Blockchain Workbench architecture

Before we dive into Azure Blockchain Workbench architecture, let's look at a web app compared to a blockchain app. This will give us a clearer understanding of the differences between traditional web apps and a blockchain app based on smart contracts.

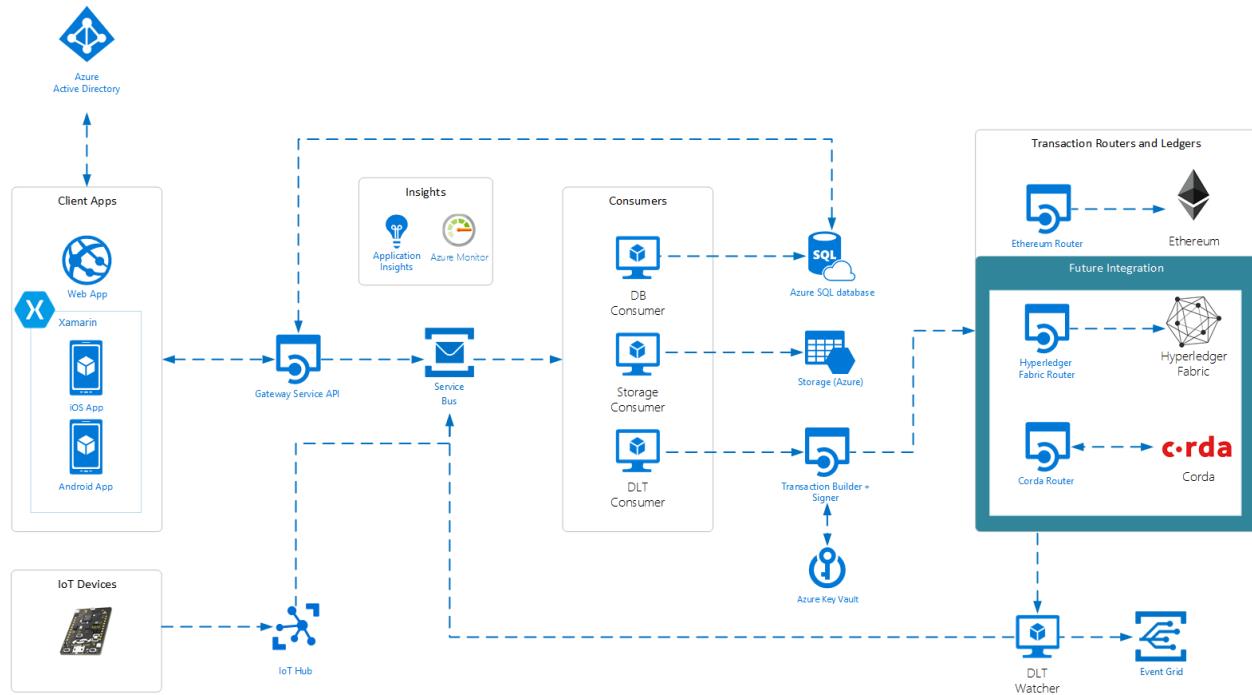


Something new that was introduced in the previous image is “Whisper,” which is used for interaction-communication within a blockchain decentralized app. A part of the Ethereum P2P protocol suite, Whisper simply is a peer-to-peer communication protocol for decentralized apps (DApps) to communicate with each other on a blockchain network. Azure Blockchain Workbench takes your DApp further by complementing and augmenting the blockchain DApp through a number of supporting services. Azure services make it easier to stand up, develop for, and run a blockchain app. When you deploy Blockchain Workbench, you don’t have to also deploy the supporting Azure services because a combination of Azure services and capabilities are dynamically deployed and configured for use with your Blockchain Workbench. They include:

- An App Service plan with two web apps and two web APIs – For admin and client applications
- An Application Insights instance – For monitoring the Blockchain Workbench architecture
- An Event Grid topic – Published endpoint that sources can send events to for routing of the events
- Two Key Vaults – For storing secrets and credentials used by Blockchain Workbench
- A Service Bus namespace – Scoping container for addressing service bus resources within Blockchain Workbench
- A SQL Server with a SQL Database – For off-chain blockchain data storage
- Two Azure Storage accounts – Storage used by Blockchain Workbench
- Two VM scale sets that consist of the ledger nodes and blockchain workbench microservices
- Two virtual network resource groups (with a virtual network, subnets, load balancers, network security groups, and public IP address)

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

Other components used by Blockchain Workbench are Azure Active Directory for identity, Azure Monitor (optional), a log analytics workspace for logging (deployed with Azure Monitor), a mobile app for both iOS and Android, and a REST-based gateway service API. Note that the cost of running Blockchain Workbench is an aggregate of the costs of these Azure services and capabilities. Here is an image of the Blockchain Workbench architecture.



Blockchain Workbench architecture. Source: <https://docs.microsoft.com/en-us/azure/blockchain/workbench/media/architecture/architecture.png>

Now let's look at these supporting services in more detail.

Identity and authentication

Blockchain Workbench can utilize Azure Active Directory (Azure AD) to federate enterprise user accounts for access to a blockchain DApp running on Blockchain Workbench. This eliminates the need to rely on a third-party identity provider. Using Azure AD, which comes integrated with Blockchain Workbench, reduces the complexity of rolling your own identity provider for your blockchain app. All the cases where interactions such as client applications (e.g., web interface, mobile app), REST-based API, and Messaging API must be authenticated are handled by Azure AD.

Client applications

When you deploy Blockchain Workbench, it includes both a web app and mobile app, which can be used to access the blockchain apps running on Blockchain Workbench. The web-based app runs as a web app in Azure App Service. The mobile apps work for both iOS and Android, and they also run on Azure. These become the front-facing UI for your blockchain app. They are automatically generated based on the smart contract metadata by Blockchain Workbench. Users that sign in to these front-facing client apps

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

will use Azure AD-based accounts to authenticate. An example of a DApp running on Blockchain Workbench is shown below.

The screenshot shows the 'Asset Transfer Contract 1' page in the Blockchain Workbench web application. The left sidebar has 'Applications' selected. The main content area includes:

- Status:** Shows a pie chart with 2 Active contracts (08/14/18 at 11:48 PM) and 1 Active contract (08/14/18 at 11:43 PM).
- Actions:** A log of recent actions, including 'Steve Buchanan called Modify' 3 months ago, with a 'Take action' button.
- Activity:** A timeline of events: 'Steve Buchanan recorded action Modify' at 11:48 PM and 'Steve Buchanan recorded action Create' at 11:43 PM.
- Details:** Contract information:
 - Created By: Steve Buchanan
 - Created Date: 08/14/18
 - Contract Id: 1
 - Contract Address: 0x093309dd21fddfb7aebd4795df7a7c2865653c98
 - State: Active
 - Description: Price adjustment
 - Asking Price: 38,000

Something else to note with the Blockchain Workbench web app is that it also doubles as the interface for the Blockchain Workbench administrator. Nothing additional is required because the web-based Administrator Console is also deployed as a part of the web-based application. Whoever has administrator permissions will see administrative functionality when they sign in to the Blockchain Workbench website (as shown in the following screenshot).

The screenshot shows the 'Applications' management page in the Blockchain Workbench web application. The left sidebar has 'Applications' selected. The main content area includes:

- Applications:** A list of deployed applications:
 - AT (Asset Transfer)
 - CR (Car Registration)
 - DL (Digital Locker)
 - HB (Hello, Blockchain!)
- Actions:** Buttons for '+ New' and 'Disable'.
- Filter:** 'Enabled applications' dropdown.
- Members:** 5 members listed as CU, C, C, C.

Gateway service API

Blockchain Workbench comes with a REST-based gateway service API that runs on Azure. Instead of only having the mobile and web app to access the blockchain applications, the API works on behalf of the authenticated user for access as well. Developers can also use this API service to integrate directly with

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

blockchain apps running on Blockchain Workbench. This API delivers messages to an event broker that talks to other components such as an off-chain SQL Database.

Message broker

Some developers, especially those who work in IoT and system-to-system integration, may have a need to send messages directly to the Blockchain Workbench service bus. This is made possible through a supporting component called the message broker a messaging API, which includes incoming and downstream paths sent to the service bus. Incoming is used for sending messages into the Blockchain Workbench. Downstream is used for sending downstream code based on an event.

Message consumers

Blockchain Workbench also comes with three message consumers: distributed ledger, database, and storage. These message consumers get messages off the service bus and send them to other supporting services.

The **distributed ledger consumer** contains distributed ledger technology (DLT) messages, which have the metadata for transactions. This consumer takes the messages and then pushes them to more supporting services including the transaction builder, signer, and router. The **database consumer** gets messages from the service bus and sends them to a SQL Database included with Blockchain Workbench. The **storage consumer** retrieves messages and sends them to Azure Storage, which is also included.

Transaction

Blockchain Workbench comes with transaction builders, signers, routers, and ledgers to help in processing blockchain transactions. Transactions are handled by these components as a part of the smart contract workflow. As previously stated, incoming messages that need to be written to the blockchain are handled by the distributed ledger consumer via distributed ledger technology. The DLT sends the messages along to the transaction builder and signer.

The **transaction builder and signer** assemble a blockchain transaction from metadata and the intended blockchain destination. Assembled transactions are signed using private keys stored in Key Vault. After the transaction is assembled and signed, it is sent to the transaction **routers and ledgers** to send to the correct blockchain.

DLT watcher

The DLT watcher monitors events on the blockchains on Blockchain Workbench such as new contracts, transaction executions, and state changes. It captures these events and sends them to the outbound message broker to downstream consumers.

Azure SQL Database

The Azure SQL Database is included in Blockchain Workbench to store replicated data from the blockchain. The goal is to have off-chain, SQL-accessible blockchain data that is easy to query, visualize, and create reporting from, and that can be used to perform other data-centric integrations.

Azure Storage

Azure Storage is used to store documents and media content that are associated with smart contracts in the blockchain. Blockchain Workbench business logic is able to have documents or media within the app. The blockchain app does not store the actual documents or media; instead, it has a hash that points to the documents or media hosted in Azure Storage. This largely removes the complexity of building a storage system in a blockchain app by providing it within Blockchain Workbench, so developers simply need to know the Blockchain Workbench business logic syntax.

Monitoring

Blockchain Workbench includes a full monitoring solution both of the blockchain network (infrastructure) and at the application level. This is provided by Azure Monitor components including Application Insights for application level monitoring and log analytics for infrastructure monitoring and logging. The information that is monitored and logged includes warnings, errors, successes, and health of the blockchain network.

Blockchain network

At the time of writing this white paper, Ethereum is the only blockchain network supported by Azure Blockchain Workbench. Microsoft plans to add support for Hyperledger and Corda blockchain networks in the future.

Now that you have a better understanding of Azure Blockchain Workbench architecture, let's explore what it takes to deploy Blockchain Workbench and decentralized apps to it.

Deploying Azure Blockchain Workbench

Prerequisites

1. Register an Azure AD application for Blockchain API

To register, you need to create a new app, which is basically a registration in Azure AD that sets the required permissions for users to access the Azure tenant. Within the Azure portal, go to **Azure Active Directory > Select App registrations > New application registration**. Create the app registration (as shown in the screenshot below):

The screenshot shows the 'Blockchain API' app registration in the Azure Blockchain Workbench. The 'Settings' tab is selected. Key details shown include:

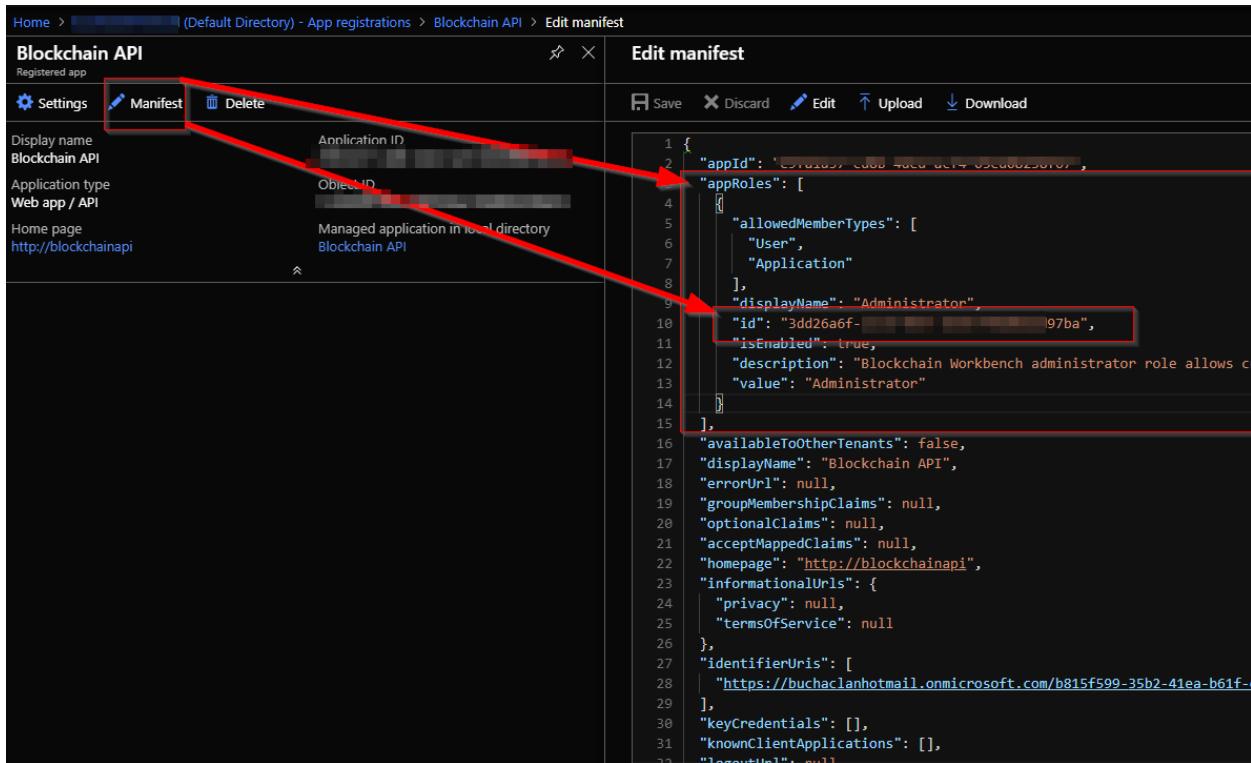
Setting	Value
Display name	Blockchain API
Application type	Web app / API
Home page	http://blockchainapi
Application ID	e9fa1[REDACTED]7
Object ID	7c7[REDACTED]3
Managed application in local directory	Blockchain API

2. Modify the application manifest

Next you need to modify the application manifest of the app registration to use application roles within Azure AD to specify Blockchain Workbench administrators.

Within the app registration, click **Manifest**. Modify the "appRoles" section of the JSON adding a new GUID after "id": (as shown in the following screenshot).

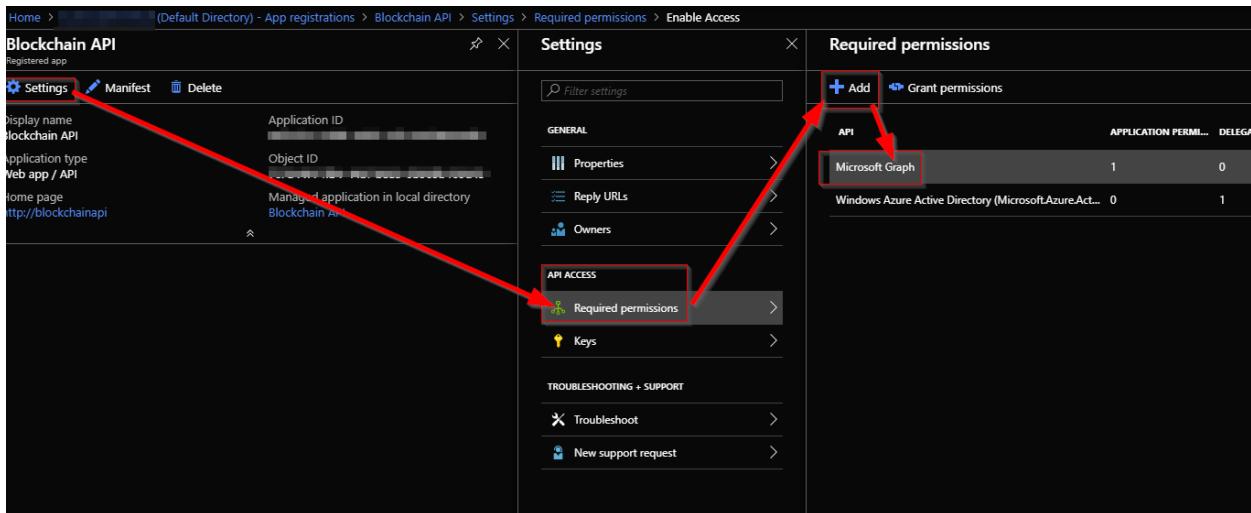
Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench



```
1 "appId": "C0d4a0f2-0000-0000-0000-000000000000",
2 "appRoles": [
3     {
4         "allowedMemberTypes": [
5             "User",
6             "Application"
7         ],
8         "displayName": "Administrator",
9         "id": "3dd26a6f-0000-0000-0000-000000000000",
10        "isEnabled": true,
11        "description": "Blockchain Workbench administrator role allows c",
12        "value": "Administrator"
13    }
14 ],
15 "availableToOtherTenants": false,
16 "displayName": "Blockchain API",
17 "errorUrl": null,
18 "groupMembershipClaims": null,
19 "optionalClaims": null,
20 "acceptMappedClaims": null,
21 "homepage": "http://blockchainapi",
22 "informationalUrls": {
23     "privacy": null,
24     "termsOfService": null
25 },
26 "identifierUris": [
27     "https://buchatechmail.onmicrosoft.com/b815f599-35b2-41ea-b61f-e",
28 ],
29 "keyCredentials": [],
30 "knownClientApplications": [],
31 "tags": []
32 }
```

3. Add Graph API required permissions

The Blockchain API application needs to request permission from users to access the Azure tenant directory. To set this, navigate back to the Blockchain API app registration. Select **Settings > Required permissions > Add > Select an API > Microsoft Graph**.



API	APPLICATION PERMISSIONS	DELEGATED PERMISSIONS
Microsoft Graph	1	0
Windows Azure Active Directory (MicrosoftAzureAct...	0	1

Under **Enable Access > Application permissions**, click the **Read all users full profiles** checkbox to enable it.

Enable Access	
Microsoft Graph	
<input type="button" value="Save"/>	<input type="button" value="Delete"/>
Read mail in all mailboxes	<input checked="" type="checkbox"/> Yes
Read and write mail in all mailboxes	<input checked="" type="checkbox"/> Yes
Send mail as any user	<input checked="" type="checkbox"/> Yes
Read contacts in all mailboxes	<input checked="" type="checkbox"/> Yes
Read and write contacts in all mailboxes	<input checked="" type="checkbox"/> Yes
Read all groups	<input checked="" type="checkbox"/> Yes
Read and write all groups	<input checked="" type="checkbox"/> Yes
Read directory data	<input checked="" type="checkbox"/> Yes
Read and write directory data	<input checked="" type="checkbox"/> Yes
Read and write devices	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/> Read all users' full profiles	<input checked="" type="checkbox"/> Yes

NOTE: To set the permissions, it is critical to click the **Grant Permissions** button under Required permissions and click **Yes** when prompted. This permission is what gives the Blockchain API registered app the ability to read users from the Azure Active directory tenant.

4. Add Graph API key to application

Blockchain Workbench uses a Graph API key to authenticate with Azure AD. This is needed because Azure AD serves as the identity management system for the blockchain apps running on Blockchain Workbench. To get the key, follow these steps:

In the Blockchain API app registration, click **Settings** and then **Keys**.

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

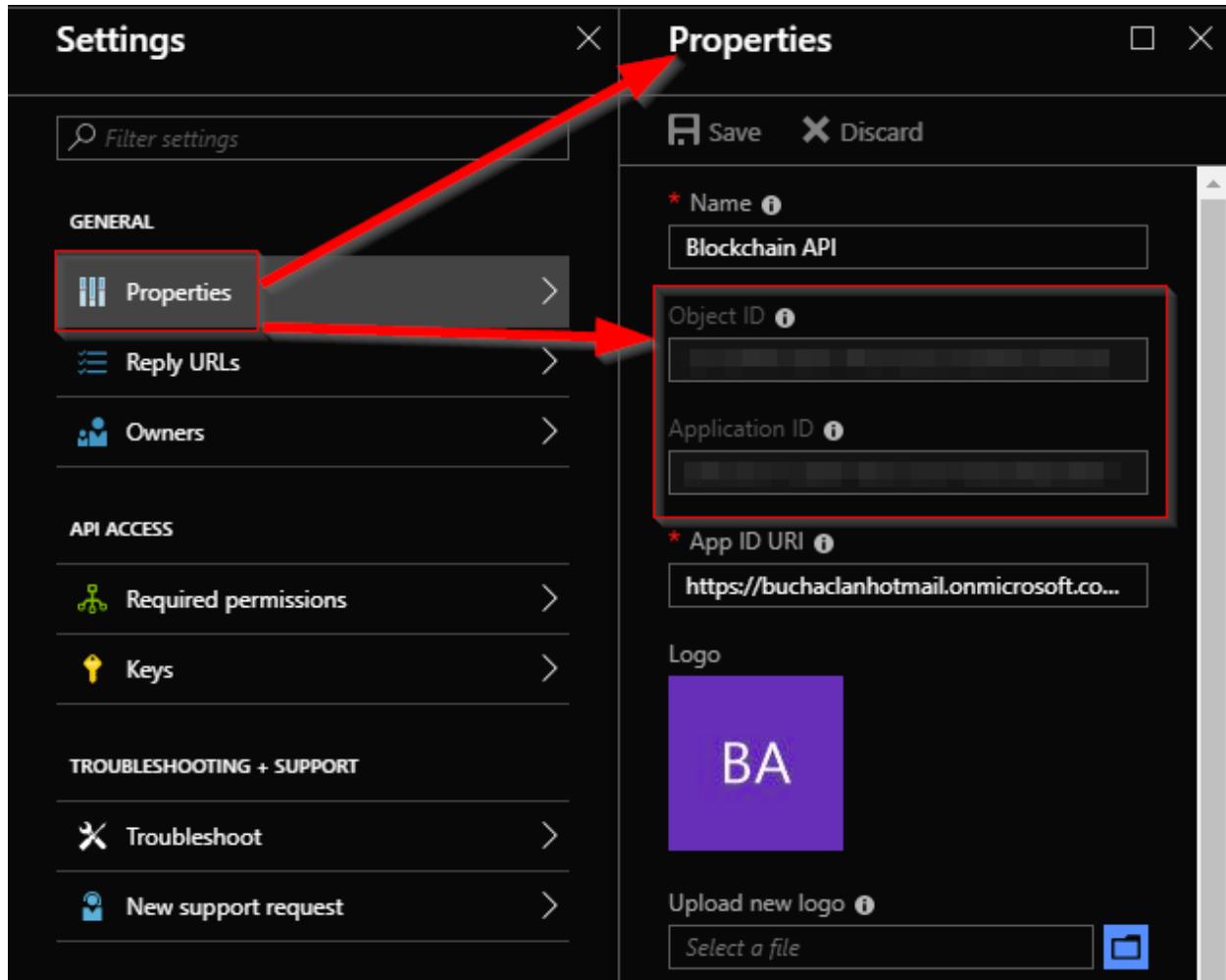
The screenshot shows the Azure Blockchain Workbench interface. On the left, under 'Blockchain API' settings, the 'Settings' tab is selected. It displays basic app information: Display name (Blockchain API), Application type (Web app / API), and Home page (http://blockchainapi). To the right, a 'Settings' blade is open with sections for GENERAL (Properties, Reply URLs, Owners), API ACCESS (Required permissions, Keys), and TROUBLESHOOTING + SUPPORT (Troubleshoot, New support request). A red arrow points from the 'Settings' tab on the left to the 'Keys' section in the blade on the right.

Under Passwords, add a name to Service, set when it expires, and click **Save**. The key is then revealed to you. Copy the key to Notepad because it is only displayed to you once, right after the save button is clicked.

The screenshot shows the 'Keys' blade. At the top, there are 'Save' and 'Discard' buttons, and a 'Upload Public Key' link. A warning message '⚠️ Copy the key value. You won't be able to retrieve after you leave this blade.' is highlighted with a red box. Below this, the 'Passwords' section lists a single entry: 'Service' (Description), '8/13/2019' (Expires), and 'Hidden' (Value). A red arrow points from this row down to the 'Value' field, which contains the key value '4LMUPjUeocO24aVko5NuJF7rY7dPC8c5boMECR9vFE='. At the bottom of the blade, there are input fields for 'Key description', 'Duration', and 'Value will be displayed on save'. Below this, the 'Public Keys' section shows a table with columns: THUMBPRINT, START DATE, and EXPIRES. The table displays the message 'No results.'

5. Get application ID and tenant domain name

Next you will need to get the application ID and tenant information that you will use in the deployment. In the Blockchain API app registration, click **Settings** and then **Properties**. Within Properties, copy the **Object ID** and the **Application ID**.



That concludes the prerequisites for Blockchain Workbench. Now you are all set to deploy the Blockchain Workbench.

Configuring Blockchain Workbench administrators

It is important to make sure that your account and any other account for which you want to be an administrator of Blockchain Workbench has been set up properly. Blockchain Workbench administrators can create blockchain applications, add users, assign users, and remove users from Blockchain Workbench. Note that users need to be in the Azure Active Directory that Blockchain Workbench was deployed to before continuing.

To be a Blockchain Workbench administrator, you need to be a user within the Administrator group of the Azure AD Blockchain API application. The Administrator group of the Blockchain API application is

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

associated with the Administrator application role in the Blockchain Workbench. To set this, follow these steps:

1. Sign in to the Azure portal.
2. Navigate to your Azure AD tenant.
3. Select **Azure Active Directory** and then **Enterprise applications**.
4. Select the **Blockchain API** application.
5. Select **Users and groups** and then click **Add user**.
6. Within Add Assignment, make sure Select Role is set to **Administrator**.
7. Within Add Assignment, click **Users** and search for the user you want to add.
8. Place a check mark next to the user you want to add and then click **Select**.

Deploy Blockchain Workbench

Microsoft has greatly simplified the deployment of the Azure Blockchain Workbench by supplying a wizard UI experience in the Azure portal that asks a series of questions in four areas for your Blockchain Workbench settings and deploys it from there. In fact, satisfying the prerequisites is more work than the deployment. Here are the steps for deploying Blockchain Workbench from within the Azure portal:

1. Open the Azure portal.
2. Click **Create a resource**.
3. Type in **Azure Blockchain Workbench** and click **Search**.
4. Select **Azure Blockchain Workbench** and click **Create**.

The screenshot shows the Azure Marketplace interface. On the left, a search bar contains 'Azure Blockchain Workbench'. Below it, filters for Pricing (All), Operating System (All), and Publisher (All) are applied. The results table shows one item: 'Azure Blockchain Workbench' by Microsoft, categorized under Compute. A red arrow points from the search bar to the result row. On the right, the product details page for 'Azure Blockchain Workbench' is displayed. It includes a brief description, a note about the deployment experience, a list of services included (App Service Plan, Application Insights, Event Grid Topic, Azure Key Vaults, Service Bus Namespace, SQL Databases, Azure Storage accounts, Virtual Machine scale sets, Virtual Network resource groups, and Optional: Azure Monitor), and a note about costs. At the bottom, a 'Create' button is highlighted with a red arrow. A dropdown menu above the 'Create' button shows 'Resource Manager' selected.

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

You will then see the screen for Blockchain Workbench creation. It will have four sections of settings you need to complete (the first four in the following screenshot).

The screenshot shows the 'Create Azure Blockchain Workbench' wizard with the following steps visible on the left:

- 1 Basics**: Configure basic settings.
- 2 Azure Active Directory setup**: Enter the parameters for this d...
- 3 Network Settings and Perform...**: Define the network settings for...
- 4 Azure Monitor**: Create new or connect to existi...
- 5 Summary**: Azure Blockchain Workbench
- 6 Buy**

The right side of the screen displays the 'Basics' configuration page with the following fields:

- * Resource prefix
- * VM user name
- * Authentication type Password SSH public key
- * Password
- * Confirm password
- * Database password
- * Confirm database password
- * Deployment region East US
- Subscription MVP Account 0
- * Resource group Select existing... Create new
- * Location South Central US

A large blue 'OK' button is located at the bottom right of the configuration area.

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

The following tables lay out the settings you will need to specify as a part of the four sections when creating a Blockchain Workbench instance.

#1 Basics

Setting	Description
Resource prefix	A unique identifier for your Blockchain Workbench deployment.
VM username	The username used as administrator for all virtual machines.
Authentication type	Select if you want to use a password or key for connecting to VMs.
Password	Password used for connecting to Blockchain Workbench VMs.
SSH	An SSH key here, which can be generated using ssh-keygen on Linux and OS X, or by using PuTTYgen on Windows.
Database password / Confirm database password	Password for access to the database created as part of the deployment.
Deployment region	Azure location to deploy the Blockchain Workbench resources.
Subscription	Azure Subscription to be used for the deployment.
Resource groups	Resource group to be used for the deployment.
Location	Azure region where you will deploy the Blockchain Workbench framework. Best practice is that this be the same as the deployment region setting.

#2 Azure Active Directory setup

Setting	Description
Domain Name	Azure AD tenant from the tenant domain name prerequisite.
Application ID	Application ID from the blockchain client app registration prerequisite.
Application Key	Application key from the blockchain client app registration prerequisite.

#3 Network settings and performance

Setting	Description
---------	-------------

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

Ethereum RPC endpoint	If you select "Use Existing," specify the RPC endpoint of an existing PoA blockchain network. The endpoint starts with http:// and ends with a port number. For example, http://domain.onmicrosoft.com:4545.
Storage performance	Preferred VM storage performance for the blockchain network.
Virtual machine size	Preferred VM size for the blockchain network.

#4 Azure Monitor

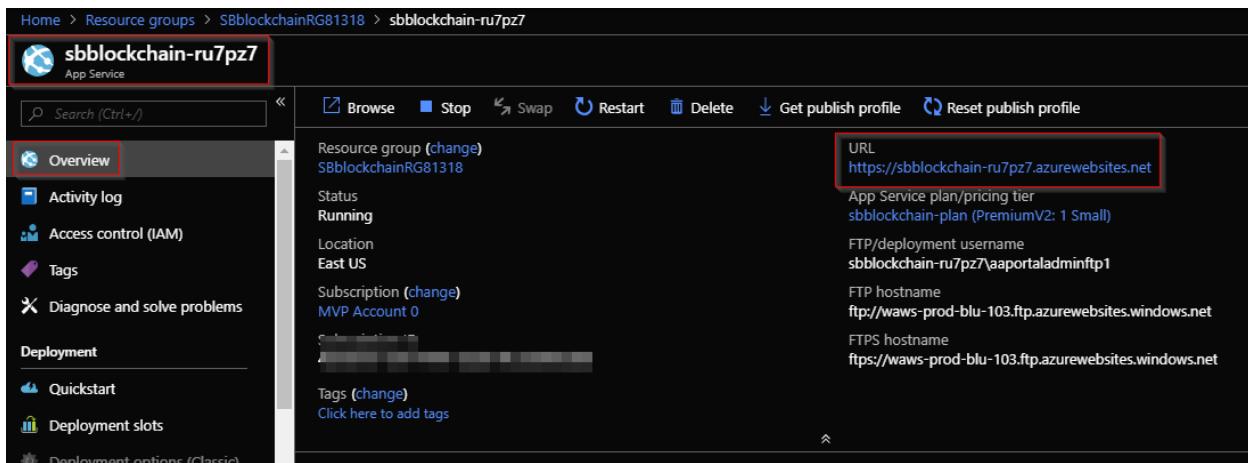
Setting	Description
Monitoring	Choose whether you want to enable Azure Monitor to monitor your blockchain network.
Connect to existing Log Analytics instance	Specify an existing Log Analytics workspace or create a new one. If you choose an existing Log Analytics workspace, enter the workspace ID and primary key.

Deploy a blockchain application

After your Blockchain Workbench has been deployed, you now can sign in to the web portal to deploy blockchain apps to it. A URL is added during the deployment. To get the URL, follow these steps:

1. In the Azure portal, go to the new resource group for the deployed workbench.
2. Sort the resources by type and locate the App Service resources.
3. There will be two App Service resources. Click the App Service that does not have *-api* at the end of its name.
4. Within the App Service overview pane, you will see the Blockchain Workbench web portal address under URL. The URL will be like https://APPSSERVICERESOURCENAME.azurewebsites.net. Go ahead and click this URL (as shown in the following screenshot).

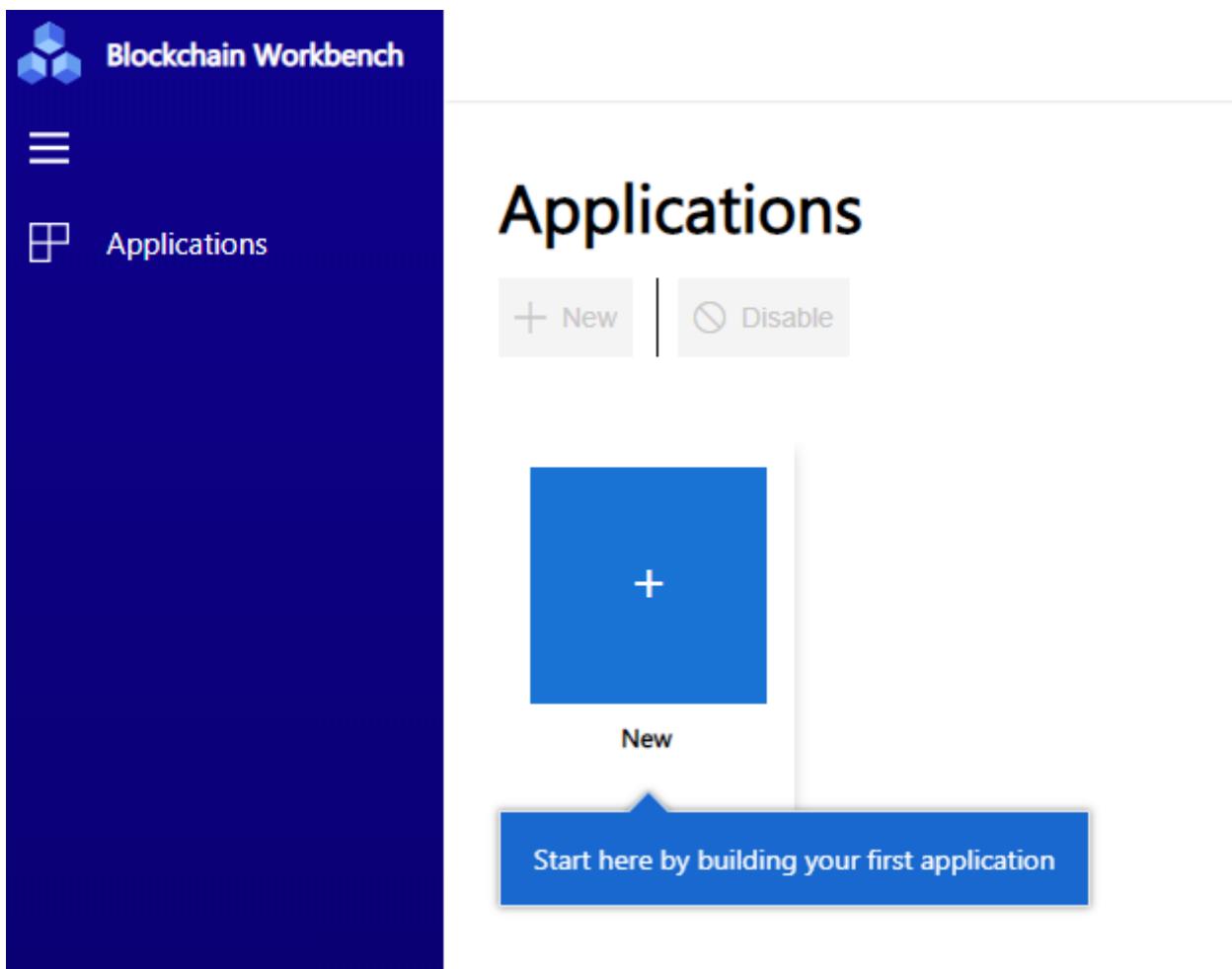
Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench



The screenshot shows the Azure portal's 'Resource groups' section. A specific app service named 'sbblockchain-ru7pz7' is selected. The 'Overview' tab is active. Key details shown include:

- Resource group:** SBlockchainRG81318
- Status:** Running
- Location:** East US
- Subscription:** MVP Account 0
- URL:** <https://sbblockchain-ru7pz7.azurewebsites.net>
- App Service plan/pricing tier:** sbblockchain-plan (PremiumV2: 1 Small)
- FTP/deployment username:** sbblockchain-ru7pz7@aportaladminftp1
- FTP hostname:** <ftp://waws-prod-blu-103.ftp.azurewebsites.windows.net>
- FTPS hostname:** <https://waws-prod-blu-103.ftp.azurewebsites.windows.net>

As long as permissions are configured properly, you will be signed in to the Blockchain Workbench portal as the Blockchain Workbench administrator. You will not have any blockchain applications deployed yet.



The screenshot shows the Microsoft Blockchain Workbench portal. The left sidebar has a dark blue background with a navigation menu. The 'Applications' option is selected and highlighted in blue. The main content area has a white background with a large blue button containing a white plus sign. Below the button, the word 'New' is written. At the bottom of the main area, there is a blue call-to-action button with the text 'Start here by building your first application'.

Microsoft hosts a GitHub repository of blockchain applications and contract samples. These can be deployed to Azure Blockchain Workbench as a starting point. These are fully functioning blockchain apps that can be used as is or modified for other scenarios. Each application will consist of a *.json* and *.sol* file.

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

The JSON is the contract configuration and the SOL is the contract code. They are both needed when deploying the application to Blockchain Workbench.

The *.json* configuration file contains metadata that defines the workflows including stages and actions of the smart contract in the blockchain application. The configuration file is used to set the application name, description, application roles, and the workflows including functions and states.

The *.sol* file is the smart contract code file. It defines the business logic of the smart contract in the blockchain application. Solidity, the programming language for *.sol* files, is a contract-oriented language like object-oriented. Smart contracts in Solidity are just like classes. Solidity is used by Ethereum as it targets the Ethereum Virtual Machine. Solidity has a plugin for Microsoft Visual Studio Code that you can find here: <http://juan.blanco.ws/solidity-contracts-in-visual-studio-code>.

Sample Blockchain Workbench applications can be found here: <https://github.com/Azure-Samples/blockchain/tree/master/blockchain-workbench/application-and-smart-contract-samples>. At the time of writing this white paper, the available smart contract blockchain apps are:

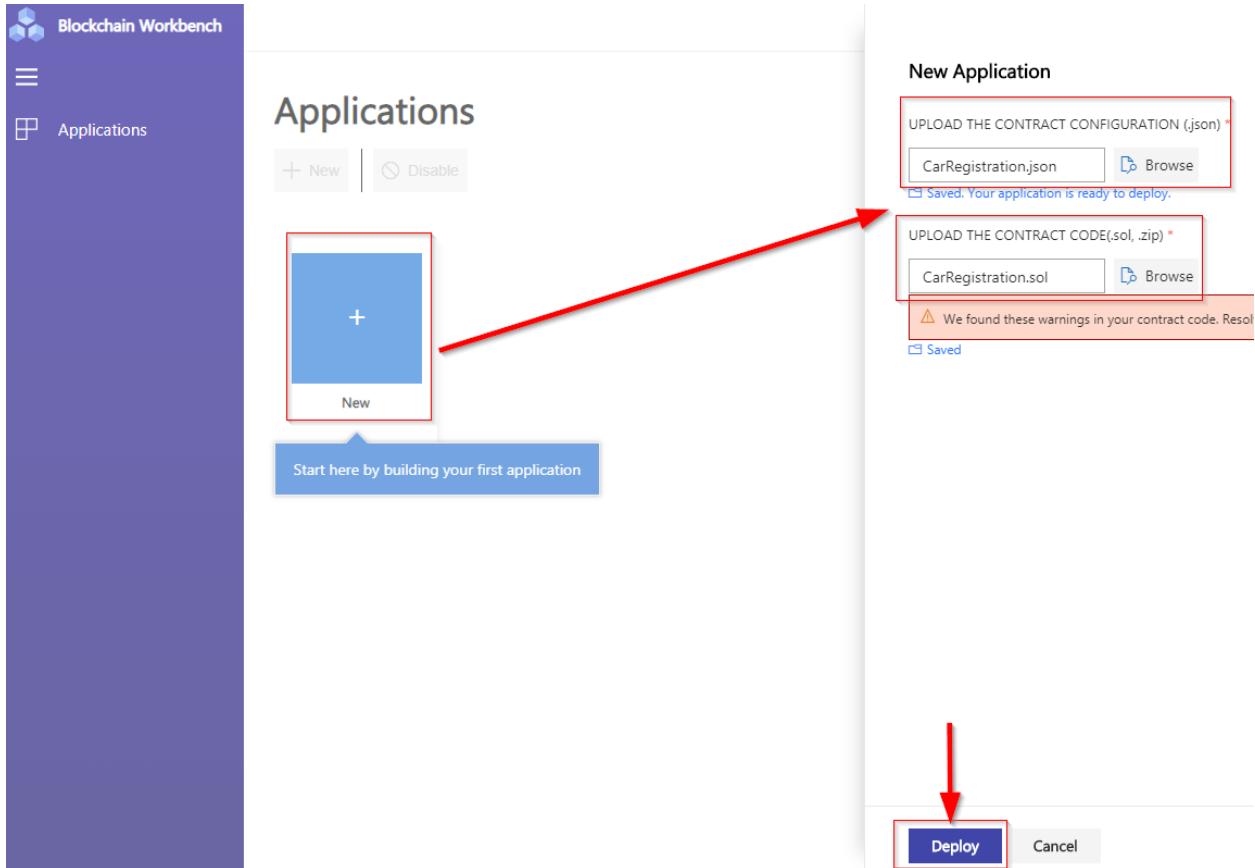
- Hello Blockchain – A workflow between a person sending a request and a person responding to the request.
- Simple Marketplace – A workflow for a simple transaction between an owner and a buyer in a marketplace.
- Basic Provenance – A workflow for a simple record of ownership or responsibility.
- Digital Locker – This workflow shares digitally locked files where the owner of the files controls the access to these files.
- Refrigerated Transportation – A workflow where a supply chain transport requires certain compliance rules must be met throughout the duration of the transportation process.
- Asset Transfer – A workflow for buying and selling high value assets between a buyer and a seller but also requires an inspector and appraiser.
- Bazaar Item Listing – This workflow consists of two workflows to demonstrate how a blockchain application can consist of more than one workflow and how a contract can deploy other contracts.
- Room Thermostat – A workflow where someone will install a thermostat and indicate the intended user for the thermostat, who becomes the assigned user and can set the mode and temperature of the thermostat.
- Defective Component Counter – A workflow that where a manufacturer gets the total number of defective components over a 12-month period.
- Frequent Flyer Rewards Calculator – This workflow binds an airline representative, a flyer, and the number of rewards per mile in a smart contract.

To deploy a new app to Azure Blockchain Workbench, follow these steps:

1. Download the *.json* and *.sol* files for the blockchain app locally.
2. Sign in to the Blockchain Workbench website. Make sure you sign in as an Azure Blockchain Workbench administrator.
3. Click **Applications** and **New**.
4. Click **Upload the contract configuration** and upload the blockchain app's *.json* file. The file will be validated.

5. Upload the contract code. The file will be validated.
6. Click **Deploy** to install the blockchain application.

The following screenshot shows what the app deployment looks like.



After a successful deployment, the blockchain app will look like the following screenshot.

The screenshot shows the Azure Blockchain Workbench interface. On the left, there's a dark sidebar with a blue hexagonal logo, the text 'Blockchain Workbench', a three-line menu icon, and an 'Applications' icon. The main area has a light gray header with the word 'Applications' in large, bold, dark blue letters. Below the header are two buttons: '+ New' and 'Disable'. A large yellow square card contains the letters 'CR' in white. Below the card, the text 'Car Registration' is in blue, followed by 'Deployed 09/15/18' in gray, and 'Enabled' in blue at the bottom.

After your blockchain app is deployed, click it. You will notice that you will have 0 members assigned to the smart contract app. You will need to add a user (or users) as a member in the blockchain app. To add a new member to your blockchain app, use the following steps:

- Click the members link in the upper right corner of the blockchain web portal.
- Next select or type the member's name you want to add. Note that only users in the Azure AD tenant in the Blockchain Workbench tenant will be available to add.
- Now select the role you want this member to have. For example, the first user would be a requester.
- Click the **Add** button.

Enterprise ready

As we have seen, smart contract-based blockchain is ready for prime time. Azure Blockchain Workbench can get you from nothing to a blockchain application faster than you could on your own. With some imagination and hard work to build blockchain applications for today's and tomorrow's scenarios,

Blockchain Beyond Cryptocurrency: A white paper on Azure Blockchain Workbench

blockchain is well on its way to becoming a norm in the enterprise. Use cases such as Voting, Blind Auction, Remote Purchase, Supply Chain, and Validation for the enterprise are just the beginning of this wave of technologies.

Get started with Blockchain Workbench today by visiting these links:

[Azure Blockchain Workbench](#)

[What Is Azure Blockchain Workbench?](#)

[Azure Blockchain Development Kit](#)

[Azure Blockchain Service](#)