CHAPTER 14

Blockchain-as-a-Service

In an earlier chapter, we talked about using lean methodologies to discover whether integrating a blockchain is the right choice; we also provided an open source avenue in the form of open source DLTs available through Hyperledger. This chapter extends our prior discussion to Blockchain-as-a-Service: a more immersive environment for testing a blockchain deployment and smart contracts in the cloud.

Blockchain-as-a-Service (BaaS) is a cloud-based blockchain service that provides a development environment for customers to write smart contracts, rapidly prototype blockchain applications, and deploy blockchain-based consortia with ease. It eliminates barriers to entry, particularly the massive up-front costs of hardware and expertise needed to deploy a blockchain network. This allows startups and even large companies trying to experiment with the blockchain to better understand the finances of integrating a blockchain into the existing business model. The role of a BaaS provider is analogous to that of a traditional web host. The provider manages the infrastructure, performs hardware upgrades, and keeps the network operational for customers to build complex business applications that rely on a blockchain. In terms of maintenance, the provider relies on networking tools for proper allocation of resources, manages bandwidth, and provides support for other hosting requirements. Using a BaaS service can help with mass adoption of blockchain technology as customers can focus on their application and core development while avoiding the technical complexities of creating a blockchain network and infrastructure-related support issues.

A very interesting international example of the BaaS phenomenon helping with mass adoption is the Blockchain-based Service Network (BSN): a tailored infrastructure for blockchain development with managed support in the cloud. BSN was the result of a large collaborative effort between local blockchain companies in China with a vested interest in cloud infrastructure. BSN has already integrated with six public chains, including Tezos, NEO, Nervos, EOS, IRISnet, and Ethereum. This project aims to include more than ten public chains in the near future and offer more security features with technical services that allow developers to streamline the development process.

V. Dhillon et al., *Blockchain Enabled Applications*, https://doi.org/10.1007/978-1-4842-6534-5_14

In this chapter, we study the Blockchain-as-a-Service (BaaS) offerings from four providers (three major players in the cloud blockchain world and one ConsenSys-backed startup): Microsoft Azure, Amazon Web Services, Oracle, and Kaleido. Our main focus will be on the unique aspects of BaaS providers and how each provider is supporting the growth of new organizations/startups in the cloud.

Service Providers

At the time of writing, there are ten or so BaaS providers, all offering varying degrees of services and support. While the list of criteria for evaluating a BaaS provider can be endless, each business application will have specific needs. There are two dimensions of evaluation here: the cloud provider hosting your BaaS instance, and the BaaS instance itself. Here, we want to focus on four areas to keep in mind when selecting a BaaS provider:

- Prior experience in setting up blockchain infrastructure: As the BaaS provider market expands, it is crucial to choose providers with an established track record of developing and deploying blockchain infrastructure. The reliability of infrastructure becomes more apparent and crucial as business applications start to scale with more customers. For instance, a cloud provider with experience with blockchain transaction volume can set up load-balancers that can respond appropriately to a scaling BaaS setup.
- Security and data-redundancy standards: It is paramount to get familiar with the security and backup/disaster-recovery policies of the cloud platform hosting your BaaS instance. This comes down to understanding the security of private–public keys being stored and used in the cloud, on-site security team handling cloud provider's BaaS instances, being compliant with security regulations, and redundancy of backups holding your blockchain instance.
- Integrations: Does your BaaS provider have plans for the integration of services that can enable new features and applications to be built on the blockchain? This is crucial for new startups as they pivot to find new avenues, or add new features to their minimum viable product. Additionally, developers must make sure that new features

and integrations are easy to adopt into existing workflows. This reduces the additional overhead of becoming familiar with a new cloud system.

- Pricing and tiers of support: Transparent pricing is key in any BaaS provider; this also helps in planning the finances around your BaaS instance. Many providers offer tiers of pricing for using the infrastructure combined with levels of support. Fully managed instances require less in-house management of the instance once it moves online. This option is suitable for startups or companies without domain expertise in setting up BaaS or networking and allows them to focus only on blockchain development. Partially managed options are better suited for startups that are heavily tech-focused and have in-house expertise.

Figure 14-1 provides a graphical summary of the main BaaS providers on the market.

Hosting Platforms	Ethereum	Quorum	Corda	Fabric	MultiChain	Digital Asset
AWS	✓	✓[1]	✓[2]	✓		
Azure	✓	✓	✓	✓	✓[3]	
Google	✓[4]			✓		✓
HPE			✓			
IBM				✓		
Oracle				✓		
SAP				✓	✓	

1. Offered via Kaleido in AWS marketplace
2. Full server offering, not a container module
3. An older template from 2006
4. Full Ethereum version coming out later in the year

Figure 14-1. *BaaS hosting summary*

Microsoft Azure

Azure offers a Blockchain-as-a-Service platform by providing customers with easy-to-deploy enterprise-grade templates for four major distributed ledger protocols, including Ethereum, Quorum, Fabric, and Corda (and many more planned). The Azure BaaS is divided into three main components: a fully managed Azure Blockchain Service, a development and management platform called Azure Blockchain Workbench, and integrations bundled in Azure Blockchain Development Kit. This section was prepared with reference material from the Azure Blockchain Service documentation.

Azure Blockchain Service

This is Microsoft Azure's fully managed BaaS offering. Customers can create and deploy a permissioned Quorum network with a few clicks and manage the network policies along with security options using a GUI in the Azure Portal. In addition, Microsoft is releasing a Visual Studio extension that assists users in 1) writing and compiling Ethereum smart contracts; 2) deploying them to a consortium network through Azure Blockchain service or the public main-net; and 3) managing them through Azure Portal. This service enables organizations and consortia to grow and scale completely in the cloud with Azure without having to worry about underlying provisioning. Granular network governance and simplified infrastructure management allow for simple network deployment, easy-to-manage network operations and security, and the development of smart contracts with developer-friendly tools such as Visual Studio.

Currently, Azure offers Quorum ledger with the Istanbul Byzantine Fault Tolerance consensus mechanism. Using a managed service allows customers to focus on developing their core products and business logic without worrying about the underlying virtual machine infrastructure. As with most cloud services, the use of virtual machines allows for redundant backups of the whole blockchain and the possibility of network restoration from a specific time slot if necessary.

Azure Blockchain Workbench

Azure Blockchain Workbench is a collection of Azure support services that help deploy and manage blockchain applications and share business processes with other organizations on a network. This workbench provides underlying infrastructure to capture business models and codify them into smart contracts that can be deployed to an existing blockchain network. It also provides developers with automation capabilities for redundant tasks. The workbench simplifies setting up a consortium blockchain network by providing a template within Azure Resource Manager; for now, the template only supports Ethereum, but more ledger protocols are under development. The key feature of Workbench is its integrations with existing Azure components; for instance, the REST API as explained by the Azure Blockchain documentation:

You can use the Blockchain Workbench REST APIs and message-based APIs to integrate with existing systems. The APIs provide an interface to allow for replacing or using multiple distributed ledger technologies, storage, and database offerings.

Blockchain Workbench can transform messages sent to its message-based API to build transactions in a format expected by that blockchain's native API. Workbench can sign and route transactions to the appropriate blockchain.

Workbench automatically delivers events to Service Bus and Event Grid to send messages to downstream consumers. Developers can integrate with either of these messaging systems to drive transactions and to look at results.

Azure Blockchain Workbench makes it easier to analyze blockchain events and data by automatically synchronizing data on the blockchain to off-chain storage. Instead of extracting data directly from the blockchain, you can query off-chain database systems such as SQL Server. Blockchain expertise is not required for end users who are doing data analysis tasks.

Azure Blockchain Development Kit

The Blockchain Development Kit is a comprehensive GitHub repository that includes code samples with a particular focus on integrations with Azure intelligent services. The development kit could be extended in the near future to build machine learning apps that run on a blockchain powered by Azure. Here, we will focus on two such examples: IoT Central and Cognitive Search.

Note Machine learning capabilities are already available in Azure; however, once data from a blockchain system can be imported, it can be accessed by other Azure services. In the future, machine learning services can "scan" the blockchain and have some level of event reporting that allows for network-wide alerts and notifications.

The Cognitive Search sample shows a user how to take ledger data from Ethereum and import that information to Azure Search. Once imported, it becomes available to a wide range of Enterprise applications, and the smart search options allow for the possibility of data-mining and learning from your users. The IoT example demonstrates how to connect physical products to a digital cloud and import data into your blockchain consortium. In IoT Central, you can create simulated devices, provision new devices, manage and troubleshoot physical devices, and define custom rules for incoming data and custom actions for data.

Amazon Web Services

Amazon Managed Blockchain is a fully managed BaaS offering from AWS that makes it easier to build and deploy blockchain networks in the cloud based on open source DLTs, such as Hyperledger Fabric and Ethereum, soon coming to BaaS. Presently, building scalable blockchain networks from scratch and deploying applications is technically challenging. Adding multiple parties to your network extends that challenge as each new network member needs to install software, create new security certificates or private keys, manually provision hardware, and configure networking to support the network. In addition, once a blockchain network is online, the developers have the additional task of monitoring for increased transaction requests, new members joining, and infrastructure health. Amazon Managed Blockchain streamlines this process and reduces the technical overhead. The network can scale, adapting to increased load and transaction volume. In addition, new members can join the cloud with ease without needing any special hardware.

Ease of Setup (taken from Amazon Managed Blockchain documentation)

In the Managed Blockchain offering, launching a new network can be done via templates within minutes, without the need for extensive configuration. After the network is available, new members can join the network with an invite. Once a new member accepts membership, they can configure peer nodes that provide computational storage and memory to execute DApps and maintain a copy of the ledger, all in the cloud. This bypasses the need for any local hardware throughout the process. During periods of heavy transaction load and network traffic, scaling an application simply requires additional peer nodes to be added. New peer nodes can be provisioned within the cloud network with ease. New nodes can be instantiated to support your increased workload via a variety of instance families pre-configured into AWS Cloud Services.

Security (taken from Amazon Managed Blockchain documentation)

Managing private–public keys is crucial for blockchain-based platforms. The keys are needed for accessing your wallet on a blockchain instance, signing contracts, and enrolling in security certificates and identity management. Managed Blockchain relies on AWS Key Management Service to act as the certificate authority for Hyperledger Fabric. To that end, new users signing onto a Managed Blockchain will not have to worry about setting up hardware security modules.

Immutable Ordering Service (taken from Amazon Managed Blockchain docs)

The default ordering service for Hyperledger Fabric to support propagation of transactions across the network is Apache Kafka. Even though Kafka is a messaging service that delivers transactions (and the associated information) sequentially across the network, it is not optimized to create a sequential log of transactional history. Therefore, in case of network failure, Kafka makes it difficult to retrieve historical transactions. Managed Blockchain has a new ordering service built on Quantum Ledger Database (QLDB) that provides an immutable log and maintains a complete network history of all uncommitted transactions in a blockchain network. This QLDB service uses a centralized trusted authority for maintaining histories and supplements the decentralized ledger.

Oracle

Permissioned blockchain networks begin with a few core-member organizations, and network governance is easier; however, as the organization starts to expand with new members, fair allocation of underlying hardware resources becomes more intricate and complex. This is particularly true in the context of ordering responsibilities and selecting members for private channels in Fabric implementations. Recall that a set of ordering nodes bundles current transactions into blocks, finalizes the block, and broadcasts a block to the network. Under an older consensus implementation, a leader is selected at random from the global pool of ordering nodes for creating the next block. The orderers also maintain a list of organizations that are allowed to create private channels.

A channel is created by an ordering node, and it functions as a private subnet on the Fabric blockchain where network members can conduct private and confidential transactions. In the past, a cluster of Kafka and Zookeeper implementations was managing the ordering nodes; however, in the cloud, this setup is more resource intensive. Recently, a new consensus plugin called RAFT was introduced to manage the ordering service and make it ready for enterprise-grade production networks. RAFT is a dynamic leader-based protocol where a set of ordering nodes (called the consenter set) cooperates in an ordering cluster to create blocks. This ensures a more fair and uniform participation of member organizations, even with a large number of members. Additionally, an ordering cluster is configured such that it provides enhanced privacy: any node from the ordering cluster can create a channel with specific member organizations to handle a specific private transaction. This allows more granular control over transaction volume by selecting from specific members, rather than using a global ordering queue. Using RAFT also streamlines computational resource usage by eliminating the need for a resource-intensive cluster in the cloud. Next, we will talk about the main features of Oracle BaaS from three major categories: infrastructure maintenance, identity, and backups.

Maintenance (taken from Oracle Support Documentation)

- Includes Oracle operations monitoring
- Has zero downtime managed patching and updates
- Includes embedded ledger and configuration backups
- Provides a comprehensive, intuitive web user interface and wizards to automate many administration tasks. For example, adding organizations to the network, adding new nodes, creating new channels, deploying and instantiating chaincodes, browsing the ledger, and more.

Identity (taken from Oracle Support Documentation)

- Supports identity federation and third-party client certificate support to enable consortia formation and simplify member onboarding

- Built-in integration with Oracle Identity Cloud Service for user authentication, roles management, and identity federation immediately leverages Oracle Identity Cloud Service accounts and enables easy onboarding of consortium members who prefer using SAML-based federation for authentication against their own identity providers

Backups (taken from Oracle Support Documentation)

- Peer-node containers distributed across multiple VMs to ensure resiliency if one of the VMs is unavailable or is being patched
- Orderers, fabric-ca, console, and REST proxy nodes are replicated in all VMs for transparent takeover to avoid outages.
- Isolated VM environments for customer chaincode execution containers for greater security and stability
- Autonomous monitoring and recovery agents in all components, leveraging dynamic object store backups of all configuration updates and ledger blocks to enable autonomous recovery

Kaleido

Kaleido is a ConsenSys-backed startup that has created two major blockchain offerings: an Ethereum-based software package that can run on Amazon Cloud to simplify development of blockchain applications, and a fully managed Blockchain-as-a-Service cloud called Kaleido Core. There are two main facets to Kaleido that we will discuss in this section.

Network Governance (taken from Kaleido blog on BaaS)

In the managed Kaleido service, a major theme is the creation of templates dedicated to common governance tasks and the provision of them to users through a simple UI. This includes a new-member onboarding workflow that can set up new members and peer-nodes with a few clicks. In addition, once the new members are on-boarded and familiar with the network, they have ownership of their node and key data. This allows for more granular control over node settings and flexibility in opting out of optional network

features. Kaleido also provides a network-wide address book that lists all the users and their affiliations (within the consortia) and access permissions. For a more transparent history of transactions and network activity, a standard block explorer and token explorer for the ledger running in BaaS are also included. Finally, a native smart-contract manager is embedded into network governance. This includes common smart-contract tasks such as workflow for security tasks, firewall protection, versioning, and contract deployment to a ledger.

High Availablility and Disaster Recovery (taken from Kaleido blog on BaaS)

A blockchain network hosted in the cloud needs to be adaptive so that spikes in traffic or transaction load can be handled without the network going offline. To that end, high availability is crucial for the consensus algorithms, especially in a decentralized network. This is especially important for provisioning new nodes; for instance, if five new nodes join a network, they should be dynamically added to the consensus algorithm in order to prevent overload on the network. Kaleido's Managed Blockchain also comes with network protections against new/untested consensus algorithms in order to provide some level of disaster recovery support. In the near future, this will allow developers to deploy and hot-switch new consensus algorithms and collect data on how the algorithm behaves during a period of high transaction volume.

Summary

In this chapter, we reviewed the basic concept of a Blockchain-as-a-Service platform and talked about four major platforms. We discussed the main features of each offering, the managed support given by each provider, and opportunities for blockchain companies to grow completely in the cloud.

References

The main references for this chapter were based on developer documentation from Microsoft Azure Blockchain docs (https://azure.microsoft.com/en-us/solutions/blockchain/), Blockchain AWS docs (https://aws.amazon.com/blockchain/), Oracle Platform docs (https://www.oracle.com/application-development/cloud-services/blockchain-platform/), and Kaleido docs (https://docs.kaleido.io/).