Blockchain Enabled Applications

Understand the Blockchain Ecosystem and How to Make it Work for You

—

Second Edition

—

Vikram Dhillon
David Metcalf
Max Hooper

Apress®

Blockchain Enabled Applications

Understand the Blockchain Ecosystem and How to Make it Work for You

Second Edition

Vikram Dhillon
David Metcalf
Max Hooper

Apress®

Blockchain Enabled Applications: Understand the Blockchain Ecosystem and How to Make it Work for You

Vikram Dhillon
Orlando, FL, USA

David Metcalf
Orlando, FL, USA

Max Hooper
Orlando, FL, USA

ISBN-13 (pbk): 978-1-4842-6533-8
ISBN-13 (electronic): 978-1-4842-6534-5
https://doi.org/10.1007/978-1-4842-6534-5

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Spandana Chatterjee
Development Editor: Laura Berendson
Coordinating Editor: Shrikant Vishwakarma

Cover designed by eStudioCalamar

Cover image designed by Pexels

Distributed to the book trade worldwide by Springer Science+Business Media LLC, 1 New York Plaza, Suite 4600, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, email orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science+Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-6533-8. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

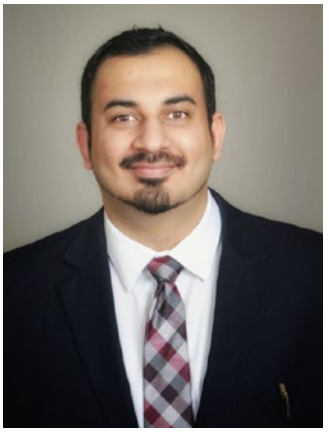
Vikram Dhillon would like to dedicate this work to Aaron Hillel Swartz and his legacy.

David Metcalf would like to thank Katy, Adam, and Andrew for their patience during the extended hours and effort required while putting the book together, and the colleagues and students at UCF and through the NSF I-Corps program who identified the power of Bitcoin and blockchain technology years ago and shared their knowledge and future strategies that inspired us early on to pursue this area of research. Thank you to my co-authors and our outside collaborators and contributors, and of course to God for the wisdom, ability, and grit needed to bring this effort to life.

Max Hooper would like to thank his co-authors and colleagues at UCF/METIL Lab, along with special thanks to Mindy Hooper for her help and support. Additionally, he thanks God for his inspiration, guidance, direction, and wisdom. He would like to acknowledge His leadership.

Table of Contents

About the Authors



Vikram Dhillon is an internal medicine resident physician at Wayne State University, Detroit Medical Center, as well as a research fellow at the Institute of Simulation and Training, University of Central Florida (UCF). A graduate of the University of Central Florida, he holds a Bachelor of Science degree in molecular biology with a focus in bioinformatics, as well as a Doctor of Osteopathic Medicine doctoral degree from Nova Southeastern University. He has authored four textbooks on blockchain design and published several scientific papers in the fields of computational genomics and public health. In the past, he worked as a software and business development coach at the Blackstone Launchpad to mentor young entrepreneurs and startups through the process of building technology products. He was previously funded by the National Science Foundation through the Innovation Corps program to study customer discovery and apply it to commercializing high-risk startup ideas. Currently a member of the Linux Foundation, he has been involved in open-source projects and initiatives for the past several years. He often speaks at local conferences on the topics of blockchain design, security, and entrepreneurship. He can be reached on Twitter @opsbug.

David Metcalf is a serial entrepreneur who has launched multiple successful ventures and spinoff companies. He has reviewed thousands of emerging technology companies as an advisor and investor. He is the director of the Mixed Emerging Technology Integration Lab at UCF's Institute for Simulation and Training. His past projects involving XR and IoT span across education, health, space, cyber, and transportation. Current efforts include smart cities, blockchain, and enterprise learning transformation for government and industry. He is the co-editor/author of

Voice Technology in Healthcare (2020) and *Blockchain in Healthcare* (2019) as part of the HIMSS Emerging Technology Series, *Blockchain Enabled Applications* (2018), *Connected Health* (2017), *HIMSS mHealth Innovation* (2014), and the HIMSS best-seller *mHealth: From Smartphones to Smart Systems* (2012).

Max Hooper is the chief executive officer of Merging Traffic. He is responsible for the company's management and growth strategy, serving as the corporate liaison to the financial services industry and various capital formation groups. Prior to starting the company, he was co-founder of Equity Broadcasting Corporation (EBC), a media company that owned and operated more than 100 television stations across the United States. He was responsible for activities in the cable, satellite, investment banking, and technology industries, and during his tenure, EBC grew to become one of the top ten largest broadcasting companies in the country. He is a lifelong learner and has earned five doctorate degrees—PhD, DMin, PhD, ThD, and DMin—from a variety of institutions. Hooper studied financial technology with cohorts at MIT, and cryptocurrency and business disruption with cohorts at the London School of Economics. As an avid runner, he has completed more than 100 marathons and an additional twenty ultra-marathons, which are 50- or 100-mile runs. He has completed the Grand Slam of Ultra Running. He is committed to his family and is a husband, father to five children, and grandfather to seven grandsons. He is active in many organizations and serves on various boards of directors. He works globally with several ministries and nonprofit aid groups and was honored to speak at the United Nations in New York in 2015.

About the Technical Reviewer

Prasanth Sahoo is a thought leader, an adjunct professor, a technical speaker, and a full-time practitioner in blockchain, cloud, and scrum working for Tata Consultancy Services. He has worked on various cloud platforms, including Azure and Google Cloud, and also led cross-functional teams to achieve goals using agile methodologies. He is passionate about driving digital technology initiatives by handling various community initiatives through coaching, mentoring, and grooming techniques.

He is a Certified Blockchain Expert and Certified Admin from Microsoft and is a working group member in the Blockchain Council, CryptoCurrency Certification Consortium, Scrum Alliance, Scrum Organization, and International Institute of Business Analysis. He has also received accolades for his presentation at the China International Industry Big Data Expo 2018, run by the Chinese government.

Acknowledgments

The authors would like to acknowledge our editors, Shrikant Vishwakarma and Spandana Chatterjee, for their help and guidance throughout the writing process.

The figures throughout this book were made with the help of Lucidchart. All figures from external sources were used with permission.

In addition, we wish to acknowledge the authorship and stellar contributions from the following authors toward this book:

Chapter 6: Colin Forward

Chapter 13: John Bass

Chapter 15: Katherine Kuzmeskas

Chapter 16: Heather Flannery, Jonathon Passerat-Palmbah, and Sean T. Manion

Chapter 17: Tory Ceraj

Without their resilience and tireless hard work, our book would be less complete.

Introduction

Blockchain technology is poised to fundamentally change our online world. Bitcoin was the first implementation of the blockchain; however, it has ushered in a fundamental shift for the offline world by allowing the transfer of value across the World Wide Web without the need for a centralized authority. Digitization and democratization of trust via the blockchain is enabling a new class of applications and companies to grow. Marc Andreessen famously authored a piece in the *Wall Street Journal* describing the theme of "software eating the world," where numerous industries are undergoing a rapid transformation after being consumed by software. To that end, we hope that our book provides an outlook to the world as blockchain actively transforms enterprises and creates entirely new verticals.

The fundamental shift that blockchain technology represents is a method for moving away from having a central trusted authority in a massively distributed network. Instead, it allows for having multiple sources of trust that must all agree, based on an algorithm, that this transaction can be trusted as valid. Furthermore, most blockchain solutions offer an immutable and enduring record of a transaction that is hard for any source, trusted or not, to change or modify. This presents a completely new level of security, privacy, and trust to our online world. As you will see throughout this book, a variety of uses, protocols, and standards make up the current blockchain ecosystem.

We also strive to strike the perfect balance between being a technical reference and a how-to handbook that shows practical examples of both current- and future-state use cases. While not comprehensive, we do select several high-promise areas where blockchain technology is beginning to enable applications for entirely new industry segments. We hope this book will inform you and provide a roadmap to your success in leveraging blockchain technology to enable new applications for your business.

Throughout the book, you will see many examples of applications to reinforce key points. Early examples extend beyond financial transactions to cover other aspects of FinTech, RegTech (regulation), InsuranceTech, GovTech (eVoting, licensing, records, and certification), HealthTech, and many others.

In order to understand these early examples, it is necessary to explore blockchain's history; fundamentals of distributed trust; consensus; hardware; software; and encryption in the early chapters. Next, you'll learn about the network transactions and simplified payments in blockchain fundamentals. We'll compare this with the extended capabilities of Ethereum and specific characteristics like how gas works and DApps, along with examples of Blockchain-as-a-Service. To further extend these capabilities, two chapters are devoted to DAO/Decentralized Organizations and the details and examples in these areas. In Chapter 7, Ethereum tokens are highlighted for value creation, with various technology and business-sector examples that highlight the power of smart contracts to allow multiple sources of value and rules to be directly embedded in the transactions. The next three chapters—8, 9, and 10—provide updates on blockchain in science and blockchain in healthcare, and details on the structure of the Hyperledger Project, respectively. Chapter 11 focuses on many recent developments, such as the EOS blockchain, Enterprise Ethereum Alliance, Quorum, R3, and Corda. Chapter 12 focuses particularly on ICOs and their effect on financial markets and processes. The next chapter is a conversation between the authors and John Bass from Hashed Health on building a healthcare consortium. Chapter 14 highlights the new cloud-blockchain computing landscape with an emphasis on Blockchain-as-a-Service. The following four chapters—15, 16, 17, and 18—are interviews with prominent figures from the blockchain world discussing the transforming roles of blockchain in education, artificial intelligence, machine learning, and quantum simulations.

Presently, during the COVID-19 pandemic, blockchain-based applications are being repurposed to help with contract tracing, developing "health passports" for the general public, and tracking physician burnout. In the near future, supply-chain applications built on the blockchain could assist in maintaining personal protective equipment (PPE) inventory and even therapeutics such as vaccines. We hope you find the information in this book useful as well as enjoyable as you explore the fundamentals, current best practices, and future potential of blockchain-enabled applications. We welcome your feedback at `info@metil.org`.

CHAPTER 1

Behold the Dreamers

Anxiety is perhaps the best way to describe the attitude that dominated the minds of investors and the general public concerning the financial markets toward the end of 2008. The 2008 financial crisis is considered by numerous economists to have been the worst financial crisis since the Great Depression. The years leading up to the crisis saw a flood of irresponsible mortgage lending and a massive systemic failure of financial regulation and supervision. The fallout was so immense that it threatened the collapse of large financial institutions, and national governments had to intercede to bail out the major banks. In this chapter, we will begin our discussion with an overview of the 2008 financial crisis and its aftermath: an environment where a new banking system and an alternative currency such as Bitcoin could thrive. Then, we will dive into the technology stack that powers Bitcoin. Remarkably, the components of this stack are not completely new, but have been integrated in a very intricate design to build a new system. Finally, we will end the discussion by talking about the heightened interest in blockchain, a major technical breakthrough that has the potential to revolutionize several industries. Imbolo Mbue wrote a book (Random House, 2017), which has the same name as this chapter, and tells the story of "dreamers" in New York City going through the financial crisis, and how their lives had changed as a result. This book chronicles the dreamers who envisioned building a more resilient financial system.

V. Dhillon et al., *Blockchain Enabled Applications*, https://doi.org/10.1007/978-1-4842-6534-5_1

Paradigm Shift

Revolutions often look chaotic, but this one was brewing quietly, headed by an unknown individual(s) under the name Satoshi Nakamoto who dreamt of changing the financial world. Any number of parties can be blamed for the financial crisis; however, the common denominator was that fundamental financial and accounting instruments used to maintain the integrity of the entire system became too complex to be used efficiently. Trust, the ultimate adhesive of all financial systems, began to disappear in 2008. The regulations have since changed to not allow similar circumstances to arise; however, it was clear that there was a dire need for auto-regulation of trust between counterparties and transparency into their ability to enter into any type of sales contract. A **counterparty** is essentially the other party in a financial transaction. In other words, it's the buyer matched to a seller. In financial transactions, one of the many risks involved is called **counterparty risk**—the risk that the *other* party involved in a contract may not be able to fulfill its side of the agreement. The systemic failure referenced earlier can now be understood in terms of counterparty risk: both parties in the transaction were accumulating massive counterparty risk, and in the end, both parties collapsed under the terms of the contract. Imagine a similar transaction scenario involving multiple parties, and now imagine that every single player in this scenario is a major financial institution, a bank or an insurance company that further holds millions of customers. This is what happened during the 2008 crisis.

The next issue we need to discuss is that of **double spending**. We will revisit this topic again strictly in the context of Bitcoin, but let's get a basic understanding of the concept by applying it to the financial crisis. The principle behind double spending is that resources committed to one transaction cannot be simultaneously allocated to a second disparate transaction. This concept has obvious implications for digital currencies; however, it can also summarize the central set of problems during the 2008 crisis.

Here's how it started: Loans (in the form of mortgages) were given out to borrowers with poor credit histories, who struggled to repay them. These high-risk mortgages were sold to financial experts at the big banks, who packaged them into low-risk public stocks by putting large numbers of them together in pools. This type of pooling would work when the risks associated with each loan (mortgage) are not correlated. The experts at big banks hypothesized that property values in different cities across the country would change independently, and therefore pooling was not risky. This proved to be a massive mistake. The pooled mortgage packages were then used to purchase a type of stock

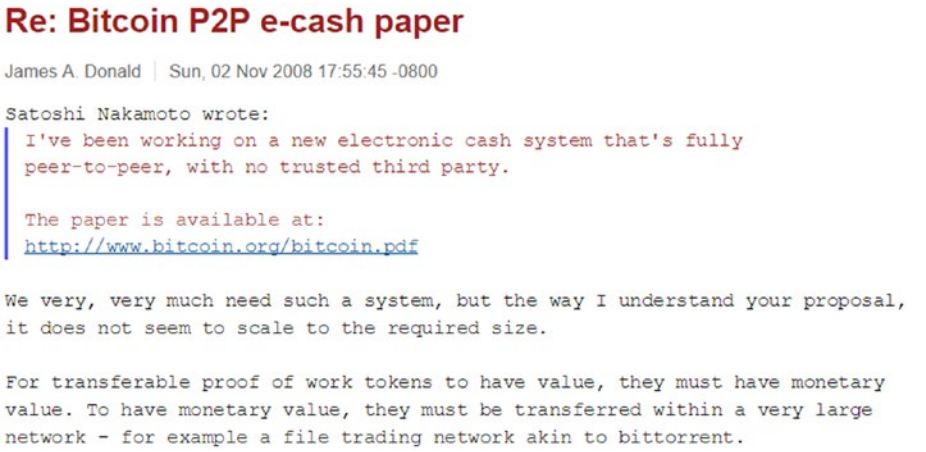
called collateralized debt obligations (CDOs). The CDOs were divided into tiers and sold to investors. The tiers were ranked and rated by financial standards agencies, and investors bought the safest tiers based on those ratings. Once the housing market in the United States turned, it set off a domino effect, destroying everything in the way. The CDOs turned out to be worthless, despite the ratings. The pooled mortgages collapsed in value, and all the packages being sold around instantly vaporized. Throughout this complex string of transactions, every sale increased the risk and incurred double spending at multiple levels. Eventually, the system equilibrated, only to find massive gaps, and collapsed under the weight. Following is a brief history of 2008. This timeline was made following a presentation by Micah Winkelspech at Distributed Health, 2016:

- January 11: Bank of America buys the struggling Countrywide
- March 16: Fed forces the sale of Bear Stearns
- September 15: Lehman Brothers files for Chapter 11 bankruptcy
- September 16: Fed bails out American International Group (AIG) for $85B
- September 25: Washington Mutual fails
- September 29: Financial markets crash, Dow Jones Industrial Average falls 777.68 points, and the whole system on the brink of collapse
- October 3: US government authorizes $700B for bank bailouts

The bailout had massive economic consequences, but more important, it created the type of environment that would allow Bitcoin to flourish. In November 2008, a whitepaper (`https://bitcoin.org/bitcoin.pdf`) was posted on the Cryptography and Cryptography Policy Mailing List titled "Bitcoin: A Peer-to-Peer Electronic Cash System," with a single author named Satoshi Nakamoto. This whitepaper detailed the Bitcoin protocol, and along with it came the original code for early versions of Bitcoin. In some manner, this whitepaper was a response to the economic crash that had just happened, but it would be some time before this technological revolution caught on. Some developers were concerned with this electronic cash system failing before it could ever take hold, and their concern was scalability, as we can see pointed out in Figure 1-1.

So, who is Nakamoto? And what is his background? The short and simple answer is that we don't know. In fact, it is presumptuous to assume that he is actually a "he." The name Satoshi Nakamoto was largely used as a pseudonym, and "he" could have been a

"she" or even a large group. Several reporters and news outlets have dedicated time and energy in digital forensics to narrow down candidates and find out the real Satoshi, but all the efforts so far have been wild-goose chases (https://www.technologyreview.com/s/527051/the-man-who-really-built-bitcoin/). In this case, the community is starting to realize that maybe it doesn't matter who Satoshi is, as the nature of open source almost makes it irrelevant. Jeff Garzik, one of the most respected developers in the Bitcoin community, described it as follows, "Satoshi published an open-source system for the purpose that you didn't have to know who he was, and trust who he was, or care about his knowledge." The true spirit of open source makes it so that the code speaks for itself, without any intervention from the creator/developer.



Re: Bitcoin P2P e-cash paper

James A. Donald | Sun, 02 Nov 2008 17:55:45 -0800

Satoshi Nakamoto wrote:

> I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.
>
> The paper is available at:
> http://www.bitcoin.org/bitcoin.pdf

We very, very much need such a system, but the way I understand your proposal, it does not seem to scale to the required size.

For transferable proof of work tokens to have value, they must have monetary value. To have monetary value, they must be transferred within a very large network - for example a file trading network akin to bittorrent.

Figure 1-1. *Initial reception of the Bitcoin protocol. Concerns about scalibility and realistic prospects of Bitcoin*

Technology Stack

Satoshi's real genius in creating the Bitcoin protocol was solving the Byzantine Generals Problem. The solution was generalized to financial transactions with components and ideas borrowed from the cyberpunk community. We will briefly talk about three of those ideas, how the components work, and how they help the Bitcoin protocol: Hashcash for proof-of-work, Byzantine fault tolerance for the decentralized network, and the blockchain for removing the need for centralized trust or a central authority. Let's dive into each one, starting with Hashcash.

Hashcash was devised by Adam Black in the late nineties to limit email spam with the first of its kind proof-of-work (PoW) algorithm. The rationale behind Hashcash was to attach some computational cost to sending emails. Spammers have a business model that relies on sending large numbers of emails with very little cost associated with each message. However, if there is even a small cost for each spam email sent, that cost multiplies over thousands of emails, and their business becomes unprofitable. Hashcash relies on the idea of cryptographic hash functions—a type of hash function (in the case of Bitcoin, it's SHA1) that takes an input and converts it into a string and generates a message digest, as shown in Figure 1-2. The hash functions are designed to have a property called one-way functions, which implies that a potential input can be verified very easily through the hash function to match the digest, but reproducing the input from the digest is infeasible. The only possible method of recreating the input is by using brute force to find the appropriate string of input. In practice, this is the computationally intensive element of Hashcash and has been imported into Bitcoin. This principle has become the foundation behind the proof-of-work (PoW) algorithms powering Bitcoin today, and most cryptocurrencies. The PoW for Bitcoin is more complex and involves new components that we will talk about at length in a later chapter.

Figure 1-2. *Mechanism of a cryptographic hash function. It takes an input and consistently converts it to a string of an output digest*

The next idea we need to discuss is the Byzantine Generals Problem. It is an agreement problem between a group of generals, with each one commanding a portion of the Byzantine army, ready to attack a city. These generals need to formulate a strategy for attacking the city and communicate it to each other adequately. The important task is that every general must work toward the same action, as a tepid attack by a few generals would be worse than a coordinated attack or a coordinated retreat. The crux of the problem is that some of the generals are traitorous. They may cast a vote to deceive the other generals and ultimately lead to a suboptimal strategy. Let's take a look at an example: In a case of odd-numbered generals, say seven, three support attacking and three support retreat. The

seventh general might communicate an agreement to the generals in favor of retreat, and an agreement to attack to the other generals, causing the whole arrangement to fall apart. The attacking forces would fail to capture the city because no intrinsic central authority could verify the presence of trust among all seven generals.

In this scenario, Byzantine fault tolerance can be achieved if all the loyal generals can communicate effectively to have an indisputable agreement on their strategy. If so, the misleading (faulty) vote by the traitorous general would be revealed and would fail to perturb the system as a whole. In the Bitcoin protocol, Satoshi's key innovation in enabling Byzantine fault tolerance was to create a peer-to-peer network with a ledger that could record and verify a majority approval, thereby revealing any false (traitorous) transactions. This ledger provides a consistent means of communication and further allows for the removal of trust from the whole system. The ledger is also known as the blockchain. With blockchain attached, Bitcoin became the first digital currency to solve the double-spending problem network-wide. In the remainder of this chapter, we will present a broad overview of the technology, and of the concept of a blockchain-enabled application.

A blockchain is primarily a recording ledger that provides all involved parties with a secure and synchronized record of transactions from start to end. A blockchain can record hundreds of transactions very rapidly, and has several cryptographic measures intrinsic to its design for data security, consistency, and validation. Similar transactions on the blockchain are pooled together into a functional unit called a **block** and then sealed with a timestamp (a cryptographic fingerprint) that links the current block to the one preceding it. This creates an irreversible and tamper-evident string of blocks connected together by timestamps, conveniently called a blockchain. The architecture of blockchain is such that every transaction is very rapidly verified by all members of the network. Members also contain an up-to-date copy of the blockchain locally, which allows for consensus to be reached within the decentralized network. Features such as immutable record-keeping and network-wide consensus can be integrated into a stack to develop new types of applications called decentralized apps (DApps). Let's look at a prototype of a DApp in Figure 1-3, in the context of the Model-View-Controller (MVC) framework. The Model-View-Controller framework is a software design concept that separates an application into three components: The model (that contains the data-related logic), the view (UI component that customers often interface with), and the controller (an interface that interacts between the model and the view). This framework is frequently used to design traditional web applications that are extensible and scalable. Here, we want to extend an industry standard and use it to explain a DApp.

Note The first block of the blockchain is called the Genesis block. This block is unique in that it does not link to any blocks preceding it. Satoshi added a bit of historical information to this block as context for the current financial environment in the United Kingdom, "*The Times 03/Jan/2009 Chancellor on brink of second bailout for banks*." This block not only proves that no Bitcoins existed before January 3, 2009, but also gives a little insight into the mind of the creators.

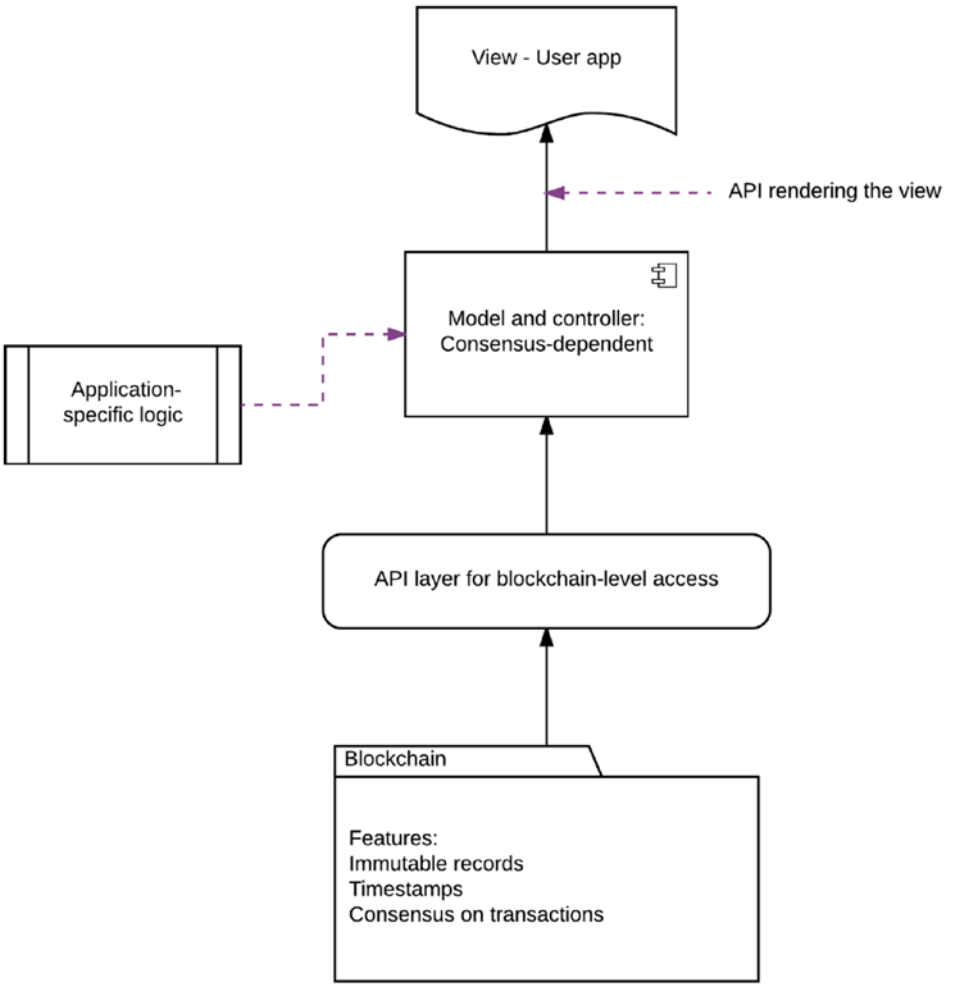


Figure 1-3. *This figure presents a simple prototype of a decentralized application that interacts with the end user at the final steps*

The model and controller here rely on the blockchain for data (data integrity and security) and accordingly update the view for the end user. The secret sauce in this prototype is the API, which works to pull information from the blockchain and provides it to the model and controller. This API provides opportunities to extend business logic and add it to the blockchain, along with basic operations that take blocks as input and provide answers to binary questions. The blockchain may eventually have more features such as oracles that can verify external data and timestamp it on the blockchain itself. To better understand the concept of blockchain-enabled applications, we have to appreciate the full stack of services that could power an end-user application; this is demonstrated in Figure 1-4.

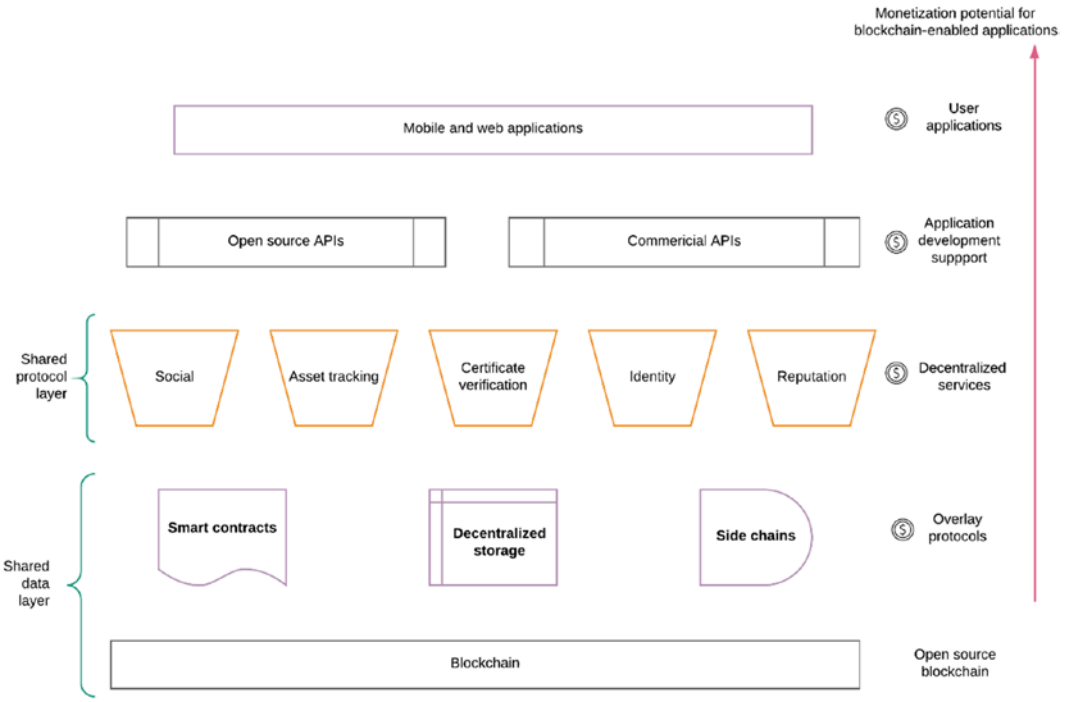


Figure 1-4. *The blockchain-enabled application stack*

Summary

In this chapter, we started talking about the history of Bitcoin and the financial environment around the time it came to exist. We will continue our discussion of blockchain and specific features of the peer-to-peer network, such as miners and more, in the upcoming chapters.

CHAPTER 2

The Gold Rush: Mining Bitcoin

Mining is a key operational concept in understanding how the Bitcoin protocol functions. It refers to a decentralized review process performed on every block of the blockchain to reach consensus, without the need for a central authority to provide trust. In other words, mining is the computational equivalent of peer review in a decentralized environment where neither party involved trusts the other. We will continue our discussion of the hash function here in more depth, as it refers to mining and solving proof-of-work functions. Then, we will integrate the concepts of block target values and network difficulty with mining and how mining has evolved to keep up with the increasing network difficulty. This will lead us further into talking about the different types of hardware mining that have been developed recently. We will end the chapter with an analysis of startups that began selling dedicated hardware for mining, leading to the Bitcoin mining arms race and the startups' eventual failure.

Reaching Consensus

Mining is central to the Bitcoin protocol and has two primary motivations: add new Bitcoins to the overall economy and verify transactions. In this chapter, we will look at the mechanisms behind these two processes. Essentially, mining is the appropriate solution to the double-spending problem that we discussed previously. To remove the need for a central authority, individuals running the Bitcoin client on their own machines (called miners) participate in the network and verify that transactions taking place between two parties are not fraudulent. Mining is actually a computationally intensive activity, but what incentive does anyone have to help mine for new Bitcoins? The key incentive for miners is getting a reward in the form of Bitcoins for their participation. Let's look at a simplified view of the mining process in Figure 2-1.

V. Dhillon et al., *Blockchain Enabled Applications*, https://doi.org/10.1007/978-1-4842-6534-5_2

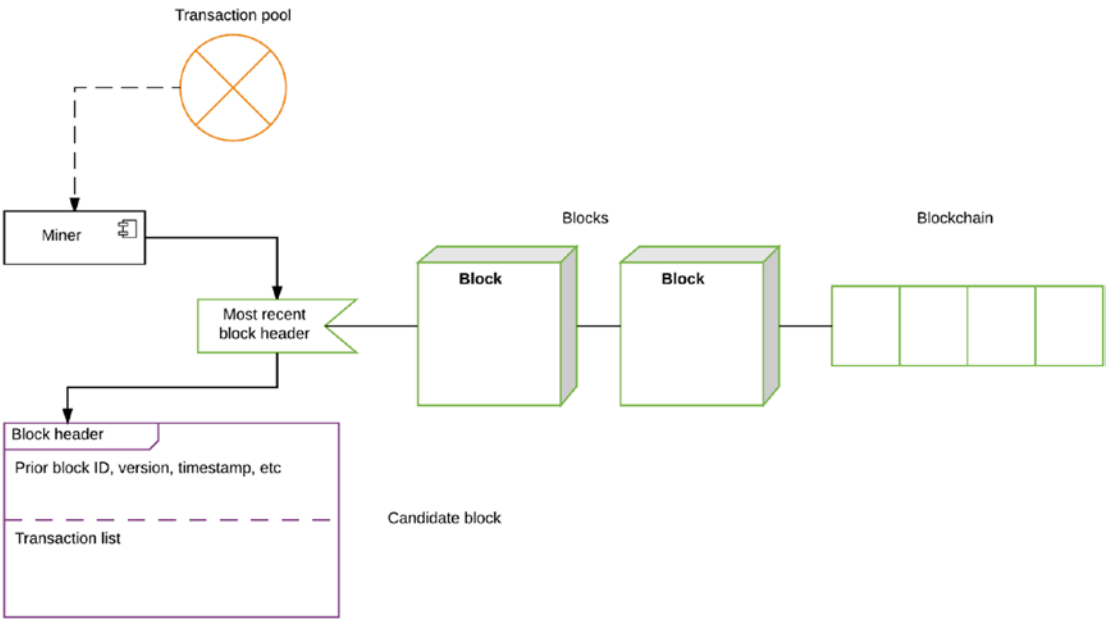


Figure 2-1. *A simplified overview of the mining process*

Unpackaged transactions that have occurred recently on the Bitcoin network remain in the transaction pool (also known as the mempool, where all valid transactions wait to be confirmed by the Bitcoin network) until they are picked up by a miner to be packaged into a block. A miner selects transactions from the transaction pool and packages them in a block. After the block has been created, it needs a header before it can be accepted by the blockchain. Think of *this* as shipping a package: once the package has been created, it needs to be stamped so that it can be shipped. A miner uses the header of the most recent block in the blockchain to construct a new header for *this* current block. The block header also contains other elements such as a timestamp, version of the Bitcoin client, and an ID corresponding to the previous block in the chain. The resulting block is called a candidate block, and it can now be added to the blockchain if a few other conditions are satisfied.

The process of mining is very involved, and Figure 2-1 only served to paint a broad picture regarding the participation of miners in the protocol. Next, we will explore the technical aspects of the stamp (in the analogy just referenced) and the mechanism of stamping a package. Keep in mind that mining is a competitive process. Figure 2-1 only describes this process for one miner, but in reality, a very large number of miners from the network participate simultaneously. The miners compete with each other to find a stamp for the package (block) that they created, and the first miner to discover the stamp

wins. The race between miners to find a stamp is concluded within ten minutes, and a new race begins in the next ten minutes. Once the stamp is discovered, the miner can complete the block and announce it to the network. Now the block can be added to the blockchain. Let's take a look at the process behind searching for the stamp, better known as a block header, in Figure 2-2.

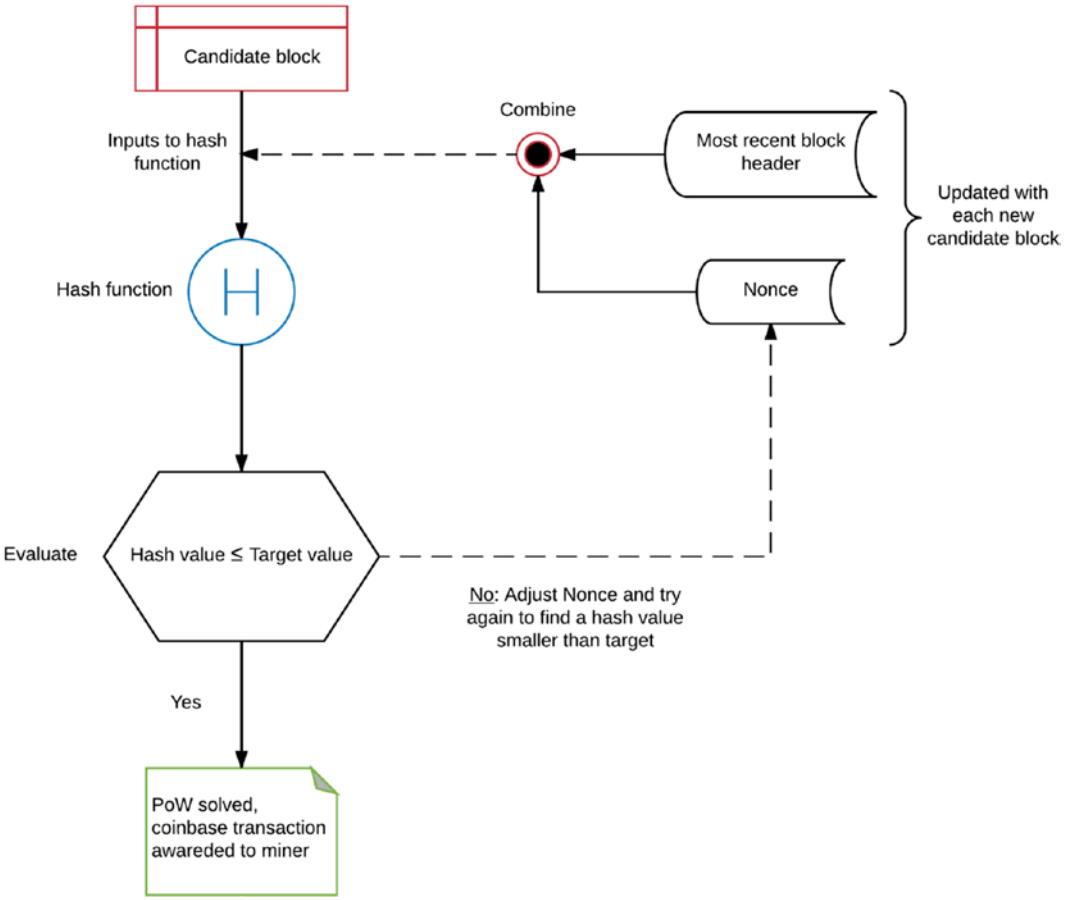


Figure 2-2. *Generating a block header by solving proof-of-work (PoW)*

The package created by a miner is almost a block, but it is missing a header. It's called a candidate block and can only be added to the blockchain after the stamp, or the header, is added. The header from the most recent block in the blockchain is retrived and combined with a 32-bit value called nonce. This combination is directed to the hash function (SHA-256) as an input. The hash function computes a new resulting hash as an output. This generated hash is then compared to the target value of the network (at the given time).

If the hash value is larger than the target value, then the nonce is readjusted and a new input is sent to the hash function to obtain a new potential output. The problem of finding the appropriate hash value that is smaller than the target value is at the heart of PoW, and it can only be solved using brute force. Once a hash value smaller than the target value is discovered by a miner, this hash can now be used in the block header for the candidate block. The first miner to discover the hash is considered the winner. The winning miner has shown proof of work that she did to discover the hash; therefore, the transactions contained within the block are now considered valid. This block can now be added to the blockchain. Additionally, the winning miner also wins a reward for solving the PoW problem, which is a certain number of Bitcoins. This whole process from packaging transactions into a block, to finding the hash and announcing the block to the Bitcoin network, repeats itself approximately every ten minutes.

We introduced some new terminology in Figure 2-2; let's describe them here properly for the sake of completion:

- **Candidate block:** An incomplete block, created as a temporary construct by a miner to store transactions from the transaction pool. It becomes a complete block after the header is completed by solving the proof-of-work problem.
- **PoW:** The problem of discovering a new hash that can be used in the block header of the candidate block. A computationally intensive process that involves evaluating a hash taken from the most recent block and appending a nonce to it against the target value of the network. This problem can only be solved using brute force; i.e., multiple trials of using the hash (from most recent block header) and nonce's being adjusted each time are necessary to solve the PoW problem.
- **Nonce:** A 32-bit value that is concatenated to the hash from the most recent block header. This value is continuously updated and adjusted for each trial, till a new hash below target value is discovered.
- **Hash function:** A function used to compute a hash. In the Bitcoin protocol, this function is the SHA-256.
- **Hash value:** The resulting hash output from a hash function.

- **Target value:** A 256-bit number that all Bitcoin clients share. It is determined by the difficulty, which will be discussed shortly.
- **Coinbase transaction:** The first transaction that is packaged into a block. This is a reward for the miner to mine the PoW solution for the candidate block.
- **Block header:** The header of a block, which contains many features such as a timestamp, PoW, and more. We will describe the block header in more detail in the following chapter.

Note After going over the terms defined, revisit Figures 2-1 and 2-2. Some concepts that were abstracted out will become clear now, and the information will integrate better.

Now that we have a better idea of how mining works, let's take a look at mining difficulty and target values. These two concepts are similar to knobs that are adjusted over the course of time for the network, and all Bitcoin clients get updated to follow the latest values. So, what is mining difficulty? Essentially, it can be defined as the difficulty of finding a hash below the target value as a miner is solving the proof-of-work problem. An increase in difficulty corresponds to a longer time needed to discover the hash and solve the PoW, also known as mining time. The ideal mining time is set by the network to be approximately ten minutes, which implies that a new block is announced on the network every ten minutes. The mining time is dependent on three factors: the target value, number of miners in the network, and mining difficulty. Let's look at how these factors are interconnected:

1. An increase in mining difficulty causes a decrease in the target value to compensate for the mining time
2. An increase in the number of miners joining the network causes an increase in the rate at which PoW is solved, decreasing the mining time. To adjust for this, mining difficulty increases and the block-creation rate returns to normal.
3. The target value is recalculated and adjusted every 2016 blocks created, which happens in approximately two weeks.

As we can see, there is a common theme of self-correction in the Bitcoin network that allows it to be very resilient. Miners are the heartbeat of the Bitcoin network, and they have two main incentives for participation:

- The first transaction to be packaged in a block is called the coinbase transaction. This transaction is the reward that the winning miner receives after mining the block and announcing it on the network.
- The second reward comes in the form a fee charged to the users of the network for sending transactions. The fee is given to the miners for including the transactions in a block. This fee can also be considered a miner's income because as more and more Bitcoins are mined, this fee will become a significant portion of their income.

Now we can put these concepts together in the form of another flowchart in Figure 2-3. This will help solidify the process of mining in the context of difficulty and target values.

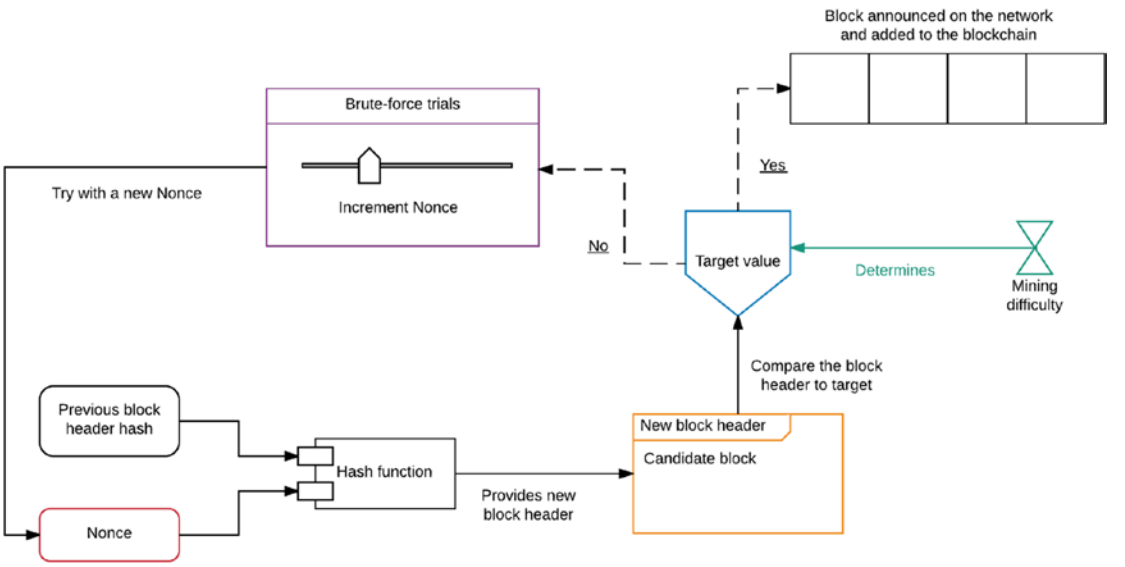


Figure 2-3. *Solving the PoW problem*

Miners across the network compete to solve the problem, and the winning miner announces the block to the network, which then gets incorporated in the blockchain. To solve the PoW, a miner has to keep generating new hash values (through the hash function) using the incremented nonce until a hash below the target value is discovered. In this case, notice that the nonce is the only adjustable value. This is a simplified PoW scheme, and there are small differences in its implementation versus reality.