Case Study #2

Becoming a Blockchain Developer

Sanchit Singhal

Fall 2018 – November 23, 2018

INF385T – Introduction to Blockchain w/ Lance Hayden

School of Information

As blockchain technology continues to permeate itself into all sorts of different use cases, it is clear that the “era of the blockchain” is still in its infancy. [8] More than 80,000 projects that use blockchain technology have been launched since 2015. However out of these, only 8% are thought to be active today. Research shows that the average lifespan of a blockchain project is roughly 1.22 years – well below the cut-off of what can be deemed as a success. Entrepreneurs in the space seem too keen to launch start-ups – never short of an idea – but more often than not, their attempts die out quickly and the projects fail to achieve any real-life application. [8] Investors have reportedly spent more than $ 1 billion dollars funding blockchain related missions, but it is still largely unclear how much of this amount has been truly recouped through business value gained. There is certainly a case to be made that many of these new endeavors are in fact fraudulent and merely an orchestrated attempt at hoarding investment with no real plan of delivering any product. While we, as a society, continue to work through setting up appropriate regulations to counter this, not all failed projects are scams of course. Many of the business plans generated through blockchain incubation are both game changing and viable in their respective industries however fail simply due to a lack of technical know-how of the implementation. As is usually the case, the devil is in the details – while it takes leaders and business acumen to plan, pitch, and push new product offerings, these individuals typically do not have the skillset to figure out exactly how the blockchain will be architected and coded. The “give me money, and I’ll figure it out” approach, which initially flooded the market, is quickly losing popularity as investors are becoming more carefully with their assets. While it is evident that blockchain has the potential to disrupt, it is prudent that we groom more blockchain developers that can actually take up the challenge of learning, practicing, coding, and experimenting with the technical side of the technology. This case study will explore various areas that such aspiring developers can start their journey in, the skill set that is recommended for them, and provide a basic framework that can help guide their ambition of getting involved in the blockchain era.

The first step for any technical individual looking to develop on the blockchain would be to familiarize themselves with the existing system. This can be done through a wide variety of mediums such as university classes, online videos, or even just reading articles. It important to understand, not only the terminology involved, but also the higher level concepts such as peer-to-peer networks, consensus mechanisms, distributed ledgers, and hashed Merkle trees. Getting caught up with some of the philosophies of the original whitepapers can also be a meaningful step towards fostering an understanding of the culture and ethics of the blockchain. Once a solid foundation has been set, one is well-versed to at least beginning investigating more advanced subject matters that they are interested in. There are communities of practice around topics such as cryptography or Web 3.0 that one can get involved with. By being able to interact and connect with other entities, developers can join groups that shared similar visions or collaborate on projects that enhance their understanding of the tools and techniques. It is also highly endorsed that one should learn how the process works from the user’s point of view. If one is interested in cryptocurrency, then setting up a wallet and purchasing a few coins might be a useful exercise to get acquainted with. Coinbase is an exchange that allows for a rather simple interface to do this – although it is strongly discouraged to not store too much value in such services because your private key is essentially on their servers and any hacking can lead to a loss of your portfolio. Exploring cold wallets, in which the key is kept offline, or multi-sig wallets, in which more than one key is required for authorization, are another option for those looking to really understanding wallet best practices. Once the developer is confident in their knowledge of the existing ecosystem, they are finally ready to start preparing for learning how to start coding on the blockchain.

As a blockchain developer, there are three major areas of software development that are essential to the success of the implementation and a budding developer should be comfortable with these. Security is one of the most fundamental elements of a public blockchain. Because the source code lies in an open forum, it is likely that there will be thousands, if not millions, of people scrutinizing each block of code looking for bugs and vulnerabilities to exploit. Any exposure to risk for users could tarnish the new platform and will most certainly kill the project immediately if not rectified immediately. [6] One such example is a hack on Ethereum, in which The DAO was manipulated through a bug in the code and more than 15% of all ether in circulation at the time was drained. The only way to recover the funds was to go back in time and perform a hard fork on the codebase – arguably irreversibly damaging the integrity of the blockchain. Performance and resource management is another key component that the developer should keep in mind when developing. The network should be able to keep up with demands and handle task requests in a manner that matches the appropriateness of the use case. The selected programming language should be also be robust enough to offer benefits of parallelism and perform multiple processes at the same time to enable efficient dispensation of modules. [1] An example of this can be a task such as a digital signature verification – which requires only the key, transaction, and signature data points– and can in theory be conducted in a parallelized manger. Though, each such operation must be carefully thought out: transactions probably shouldn’t happen in tandem to avoid errors like double spends. The level of parallelism needed can impact the choice of language – some are better than others at it – and therefore, the developer much be diligent in their selection. Isolation of certain code segments is another key feature that must be kept in mind: by separating transactions from non-deterministic elements, the developer must ensure the code performs in a deterministic manner and performs the same across all machines. While there are many programming languages that can match these criteria, most previously implementation of blockchain has been written in C++ (such as bitcoin), and Javascript. There are many custom libraries in JS that allow are ease of coding such s crypto-js and is a logical first language that should be learned.

A typical block contains the index, a timestamp, the data to be stored, and the hash of the previous block. An object is initialized within the class, converted into a JSON string and finally converted into a hash using the SHA256 function. When connecting these blocks into a chain – a genesis block function must be invoked as well: which obviously will not contain the hash to a previous block and therefore must be manually fed with data. When linking the blocks after the first one is created, each block must compare the previous hash value to the value in the new block to verify the legitimacy of the block. Usually the entire chain itself is verified at this point as well using a “for” loop from block 0, the genesis block, all the way unto the current block. Larger chains only verify a certain number of blocks as it is infeasible to check the entire chain all the time and does not provide much additional benefit after a certain threshold for the number of blocks as been reached – this can also be coded into the script. Such an implementation is a relatively rudimentary outline of a blockchain source code and fancier methods can be added for custom functionality if desired. These will most probably developed through an iterative testing software lifecycle. Familiarity with the various Javascript modules built specifically for creating blockchain can be a huge benefit for potential developers and reduce the time needed for development.

There is another quite popular way that developers are getting involved with blockchain: through developing smart contracts that run on existing infrastructure as opposed to developing new blockchains all together. [1] These smart contracts are a “protocol intended to facilitate, verify, or enforce” an agreement between users through computer automated code. When developing these programs, it is important to keep in mind some of the same principles as above such as determinism and avoid dynamic calls. But over and above these, it is vital to make sure the program is terminable. [1] In 1936, Alan Turing deduced, using Cantor’s Diagonal Problem, that there is no way to know whether a piece of software can finish in a certain time frame or not. Therefore, there are measures that developers have come up with to ensure software does not enter an infinite loop that ends up draining all available resources: most prominently setting a pre-determined timer before it is externally aborted or having a “step and fee meter” in which the number of steps is recorded and each has an associated cost that is deducted from its allocated budget. Perhaps the most important constituent of a smart contract is its sandbox in the ecosystem that it resides in and provides isolation such that a bug does not bring does the entire system. Ethereum for example uses Virtual Machines whereas more recent efforts such as Fabric utilize Docker. While Fabric primarily requires competence in Python, Ethereum, the most popular second-generation blockchain, uses Solidity. For anyone interesting in developing decentralized apps, Solidity is a required expertise. [1] Originally created by the core contributors of Ethereum, it is a slimmed down, loosely-typed version of ECMAScript. It uses a stack-and-memory model with a 32-byte word size and an EVM as its register space that allows for expandable temporary memory which feeds the permanent storage mechanism. Solidity developers specialize in maintaining DAPPs and one must become proficient if they want to start working with Web 3.0.

While, this case study examines some of the prerequisites that are generally agreed upon for today’s blockchain environment it is hard to approximate future tools that will be needed for technical individuals. A budding developer should be proactive in staying in the mix and constantly surf forums, Github pages, and Stack Exchange to remain up-to-date with new developments and fine-tune their skill sets based on market needs and personal interests. [4] For those more intrigued by mining the blocks themselves, learning more about setting up a node and joining pools that utilize explicitly designed hardware might also be of curiosity. Becoming a blockchain developer can be a lucrative prospect, but not only for the individual as a career path, but also for blockchain projects. With a shortage of supply, such initiatives are rapidly trying to grow their technical teams that can deliver tangible output and make good on the promises to their investors. It is an exciting time in the field as more and more developers are making the switch and hopefully this case study provides a supportive overview on how to go about it.

References

[1] Garber, M. (2017, March 01). Blockchain Tutorial | How To Become A Blockchain Developer:. Retrieved November 23, 2018, from https://blockgeeks.com/guides/blockchain-developer/

[2] Marco Conoscenti, Antonio Vetrò, Juan Carlos De Martin, "Blockchain for the Internet of Things: A systematic literature review", *Computer Systems and Applications (AICCSA) 2016 IEEE/ACS 13th International Conference of*, pp. 1-6, 2016.

[3] Melanie Swan, Blockchain: Blueprint for a New Economy, O'Reilly Media, Inc., 2015

[4] Mining?, W. I. (n.d.). Bitcoin Mining Guide. Retrieved November 23, 2018, from https://www.bitcoinmining.com/getting-started/

[5] N. Caes, R. Williams, E. H. Duggar, and M. R. Porta, "Robust, cost-effective applications key to unlocking blockchain's potential credit benefits," 2016.

[6] New Alchemy. (2018, February 08). A short history of smart contract hacks on Ethereum. Retrieved November 23, 2018, from https://medium.com/new-alchemy/a-short-history-of-smart-contract-hacks-on-ethereum-1a30020b5fd

[7] Shinsaku Kiyomoto, Mohammad Shahriar Rahman, Anirban Basu, "On blockchain-based anonymized dataset distribution platform", *Software Engineering Research Management and Applications (SERA) 2017 IEEE 15th International Conference on*, pp. 85-92, 2017.

[8] Team, I. (2018, September 05). How Blockchain May Impact Logistics, Supply Chain And Transportation: A Conversation With The Blockchain In Transport Alliance. Retrieved November 23, 2018, from https://www.forbes.com/sites/insights-penske/2018/09/04/how-blockchain-may-impact-logistics-supply-chain-and-transportation-a-conversation-with-the-blockchain-in-transport-alliance/#5bdff02ff2b3

[9] Unicredit, "Blockchain technology and applications from a financial perspective," 2016.