**Exploring Blockchain Ecosystems**

Final Research Paper

Team: Blockchain Ecosystems

Jamie Kerr, Joe Cucuzza

{kerrj, cucuzzaj}@wit.edu

Github repo: <https://github.com/Jaromeat/BlockchainEcosystems.git>

**Abstract**

We explored Ethereum, Bitcoin, and Tezos as smart contract ecosystems to determine their strengths and weaknesses based on popularity, features, development tools, and future. We then explain our experience designing and developing a contract for each ecosystem. Then we looked at two up and coming ecosystems to determine where the future of cryptocurrency is headed. We discovered that blockchain ecosystems are moving away from proof of work towards proof of stake and other alternatives to ensure blockchain ecosystems remain scalable, fast, and secure.

**Introduction**

Blockchain technology was invented for digital currency. It allows two or more users to securely transfer money to one another over servers. One of the most important aspects of blockchain is that there isn’t an online bank or banks in charge of money. Money can be transferred from anywhere at any time. All that is required is an internet connection. This concept is referred to as decentralization. Freedom from a centralized provider spurred the birth of Bitcoin, a digital currency reliant entirely on the merits of blockchain. However, Bitcoin is not alone. Currencies like Ethereum managed to stay ahead of the competition with the introduction of smart contracts, a framework for developing apps that run entirely on the blockchain, run by the blockchain miners at a cost. The introduction of smart contracts has led many cryptocurrencies to blossom into blockchain ecosystems, complex networks of interwoven contracts and decentralized applications (Dapps). The smart contract has become an important tool in expanding the use of cryptocurrencies, but not all ecosystems are created equal. Which ecosystem should a developer choose when designing a Dapp? We plan to define the benefits and downfalls of blockchain ecosystems of widely varying structure to inform development decisions and predict the future of blockchain ecosystems.

We have researched 5 cryptocurrencies: Ethereum, Bitcoin, Tezos, Cardano, and Diem (Libra) for comparison. For those that have fully implemented smart contracts, we designed a contract to facilitate the functionality of a trade board. The requirements of the trade board are listed:

* The seller must be able to add a trade request to the board​
* The buyer must be able to see requests and make offers​
* Secure transfer of funds must occur between the buyer and seller​

Following these requirements we designed and developed 3 smart contracts portraying the differences between smart contract scripts, smart contracts on a proof of work system, and smart contracts on a proof of stake system. To begin, let us explain consensus algorithms.

Blockchain ecosystems rely on the integrity of blockchain to build security and credibility. Changes to the blockchain can have sweeping effects on the overarching cryptocurrency and smart contract system. The most dramatic variation in blockchain makeup comes in the form of the consensus protocol. A consensus protocol ensures that the blockchain remains secure by requiring proof that a new block is valid, that the miner is allowed to add it to the chain, and that the network agrees on the make-up of the chain. Following is a description of several popular consensus protocols including their strengths and weaknesses starting with a naive solution, proof of authority.

**Proof of Authority**

Proof of Authority functions as the simplest consensus algorithm. It functions by allowing only authorized users, known as validators, to validate blocks added to the chain. All validators are rewarded equally for each block they add and are authorized to Validators are public figures that are encouraged to maintain security to both maintain their position and their reputation[1]. If a validator were to exploit the system, their position would be revoked and they would earn a bad reputation. Users can earn the right to become a validator in this system. Of the consensus protocols, proof of authority is the least secure. Position and reputation are potentially flimsy motivators, and nothing prevents validators from pooling to form a majority. A majority of validators can agree to change the blockchain however they please. This exploit is called a 51% attack[2]. Proof of Authority requires trust in its validators. For this reason, proof of authority is commonly used only on private ecosystems[1].

**Proof of Work**

Proof of work stands as the most popular consensus protocol today, and was the first consensus protocol implemented successfully. In proof of work, any miner can add a block to the chain as long as their block is valid and accepted by other miners. Proof of work ensures the validity of blocks in the chain by ensuring miners have performed the necessary amount of ‘work’ to create the block[3]. Blocks in a proof of work system require a hash of the previous block in the chain, all data stored in the block, a timestamp, and an arbitrary integer value known as the nonce. To prove a new block is valid in most proof of work systems, miners must hash the block in its entirety to get a hash value with a number of leading zeroes. Miners must hash the block repeatedly, modifying the nonce value, until they find the golden nonce, resulting in a hash value containing enough leading zeroes. This process is computationally expensive and ensures that miners have investment in the currency. The number of leading zeroes actively changes to ensure a consistent mining time. Proof of work is secure due to the intensive computational requirements to mining. A 51% attack on a proof of work protocol is not feasible due to the barrier of entry. However, proof of work inherently restricts traffic speeds[3]. This leads to slow transaction times which can affect the speed and design of a smart contract.

**Proof of Stake**

Proof of stake is the second most popular consensus protocol behind proof of work, and is being implemented into many younger cryptocurrencies. Proof of stake does away with hashing and instead raffles blocks out to its miners based on the amount of currency the miner or mining pool has. At a regulated time interval, miners are selected at random to add a block to receive the reward of a new block[4]. A miner with more currency is more likely to receive the reward. This means miners do not need to invest in computational assets, they are instead rewarded for remaining loyal to the currency. Proof of stake works on the philosophy that those with more currency, or stake, are less likely to exploit the currency and potentially ruin their stake[4]. This system provides far faster transaction times than proof of work, but is inherently at more risk to a 51% attack. Miners in a proof of stake system are encouraged to form pools to increase their earnings. Without regulation, nothing stops a pool from growing to the point it possesses more than 50 percent of the stake and thus can exploit the chain[4]. Regulation is very important for proof of stake to remain secure. Commonly, regulation comes in the form of capping the amount of stake a pool will be rewarded for, thus discouraging pools from growing larger than a predetermined size.

**Proof of Activity**

Proof of activity is far less popular than proof of work and proof of stake, but provides the most security. Proof of activity works as both proof of work and proof of stake. First, proof of work miners must show they have invested computationally by creating a block with a golden nonce, then the block is raffled off to proof of stake validators. Both the miners and the validators are rewarded for the block[5]. The inclusion of mining rewards reduces the concern for a 51% attack from proof of stake alone, but does not completely minimize it. Validation must happen both through two systems, so proof of activity is even slower than proof of work. For these shortcomings, it is unlikely proof of activity will not become popular in its current iteration[5].

**Ethereum**

Ethereum is widely popular for first implementing the smart contract, a set of instructions that can be written into a blockchain and tied to an address so parameters sent to the contract address will execute on a miner’s machine and potentially return a response for a fee[6]. The smart contract is revolutionary for increasing complexity and potential for blockchain ecosystems. For this reason, Ethereum and its coin, Ether, are second only to Bitcoin for market capitalization according to coinmarketcap.com. Ethereum utilizes a proof of work consensus algorithm resulting in only 15 transactions per second[6].

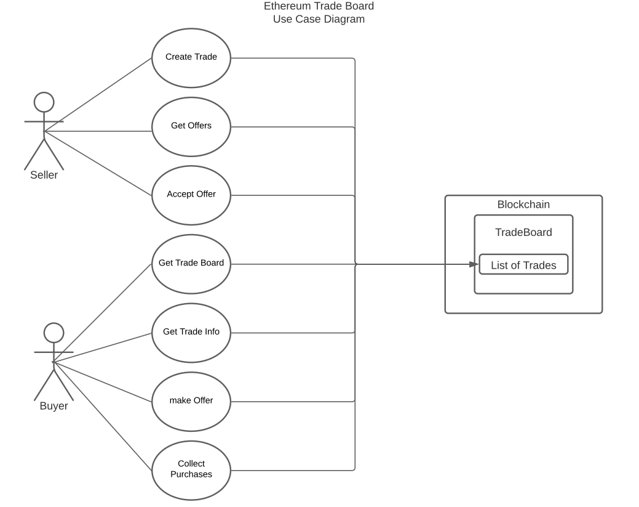


Figure 1. Use Case Diagram for Tradeboard Implementation in Ethereum.

Ethereum provides many tools for development. Code is written in Solidity, a Turing complete, functional language. Ethereum provides an in browser ide called Remix for writing and testing Solidity code. The Ethereum Virtual Machine allows for distributed applications (Dapps) to be developed that live on the blockchain. For further testing, Ganache CLI works as a local test network, and on public servers, there are several testnets available for deployment. Truffle can be used for deployment to either test nets or the main blockchain.

Design of a dapp for Ethereum is similar to design for any function language. Solidity is an easy language to pick up for those fluent in Java or Javascript. Listed structs allow for the tradeboard to exist entirely in the smart contract. A major design decision arose when considering transaction speed. Ethereum has a slow transaction throughput, so the contract must handle the event where a second buyer buys before the first buyer’s transaction completes. To handle this, as seen in Figure 1., the buyer can not purchase, they must make an offer. The seller must look at the listed offers and accept an offer. At this point, a buyer can collect purchases that their offers were accepted in. For this reason, the Trade struct must include the additional accepted boolean to denote the state in which the buyer has been accepted but has not yet collected the purchase. This is shown in Figure 2.

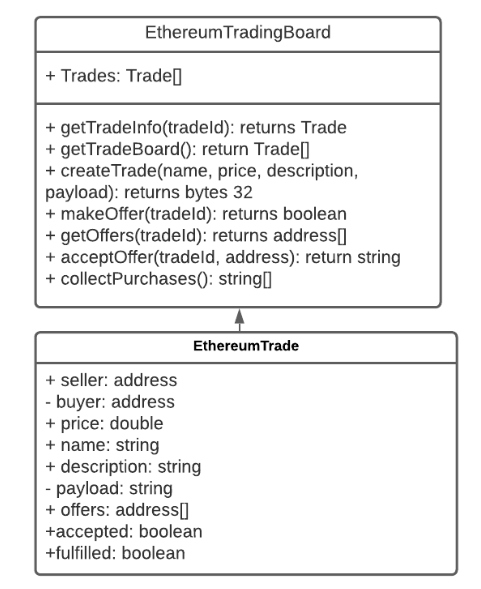


Figure 2. UML Class Diagram for a Tradeboard implementation in Ethereum

The development process of Ethereum is the best of its kind likely due to its age compared to other dapp and smart contract systems. This is likely to remain true until the introduction of so-called ‘gen 3’ cryptocurrencies arrive. Ethereum has some promising additions on the horizon. Ethereum 2.0 is Ethereum’s ‘gen 3’ upgrade and will bring with it secondary blockchain structures to grow alongside Ethereum known as sidechains[7]. One such sidechain is the Beacon chain, a sidechain that will utilize proof of stake to speed up transaction times and lower mining costs. The end goal of the Beacon chain is to merge into the main chain and end proof of work consensus for the entire ecosystem[7]. Ethereum is likely to remain popular for the foreseeable future as the best language for smart contracts and dapps.

**Bitcoin**

The first cryptocurrency, Bitcoin possesses the highest market capitalization of all cryptocurrencies according to coinmarketcap.com. Its focus lies on creating a fully digital currency that will be used for anonymously transferring money between two users. Bitcoin allows its users to handle currency without going through banks in a secure manner[8]. Bitcoin functions on a proof of work consensus algorithm. This fact alongside slow security and mining measures means Bitcoin’s transaction throughput is a sluggish 4.5 transactions per second[9].

Development for Bitcoin is heavily restricted by Bitcoin’s lowest level language, Bitcoin Script. Bitcoin Script is a Turing incomplete language which significantly hinders or prevents the development of complex smart contracts. Above Bitcoin Script, MiniScript and finally Min.sc can be run, Min.sc being the highest tiered language. Restrictions on Bitcoin Script apply to Min.sc, but Min.sc’s libraries make functions, time, and gas preferences easily accessible. Smart contracts for Bitcoin are restricted to simple scripts, and without a virtual environment, dapps are impossible to develop. A Tradeboard implementation on Bitcoin can be shown in Figure . The board and all trade interactions must be stored externally, and the Min.sc code only handles simple transactions. For the Tradeboard, two scripts were deemed most useful, Authenticated Transaction and Lock Funds. Authenticated Transaction requires 3 way verification between the buyer seller and tradeboard itself to perform the Bitcoin transfer. Lock Funds will take any funds transferred and hold them until either the buyer enters a secret key, or a certain amount of time has passed and the seller refunds the money. The website Min.sc acts as a Min.sc compiler into a contract address, policy, Bitcoin Script, and MiniScript code. Deployment is not as complex as a Dapp focused system. Simply send the code with policy and Bitcoin Script to the contract address, and the contract is live. This can be done via JavaScript via the minsc and bitcore packages to create a transaction. For testing of scripts, Bitcoin has a public Testnet.

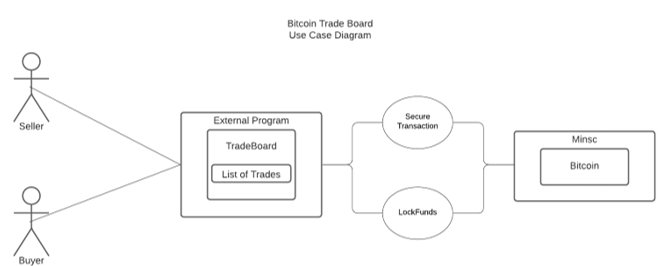


Figure 3. Use Case Diagram for Bitcoin Implementation utilizing Min.sc

Bitcoin’s future is uncertain due to a myriad of scalability and usability concerns. Bitcoin is not competitive with its competition for smart contract development, as the system was not designed with complex smart contracts in mind. Bitcoin also faces serious issues of scalability. Proof of work has a shelf life as mining costs overtake mining profits[10]. Bitcoin is the slowest major crypto on the market. The only leg Bitcoin is standing on is its popularity and price, which far exceed its rivals, but this advantage will not last. Unless Bitcoin can make sweeping changes to the ecosystem’s architecture, Bitcoin is not the coin of the future.

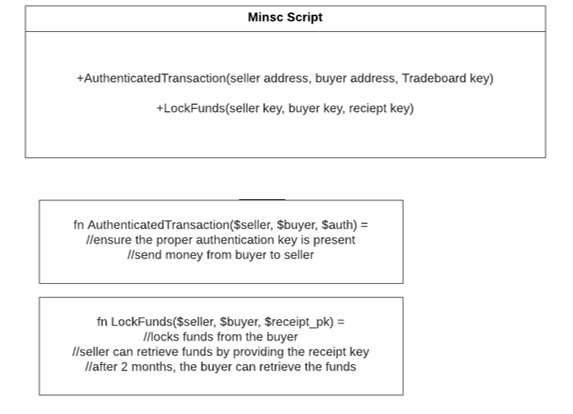


Figure 4. Minsc script diagram and code stubs for Bitcoin Tradeboard implementation.

**Tezos**

The main idea of Tezos is to allow companies to make accounting books based on its chain. Tezos is based on the OCaml Language. Tezos has its own advanced blockchain with a unique feature of an on-chain governance model that allows stakeholders to reach an agreement on proposed protocol amendments. It also auto-regulates its transactions. Other features of Tezos include self-amendment, proof of stake, and delegation. With self-amendment, Tezos is able to update itself without having to fork or split the blockchain into two different chains. Proof of stake and delegation are used to help the blockchain as a whole reach consensus on the state of the blockchain.

**Self-Amending Blockchain** – protocol that approves blocks and that modifies its own algorithm.

**Proof-of-Stake** – takes into consideration the number of tokens (the stake), a user has to push a block. Pushing a block in Tezos means to bake a block. These users are bakers. Although Tezos is relatively newer than established currencies like Bitcoin, its consensus protocol of Proof-of-Stake is more advanced than those cryptocurrencies. For example, unlike Bitcoin, Tezos handles potential malicious behaviors from bakers. The threats to the ecosystem include double baking and double endorsement. Double baking pushes two different blocks at the same time. Double endorsement is when two different blocks are placed at the same level. The protocol is designed to give out a penalty whenever one of these two acts of cheating occurs. Proof-of-Stake is designed as the baker placing an entry of 256 tokens. The system holds these tokens, now inaccessible to the baker and anyone else. During the two weeks, if the baker is caught cheating, the 256 tokens and any rewards made will be disregarded from the baker.

**Formal Verification and the use of OCaml Language** – Tezos is programmed to strictly eradicate any runtime errors or attacks. For this reason, Tezos’ base is mainly coded in the programming language of OCaml. The language’s features of static type system and memory management system are used to minimize null pointer exceptions and buffer overflows.

\*Tezos seems to be more advanced than cryptocurrencies like Bitcoin when it comes to preventing attacks and preventing programming errors. The Tezos ecosystem actually has an entire public network used for testing, called **Zeronet.** Other two public tezos networks: Mainnet – runs with real tezos tokens that have been baked or allocated, Alphanet – uses free tokens, a reference network for developers.

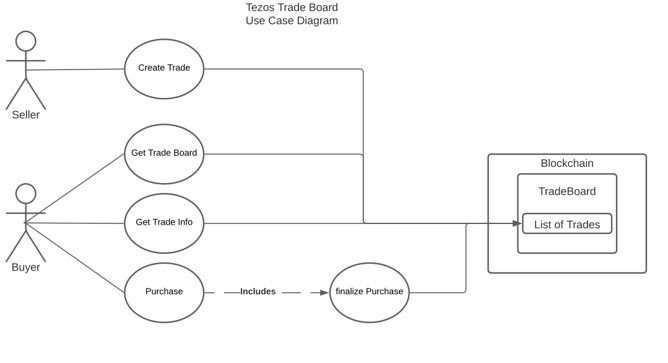


Figure 5. Use Case Diagram for Tezos Tradeboard Implementation

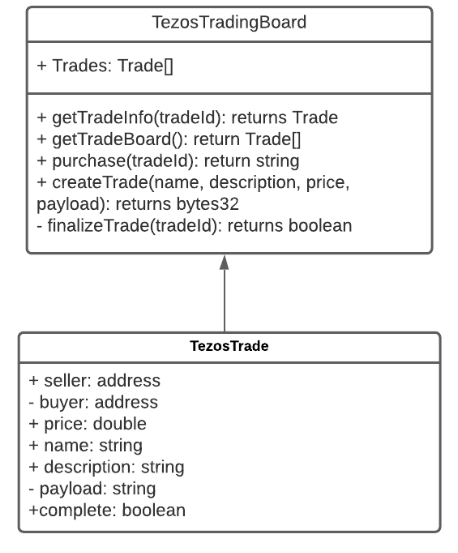


Figure 6. UML Class Diagram for Tezos Tradeboard Implementation

Tezos has several tools for development. Smartpy is an online IDE and testing environment that allows Python code to be compiled into OCaml. Smartpy, however, has very little documentation, so development can be challenging. Testing in Smartpy is done by utilizing internal functionality to build an html output that is displayed in window. Documentation on the internal functionality is undersupported. Deployment can be done via Truffle very similarly to Ethereum.

Implementation of a Tradeboard in Tezos is simpler than Ethereum due to increased transaction throughput from a proof of stake algorithm. Tezos can handle 40 tx/s. While this would not entirely remove the concern of double buying, it is far less likely to happen. For this reason, the Tezos contract does not need to handle offers and buyers can simply purchase without seller intervention. This is shown in Figures 5 and 6.

Tezos utilizes several methods to get ahead of the competition. Proof of Stake appears to be the future of cryptocurrencies due to its scalability, and with self-amending blockchain, Tezos is far ahead of the curve in terms of scalability. However, Tezos does not have the popularity of Ethereum or Bitcoin. While it seeks to contend with Ethereum in smart contracts, Tezos’ development tools are overshadowed by those of Ethereum.

**Cardano**

Cardano is an up and coming ‘gen 3’ crypto that looks to compete with real currency transaction companies. The currency, known as the ada, is currently ranked 8th in the world in market capitalization by coinmarketcap.com. At roughly $0.15 usd ada is a cheap currency that has very few current stand-out features. Cardano is proof of stake utilizing a strategy to encourage large stake pooling by utilizing proof of stake’s natural incentive for pooling and decentivising mega pools. The result is an environment of about 300 significant stake pools averaging a stake total of roughly 30 million ada[13]. The coin is sparse in features, but Cardano’s ambitious and active roadmap brings eyes to the currency.

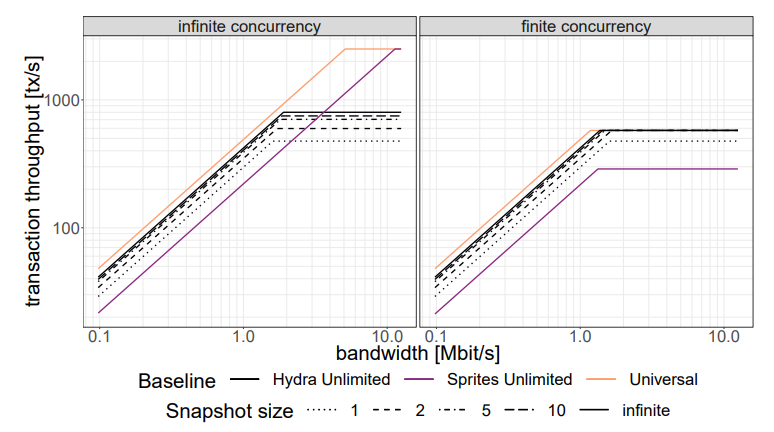
From launch in 2017, Cardano set high sights to set up ada as a true global payment system. To do this, Cardano must compete with real world payment systems. Visa for example holds an average of over 1600 transactions per second, a far cry from Bitcoin’s 4.5 transactions. In order to fulfill this requirement, Cardano performed research on the expansion of proof of stake and proposed several solutions. On launch, Cardano’s estimated best transaction throughput was 256 tx/s [12], but new research into a protocol labeled hydra shows great promise in scalability. By utilizing the stake pools formed under ada, Cardano can implement ‘hydra heads’ that handle most of the computational bandwidth with a process known as sharding. Sharding offloads work from the main blockchain to side chains or external ledgers. This means each stake pool can handle computation simultaneously. The throughput of a single hydra head can be visualized in Figure 7. Hydra protocol handles transactions in groups known as snapshots. The snapshot size defines how many transactions are in each snapshot. Comparing hydra to the previous best baseline, sprites, and the least secure, least restrictive baseline, universal, with infinite concurrency shows hydra is limited at high bandwidth. However, in a more realistic world of finite concurrency, hydra protocol far outpaces sprite protocol and nearly matches universal. Hydra is shown to have a peak of about 500-600 tx/s per ‘hydra head’ depending on snapshot size[14].

Figure 7. Simulation graph of transaction throughput by bandwidth for various baselines with infinite concurrency and finite concurrency. Universal is a baseline with the weakest possible requirements to maximize performance. Sprites Unlimited is the pre-Hydra best baseline. Hydra Unlimited is the new best baseline and is visualized in 4 lines based on the number of transactions in each snapshot. Graph provided by Cardano in a study of Hydra[14].

Hydra was theorized based on Cardano’s up and coming smart contract language, Haskell. Haskell is a low level functional language for building Cardano smart contracts and is yet to be released. Cardano plans to implement a full suite of tools for smart contract development. A development and execution platform is planned called Plutus which will utilize the low level scripting language Plutus Core. Marlowe will be a domain-specific language for non-technical accessibility to create financial smart contracts. IELE and KEVM are both proposed virtual machines for Cardano smart contracts designed for interoperability with Ethereum’s virtual machine and Solidity. Cardano’s goal moving into smart contracts is to both rival Ethereum in scope and collaborate with Ethereum to build a larger shared ecosystem. [15]

As an ecosystem, Cardano is still very young. Many of the key features of Cardano have yet to be implemented. So as it stands, Cardano is simply a proof of stake payment system that is faster than its competitors. Excitement around Cardano stems from its research. Many of the features Cardano looks to implement are ground breaking. They will change the state of cryptocurrencies and blockchain alike, but only they are far from trivial. Cardano is an ecosystem to look out for, but until it delivers on its ambitious promises, its future remains unclear.

**Diem (Libra)**

On June 18, 2019, Facebook announced its first venture into blockchain technologies, a payment system and private cryptocurrency, Libra. The goal of Libra is to create a collection of stable cryptocurrencies backed by real world currencies and US Treasury securities[16]. Unlike traditional cryptocurrencies, Libra is not decentralized. It is instead backed by the Libra Association, a trusted union of Libra service partners. Transactions will be performed at each service partner and reconciliation between partners will occur on the private blockchain and verified through proof of authority. Each member of the Libra Association provides 10 million usd to prove trust and provide full funding at launch. Libra as a currency will provide faster and cheaper long distance money transfers than traditional currencies, without the volatility that cryptocurrency is known for[16]. Sadly shortly after announcement, the development of Libra was blocked by both the US government and the European Union. Government concerns arose over privacy and security of blockchain technologies. Resolution of security and privacy concerns required a complete restructure of Libra, and by early 2020, many of the Libra Association’s most notable members including Visa, Mastercard, and Paypal abandoned the project.

On December 1, 2020, the Libra Association announced the rebranding of Libra to Diem. The switch comes with a new plan for the currency that is far less ambitious than the last. Diem is still a centralized cryptocurrency that will provide a cheap and fast alternative to physical currency that is backed by physical currency and US securities[17]. However, the scope of the project is smaller. Diem does not try to replace fiat currencies, it looks to compliment them. This can be seen with Diem’s single stablecoin the Diem Dollar which will match the usd[18].

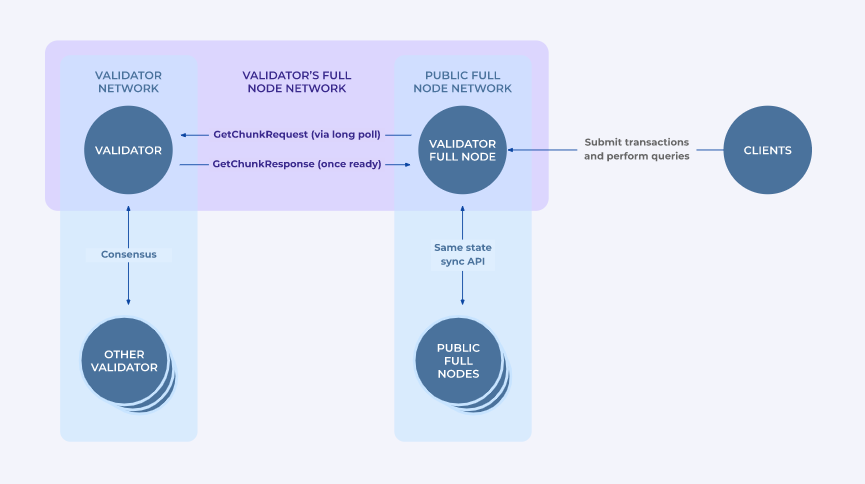
Diem is structured around a private blockchain that is maintained by Diem service partners. Each service partner is prepared to fulfill external transactions independently from the blockchain. Transactions between partners are handled over the private blockchain, which can be accessed through Diem Nodes. Diem Nodes act as validators and handle the vast majority of Diem overhead. They handle Json-RPC services, mempools, storage, consensus between validators, execution, state synchronization, and Diem virtual machine[16]. In Figure 8., the Diem Node architecture is expanded. Clients interact with a Validator fullNode either through the node itself of a state synchronized public fullNode. The validator fullNode will then send transactions to a validator node that will handle all validator network interactions and return a response.

Figure 8. Diem Node Network Architecture[16].

Development on Diem is still in very early form. Diem has three SDKs released for writing smart contracts in Java, Go, and Python with Rust on the way, but the coin has not launched yet, so all that can be done is early testing in the Diem CLI [18]. Due to the private nature of the blockchain, deploying a Dapp may require licensing and permissions, though this has yet to be revealed. It is still far too early to determine the path of Diem. It fulfills the desire for stable crypto and has backing from large tech and marketplace companies. Should the currency survive its early years, Diem will likely be in the picture for a long time due to its stability and backing. Once the currency is on its feet, expect further development to be more in line with Libra’s goals.

**Conclusion**

In conclusion, we have researched 5 cryptocurrencies: Ethereum, Bitcoin, Tezos, Diem, and Cardano. For those that have implemented smart contracts, we implemented a tradeboard to better understand the design process and development path of each environment.. Through our research we found that the strengths and weaknesses of blockchain ecosystems lie in their popularity, development tools, features, and future. Of the currently implemented smart contract systems, Bitcoin is the least promising. Bitcoin has no support for dapps, severe contract limitations due to a Turing incomplete language, and a dark future due to concerns of scalability. Bitcoin’s only saving grace is its popularity. Tezos is the next best ecosystem. Tezos leads the pack in proof of stake scalability, though it does not entirely solve the issue of slow transaction throughput. Tezos’ development tools allow the development of dapps, but lack of documentation and limited tools make the experience more difficult than it could be. Tezos’ lead as a scalable proof of stake ecosystem is shrinking as Ethereum 2 and Cardano come closer to fruition. It is unclear if Tezos can keep up. The best blockchain ecosystem to develop in currently is Ethereum. While it is restricted by the shortcomings of proof of work, Ethereum leads the pack in smart contract development. Its popularity far exceeds that of other dapp ecosystems. With the coming of Ethereum 2, Ethereum will solve the concerns of scalability and speed. For development of today and the near future, Ethereum still pulls out ahead.

Two ecosystems we researched have yet to be fully realized; Cardano, Libra(Diem). Cardano set ambitious goals as a proof of stake currency that can handle over 1000 transactions per second. If Cardano can deliver, the ada will be the fastest cryptocurrency by far providing the tools of Ethereum and the speeds of real world payment systems, but Cardano’s systems are still in early development. If Cardano delivers all that it plans, it will likely surpass Ethereum. If it fails to deliver, ada will likely fade into obscurity. Diem is very similar. Diem’s centralized and stable cryptocurrency is very unique and has the potential to solve both the crypto problem and the fiat currency problem, and if it can make it off the ground, the support of Facebook and the Diem Association will likely keep it afloat. However, government bodies look at Diem with apprehension. The currency may be struck down before it can reach its goals. Both Cardano and Diem are currencies to watch as they grow, but their futures are still unclear.

Blockchain ecosystems of the future are moving away from proof of work towards proof of stake to ensure blockchain ecosystems remain scalable, fast, and secure.

**Works Cited**

1. Toyoda, K., Machi, K., Ohtake, Y., & Zhang, A. N. (2020). Function-level bottleneck analysis of private proof-of-authority ethereum blockchain. *IEEE Access*, *8*, 141611-141621.
2. Lovejoy, James, and Anne Ouyang. (2020). 51% Attacks. *MIT Digital Currency Initiative*, MIT, dci.mit.edu/51-attacks.
3. Vukolić, M. (2015, October). The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication. In *International workshop on open problems in network security* (pp. 112-125). Springer, Cham.
4. Saleh, F. (2020). Blockchain without waste: Proof-of-stake. *Available at SSRN 3183935*.
5. Bentov, I., Lee, C., Mizrahi, A., & Rosenfeld, M. (2014). Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract] y. *ACM SIGMETRICS Performance Evaluation Review*, *42*(3), 34-37.
6. *Ethereum White Paper*. https://ethereum.org/en/whitepaper/.
7. *Eth2 Upgrades.* <https://ethereum.org/en/eth2/>.
8. Nakamoto, S. (2008). Bitcoin whitepaper. *URL: https://bitcoin. org/bitcoin. pdf-(Дата обращения: 17.07. 2019)*.
9. *Transactions per Second.* <https://www.blockchain.com/charts/transactions-per-second>.
10. Croman, Kyle; Eyal, Ittay (2016). ["On Scaling Decentralized Blockchains"](http://www.comp.nus.edu.sg/~prateeks/papers/Bitcoin-scaling.pdf) (PDF). *Financial Cryptography and Data Security*. Lecture Notes in Computer Science. **9604**. pp. 106–125.
11. Allombert, V.(2019, September). Introduction to the Tezos Blockchain.
12. *Hoskinson, C. (2019) Cardano Whitepaper.*
13. Aydinli, K. Performance Assessment of Cardano.
14. Chakravarty, M. M., Coretti, S., Fitzi, M., Gazi, P., Kant, P., Kiayias, A., & Russell, A. (2020). Hydra: Fast Isomorphic State Channels. *IACR Cryptol. ePrint Arch.*, *2020*, 299.
15. *Goguen*. Cardano roadmap. (2020). https://roadmap.cardano.org/en/goguen/.
16. *Diem Whitepaper.* https://www.diem.com/en-us/white-paper/.
17. *Announcing the name Diem. Executive leadership in place in preparation for launch.* Diem Association. (2020, December 1). https://www.diem.com/en-us/updates/diem-association/.
18. *The Diem Association*. Home Page | Diem Association. https://www.diem.com/en-us/.