XFolio: A Decentralized Cryptocurrency Portfolio Manager

Henry Stolz, Radia Soulmani, and Professor Michael J. Freedman
Princeton University

Abstract

Modern internet applications rely on centralized servers to store data and exchange information with other users. Due to this centralization of data, these servers are a prime target for digital thieves, and there is no juicier target than money. Many times a year, we learn of high-profile data leaks that release millions of credit cards, passwords, and social security numbers into the wild, and we lose more and more faith in the safety of our personal digital information.

In this paper, we introduce XFolio, a decentralized cryptocurrency portfolio application. The app solves two problems. Firstly, it is a proof-of-concept decentralized web application—it allows users to choose where their information is stored rather than being forced to trust in a central authority. Secondly, it satisfies the need for a cryptocurrency management app that embodies these decentralized properties and allows users to monitor market trends and keep track of their varied cryptocurrency holdings.

1 Introduction

1.1 The Centralized Internet

The internet in its current form is markedly centralized, not only in terms of the storage architecture of web applications but also traffic and cloud platform services.

To understand what centralization means vis-à-vis web application architecture, we take Flickr as an example. When you load your private photo album in your web browser, you request that information from the Flickr server, who verifies that you are who you claim, ensures you have access to the album you are requesting to view, and subsequently sends you the pictures from a database. As a user, you have no ability to choose how or specifically where your photos are stored, and you must *trust* that Flickr has done so for you in a way that ensures your privacy and anonymity. Thus, all users have photos stored in a *central* location and trust that Flickr has done its job securing them.

Statistics tell a striking tale in terms of the centralization of web traffic and cloud platform services. On the traffic side of things, some estimates pin the U.S. web traffic flowing through Google and Facebook at more than 25% of the country's total [1]. Meanwhile, as recently as 2015, Netflix consumed more than one third of America's total internet bandwidth [2]. In the cloud infrastructure market, the top four players (Amazon, Microsoft, IBM, and Google) control 58% of the market, a portion that only continues to increase [3]. With these few players occupying such a large portion of the internet landscape, it is difficult to avoid entrusting one's data to, for instance, Facebook—with no direct means of controlling what Facebook can or cannot do with that data [4].

The fear of centralization is not an empty one. The past five years have seen a staggering nine billion data records lost or stolen. Breaches from Equifax, Yahoo, and JP Morgan Chase [5, 6, 7] divulged the personal information—such as social security numbers from Equifax—from billions of accounts. These are prime examples of the fact that the impossibility of guaranteed security coupled with lure of troves of personal information makes a perfect target for hackers.

1.2 Centralized Currency

In much the same manner by which internet traffic flows through and is otherwise controlled by centralized parties, the flow of currency in the digital world is likewise centralized. The validity and authenticity of transactions is determined by institutions built on trust—think PayPal, Visa, Chase, Bank of America—who privately track users' funds to make these determinations.

1.3 Decentralized Currency

Almost a decade ago, Bitcoin, the first decentralized cryptocurrency, was born [9]. From its humble beginnings, Bitcoin has come into the spotlight over the past few years, leading the price

of one bitcoin to soar to an all-time high of $19,783.06 in December of 2017 [10].

Based on *blockchains*—a public, distributed ledger of transactions—the flow of bitcoins is not regulated by any central institution of trust. When members of the bitcoin network wish to make a transaction, the members of the network decide, based on the records stored within the blockchain, whether or not a transaction is valid. The clever design of the Bitcoin protocol ensures that transactions are valid and become immutable once accepted into the Bitcoin blockchain.

While we encourage the reader to study the Satoshi Whitepaper [9] to more fully understand the workings of Bitcoin and blockchains, it suffices to understand that Bitcoin gives people a system for making transactions such that one need not put trust in a central authority or any other individual. Instead, users trust the *technology* to ensure that transactions are valid.

1.4 Decentralized Web Applications

As we mentioned in section 1.1, the deep centralization of the web has introduced a number of undesirable effects: namely, the way in which users are forced to trust that their data is stored securely, the omnipresence of companies like Facebook and Google, and the large silos of personal information that too often are cracked open by malicious hackers.

In the same way that Bitcoin offers a decentralized, trust-free transaction system as an alternative to conventional, centralized systems like PayPal, we wanted to create a web application that does not obligate its users to entrust their data to one party, as is the case with the Flickr example in section 1.1 and most other existent web applications. We achieved this aim by building our application using Blockstack [11], the details of which will be discussed in section 2.1.

1.5 Cryptocurrency Management

Since the advent of Bitcoin, countless other cryptocurrencies have sprung into existence. Just as with fiat currencies or financial instruments, users need a means by which they can track their holdings of various cryptocurrencies and examine the performance of these various assets.

It would be fair game to create a conventional, centralized web application to accomplish these tasks; indeed, we overview such applications in section 4 of this paper. However, a more desirable solution is a *decentralized* web application with the same functionality. Indeed, it is only fitting that a cryptocurrency portfolio management app is as decentralized as the assets that it manages.

1.6 Goal

With these considerations in mind, we outlined our desired properties for XFolio, a decentralized cryptocurrency portfolio manager:

- Eschew conventional centralization in favor of an application architecture that allows users to choose where their data is stored
- Provide functionality for inputting and modifying the value of crypto asset holdings, namely Bitcoin, Ether, and Litecoin
- Support the viewing of detailed, interactive graphs that display crypto asset performance on various time scales

2 Approach and Implementation

2.1 Blockstack

We achieve our goal, particularly the aspect of user-designated storage, using Blockstack, “a new internet for decentralized applications that enables users to own their application data directly” [11]. Blockstack derives its functionality from a series of layers—*virtualchain*, peer discovery, and data—that overlay an underlying blockchain. These three layers work together, allowing users to register an identity in the distributed DNS coined as the “Blockstack Naming System” (BNS).

Information concerning Blockstack operations, such as the registration of names, is stored in an underlying blockchain. Parsing these elements of Blockstack data out of the blockchain reveals a *virtualchain* that records the state of the BNS. The chain and peer discovery layer in tandem provide a mapping between user identities and data (stored in the third and final layer) that has been associated with said identity. By building atop the Bitcoin blockchain, the identity system has the same properties (verifiable, immutable, decentralized, trust-free) as the underlying blockchain. The Blockstack Whitepaper [11] talks in more detail, including diagrams, about the technical architecture of Blockstack should the reader be so inclined.

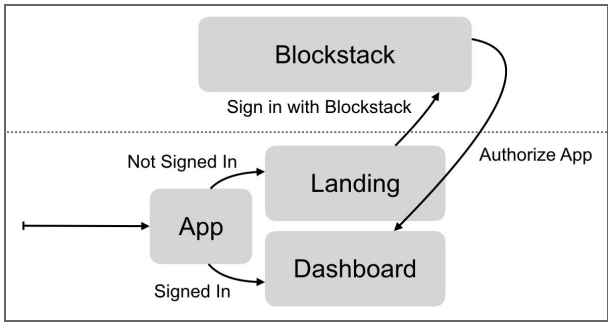

Figure 1: XFolio login flow using Blockstack

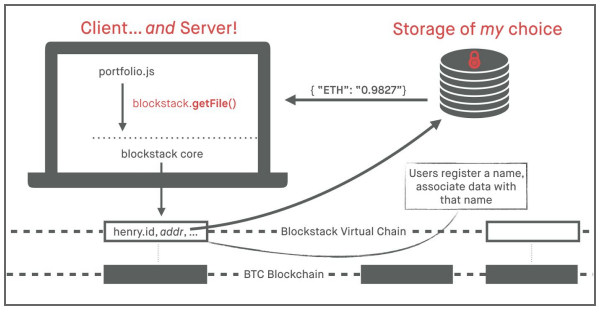

Figure 2: Blockstack file retrieval

In a simple way, you can think of a Blockstack identity as a universal, distributed version of a conventional user account. First, recall a typical account—a Facebook account, for instance. Next, imagine that you could log in to every website (Gmail, Facebook, Twitter, etc.) using a single account. Then, imagine that information about the account is not stored in any central location—instead, numerous copies of the account (and, in fact, all accounts) are shared among various nodes in the account network. Finally, imagine that, when using your account to access a web application like Facebook, your data was stored in a datastore of your choosing, rather than the database run by Facebook. That gives an idea of the potential use of Blockstack identities.

By choosing Blockstack as the architecture upon which to build XFolio, we allow users to log in to XFolio using their owned Blockstack identity (Blockstack ID). As seen in Figure 1, if you have not yet signed in with your Blockstack ID, XFolio redirects you to the Blockstack homepage, where you will be prompted to authorize XFolio to access your identifying information. Once accepted, you are redirected back to XFolio, this time to the Dashboard (since you are now signed in).

The location from and to which the application reads and writes data is determined by the storage address a user chooses to associate with their ID. A simplified diagram of the file retrieval process can be seen in Figure 2. Blockstack core, a client side server, listens for Blockstack API calls from XFolio. To handle a getFile call, Blockstack core looks up the current user's ID in the virtualchain. Not shown in Figure 2, the name has an associated hash that allows the user data address to be found via the intermediate peer discovery layer. Blockstack core can then read the file from the storage location that was chosen by the user and provide it to the client portfolio app.

Blockstack is not the only architecture that utilizes blockchains to provide functionality above and beyond pure cryptocurrency transactions. The most popular alternative is Ethereum, "an open blockchain platform that lets anyone build and use decentralized applications (dapps) that run on blockchain technology" [12]. Ethereum focuses on *smart contracts*, which are essentially pieces of code that are distributed and executed among nodes in the Ethereum network.

One example of the way in which smart contracts can be used is vDice [13], an Ethereum-based gambling dapp that mimics betting on a dice roll. A player sends some Ether (the cryptocurrency coin used on the Ethereum blockchain) to an address on the Ethereum blockchain where a variation of the vDice smart contract is stored. Nodes execute the smart contract, paying out only if the random number 'rolled' is below a target value. Having the code of the contract stored in the blockchain allows anyone to understand its behavior, and executing the contract in parallel over multiple nodes ensures that validity of the contract result.

While this architecture provides a means for certain forms of dapps, Ethereum does not have a direct focus on the decentralized storage we so desired in XFolio, which led us to select Blockstack as our platform of choice.

2.2 Architecture

XFolio is a single page JavaScript application built with Vue.js, a younger, more lightweight competitor to Angular and React. Using Vue [14], we were able to construct XFolio from a series of components that encapsulate specific, focused purposes, such as the landing page, dashboard, graph, and displaying and editing cryptocurrency holdings.

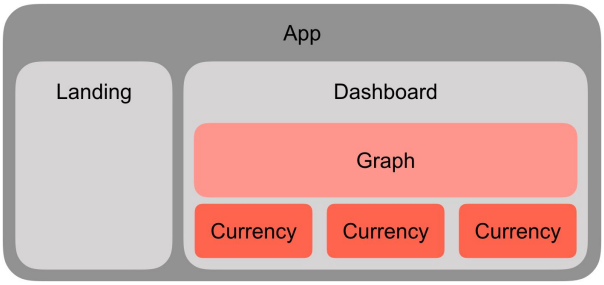

Figure 3: XFolio component hierarchy

2.2.1 Components

The hierarchy of components can be seen above in Figure 3. The main focus is on the dashboard, which is the main point of interaction for the user.

The dashboard has graph and currency child components, to whom it passes down data through what in Vue are termed *props*. For instance, the graph component is passed 'timescale', depending on the status of the time scale radio button group, this prop might have values such as 'All,' 'Month,' or 'Hour.' This allows the graph component to make the correct http request to the CryptoCompare API [15] and populate the graph with the received data. Props bind data such that, when the data changes in the parent, it does so for the child as well.

Props allow a parent component to share data with child components; passing information from child to parent component is achieved via use of *events*. For instance, each currency component emits an event when the user updates the balance (Clicking the 'Update' button of the modal in Figure 5c) of one of their cryptocurrency holdings. The dashboard component handles these events by making the relevant calls to blockstack.putFile().

The **dashboard** interface can be viewed via the Figure 4 (which doubles as a link the the demo video), showing how child components are composed visually within the single page app. The layout, styling, and standard components were adapted from the Vue-centric iView IU toolkit [16].

Figure 5a showcases the **graph** component, including the tooltip activated on mouse hover that gives precise price data. The graph component utilizes vue-echarts [17], a Vue-optimized spinoff of the popular ECharts [18] visualization library.

The **currency** component is viewable via Figure 5b. Using WebSockets, the real-time value of each coin (and the value of the user's holding) is shown. Users can click 'Edit' to bring up a modal (Figure 5c) where they update their asset balance.

Figure 4: XFolio Dashboard interface
Click to view demo

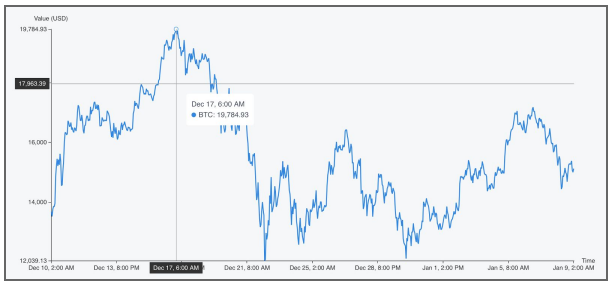

Figure 5a: Graph component

Figure 5b: Currency component

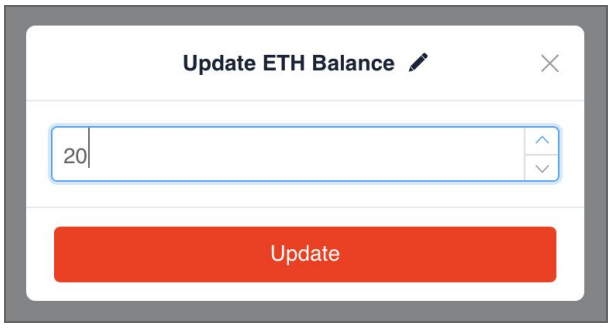

Figure 5c: Balance update modal

2.2.2 Networking

XFolio achieves its networking capabilities from a trio of technologies: Blockstack, WebSockets, and Axios. We rely on blockstack to achieve the

storage of user holdings in a decentralized manner, as detailed in section 2.1. In order to provide real-time prices for each of the three currently supported cryptocurrencies (Bitcoin, Ether, and Litecoin), we utilize WebSockets, which offer a lightweight means of getting frequent updates from a server, in this case, the GDAX API [20]. Finally, we utilize Axios, a simple HTTP client, to cleanly handle asynchronous requests to the CryptoCompare API [15] for historical price data to be graphed.

2.3 Technology Recap

- **Blockstack** — dapp architecture providing decentralized identity, storage (see section 2.1) [11]
- **Vue.js** — lightweight JavaScript framework, allows for app development using discrete components [14]
- **iView** — Vue-centric UI toolkit [16]
- **vue-echarts** — highly customizable Vue adaption of the ECharts graphing library [17]
- **WebSocket, Axios** — real-time and asynchronous data retrieval [19]

3 Results

In this evaluation, we will lay out the results obtained by XFolio, demonstrating that we have reached the goals laid out in section 1.6 of this paper. Each subsection 3.1-3 will give data to support our success on the corresponding goal from section 1.6. In addition to the data presented below, we encourage the reader to view the demo video linked **here** or as part of Figure 4.

3.1 Decentralization

As discussed in section 2.1, using Blockstack as the basis of our application ensures that the data stored by the application is located in a location determined by the user. Users can currently only choose Blockstack's own Gaia storage network, but there are plans to add support for providers such as Dropbox, IPFS, OneDrive, self-hosted storage. Because no data is stored centrally by XFolio, the concerns of having to maintain and protect centralized user information from data breaches is essentially mitigated.

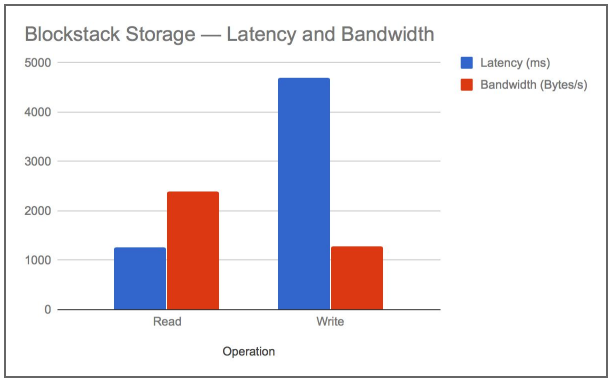

Figure 6: Blockstack read/write performance

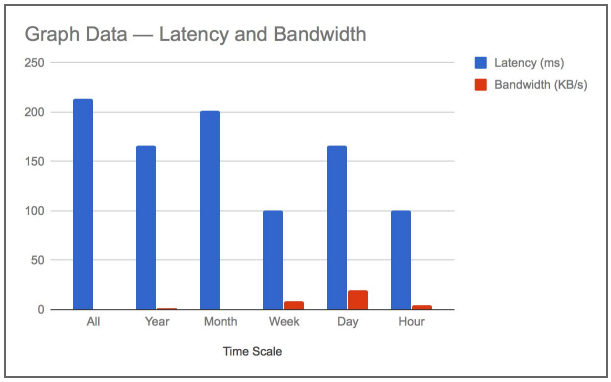

Figure 7: CryptoCompare API performance, for graph data

3.2 Read/Write Functionality

We succeeded in providing the functionality for inputting and modifying crypto asset holdings for Bitcoin, Ether, and Litecoin, as seen in Figure 5c. We also ran benchmarks to determine the read and write performance of this data, which was stored via Blockstack Gaia storage. Benchmarks were run on macOS running Safari 11, connected to the Princeton University eduroam wireless network. The browser cache was disabled and timing was determined via Safari developer tools. Data resulted from averaging over five trials.

The results above speak to the less-than-stellar read/write performance of Blockstack currently. In particular, there was consistent latency of up to five seconds to store updated holding values, which led to a deteriorated user experience in waiting for the updated value to take place. However, read performance was not low enough to affect usability greatly. It is unclear whether the low bandwidth is

a result of the small size of the files being transferred (on the order of hundreds of bytes).

One significant problem encountered with writing to Gaia storage, not displayed in the figures, is the occasional failure for written data to persist, despite the apparent success of the blockstack.putFile() method call. Further investigation will be necessary to determine the root of this problem.

3.3 Graphs

We also succeeded in providing detailed, interactive graphs that display crypto asset performance at varying time scales, which is visible in the Figure 4 demo video.

We ran benchmarks to determine the performance of the graph data calls, with the results visible in Figure 7. Benchmarking methods were identical to those in section 3.2. The coin whose data was requested was Bitcoin.

In contrast to the lengthy latency and low bandwidth performance of the Blockstack storage, the CryptoCompare API provided nearly instantaneous updates to our graphs, though this is not to be wholly unexpected of a popular public API. The better performance of the API as the time scale grew more and more recent is not unexpected, considering that the API likely receives more requests for the past 24 hours of data (and thus caches it or otherwise prioritizes it) versus the past year. At this time, the performance of the Cryptocompare API seems adequate to support XFolio's graph functionality.

4 Related Work

As mentioned in section 1.5, there exist many other cryptocurrency portfolio management applications, such as the prominent Coinbase [10], or the mobile-based Lawnmower [21]. However, we were unsatisfied with these solutions, because they still were built on the same underlying, centralized web app architecture, doing nothing to break away from the model that entices hackers to find security holes and extract precious information.

When we set out to create XFolio, there existed no cryptocurrency portfolio management dapp built with Blockchain; however, since then, more than ten have appeared, one example being Coinfort [22]. This is likely due to the bounty [23] posted by Blockstack to encourage the development of such an app; we, too set out to create XFolio after being inspired by the challenge, though had no realistic hopes for the prize itself.

5 Conclusion

In this paper, we presented XFolio, which not only is a proof-of-concept for trust-less, decentralized web applications, but also a serves an immediate need in the niche of cryptocurrency portfolio management dapps. We showed the means by which blockchains and Blockstack enable us to create dapps in which users need not blindly trust in a centralized party to store and safeguard their personal information against security threats. We explained the composition and displayed the functionality that XFolio offers through demo videos and timing data. As of writing, XFolio is still far from a complete portfolio management solution, with challenges ranging from inconsistent data read and write performance to lack of support for a broader range of cryptocurrencies. The newest version of XFolio can be cloned from the public Github repository at **https://github.com/hstolz/xfolio**.

References

[1] Experian. Most Popular Websites in The United States as of February 2016, Based on Share of Visits. Retrieved January 9, 2018 from https://www.statista.com/statistics/265770/most-popular-us-websites-by-market-share-of-visits/

[2] Claire Groden. 2015. See How Much Bandwidth Netflix Consumes in One Chart. (October 2015). Retrieved January 9, 2018 from http://fortune.com/2015/10/08/netflix-bandwith/

[3] Christine Hall Hall. 2017. Microsoft's Cloud Market Share Grew More than Anyone Else's Last Quarter – Analysts. (August 2017). Retrieved January 9, 2018 from http://www.datacenterknowledge.com/business/microsofts-cloud-market-share-grew-more-anyone-elses-last-quarter-analysts

[4] Laura Shin. 2017. Blockstack On How To Take Control From Google, Facebook And Amazon. (September 2017). Retrieved January 9, 2018

from https://www.forbes.com/sites/laurashin/2017/09/05/blockstack-on-how-to-take-control-from-google-facebook-and-amazon/#4535de4d6abf

[5] Data Breach Statistics. Retrieved January 9, 2018 from http://breachlevelindex.com/

[6] Stacy Cowley. 2017. 2.5 Million More People Potentially Exposed in Equifax Breach. The New York Times (October 2017).

[7] Natt Garun. 2017. Yahoo says all 3 billion user accounts were impacted by 2013 security breach. (October 2017). Retrieved January 9, 2018 from https://www.theverge.com/2017/10/3/16414306/yahoo-security-data-breach-3-billion-verizon

[8] Jessica Silver-Greenberg, Matthew Goldstein, and Nicole Perlroth. 2014. JPMorgan Chase Hacking Affects 76 Million Households. The New York Times (October 2014).

[9] Satoshi Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” tech report, 2009. https://bitcoin.org/bitcoin.pdf.

[10] Anon. Bitcoin, Ethereum, and Litecoin Price. Retrieved January 9, 2018 from https://www.coinbase.com/charts

[11] Muneeb Ali, Ryan Shea, Jude Nelson, and Michael J. Freedman. 2017. Blockstack: A New Internet for Decentralized Applications. (October 2017). Retrieved January 9, 2018 from https://blockstack.org/whitepaper.pdf

[12] Ethereum community. Ethereum Introduction. Retrieved January 9, 2018 from http://www.ethdocs.org/en/latest/introduction/

[13] Anon.Retrieved January 9, 2018 from https://www.vdice.io/

[14] Vuejs. Vue. Retrieved January 9, 2018 from https://github.com/vuejs/vue

[15] Anon. CryptoCompare API Introduction. Retrieved January 9, 2018 from https://www.cryptocompare.com/api/

[16] iView. iView. Retrieved January 9, 2018 from https://github.com/iview/iview

[17] Justineo. vue-echarts. Retrieved January 9, 2018 from https://github.com/Justineo/vue-echarts

[18] ecomfe. ECharts. Retrieved January 9, 2018 from https://github.com/ecomfe/echarts

[19] axios. axios. Retrieved January 9, 2018 from https://github.com/axios/axios

[20] Anon. GDAX API Reference. Retrieved January 9, 2018 from https://docs.gdax.com/

[21] Anon.Retrieved January 9, 2018 from https://lawnmower.io/

[22] Anon.Retrieved January 9, 2018 from http://www.coinfort.io/

[23] Anon. Signature Bounties: Encrypted Token Portfolio App. Retrieved January 9, 2018 from https://www.eventbrite.com/e/signature-bounties-encrypted-token-portfolio-app-registration-38154648581#