Federal Reserve Bank of Boston®

Payments Innovation

February 3, 2022

Project Hamilton Phase 1 Executive Summary

Federal Reserve Bank of Boston and Massachusetts Institute of Technology Digital Currency Initiative

Contents

The views expressed in this paper are those of the author and do not necessarily represent those of the Federal Reserve Bank of Boston or the Federal Reserve System.

Introduction

In light of continued innovation in money and payments, many central banks are exploring the creation of a central bank digital currency (CBDC), a new form of central bank money which supplements existing central bank reserve account balances and physical currency [5]. CBDCs could exist in various forms depending on a central bank's objectives, including a general-purpose CBDC that can be made available to the public for retail, e-commerce, and person to person payments. Central banks, researchers, and policymakers have proposed various objectives including fostering financial inclusion, improving efficiency in payments, prompting innovation in financial services, maintaining financial stability, and promoting privacy [2,3,9,19].

Because the CBDC research process is still in early stages in many jurisdictions, several technical design questions remain open for investigation. The answers to these questions will have meaningful implications and consequences for what options are, or are not, available to policymakers.

The Federal Reserve Bank of Boston (Boston Fed) and the Massachusetts Institute of Technology's Digital Currency Initiative (MIT DCI) are collaborating on exploratory research known as Project Hamilton, a multiyear research project to explore the CBDC design space and gain a hands-on understanding of a CBDC's technical challenges and opportunities. This paper presents the project's Phase 1 research. Our primary goal was to design a core transaction processor that meets the robust speed, throughput, and fault tolerance requirements of a large retail payment system. Our secondary goal was to create a flexible platform for collaboration, data gathering, comparison with multiple architectures, and other future research. With this intent, we are releasing all software from our research publicly under the MIT open source license.[1]

By focusing Phase 1 on the feasibility and performance of basic, but resilient transactions, we aim to create a foundation for more complex functionality in Phase 2. The processor's baseline requirements include time to finality of less than five seconds, throughput of greater than 100,000 transactions per second, and wide-scale geographic fault tolerance. Topics left to Phase 2 include critical questions around high-security issuance, systemwide auditability, programmability, how to balance privacy with compliance, technical roles for intermediaries, and resilience to denial of service attacks.

As exploratory research on the implications of different design choices, this work is not intended for a pilot or public launch. That said, we consider performance under a variety of extensive, realistic workloads and fault tolerance requirements.

[1] https://github.com/mit-dci/opencbdc-tx

Core Design and Results

In Phase 1, we created a design for a modular, extensible transaction processing system, implemented it in two distinct architectures, and evaluated their speed, throughput, and fault tolerance. Furthermore, our design can support a variety of models for intermediaries and data storage, including users custodying their own funds and not requiring storing personally identifying user data in the core of the transaction processor.

In our design users interact with a central transaction processor using digital wallets storing cryptographic keys. Funds are addressed to public keys and wallets create cryptographic signatures to authorize payments. The transaction processor, run by a trusted operator (such as the central bank), stores cryptographic hashes representing unspent central bank funds. Each hash commits to a public key and value. Wallets issue signed transactions which destroy the funds being spent and create an equivalent amount of new funds owned by the receiver. The transaction processor validates transactions and atomically and durably applies changes to the set of unspent funds. In this version of our work, there are no intermediaries, fees, or identities outside of public keys. However, our design supports adding these roles and other features in the future.

The flexibility, performance, and resiliency challenges of this design are addressed with three key ideas. The first idea is to decouple transaction validation from execution, which enables us to use a data structure that stores very little data in the core transaction processor. It also makes it easier to scale parts of the system independently. The second idea is a transaction format and protocol that is secure and provides flexibility for potential functionality like self-custody and future programmability. The third idea is a system design and commit protocol that efficiently executes these transactions, which we implemented with two architectures.

Both architectures met and exceeded our speed and throughput requirements. The first architecture processes transactions through an ordering server which organizes fully validated transactions into batches, or blocks, and materializes an ordered transaction history. This architecture durably completed over 99% of transactions in under two seconds, and the majority of transactions in under 0.7 seconds. However, the ordering server resulted in a bottleneck which led to peak throughput of approximately 170,000 transactions per second. Our second architecture processes transactions in parallel on multiple computers and does not rely on a single ordering server to prevent double spends. This results in superior scalability but does not materialize an ordered history for all transactions. This second architecture demonstrated throughput of 1.7 million transactions per second with 99% of transactions durably completing in under a second, and the majority of transactions completing in under half a second. It also appears to scale linearly with the addition of more servers. In order to provide resilience, each architecture can tolerate the loss of two datacenter locations (for example, due to natural disasters or loss of network connectivity) while seamlessly continuing to process transactions and without losing any data.

Learnings

Phase 1 has surfaced several key learnings on the potential design of a CBDC:

Select ideas from cryptography, distributed systems, and blockchain technology can provide unique functionality and robust performance. We suspect existing database and distributed systems technology is sufficient to provide a more traditional payment architecture for CBDC where one actor stores users' accounts, users cannot custody their own funds, and there is no transaction scripting functionality. We created a new design to offer both these features and new opportunities for different intermediary roles.

A CBDC can provide functionality that is not currently possible with either cash or bank accounts. For example, a CBDC could support cryptographic proofs of payment, more complex transfers to or from multiple sources of funds, and flexible forms of authorization to spend, such as varying transaction limits.

We found that separating a transaction processor into modular components improves system scalability and flexibility; for example, we can scale and replicate transaction validation independently from preventing double spending and committing transactions, and our architecture can support many future designs for programmability and privacy.

Despite using ideas from blockchain technology, we found that a distributed ledger operating under the jurisdiction of different actors was not needed to achieve our goals. Specifically, a distributed ledger does not match the trust assumptions in Project Hamilton's approach, which assumes that the platform would be administered by a central actor. We found that even when run under the control of a single actor, a distributed ledger architecture has downsides. For example, it creates performance bottlenecks, and requires the central transaction processor to maintain transaction history, which one of our designs does not, resulting in significantly improved transaction throughput scalability properties.

CBDC design choices are more granular than commonly assumed. Currently, CBDC designs are categorized as direct, two-tier, or hybrid models, with "token" or "account" access models [1, 2, 7, 12, 15]. We found these limited categorizations lacking and insufficient to surface the complexity of choices in access, intermediation, institutional roles, and data retention in CBDC design [10]. For example, wallets can support both an account-balance view and a coin-specific view for the user regardless of how funds are stored in the database.

By breaking transaction processing into steps like creation, authorization, submission, execution, and storing history, CBDC designers can consider the potential roles for intermediaries at each stage, creating opportunities for innovation.

By implementing a robust system, we identify new questions for CBDC designers and policymakers to address, regarding tradeoffs in performance, auditability, functionality, and privacy. Our work raised important questions to address in how the technical architecture might affect the use and function of CBDC in payments. For example, it is an open question how important from an economic perspective it might be to support *atomic transactions*. In database parlance, this implies multiple operations to different pieces of

the data are applied in a way that appears instantaneous (atomic), or the set of updates does not happen at all; there is no partial application [4,14]. In the context of a payment processor, this means users could reliably issue payments that might transfer multiple bills (or funds from multiple accounts) entirely, and would never see partial transfers, even if there are crashes or system errors. We chose to implement atomic transactions, which has a direct impact on the performance of the system [8].

The main functional difference between our two architectures is that one materializes an ordered history for all transactions, while the other does not. This highlights initial tradeoffs we found between scalability, privacy, and auditability. In the architecture that achieves 1.7M transactions per second, we do not keep a history of transactions nor do we use any cryptographic verification inside the core of the transaction processor to achieve auditability. Doing so in the future would help with security and resiliency but might impact performance. In the other architecture, we can audit the set of unspent funds to make sure they were created correctly. Storing the history of transactions implies the central transaction processor can reconstruct the transaction graph, which, in combination with other data sources, could reveal sensitive user information [16,17]. In the next phase of work, we will focus on adding privacy-preserving designs for auditability.

Similarly, our goals of supporting self-custody and reducing data stored in the core of the transaction processor had direct implications on data users might be required to store, failure scenarios, recovery protocols, and on what types of payment functionality we can support.

Phase 2

In Phase 2 of Project Hamilton, the Boston Fed and MIT DCI will explore new functionality and alternative technical designs. Research topics may include cryptographic designs for privacy and auditability, programmability and smart contracts, offline payments, secure issuance and redemption, new use cases and access models, techniques for maintaining open access while protecting against denial of service attacks, and new tools for enacting policy. In addition, we hope to collaborate and explore these challenges with other technical contributors from a variety of backgrounds in the open source repository.

Through the development and testing of its own custom software, Project Hamilton provides unique insight into the technical considerations and tradeoffs involved with the development of a core processing engine for a CBDC. Project Hamilton's research and experimentation with a fast, highly scalable, resilient, and secure technical architecture will supplement previous work by central banks including policy and economic research [13], proofs-of-concept and pilot testing [11, 18], as well as CBDCs which have been made available to the public [6].

References

[1] R. Auer and R. Böhme. The technology of retail central bank digital currency. *BIS Quarterly Review*, March, 2020.

[2] Bank for International Settlements. CBDCs: an opportunity for the monetary system. *BIS Annual Report Economic Report 2021*, pages 65–91, 6 2021.

[3] Bank of Canada et al. Central bank digital currencies: foundational principles and core features. BIS Working Group, 2020. https://www.bis.org/publ/othp33.pdf.

[4] P. A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency control and recovery in database systems*, volume 370. Addison-wesley Reading, 1987.

[5] C. Boar and A. Wehrli. Ready, steady, go? results of the third BIS survey on central bank digital currency. *BIS Papers No 114, 2021*. https://www.bis.org/publ/bppdf/bispap114.htm.

[6] Central Bank of The Bahamas. Sand dollar. https://www.sanddollar.bs.

[7] Committee on Payments and Market Infrastructures Markets Committee. Central bank digital currencies. BIS Quarterly Re-view, March 2018.

[8] European Central Bank. Work stream 3: A new solution –blockchain & eID, 2021. https://haldus.eestipank. ee/sites/default/files/2021-07/Work%20stream%203%20-%20A%20New%20Solution%20-%20Blockchain%20and%20eID_1.pdf.

[9] R. Garratt, M. J. Lee, et al. Monetizing privacy with central bank digital currencies. Technical report, Federal Reserve Bank of New York, 2020.

[10] R. Garratt, M. J. Lee, B. Malone, A. Martin, et al. Token- or Account-based? A digital currency can be both. Technical report, Federal Reserve Bank of New York, 2020.

[11] J. C. Jiang and K. Lucero. Background and implications of China's central bank digital currency: E-CNY. *Available at SSRN 3774479, 2021*.

[12] C. M. Kahn, F. Rivadeneyra, and T.-N. Wong. Should the central bank issue e-money? *Money*, pages 01–18, 2019.

[13] J. Kiff, J. Alwazir, S. Davidovic, A. Farias, A. Khan, T. Khiaonarong, M. Malaika, H. Monroe, N. Sugimoto, H. Tourpe, and P. Zhou. A survey of re-search on retail central bank digital currency, 2020. https://www.elibrary.imf.org/view/journals/001/2020/104/001.2020.issue-104-en.xml.

[14] B. W. Lampson. Atomic transactions. In *Distributed Systems Architecture and Implementation*, pages 246–265. Springer, 1981.

[15] T. Mancini-Griffoli, M. S. M. Peria, I. Agur, A. Ari, J. Kiff, A. Popescu, and C. Rochon. Casting light on central bank digital currency. *IMF staff discussion note*, vol 8, 2018.

[16] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. Mc-Coy, G. M. Voelker, and S. Savage. A fistful of bitcoins: charac-terizing payments among men with no names.

In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140, 2013.

[17] D. Ron and A. Shamir. Quantitative analysis of the full bit-coin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer, 2013.

[18] Sveriges Riksbank. E-krona pilot phase 1. *Sveriges Riks-bank Report*, 2021. https://www.riksbank.se/en-gb/payments--cash/e-krona/technical-solution-for-the-e-krona-pilot/.

[19] A. Usher, E. Reshidi, F. Rivadeneyra, S. Hendry, et al. The positive case for a CBDC. *Bank of Canada Staff Discussion Paper*, 2021.