



Archetypes for a retail CBDC

Staff Analytical Note 2022-14 (English)

Sriram Darbha

October 2022

Key messages

- A wide variety of technology designs have been, and continue to be, proposed to underpin retail central bank digital currency (CBDC) systems. Central banks need a common framework to analyze and compare the different possible designs, independent of vendor, platform and technology.
- We consider the fundamental perspective of how information, or state, related to a retail CBDC instrument is organized among participating entities. From this perspective, we develop five archetypes—common patterns that recur in system designs—and discuss their trade-offs from the perspective of privacy, compliance, visibility, scalability, resilience, extensibility, online and offline payments.
- These archetypes can be combined and are general enough to collectively cover a wide range of possible CBDC designs. We anticipate that some CBDC system designs will closely align with one archetype, some will be variations of an archetype, and others will combine aspects of multiple archetypes.
- Although our fundamental perspective is how information is organized, we note that the space of possible designs could be analyzed differently. A closely related perspective is the representation of money (i.e., what the structure of information is).
- Our analysis suggests that no archetype scores highly across all criteria, so a design based on a single archetype is not likely to satisfy all policy goals. Instead, guided by the policy goals of their jurisdiction, policy-makers should consider a design that combines aspects of multiple archetypes.

Introduction

A CBDC system's state is the information the system maintains about the supply and ownership of the digital currency. The space of storage and database systems to underpin a CBDC system is vast, with hundreds of technology platforms ranging from general-purpose ones, such as relational databases, to niche ones, such as NoSQL databases for social networks and blockchain systems for cryptocurrencies.

This raises a few questions:

- What types of storage systems could support a retail CBDC system?
- Are certain types of storage systems particularly well suited (or unsuitable) for a CBDC?
- Do blockchain systems present compelling benefits for retail CBDC systems because they are prevalent in cryptocurrency applications?
- Can the large variety of possible system designs be framed and analyzed based on some common design ideas?

After studying a variety of CBDC system designs, we believe it is useful to organize the possible CBDC designs according to a few archetypes that are independent of vendor, platform and technology. These archetypes will be of interest to a technical audience. However, we present them in a manner accessible to non-technical audiences such as policy-makers and members of the general public who can recognize the range of possible system designs and their trade-offs.

Before developing the archetypes, we list some assumptions that define the scope of choices we considered.

A CBDC system must store state and maintain it in a way that preserves the integrity of the CBDC supply. A change in the state happens any time an operation that affects the CBDC supply is conducted, including transactions or issuing additional currency. The change involves updating the system state with the information required to record the settlement for that operation. Regardless of the state's structure, the CBDC system must ensure that every update to the state is applied in a way that preserves the integrity of the supply. For example, if the state is modelled as a set of account balances, a change in state triggered by a transaction must update the balances of the parties involved in the transaction so that the debits offset the credits exactly and the correct parties—and no others—are debited and credited.

We assume the CBDC system is self-sufficient and contains all the information related to CBDC liabilities. Other designs are possible where some state is held outside the CBDC system.¹ Settling CBDC obligations in such systems would require that the parties involved comply with technical and governance obligations outside the system's control. Such systems are beyond the scope of this analysis.

We assume that a CBDC system must achieve global consistency of updates with near-immediate finality. Consistency is a system property where all read operations must return the most recently written value (Gilbert and Lynch 2002). Some systems sacrifice consistency to achieve high throughput, or transactions per second; in this case, it is possible that parts of the system may not return the most recently written value until some time after an update. A CBDC system must always be globally consistent (i.e., all parts of the system must return the same value on a read operation) and achieve consistency quickly (i.e., return the most recent value almost immediately after an update).²

A range of entities will participate in a retail CBDC system—the central bank, money services businesses (MSBs) such as financial institutions and other private sector participants, end users and merchants. Organizing state in a CBDC system involves deciding how the system state is stored across these various entities. Further, it involves establishing how oversight over a proposed update to that state is provided—in other words, how a proposed update is validated, approved or rejected so the information needed to update the system state can be recorded if the transaction is approved.

Archetypes

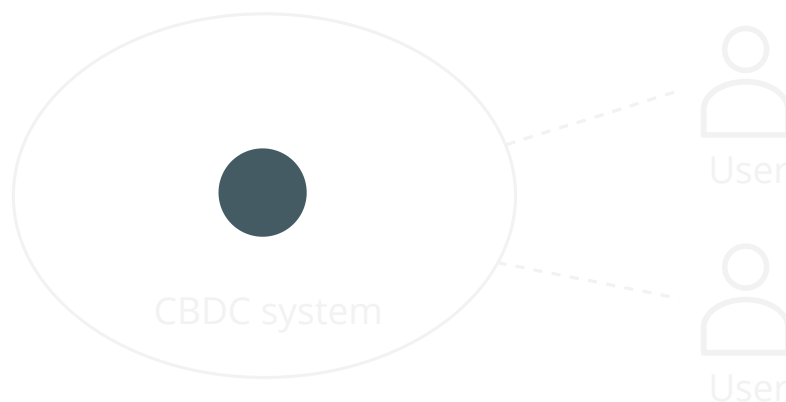
In this section, we discuss the options for storing and updating the system state and develop five archetypes:³

- Centralized
- Leaderless
- Macro-partitioned
- Micro-partitioned
- Direct

Centralized

The distinguishing feature of the centralized archetype is that the total system state is within the trust zone of, and controlled by, one entity. This entity would have the power to approve and apply each update or deny it. Users connecting to the system do not hold any state, only credentials that authorize access. An operation, such as a transaction or issuance, could be fully recorded as a change to information within the zone.

Figure 1: Centralized archetype



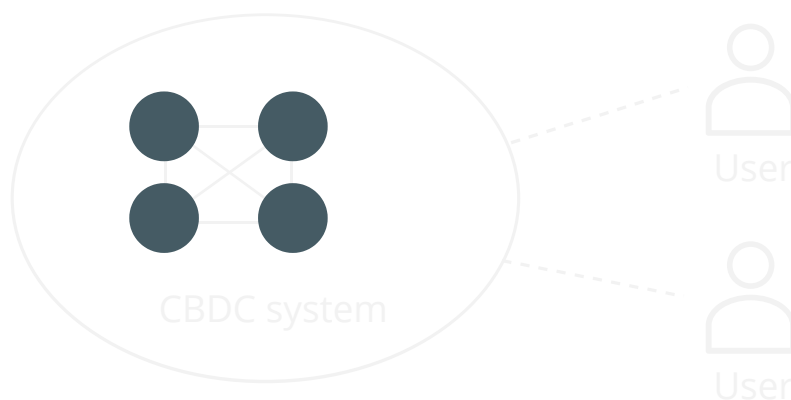
In practice, a CBDC system would not be deployed as a single instance. Instead, it would replicate state, possibly to different geographic sites, for backup and availability. Nevertheless, if the instances are under a single entity's authority, the system can be considered centralized because it preserves the essential characteristic of the total state controlled by one party.

A system based on a centralized archetype could take other forms, such as a collection of multiple components that operate together as one logical instance and are within the trust zone of one entity. A range of platforms—from traditional databases to purpose-built platforms such as the Guardtime KSI (Eesti Pank 2021)—can be deployed as centralized systems.

Leaderless

Another way to organize state is to replicate it in its entirety to multiple identical instances that are controlled by different entities and organized to be leaderless (i.e., no instance is the leader). The crowd of instances collectively provides oversight over each update, ensuring that it is applied by all replicas in the same manner, progressing all of them from the same current state to the same next state. The process to achieve agreement on updates is called consensus, and numerous consensus algorithms are known (Cachin and Vucolic 2017). This is the leaderless archetype, exemplified by systems such as Bitcoin (Nakamoto 2009), Ethereum (Buterin 2014) and Algorand (Chen and Micali 2017).

Figure 2: Leaderless archetype



Since all instances in a leaderless system are copies of the total system state, every update must be applied to all instances. This can lead to a high overhead of communications (**Figure 2**, solid lines) and duplication of storage and computation. Again, users connecting to the system do not hold any state, only credentials.

A key aspect of leaderless systems is their ability to operate in low-trust environments, so users do not have to trust one state instance or institution. In general, the lower the trust level, the higher the overhead and complexity of consensus required. We assume that most retail CBDC systems will not operate in low-trust environments because institutions performing system roles will almost certainly be required to be granted permission and be trusted by both the central bank and regulators.

The need for strong central bank roles and functions must be assessed against the default posture of leaderless systems. In most jurisdictions, a CBDC system would be required to limit certain functions; for example, the central bank would be the exclusive issuer of the CBDC. This contrasts with leaderless systems where the authority over those functions (i.e., over the technology mechanisms that encode those functions) is shared between multiple entities. Leaderless systems must therefore be evaluated to ensure that there are no security concerns for the roles and functions of the central bank.⁴

Macro-partitioned and micro-partitioned

An alternative to replicating the system state is to partition it, that is, to divide it into partitions or slices, each controlled by a different entity. Any two partitions (**Figure 3**, grey circles) would differ from each other, while possibly overlapping in content related to their shared interactions. Each partition may also contain private state that is not shared with other partitions. We call these “partitioned-state systems.” In contrast to centralized and leaderless systems, no instance presents the total system state in partitioned-state systems.

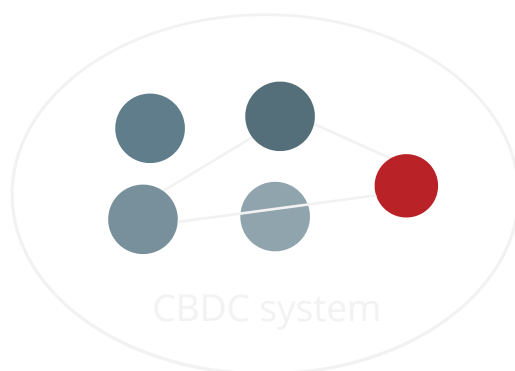
In a partitioned-state system, only some partitions change during an update. The system must ensure that the updates applied to those partitions are valid, that double-spends and fake issuance are disallowed even if owners of those partitions try to collude. A third-party function integrated with all partitions could provide oversight of updates.

One way to implement that third-party function is as a component that communicates with all partitions. Suppose that two partitions are involved in a transaction (**Figure 3**, solid lines). They would interact with the third party (**Figure 3**, red circle) to gain its approval and then apply the update to their respective states, while the third party records some shared state.⁵ Other partitions may not be aware that an update occurred. A third party that holds some shared state and provides oversight may, however, introduce a degree of centralization.

Another way is by distributing that function between all partitions as a decentralized mechanism or protocol. In that case, transacting partitions participate in the oversight protocol to generate state information to prove the validity of the update, which is then shared with all partitions.⁶

Regardless of the design of the oversight function, it can be considered a protocol and some shared state that all partitions have access to.

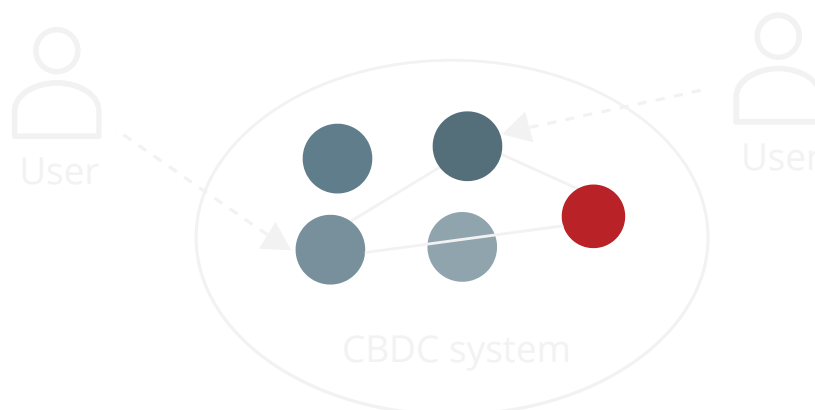
Figure 3: Partitioned state



We describe two archetypes based on the notion of partitioned state: the macro-partitioned and the micro-partitioned.

In the macro-partitioned archetype, MSBs own and operate partitions. The number of partitions would be small (for example, in the dozens), while each partition would be large enough to represent a percentage of the total system state. Further, partitions could be assumed to be always on and connected to other system entities. Many platforms can be configured as macro-partitioned designs, for example, OpenCBDC, R3 Corda and HyperLedger Fabric (Androulaki et al. 2018).

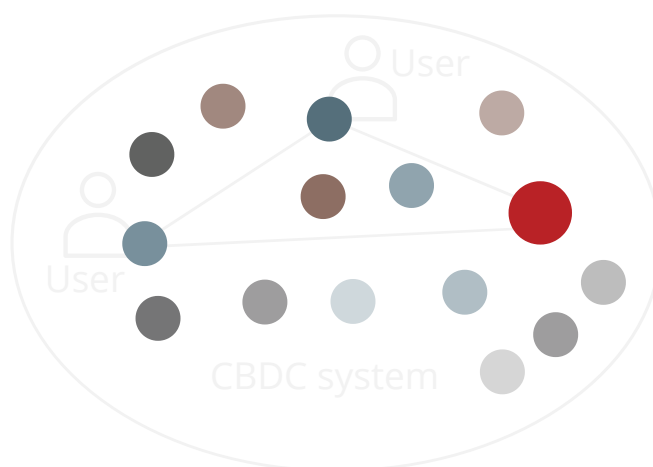
Figure 4: Macro-partitioned archetype



In the micro-partitioned archetype, the number of partitions would be large (in the millions), but each partition would be a small slice of the total state. The function of holding state is pushed out to the edge of the system so that end users (i.e., individuals, merchants and corporations conducting the transactions) maintain the different partitions. Hence, as shown in **Figure 5**, end users hold system state, not just credentials. For this reason, partitions cannot be assumed to be always on and reachable. Examples of platforms deployable as micro-partitioned designs include TODAQ (Gravitis, Goh and Toliver 2019) and OpenCBDC (MIT Digital Currency Initiative 2022).

In general, a platform designed to be deployed as a micro-partitioned system could also be deployed as macro-partitioned (by having institutions maintain custody of the state of end users). The reverse may not be feasible.

Figure 5: Micro-partitioned archetype

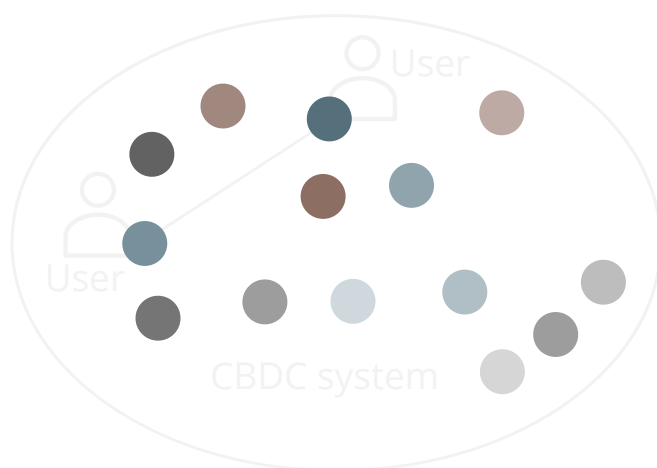


Direct

An alternative partitioned-state system—the direct archetype—is one in which transacting partitions directly provide their own oversight.⁷ That is, transacting parties would communicate between themselves, without involving other parties, to exchange and record settlement information in their respective partitions. The technology and protocols would preserve integrity of state, even though the update is not overseen by a third party.

The direct archetype is the only one that achieves cash-like person-to-person transactions, where one party can interact with a second and settle a transaction without involving a third party, as shown in **Figure 6**.

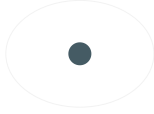



Figure 6: Direct archetype



The lack of third-party oversight has security implications. To our knowledge, achieving a design for a direct archetype requires using secure, tamper-resistant hardware to maintain and update the state.⁸ Secure hardware-based solutions have been deployed at a relatively small scale in closed-loop systems, such as university campuses and public transit. However, it is unknown whether their security can be hardened sufficiently to support a general-use fiat currency system at the scale of a national population. The worst-case risk is that a compromised partition could be used as a sort of printing press to issue CBDC fraudulently without the central bank finding out.

Figure 6 shows partitions in a direct archetype being operated by end users because this is the most likely scenario. However, a direct archetype-based system with institutional partitions is also possible. We do not describe this separately, but all direct archetype-based systems must address the same security concerns and maintain state within tamper-resistant hardware.

Table 1: Summary of archetypes

	Centralized	Leaderless	Macro-partitioned Micro-partitioned	Direct
				
How state is maintained	A single entity maintains the state within its trust zone.	The state is fully replicated to multiple identical instances owned by different entities in a leaderless system.	The state is split across different partitions owned by multiple entities, with a small, shared state.	The state is split across different partitions owned by multiple entities.
How state updates are overseen	The owner of the trust zone oversees updates.	A state update changes all replicas. The crowd of replicas oversees the updates collectively.	An update to the state changes only some partitions. An oversight function with some shared state is integrated with all partitions.	An update to the state changes only some partitions, which oversee their updates directly. No third-party oversight function is involved.

Analysis

In this section, we analyze the five archetypes and compare their privacy, compliance, visibility, scalability, resilience, extensibility, and online and offline payments.

Privacy

Privacy refers to how many details the system entities know about user transactions. It also covers how much one institution knows about the data of other institutions. A strong privacy posture means that user data are visible only to the user and as few institutions as required.

In leaderless systems, all institutions can see the total system state, so the potential for a breach of user privacy is quite high. Further, the data of one institution are visible to other institutions. The default privacy posture of leaderless systems is therefore weak. Efforts have been made to address this with privacy-enhancing technologies such as zero-knowledge proofs. But whether these technologies are ready for mass market deployment at scale and what trade-offs they bring are unclear.

In direct systems, only the parties involved in the transaction record the settlement. The direct archetype therefore provides users with the strongest privacy. Even if such a system requires users to periodically share data with a system entity, users may be able to influence their privacy through behaviour for example, by remaining disconnected for extended periods.

In general, archetypes that allow end users rather than institutions to hold information (the direct and micro-partitioned) can more easily achieve higher privacy.

Compliance

Compliance refers to the ease with which system entities can collect sufficient data to satisfy legal requirements such as anti-money laundering regulations. A strong compliance posture means that the required data can be collected more easily. In contrast, a weak posture means that data collection is difficult or infeasible. Compliance and privacy are policy goals in opposition—the easier one is to achieve, the harder the other is.

In both centralized and leaderless archetypes, operators can access the total system state. In those systems, it is relatively straightforward to include mechanisms in the design for regulators to gather the necessary compliance information.

The lack of an intermediary in the transaction flow makes it difficult to reliably collect the data required for compliance. This is the case in the direct archetype. In other archetypes, the oversight or validation authority could be designed to enforce the required compliance.

Visibility

Visibility refers to the extent to which the central bank can see the state of the CBDC supply. In general, higher visibility allows the central bank to know more about the state of its liability and to exercise stronger oversight.

In systems where operators have access to the total state (the centralized and leaderless), visibility is strong because it is impossible for an entity to modify the state (say, to corrupt the supply of CBDC) without the central bank becoming aware of it. Combining high visibility with the right controls can allow the central bank to enforce strong integrity checks over the supply.

If the transactions are not visible to the central bank and operators cannot see transactions, the central bank may not notice attempts to modify or corrupt the CBDC supply. This is a drawback of the direct archetype, where transactions can be finalized without third parties.

In systems with strong visibility, the technology could potentially do the heavy lifting of ensuring integrity. This lessens the regulator's burden of overseeing private entities. In contrast, in systems with weak visibility, the burden may fall on non-technical means of regulating private entities.

Scalability

Scalability refers to how efficiently the throughput of a system can be increased as the demands on the system grow.

A weakness of the leaderless archetype is that the high degree of replication of state and compute hampers performance. Systems that replicate the state almost always mitigate the overhead by reducing the instance count of validators or using techniques such as sharding to break up the data into smaller chunks (Buterin 2021) to minimize the degree of replication, for example.

Because the direct archetype allows parties to finalize updates without a third party, these systems could in theory scale limitlessly, much like a cash system. In practice, however, some bottlenecks would occur around distribution and renewal and refresh operations, which would place upper bounds on scalability.

It is important to note that operations with certain money representations, such as UTXO, may scale more easily than those with different money representations, such as account balance (Buterin 2016). Therefore, while the choice of architecture influences a system's scalability, the choice of the money representation is also important and must be considered in an overall scalability design. We discuss representations of money later.

Resilience

Resilience refers to a system's ability to avoid loss of service and loss of data if a failure occurs. Cash is considered highly resilient because it can function even during natural disasters. Further, the loss of one person's cash does not compromise the system or affect others' holdings or their ability to carry out transactions.

The leaderless archetype achieves high resilience by replicating system state fully to multiple instances. If a few instances are lost, the system state can be fully recovered from other instances. Further, its resilience is stronger than that of other archetypes because users are not bound to specific MSBs. However, this comes at the cost of weak scalability.

The loss of an individual's information in the direct archetype does not affect the holdings or ability to transact of another. Therefore, like cash, it is highly resilient. Also, this is the only archetype capable of offline settlement (discussed later), which makes it the most resilient to outages such as natural disasters.

The centralized and partitioned-state archetypes employ oversight functions or mechanisms. These functions, if localized logically in one entity, can present single points of failure. If so, they must be designed with adequate redundancy because their loss can halt the system.

Extensibility

Extensibility refers to how the system's basic functionality can be enhanced to support richer services. In retail CBDC systems, it refers to how entities in the private sector could design and offer innovative services on the core CBDC platform.

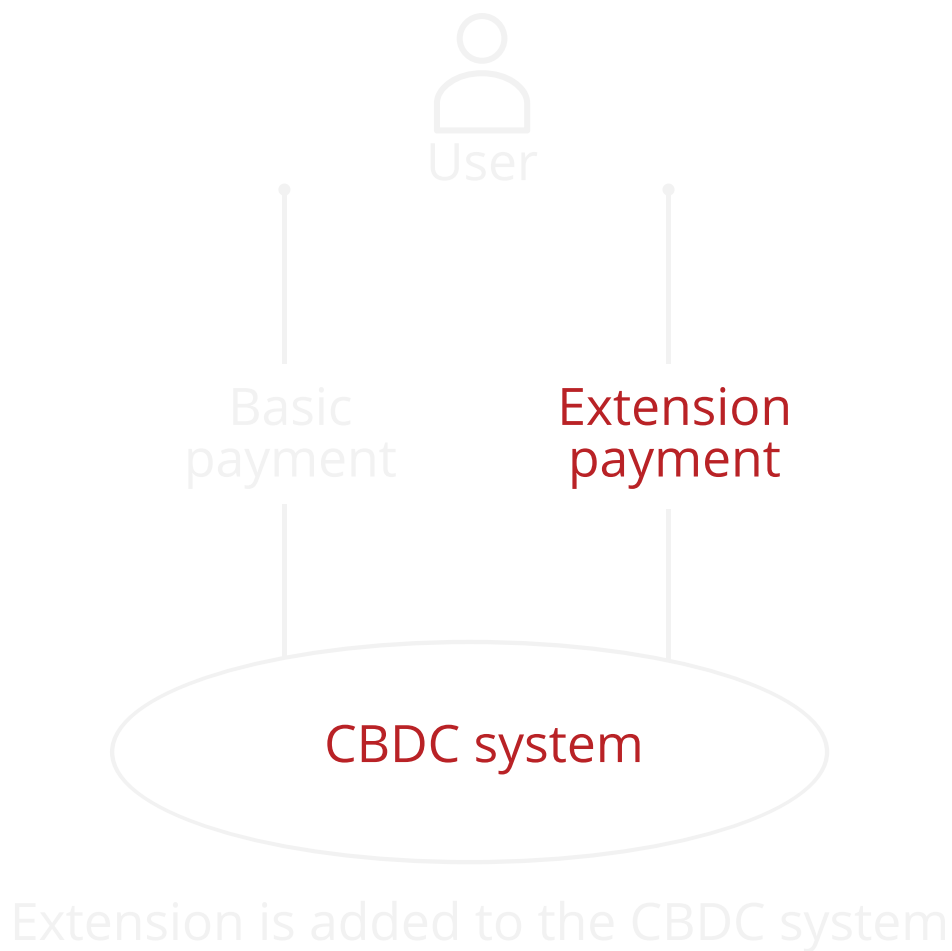
We illustrate this discussion with a simple example. Consider a CBDC system that supports one primitive operation, the basic payment. We seek to extend the system by offering another operation, the loan payment, that combines a basic payment from a lender to a borrower with a second basic payment in the reverse direction a given time later, the reverse amount being the original plus some given interest. We describe two forms of extensibility and discuss them in the context of the archetypes.

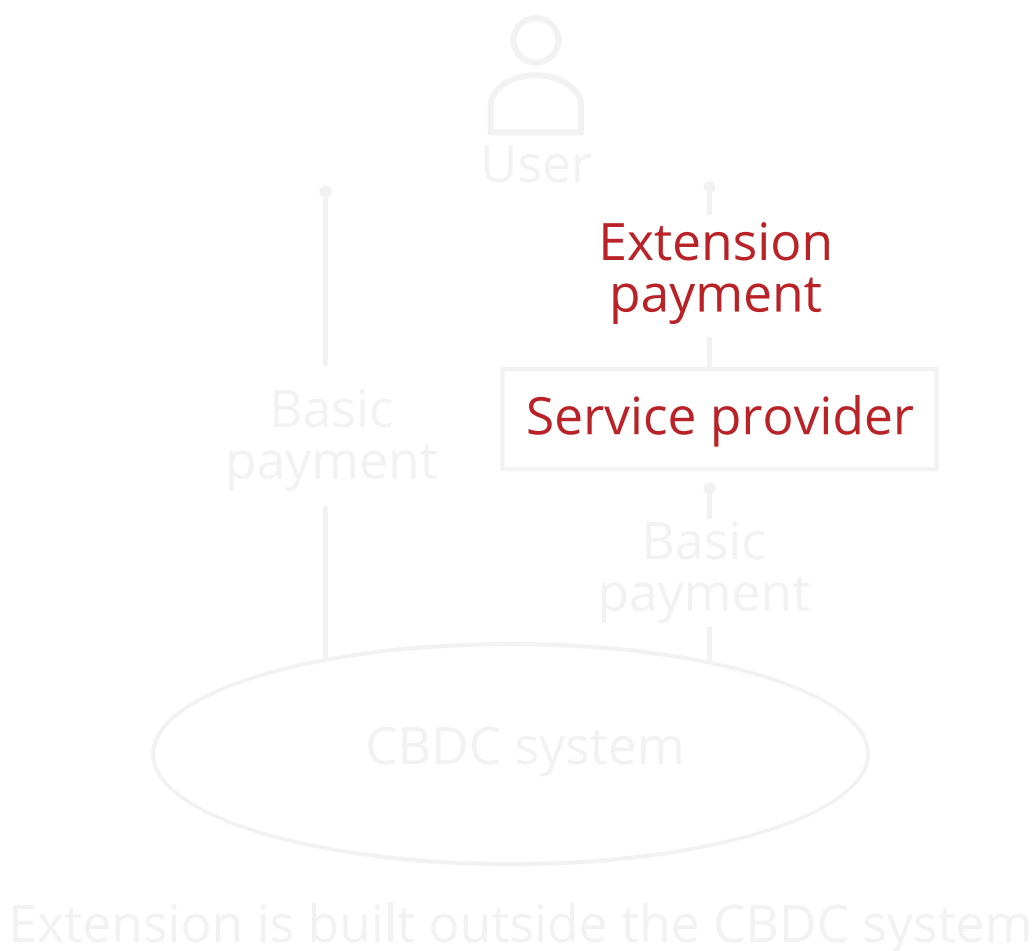
The first form of extensibility is to add the new operation to the CBDC system. In our example, then, the loan payment is added as a second operation to the core CBDC system. The system executes, settles and records all parts of a loan payment as it does for basic payments. Users could query the system for details, for example:

- When is the repayment due?
- What is the repayment amount?

In other words, the CBDC system has complete knowledge of the loan payment extension (i.e., its logic and settings) and guarantees its execution as specified.

Another way is to add the loan payment as an extension built outside the core CBDC system. In the loan payment example, an external service provider would maintain the logic and settings of the loan payment operation. The CBDC system would offer only the basic payment (and some locking or escrow functions, as required). Users would connect to the service provider to participate in a loan. The service provider would perform automated basic payments on the CBDC system for the initial payment and, a specified time later, the repayment. The CBDC system would not be aware of the loan payment operation, so it would be unable to answer queries about the repayment due date or amount. In this model, the CBDC system does not have visibility into extensions and so cannot guarantee their correctness or execution.

Figure 7: Two forms of extensibility



Adding the loan payment operation—the extension—to the CBDC system requires sharing state. Specifically, system entities must become aware of the new payment type, how it is processed (as two basic payments separated in time), its contingencies (a time trigger) and its parameters (the interest rate)—this knowledge is the new shared state. This is best accomplished in archetypes that naturally support sharing state globally: the centralized and the leaderless archetypes.

The macro-partitioned archetype could also support installing extensions. However, more work may be required because the process to settle a transaction updates only some partitions, while the installation of an extension requires updating all partitions.

The second form, building an extension outside the CBDC system, requires programmatic invocation and transacting parties to be always reachable. The archetypes that can ensure that participants' availability is always on—the centralized, leaderless and macro-partitioned—are also well-suited to support extensions outside the CBDC system. They would need to present an application programming interface that allows invocation of primitive CBDC operations. Whether extensions could be supported in archetypes where users hold system state—micro-partitioned and direct—is unclear because guaranteeing user availability is impossible.

From a technical perspective, the first form of extensibility may seem preferable—the CBDC system has full knowledge of extensions and guarantees their behaviour. However, depending on the risk appetite and policy goals of a jurisdiction, separating the extensions from the core CBDC functions may be preferable.

Online payments

We consider two online payment use cases important for a retail CBDC: billboard payments and online retail payments. We assess how well each archetype could support these use cases. Billboard payments, also known as receiver-not-present payments, could occur in scenarios such as stimulus payments or tax refunds from governments to citizens. Online retail payments involve users paying online merchants.

Only archetypes where institutions hold system state (the centralized, leaderless and macro-partitioned archetypes) can ensure that parties are always reachable using a unique identifier such as an account number or address. Only these systems can support billboard payments. Archetypes where users hold system state (micro-partitioned and direct) generally cannot support billboard payments.

The three archetypes that support billboard payments are also best suited for online retail payments. While the other archetypes (micro-partitioned and direct) could support retail payments in theory, in practice, this would require online retailers to invest in new technology to keep custody of their CBDC holdings just to receive CBDC payments. This is almost certainly not viable from a business perspective.

Offline payments

Offline refers to the ability of two parties to transact when connected only to each other and no other party. Researchers at the Bank of Canada suggest two forms of offline capability (Minwalla et al. forthcoming): extended offline (i.e., offline clearing and settlement) and intermittent offline (offline clearing only). In intermittent offline, transactions clear instantly, but funds do not settle until the payee resumes connectivity with a remote service. In extended offline, the funds transfer completes so that the payee can spend the funds received right away without connectivity to a network service.

Extended offline involves completing the transfer of money offline with finality—the payee receives the funds offline. Only the direct archetype achieves offline settlement. Offline settlement enables a cash-like user experience, which is arguably the primary benefit of the direct archetype.

Intermittent offline involves the payee confirming offline that the payer has sufficient funds. The payee lays a claim to a portion of the payer's CBDC funds, to which the payer attests, locking those funds from further spending by the payer (Minwalla et al. forthcoming). The payee would be able to validate the attestation offline but would need to present it online to trigger settlement. Archetypes other than the direct could support intermittent offline if they can:

- lock funds online
- generate proof of locked funds
- store the proof offline securely at the payer's end

These archetypes cannot support extended offline because they require the transacting parties to connect to some third-party entity for settlement.

Ratings

Based on the preceding sections, we assign a qualitative rating to each archetype for each criterion (**Table 2**). The ratings of (***), (**), and (*) align with strong, average and weak postures that the archetype can achieve, respectively, while (-) suggests achieving even the weak posture is difficult.

Table 2: Archetype ratings

Criterion	Archetype				
	Centralized	Leaderless	Macro-partitioned	Micro-partitioned	Direct
Privacy	*	-	*	**	***
Compliance	***	***	**	**	-
Visibility	***	***	**	**	-
Scalability	**	*	**	**	***

Criterion	Archetype				
	Centralized	Leaderless	Macro-partitioned	Micro-partitioned	Direct
Resilience	**	***	**	**	***
Extensibility	***	***	**	*	-
Online payments	***	***	***	*	-
Offline payments	**	**	**	**	***

In practice, multiple system designs will be based on one archetype, even though they vary in their implementations of privacy, extensibility and other aspects. The rating given to an archetype should therefore be interpreted as the typical rating achievable by systems of that archetype, while allowing for variation between systems that results from design choices particular to each system.

The ratings should not be considered fixed; they can be changed through design choices. For example, leaderless systems have a weak privacy rating, but the use of advanced privacy-enhancing techniques could strengthen their privacy. However, such design choices may add complexity or modify the system's posture for other criteria, such as scalability. A weak rating is not necessarily a reason to rule out an archetype; instead, it suggests an area for design investigation.

We observe that no archetype achieves online and cash-like offline payments use cases. Other trade-offs can be discerned, e.g., cash-like offline versus compliance, and extensibility versus offline. These suggest that a system based on a single archetype will not likely satisfy all policy goals of a retail CBDC system. It is more likely that a hybrid system combining aspects of more than one archetype would be needed to meet the central bank's goals.

Representations of money and state

We have described five archetypes based on the fundamental consideration of how state is organized in a retail CBDC system. A closely related and equally important consideration is what that state is—the representation of money and the structure of state.

Money can be represented in several ways, such as account balance (the instrument's value changes while its ownership is fixed), digital bills (the instrument value is fixed while its ownership changes) or UTXOs (the instrument is short-lived and its ownership changes) (Buldas et al. 2021). Further, the structure of state can include aspects such as whether it is append-only or cryptographically immutable. We envision the organization of state as a first-order consideration that gives rise to the archetypes, and the structure of state as a second-order consideration that leads to the specialization of archetypes into a system design.

The UTXO model may allow the origin of money objects to be traced to unrelated parties, affecting their privacy. If Alice pays Bob some money, and Bob later uses that money to pay Charlie, it may be possible to ascertain that the money used in Charlie's transaction originally came from Alice. In this way, different money representations may result in different levels of privacy.

The UTXO money representation comes with fragmentation. As the system ages, a set of UTXOs tends toward a larger and larger number of smaller-value instruments. Over time, as these accumulate, even routine transactions may involve high overhead, such as a \$100 transaction using 1 cent objects, slowing down the system.

Another aspect of performance is the ease of scalability of some representations of money over others, for example, UTXOs scale better than accounts, and the bill-based model has been noted as superior to both UTXO and account models (Buldas et al. 2021). These and other examples suggest that the design choices in the structure of state can influence the performance.

Lastly, some aspects of programmability may be easier to implement in an account model than in a UTXO model (Buterin 2016). We note these as areas for future research.

Conclusions

Our analysis of the five retail CBDC archetypes and their trade-offs suggests that a design based on a single archetype is unlikely to achieve all policy goals. We consider a few possible ways to use and advance this work.

Firstly, CBDC teams can identify archetypes that are closely aligned to their policy goals. This allows them to focus their efforts on classes of systems that fit those archetypes. Secondly, the trade-offs we identify point to possible areas for technical experimentation, including privacy and scalability of leaderless systems, safe implementations of extensibility, and compliance and security in direct systems. Thirdly, an important extension to this work is understanding how different representations of money can be combined with the archetypes and their impact on the criteria set out in **Table 2**. Fourthly, the set of criteria could be expanded, e.g., cross-border payments is an important area. We leave it to future work to investigate whether some archetypes are better suited to cross-border payments than others. Lastly, given the large variety of retail CBDC system designs in both industry and academia, these archetypes provide a common framework and terminology for technical teams to analyze and evaluate designs.

We trust that these archetypes will be useful to central banks considering the issuance of a retail CBDC instrument.

Endnotes

1. For example, state could be held in wholesale accounts of a real-time gross settlement system.[↔]
2. For example, the NoSQL database Cassandra permits *eventual* consistency, where the data returned on a read operation will eventually be the most recently written value. Until then, different parts of the system may return, on a read operation, a value other than the most recent. We note that anything other than fast global consistency is a poor fit for CBDC systems.[↔]
3. Archetypes in CBDC systems are presented in the same spirit as the widely known design patterns in software engineering (Gamma et al. 1994), in that both represent recurring ideas in their respective fields. No familiarity with design patterns is required to understand archetypes.[↔]
4. Incidentally, minting of the Marshall Islands CBDC is fixed in legislation at a “predefined annual growth rate” rather than being in the central bank’s sole control. (Republic of the Marshall Islands 2018)[↔]
5. The Notary in an R3 Corda (Hearn and Brown 2019) network is an example of such an oversight component.[↔]
6. The TODAQ (Gravitis, Goh and Toliver 2019) protocol to compute a global Cycle Trie root in an update cycle is an example of a third-party oversight mechanism.[↔]
7. To be clear, the trust is not provided by the partition operators (end users or institutions). Instead, trust is provided by the secure protocols and execution environments within which the partitions operate, and, ultimately the authority (such as the central bank) that approves those environments and protocols.[↔]
8. An implementation of the direct archetype that does not require secure hardware has been proposed—money based on unclonable quantum states (Aaronson and Christiano 2012). While the idea is intriguing, the technology is unproven.[↔]

Acknowledgements

I would like to thank members of the CBDC program at the Bank of Canada for their comments, in particular Rakesh Arora for privacy and compliance, Han Du for programmability and Cyrus Minwalla for offline payments. Special thanks to Rakesh Arora and Dinesh Shah for many insightful discussions to refine these ideas.

References

Aaronson, S. and P. Christiano. 2012. "Quantum Money from Hidden Subspaces." *STOC '12: Proceedings of the Forty-Fourth Annual ACM Symposium on the Theory of Computing*. New York: ACM. 41–60.

Androulaki, E., A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. A. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. Weed Cocco and J. Yellick. 2018. "**Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains.**"

Buldas, A., M. Saarepera, J. Steiner, L. Ilves, R. Olt and T. Meidla. 2021. ***A Formal Model of Money Schemes and Their Implications for Central Bank Digital Currencies.***

Buterin, V. 2014. "**Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform.**"

Buterin, V. 2016. "**Thoughts on UTXO.**"

Buterin, V. 2021. "**Why Sharding is Great: Demystifying the Technical Properties.**"

Cachin, C. and M. Vucolić. 2017. "Blockchain Consensus Protocols in the Wild." *31st International Symposium on Distributed Computing (DISC 2017)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 1:1–1:16.

Chen, J. and S. Micali. 2017. "**Algorand**" version 9.

Eesti Pank. 2021. "**Eesti Pank Ran an Experiment to Investigate the Technological Possibilities of a Central Bank Digital Currency Based on Blockchain.**" December 13.

Gamma, E., R. Helm, R. Johnson and J. Vlissides. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Germany: Addison-Wesley Professional.

Gilbert, S. and N. Lynch. 2002. "Brewer's Conjecture and the Feasibility of Consistent Available Partition-Tolerant Web Services." *ACM Sigact News* 33 (2): 51–59.

Gravitis, A., N. Goh and D. Toliver. 2019. "**TODA Primer: The Promise of Blockchain**." TODAQ Holdings Inc.

Hearn, M. and R. Gendal Brown. 2019. "Corda: A Distributed Ledger."

Minwalla, C., J. Miedema, S. Hernandez and A. Sutton-Lalani. Forthcoming. "Designing a CBDC for Offline Payments." Bank of Canada Staff Analytical Note.

MIT Digital Currency Initiative. 2022. "**OpenCBDC**."

Nakamoto, S. 2009. "Bitcoin: A Peer-to-Peer Electronic Cash System."

Republic of the Marshall Islands. 2018. "**Declaration and Issuance of the Sovereign Currency Act 2018**." SOV. March 1. Accessed August 31, 2022.

Disclaimer

Bank of Canada staff analytical notes are short articles that focus on topical issues relevant to the current economic and financial context, produced independently from the Bank's Governing Council. This work may support or challenge prevailing policy orthodoxy. Therefore, the views expressed in this note are solely those of the authors and may differ from official Bank of Canada views. No responsibility for them should be attributed to the Bank.

Content Type(s): **Staff research, Staff analytical notes**

Topic(s): **Central bank research, Digital currencies and fintech**

JEL Code(s): **E, E4, E42, E5, E51, O, O3**

DOI: <https://doi.org/10.34989/san-2022-14>