STAFF
REPORTS

NO. 1180
FEBRUARY 2026

Programming Money Without Programmable Money

Michael Junho Lee | Antoine Martin

FEDERAL RESERVE BANK *of* NEW YORK

Programming Money Without Programmable Money

Michael Junho Lee and Antoine Martin

Federal Reserve Bank of New York Staff Reports, no. 1180
February 2026
https://doi.org/10.59576/sr.1180

Abstract

Programmability is at the heart of ongoing work on the future of money and payments by central banks around the world. Despite its potential, there is growing concern that programmability conflicts with the provision of "good" money. This paper overviews key principles of "good" money and argues that the discourse on programmability inadequately differentiates between programmable money, which is generally negatively viewed, and programmable payments, which is generally accepted as part of the future. We provide a framework for programmable monetary systems that sharply distinguishes between programmable money and programmable payments. We show that our framework nests a broader set of financial arrangements and revisit the debate on programmability in the design of monetary systems.

JEL classification: E42, E58, G28
Key words: digital money, programmability, payments, monetary systems

---

Lee: Federal Reserve Bank of New York (email: michaeljunholee@gmail.com). Martin: Swiss National Bank (email: antoine.martin@snb.ch).

This paper presents preliminary findings and is being distributed to economists and other interested readers solely to stimulate discussion and elicit comments. The views expressed in this paper are those of the author(s) and do not necessarily reflect the position of the Federal Reserve Bank of New York, the Federal Reserve System, or Swiss National Bank. Any errors or omissions are the responsibility of the author(s).

To view the authors' disclosure statements, visit
https://www.newyorkfed.org/research/staff_reports/sr.html.

1 Introduction

Innovations in money and payments have brought forth the potential for enabling programmability. This could support a wide range of arrangements and transactions that automatically execute and settle when certain conditions are met. As governments, central banks, and private companies around the world explore the development of tokenized money, questions have emerged on how exactly programmability can be incorporated into a monetary system.

An emerging question is whether countries should consider a system with programmable payments, programmable money, both, or neither? For example, the Riksbank distinguishes between programmable payments and programmable money in its report on the third phase of the e-Krona project, and argues that programmable payments may be desirable but not programmable money. The report reasons that money that has been programmed may lose some of the basic functions of money, notably being exchangeable with other forms of money. The Riskbank's intended goal is to build a payment system in which programmability is enabled, but to avoid the circulation of different types of tokenized money that do not trade at par. Relatedly, in a speech in January 2023, Fabio Panetta, member of the ECB's Executive Board also made the distinction between programmable payments and money. He stated: "Let me be clear, the digital euro would never be programmable money. The ECB would not set any limitations on where, when or to whom people can pay with a digital euro. That would be tantamount to a voucher. And central banks issue money, not vouchers."

Our paper makes three points. First, we review the underlying principles for the implementation of programmability for tokenized money. Second, we argue that the distinction between programmable payments and programmable money made by academics and policymakers alike is blurry at best. So, how exactly are programmable payments different from programmable money? Third, we propose a framework that allows us to distinguish more precisely between programmable payments and programmable money. We use this framework to review some arguments made about programmable money.

2 Qualities of Good Money

To understand why we may care about programmable money, it is useful to briefly review the qualities of good money. A key quality of money is that it be accepted "no questions asked" (NQA). This idea, introduced in Holmstrom (2015), is also discussed in Gorton, Ross and Ross (2022). Money satisfies the NQA property if households and merchants don't have to be concerned about the value of the money, about where and whether it will be accepted. In particular, no effort or due diligence is required to ascertain the value of money that satisfies NQA. Assets can be more or less "money-like" depending on how close they are to NQA Gorton et al. (2022) consider empirical measures of the distance to NQA).

The NQA property is related to two properties of "good" money, uniformity and fungibility. Uniformity of money refers to the property whereby different individual units of money have the same value. For example, a 10 dollar bill, regardless of the physical condition of the currency, will be accepted for goods and services worth 10 dollars in value by any vendor. Similarly, a 10 dollar bill has the same value as 10 dollars in a bank account. This uniformity allows currency to facilitate efficient payments without users and potential receivers being concerned about the value of money. Hence, money that doesn't have the property of uniformity would also fail to have the NQA property.

Fungibility of money refers to the property whereby individual units, as well as parts, of money are interchangeable. Continuing with the example, 10 dollars in a single bill is equivalent to 10 dollars in 10 one-dollar bills. Fungibility is also important in the context of accounts holding money. Each dollar in an account holding 10 dollars is indistinguishable from one another, and no single "part" of the 10 dollars are treated differently. Fungibility allows currencies to be valued not by their representation, but by the value that they represent. As with uniformity, money that doesn't satisfy the fungibility property doesn't satisfy NQA.

A new monetary system with programmability may provide the infrastructure necessary to support the future of the financial system and foster innovation. It has, however, raised reasonable concerns that programmability may undermine the NQA property. In particular, a government might be concerned that programmable money could circulate at different prices than "regular" money and could command different levels of liquidity. For example, before accepting money that can only be used to purchase specific goods and services, or can only be

used under specific conditions, households and businesses may want to undertake some costly due diligence, which would undermine NQA. To further investigate these concerns, we now turn to a framework designed to help us think about programmability.

3 A framework to think about programmability

Our framework aims to distinguish between different types of programmability. A digital monetary system with programmability consists of two layers: an "asset" layer and a "program" layer. We consider these layers from a conceptual perspective, rather than a technological one. In particular, we do not take a stand on whether these layers are separate or not (this topic is discussed in Aldasoro, Doerr, Gambacorta, Garratt and Wilkens (2023)). We are also agnostic about the underlying technology, which could in principle be based on a blockchain, distributed ledger, or a standard database.

The asset layer can be thought of as a ledger that records the ownership of the asset. For example, the asset layer could be the set of all Fedwire accounts or the set of all Bitcoin addresses. The set of deposit accounts at depository institutions also constitutes an asset layer, even if those deposit accounts reside on the private ledgers of different institutions.

The program layer is the universe of programs that users can potentially access to manipulate the behavior of assets programmatically. A program represents an instruction to change the allocation of control and/or balances from one account to another under certain conditions. This includes simple programs, such as those that transfer ownership (e.g. payments), and more sophisticated programs, such as those that change ownership conditional on some states (e.g. insurance). For example, send $10 from account A to account B if condition X is satisfied, where condition X could be "it rains in NYC on Thursday". The program layer includes all allowable programs. An example of a common type of program that already exist is direct debit arrangements, such as the ability to pay one's rent or utility bills automatically every month.

What are "payments" and "money" in this framework? Money is the asset recorded in the asset layer. It is a stock. In this paper, the asset recorded is "plain-vanilla" money, either issued by the central bank, as in the case of Fedwire, or readily convertible into such money, in the case of deposit accounts, since they

allow depositors to obtain currency. In particular, the money recorded on the asset layer is not programmed and, thus, satisfies the NQA property. A payment is the transfer of money from one account to another in the asset layer. It is a flow. Programs that transfer money are recorded on the program layer.

Defining programmable money as "money that is programmed to behave in a certain way" is problematic. Under such a definition, any activity in a monetary system with programmability capabilities involves programmable money. This definition leaves no room to differentiate between programmable money and programmable payment. In addition, under this definition, programmable money is already a reality, for example in the form of direct debit arrangements, which is clearly not what economists are concerned about when they think about programmable money.

To make progress, we find it useful to consider properties of programs that might be applied to the asset layer. This allows us to think more systematically about the type of programs that could impact whether money satisfies the NQA property. This reflects our view that the key question is not whether "programming" money should be feasible (after all, all programs "program" money), but rather which types of programs would be desirable in a monetary system.

4 A classification of programmability

In this section, we propose a taxonomy for classifying different types of programs. This allows us to be more precise about what functionalities would be desirable, depending on the objective of a system designer.

To make things concrete, we continue with the two-layer system described above. The asset we consider is "plain-vanilla" money, which we call "basic" money. We start by considering an implementation that involves "pre-paying" into programs for future exchange, as is the case in Kahn and Van Oordt (2022). This specific implementation allows us to sidestep questions regarding credit risk. Pre-payment is not a general requirement, however, and we consider a system with credit in the next section.

We now describe one implementation through which basic money can be programmed:

1. Basic money is "locked" in a program.

2. The program specifies the conditions under which basic money is released.[1]

3. The program issues a "certificate." The certificate is a new asset that can potentially be used in exchange.

We start with an example to make things concrete: a program that enables basic money to be used only for a specific purpose. Under step 2, the program could specify that the basic money locked in the program will be released only to a merchant selling coffee, but not a merchant selling other goods or services, like hot dogs. The ownership and control of the program, and the basic money that is locked in it, can be represented as a new asset, which we call a certificate. Importantly, the certificate grants its owner a set of rights and determines feasible actions allowed under the program.

An analogy can be drawn to prepaid gift cards, which satisfy the steps described above for programs. The physical card would be the certificate; the holder of the physical card can use the card for certain purchases, say, coffee. In this case, since the seller receives basic money in exchange for the cup of coffee, we assume that a seller is indifferent between being paid in basic money or with the gift card. The holder of the physical card can also, in principle, exchange the card for some other goods or services, such as the purchase of a hot dog. In this case, the seller of the hot dog would not receive basic money but, instead, a gift card that can be used to purchase coffee. For that reason, the seller may not accept the card at face value. The seller, as the new owner of the card, can then use it and is bound by the rules limiting the use of the card.

Classifying programmability boils down to defining the features that the program enables. We discuss two: transferability and convertibility. We use transferability to refer to a program in which the ownership of the certificate can be transferred. With the transfer of ownership of the certificate, the new owner inherits the rights designated by the program. A gift card is transferable; the card, and the ability to use it, can be passed on to anyone. Put another way, transferability means that the program is a "bearer" instrument. Having the certificate is enough to be able to execute the program. A certificate could be "nontransferable," if use of the certificate is limited by the program. An example would be a gift card that

[1] If the amounts of funds locked in a certificate exceeds the price of a purchase, then the certificate's new available amount equals the difference between the original amount and the price of the purchase. This is similar to the practice with gift cards.

can only be used by a specific individual who would need to show proof of ID before being able to use the card. Note that if a certificate is transferable, then there must be an asset layer that keeps track of the ownership of certificates. In principle, there could also be programs that specify conditions under which certificates can be used. We briefly discuss those below.

We use convertibility to refer to a program in which certain uses of the certificate release basic money. In other words, when the conditions specified in the program are satisfied, the certificate is converted into basic money. A gift card is convertible; when the card is used for the permitted purpose (e.g. retail shop, etc.), the vendor receives basic money, not the gift card itself, and can spend that money on anything they want. A certificate is not convertible if the conditions imposed on the money by the program remain. For example, if the owner of a non-convertible certificate purchases a coffee, the barista would receive a certificate that can only be used to purchase coffee. The owner of a hot dog stand would have no obligation to accept the certificate in exchange for a hot dog.[2] A non-convertible certificate would never release basic money under any circumstances.

5 Programmable money vs programmable payments

In our framework, "money" is an asset represented on the asset layer that can be transferred in exchange for goods or services. Hence, any transferable certificate is potentially programmable money. It is programmable, by virtue of being a certificate and it is money if and when it is transferred in exchange for goods and services. In our framework, a "payment" is defined to be a (convertible) transfer of basic money. That is, the payee obtains basic money, either from the transfer of basic money, or from a certificate that is converted to basic money. The main distinction is that with programmable money, the asset that is sent is the same as the asset that is received – with programmable payments, the receiver receives basic money.

Note that if the certificate itself is exchanged for good or services, it constitutes a payment but not a programmable payment. In this case, the seller would accept

[2]Here we assume that the barista is required to accept the certificate. If there are no requirements or incentives for coffee sellers to accept non-convertible certificate that give the right to purchase a coffee, then the restrictions of the program are meaningless, in the sense that the certificate could be accepted or rejected with equal likelihood everywhere.

the certificate as payment for the goods or services offered regardless of the conditions embedded in the program. And, because by assumptions those conditions are not met, the seller doesn't receive basic money. Continuing our example, this would be akin to purchasing a hot dog in exchange for a coffee gift card.

Similarly, any convertible certificate is potentially a programmable payment. It is a payment whenever the conditions for convertibility are satisfied, and the seller receives basic money. This is because satisfying the conditions of the program leads to the "programmed" transfer of basic money to the seller's account. In turn, the seller is now able to enter any future transfers using money unencumbered by the nature of its origins.

In our framework, a certificate is "pure" programmable money if it is non-convertible but transferable. An example would be a certificate that can only be used to purchase coffee and never releases basic money (non-convertible) but can be given to anyone, i.e., is a bearer instrument (transferable). The money backing the program is held in perpetuity in the program, and the program does not allow any receiver to convert the certificate into basic money. These certificates could circulate in perpetuity, as a form of money used only in certain circumstances and/or to purchase specific goods and services. In a financial context, certificates could also be a share of a closed-end fund or a company, which could be purchased and sold to third parties, but not used to redeem value as with an open-ended mutual fund.

A certificate is "pure" programmable payment if it is non-transferable but convertible. Under certain circumstances, like the purchase of a coffee, its use endows the receiver with basic money. The basic money received by the barista can then be used without restriction. However, the certificate itself cannot be transferred to anyone else and can only be used by the person to whom it was issued. In a financial context, direct debit arrangements described before are examples of pure programmable payments.

A transferable and convertible certificate represents a hybrid that could be either programmable payment or programmable money, depending on its use. If the certificate is used such that the conditions of convertibility are satisfied and the seller receives basic money, then it is a payment. If, instead, it is used such that the conditions of convpertibility are not satisfied, and the seller receives the certificate, then it is money.

We think of transferability as an intrinsic property of money. It relates to the

	Convertible	Non-Convertible
Transferable	Hybrid, potentially programmable money or payment. *Examples:* Prepaid cash card, Gift card	Pure programmable money *Examples:* Closed-end fund shares, securities
Non-Transferable	Pure programmable payment. *Example:* Direct debit arrangements	Neither money nor payment. *Example:* Non-transferable ownership share

Table 1: Classification by Programmability.

ability of money to be exchanged between users or, in other words, to effectuate a payment. In contrast, convertibility is an intrinsic property of payments. Indeed, the transfer that occurs when the conditions of the program are met adds basic money to the receiver's account, which can then be used to make unrestricted transactions.

Finally, an object that is non-transferable and non-convertible is neither money nor (programmable) payment. Table 1 summarizes this discussion and presents the 4 different types of certificates that can arise from considering transferability and convertibility.

Higher order certificates

Certificate themselves could be programmed. In this case, following the steps described in Section 5, the program would issue a new certificate. We can think of this certificate as a 2nd degree certificate. If such a certificate is convertible, it would release the original (1st degree) certificate, when the conditions of the program are met, rather than basic money. The 1st degree certificate, if it is convertible, could release basic money when the conditions of its program, which could be different from the conditions for convertibility of the 2nd degree certificate, are met.

As noted above, an nth degree certificate that is transferable and convertible could be used either as programmable money or programmable payment. If the conditions of the program are met and the certificate is converted into the certificate of degree n-1, then it is a programmable payment. If, instead, the conditions of the nth degree certificate are not met, but the certificate is exchange for goods and services, as part of a nonprogrammable payment, then the certificate is used as programmable money.

Certificate issuance

Which entities are allowed to issue new certificates, and under which conditions, is potentially an important consideration. While a full treatment of this question is beyond the scope of this paper, we briefly discuss the issue.

A conservative approach is to restrict the right to issue certificates to specific institutions, such as the central bank as well as perhaps some other official sector institutions. Such restriction would limit the circulation of certificates that are either badly designed and/or poorly understood, which could have unintended consequences.

At the opposite side of the spectrum, issuance of certificates could be allowed for anyone. This would likely maximize the potential benefits of innovation that programmability could provide. But this could come at the cost of problems such as faulty certificates or fraud, as illustrated by the world of DeFi.

An intermediate approach could be to allow different sets of institutions to issue different types of certificates. The type of certificates could depend on whether they are transferable, convertible, and whether they are pre-paid, among other. Authorization to issue certificate that are pre-paid, nontransferable, and involving simple programs, could be granted to a large set of institutions and individuals. Authorization to issue of other types of certificates, particularly transferable ones since they have the potential to be used as money, could be limited to a smaller set.

6 Open certificates

So far, we have considered only "pre-paid" certificates, which are not subject to credit or liquidity risk. A plethora of possible certificates do not meet this restriction. We call them "open" certificates. We start by considering how the steps of the implementation described in Section 5 need to be modified. First, instead of locking basic money into the program, the first step would specify an account from which basic money would be debited when the conditions of the program are satisfied. Second, because the certificate is not prepaid, the program would also need to specify the amount of basic money that is released when the conditions of the program are met. The exact amount does not need to be determined at the time the program is created, but clear instructions that unambiguously determine that

amount are necessary, related to the concept of "complete programs" described in Lee, Martin and Townsend (2024). The steps become:

1. The account from which basic money is debited when the conditions of the program are met is specified.

2. The program specifies the conditions under which basic money is released and the exact amount that is released.

3. The program issues a certificate.

A variety of common financial instruments resemble open certificates. We provide some examples. A direct debit arrangement is a convertible, nontransferable, open certificate. Such an arrangement can specify that a credit card's balance is paid in full every month. The corresponding program specifies the account from which balances are debited, say a checking account, the conditions under which this occurs, the due date, and the amount that will be debited, the full balance. Since such arrangements are convertible and nontransferable, they are pure programmable payments in our framework.

A Treasury security is a convertible, transferable, open certificate. It is convertible, since at expiration of the security, basic money is released to its owner. It is transferable since it can be bought or sold. It is open since the Treasury's account isn't debited until the maturity date. In our framework, Treasury securities would be considered as a hybrid, which can represent a programmable payment or programmable money depending on how it is used. For a buy-and-hold investors, a Treasury security is akin to a programmable payment, that provides basic money at a specific date. While Treasury securities are not typically used to purchase goods and services, for an institutional investor they could be considered programmable money if they are sold or repo-ed.

Finally, equity resembles a nonconvertible, transferable and open certificate. Equity is not convertible because there are no conditions that could trigger a principal payment.[3] It is transferable. Finally, it is open, in the sense that no basic money is locked in.

[3]The payment of dividends should not be considered convertibility, since such payments do not modify the security; e.g., reduce the principal amount.

7 Should central banks be concerned about programmable money?

With our framework in place, we can ask whether central banks should be concerned about programmable money. Of course, the answer to that question could depend on the type of concerns a central bank may have. One concern could be that programmable money is unlikely to satisfy the NQA property or, as stated by the Riksbank, trade "at par". While being transferable is a necessary condition for an asset to be money, there is considerable evidence that it is not a sufficient condition. For example, some of the safest securities, such as AAA bonds, are transferable but don't trade as money, in part because they carry some credit risk and trade in markets that can be illiquid. Of course, markets view AAA bonds as more money-like than "junk" bonds. AAA bonds are less informationally sensitive, and valuing a AAA bond involves less due diligence than a junk bond.

This provides some potential lessons for programmable money. Whether transferable certificates are used as money would likely depend in large part on the due diligence necessary to understand whether the certificates would be accepted by others and at what price. The cost of that due diligence is likely to increase with the complexity of the program underlying the certificate. And, even if a transferable certificate is used as money, the example of Treasury securities suggests that it is unlikely that such certificate would trade at par. In that sense, the concerns of some central banks appear to be warranted.

That said, as noted above, Treasury securities satisfy our definition of programmable money and are a close substitute for money, at least for institutional financial firms. This doesn't seem to cause a problem. Moreover, a number of central banks have the authority to issue Bills, and some do. Issuing such instruments, which are arguable just as close substitutes for money as government securities, does not appear to create an issue. This suggests that maybe issuing programmable money is not a big concern. Or perhaps a finer distinction needs to be made between different types of programmable monies, some of which may create problems while others do not.

A related reason why issuing programmable money may not be a concern is that just because an asset is transferable, and thus potentially money, doesn't mean that it will be used as money. Programmable money may not circulate as

money because of concerns about trading with better informed counterparties. At the same time, a money system that bars transferable programs may prohibit desirable applications and innovation. Transferability may be key to certain financial arrangements, in which basic money is used to create synthetic assets and contracts that can then be collateralized or traded. If this were to be true, it might be desirable for central banks to allow for transferable programs despite their concerns that these could potentially be used as programmable money. An important question for central banks to consider, in that regard, is whether there are benefits of programmable money that cannot be achieved with programmable payments?

Finally, it might be worth noting that if central banks are primarily concerned with making sure their liabilities maintain a constant relative value, then programmable money may not have any benefits. Indeed, traditionally central banks have maintained the relative value of their liabilities by exchanging them inelastically 1 for 1. For example, banks can exchange currency for reserves and vice versa. A central bank could maintain parity between basic money and programmable money it issues by doing the same thing. While this would preserve the NQA property, it would seem to undo any potential benefit from programmable money, since it would always be possible to exchange programmable money for basic money, which doesn't have any restrictions, at par.

8 Conclusion

We have developed a framework to distinguish between programmable money and programmable payment. Our framework suggests that programmable money may not create serious concerns. There already exists assets that satisfy our definition of programmable money that do not seem to be creating problems.

That said, the main reason why concerns may be limited is that programmable money would be unlikely to satisfy the NQA property and, for that reason, would likely not be used as money.

References

Aldasoro, Iñaki, Sebastian Doerr, Leonardo Gambacorta, Rodney Garratt, and Priscilla Koo Wilkens, "The tokenisation continuum," Technical Report, Bank for International Settlements 2023.

Gorton, Gary B, Chase P Ross, and Sharon Y Ross, "Making money," Technical Report, National Bureau of Economic Research 2022.

Holmstrom, Bengt, "Understanding the role of debt in the financial system," 2015.

Kahn, Charles M and Maarten RC Van Oordt, "The demand for programmable payments," Technical Report, Tinbergen Institute 2022.

Lee, Michael Junho, Antoine Martin, and Robert M Townsend, "Zero settlement risk token systems," Technical Report, Staff Reports 2024.