Grassroots Bonds: Financing by the People, for the People (Full Version)

EHUD SHAPIRO, London School of Economics and Weizmann Institute of Science, UK and Israel

Grassroots currencies turn mutual trust into liquidity: a grassroots coin is a unit of its issuer's debt, backed by the issuer's goods and services, which the issuer must redeem, 1-for-1, against any coin they hold; liquidity arises from mutual credit lines, formed by the voluntary exchange of coins among persons who trust each other. As coins are redeemable 1-for-1, the exchange must be 1-for-1 as well, lest prompt redemption after it leave one party with undue profit. Thus, grassroots coins are incongruent with interest-bearing credit.

Here, we extend grassroots currencies to include also grassroots bonds, units of their issuer's debt due at a later date. Upon maturity, the bearer of the bond may redeem it against a coin of the issuer, so liquid coins can be lent against interest-bearing bonds. We specify bonds by adding Date and Escrow clauses to the social contract of grassroots currencies: the contract enforces the redemption of a bond when it is mature according to the date stated by the issuer, who undertakes to keep it current.

We show that the voluntary swap of coins and bonds can realise the basic financial instruments: loans, sale of debt, and forward contracts, and, with an escrow agent, loans with payment schedules, options, collateral, guarantees, insurance, credit default swaps, letters of credit, and credit lines. We extend the liquidity ratios of corporate finance to bonds, with maturity as asset class, and prove that a community clears its debts by redemptions exactly when no member owes more than they hold, and that it does so without coordination once dates advance and persons act on their rights.

Grassroots currencies that include coins and bonds are implemented in GLP, a concurrent logic programming language running on Dart, as a program derived from the contract and demonstrated by a village market of six agents and an escrow agent.

CCS Concepts: • **Applied computing** → **Digital cash**; **Economics**; • **Computer systems organization** → *Peer-to-peer architectures.*

Additional Key Words and Phrases: Grassroots bonds, Grassroots currencies, Mutual credit, Financial instruments, Social contracts, Liquidity, Peer-to-peer credit

Note. Grassroots coins, and the contract, results, and implementation of grassroots currencies that this paper extends, are the subject of a separate paper [24].

1 Introduction

Grassroots currencies. Grassroots coins [24] are units of debt that can be issued and traded digitally by any person—natural or legal—including people, communities, cooperatives, corporations, banks, municipalities and governments. A coin is backed by the goods and services of its issuer, priced in their own coins, and by the coins they hold: upon request, a person must redeem any coin they have issued, 1-for-1, against any coin they hold. Liquidity arises from mutual credit lines, formed by the voluntary exchange of coins among persons who know and trust each other, and coin redemption pegs mutually-liquid currencies at par and carries payments along chains of holdings. Grassroots currencies aim to provide a foundation for grassroots digital economies, allowing them to emerge without initial capital or external credit, operating solely on the networked smartphones of their members [24].

Coins are not interest-friendly. Since grassroots coins can be redeemed 1-for-1, mutual credit must also be 1-for-1, lest closing it right after formation would leave the parties with undue profit or loss. Concretely, if p opens a mutual line of credit with q with an upside for p, e.g. 100 p-coins against 110 q-coins, then coin redemption would allow p to immediately redeem the 100 q-coins they hold, leaving p with 10 q-coins and q with nothing; this is as if a bank could grant a loan with an upfront interest payment then immediately recall the loan while keeping the interest. More

Credit on the Web

Credit on the Web is provided by third parties—banks, payment and lending platforms, and the validators of decentralised-finance protocols on blockchains—through whom every transaction passes. Here we present credit on the Web provided without third parties, by mutual trust: each person issues their own coins and bonds on their smartphone, and a loan, a sale of debt or an option is a transaction among its parties. We specify this credit market by a social contract, analyse its liquidity and solvency, and implement it as a smartphone app.

generally, 1-for-1 redemption precludes associating interest with mutual credit lines denominated in grassroots currencies.

Grassroots bonds. In this paper we extend grassroots currencies to include also grassroots bonds, units of their issuer's debt due at a later date: upon maturity, the bearer of a bond may redeem it against a coin of the issuer, so liquid coins can be lent against interest-bearing bonds. Redemption generalises the coin-for-coin redemption of grassroots currencies, the bearer of a coin taking any coin or bond its issuer holds. We show that the voluntary swap of coins and bonds can realise the basic financial instruments: loans, sale of debt, and forward contracts (Section 4), and, with an escrow agent, loans with payment schedules, options, collateral, guarantees, insurance, credit default swaps, letters of credit, and credit lines (Section 5 and Appendix C); and that the liquidity ratios of corporate finance apply to grassroots bonds, with maturity as asset class (Section 6).Those of Section 5 employ an escrow agent, trusted by the parties for that agreement, and, where the outcome turns on an outside event, an oracle attesting it; each construction states the trust it requires.

Grassroots bonds with pen and paper. Grassroots currencies were introduced as cryptocurrencies and reintroduced as a physical currency, realised with pen and paper, a thought experiment that simplifies their exposition and uncovers their inherent properties as a currency [24]; we follow it.A *paper bond* is a paper coin with a maturity date written on it: a promissory note stating its issuer, its maturity date, and three standing obligations of its issuer—to price their offerings in their own coins and accept them in payment at these prices; to price every coin and bond they hold, 1-for-1, against their own coins, on standing offer, with redemption the bearer exercising the offer; and to redeem the bond as a coin once it has matured. Each person keeps a calendar, and a bond matures by the calendar of its issuer; a bond is transferable at any time, is not paid, and once it is mature its bearer redeems it as a coin (Section 2).

Specification by a grassroots social contract, and the issuer's date. We specify grassroots bonds by a *grassroots social contract* [10] that extends the contract of grassroots currencies [24] with two clauses: Date, by which a party keeps their date at the current date; and Escrow, by which a party deposits coins and bonds with a second party for a third (Section 3). The clauses that describe acts are written as act schemas, from which the volition-guarded multiagent atomic transactions [17] realising the contract are compiled, and each clause is enforced, attested, or undertaken by the contract [10].The four acts of grassroots currencies are enforced—the runs determine whether their clauses are kept and every correct run keeps them—and the pricing undertaking is witnessed by every coin in another's hands. The Date clause is undertaken: in an asynchronous system no record can attest the date a party keeps. Whether a bond is mature is subject to its issuer advancing their date. So the contract enforces the redemption of a bond only against the date its issuer keeps, and the obligation of a bond as a whole—to redeem it once its maturity date has come—is undertaken. A coin's redemption is enforced by the contract; a bond's obligation is enforced by the contract once its issuer's date has reached its maturity, and before that by the law, on the record the contract

keeps. One contract has two realisations, with pen and paper and digital, under which the results hold alike [24]; here we give what dates add to each.

Solvency and clearing. We prove that a community closed but for one member—e.g., a community bank—clears its debts by redemptions exactly when no other member holds fewer coins and bonds than they owe, of all maturities; and that when it can, members who act on their rights do, in any order, without coordination, once dates advance (Section 6).Clearing with bonds needs the dates to advance, as a bond returns to its issuer only as a coin: the possibility result quantifies over what can happen, and the liveness result over computations in which every member's date advances beyond any bound and members who can redeem eventually do. What a group can obtain by its own rights now, and whether a redemption run is met, concern coins, and the results of grassroots currencies [24] apply to the coins among the bonds unchanged.

Implementation in GLP. We implement grassroots bonds in GLP [25], a multiagent concurrent logic programming language running on Dart for smartphone deployment, as a program derived from the contract, transaction by transaction.What dates add to the digital realisation is that each agent keeps its own local date, the issuer judges the maturity of one of their bonds by their own date, and an escrow is released by the date of the escrow agent (Section 7).The implementation is demonstrated by a village market of six agents and an escrow agent that exercises symmetric mutual credit, a loan with interest, payments, a portfolio swap, escrow, redemption, sale of debt, and the redemption of a matured bond.Making the implementation secure against major faults—the loss of the private key or the machine—using identity and state custodians is the subject of prior work [8].

Paper structure. Section 2 presents grassroots bonds with pen and paper: the paper bond and its maturity, holdings and dates, the operations, and the loan and the credit line. Section 3 presents their grassroots social contract as the contract of grassroots currencies with the Date and Escrow clauses, the paper-and-pen realisation, and the transactions with dates the clauses compile to. Section 4 gives the instruments that are bond swaps, and Section 5 those that employ an escrow agent, the exchange of several lots at once among them. Section 6 extends the liquidity ratios to bonds and states the clearing results. Section 7 gives the digital realisation—what dates add to it, and the village market run on the program—Section 8 relates the work to others, and Section 9 concludes. The appendices carry the act schemas, the modality of each clause and the proof that the contract is grassroots (Appendix A); the deferred proofs (Appendix B); the further escrow instruments (Appendix C); the worked example of the ratios (Appendix D); the community bank with bonds (Appendix E); and the derivation of the program from the transactions (Appendix F).

2 Grassroots Bonds

Grassroots currencies were introduced [24] and implemented [16] as cryptocurrencies, and are presented in [24] as a physical currency, realised with pen and paper: a grassroots coin is an IOU—inside money [5]—issued freely by its issuer, whose offerings and holdings back it, and every transaction is a voluntary act of all its participants, with the rights and duties of the parties specified by the coins. Here we add a second instrument, the bond—a debt of its issuer due at a later date—and present it in the same way: first as a physical currency, then, from Section 3 on, as a grassroots social contract that extends the contract of grassroots currencies, whose clauses compile to volition-guarded multiagent atomic transactions [10, 17]. In the following, formal statements can be skipped in favour of their adjacent informal readings.

Persons. A person—natural or legal, with the gender-neutral pronoun "they"—is identified by their name and signature. We assume a potentially infinite set of persons Π; we consider only finite subsets of Π.

A grassroots coin is a unit of its issuer's debt, due now, backed by the issuer's offerings and by the coins and bonds they hold [24], and a grassroots bond is such a unit due at a later date. The two are distinct instruments: a coin carries no date, and a bond is minted with a date later than the date of its issuer.

Definition 2.1 (Grassroots Coins and Bonds). A *grassroots p-coin*, denoted $¢_p$, is a unit of debt issued by $p \in \Pi$, due now. A *grassroots p-bond with maturity date d*, denoted $¢_{p,d}$, is such a unit due at the date $d \in \mathbb{N}$. We write $¢_p^k$ and $¢_{p,d}^k$ for multisets of k of each.

A coin and a bond are units, and a holding is a whole number of them; in a practical implementation each p-coin and p-bond has a serial number and is signed by p.

Within the pen-and-paper scenario, we assume that signatures cannot be forged and that signed papers cannot be duplicated ("double-spent"). A pen-and-paper grassroots p-bond—a *paper bond*—is the paper coin of grassroots currencies [24] with a maturity date written on it: a promissory note, written on paper and signed by p, stating the issuer p, the maturity date d, and three standing obligations:

(1) p prices their offerings in their own coins and accepts them in payment at these prices;
(2) p prices every coin and bond they hold, of any issuer and maturity, 1-for-1 against their own coins, on standing offer; and
(3) once the date of p has reached the maturity date, the bearer may present the bond to p as they would a p-coin.

Obligations 1 and 2 are those of a paper coin [24], and obligation 3 is the bond's. A grassroots coin is a legally binding debt of its issuer, due now. Its bearer may present it to its issuer and take, at the written 1-for-1 price, any coin or bond the issuer holds, of the bearer's choice. A bearer may take a bond rather than a coin: a debt due later of a person the bearer trusts can be worth more than a debt due now of one the bearer trusts less. Taking an instrument the bearer issued is set-off: it returns the bearer's own debt, indisputable at par. A presentation left unfulfilled evidences the issuer's default, which in turn may cause a redemption run on p and if p turns out to be insolvent, the depreciation of the market value of p's coins and bonds. A coin or a bond is transferred by its holder signing it over to the new holder and delivering it, so any two persons transfer any coins and bonds between themselves, with no third party; the signatures on the paper record its provenance, and whoever holds it knows its issuer by the issuer's signature.

Holdings and dates. The state of a person p comprises c_p, the multiset of coins and bonds in p's possession (initially empty); $d_p^* \in \mathbb{N}$, p's local current date (initially the day of their calendar), advanced by the calendar. A bond is minted with a date later than the date of its issuer, so no bond is born mature. A bond $¢_{q,d}$ is *mature* iff $d \leq d_q^*$, and its bearer may then redeem it as a coin: whether q owes yet is a fact about q, so maturity is settled by q's date and the dates of the holders play no part. A coin carries no date, so no act of an issuer makes or unmakes one.

A transaction changes the states of its participants and no one else's. The four operations of grassroots currencies—mint, voluntary swap, pay, and redeem—gain advance-date, and are the acts of the grassroots social contract of Section 3, whose schemas compile to volition-guarded multiagent atomic transactions over the states above (Section 3.3):

(1) *Mint*: a person may create, at will, coins of their own, of any amount, or bonds of their own, of any amount and of any maturity later than their own date—with pen and paper, by writing and signing k papers;
(2) *Advance-date*: local time advances mechanically, by the calendar;

(3) *Voluntary swap*: two persons may exchange coins and bonds by mutual consent—the parties meet and exchange the agreed papers by simultaneous signing-over and delivery, so the swap is physically atomic;
(4) *Pay*: a holder may pay coins of the recipient to the recipient, at will—delivering the recipient's coins to the recipient;
(5) *Redeem*: the bearer of a coin, or of a matured bond, may present it to its issuer, at will, and take for it any coin or bond the issuer holds—the issuer takes the instrument back and gives one of the bearer's choice among those they hold.

Only coins are paid. A coin carries no date, so its issuer can neither refuse a payment of their own coins nor refuse a presentation of one, as obligations 1 and 2 require, and no date is read in either. Redemption generalises the coin-for-coin redemption of grassroots currencies [24] by allowing the bearer of a coin to choose any coin *or bond* held by the issuer of the coin. A bond is minted, swapped, escrowed and sold, and an issuer may fulfil a presentation with one; it is not paid, and once it is mature its bearer may redeem it as a coin. The issuer's date is the precondition of that redemption, and of no other act (Section 3); it is a machine precondition and not a guard, so the issuer cannot refuse, as obligation 3 requires. The holder of a bond may sell the debt to a third party, as shown in Section 4. A coin or a bond that reaches its issuer is self-held; whether the issuer keeps or cancels the paper is immaterial, as self-held instruments carry no value (Section 6).

Voluntary transactions and the obligations of the coins and bonds. Every transaction is a voluntary act of all its participants: minting and advancing one's date are acts of one person; a swap, a payment and a redemption take place at a meeting of the two, each performing their part. A voluntary swap is a contract, made at the meeting; either party may decline it. A payment and a redemption are the bearer's rights under the written obligations: the recipient of a payment must accept their own coins (obligation 1), and the issuer must fulfil a presentation of their coin (obligation 2) and of their matured bond (obligation 3); declining any of the three is a breach of the terms, actionable at law. The issuer assumed these obligations voluntarily, by minting the coin or the bond. The coins and bonds do not determine whether persons act on their rights, nor whether they advance their dates; the analysis of Section 6 makes both explicit assumptions: dates advance, and persons who can redeem eventually do.

Financial instruments as swaps of coins and bonds. Loans, credit lines, sale of debt, collateral, escrow, and further financial instruments are realised as voluntary swaps of grassroots coins and bonds, and as escrow agreements (Sections 4 and 5). Two are used repeatedly in this paper. A *loan* from lender p to borrower q is a swap of freshly-minted p-coins against q-bonds maturing later; a face value exceeding the grassroots coins lent expresses interest. A *credit line* from p to q is a standing commitment, held by an escrow agent, to advance p-coins against q-bonds up to an agreed limit, with drawn amounts carrying interest and undrawn capacity returned at expiry. A holder of a q-bond who loses trust in q may sell the debt in the free market, at a discount, as with coins [24].

Local dates. As there is no global clock, each person's calendar is their own, and persons may disagree on the date. A bond matures by the calendar of its issuer, so an issuer who does not advance their date does not yet owe on their bonds. The contract obliges each party to keep their date at the current date, an undertaking, as in an asynchronous system no record can attest the date a party keeps (Section 3). The value of a coin or a bond rests on trust in its issuer, and an issuer whose date lags loses it: their bonds never mature, their debts never fall due, and others price their currency accordingly or decline to hold it at all. In the ratios of Section 6 each person classifies their holdings against their own date, so a ratio is the view of the person computing it. For an option, exercise is judged by the escrow agent's date (Section 5), so the parties need agree only with the escrow agent. We assume no bound on the disagreement: it delays a transaction and never compels one.

3 The Grassroots Social Contract

A grassroots social contract [10] is a finite list of clauses its parties undertake towards one another. A clause that describes an act is written as an *act schema* [10]: the act's roles, which of them must will it, and the atoms it requires, forbids, adds and deletes at each role; the volition-guarded transactions [17] realising the contract are compiled from the schemas, three syntactic conditions on the schemas suffice for the protocol so realised to be volitionally grassroots, and each clause is *enforced*, *attested* or *undertaken* by the contract according to what its runs determine and record. Grassroots currencies are specified by a contract of five clauses [24]. Here we write the contract of grassroots bonds as that contract with two further clauses, Date and Escrow, its paper-and-pen realisation, and the transactions with dates its schemas compile to. Its act schemas, the modality of each clause and the proof that it is grassroots are in Appendix A, and the program every participant runs is given in Section 7.

3.1 The Grassroots Bonds Social Contract

A *party* is a person, natural or legal (with the gender-neutral pronoun 'they'), with *Alice*, *Bob* and *Carol* as variables naming distinct parties to the contract. A *grassroots coin* is a unit of debt that names its issuer and is due now, and a *grassroots bond* one that names a maturity date besides and is due then, with *Alice-coin* and *Alice-bond* referring to ones issued by Alice. Each party keeps a date, and a bond is *mature* when its maturity date is at most the date of its issuer. The contract extends GKK, the grassroots keys and knowledge contract [10], whose clauses it cites and does not restate: a person establishes their identity by choosing their own key pair, and makes themselves known by the public key (GKK:1); two persons may come to know each other by an act of both (GKK:2); and a person who knows two persons may introduce one to the other, by an act of the introducer and the one introduced to (GKK:3). A person *knows* another when they hold the other's public key; a coin or a bond carries its issuer's public key, so whoever holds one knows its issuer. Each clause opens with its legal modality [10]: ■ *enforced*, no act of the contract can breach it; ○ *undertaken*, its performance lies outside the contract, which records only the undertaking. The box ends with the contract's *pragmas*, which the app compiled from the contract follows [10]: a tab of the app for each kind of instrument it lists, and for each clause the person proposing an act, the button in brackets, pressed in the row of the person who takes the act's other role, a field in angle brackets for each variable of the act's schema, a maturity field left empty for a coin, and, indented, each other proposer with their button and each person asked with the question put to them; Date carries no line, its act being willed by no one.

GB: Grassroots Bonds

Alice, Bob and Carol are persons.

1. ■ *Minting.* Alice may decide to issue Alice-coins, and Alice-bonds of a maturity later than their own date.
2. ○ *Date.* Alice keeps their date at the current date.
3. ■ *Swap.* Alice and Bob, who know each other, may jointly decide to swap coins and bonds they hold.
4. ■ *Payment.* Alice may decide to pay Bob-coins they hold to Bob, whom they know.
5. ■ *Redemption.* Alice may decide to redeem a Bob-coin, or a matured Bob-bond, they hold against any coin or bond held by Bob, whom they know.
6. ○ *Pricing.* Alice prices their offerings in Alice-coins and accepts them in payment at those prices.

7. ■ *Escrow.* Alice may deposit coins and bonds with Carol for Bob, on a condition the three agree; Carol and Bob may jointly decide to release them to Bob when the condition is met, and Carol and Alice may jointly decide to return them to Alice when it fails; each act is between two parties who know each other.

Pragmas

Coins: the coins Alice holds.
Bonds: the bonds Alice holds.
Minting: `Alice [Mint coin] <amount> [Mint bond] <maturity> <amount>`
Swap: `Alice [Swap] <issuer given> <maturity given> <amount given> <issuer wanted> <maturity wanted> <amount wanted>`
`Bob: Alice asks to swap coins or bonds with you.`
Payment: `Alice [Pay] <amount>`
Redemption: `Alice [Redeem] <maturity given> <issuer wanted> <maturity wanted>`
Escrow: `Alice [Deposit] <issuer> <maturity> <amount>`
`Carol: Alice asks to deposit coins or bonds with you.`
`Carol [Release] <issuer> <maturity> <amount>`
`Bob: Carol asks to release coins or bonds to you.`
`Carol [Return] <issuer> <maturity> <amount>`
`Alice: Carol asks to return coins or bonds to you.`

Clauses 1, 3, 4, 5 and 6 are the contract of grassroots currencies [24] with a bond beside the coin where the act allows one, clauses 2 and 7 are new, and clause 5 redeems a matured bond as it does a coin. Clauses 1–5 and 7 describe acts; clause 6 describes none. Carol in clause 7 is the escrow agent. A swap in which each party gives coins of their own issue opens a mutual credit line; one in which a party gives coins and the other their own bonds of larger face value maturing later is a loan (Section 2).

Modalities. Clauses 1, 3, 4, 5 and 7 are enforced: each permits an act and obliges nothing beyond it, and the obligations of the payee in clause 4 and of the issuer in clause 5 are self-executing, as neither is asked. Clauses 2 and 6 are undertaken: keeping one's date at the current date and selling at one's prices are performed outside the contract, and a bond or a coin of a party in another's holding is the record of their undertaking. The contract therefore enforces the redemption of a bond only against the date its issuer keeps, and the obligation of a bond as a whole—to redeem it once its maturity date has come—is undertaken, where the obligation of a coin is enforced. The definitions [10], and the argument for each clause, are in Appendix A.2.

3.2 The Paper-and-Pen Realisation

The contract has two realisations. In the first, people read the clauses and follow them with paper and pen, and no machine takes part.

Definition 3.1 (Paper-and-Pen Realisation). The *paper-and-pen realisation* of the contract is its performance by its parties on signed notes: a party mints by writing and signing notes of their own, a coin note or a bond note stating its maturity date (clause 1); a party keeps a calendar (clause 2); two parties swap notes by signing them over and delivering them to each other at a meeting, both at once (clause 3); a party pays by signing over and delivering to another party coin notes that party issued (clause 4); a party presents a coin note, or a matured bond note, to its issuer, who takes it back and gives in return a note of the presenter's choice among those they hold (clause 5); a party prices their offerings in their own notes (clause 6); and a party deposits notes with the escrow

agent, who releases or returns them, each transfer by signing over and delivering at a meeting of the two (clause 7). A party's holding is the notes in their possession, their date is that of their calendar, and their consents are their own.

Under this realisation, safety and liveness are matters of law and of persons. An act the clauses do not permit is a breach, actionable at law, and so is declining a payment or a presentation, or letting one's date lag (Appendix A.2); and whoever persistently wills a permitted act with a willing counterparty eventually performs it, and the calendar advances—assumptions about persons. In the digital realisation (Section 7) both are properties of the program every participant runs, proven for Grassroots Flash [16]. The results of this paper are theorems about the transactions the contract compiles to (Section 3.3); the acts of the paper-and-pen realisation are those transactions performed on notes, so the results hold under both realisations.

3.3 The Transactions

The digital realisation begins by compiling the act schemas to volition-guarded multiagent atomic transactions [10, 17]. Each agent is a person operating a machine, and here the machine state of a party p is their holdings c_p, a multiset of coins and bonds, initially empty; their date d^*_p, initially the day of p's calendar; and the set $K_p \subseteq \Pi$ of the persons p knows, initially empty, which the transactions of GKK [10] add to, and to which a transaction below that gives p an instrument of issuer u adds u, a coin or a bond carrying its issuer's public key; K_p is the set of persons the *knows* atoms of p's state name (Appendix A), an atom added again or naming p changing it not at all.A transaction changes the machine states of its participants and no one else's, is enabled when its precondition holds, and is *guarded* by the persons among its participants whose consent it takes: it is taken only when every guard is willing. Transactions that are the same act in different configurations are *equivalent*. A run is *safe* if every step is a transaction of the protocol, *live* if a class of equivalent transactions that is enabled and willed by its guards is eventually taken, and *correct* if it is both; a protocol is *volitionally grassroots* if any group of persons may operate it on its own, and two groups become connected only by a transaction willed by a member of each [17]. Multiset sum and difference are written $\uplus$ and $\setminus$. The transactions of grassroots currencies [24] are the four acts over holdings of coins; the acts of grassroots bonds are the following transactions, in which mint takes two forms, a swap carries coins and bonds alike, redeem presents a coin or a matured bond and takes a coin or a bond, and advance-date and the three escrow transfers are new.

Definition 3.2 (The Transactions of Grassroots Bonds). For persons $p \neq q \in \Pi$, $e \in \Pi$ distinct from p and q, $u, v, r \in \Pi$, maturity dates $d, d' \in \mathbb{N}$, and amounts $k, k' \geq 1$, the *transactions of grassroots bonds* are the volition-guarded transactions $(c \rightarrow c', Q')$ below, over the participants named, in which c satisfies the precondition stated, a participant who receives an instrument of issuer u has $K'_p = K_p \cup \{u\}$, and every other component of a participant's state not named is unchanged. We write $¢^k_{u,*}$ for k instruments of u of one kind and maturity, coins or bonds, where the transaction does not distinguish them:

(1) $Mint_p(k)$ and $Mint_p(d, k)$, over $\{p\}$, guarded by $\{p\}$: the second with precondition $d > d^*_p$, so no bond is born mature; $c'_p = c_p \uplus ¢^k_p$ and $c'_p = c_p \uplus ¢^k_{p,d}$ respectively.

(2) $Advance_p$, over $\{p\}$, guarded by $\emptyset$: precondition that the day of p's calendar is past d^*_p; $d'^*_p = d^*_p + 1$, the next day; a calendar that has moved on by several days is as many advances.

(3) $Swap_{p,q}(¢^k_{u,*}, ¢^{k'}_{v,*})$, over $\{p, q\}$, guarded by $\{p, q\}$, with the two lots of different issuers, kinds or maturities: precondition $q \in K_p$, $p \in K_q$, $¢^k_{u,*} \subseteq c_p$ and $¢^{k'}_{v,*} \subseteq c_q$; $c'_p = c_p \setminus ¢^k_{u,*} \uplus ¢^{k'}_{v,*}$ and $c'_q = c_q \setminus ¢^{k'}_{v,*} \uplus ¢^k_{u,*}$.

(4) $Pay_{p,q}(k)$, over $\{p, q\}$, guarded by $\{p\}$: precondition $q \in K_p$ and $¢^k_q \subseteq c_p$; $c'_p = c_p \setminus ¢^k_q$ and $c'_q = c_q \uplus ¢^k_q$. Only coins are paid.
(5) $Redeem_{p,q}(x, r, *)$, over $\{p, q\}$, guarded by $\{p\}$, with x a q-coin or a q-bond with maturity $d \leq d^*_q$, and the instrument taken not x: precondition $q \in K_p$, $x \in c_p$ and $¢_{r,*} \in c_q$; $c'_p = c_p \setminus x \uplus ¢_{r,*}$ and $c'_q = c_q \setminus ¢_{r,*} \uplus x$. The bearer presents a coin or a matured bond, and takes a coin or a bond, at their own choice.
(6) $Deposit_{p,e}(¢^k_{u,*})$, over $\{p, e\}$, guarded by $\{p, e\}$: precondition $e \in K_p$, $p \in K_e$ and $¢^k_{u,*} \subseteq c_p$; $c'_p = c_p \setminus ¢^k_{u,*}$ and $c'_e = c_e \uplus ¢^k_{u,*}$.
(7) $Release_{e,q}(¢^k_{u,*})$ and $Return_{e,p}(¢^k_{u,*})$, over $\{e, q\}$ and $\{e, p\}$, guarded by both participants: precondition that the two participants know each other and $¢^k_{u,*} \subseteq c_e$; the instruments are deleted at e and added at the other participant.

Equivalence is the least equivalence on the underlying machine transactions under which two transactions with the same name and arguments, differing only in the configurations c and c', are equivalent.

Each is a schema of Appendix A at a binding: the participants are the roles, the guards are the guarding roles, the precondition is what the roles require, and the effect is what they add and delete.The equivalence is the one generated by one schema at one binding [10], except that the dates a role holds, t and t' in the schemas, are not arguments of the transaction, so a mint or a redemption of a matured bond remains one class while dates advance, and the advances of a person are one class. One machine transaction can be several of these transactions, which are then equivalent, and their class is guarded by each of their guard sets: a redemption is also the swap of the same two lots, so its class is guarded by the bearer; p's q-coin taken for q's p-coin is a redemption by either of them, so its class is guarded by either; and a deposit, a release and a return of one lot from one person to another are one class, guarded by both, and by the giver if the lot is coins of the receiver, that being a payment.The *grassroots bonds* protocol *GB* is the protocol over these transactions and those of GKK, which add to the sets K_p and touch no instrument [17]: the protocol realising the contract [10].Mint, pay and redeem are guarded by the initiator—an obligation of the counterparty under the coins and bonds, so their volition is not required; a swap and each escrow transfer are contracts, guarded by both; advance-date is unguarded, as local time advances mechanically.In a correct run, a class of transactions that is enabled and willed by its guards is eventually taken, so every date advances beyond any bound, and a redemption that is enabled and willed by its bearer is eventually taken; whether the bearer wills it is an assumption about persons, under both realisations, and the two assumptions of Section 6 are these.

Lemma 3.3 (Knowledge of the Issuer). *In any safe run, a person who holds a coin or a bond of another issuer knows that issuer.*

Proof. Initially every holding is empty. Mint and pay give a person instruments of their own issue; swap, redeem, deposit, release and return give a participant instruments of an issuer u and add u to that participant's K; and no transaction removes a person from a K. □

Proposition 3.4 (Conservation of Money and Chain Redemption). *Conservation of Money and Chain Redemption of grassroots currencies [24] hold for the transactions of grassroots bonds: in any safe run, the p-coins and p-bonds in any configuration are exactly those p has minted so far, and the debts of p are those of them held by others; and if each person of a sequence of distinct persons holds a coin of the next, the first obtains a coin of the last by redeeming along the sequence.*

Proof. Mint adds new p-coins and p-bonds at p; swap, pay, redeem, deposit, release and return move coins and bonds between two agents without creating or destroying them; advance-date

changes no holding, and neither does a change of volition. Hence the p-coins and p-bonds across all agents are those p minted, and a self-held instrument is not a debt (Section 2). For the sequence, p_0 knows p_1 (Lemma 3.3) and redeems the p_1-coin they hold against the p_2-coin held by p_1, so the sequence without p_1 satisfies the hypothesis with one person fewer, and induction on its length completes the proof. □

Grassroots. Bonds among the members of a group require nothing outside the group, so any group may operate grassroots bonds on its own, and two groups may transact once their members meet: the grassroots bonds protocol is grassroots, and volitionally grassroots.

Corollary 3.5 (Volitionally Grassroots). *Grassroots bonds are volitionally grassroots.*

Proof. The contract is syntactically grassroots (Appendix A); the protocol realising a syntactically grassroots contract is volitionally grassroots [10]. □

4 Financial Instruments via the Voluntary Swap of Bonds

Financial instruments are expressed via the voluntary swap of grassroots bonds and via escrow agreements. The instruments below are single swaps; each is specified as a transaction of the model, and further instruments are combinations of them. A swap exchanges one lot—coins of one issuer, or bonds of one issuer and maturity—against another (Section 3); the instruments that exchange several lots at once, and those needing a third party—to hold bonds, or to judge an outside event—are developed in Section 5, where the escrow agent appears as an agent of the model. Each instrument below is a swap between agents p and q, guarded by the two agents, specified by the instruments x transferred from p to q and the instruments y transferred from q to p. Symmetric mutual credit, asymmetric mutual credit, and the sale of debt are swaps of coins [24]; what a maturity date adds is the following.

An *oracle* is an agent the parties agree to trust for the value of a quantity outside the protocol—a reference interest rate, a delivery, the default of a third person. An oracle holds no coins or bonds and takes no transaction: it attests, and the agents who trust it act on the attestation. The oracle is the information source of the escrow constructions of Section 5, and the trust the parties place in it is part of the agreement.

(1) **Zero-coupon loan.** Lender p mints $k' < k$ coins, borrower q mints a bond maturing at d: $x = ¢_p^{k'}$, $y = ¢_{q,d}^{k}$. The premium of asymmetric mutual credit [24] is interest here: a bond is redeemed only once it is mature, so the loan cannot be recalled before its date, whereas the borrower may repay it early, by redeeming the lender's coins against their own bond.

(2) **Sale of debt before maturity.** p sells r-bonds maturing at d to q for $k' < k$ q-coins: $x = ¢_{r,d}^{k}$, $y = ¢_q^{k'}$. The debtor r is not a party; the discount prices the time to maturity as well as the doubt about r's solvency.

(3) **Forward contract.** p and q exchange bonds both maturing at d: $x = ¢_{p,d}^{k}$, $y = ¢_{q,d}^{k'}$. The bonds are swapped at inception and mature together, so at d each party holds a claim on the other of the agreed face value; the amounts due are those of a forward settled at d. The risk differs: between inception and d each party holds the other's debt, so a default in that interval falls on the counterparty, whereas a forward settled at d leaves nothing outstanding until then.

The instruments cover loans (item 1), debt trading (item 2), and derivatives (item 3). Currency swaps and repurchase agreements (repos) are realised as sequences of two swaps; factoring (selling receivables at a discount) is an instance of sale of debt; a mortgage is a loan combined with collateral. Loans with payment schedules—balloon and fixed-payment loans—and the interest rate swap exchange several lots at once, and are developed in Section 5 as packaged exchanges, together

with credit lines; collateral, guarantees, options, insurance, credit default swaps, and letters of credit are in Appendix C.

5 Escrow-Based Financial Instruments

An escrow agent e holds coins and bonds on behalf of the parties to an escrow agreement and transfers them according to its agreed-upon conditions; it issues no coins or bonds of its own, so no redemption reaches what it holds. Each escrow transaction is a transfer guarded by the two agents it involves—the agent giving the bonds and the agent receiving them; as e is party to every transfer, each escrow transaction is guarded by e and one of the parties. Thus e is a volitional agent, whose volitions are determined by the escrow program implementing the agreement: for a date-based condition, by the local date d_e^* of e; for an event-based condition, by the judgment of e, informed where needed by an oracle agreed by the parties (Section 4). The escrow constructions assume that both parties trust e to follow the escrow program and, for event-based conditions, trust e as adjudicator of the condition. We write $\{(a, x),\ (b, y)\}$ for the exchange in which a gives the instruments x and b gives the instruments y, guarded by a and b; $y = \emptyset$ makes it a one-sided transfer. The contract has four: the deposit, the release and the return of clause 7, and the payment of clause 4, which carries only coins of the party receiving them. Coins and bonds therefore do not pass one way from one ordinary party to another; an exchange has two sides.An exchange or transfer of one lot—coins of one issuer, or bonds of one issuer and maturity—is one act of the contract (Section 3); one of several lots is the corresponding sequence of acts, executed all-or-nothing by the packaged exchange below. An escrow whose agent is a committee, acting when a threshold of its members agree, is not expressible in the present model, in which every agent is one person; committee agents are left to further work.

Mechanism. An escrow arrangement has a depositor, a beneficiary, and an escrow agent e. The depositor transfers coins or bonds to e, guarded by the depositor and e. Subsequently e either releases them to the beneficiary or returns them to the depositor, each guarded by e and the recipient, as the escrow program determines from the agreed conditions. In the transfers below, the two named agents are the participants in, and guards of, the transaction.

(1) **Packaged exchange:** Parties p and q agree the lots X that p gives, the lots Y that q gives, and an establishment window T_0:
   (a) **Deposit.** $\{(p, X),\ (e, \emptyset)\}$ and $\{(q, Y),\ (e, \emptyset)\}$: each party deposits its lots with e, one transfer per lot.
   (b) **Release.** Once every lot of both is deposited, $\{(e, Y),\ (p, \emptyset)\}$ and $\{(e, X),\ (q, \emptyset)\}$: e releases the deposits crosswise.
   (c) **Return.** If the window passes with the lots of a party incomplete, e returns every deposited lot to its depositor.

   Neither party holds the other's lots before both deposits are complete, and a party whose counterparty's lots do not all arrive by T_0 has theirs returned; once both deposits are complete, each party takes the other's lots at their own will and neither can take theirs back. The packaged exchange realises the instruments whose sides are several lots:
   (a) **Balloon loan.** Lender p gives $X = ¢_p^k$; borrower q gives $Y = \bigcup_{j=1}^{n} ¢_{q,d_j}^{k_j} \cup ¢_{q,d}^{k}$—coupon bonds $¢_{q,d_j}^{k_j}$ and a principal bond $¢_{q,d}^{k}$.
   (b) **Fixed-payment loan.** Lender p gives $X = ¢_p^k$; borrower q gives $Y = \bigcup_{j=1}^{n} ¢_{q,d_j}^{k_j}$, bonds with fixed payments k_j at dates d_j.

(c) **Interest rate swap.** $X = \bigcup_j \¢^{k_j}_{p,d_j}$, $Y = \bigcup_j \¢^{k'_j}_{q,d_j}$, where p pays fixed amounts k_j while q pays amounts k'_j that vary according to a reference rate, attested by the interest-rate oracle. Periodic settlement exchanges realise the net difference at each date.

(2) **Further escrow-based instruments:** Collateral, guarantees, options, insurance, credit default swaps, and letters of credit are escrow constructions of the same three transfers—deposit, release, return; they are given in Appendix C.

(3) **Credit line:** A lender p commits to advance up to k coins to borrower q, with periodic interest on drawn credit and line expiry at date T. The agreement specifies the limit k, interest rate ρ (assumed to yield integer interest amounts), payment dates $d_1, \ldots, d_n$, and expiry T.

(a) **Establishment.** p mints k coins and deposits them with escrow agent e: $\{(p, \¢^{k}_{p}),\ (e, \emptyset)\}$. e tracks the drawn amount k_d, initially 0.

(b) **Draw.** When q draws $k' \leq k - k_d$, q mints a principal bond $\¢^{k'}_{q,T}$ (maturing at line expiry) and coupon bonds $\¢^{\rho k'}_{q,d_j}$ for each d_j later than the dates of e and of q, as mint requires of q, and deposits them with e, one transfer per lot: $\{(q, \¢^{k'}_{q,T} \cup \bigcup_{d_j > d^*_e} \¢^{\rho k'}_{q,d_j}),\ (e, \emptyset)\}$. Once every lot is deposited, e releases them to p, one transfer per lot, $\{(e, \¢^{k'}_{q,T} \cup \bigcup_{d_j > d^*_e} \¢^{\rho k'}_{q,d_j}),\ (p, \emptyset)\}$, and releases the coins to q, $\{(e, \¢^{k'}_{p}),\ (q, \emptyset)\}$—the release of a packaged exchange (item 1) whose lender's side e already holds—and updates the drawn amount k_d to $k_d + k'$.

(c) **Repayment.** q acquires p-coins (e.g. by trade with third parties or from their own holdings) and redeems them from p against their principal bond, coin by coin (Section 3, Redemption): repayment is q's right under the written obligations, and p cannot refuse it. p then deposits the p-coins with e via $\{(p, \¢^{k_{\text{rep}}}_{p}),\ (e, \emptyset)\}$, restoring capacity: e updates k_d to $k_d - k_{\text{rep}}$. Repayment may be partial: any $k_{\text{rep}} \leq k_d$ restores capacity by k_{rep} and leaves $k_d - k_{\text{rep}}$ drawn. The coupon bonds are q's debt from the draw and remain due at their dates; returning the unaccrued ones is a further transfer from p to q.

(d) **Expiry.** Once $d^*_e \geq T$, e returns the undrawn balance to p: $\{(e, \¢^{k-k_d}_{p}),\ (p, \emptyset)\}$.

The credit line is the one construction in which the escrow agent carries state across transactions: the drawn amount k_d. It is not extra state: it is the limit less the p-coins e holds.

Proposition 5.1 (Credit Line). *Let k be the limit, T the expiry, and k_d the drawn amount of the credit line above. In any safe run in which e follows the escrow program and p and q transact only as the agreement specifies, between the operations of the agreement—each packaged exchange or transfer complete—e holds $\¢^{k-k_d}_{p}$, p holds $\¢^{k_d}_{q,T}$ together with the coupon bonds of the draws, and $0 \leq k_d \leq k$. At expiry with $k_d = 0$, p holds $\¢^{k}_{p}$, and the q-bonds p holds are the coupon bonds of the draws.*

The coins e holds and the principal p holds sum to the limit, so q cannot draw beyond k and the principal p holds equals the amount drawn. The proof is in Appendix B.

6 Liquidity Ratios, Solvency, and Clearing

This section extends the liquidity ratio of grassroots currencies [24] to bonds and proves the clearing results for bonds. The ratios are per person: the cash, quick, and current liquidity ratios of corporate finance apply to grassroots bonds, with bond maturity playing the role of asset class. The results are per community, stated against its community bank or any designated member: whether its debts, of all maturities, can be extinguished by redemptions, and whether they are, once dates advance and persons act on their rights.The persons act independently, each by their own rights, so we state and prove the answers in graph terms: the holdings form a graph, and solvency depends on the graph

degrees. What a group can obtain of a member's coins now, and whether a redemption run is met, are questions about coins, answered in [24]; the section ends with how the answers apply to bonds.

Liquidity ratios. For grassroots coins the cash, quick, and current liquidity ratios of corporate finance coincide in one liquidity ratio, as every coin is redeemable now [24]. Here we recover the three ratios exactly, with maturity (a coin, a near-term bond, a long-term bond) playing the role of asset class. Recall that d_p^* denotes the local current date of person p (Section 2). Let $\nu_p(q)$ be the number of p-coins held by q at present, and $\nu_{p,t}(q)$ the number of p-coins and p-bonds maturing at or before t held by q at present, both zero if $p = q$, as instruments p holds of their own carry no value. Let δ denote a near-term horizon (typically 90 days) and Δ the operating cycle (typically one year), with $0 < \delta < \Delta$. The current liabilities of p are their coins and their bonds maturing within the operating cycle, $\sum_q \nu_{p,\, d_p^*+\Delta}(q)$—the common denominator of the three ratios of p:

$$\textit{Cash Ratio: } \frac{\sum_r \nu_r(p)}{\sum_q \nu_{p,\, d_p^*+\Delta}(q)} \qquad \textit{Quick Ratio: } \frac{\sum_r \nu_{r,\, d_p^*+\delta}(p)}{\sum_q \nu_{p,\, d_p^*+\Delta}(q)}$$

$$\textit{Current Ratio: } \frac{\sum_r \nu_{r,\, d_p^*+\Delta}(p)}{\sum_q \nu_{p,\, d_p^*+\Delta}(q)}$$

With the common denominator and progressively wider numerators, the ratios nest: cash ratio $\leq$ quick ratio $\leq$ current ratio. A worked example on the village market is in Appendix D; the creditworthiness measures of grassroots currencies—foreign debt, current account, and velocity—apply unchanged, counted over coins and bonds of all maturities [24].

Insolvency and illiquidity. The ratios separate two failures. A person is *insolvent* when their liabilities exceed their assets: they cannot repay their debt, and no loan remedies that—the current ratio, all assets against all liabilities of the operating cycle, bears on it. A solvent person is *illiquid* when the coins they hold cannot meet the claims arriving now, though their assets suffice at their maturities—the cash ratio measures exactly this, its numerator the person's immediate means of meeting redemptions. For coins the two failures are one, as every asset is a coin; the bond separates them. Default is sanctioned by the community [1]: holders redeem what the defaulter still holds, no one extends them new credit, they lose set-off and the ability to trade in their own currency, and the market value of their coins and bonds depreciates.

Computations. A *configuration* records, for every person, their holdings, the persons they know and their local current date; a *transition* leads from one configuration to the next—a transaction of the contract (Section 3.3) taking place. A *computation* is a sequence of configurations, each reached from the previous by a transition; it is *safe* if every step is one, and a transaction is *enabled* in a configuration if its preconditions hold there.The possibility result below (Proposition 6.3) quantifies over safe computations—what can happen; the liveness result (Proposition 6.5) quantifies over computations in which dates advance and persons act on their rights—persons who can redeem eventually do (Definition 6.4)—and states what must then happen. In the volitional framework, liveness takes an enabled class that is willed by its guards, and advance-date is willed by no one, so dates advance; that a bearer wills an enabled redemption is an assumption about persons (Section 3.3).The proofs of the propositions below are in Appendix B.

Graph solvency. Can a community extinguish all its internal debts? A debt is extinguished when the coin or the bond returns to its issuer, and a redemption does exactly that: the bearer returns a coin to its issuer and takes in exchange any coin or bond the issuer holds—as of right (Section 2), without transferring value, and without changing any person's assets net of liabilities.The propositions quantify over safe computations and do not assume a holder can inspect the debtor's holdings; in practice a debtor incapable of set-off discloses their holdings to the redeemer.The results are

stated relative to a designated person $s \in \Pi$, the *sink*: the member with whom the community's net position is left—e.g., a community bank in a village (Appendix E).Here we define the debts of a community as a graph, one edge per coin and per bond from holder to issuer, and prove: the community can extinguish all its debts, by redemptions, exactly when every member holds at least as many coins and bonds as they owe (Proposition 6.3); and when it can, members who act on their rights do, in any order, without coordination, once dates advance (Proposition 6.5).Clearing uses Redeem and Advance-date (Section 3): a Pay transfers value and a swap requires the counterparty's consent, while Advance-date brings a debtor's date up to the maturity of their bonds, after which their bearers redeem them. For coins, clearing uses Redeem [24]; a bond is redeemed once it is mature, which needs the debtor's date to reach its maturity.

Definition 6.1 (Holding Graph, Sink-Closed). Given a configuration and a designated person $s \in \Pi$, the *sink*, the *holding graph* is the directed multigraph with the persons as vertices and an edge $u \to v$ for each v-coin and each v-bond that person $u \neq v$ holds. A set of vertices containing s is *sink-closed* in a directed multigraph over the persons if every edge between two vertices other than s has both endpoints in the set or neither.

We write a_p and ℓ_p for the out-degree and in-degree of person p in the holding graph: their assets and liabilities in circulation, counted in coins and bonds of all maturities. A sink-closed set of persons is a community whose only claims across its boundary are with the sink; the sink is a member of every such community. A Redeem transaction in which the debtor is not s and the instrument taken is not issued by the debtor is a *clearing redemption*; it presents a coin or a matured bond.

Definition 6.2 (Clearing). Given a configuration and a sink-closed set V of the holding graph, the holdings of V *clear* if some finite safe computation of clearing redemptions among members of V and Advance-date transitions of members of V (Section 3) leads to a configuration in which every coin and bond of an issuer in V other than s is held by its issuer.

Proposition 6.3 (Graph Solvency). *The holdings of a sink-closed set V clear if and only if $a_p \geq \ell_p$ for every member p of V other than s, and cleared holdings leave each such member holding exactly $a_p - \ell_p$ coins and bonds of the sink.*

The liveness counterpart replaces "some computation clears" by "every computation in which dates advance and persons act on their rights clears". Under both realisations, that persons act on their rights is an assumption about persons; that the calendar advances is one with pen and paper, and the digital realisation proves it of the protocol, as advance-date is willed by no one (Section 7).

Definition 6.4 (Eventual Redemption). An infinite safe computation of clearing redemptions among members of a sink-closed set V and Advance-date transitions of members of V satisfies *eventual redemption* if (*i*) the local current date of every member of V advances beyond any bound, and (*ii*) no clearing redemption among members of V is enabled in every configuration of some suffix in which none occurs.

Proposition 6.5 (Clearing Liveness). *If $a_p \geq \ell_p$ for every member p of V other than s, then every infinite safe computation of clearing redemptions among members of V and Advance-date transitions of members of V that satisfies eventual redemption reaches cleared holdings of V, after at most $\sum_{p \in V \setminus \{s\}} \ell_p$ redemptions.*

Proposition 6.3 has three special cases: (*i*) V is all persons: the economy clears exactly when every person's assets in coins and bonds cover their liabilities in them; when every bond matures within the operating cycle, the condition reads: every person has a current ratio of at least 1. (*ii*) V

has no claims on or by the sink: every coin and bond then counts once as an asset and once as a liability within V, so the condition holds only with equality throughout—the community clears exactly when every member holds precisely as much as they owe, and each member ends holding only their own coins and bonds. (*iii*) V is a cycle of mutual debt (with the sink an isolated member): the cycle clears with no coin or bond of the sink, as chain redemption around the cycle ends in a set-off. A sink-closed set is the sink together with any union of connected components of the holding graph without it, so one sink serves many communities, each clearing on its own. The propositions concern one configuration; over time, liabilities grow with interest, which enters through minting and swaps, not through clearing.

Coins among the bonds. Two further questions of grassroots currencies [24] concern what can be redeemed now, and their answers apply to the coins and the matured bonds. The *liquid holding graph* of a configuration is the subgraph of the holding graph consisting of the edges of coins and of bonds mature by their issuer's date. What a group A of persons can obtain of the sink's coins by its own redemptions, now—with no date advancing, as advancing the dates of others is outside the group's control—is the maximum flow from A to s in the liquid holding graph: a coin and a matured bond are presented alike, either may be taken on the way to a coin of the sink, and a bond not yet mature cannot be presented, so the maximum-flow proposition of [24] applies to this graph as it does to the holding graph of coins.For a single person, the numerator of their cash ratio counts the coins they hold, and the maximum flow counts the matured bonds they hold besides.A redemption run—all coins and matured bonds of a person presented within a single day—is met in full exactly when the coins and bonds the person holds of other issuers are at least as many as the instruments presented, as each presentation takes one of them, of the bearer's choice: the redemption-run proposition of [24] applies with the instruments held as the assets and the instruments presented as the claims. When it is not met, the presentations that cannot be met stand unfulfilled, evidencing the person's default (Section 2).

7 Digital Realisation

The digital realisation of grassroots currencies—the issuer as the sole point of trust, the digital social contract every participant runs, finality by the issuer's signature and recovery by custodians, and the program derived from the contract, transaction by transaction—is given in [24]. Here we give what dates and escrow add to it, and run the bonds village market on the program.

Volitions and guards. In the pen-and-paper scenario, a person consents by taking part in the transaction; digitally, standing consents are recorded. The assumptions of Section 6—that dates advance and that persons act on their rights—become the correctness requirement of the framework: in a correct run, a transaction that is enabled and willed by its guards is eventually taken, and advance-date is willed by no one [17]. The protocol so specified is *volitionally grassroots*: two groups become coupled only by a first transaction willed by a member of each (Corollary 3.5).

The digital social contract. The contract of Section 3 is realised by the *digital social contract* implementing the currency: the multiagent program that every participant runs, its instances communicating only with each other [4]. Its enforced clauses are self-executing, as for coins [24], where the issuer signs every transaction in their own coins and bonds, and here applies their own date to them: a bond presented to its issuer is mature by the issuer's date or it is not, and no party needs to see another's date. The escrow transfers (clause 7) are guarded by the escrow agent and the party receiving, and the escrow program of the agent wills release or return as its own date, or the attestation of an oracle, determines (Section 5). The pricing undertaking (clause 6) is performed outside the contract.Finality by the issuer's signature [16], recovery from the loss of a

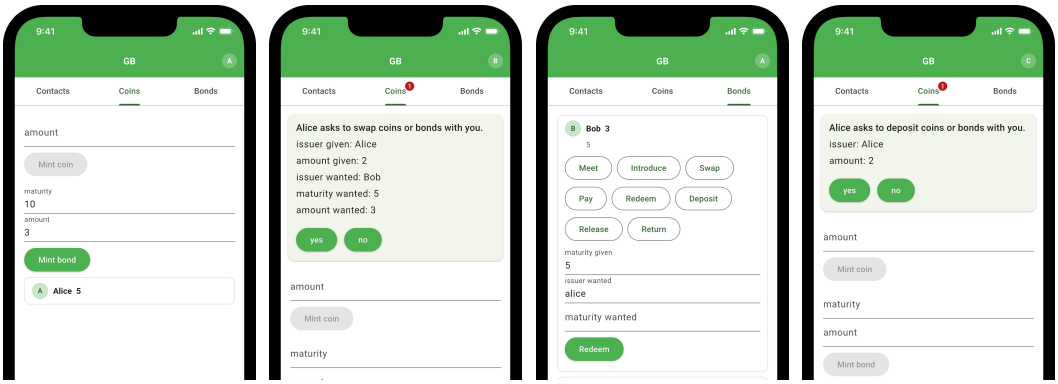


Fig. 1. The GB app. Alice mints three Alice-bonds maturing on day 10 (GB:1); Bob is asked to swap three Bob-bonds maturing on day 5 for two Alice-coins, a loan from Alice to Bob (GB:3); Alice redeems a Bob-bond, matured by Bob's date, against an Alice-coin Bob holds (GB:5); Carol is asked to accept Alice's deposit of two Alice-coins in escrow (GB:7). This is a screenshot of the GB mini-app, compiled from the GB social contract down the abstraction cascade [10, 12]. One instance of the app on an iPhone simulator runs the mini-app of every agent; the agents communicate over channels within it, and the screenshot shows one person's screen. All source code and all artefacts, including the social contracts and the artefacts the abstraction cascade produces from them, are in the public repository https://github.com/EShapiro2/GLP.

key or a machine by identity and state custodians [8], enforcement by mutual attestation, default and identity escape, the digital platforms a currency can run on, and privacy are as for coins [24].

Implementation in GLP. Grassroots Logic Programs (GLP) [25] is a multiagent, concurrent, logic programming language running on Dart for smartphone deployment. A grassroots specification is realised as a GLP program, which people load and activate voluntarily on their devices. Here we implement the contract of Section 3 as a program in vGLP [12], GLP with volitional procedures, which put a question to the person: the agent's question is the stream of its person's requests, and it has one clause for each transaction of Section 3.3 at the party who initiates it. The vGLP compiler compiles it to the GLP program every participant runs, together with the interface constructs it derives from the program; the derivation, transaction by transaction, is in Appendix F. What dates add to the digital realisation is three things: each agent keeps its own local date, advanced by Advance-date from its calendar, with no global clock; the issuer tests a bond presented to them by their own date, a presentation they must fulfil; and an escrow agent releases or returns what is deposited with it as the escrow program wills, for a date-based condition by the escrow agent's own date. The program of grassroots bonds is a mini-app the Grassroots Super-App [7] installs, as the program of grassroots currencies is [24]; the run below is of the compiled program with scripted persons in place of the interface. Figure 1 shows four of the app's screens.

7.1 Village Market Demonstration

To demonstrate that the instruments of Sections 4–5 compose into a functioning economy, we run a village market of six agents and an escrow agent on the program: Alice (baker), Bob (farmer), Charlie (carpenter), Diana (doctor), Eve (teacher), and Frank (fisherman). The scenario unfolds over a month and exercises seven distinct financial operations:

(1) **Symmetric mutual credit.** Alice and Bob each mint 15 coins and swap them 1-for-1, establishing mutual liquidity. Alice and Charlie do the same with 15 coins each, Charlie and Eve with 10, and Eve and Frank with 10.

Party	Holdings after the scenario
Alice	5 bob-coins, 8 alice-coins, 15 charlie-coins, 4 frank-coins
Bob	10 alice-coins, 20 diana-coins
Charlie	10 alice-coins, 10 eve-coins, 6 charlie-coins
Diana	24 bob-bonds (day 25), 12 frank-bonds (day 28), 8 diana-coins, 1 frank-coin
Eve	4 charlie-coins, 1 frank-coin, 2 alice-coins, 10 bob-coins
Frank	7 diana-coins, 10 eve-coins, 6 frank-bonds (day 28), 4 frank-coins
Escrow agent	nothing

Table 1. The holdings of the seven parties at the end of the village market, as each party's screen reports them, self-held coins and bonds included; at the end Bob's date is 25 and Frank's 28, so every bond listed is mature.

(2) **Asymmetric credit (loan).** Diana lends Bob 20 diana-coins in exchange for 24 bob-bonds maturing on day 25—a 20% interest rate. Diana similarly lends Frank 15 coins for 18 bonds maturing day 28.
(3) **Pay.** Bob pays Alice 5 alice-coins for bread; Eve pays Charlie 6 charlie-coins for a bookshelf; Frank pays Diana 3 diana-coins for a medical checkup; Eve pays Alice 3 alice-coins for bread.
(4) **Portfolio swap.** Eve trades 5 frank-coins to Charlie for 5 alice-coins, acquiring the currency she needs to buy bread from Alice.
(5) **Escrow.** Charlie hires Frank to build a dock and deposits 5 frank-coins with the escrow agent for Frank, to be released when the escrow agent's date reaches day 30; the escrow agent's date reaches day 30, and it releases the coins to Frank as payment for the dock.
(6) **Redeem.** Frank redeems 5 diana-coins from Diana, taking back five of his own bonds she holds—set-off, reducing his exposure to Diana's currency.
(7) **Sale of debt.** Alice sells 10 bob-coins to Eve for 4 frank-coins—a discounted sale reflecting Alice's desire to reduce exposure to Bob's debt.

Each party runs the compiled program of Appendix F with its dispatcher, and the seven run concurrently in one process, a scripted person submitting each party's requests and answering its cards in place of the interface, each act once the one before it has taken effect, and the escrow program answering the escrow agent's; each party's date advances by its own calendar, as the operations need it. Shown once, on Frank's bonds, is what dates add: before Frank's date reaches day 28, Diana presents one of the frank-bonds she holds, and Frank's agent returns it, as it is not mature by Frank's date; once Frank's date has reached day 28, Diana presents a frank-bond again and redeems it against a frank-coin Frank holds. The run performs the seven operations and the redemption of the matured bond, and ends with the holdings of Table 1. They reconcile with Conservation of Money (Proposition 3.4): the coins and bonds of each issuer held across the seven parties are those they minted—30 alice-coins; 15 bob-coins and 24 bob-bonds; 25 charlie-coins; 35 diana-coins; 20 eve-coins; and 10 frank-coins and 18 frank-bonds—and are those the seven operations and the redemption produce.

7.2 Implementation Summary

The village market of Section 7.1 is a GLP script that drives the seven parties through the seven operations. The implementation is open-source: the bond agent, its compilation and the escrow programs, as a mini-app, and the plays are at https://github.com/EShapiro2/GLP, in `programs/currencies/bonds`.

The proof-of-concept represents each coin and each bond as a unit. Each agent keeps its own local date d_p^*, advanced via Advance-date, with no global clock. Redeeming a bond requires it to be mature by the calendar of its issuer (Section 3); the issuer applies their own date, so no party needs to see another's. The scenario runs with every party acting as the agreements specify. Four things are left to further work: an issuer who defaults by refusing a presentation they are obliged to fulfil, and the effect on the liquidity ratios of the holders of their bonds; counterparties who deviate from an escrow program; concurrent redemptions contending for the same coins and bonds, where the menu of Appendix F is contested; and populations of hundreds of agents, at the scale of the village of Appendix E.

8 Related Work

The relation of grassroots currencies to global cryptocurrencies and the attacks each design is open to, to credit networks, IOU networks and mutual credit systems, to personal currencies, to microfinance and informal community finance, to payment and state channels, and to monetary theory is given in [24]. Here we relate grassroots bonds and the instruments built on them to decentralised finance, to financial contract specification, to tokenised and traditional bonds, and to bills of exchange.

Financial instruments on blockchains. The programmability of blockchains via smart contracts [2, 23, 26] has given rise to Decentralised Finance (DeFi) [22, 27], which re-implements traditional financial instruments on-chain. Lending protocols [11, 15] offer overcollateralised loans with algorithmically-determined interest rates; decentralised exchanges provide token trading via automated market makers; and stablecoins offer price-stable media of exchange.

These DeFi instruments have analogues among the grassroots bond instruments of Sections 4–5: loans, credit lines, sale of debt, and options are all expressible as voluntary swaps of grassroots bonds and escrow agreements, without requiring a blockchain or consensus. Two differences are noteworthy. First, DeFi instruments require overcollateralisation enforced by smart contracts because counterparty identity is pseudonymous and trust is absent; grassroots instruments operate among persons with mutual trust, and collateral (Appendix C) is available but not mandatory. Second, DeFi instruments incur gas fees for every state transition; grassroots instruments incur no fees beyond bilateral communication. Flash loans [19], which exist within a single atomic blockchain transaction, have no counterpart: grassroots bonds have no global atomic transaction spanning many parties, and a loan is a swap of two, settled by redemption at maturity.

Financial contract specification. The foundational work on composable financial contract specification predates blockchain: Peyton Jones et al. [18] introduced a combinator library for describing financial contracts declaratively, inspiring subsequent blockchain domain-specific languages. Marlowe [14], Cardano's financial contract DSL, specifies instruments declaratively with exhaustive static analysis; Scilla [23] introduced automata-based contracts with formal safety guarantees. These languages demonstrate that financial instruments can be specified declaratively, yet they compile to consensus-bound smart contracts. Grassroots bonds take the same declarative approach—each instrument is composed of the acts of a grassroots social contract (Section 3), a swap of one lot, or a packaged exchange of several held by an escrow agent—but enforce it bilaterally rather than via consensus, with each issuer guaranteeing the security of transactions in their currency [16]. Discreet Log Contracts [6] are bilateral derivatives on Bitcoin whose payout is fixed by an oracle signature, with the terms of the contract invisible on chain; the escrow instruments of Section 5 share the bilateral form and the oracle, and settle with no chain at all—the payout is a transfer of bonds by the escrow agent, enforced by the written obligations of the bonds rather than by a spending transaction.

Tokenised bonds on blockchain. On-chain bond issuance represents the blockchain-native approach to the same problem grassroots bonds address. A Federal Reserve analysis [3] examines tokenisation of real-world assets including the Santander (2019) and EIB (2021) bond issuances on Ethereum, identifying smart contract designs and financial stability implications. Security token offerings (STOs) extend this trend to debt instruments more broadly [13]. These tokenised bonds and STOs differ from grassroots bonds in three respects: they require a public blockchain and its associated gas costs; they rely on custodians to bridge on-chain tokens with off-chain legal obligations; and they are issued by institutions rather than by any person.

Traditional bond instruments. The loan structures of Sections 4 and 5—zero-coupon, balloon, and fixed-payment loans—correspond to standard fixed-income instruments as classified by Fabozzi [9]. The escrow-based instruments of Section 5 and Appendix C—collateral, guarantees, and options—likewise have standard counterparts. Grassroots bonds differ from traditional bonds not in their financial structure but in three respects: they are issued by any person rather than by institutions, traded bilaterally rather than on exchanges, and enforced bilaterally, on the record of the issuer's own date, rather than solely by legal contract. The economic effect is to democratise access to the bond market: any person can issue, hold, and trade bonds, with the community bank of Appendix E illustrating how a village can achieve the financial functionality of a bank using only grassroots bonds.

Bills of exchange and promissory notes. Before banks dominated payment, debts circulated as bills of exchange and promissory notes—signed promises to pay a fixed sum of a currency at a set date—traded and discounted before maturity [20, 21]. A grassroots bond is such an instrument in digital form: a signed, dated, transferable debt, sold before maturity at a discount (sale of debt, Section 4), with the protocol serving the purpose that transfer and proof-of-holdership rules serve for paper notes [24]; and each bond is the debt of its issuer: past holders bear no liability for it. A bill of exchange falls due by the calendar; a grassroots bond falls due by the calendar of its issuer, who undertakes to keep it current (Section 3).

9 Conclusion

Grassroots bonds add to grassroots coins a second instrument, a debt of its issuer with a maturity date, and with it the interest that 1-for-1 coin redemption precluded. A coin is money and a bond is credit, and a matured bond is redeemed as a coin is; the voluntary swap of coins and bonds and the escrow transfers are the acts from which the instruments of Sections 4 and 5 are composed. The contract of Section 3 is the contract of grassroots currencies with two clauses, Date and Escrow; the Date clause is undertaken, so the contract enforces the redemption of a bond only against the date its issuer keeps, and the obligation of a bond as a whole is undertaken, where a coin's is enforced. The ratios of Section 6 are the classical ones, with maturity as asset class, computed by each person on their own holdings; the clearing results are properties of the holdings, and hold once dates advance and persons act on their rights, with no coordination among them.

The contract has two realisations. With pen and paper the obligations are written on the notes, the date is that of the calendar, and both are enforced at law. Digitally they are enforced by the program every participant runs, in which the issuer applies their own date to their own bonds, and no party needs to see another's date.

Further work is the economy that is not a demonstration: issuers who default, counterparties who deviate from an agreement, and populations of thousands rather than six. Securing the implementation against the loss of a signing key or a device is the subject of prior work [8].

AI Disclosure. The authors used Anthropic's Claude models to assist with two parts of this work. First, the GLP code (Section 7) was derived from the specification of Sections 2 and 3

with substantial Claude assistance: the authors specified the transactions, supplied GLP language documentation, iteratively reviewed the generated code against the specification and against a running implementation, and made all design decisions. Second, the prose of the paper received Claude assistance for drafting and proofreading under the same review discipline. The authors verified the correctness and originality of all content, including all references. Responsibility for all content rests with the authors.

References

[1] Timothy Besley and Stephen Coate. 1995. Group Lending, Repayment Incentives and Social Collateral. *Journal of Development Economics* 46, 1 (1995), 1–18.

[2] Vitalik Buterin. 2014. A next-generation smart contract and decentralized application platform. *white paper* 3, 37 (2014). https://ethereum.org/whitepaper/

[3] Francesca Carapella, Grace Chuan, Jacob Gerszten, Chelsea Hunter, and Nathan Swem. 2023. *Tokenization: Overview and Financial Stability Implications.* Finance and Economics Discussion Series 2023-060. Board of Governors of the Federal Reserve System. doi:10.17016/FEDS.2023.060r1

[4] Luca Cardelli, Liav Orgad, Gal Shahaf, Ehud Shapiro, and Nimrod Talmon. 2020. Digital social contracts: A foundation for an egalitarian and just digital society. In *CEUR Proceedings of the First International Forum on Digital and Democracy*, Vol. 2781. CEUR-WS, 51–60. https://ceur-ws.org/Vol-2781/paper5.pdf

[5] Ricardo de O Cavalcanti and Neil Wallace. 1999. Inside and outside money as alternative media of exchange. *Journal of Money, Credit and Banking* (1999), 443–457.

[6] Thaddeus Dryja. 2017. Discreet Log Contracts. https://adiabat.github.io/dlc.pdf MIT Digital Currency Initiative.

[7] Ohad Eitan, Idit Keidar, and Ehud Shapiro. 2026. The Grassroots Super-App: A Secure Operating System for Grassroots Mini-Apps. arXiv:XXX:XXXXX. arXiv:XXX:XXXXX [cs.DC]

[8] Ohad Eitan, Idit Keidar, and Ehud Shapiro. 2026. Securing People and their Machines Against Major Faults (Full Version). *Proceedings of The 28th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2026), to appear* (2026). https://arxiv.org/abs/2607.02304 arXiv:2607.02304.

[9] Frank J. Fabozzi and Francesco A. Fabozzi. 2021. *Bond Markets, Analysis, and Strategies* (10th ed.). MIT Press.

[10] James Golike, Andy Lewis-Pye, and Ehud Shapiro. 2026. Grassroots Social Contracts: From Legal Text to an App (Full Version). *arXiv:2609.06231* (2026).

[11] Lewis Gudgeon, Sam M. Werner, Daniel Perez, and William J. Knottenbelt. 2020. DeFi Protocols for Loanable Funds: Interest Rates, Liquidity and Market Efficiency. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies (AFT '20)*. ACM, 92–112. doi:10.1145/3419614.3423254

[12] Idit Keidar and Ehud Shapiro. 2026. Volition Elicitation: Operational Semantics for People and Their Machines. arXiv:2607.14138. arXiv:2607.14138 [cs.DC] https://arxiv.org/abs/2607.14138

[13] Thomas Lambert, Daniel Liebau, and Peter Roosenboom. 2022. Security Token Offerings. *Small Business Economics* 59 (2022), 299–325. doi:10.1007/s11187-021-00539-9

[14] Pablo Lamela Seijas and Simon Thompson. 2018. Marlowe: Financial Contracts on Blockchain. In *Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2018) (LNCS, Vol. 11247)*. Springer, 356–375. doi:10.1007/978-3-030-03427-6_27

[15] Robert Leshner. 2020. Compound Governance. Blog post, Compound Labs. https://medium.com/compound-finance/compound-governance-5531f524cf68

[16] Andrew Lewis-Pye, Oded Naor, and Ehud Shapiro. 2023. Grassroots Flash: A Payment System for Grassroots Cryptocurrencies. *arXiv preprint arXiv:2309.13191* (2023). https://arxiv.org/abs/2309.13191

[17] Andy Lewis-Pye and Ehud Shapiro. 2026. Volition-Guarded Multiagent Atomic Transactions: Describing People and their Machines. *arXiv preprint arXiv:2604.25596* (2026). https://arxiv.org/abs/2604.25596

[18] Simon Peyton Jones, Jean-Marc Eber, and Julian Seward. 2000. Composing Contracts: An Adventure in Financial Engineering (Functional Pearl). In *Proceedings of the 5th ACM SIGPLAN International Conference on Functional Programming (ICFP '00)*. ACM, 280–292. doi:10.1145/351240.351267

[19] Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. 2021. Attacking the DeFi Ecosystem with Flash Loans for Fun and Profit. In *Financial Cryptography and Data Security – 25th International Conference, FC 2021*. Springer, 3–32. doi:10.1007/978-3-662-64322-8_1

[20] James Steven Rogers. 1995. *The Early History of the Law of Bills and Notes.* Cambridge University Press.

[21] Veronica Aoki Santarosa. 2015. Financing Long-Distance Trade: The Joint Liability Rule and Bills of Exchange in Eighteenth-Century France. *The Journal of Economic History* 75, 3 (2015), 690–719.

[22] Fabian Schär. 2021. Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets. *Federal Reserve Bank of St. Louis Review* 103, 2 (2021), 153–174. doi:10.20955/r.103.153-74

[23] Ilya Sergey, Amrit Kumar, and Aquinas Hobor. 2019. Scilla: a smart contract intermediate-level language. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*. ACM, 366–381.
[24] Ehud Shapiro. 2022. Grassroots Currencies: Turning Mutual Trust into Liquidity. *arXiv preprint arXiv:2202.05619* (2022). https://arxiv.org/abs/2202.05619
[25] Ehud Shapiro. 2025. GLP: A Grassroots, Multiagent, Concurrent, Logic Programming Language for AI (full version). *arXiv preprint arXiv:2510.15747, Summary in Proc. of ICLP'26, EPTCS 450, pp. 119–133* (2025). https://arxiv.org/abs/2510.15747
[26] Nick Szabo. 1997. Formalizing and Securing Relationships on Public Networks.
[27] Sam M. Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J. Knottenbelt. 2022. SoK: Decentralized Finance (DeFi). In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies (AFT '22)*. ACM, 446–461. doi:10.1145/3558535.3559780

A The Act Schemas of the Contract

This appendix writes the clauses of Section 3.1 as act schemas [10], states the modality of each clause, and shows the contract syntactically grassroots. Act schemas allow numbers [10]: an atom's argument may be a number, a schema may carry a constraint on its numeric variables, written after its head, which every binding satisfies, and an atom added or deleted may carry a multiplicity. A number names no person, so the three conditions and the theorem of [10] concern the persons a schema names. In the repository the schemas are a term the checker of [10] reads, `programs/currencies/bonds/GB.glp`, in which a coin $¢(u, \bot)$ is $coin(u)$ and a bond $¢(u, d)$ is $bond(u, d)$, so a schema whose maturity arguments range over both is written once for each kind, and the table below writes mint and redeem once for a coin and once for a bond, as there a coin reads no date. The contract extends GKK [10], so it holds GKK's three schemas, cited from there and not restated.Establish adds *identity* at its one role. Meet is guarded in both roles and adds *knows* of the other at each. Introduce is guarded in both roles: the first requires *knows* of the second and of a third person c, and the second gains $knows(c)$. Every role but establish's requires *identity*, which the table below leaves out. The contract of grassroots currencies has four schemas—mint, swap, pay and redeem—over one predicate, ¢ of arity one, a coin naming its issuer [24]. Here ¢ gains a maturity date, which a coin takes as $\bot$, redeem presents a coin or a matured bond and takes a coin or a bond, and advance and the three escrow transfers are new schemas.

A.1 The Act Schemas

The predicates are GKK's *identity*, of arity zero, and *knows*, of arity one, $knows(u)$ in the state of p saying that p holds u's public key, and two of the contract's own. ¢, of arity two: $¢(u, d)$ in the state of p is a bond issued by u with maturity date d, held by p, and $¢(u, \bot)$ is a coin of u, due now; d ranges over $\mathbb{N} \cup \{\bot\}$. *date*, of arity one: $date(t)$ in the state of p is the date of p, and *dated*, of arity zero, that p has a date. $today(\delta)$ in the state of p is the day: an *environment predicate* [10], held by the machine from startup and kept current by the platform from the device's calendar [12], which no schema adds or deletes and a schema may require. The roles are p, q and e, Alice, Bob and Carol of Section 3.1; the party variables are u, v and r; the numeric variables are d, d' for maturity dates, t, t' for dates, δ for the day, and k, k' for amounts. The six clauses that describe acts become eleven schemas: mint, redeem and Date two each, a coin and a bond, the day and the next day, and escrow three.

"Alice may decide" is the single guarding role of mint, of pay and of redeem; "Alice and Bob may jointly decide" is the two guarding roles of the swap, and "Carol and Bob" and "Carol and Alice" those of release and return; a deposit with Carol is guarded by Alice and Carol, as the agent takes it; first_date and advance are guarded by no role, as the calendar asks no one: first_date gives a party the day as their date, once, and advance takes their date to the next day while the day is not past, so a calendar that has moved on several days is as many advances. "Who know each other" is

schema	*at its first role*	*at its second role*
$mint(p?;\, k)$:	$+k \cdot ¢(p, \perp)$	
$mint(p?;\, d, k, t) \mid d > t$:	$date(t),\ +k \cdot ¢(p, d)$	
$first_date(p;\, \delta)$:	$today(\delta),\ \neg dated,\ +dated,\ +date(\delta)$	
$advance(p;\, t, t', \delta) \mid t' = t + 1 \wedge t' \leq \delta$:	$today(\delta),\ -date(t),\ +date(t')$	
$swap(p?, q?;\, u, d, k, v, d', k')$:	$knows(q),\ -k \cdot ¢(u, d),\ +k' \cdot ¢(v, d'),\ +knows(v)$	$knows(p),\ -k' \cdot ¢(v, d'),\ +k \cdot ¢(u, d),\ +knows(u)$
$pay(p?, q;\, k)$:	$knows(q),\ -k \cdot ¢(q, \perp)$	$+k \cdot ¢(q, \perp)$
$redeem(p?, q;\, r, d')$:	$knows(q),\ -¢(q, \perp),\ +¢(r, d'),\ +knows(r)$	$-¢(r, d'),\ +¢(q, \perp)$
$redeem(p?, q;\, d, r, d', t) \mid d \leq t$:	$knows(q),\ -¢(q, d),\ +¢(r, d'),\ +knows(r)$	$date(t),\ -¢(r, d'),\ +¢(q, d)$
$deposit(p?, e?;\, u, d, k)$:	$knows(e),\ -k \cdot ¢(u, d)$	$knows(p),\ +k \cdot ¢(u, d),\ +knows(u)$
$release(e?, q?;\, u, d, k)$:	$knows(q),\ -k \cdot ¢(u, d)$	$knows(e),\ +k \cdot ¢(u, d),\ +knows(u)$
$return(e?, p?;\, u, d, k)$:	$knows(p),\ -k \cdot ¢(u, d)$	$knows(e),\ +k \cdot ¢(u, d),\ +knows(u)$

that each role of the swap, and of each escrow transfer, requires *knows* of the other, and "whom they know" that pay and redeem require $knows(q)$ at p; and a coin or a bond carrying its issuer's public key (Section 3.1) is that swap, redeem and each escrow transfer add *knows* of the issuer at the role that receives the instrument, while pay adds none, the payee receiving coins of their own issue. The amount minted is the argument k of mint, and "of a maturity later than their own date" is the constraint of the mint of a bond, a date past the minter's own, so no bond is born mature; the mint of a coin reads no date. A bond is mature when its maturity date is at most its issuer's date: the redeem of a bond requires $date(t)$ of the issuer q, in q's role, under the constraint $d \leq t$, so maturity is judged by the issuer's date and the holder's date plays no part, and it records no date. It is the one schema that reads a date of another role; the redeem of a coin, like pay, reads no date. "Bob-coins they hold" is that pay requires of p coins of q, and "a Bob-coin, or a matured Bob-bond, they hold" that redeem requires of p an instrument of q; "any coin or bond held by Bob" is redeem's party variable r and maturity d', unconstrained.The mutual credit line of clause 3 is the swap at a binding sending u to the party at p, v to the party at q, d and d' to $\perp$, and k' to k; the loan is the swap at a binding with $k' > k$, $d = \perp$, and d' later than the borrower's date. Release and return differ in who receives: the party at q in release, the depositor at p in return; which of the two the agent wills is fixed by the escrow agreement (Section 5). The swap exchanges one lot—coins of one issuer, or bonds of one issuer and maturity—against another; an instrument that exchanges several lots at once is an escrow agreement, the packaged exchange of Section 5. A binding at which a role adds what it deletes—the swap of a lot for the same lot or for more of it, or the redemption of a q-coin for a q-coin—determines no transaction [10], which is why the two lots of a swap are of different issuers, kinds or maturities.

A.2 The Modality of Each Clause

The definitions are [10]'s.A *clause* of a contract is a proposition about the conduct of its parties, kept or breached by that conduct, and a party it names as answerable is its *obligor*; the runs of the protocol realising the contract *determine* the clause if whether it is kept is a function of the run, and a transaction *may breach* it if it occurs in some correct run that breaches it. A transaction is *witnessed against* a party if after it another of its participants holds an atom naming that party. A clause with obligor o is *enforced* if the runs determine it and every correct run keeps it; *attested* if some transaction may breach it and every transaction that may is witnessed against o; and *undertaken* if no transaction may breach it, the runs do not determine it, and some transaction is witnessed against o or o's undertaking is held by another party.

Clauses 1, 3, 4, 5 and 7 are enforced. Each permits an act and obliges nothing beyond it, so the runs determine it and every correct run keeps it. Two of them oblige a counterparty, and the schemas above make the obligation self-executing: "Alice may decide" in clause 4 is that pay is guarded by the payer and adds the coins at the payee, and in clause 5 that redeem is guarded by the bearer and deletes the instrument taken at the issuer. Neither the payee nor the issuer is asked.

Each of the three transfers of clause 7 is guarded by the two parties to it. No clause of the contract is attested: no act of it carries a warranty, and pay adds a coin naming its issuer and not its payer, so no clause against the payer could be [10]. The modalities of GKK:1 to GKK:3 are stated in [10].

Clause 6 is undertaken, as in grassroots currencies [24].Selling at an advertised price is no act of the contract, so no transaction of it may breach the clause and no run determines whether it is kept; and the swap opening a mutual credit line adds $¢(p, \perp)$ in the state of the party at q, naming the issuer p, so the coin in another party's holding is the record of the undertaking.

Clause 2 is undertaken. It relates the date a party keeps to the calendar, which is outside the contract, so no transaction of it may breach the clause and no run determines whether it is kept; in an asynchronous system no record can attest the date a party keeps. A q-bond in the holding of another party is the record of q's undertaking, as a q-coin is of pricing. The redeem of a bond reads the date of the issuer and records nothing, and no other schema reads a date of another role; first_date and advance read the day the platform writes.

Whether a bond is mature is subject to its issuer advancing their date, so the contract enforces the redemption of a bond only against the date its issuer keeps. Once the date of q has reached the maturity of a q-bond, its bearer redeems it at will, as a coin, and that act is enforced, as clause 5. Until then no transaction of the contract redeems it, and an issuer whose date lags owes nothing yet on their bonds, as none of them has matured; if the calendar is past the maturity, q is in breach of their undertaking of clause 2, and no record of the contract witnesses it. Their debts do not fall due, and others price their currency accordingly or decline to hold it (Section 2). The obligation of a bond as a whole—to redeem it once its maturity date has come—is therefore undertaken; the obligation of a coin is enforced.

A.3 The Contract Is Grassroots

A contract is syntactically grassroots if it has an unobstructed introductory act, every predicate its acts require has traceable provenance, and every act not willed by all its parties runs only between parties who already record each other; the protocol realising a syntactically grassroots contract is volitionally grassroots [10]. The three conditions quantify over the people a state records, and the numeric arguments and constraints of the schemas above name no person, so the conditions and the theorem apply to them as stated. The check for the four schemas of grassroots currencies is in [24]; a maturity date names no person, so that check stands, and what follows is the check for the whole contract, the new schemas and predicates included.

Introduction. Meet is an introductory act: it is guarded in both roles, it adds $knows(q)$ at p, naming q, and $knows(p)$ at q, naming p. It is unobstructed: what each role requires, *identity*, is added by establish, a schema of arity one, guarded in its role, requiring nothing and forbidding only the identity it adds, so a person establishes once; and no role of meet forbids anything. It is the introductory act of every contract extending GKK [10].

Provenance. *knows* has traceable provenance: meet adds it naming a role; introduce adds $knows(c)$ at q, where c is an argument of an atom p requires; and swap, redeem and each escrow transfer add *knows* of an issuer at the receiving role, the issuer being an argument of an atom the other role requires. ¢ has traceable provenance. Mint adds $¢(p, d)$ at p, naming its own role. Swap adds $¢(v, d')$ at p and $¢(u, d)$ at q, and v and u are arguments of the atoms the other role requires. Pay adds $¢(q, \perp)$ at q, naming a role. Redeem adds $¢(r, d')$ at p, where r is an argument of the atom q requires, and $¢(q, d)$ at q, naming a role. Each escrow transfer adds at its receiving role an atom naming u, an argument of the atom its giving role requires. *date*, *dated* and *today* name no person.

Volition. Of the schemas of arity two, meet, introduce, swap, deposit, release and return are guarded in both roles. Pay and redeem are guarded by p, and in each p requires $knows(q)$ and an

instrument of q, atoms of traceable provenance naming q, so the role graph is connected: a party pays or presents to an issuer they know and whose coin or bond they hold.

The contract is therefore syntactically grassroots, and the protocol realising it is volitionally grassroots (Corollary 3.5): any group of persons operates it on its own, and two groups become connected only by a meeting willed by both.

B Deferred Proofs

This appendix proves the correctness of the credit line of Section 5 and the clearing results of Section 6. The syntactic conditions establishing that grassroots bonds are volitionally grassroots (Corollary 3.5) are checked in Appendix A.

Proposition 5.1 (Credit Line) (restated). *Let k be the limit, T the expiry, and k_d the drawn amount of the credit line above. In any safe run in which e follows the escrow program and p and q transact only as the agreement specifies, between the operations of the agreement—each packaged exchange or transfer complete—e holds $¢_p^{k-k_d}$, p holds $¢_{q,T}^{k_d}$ together with the coupon bonds of the draws, and $0 \le k_d \le k$. At expiry with $k_d = 0$, p holds $¢_p^k$, and the q-bonds p holds are the coupon bonds of the draws.*

Proof. By induction on the operations of the agreement, in the order they complete in the run.

Base. At establishment p mints $¢_p^k$ and transfers them to e, and $k_d = 0$. So e holds $¢_p^k$, p holds no q-bond, and $0 \le k_d \le k$.

Draw. Let the invariant hold with drawn amount k_d, and let q draw $k' \le k - k_d$. By the invariant e holds $¢_p^{k-k_d} \supseteq ¢_p^{k'}$, so once q has deposited $¢_{q,T}^{k'}$ and the coupon bonds with e, the releases complete: they move $¢_{q,T}^{k'}$, with the coupon bonds, from e to p, and $¢_p^{k'}$ from e to q. Afterwards e holds $¢_p^{k-k_d-k'}$ and p holds $¢_{q,T}^{k_d+k'}$, and the new drawn amount is $k_d + k' \le k$. The invariant holds.

Repayment. Let the invariant hold with drawn amount k_d, and let q repay $k_{\text{rep}} \le k_d$. By the invariant p holds $¢_{q,T}^{k_d} \supseteq ¢_{q,T}^{k_{\text{rep}}}$, so each redemption is enabled. The k_{rep} redemptions move $¢_p^{k_{\text{rep}}}$ from q to p and $¢_{q,T}^{k_{\text{rep}}}$ from p to q; p then transfers the coins to e. Afterwards e holds $¢_p^{k-k_d+k_{\text{rep}}}$ and p holds $¢_{q,T}^{k_d-k_{\text{rep}}}$, and the new drawn amount is $k_d - k_{\text{rep}} \ge 0$. The invariant holds.

Expiry. Once $d_e^* \ge T$, e transfers $¢_p^{k-k_d}$ to p. With $k_d = 0$ this is $¢_p^k$, and by the invariant the q-bonds p holds beyond the coupon bonds of the draws are $¢_{q,T}^0$; as p and q transact only as the agreement specifies, p holds no other q-bond.

No other operation of the agreement changes the p-coins held by e or the principal q-bonds held by p: by hypothesis the parties take no transaction outside the agreement. Hence the invariant holds between the operations of the agreement throughout the run. □

Proposition 6.3 (Graph Solvency) (restated). *The holdings of a sink-closed set V clear if and only if $a_p \ge \ell_p$ for every member p of V other than s, and cleared holdings leave each such member holding exactly $a_p - \ell_p$ coins and bonds of the sink.*

Proof. A clearing redemption among members of V surrenders a coin or a matured bond of a member $q \ne s$ and takes a coin or a bond of q's holdings of an issuer $r \ne q$; if $r \ne s$ then $r \in V$, by sink-closedness of the edge $q \to r$. The surrendered instrument becomes self-held and the instrument taken moves to the redeemer, so every edge the redemption deletes, creates, or moves

is incident to s or has both endpoints in V; edges outside V are untouched, and V remains sink-closed along any safe computation of clearing redemptions among its members and Advance-date transitions.

Write $n_u := a_u - \ell_u$. A Redeem in which holder p surrenders a q-coin or a matured q-bond and takes an instrument y of issuer r held by q changes the counts as follows. The surrendered instrument moves to its issuer: a_p and ℓ_q decrease by 1. If $r \notin \{p, q\}$, y moves from holder q to holder p: a_q decreases and a_p increases by 1. If $r = p$, y moves to its issuer: a_q and ℓ_p decrease by 1. If $r = q$, y moves from its issuer to p: a_p and ℓ_q increase by 1. In every case n_u is unchanged for every person u, and Advance-date changes no holdings. Every n_u is therefore invariant along any safe computation of Redeem and Advance-date transitions.

Only if: cleared holdings of V have $\ell_p = 0$ for every member p of V other than s, so there $n_p = a_p \geq 0$; n_p is invariant, so $a_p \geq \ell_p$ in the given configuration.

If: by induction on $C := \sum_{p \in V \setminus \{s\}} \ell_p$. If $C = 0$ the holdings of V are cleared. Otherwise some person p holds an instrument of an issuer $q \in V \setminus \{p, s\}$, and p is a member of V, as the edge $p \to q$ is incident to s or lies in V. If that instrument is a bond, q advances d_q^* to its maturity date, so it is mature. As $a_q \geq \ell_q \geq 1$, q holds an instrument y of some issuer $r \neq q$; p knows q (Lemma 3.3), so p redeems the instrument against y—a clearing redemption among members of V. By the count changes above, the redemption decreases C by 1 or 2 and leaves every n_u unchanged, so the condition persists and the induction applies. At most C redemptions clear the holdings of V.

In cleared holdings of V, every instrument of another issuer that a member $p \neq s$ holds is a coin or a bond of the sink—its issuer is s or in V by sink-closedness, and every issuer in V other than s is cleared—and there are $a_p - \ell_p$ of them, as n_p is invariant and $\ell_p = 0$ there. □

Proposition 6.5 (Clearing Liveness) (restated). *If $a_p \geq \ell_p$ for every member p of V other than s, then every infinite safe computation of clearing redemptions among members of V and Advance-date transitions of members of V that satisfies eventual redemption reaches cleared holdings of V, after at most $\sum_{p \in V \setminus \{s\}} \ell_p$ redemptions.*

Proof. A clearing redemption among members of V decreases $C = \sum_{p \in V \setminus \{s\}} \ell_p$ by 1 or 2 and leaves every n_u unchanged (proof of Proposition 6.3), and Advance-date transitions change no holdings, so the computation takes at most C redemptions. Suppose the holdings of V after the last redemption are not cleared: some person p holds an instrument of an issuer $q \in V \setminus \{p, s\}$ in every subsequent configuration, and p is a member of V, as the edge $p \to q$ is incident to s or lies in V. If that instrument is a bond, by (*i*) d_q^* eventually reaches its maturity date, from which it is mature in every subsequent configuration. As $a_q \geq \ell_q \geq 1$ and no transaction of the suffix changes any holding, q holds an instrument y of some issuer $r \neq q$ in every subsequent configuration, and p knows q (Lemma 3.3). The clearing redemption in which p surrenders the instrument and takes y is then enabled in every configuration of a suffix, with no clearing redemption occurring in the suffix, contradicting (*ii*). □

C Escrow-Based Instruments, Continued

The escrow-based instruments deferred from Section 5: collateral, guarantees, options, insurance, credit default swaps, and letters of credit. The notation, the escrow agent e, and the trust assumptions are those of Section 5.

(1) **Collateral:** To collateralise a loan from lender p to borrower q:
   (a) **Deposit.** $\{(q, x),\ (e, \emptyset)\}$: borrower q transfers bonds x to escrow agent e.
   (b) **Release on default.** $\{(e, x),\ (p, \emptyset)\}$: e transfers collateral to lender p.
   (c) **Return on fulfilment.** $\{(e, x),\ (q, \emptyset)\}$: e returns collateral to borrower q.

(2) **Guarantee:** Guarantor g covers the obligations of borrower q to lender p:
   (a) **Deposit.** $\{(g, \¢_g^k),\ (e, \emptyset)\}$: g deposits g-coins with escrow agent e.
   (b) **Invocation on default.** $\{(e, \¢_g^k),\ (p, \emptyset)\}$: e releases g-coins to lender p.
   (c) **Release on fulfilment.** $\{(e, \¢_g^k),\ (g, \emptyset)\}$: e returns g-coins to g.

(3) **Option:** An option grants p the right, but not the obligation, to execute a specified swap with q at or before expiry date d, judged by escrow agent local date d_e^*, with establishment-window T_0 before which both parties must complete their deposits. The option premium and the underlying assets are held in escrow:
   (a) **Establishment.** p deposits the premium $\¢_p^k$ with escrow agent e: $\{(p, \¢_p^k),\ (e, \emptyset)\}$. q deposits the underlying bonds y with e: $\{(q, y),\ (e, \emptyset)\}$. If only one party deposits by establishment-window T_0, e returns the deposit to its depositor.
   (b) **Exercise.** While $d_e^* \leq d$, p may signal exercise; e releases the underlying y to p: $\{(e, y),\ (p, \emptyset)\}$, and releases the premium to q: $\{(e, \¢_p^k),\ (q, \emptyset)\}$.
   (c) **Expiry.** Once $d_e^* > d$, if p has not exercised, e returns the underlying to q: $\{(e, y),\ (q, \emptyset)\}$, and transfers the premium to q: $\{(e, \¢_p^k),\ (q, \emptyset)\}$.

   The above specifies a *call option*: p acquires the right to buy the underlying y from q. A *put option*—the right to sell bonds to q at a predetermined price—is symmetric: q deposits the strike price with e, and p deposits the bonds they wish to sell.
   Exercise is specified by a window $[d_1, d_2]$ agreed at establishment: p may exercise while $d_1 \leq d_e^* \leq d_2$, and expiry is once $d_e^* > d_2$. The specification above uses *American* exercise, the window $[0, d]$; *European* exercise is the window $[d, d_2]$, with d_2 the last date on which e accepts exercise. A window is needed because Advance-date is unguarded and may raise d_e^* by more than one day, so d_e^* can pass d without equalling it and the condition $d_e^* = d$ may never hold.
   The option above is prepaid: the premium is the whole price of y. A *strike price* is k' further p-coins that p deposits with e at establishment; on exercise e releases them to q with the premium, and on expiry e returns them to p.

(4) **Insurance:** Party p (the insured) pays a premium to escrow agent e, who disburses a payout if a specified event occurs by contract expiry T (judged by d_e^*), as attested by e:
   (a) **Establishment.** $\{(q, \¢_q^{k'}),\ (e, \emptyset)\}$: the insurer q deposits the reserve $\¢_q^{k'}$, $k' \gg k$, with e.
   (b) **Premium.** $\{(p, \¢_p^k),\ (e, \emptyset)\}$: p deposits the premium with e.
   (c) **Claim (event occurs).** $\{(e, \¢_q^{k'}),\ (p, \emptyset)\}$ and $\{(e, \¢_p^k),\ (q, \emptyset)\}$: e transfers the reserve to p as the payout, and the premium to the insurer q.
   (d) **Expiry (no event).** $\{(e, \¢_p^k \cup \¢_q^{k'}),\ (q, \emptyset)\}$: once $d_e^* \geq T$ with no claim, e transfers the premium to insurer q and returns its reserve.

   The escrow agent e serves as the claims adjudicator.A mutual insurance arrangement among a group of participants can be realised by each member depositing premiums into a shared escrow reserve.

(5) **Credit default swap (CDS):** Party p pays periodic premiums to q (the protection seller), and q compensates p if a reference entity r defaults. Escrow agent e adjudicates:
   (a) **Establishment.** $\{(q, \¢_q^K),\ (e, \emptyset)\}$: the protection seller q deposits the reserve $\¢_q^K$ with e.
   (b) **Premium payments.** Periodic swaps $\{(p, \¢_{p,d_j}^{k_j}),\ (e, \emptyset)\}$ at dates d_j judged by d_e^*: p deposits premium bonds with e, forwarded to q.
   (c) **Credit event.** $\{(e, \¢_q^K),\ (p, \emptyset)\}$: if e determines that r has defaulted, e releases the reserve coins of q to p.

(d) **No credit event.** $\{(e, \mathfrak{c}^{k_j}_{p,d_j}),\ (q, \emptyset)\}$ and, at contract expiry T (when $d^*_e \geq T$), $\{(e, \mathfrak{c}^{K}_{q}),\ (q, \emptyset)\}$: premium bonds are released to q, and the reserve coins are returned to q.

(6) **Letter of credit:** A bank b guarantees payment to seller p on behalf of buyer q, with escrow agent e verifying that contractual conditions (e.g. delivery of goods) are met:

(a) **Issuance.** $\{(b, \mathfrak{c}^{k}_{b}),\ (e, \emptyset)\}$: bank b deposits b-coins with e.

(b) **Presentation.** $\{(e, \mathfrak{c}^{k}_{b}),\ (p, \emptyset)\}$: upon e verifying that p has fulfilled the terms (e.g. delivery attestation), e releases b-coins to seller p.

(c) **Reimbursement.** $\{(q, \mathfrak{c}^{k}_{q,d}),\ (e, \emptyset)\}$ and $\{(e, \mathfrak{c}^{k}_{q,d}),\ (b, \emptyset)\}$: buyer q deposits their bonds with e, which releases them to bank b.The reimbursement goes through e because q-bonds are not coins of b and so cannot be paid to it, and there is no other act by which bonds pass one way between two ordinary parties (Section 5).

D The Liquidity Ratios: Worked Example

We compute the cash, quick, and current ratios of Section 6 for two agents of the village market scenario (Section 7.1), at the snapshot immediately after step 2 (asymmetric credit), with both agents' local date $d^* = 0$, near-term horizon $\delta = 90$ days, and operating cycle $\Delta = 1$ year. At this point the relevant holdings are as follows.

Bob holds 15 alice-coins (from step 1's symmetric mutual credit with Alice) and 20 diana-coins (from the loan Diana extended in step 2): 35 foreign coins. Bob has issued 15 bob-coins (held by Alice) and 24 bob-bonds (held by Diana, maturing day 25); the coins are due now and the bonds mature within Δ, so the current liabilities are 39.

Diana holds 24 bob-bonds (maturing day 25) and 18 frank-bonds (maturing day 28), 42 foreign bonds in total, and no foreign coins. Diana has issued 20 diana-coins (held by Bob) and 15 diana-coins (held by Frank); all 35 diana-coins are due now and hence within Δ, so the current liabilities are 35.

Applying the formulas with common denominator (current liabilities):

Cash Ratio of Bob	$= 35/39$	≈ 0.90
Quick Ratio of Bob	$= 35/39$	≈ 0.90
Current Ratio of Bob	$= 35/39$	≈ 0.90
Cash Ratio of Diana	$= 0/35$	$= 0$
Quick Ratio of Diana	$= 42/35$	≈ 1.20
Current Ratio of Diana	$= 42/35$	≈ 1.20

Bob's three ratios coincide: he holds only foreign coins, so widening the horizon adds nothing to his numerator. His common value of 0.90 reflects that his 35 foreign coins fall short of his 39 current liabilities.

Diana's cash ratio is zero—she holds no foreign coins on hand—but her quick and current ratios both reach 1.20, reflecting full coverage of her current liabilities by the bob-bonds and frank-bonds she will collect within the operating cycle: Diana is illiquid and solvent (Section 6). Her quick and current ratios coincide because all foreign bonds she holds mature within δ; in a scenario with longer-dated holdings the current ratio would exceed the quick.

E The Community Bank with Bonds

A community bank, or credit union, issues its own grassroots currency and opens credit lines with community members, whose members undertake to accept payments in the community currency; it needs no initial capital, and it manages its exposure to a member by redeeming that member's

coins against the coins of other members it holds [24]. Bonds add to it lending at interest, deposits, collateral, and risk management by maturity.

Bank lending via bonds. The bank lends community coins to member p in exchange for p-bonds with maturity and interest. For example, the bank mints 10,000 community coins for p, and p mints 10,500 p-bonds maturing in six months—500 bonds as interest. The bank holds interest-bearing claims on p; p holds liquid community coins accepted by all members. This is a loan of Sections 4 and 5 (zero-coupon, balloon, or fixed-payment), with the bank as lender, or a credit line (Section 5), with periodic interest on the amount drawn. The bank may increase its capital by charging interest on drawn credit, as well as transaction fees.

Deposits. A member may deposit community coins with the bank and receive community bonds maturing at d with interest—for example, 10,000 community coins deposited, 10,200 community bonds maturing in six months returned. The bank's income is the spread between lending and deposit rates.

Collateral. Personal coins—those acquired by community members through bilateral mutual credit—serve as collateral for bank loans. The bank may make the loan conditional on borrower p depositing non-p-coins they hold into escrow (Appendix C) as security. If p defaults, the escrowed bonds are released to the bank. If p fulfils the loan, the collateral is returned. The bank does not hold or manage personal coins directly; it holds p-bonds and, when needed, escrowed collateral.

Risk management. The bank manages risk through the maturity structure of its loan portfolio, collateral requirements at origination, the liquidity ratios of Section 6 applied to its borrowers, and the ability to sell debt at a discount (Section 4, sale of debt). As in traditional banking, a fixed-term loan cannot be recalled before maturity; the bank must arrange protection at origination or sell the debt to a third party. Risk management by redemption [24] reaches the member's coins and their matured bonds.

F From the Transactions to the Program

The program of Section 7 is written in vGLP [12], GLP with volitional procedures, which put a question to the person, and is derived from Definition 3.2 as the program of grassroots currencies is derived from its four transactions [24]. The agent is a volitional procedure whose question is the stream of its person's requests, one request for each transaction a party initiates; `serve`, the agent with the requests not yet served, has one clause per request, and `on_act` has one clause per act that arrives from another party. The swap and its responder, pay and redeem, and what the vGLP compiler derives from the source—the interface constructs and the dispatcher—are as in the coins paper, and are not restated; here is what dates and escrow add, and the channel two parties transact over. The program is at https://github.com/EShapiro2/GLP, in `programs/currencies/bonds`: the mini-app in `dated`—the source `bonds_agent.vglp`, its compilation `bonds_agent.glp`, the escrow programs `escrow.glp` and the entry `miniapp.glp`—and the plays `village.glp` and `credit_line.glp`.

The agent. Each party runs a *bond agent*, the procedure `agent` of the source, whose state is the machine state of Definition 3.2: its holdings, each coin as `coin(Issuer, Serial)` and each bond as `bond(Issuer, Maturity, Serial)`; its date; and the serial its next instrument takes. Its arguments are its name, its own input, the input from its counterparties, the conversations the social graph hands it, the calendar, its outputs, and the three components of its state. Its one clause asks its question once and passes the stream to `serve` with these arguments, and every clause of `serve` that continues the agent passes on the rest of the stream, `Reqs`. A lot—coins of one issuer, or bonds of one issuer and maturity—is `lot(Issuer, Due, Count)`, the $¢^{k}_{u,*}$ of the definition, its due `now` for coins and the maturity for bonds.

Conversations. Two parties transact over the root channel of the conversation the super-app opens between their two executions of the program, at an invitation of one friend by the other, as in the coins paper: the agent takes each conversation from the social graph and adopts its channel; a conversation is between two persons who know each other, so the preconditions on K of Definition 3.2 hold of every counterparty; and an act willed towards a party with whom no conversation is open is not sent—the instruments stay, and the screen says so.

Mint. $\mathrm{Mint}_p(k)$ is as in the coins paper. $\mathrm{Mint}_p(d,k)$ is the request `mint(K, D)`, whose two fields are the amount and the maturity; the guard of its clause carries the precondition $d > d_p^*$, so no bond is born mature, and its effect is $c_p' = c_p \uplus \mathfrak{c}_{p,d}^k$:

```
serve([mint(K, D)|Reqs], Id, UserIn, PeerIn, Convs, Cal, Outs,
      Holdings, Date, Serial) :-
    ground(Id?), integer(K?), K? >= 1, integer(D?), D? > Date? |
    create_bonds(Id?, D?, K?, Serial?, New),
    Serial1 := Serial? + K?,
    append_instruments(Holdings?, New?, Holdings1),
    show(minted(K?, D?), Outs?, Outs1),
    after_change(Reqs?, Id?, UserIn?, PeerIn?, Convs?, Cal?,
                 Outs1?, Holdings1?, Date?, Serial1?).
```

Advance-date and the calendar. $\mathrm{Advance}_p(t)$ is guarded by no one, so it is a clause of `serve` that reads no request: the calendar reaches the agent as a stream of dates, the sixth argument of `serve`, and a date past the agent's advances it by one day, the effect $d_p'^* = d_p^* + 1$, and stays at the head of the calendar, so a date k days past the agent's is taken as k advances, while one not past it is dropped:

```
serve(Reqs, Id, UserIn, PeerIn, Convs, [T|Cal], Outs, Holdings,
      Date, Serial) :-
    ground(Id?), T? > Date? |
    Date1 := Date? + 1,
    show(date_advanced(Date1?), Outs?, Outs1),
    serve(Reqs?, Id?, UserIn?, PeerIn?, Convs?, [T?|Cal?], Outs1?,
          Holdings?, Date1?, Serial?).

serve(Reqs, Id, UserIn, PeerIn, Convs, [T|Cal], Outs, Holdings,
      Date, Serial) :-
    ground(Id?), T? =< Date? |
    serve(Reqs?, Id?, UserIn?, PeerIn?, Convs?, Cal?, Outs?,
          Holdings?, Date?, Serial?).
```

Who writes the calendar is the platform's affair, not the program's: on a device the clock writes it; in the headless run below the play writes each party's, so that the dates advance as the scenario needs them. Two parties may therefore disagree on the date, as Section 2 allows, and nothing in the program compares one party's date with another's.

Pay and redeem. $\mathrm{Pay}_{p,q}(k)$ reads no date: a payment carries q's coins, which q's agent takes without asking its person, returning anything else. $\mathrm{Redeem}_{p,q}(x,r,*)$ is guarded by p: the request `redeem(Q, D, R, D1)`, whose four fields are the issuer presented to, the due of the instrument presented, `now` for a coin and the maturity for a bond, and the issuer and the due of the instrument wanted; in its clause the precondition $x \in c_p$ is the selection of the instrument presented:

```
serve([redeem(Q, D, R, D1)|Reqs], Id, UserIn, PeerIn, Convs, Cal,
```

```
      Outs, Holdings, Date, Serial) :-
    ground(Id?), ground(Q?), ground(D?), ground(R?), ground(D1?) |
    take_instrument(Q?, D?, Holdings?, Found, Rest),
    do_redeem(Found?, Id?, Q?, D?, R?, D1?, Rest?, Reqs?, UserIn?,
              PeerIn?, Convs?, Cal?, Outs?, Date?, Serial?).
```

The instrument reaches q with a response variable, and q's agent answers without asking its person, as clause 5 obliges. A coin of q it takes back, and a bond of q mature by q's date—the precondition $d \leq d_q^*$—likewise, and gives the instrument wanted; a bond of q not mature by q's date goes back:

```
fulfil(bond(I, M, S), R, D1, Resp?, Id, From, Reqs, UserIn,
       PeerIn, Convs, Cal, Outs, Holdings, Date, Serial) :-
    I? =?= Id?, M? =< Date? |
    take_instrument(R?, D1?, Holdings?, Found, Rest),
    give(Found?, bond(I?, M?, S?), R?, D1?, Resp, Id?, From?,
         Reqs?, UserIn?, PeerIn?, Convs?, Cal?, Outs?, Rest?,
         Date?, Serial?).

fulfil(bond(I, M, S), _, _, redeem_not_mature(bond(I?, M?, S?)),
       Id, From, Reqs, UserIn, PeerIn, Convs, Cal, Outs, Holdings,
       Date, Serial) :-
    I? =?= Id?, M? > Date? |
    show(not_mature_for(From?, M?), Outs?, Outs1),
    serve(Reqs?, Id?, UserIn?, PeerIn?, Convs?, Cal?, Outs1?,
          Holdings?, Date?, Serial?).
```

Where q holds no instrument of the issuer and due wanted, it returns the presented instrument with the lots of other issuers it holds, the menu of the coins paper with the due of each lot, for the bearer to choose among. A bond returned records nothing, as no transaction took place.

The escrow transfers. $\text{Deposit}_{p,e}$, $\text{Release}_{e,q}$ and $\text{Return}_{e,p}$ are transfers of one lot guarded by both participants, so each is a request at the giver and a responder at the receiver, as the swap is. The clauses of the three requests differ in the kind of transfer they name, which the receiver's card shows; the deposit's is

```
serve([deposit(E, U, D, K)|Reqs], Id, UserIn, PeerIn, Convs, Cal,
      Outs, Holdings, Date, Serial) :-
    ground(Id?), ground(E?), ground(U?), ground(D?), integer(K?),
        K? >= 1 |
    select_instruments(U?, D?, K?, Holdings?, Status, Give, Rest),
    propose_transfer(Status?, deposit, Id?, E?, lot(U?, D?, K?),
                     Give?, Rest?, Reqs?, UserIn?, PeerIn?,
                     Convs?, Cal?, Outs?, Date?, Serial?).
```

and the responder is a volitional procedure whose question is a card: its clause writes the offer on the card at once and hands the person the writer of their answer, which `decide_transfer` reads with a clause per answer:

```
(card(offer_transfer(Kind?, From?, Given?),
      Answer))*respond_transfer(offer_transfer(Kind, From, Given),
      Resp?, Ans?) :-
    ground(Kind?), ground(From?), ground(Given?) |
    decide_transfer(Answer?, Kind?, From?, Given?, Resp, Ans).
```

```
decide_transfer(yes, Kind, From, Given, Resp?, Ans?) :-
    accept_transfer_answer(Kind?, From?, Given?, Resp, Ans).
decide_transfer(no, Kind, From, Given, Resp?, Ans?) :-
    decline_transfer_answer(Kind?, From?, Given?, Resp, Ans).
```

The instruments given are set aside at the giver until the receiver answers, and go to the receiver on acceptance or back on refusal; the transfer is taken only when both guards are willing, as the definition states.

The escrow program. The escrow agent *e* of Section 5 is an ordinary party running the same program. What makes it an escrow agent is the *escrow program* of `escrow.glp`, which takes the place of its person: at every transfer card it answers whether *e* accepts, and at every message of *e*'s screen it says which releases and returns *e* wills, which are then submitted as *e*'s release and return requests, as a person would submit them. For a date-based condition it wills by *e*'s date. An escrow agent holding several agreements runs their programs together: a deposit is taken by the first agreement that provides for it, and every other message of its screen, a date advance included, goes to every agreement, so two escrows open at once are each released on their date. The village market's escrow agreement names the depositor, the beneficiary, the lot and the release date; the deposit of that lot by that depositor is accepted, and once the lot is held and *e*'s date has reached the release date, the release is willed:

```
dated_check(Date, T, Dep, Ben, lot(U, D, K), deposited,
            [release(Ben?, U?, D?, K?)],
            dated_escrow(Dep?, Ben?, lot(U?, D?, K?), T?,
                         released, Date?)) :-
    Date? >= T? | true.
dated_check(Date, T, Dep, Ben, Lot, Ph, [],
            dated_escrow(Dep?, Ben?, Lot?, T?, Ph?, Date?)) :-
    otherwise | true.
```

The credit line of Section 5 is the second escrow program of the file, and the one in which *e* keeps state across the transactions, the drawn amount: the lender's deposit of the limit opens the line; the borrower's deposits of a principal lot maturing at the expiry, within the undrawn amount, and of its coupon lot are a draw, on which *e* wills the release of the principal and the coupon bonds to the lender and of that many of the lender's coins to the borrower; the lender's later deposits of their coins, redeemed from them by the borrower against the principal, are repayments, which restore the capacity; and *e*'s date reaching the expiry is the expiry, on which *e* wills the return of the undrawn coins. A deposit the program cannot yet accept stays open and is put to it again at every change of its state, so the order in which the parties reach the escrow agent does not matter.

The village market. The play `village.glp` runs the six villagers and the escrow agent of Section 7.1 in one process: each with its dispatcher, the conversations the play opens between every two parties who transact, its calendar, and a scripted person in place of the interface, which reads the agent's request form and the cards the dispatcher draws, and the screen the agent writes, and submits its requests and answers the cards in the order of its script, each act once the one before it has taken effect on its screen; at the escrow agent the escrow program answers the transfer cards and the person submits the release it wills. The play drives the dates: the escrow agent's reaches day 30 once the deposit is held, Bob's reaches 25 and Frank's 28 by their own calendars. Shown once is what dates add: before Frank's date reaches 28, Diana presents one of his bonds maturing on day 28, and Frank's agent returns it, as it is not mature by his date; once his date has reached 28, she presents one again and redeems it against a frank-coin he holds. The screen messages are logged,

and the final holdings of Section 7.1 are read off the log. The play `credit_line.glp` runs the credit line on the escrow program—establishment, a draw, repayment in two parts, and expiry—and the screens show what Proposition 5.1 states: after the establishment, the draw and each repayment the escrow agent holds the limit less the amount drawn; after the draw the lender holds the principal and the coupon bonds; and at expiry, with nothing drawn, the lender holds the limit in her own coins and no principal bond.