

Grassroots Platforms: Beyond Decentralisation

(Preliminary Draft)

Andrew Lewis-Pye^a and Ehud Shapiro^{a,b}

^aLondon School of Economics, UK, and ^bWeizmann Institute of Science, Israel

Abstract

Decentralisation arose as an antidote to centralised control, with Bitcoin and Ethereum as the prime examples of decentralised global platforms. However, from the perspective of ordinary people, all global platforms—centralised and decentralised alike—appear as monolithic entities, owned, operated and governed by others.

Grassroots platforms are designed to be owned, operated and governed by their participants. They are person-to-person smartphone apps that realise Proudhon’s two-centuries-old concept: A social contract that can have multiple, independent, volitionally-forming instances that may coalesce by mutual consent, possibly into a single global instance.

Here we compare decentralised platforms with grassroots platforms along multiple dimensions: global state; ledger; operation; trust; Sybil resilience; contract; efficiency; scalability; governance; essential agents;¹ and implementation.

We then compare global cryptocurrencies with grassroots currencies. First grassroots coins against Bitcoin and Ether: unit and backing; issuer; ledger; consensus; transaction cost; double spending; liquidity; unit of account; loss of key or device; and implementation. Then, grassroots bonds against decentralised finance: credit; interest; instruments; trust; order and atomicity; time; clearing; and implementation. Finally, sovereign and non-sovereign grassroots currencies against stablecoins: unit; issuer; backing and price; record and parties to a payment; credit creation; interest; monetary policy; runs; and implementation.

In doing so, we illustrate the many ways in which grassroots platforms go beyond decentralisation. We also follow grassroots coins from their social contract down the abstraction cascade, through volition-guarded transactions to the vGLP program every participant runs.

We have developed grassroots theory for a decade but could not muster the resources needed to turn it into competitive software, given the not-for-profit, common-good nature of our project. Now, AI is piercing through this conundrum: our scientific papers serve as the evolving interface between us humans and AI, which codes from them. We have open-source early prototypes of grassroots social networks and currencies and are on course to develop grassroots platforms on par with incumbent global platforms.

2012 ACM Subject Classification Computer systems organization → Peer-to-peer architectures; Theory of computation → Distributed computing models; Human-centered computing → Collaborative and social computing; Applied computing → Digital cash

Keywords and phrases Grassroots Platforms, Decentralisation, Blockchain, Consensus, Sybil Resilience, Trust, Smart Contracts, Social Contracts, Federation, Digital Sovereignty, Cryptocurrencies, Grassroots Currencies, Stablecoins, Decentralised Finance

¹ A minimal set of agents without which communication cannot take place. In centralised global platforms it is the server; in decentralised ones these are the hardwired boot nodes, more than one for resilience; in grassroots, any two remaining agents can communicate.

1 Introduction

Decentralisation arose as an antidote to centralised control, with Bitcoin [22] and Ethereum [44] as the prime examples of decentralised global platforms. However, from the perspective of ordinary people, all global platforms—centralised and decentralised alike—appear as monolithic entities, owned, operated and governed by others.

Grassroots platforms [29, 31] are designed to be owned, operated and governed by their participants. They are person-to-person smartphone apps that realise Proudhon’s two-centuries-old concept: A social contract that (i) can have multiple, independent, volitionally-forming instances that (ii) may coalesce by mutual consent, possibly into a single global instance [35, 21]. In Proudhon’s words [24, Sixth Study], one could (i) “make a contract with all, as ... with some”; and (ii) “each group of citizens ... formed by a like contract ... could thereafter, and always by a similar contract, agree with every and all other groups”.

Global platforms can have only a single instance—one Facebook, one Bitcoin—as two instances would clash over the global resources they use—domain name, port number, or boot nodes—hardwired into their code. A grassroots platform is constituted from its participants’ smartphones and the grassroots app each participant runs, and requires no global resource but the network: no hardwired servers, no global authority, and nobody’s permission to start [29, 31].

Here we compare decentralised platforms with grassroots platforms along multiple dimensions (Section 3, Table 1): (1) global state (replicated vs. distributed [30, 28]), (2) ledger (blockchain [22] vs. attested streams forming a blocklace [2]), (3) operation (miners/validators armed with high-performance servers vs. participants armed with smartphones), (4) trust (trustless vs. trust-based [30, 42]), (5) Sybil resilience (proof of work [22]/stake [17] vs. grassroots social graph [23, 27, 30, 11]), (6) contract (smart contract [9], among accounts, operated by third-parties, requires consensus/SMR vs. social contract [6, 12], among people, operated by the participants, consensus as needed [14, 19, 2, 15]), (7) efficiency (global vs. local transactions [30, 28]), (8) scalability (global state synchronisation vs. network or federation of independent local states [42]), (9) governance (plutocratic, one coin – one vote [5, 4], vs. democratic, one person – one vote [15, 41]), (10) essential agents [32] (finite, > 1 , vs. infinite, all but one), and (11) implementation (deployed at global scale [22, 44] vs. a prototype [40], simulated with stochastic agents [37], not yet deployed). Each dimension states what the decentralised side accomplishes and where the grassroots side falls short, and the converse.

We then compare global cryptocurrencies with grassroots currencies (Section 4, Tables 2A, 2B and 2C). First grassroots coins [28] against Bitcoin [22] and Ether [3]: (1) unit and backing (unbacked, scarcity fixed by the protocol vs. the issuer’s debt, backed by their offerings and redeemed at par [28]), (2) issuer (the protocol vs. any person, natural or legal), (3) ledger (one, replicated by all vs. none shared, the issuer’s signed log, replicated by their state custodians [19, 11]), (4) consensus (on every transaction vs. none), (5) transaction cost (computing or locked capital and a fee vs. operating a smartphone [28]), (6) double spending (excluded by the total order vs. by the issuer’s countersignature [19]), (7) liquidity (a market among strangers vs. mutual credit lines and chains of holdings [28]), (8) unit of account (one vs. one per issuer), (9) loss of key or device (the holdings lost vs. recovered from custodians [11]), and (10) implementation.

Then, grassroots bonds [34] against decentralised finance [28]: (1) credit (lending pools over pooled state vs. a bond, the issuer’s debt due at a date), (2) interest (set by the pool’s protocol vs. the discount at which coins are swapped for a bond), (3) instruments (smart

contracts, automated market makers, flash loans vs. swaps and escrow: loans, sale of debt, forwards, options, collateral, guarantees, insurance, credit lines [34]), (4) trust (code and overcollateralisation vs. the counterparty, the escrow agent and an oracle, each stated), (5) order and atomicity (global vs. bilateral), (6) time (block time vs. each issuer’s attested date [34]), (7) clearing (liquidation by the protocol vs. redemptions once dates advance [34]), and (8) implementation.

Finally, sovereign and non-sovereign grassroots currencies [38] against stablecoins: (1) unit (a fiat-denominated token vs. a debt of one fiat unit, redeemable at par), (2) issuer (one party against reserves vs. any person, the central bank for sovereign coins), (3) backing and price (reserves vs. redemption at par, the arbitrage-free price one fiat unit [38]), (4) record and parties to a payment (one ledger on a blockchain vs. payer, payee and issuer), (5) credit creation (none vs. every person, as inside money [7]), (6) interest (none vs. bonds, sovereign and non-sovereign), (7) monetary policy (none vs. the central bank’s lending, rates and securities in the coins and bonds the public holds [38]), (8) runs (on the issuer’s reserves vs. on one issuer, met from their holdings), and (9) implementation.

In doing so, we illustrate the many ways in which grassroots platforms go beyond decentralisation. We also follow grassroots coins from their social contract [28, 12] down the abstraction cascade [40], through volition-guarded transactions [21] to the vGLP program [40] every participant runs (Section 2).

We have developed grassroots theory for a decade but could not muster the resources needed to turn it into competitive software, given the not-for-profit, common-good nature of our project. Now, AI is piercing through this conundrum: our scientific papers serve as the evolving interface between us humans and AI, which codes from them [39, 36]. We have open-source early prototypes of grassroots social networks and currencies [40, 10, 28, 34] and are on course to develop grassroots platforms on par with incumbent global platforms.

Section 2 presents the methodology and the abstraction cascade, Sections 3 and 4 the two comparisons, and Section 5 concludes.

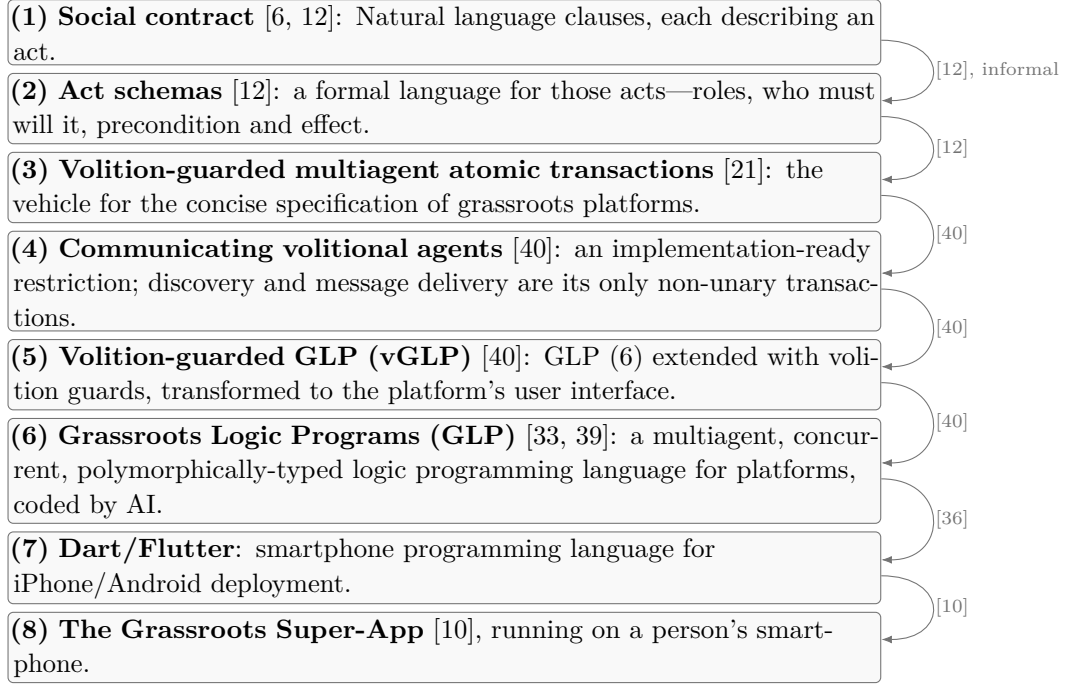
2 From Social Contract to App

2.1 The Human–Science–AI methodology and the abstraction cascade

A grassroots platform is specified by a *grassroots social contract* [6, 12]—an agreement among its participants, executed by their own machines—and is specified formally as *volition-guarded multiagent atomic transactions* [21]: each agent is a person with volitions and their personal machine (smartphone), and a transaction is enabled when the machine precondition holds and the guarding persons are willing.

In our Human–Science–AI methodology, scientific papers serve as an evolving and cooperatively-constructed interface between us humans and AI, recording our shared understanding of the subject matter; AI codes from them, the source of authority for coding, while exposing their gaps and inconsistencies, and cooperates in repairing them. We build grassroots platforms from papers describing them via a cascade of abstractions and implementations—*abstraction cascade programming* [39, 36, 40, 12]—proving each correct and fault-resilient, and coding each with AI. The rule of repair: a gap uncovered in coding is first repaired in the paper, then AI reads the revised paper and repairs the code; never by a workaround in code. The abstraction cascade is shown in Figure 1.

Here we follow one platform, grassroots coins [28], down the cascade of Figure 1: its social contract (1), its volition-guarded multiagent atomic transactions (3), and its vGLP program (5). The act schemas (2) and the communicating volitional agents (4) are in [28, 12, 40].



■ **Figure 1** The abstraction cascade, from a grassroots social contract to a running app. Each arrow represents an implementation with the reference defining it.

2.2 The social contract

The grassroots currencies social contract [28, 12] describes what parties undertake towards one another. 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, with *Alice-coin* referring to a grassroots coin issued by Alice.

1. *Minting*. Alice may decide to issue Alice-coins.
2. *Swap*. Alice and Bob may jointly decide to swap grassroots coins they hold.
3. *Payment*. Alice may decide to pay Bob-coins they hold to Bob.
4. *Redemption*. Alice may decide to redeem a Bob-coin they hold against any grassroots coin held by Bob.
5. *Pricing*. Alice prices their offerings in Alice-coins and accepts them in payment at those prices.

Clauses 1–4 are enforced by the contract’s realisation. Clause 5 is undertaken, and its breach has a social and economic cost, including the risk of a redemption run on Alice-coins; it can also be taken to court [28, 12].

2.3 The volition-guarded transactions

We assume a potentially infinite set of persons Π and consider only finite subsets of it; a grassroots p -coin, denoted c_p , is a unit of debt issued by $p \in \Pi$, and c_p^k is a multiset of k coins c_p . 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, initially empty. 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. We use $\cup$ and $\setminus$ for multiset sum and difference. The four transactions are the following [28].

► **Definition 2.1** (The Transactions of Grassroots Currencies). *For persons $p \neq q \in \Pi$, $u, v, r \in \Pi$ and amounts $k, k' \geq 1$, the transactions of grassroots currencies are the volition-guarded transactions $(c \rightarrow c', Q')$ below, Q' their guards [21], over the participants named, in which c satisfies the precondition stated and $c'_x = c_x$ for every person x not named:*

1. $\text{Mint}_p(k)$, over $\{p\}$, guarded by $\{p\}$: $c'_p = c_p \cup c_p^k$.
2. $\text{Swap}_{p,q}(c_u^k, c_v^{k'})$, over $\{p, q\}$, guarded by $\{p, q\}$: precondition $c_u^k \subseteq c_p$ and $c_v^{k'} \subseteq c_q$; $c'_p = c_p \setminus c_u^k \cup c_v^{k'}$ and $c'_q = c_q \setminus c_v^{k'} \cup c_u^k$.
3. $\text{Pay}_{p,q}(k)$, over $\{p, q\}$, guarded by $\{p\}$: precondition $c_q^k \subseteq c_p$; $c'_p = c_p \setminus c_q^k$ and $c'_q = c_q \cup c_q^k$.
4. $\text{Redeem}_{p,q}(r)$, over $\{p, q\}$, guarded by $\{p\}$: precondition $c_q \in c_p$ and $c_r \in c_q$; $c'_p = c_p \setminus c_q \cup c_r$ and $c'_q = c_q \setminus c_r \cup c_q$.

Mint, pay, and redeem are guarded by the initiator; a swap is guarded by both parties. Grassroots currencies are volitionally grassroots: any group of persons may operate them on its own, and two groups become connected only by a swap willed by a member of each [28, 21].

2.4 The vGLP program

The program every participant runs is the *currency agent*, a vGLP program [40] whose state is the person's holdings. In vGLP a procedure may put a question to the person: a *volitional procedure* is declared `procedure (T)*p(T1, ..., Tn), T its interactive type`, and each of its clauses, `(A)*p(S1, ..., Sn) :- Guard | Body`, carries an interactive term A matched against the person's answer, so a clause that reads the answer reduces only once the person has given it [40]; the interface is derived from the interactive types, and the program declares no widgets. The currency agent is a volitional procedure whose question is the stream of the person's requests, of a type with one alternative per transaction; its clause hands the stream to `serve`, ordinary GLP with one clause per transaction, guarded where the transaction is guarded, and the person's requests, given one at a time, reduce them in turn. The vGLP compiler produces from the program the GLP program every participant runs, together with its interface [40]; the Grassroots Super-App [10] runs it as a mini-app. The source and the compiled program are in the project code repository²; the request type, the agent, the clauses of the four transactions and the swap responder are the following.

```
Peer ::= Key.
Lot ::= lot(Peer, Integer).
Request ::= mint(Integer) ; swap(Peer, Lot, Lot) ; pay(Peer, Integer) ;
           redeem(Peer, Peer).
procedure (Stream(Request))*agent(Key?, Stream(PeerMsg)?, Stream(Conversation)?,
           Stream(OutputEntry)?, Stream(Coin)?, Integer?).

%% Ask the person for their requests, once, and serve them one at a time.
(Reqs)*agent(Id, PeerIn, Convs, Outs, Holdings, Serial) :-
  serve(Reqs?, Id?, PeerIn?, Convs?, Outs?, Holdings?, Serial?).

%% Mint_p(k), guarded by p: c'_p = c_p + cent^k_p.
serve([mint(Amount)|Reqs], Id, PeerIn, Convs, Outs, Holdings, Serial) :-
```

² <https://github.com/EShapiro2/GLP, directory programs/currencies/coins>.

```

ground(Id?), integer(Amount?), Amount? >= 1 |
create_coins(Id?, Amount?, Serial?, New),
Serial1 := Serial? + Amount?,
append_coins(Holdings?, New?, Holdings1),
show(minted(Amount?), Outs?, Outs1),
serve(Reqs?, Id?, PeerIn?, Convs?, Outs1?, Holdings1?, Serial1?).

%% Swap_{p,q}(cent^k_u, cent^{k'}_v), guarded by both: p proposes, giving k
%% u-coins for k' v-coins, and q answers by respond_swap. The precondition at
%% p, cent^k_u in c_p, is the selection; q's part of it is q's.
serve([swap(Friend, lot(GiveCoin, GiveAmount), lot(WantCoin, WantAmount))|Reqs],
      Id, PeerIn, Convs, Outs, Holdings, Serial) :-
  ground(Id?), ground(Friend?), ground(GiveCoin?), integer(GiveAmount?),
  GiveAmount? >= 1, ground(WantCoin?), integer(WantAmount?), WantAmount? >= 1 |
  select_coins(GiveCoin?, GiveAmount?, Holdings?, Status, Give, Rest),
  propose_swap(Status?, Id?, Friend?, lot(GiveCoin?, GiveAmount?),
    lot(WantCoin?, WantAmount?), Give?, Rest?, Reqs?, PeerIn?, Convs?,
    Outs?, Serial?).

%% Pay_{p,q}(k), guarded by p: k q-coins to q, which q accepts (clause 3 of the
%% contract). The precondition cent^k_q in c_p is the selection.
serve([pay(Friend, Amount)|Reqs], Id, PeerIn, Convs, Outs, Holdings, Serial) :-
  ground(Id?), ground(Friend?), integer(Amount?), Amount? >= 1 |
  select_coins(Friend?, Amount?, Holdings?, Status, Give, Rest),
  do_pay(Status?, Id?, Friend?, Amount?, Give?, Rest?, Reqs?, PeerIn?, Convs?,
    Outs?, Serial?).

%% Redeem_{p,q}(r), guarded by p: a q-coin presented to q for an r-coin q holds
%% (clause 4). The precondition cent_q in c_p is the selection; cent_r in c_q
%% is tested at q, which discloses what it holds where it fails.
serve([redeem(Friend, WantCoin)|Reqs],
      Id, PeerIn, Convs, Outs, Holdings, Serial) :-
  ground(Id?), ground(Friend?), ground(WantCoin?) |
  take_coin(Friend?, Holdings?, Found, Rest),
  do_redeem(Found?, Id?, Friend?, WantCoin?, Rest?, Reqs?, PeerIn?, Convs?,
    Outs?, Serial?).

%% The counterparty's guard of a swap: the offer is shown, and the person's
%% yes or no is taken.
Offer ::= offer(Peer, Lot, Lot).
YesNo ::= yes ; no.
Card ::= card(Offer, YesNo?).
SwapResponse ::= accept(Peer, Lot, Lot) ; decline(Peer, Lot).
procedure (Card)*respond_swap(Offer?, SwapResponse).
(card(Offer?, Answer))*respond_swap(Offer, Resp?) :-
  ground(Offer?) | decide(Answer?, Offer?, Resp?).
decide(yes, offer(From, Want, Offered), accept(From?, Want?, Offered?)).
decide(no, offer(From, _, Offered), decline(From?, Offered?)).

```

The counterparty's guard of a swap is the volitional procedure `respond_swap`, whose interactive type is in writer mode: its clause does not wait but writes the card at once, the offer at the position the person sees and a writer at the position `YesNo?` for the person to answer, and `decide` reduces on the answer, `yes` accepting the swap and `no` declining it. The acts arriving from counterparties—a payment, a presentation for redemption, a swap

proposal—are handled by ordinary GLP clauses of the same agent. The program has been implemented and tested on a small scale [28].

3 Comparison with Decentralised Platforms

Table 1 summarises the comparison; we discuss each dimension in turn.

Dimension	Decentralised	Grassroots
1. Global state	replicated	distributed
2. Ledger	blockchain, totally ordered, replicated by all operators [22]	attested streams, one per person, replicated by state custodians [11], interlaced into a blocklace [2]
3. Operation	miners/validators armed with high-performance servers	participants armed with smartphones
4. Trust	trustless	trust-based
5. Sybil resilience	proof of work [22]/stake [17]	grassroots social graph [23, 27, 30, 11]
6. Contract	smart contract, among accounts, operated by third-parties, requires consensus/SMR [9]	social contract [6], among people, operated by the participants, consensus as needed [14, 19, 2, 15]
7. Efficiency	global transactions	local transactions [30, 28]
8. Scalability	global state synchronisation	network or federation of independent local states [42]
9. Governance	plutocratic (one coin – one vote) [5, 4]	democratic (one person – one vote) [15, 41]
10. Essential agents	finite (> 1)	infinite (all but one) [32]
11. Implementation	Bitcoin and Ethereum, deployed at global scale since 2009 and 2015 [22, 44]	implemented and tested on a small scale [40, 10]; simulated with stochastic agents [37]; not deployed

■ **Table 1** Decentralised vs. grassroots platforms

1. Global state. A decentralised platform maintains a single global state—the ledger—replicated by every miner or validator, so every transaction, wherever it originates, is incorporated in the one shared ledger. Every participant therefore has the same view of the whole platform: the balance of every account, the supply of every token, the owner of every name. In a grassroots platform the state is distributed: each participant’s state resides on their own smartphone—their part of the social graph, their feeds, their coins—and a transaction updates only the states of its parties [30, 28]. There is no global view: a fact about the platform as a whole, the number of its participants or the coins in circulation, is held by no one.

2. Ledger. The ledger of a decentralised platform is a blockchain: one totally-ordered chain of blocks, replicated by all operators, extended by consensus [22]. The total order is what lets strangers hold and transfer an unbacked currency: every coin’s history is checked by all, so no coin is spent twice. In a grassroots platform each person has an attested stream of their own acts, signed and non-repudiable; the streams of people who interact are interlaced, each block referring to the blocks of others it has seen, and together form a blocklace [2]—partially-ordered personal blockchains. Attested streams carry the blockchain guarantees of

immutability, unforkability, and non-repudiation, without consensus; a grassroots coin is its issuer's debt, and its issuer is the single writer of its stream [28, 11]. The ledger of grassroots coins, the issuer's log, is replicated by state custodians the issuer designates among their friends, a transaction being final when a supermajority of them hold it, to achieve fault tolerance: a lost phone loses no final transaction, and the log is recovered exactly [11].

3. Operation. A decentralised platform is operated by miners and validators, compensated by fees and block rewards and equipped for the task with high-performance servers. The platform is therefore always on: a participant holds a key and need not be online for their holdings to be safe or for a payment to them to settle. A grassroots platform is operated by its participants: each runs the grassroots app on their own smartphone, and there is no other operator and no fee. A person's state is on their phone and with the friends they designate as custodians [11], so the availability of their part of the platform is that of their phone and their friends' phones.

4. Trust. Decentralised platforms are designed to be trustless: parties who know nothing of each other transact through a protocol designed to make trust unnecessary, and this makes an open global market among strangers possible. Grassroots platforms are trust-based: their social and economic transactions are local, within and among communities of people who know and trust each other [30, 42], and this locality is the source of the security and scalability of the architecture. Two people with no path of acquaintance between them have no transaction available to them until one is introduced to the other [12].

5. Sybil resilience. A decentralised platform resists fake and duplicated identities by pricing participation: Proof of Work [22] and Proof of Stake [17] make influence proportional to computation or holdings, so additional identities add no influence. It does so with no identities at all: anyone may join, anonymously, and the price is that influence follows resources. A grassroots platform resists them through the social graph: identifiers are genuine when obtained from trust built within the graph and, where the stakes are high, from mutual sureties [27, 23]; a person's friends also serve as custodians of their identity and state, so a lost key or a lost phone is recovered from peers rather than from any global resource [11]. A person is identified by a key they chose [11], and joins by being known to, or introduced by, a participant; anonymity towards one's friends is not offered.

6. Contract. A smart contract is a program among accounts, executed by the platform's operators, and requires consensus—state-machine replication—for every step [9]. Its terms are enforced by code against parties who need not know or trust each other, and contracts compose: one may call another. A digital social contract is an agreement among people, executed by the participants themselves [6], with local consensus employed only where the platform needs it [14, 19]: social networking and payments need none; a democratically-governed community has constitutional consensus among its members on the blocklace [2, 16, 20, 15], and grassroots federation and constitutional governance are being implemented without it, by reconfigurable lattice agreement [18], in which concurrent proposals are joined rather than ordered. Its clauses are of three legal types [12]: a breach of an enforced clause is impossible with a correct realisation of the contract; a breach of an attested or undertaken clause can be taken to court, with non-repudiable evidence attributable to the obligor in the case of an attested clause. A smart contract enforces by code; a grassroots social contract enforces by code where it can and by the accountability of known people where it cannot. A grassroots platform is derived from its social contract through the abstraction cascade of Section 2 [12, 21].

7. Efficiency. In a decentralised platform every transaction is global: it is disseminated to, and processed by, all operators, however few parties it concerns, and in return every

transaction has global finality. In a grassroots platform a transaction engages only the smartphones of its parties and costs no more than operating a smartphone [30, 28]; a payment to a person far in the graph is carried along a chain of holdings [28].

8. Scalability. A decentralised platform scales by synchronising its replicated global state faster, and its throughput is bounded by what every operator must process. A grassroots platform scales by a network or federation of independent local states: communities federate by mutual consent, larger ones governed by assemblies drawn by sortition [42], so capacity can grow with membership [19].

9. Governance. Permissionless consensus provides the strata for decentralised autonomous organisations, the prevailing paradigm for participant-governed digital organisations; as participants have verified resources but no trusted identities, this ecosystem is plutocratic—one coin, one vote [15, 5, 4]. Coin voting needs no identities and gives the vote to those exposed to the outcome in proportion to their exposure. Grassroots communities are governed democratically—one person, one vote—by constitutions their members adopt and amend [15, 41], and federate under them [42]; one person, one vote requires genuine identities, which the social graph supplies (dimension 5).

10. Essential agents. We have provided a formal framework for characterising global platforms [32], in which the cardinality of *essential agents*—minimal sets of agents the removal of which makes communication impossible—partitions all global platforms into four classes, centralised (Facebook), decentralised (Bitcoin) [22, 44], federated (Mastodon) [26], and grassroots (Scuttlebutt) [43], and proved that the independent formal definitions of grassroots platforms [35, 21] imply their essential-agents characterisation. In a centralised platform the essential agents are the server; in a decentralised platform, the hardwired boot nodes—finite, more than one for resilience; in a grassroots platform any two remaining agents can communicate, so all agents but one are essential. Within this taxonomy, decentralised and grassroots platforms are distinct classes: grassroots platforms lie beyond decentralisation.

11. Implementation. Bitcoin and Ethereum have been deployed at global scale since 2009 and 2015 [22, 44]. Grassroots platforms—GLP [33, 39, 36], vGLP [40], the Grassroots Super-App [10], and the social graph, social networks and currencies that run on it [30, 28, 34, 38, 11]—have been implemented and tested on a small scale, and simulated with stochastic agents in place of people [37]; they have not been deployed.

4 Global Cryptocurrencies vs. Grassroots Currencies

Global cryptocurrencies such as Bitcoin [22] and Ether [3] are unbacked and bear the cost of global consensus on every transaction [28]: their scarcity is fixed by a protocol, they keep one replicated ledger by consensus among miners or validators, and a transaction costs computing or locked capital and a fee for its place in the next block. Here we compare them to grassroots currencies: coins [28], bonds [34], and sovereign and non-sovereign currencies [38]. Tables 2A, 2B and 2C summarise the comparison—coins against Bitcoin and Ether, bonds against decentralised finance, and sovereign and non-sovereign currencies against stablecoins; we discuss each row in turn.

4.1 Coins

A1. Unit and backing. A global cryptocurrency is one unbacked unit, its scarcity fixed by the protocol, and its value is what the market gives it [22]. A grassroots coin is a claim on its issuer—any person, natural or legal, including people, communities, cooperatives,

Dimension	Bitcoin and Ether	Grassroots coins
A1. Unit and backing	unbacked; scarcity fixed by the protocol [22]	the issuer's debt, backed by their offerings and holdings, redeemed at par [28]
A2. Issuer	the protocol, by mining or staking	any person, natural or legal
A3. Ledger	one, replicated, totally ordered	none shared; the issuer's signed log, replicated by their state custodians for fault tolerance [19, 11]
A4. Consensus	on every transaction	none; final on the issuer's countersignature
A5. Transaction cost	computing or locked capital, and a fee	operating a smartphone
A6. Double spending	excluded by the total order	excluded by the issuer's countersignature [19]
A7. Liquidity	a market among strangers	mutual credit lines; payment along chains of holdings; clearing exactly when no member owes more than they hold [28]
A8. Unit of account	one global unit	one per issuer
A9. Loss of key or device	the holdings are lost	identity and log recovered from custodians [11]
A10. Implementation	in operation at global scale since 2009 and 2015	implemented in GLP from the contract; implemented and tested on a small scale [28]; simulated with stochastic agents [37]; not deployed

■ **Table 2A** Global cryptocurrencies vs. grassroots currencies, A: Bitcoin and Ether vs. grassroots coins

corporations, banks, municipalities and governments—who undertakes to accept their own coins for their offerings at the prices they post; it is a unit of its issuer's debt, which the issuer must redeem at par (1-for-1) against any grassroots coin they hold, so scarcity is controlled by the issuer rather than by a protocol, as with fiat currencies, and the coins are backed by claims on the issuer, as inside money is [28, 7].

A2. Issuer. A global cryptocurrency is issued by its protocol, to miners or validators, and no person owes anything for it. Every grassroots coin has an issuer who owes it and who must redeem it on presentation.

A3. Ledger. A global cryptocurrency keeps one replicated, totally-ordered ledger, and every participant has the same view of every balance. Grassroots currencies keep no shared ledger: the issuer signs every transaction in their own coins and keeps a log of them, replicated for fault tolerance by state custodians the issuer designates among their friends, a transaction being final when a supermajority of them hold it, so that the log survives the loss of the issuer's device and holdership of a coin is proven by the issuer's signatures, without trusting the holders it passed through [19, 11].

A4. Consensus. A global cryptocurrency reaches consensus on every transaction, which is what gives it global finality among strangers. Grassroots currencies use none: a payment needs its payer, its payee and the issuer, proceeds over any network that reaches them, and is final when the issuer has countersigned it [19].

A5. Transaction cost. A transaction of a global cryptocurrency costs computing or locked capital and a fee for its place in the next block, the fee buying priority in a global queue. A grassroots transaction costs no more than operating a smartphone, and bears no fee required

Dimension	Decentralised finance	Grassroots bonds
B1. Credit	lending pools over pooled state, over-collateralised	a bond, the issuer's debt due at a date, redeemed at maturity against a coin [34]
B2. Interest	set by the pool's protocol	the discount at which coins are swapped for a bond
B3. Instruments	smart contracts, automated market makers, flash loans [28]	swaps: loans, sale of debt, forwards; with escrow: payment schedules, options, collateral, guarantees, insurance, credit default swaps, letters of credit, credit lines [34]
B4. Trust	code and overcollateralisation; no counterparty known	the counterparty; the escrow agent; an oracle where an outside event decides; each stated
B5. Order and atomicity	global atomicity and a global order; maximal extractable value [28]	bilateral swaps; no global order
B6. Time	block time	each issuer's attested date; maturity by the issuer's date
B7. Clearing	liquidation by the protocol	the community clears exactly when no member owes more than they hold, once dates advance [34]
B8. Implementation	lending pools and automated market makers in operation at scale	implemented in GLP from the contract; implemented and tested on a small scale [34]; not deployed

■ **Table 2B** Global cryptocurrencies vs. grassroots currencies, B: decentralised finance vs. grassroots bonds

by the architecture, an issuer being free to charge for their services [28].

A6. Double spending. The total order of the ledger excludes double spending among strangers, which no party could otherwise be trusted to exclude. In grassroots currencies the issuer excludes it: every transfer of their coins is countersigned by them and recorded in their log [19].

A7. Liquidity. A global cryptocurrency is liquid in a market among strangers, and anyone who buys it holds it. In grassroots currencies liquidity arises from mutual credit lines, formed by the voluntary swap of coins among persons who know and trust each other; redemption pegs mutually-liquid currencies at par and carries payments along chains of holdings; a community clears its debts into a designated member, in any order and with no coordination, exactly when no other member owes more coins than they hold, the coins of that member a group can obtain by its own redemptions is the maximum flow from the group to that member, and a run on a person is met in full exactly when they hold at least as many coins as they owe [28]. Trade between two people with no path of mutual credit between them waits until one forms.

A8. Unit of account. A global cryptocurrency is one unit of account for all who hold it. Each grassroots issuer prices their offerings in their own coins, so there is one unit per issuer, and redemption at par is what relates them [28].

A9. Loss of key or device. Holdings of a global cryptocurrency whose key is lost are lost. A grassroots person who loses their key or device recovers their identity from their identity custodians and their log, exactly, from their state custodians, and resumes without double spending [11].

A10. Implementation. Bitcoin and Ether have been in operation at global scale since

Dimension	Stablecoins	Sovereign and non-sovereign grassroots currencies
C1. Unit	a fiat-denominated token	a debt of one fiat unit, redeemable at par [38]
C2. Issuer	one party, against reserves it holds	any person (non-sovereign); the central bank (sovereign)
C3. Backing and price	reserves; the peg held by reserves	redemption at par; arbitrage-free price one fiat unit while the issuer redeems; sovereign coins redeemable into fiat [38]
C4. Record and parties to a payment	one ledger on a blockchain; the issuer mints and redeems	payer, payee and issuer; the issuer's log [38]
C5. Credit creation	none	every person, as inside money; banks issue coins in place of deposits
C6. Interest	none to holders	bonds, sovereign and non-sovereign
C7. Monetary policy	none	the central bank lends, absorbs liquidity, sets rates, and buys and sells securities in the coins and bonds the public holds [38]
C8. Runs	a run on the issuer's reserves	a run is on one issuer and is met from their holdings; the central bank issues at its own election [38]
C9. Implementation	in operation at scale	implemented and tested on a small scale [38]; not deployed

■ **Table 2C** Global cryptocurrencies vs. grassroots currencies, C: stablecoins vs. sovereign and non-sovereign grassroots currencies

2009 and 2015. Grassroots coins are implemented in GLP as a program derived from the contract, implemented and tested on a small scale [28], simulated with stochastic agents [37]; they have not been deployed.

4.2 Bonds

B1. Credit. In decentralised finance, credit is lending over pooled state, overcollateralised, by a smart contract [28]. Grassroots coins are redeemed 1-for-1, so mutual credit is at par and the coins are incongruent with interest; a grassroots bond is a unit of its issuer's debt due at a later date, and upon maturity its bearer may redeem it against a coin of the issuer, so liquid coins can be lent against interest-bearing bonds [34].

B2. Interest. In a lending pool the rate is set by the pool's protocol. With grassroots bonds the discount at which coins are swapped for a bond, with its maturity, determines its rate of interest, agreed by the two parties [34].

B3. Instruments. Decentralised finance realises its instruments as smart contracts—lending pools [13], automated market makers [1], flash loans [25]. The voluntary swap of coins and bonds realises 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 [34].

B4. Trust. A smart contract is trusted as code, and a loan is secured by collateral exceeding it, the counterparty unknown. Each grassroots construction states the trust it requires: in the counterparty, in the escrow agent, and, where an outside event decides, in an oracle attesting it [34].

B5. Order and atomicity. Flash loans and maximal extractable value rest on global atomicity and on a global order of transactions [25, 8]. A grassroots swap is bilateral and atomic between its two parties, and there is no global order to extract value from.

B6. Time. Decentralised finance keeps time by the chain, in blocks. Each grassroots issuer keeps their own date, attested: a bond matures by the date its issuer warrants, and a party whose date lags behind the calendar breaches the contract, witnessed by the record [34].

B7. Clearing. A lending pool clears a position by liquidating its collateral. A grassroots community clears its debts by redemptions exactly when no member owes more than they hold, of all maturities, and does so without coordination once dates advance and persons act on their rights [34].

B8. Implementation. The lending pools and automated market makers of decentralised finance are in operation at scale. Grassroots bonds are implemented in GLP as a program derived from the contract, implemented and tested on a small scale [34]; they have not been deployed.

4.3 Sovereign and non-sovereign currencies

C1. Unit. A stablecoin is a token denominated in a fiat currency. A non-sovereign grassroots coin is a digital debt of one unit of fiat currency, redeemable at par, that can be issued by any person, natural or legal; a sovereign grassroots coin is such a debt issued by the central bank [38].

C2. Issuer. A stablecoin is issued by one party, against reserves it holds. Every sovereign and non-sovereign grassroots coin has an issuer who owes it, and there is no separate protocol for the central bank: the only difference between its coins and other persons' is that theirs carry some risk of default, whereas a central bank can issue by fiat as many units as needed to redeem its coins [38].

C3. Backing and price. A stablecoin's peg is held by the reserves of its issuer. The arbitrage-free price of a non-sovereign coin whose issuer redeems it on demand is one unit of the fiat currency, so, while redemptions are met, the currencies of the persons and of the sovereign are one currency, with the fiat currency as its unit of account; non-sovereign coins are redeemable at par into sovereign coins along a chain of liquidity from their issuer to the central bank, and then by it into fiat [38].

C4. Record and parties to a payment. A stablecoin keeps one ledger on a blockchain, and its issuer mints and redeems it against reserves. A payment in a grassroots currency needs its payer, its payee and the issuer, is confidential to them, and is recorded in the issuer's log; the central bank sees the payments in its own coins and no others [38].

C5. Credit creation. A stablecoin creates no credit: it is issued against reserves and redeemed into them. Under a denomination every person creates credit, as inside money secured by social collateral, and a bank issues its coins in place of deposits [38].

C6. Interest. A stablecoin pays its holders no interest. Interest in grassroots currencies is in bonds, sovereign and non-sovereign: a bank's bond takes the place of its time deposit, and the coins bear none, as cash does not [38].

C7. Monetary policy. A stablecoin has no monetary policy: its issuer mints against reserves and redeems into them. With sovereign and non-sovereign coins and bonds the central bank can lend, absorb liquidity, set its rates and buy and sell securities in the coins and bonds the public holds, choosing the counterparties and terms of its credit operations; its lending rate and its bond rate bracket the rates its counterparties face [38].

C8. Runs. A run on a stablecoin is a run on its issuer’s reserves, and the peg holds while the reserves cover the redemptions. In grassroots currencies a run is on one issuer and is met from their holdings; the central bank can always meet presentations of its own sovereign coins by issuing fiat, so there is no open-ended run into central bank money, and it issues at its own election [38].

C9. Implementation. Stablecoins are in operation at scale. Sovereign and non-sovereign grassroots coins and bonds have been implemented and tested on a small scale, and have yet to be similarly measured [38]; they have not been deployed.

5 Conclusion

Decentralised platforms achieve what they set out to achieve: a global platform with no single party in control, in which strangers transact with global finality, with no identities and no trust. Grassroots platforms achieve something else: a platform with no third party, owned, operated and governed by the people who use it, in which transactions are among people who know each other, with no global state or global ledger, and with local consensus used only where the platform needs it.

References

- 1 Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. 2021. URL: <https://uniswap.org/whitepaper-v3.pdf>.
- 2 Paulo Sérgio Almeida and Ehud Shapiro. The blocklace: A byzantine-repelling and universal conflict-free replicated data type. *arXiv preprint arXiv:2402.08068*, 2024. URL: <https://arxiv.org/abs/2402.08068>.
- 3 Vitalik Buterin. A next-generation smart contract and decentralized application platform. *white paper*, 3(37), 2014. URL: <https://ethereum.org/whitepaper/>.
- 4 Vitalik Buterin. Governance, part 2: Plutocracy is still bad, 2018. Available at <https://vitalik.eth.limo/general/2018/03/28/plutocracy.html>. URL: <https://vitalik.eth.limo/general/2018/03/28/plutocracy.html>.
- 5 Vitalik Buterin. Moving beyond coin voting governance. Blog post, August 2021. URL: <https://vitalik.eth.limo/general/2021/08/16/voting3.html>.
- 6 Luca Cardelli, Liav Orgad, Gal Shahaf, Ehud Shapiro, and Nimrod Talmon. Digital social contracts: A foundation for an egalitarian and just digital society. In *CEUR Proceedings of the First International Forum on Digital and Democracy*, volume 2781, pages 51–60. CEUR-WS, 2020. URL: <https://ceur-ws.org/Vol-2781/paper5.pdf>.
- 7 Ricardo de O Cavalcanti and Neil Wallace. Inside and outside money as alternative media of exchange. *Journal of Money, Credit and Banking*, pages 443–457, 1999.
- 8 Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927. IEEE, 2020. doi:10.1109/SP40000.2020.00040.
- 9 Primavera De Filippi, Chris Wray, and Giovanni Sileno. Smart contracts. *Internet Policy Review*, 10(2), 2021. doi:10.14763/2021.2.1549.
- 10 Ohad Eitan, Idit Keidar, and Ehud Shapiro. The grassroots super-app: A secure operating system for grassroots mini-apps. *arXiv:XXX:XXXXX*, 2026. *arXiv:XXX:XXXXX*.
- 11 Ohad Eitan, Idit Keidar, and Ehud Shapiro. 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. *arXiv:2607.02304*. URL: <https://arxiv.org/abs/2607.02304>.

- 12 James Golike, Andy Lewis-Pye, and Ehud Shapiro. Formalising and implementing grassroots social contracts: From legal text to grassroots platforms (full version). *arXiv:2609.06231*, 2026.
- 13 Lewis Gudgeon, Sam M. Werner, Daniel Perez, and William J. Knottenbelt. 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)*, pages 92–112. ACM, 2020. doi:10.1145/3419614.3423254.
- 14 Rachid Guerraoui, Petr Kuznetsov, Matteo Monti, Matej Pavlovič, and Dragos-Adrian Seredinschi. The consensus number of a cryptocurrency. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 307–316, 2019. doi:10.1145/3293611.3331589.
- 15 Idit Keidar, Andrew Lewis-Pye, and Ehud Shapiro. Constitutional consensus. *arXiv preprint arXiv:2505.19216*, 2025. URL: <https://arxiv.org/abs/2505.19216>.
- 16 Idit Keidar, Oded Naor, and Ehud Shapiro. Cordial miners: A family of simple and efficient consensus protocols for every eventuality. In *37th International Symposium on Distributed Computing (DISC 2023)*. LIPICS, 2023. arXiv preprint arXiv:2205.09174. URL: <https://arxiv.org/abs/2205.09174>.
- 17 Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual international cryptology conference*, pages 357–388. Springer, 2017. doi:10.1007/978-3-319-63688-7_12.
- 18 Petr Kuznetsov, Thibault Rieutord, and Sara Tucci-Piergiovanni. Reconfigurable lattice agreement and applications, 2020. URL: <https://arxiv.org/abs/1910.09264>, arXiv:1910.09264.
- 19 Andrew Lewis-Pye, Oded Naor, and Ehud Shapiro. Grassroots flash: A payment system for grassroots cryptocurrencies. *arXiv preprint arXiv:2309.13191*, 2023. URL: <https://arxiv.org/abs/2309.13191>.
- 20 Andrew Lewis-Pye and Ehud Shapiro. Morpheus consensus: Excelling on trails and autobahns. *Proc. 29th Conference on Principles of Distributed Systems, OPODIS 2025*. *arXiv preprint arXiv:2502.08465*, 2025. doi:10.4230/LIPIcs.OPODIS.2025.35.
- 21 Andy Lewis-Pye and Ehud Shapiro. Volition-guarded multiagent atomic transactions: Describing people and their machines. *arXiv preprint arXiv:2604.25596*, 2026. URL: <https://arxiv.org/abs/2604.25596>.
- 22 Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. URL: <https://bitcoin.org/bitcoin.pdf>.
- 23 Ouri Poupko, Gal Shahaf, Ehud Shapiro, and Nimrod Talmon. Building a sybil-resilient digital community utilizing trust-graph connectivity. *IEEE/ACM transactions on networking*, 29(5):2215–2227, 2021. doi:10.1109/tnet.2021.3084303.
- 24 Pierre-Joseph Proudhon and John Beverley Robinson. *General idea of the revolution in the nineteenth century*. Courier Corporation, 2004. Original work published 1851.
- 25 Kaihua Qin, Liyi Zhou, Benjamin Livshits, and Arthur Gervais. Attacking the DeFi ecosystem with flash loans for fun and profit. In *Financial Cryptography and Data Security – 25th International Conference, FC 2021*, pages 3–32. Springer, 2021. doi:10.1007/978-3-662-64322-8_1.
- 26 Eugen Rochko. Mastodon: A decentralized social network based on open web protocols. <https://joinmastodon.org>, 2016–. URL: <https://joinmastodon.org/>.
- 27 Gal Shahaf, Ehud Shapiro, and Nimrod Talmon. Genuine personal identifiers and mutual sureties for sybil-resilient community growth. In *International Conference on Social Informatics*, pages 320–332, EU, 2020. Springer, Springer. doi:10.1007/978-3-030-60975-7_24.
- 28 Ehud Shapiro. Grassroots currencies: Turning mutual trust into liquidity. *arXiv preprint arXiv:2202.05619*, 2022. URL: <https://arxiv.org/abs/2202.05619>.
- 29 Ehud Shapiro. Grassroots distributed systems: Concept, examples, implementation and applications (brief announcement). In *37th International Symposium on Distributed Computing*

- (DISC 2023). (Extended version: *arXiv:2301.04391*), pages 47:1–47:7, Italy, 2023. LIPICS. URL: <https://arxiv.org/abs/2301.04391>.
- 30 Ehud Shapiro. Grassroots social networking: Serverless, permissionless protocols for twitter/linkedin/whatsapp. In *OASIS '23*. Association for Computing Machinery, 2023. doi:10.1145/3599696.3612898.
 - 31 Ehud Shapiro. A grassroots architecture to supplant global digital platforms by a global digital democracy. *arXiv:2404.13468, Proceedings of DAWO'24*, 2024. URL: <https://arxiv.org/abs/2404.13468>.
 - 32 Ehud Shapiro. Characterising global platforms: Centralised, decentralised, federated, and grassroots. *arXiv preprint arXiv:2511.03286*, 2025. URL: <https://arxiv.org/abs/2511.03286>.
 - 33 Ehud Shapiro. GIp: 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. URL: <https://arxiv.org/abs/2510.15747>.
 - 34 Ehud Shapiro. Grassroots bonds: Financing by the people, for the people. *arXiv:2603.13671*, 2026. URL: <https://arxiv.org/abs/2603.13671>, *arXiv:2603.13671*.
 - 35 Ehud Shapiro. Grassroots platforms with atomic transactions: Social graphs, cryptocurrencies, and democratic federations. In *Proceedings of the 27th International Conference on Distributed Computing and Networking*, pages 71–81, 2026. *arXiv preprint arXiv:2502.11299*. doi:10.1145/3772290.3772309.
 - 36 Ehud Shapiro. Implementing grassroots logic programs with multiagent transition systems and ai (full version). *arXiv:2602.06934. Summary to appear in Proc. of LOPSTR+PPDP'26*, 2026. URL: <https://arxiv.org/abs/2602.06934>.
 - 37 Ehud Shapiro. sGLP: Specifying stochastic agents that simulate people. *arXiv:XXX:XXXX*, 2026. *arXiv:XXX:XXXX*.
 - 38 Ehud Shapiro. Sovereign Grassroots Currencies: A CBDC Architecture for Credit and Monetary Policy. *arXiv:2609.27727*, 2026. URL: <https://arxiv.org/abs/2609.27727>, *arXiv:2609.27727*.
 - 39 Ehud Shapiro. Types for grassroots logic programs (full version). *arXiv preprint arXiv:2601.17957*, 2026. URL: <https://arxiv.org/abs/2601.17957>.
 - 40 Ehud Shapiro. Volition elicitation: Operational semantics for people and their machines. *arXiv:2607.14138*, 2026. URL: <https://arxiv.org/abs/2607.14138>, *arXiv:2607.14138*.
 - 41 Ehud Shapiro and Nimrod Talmon. Constitutional governance in metric spaces. *Proceedings of the International Conference on Algorithmic Decision Theory (ADT 2026)*, to appear, 2026. *arXiv:2605.13362*. URL: <https://arxiv.org/abs/2605.13362>.
 - 42 Nimrod Talmon and Ehud Shapiro. Grassroots federation: Fair democratic governance at scale. *arXiv preprint arXiv:2505.02208; also Proc. of AAMAS'26*, 2025. URL: <https://arxiv.org/abs/2505.02208>.
 - 43 Dominic Tarr, Erick Lavoie, Aljoscha Meyer, and Christian Tschudin. Secure scuttlebutt: An identity-centric protocol for subjective and decentralized applications. In *Proceedings of the 6th ACM conference on information-centric networking*, pages 1–11, 2019. doi:10.1145/3357150.3357396.
 - 44 Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014. URL: <https://ethereum.github.io/yellowpaper/paper.pdf>.