UCL

UCL Centre for Blockchain Technologies

Discussion Paper Series

Q1 2020

CBT

Foreword

It is a real pleasure to start this discussion paper series with very interesting contributions ranging from smart contracts to stablecoins. We have received a large number of submissions, and we include in this first series five papers, of which three published in this bulletin and two published online. The purpose of the series is to showcase the activities on blockchain studies from our CBTs associates and researchers around the world. These are not journal publications in a strict sense. They are instead a selection of publications and preprints that we chose to endorse and promote.

In this series, we reprint the work of Christian Sillaber and co-authors which has been already published in a known international journal, and we report it to support its diffusion. The paper lies the foundations for the engineering design and deployment of smart contracts for DLT applications. Another work is the paper by Silvia Bartolucci and co-authors, which is a pre-print still in its submission phase to a journal. It reports a very clean and insightful modelling of the Lightning Network reporting on the emergence of a transition between a sustainable and unsustainable network controlled by transaction volumes. The pre-print paper by Paolo Giudici and co-authors is looking into the very important topic of stablecoins proposing how to construct stable baskets better using techniques form optimal control problems and financial mathematics. The paper by Alexi Anania and Ken Hodler is providing an insightful contribution to current anonymity issues in the bitcoin transaction network and the consequence of planned future technological innovations in this domain. Finally, Conny Weber provides a collection of case studies from the EU-funded European Crowdfunding Network AISBL that we re-publish online to provide further visibility to these useful contributions.

This is the first of a large number of future discussion paper series. Indeed, we intend to continue with this discussion paper series making it an important quarterly appointment. We welcome submission of recent works, either pre-prints or published papers, that we will gladly help to reach more significant impact and influence. The CBT is a very heterogeneous institution with hundreds of researchers scattered across departments and universities. This discussion paper series will reflect such a splendid diversity promoting works in all areas related to blockchain technologies. In this way, we aim to encourage the exchange of ideas across disciplines and also to create common languages and mutual understanding to move forward and at a higher level.

Enjoy your reading

Tomaso Aste
UCL CBT Scientific Director & Chairman of the Editorial Board
April 2020

Acknowledgement
The editorial board wishes to thank Anna Gorbatcheva for her high-quality work and strong perseverance, without which this discussion paper series would have never been possible.

Discussion Paper Series Contents

The following are published in this edition:

#1

A percolation model for the emergence of the Bitcoin Lightning Network

Silvia Bartolucci, Fabio Caccioli & Pierpaolo Vivo

#2

Laying the foundation for smart contract development: an integrated engineering process model

Christian Sillaber, Bernhard Waltl, Horst Treiblmaier, Ulrich Gallersdörfer & Michael Felderer

#3

Libra or Librae? Basket based stablecoins to mitigate foreign exchange volatility spillovers

Paolo Giudici, Thomas Leach & Paolo Pagnottoni

*The following are **published online only:***

#4

The impact of Tatroot Schnorr on address clustering analysis of Bitcoin transactions

Alexi Anania & Ken Hodler

#5

Exploring DLT and Blockchain for alternative finance

Conny Weber

CBT

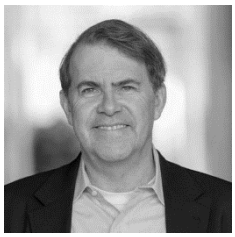
Editorial Board

Tomaso Aste

Chairman of the Editorial Board
Professor, Complexity Science, UCL

Quinn DuPont

Research Associate, University of Washington



Daniel Heller

Honorary Professor, UCL

Seongbae Lim

Professor, St. Mary's University

Ralf Wandmacher

Professor, Accadis University

Andy Yee

Public Policy Director at Visa

A percolation model for the emergence of the Bitcoin Lightning Network

Silvia Bartolucci
Department of Finance, Imperial College London Business School
South Kensington SW7 2AZ London (UK)
Centre for Blockchain Technologies, University College London

E-mail: s.bartolucci@imperial.ac.uk

Fabio Caccioli
Department of Computer Science, University College London
66-72 Gower Street WC1E 6EA London (UK)
Centre for Blockchain Technologies, University College London

E-mail: f.caccioli@ucl.ac.uk

Pierpaolo Vivo
Department of Mathematics, King's College London
Strand WC2R 2LS London (UK)

E-mail: pierpaolo.vivo@kcl.ac.uk

Abstract. The Lightning Network is a so-called second-layer technology built on top of the Bitcoin blockchain to provide "off-chain" fast payment channels between users, which means that not all transactions are settled and stored on the main blockchain. In this paper, we model the emergence of the Lightning Network as a (bond) percolation process and we explore how the distributional properties of the volume and size of transactions per user may impact its feasibility. The agents are all able to reciprocally transfer Bitcoins using the main blockchain and also – if economically convenient – to open a channel on the Lightning Network and transact "off chain". We base our approach on fitness-dependent network models: as in real life, a Lightning channel is opened with a probability that depends on the "fitness" of the concurring nodes, which in turn depends on wealth and volume of transactions. The emergence of a connected component is studied numerically and analytically as a function of the parameters, and the phase transition separating regions in the phase space where the Lightning Network is sustainable or not is elucidated. We characterize the phase diagram determining the minimal volume of transactions that would make the Lightning Network sustainable for a given level of fees or, alternatively, the maximal cost the Lightning ecosystem may impose for a given average volume of transactions. The model includes parameters that could be in principle estimated from publicly available data once the evolution of the Lighting Network will have reached a stationary operable state, and is fairly robust against different choices of the distributions of parameters and fitness kernels.

Keywords: *Blockchain, Lightning Network, Payment Networks, Percolation, Fitness Models*

1. Introduction

Bitcoin, the pioneering cryptocurrency, has brought about an unprecedented revolution in the payment industry [1]. Despite its traction and success over the last ten years, the original blockchain – the technological infrastructure underlying Bitcoin – suffers from some limitations that may hinder the future growth and adoption of the cryptocurrency. One of the major issue is the *scalability* of the system: the current number of transactions validated via this platform is between 3 and 7 transactions per second, compared for instance to thousands of transactions handled by the Visa circuit [2]. The lack of scalability is mainly caused by constraints on throughput of transactions, with the block size fixed at 1MB, and by the high latency – with a new block created on average only every ten minutes. Those limitations are imposed to safeguard the security of the platform against malicious attacks and are difficult to relax without major changes in the protocol.

The main solutions proposed to address the scalability issue include (i) changes to the main protocol (consensus algorithm, parameters) and (ii) *sidechains*∗ and second-layer solutions. Notable examples of type-(i) solutions include new consensus protocols, which would allow a faster issuance of new blocks among other new features [4]. The Lightning Network (LN), instead, is a so-called second-layer technology built on top of the Bitcoin blockchain to provide "off-chain" fast payment channels between users [5]. By off-chain we mean that not all transactions are settled and stored on the main blockchain. In a nutshell, the idea of a Lightning channel is the following: two parties lock the same amount of money as collateral and open a channel for a certain period of time. During this time, they can then exchange money back and forth through the channel, and only the netted transaction will be eventually validated and stored on the main blockchain. If one party is malicious and does not correctly update the balance, the other can keep the collateral posted by the malicious party, as a form of insurance. Any two users can open a channel and all other participants can use one or more existing channels to route transactions off-chain upon payment of a fee to channel "owners". The scalability problem could be solved if a sufficient number of channels were opened, implying that the Lightning Network spans across the whole pool of users of the main blockchain.

The Lightning Network topology is, indeed, relevant to understand the resilience of the system to attacks or random failures and its robustness. Measures of the network structure based on empirical data – such as degree distribution, assortativity, shortest paths length – provide an indication of the efficiency of payments' routing and features of the system (i.e. average number of channel per user, clusters and communities, etc.) [6]. Experiments on random or targeted nodes removal from the network give information on

∗Sidechains are blockchains "connected" to the main Bitcoin blockchain such that Bitcoins can be transferred bidirectionally between the main and side blockchain [3]. At the same time, sidechains are completely separate ecosystems whose technical features or issues would not be shared with the main blockchain.

the system resilience by monitoring when the original network is broken into multiple isolated clusters [7–9]. In the Lightning Network case, it has been shown that some types of targeted attacks – aimed at consuming, for instance, the channels' liquidity of specific nodes – may yield severe consequences for the resilience of the network in terms of average payment flow and reachability [10].

The topology of the network is in turn driven by users' economic incentives to relay transactions "off-chain". Moreover, as the Lightning fees are set by channels' owners, an important question is how high such fees should be set in order to guarantee profits but providing at the same time the right incentives for Bitcoin users to participate in the Lightning Network. In a recent work on simple network topologies (i.e. bidirectional channels and star graphs), the authors have estimated the demand for transactions on the main Bitcoin blockchain compared to the LN, the level of LN fees that would cover maintenance costs of the channel and their implication for the overall network security [11]. Indeed, transacting on the Lightning Network might impact the security of the main blockchain network by inducing a decrease in the amount of fees collected by the miners for the validation of blockchain transactions.

Fees on the main blockchain are used as incentives to miners (i.e. nodes capable of validating transactions and generating new blocks) to contribute to the security of the platform∗. Normally, users "compete" to set up the minimal fees that would ensure that their transaction be validated within a given timeframe, as miners try to maximise the total amount of fees per block. A strand of the literature has been investigating the Bitcoin fee set-up mechanisms, the miners' incentives and their potential correlation with risks of attacks and manipulation of the transaction history. In [14] the authors use a game-theoretic model to investigate the factors influencing the value of Bitcoin fees, while in [15] they also examine the interplay between fees and security of the platform, theoretically showing that the current fee model may not be sustainable on the long run. Alternative fee mechanisms have also been proposed, for instance based on auction models [14], and compared with the existing one to highlight weaknesses and possible improvements.

The Bitcoin ecosystem has been already extensively investigated using approaches based on complex networks. The transactions network has been studied to understand latency issues and propagation mechanisms in peer-to-peer systems [16] and inefficiencies of the process of permanent inclusion of the transactions on the blockchain [17] . Global and local structural properties of the users' network in Bitcoin have also provided insights on booms and bust events [18] and Bitcoin price dynamics [19]. Moreover, data on users' behaviour and spending patterns have been used to understand the global state of the crypto-economy [20] and the drivers of the growth of the network [21]. More generally, our paper taps into the growing literature on quantitative investigations of the cryptocurrencies landscape, including models of pricing and adoption of tokens [22–25], analysis of the market structure [26–32] and price prediction based on sentiment and

∗The blockchain security is associated with the platform's decentralisation, hence to the miners' total computing power [12, 13].

social interactions [33–41].

In this paper, we investigate under which conditions in terms of blockchain and Lightning fees, average wealth and volume of transactions per users, a Lightning Network that spans a sizeable fraction of Bitcoin users – thus solving the scalability problem – emerges. We model the emergence of the Lightning Network as a (bond) percolation process on a graph, exploring how different conditions may impact its feasibility [42]. In particular, we consider fitness-dependent network models [43–46] where the probability of creating a new edge depends on intrinsic node features collectively denoted *node fitness*. In the LN case, the node fitness will be defined in terms of the node wealth and activity (i.e. volume of transactions). The viability of the Lightning Network will be characterized in terms of the presence (or not) of a giant connected cluster of nodes: a non-fragmented network would, indeed, guarantee a smooth relay of payments and information between users and will incentivize off-chain transactions. Our model depends on parameters that can be all obtained – or at least estimated – from publicly available data, and is fairly robust against different choices of distributions of parameters and fitness kernels.

The paper is organized as follows. In Section 2 we provide a quick overview of the main Bitcoin blockchain and the main ideas behind LN. In Section 3 we describe our model and provide the relevant theory, which is then applied to two specific wealth distributions (uniform and exponential) in the subsections 3.1 and 3.2, respectively. In Section 4, we discuss the results of numerical simulations, and we provide some conclusions and outlook in Section 5. The Appendices are devoted to technical aspects of percolation theory on networks and are included to make the paper self-contained.

2. The Bitcoin blockchain and Lightning Network

In this section, we summarise the main features of the Bitcoin main blockchain and Lightning Network payment layer. The Bitcoin blockchain is a distributed, shared ledger that immutably records transactions among peers in the network [1, 12]. Transactions are bundled in blocks and chained together via cryptographic primitives to ensure that any change at any point in the transaction history would invalidate the full record. Transactions are validated for correctness, temporarily stored in memory pools and then arranged in the blocks data structure by *miners*: multiple miners compete using computational power to validate the next block of the chain – and therefore earn the associated reward for the service and transactions' fees–according to the Proof-of-Work consensus algorithm. Depending on the usage of the network and due to limitation in block size, waiting times can peak around 30 minutes (while the typical range is around $6-8$ minutes), while blockchain fees per transaction exhibit a broad range of variability, from a few cents to $40-50$ USD$*$.

The idea behind the creation of the Lightning Network [5] is, therefore, to devise a network for frequent and fast micro-transactions that can be performed at low

$*$Data taken from `https://www.blockchain.com/charts` .

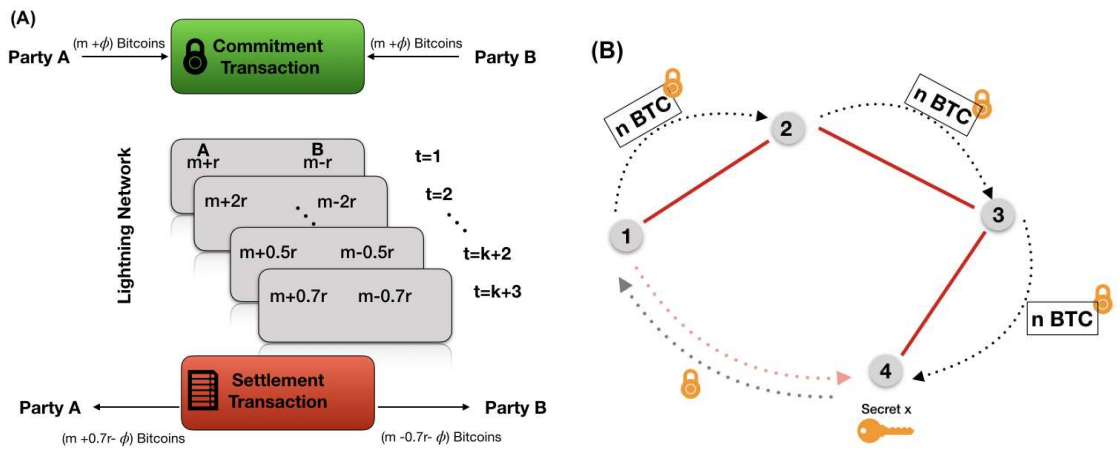


Figure 1. Panel **(A)**: Scheme of a payment channel between party A and B, including opening and closing transactions settled on the main blockchain, and intermediate transfers handled off-chain on the Lightning Network. Initially, the two parties lock m bitcoins each, plus the fee ϕ that would cover broadcasting on the main blockchain. At each time step t, users exchange an amount $\alpha(t)r$ and update their balances accordingly, by sending each other redeemable receipts without committing them on the main blockchain. Only when the two parties agree that the channel is no longer needed, they settle the net balance of funds on their original Bitcoin addresses. Panel **(B)**: Scheme of payments routing on a Lightning Network: even if party 1 and 4 are not directly connected via an existing LN channel, they can route their payments through other parties (upon payment of a fee) by choosing a suitable cryptographic lock for the Bitcoins.

transactions fees. The basic components of the Lightning Network are *payment channels* (schematically shown in Fig. 1, panel A), enabling trustless transfers between users. In the typical payment channel implementation, a theoretically unlimited amount of payments can be made, with only two transactions broadcast on the blockchain. In addition to a reduction of number of blockchain transactions and associated costs, payment channels also offer the advantage of speed and, importantly, the ability of users to recover their funds if one of the parties is malicious.

A channel is established between two parties by locking an initial amount of funds, for instance m Bitcoins for each user, on the main blockchain, which represent the maximum amount of Bitcoins that can be transferred over the channel. Funds are locked on so-called 2-of-2 multisignature addresses [12], which can be unlocked upon providing the signature of both interested parties. For instance, user A wishes to send r Bitcoins to user B: she signs a transaction, sends it to B, who will sign it and send it back to A. Only the first transaction is recorded on the main blockchain. At each time step in the lifetime of the channels, the users keep sending back and forth signed transactions that can be at any point consensually broadcast on the main blockchain to close the channel and redeem the net amount of funds. To prevent fraudulent behavior, for instance user B not acknowledging the receipt of a payment from A, a refund option is always included in any exchange. The refund option can be unilaterally unlocked

and submitted on the blockchain after a certain amount of time $t^\star$ has elapsed from the moment the channel was first established. Every new refund option is indeed signed by both parties, signalling therefore that they are in agreement with the terms of the refund, which may be exercised unilaterally at a later time. In the worst-case scenario, one party would simply submit the original refund transaction created contextually with the opening of the channel.

Payments can be relayed via the Lightning Network also if two parties are not directly connected via a Lightning channel, if there exists a path indirectly linking them via existing channels owned by third parties. Exploiting an existing path to route the payments may often prove more convenient as the two interested parties need not open a new channel, therefore saving the associated costs in blockchain fees. Channels' owners are indeed owed "routing fees" to allow payments through their channel, but at the moment those fees are very competitive (~ 4 orders of magnitude less than the Bitcoin blockchain$*$). In Fig. 1, Panel B we show an example of an indirect routing path between user 1 and 4. One of the biggest issues of the Lightning Network is the limitation in liquidity. Payments are made by effectively having intermediaries forwarding collateral across multiple channels: this means that if party 1 is transferring ℓ Bitcoins to party 4, each relaying channel needs to have at least ℓ Bitcoins available in the direction of the payment.

To prevent dishonest behavior in the transfer from party 1 to 4 via party 2 and 3 (see Fig. 1, Panel B), Party 1 will lock the Bitcoins with a secret key known only by the receiver: when party 4 receives the Bitcoins from party 3, the secret is revealed and every player can collect their coins and fees [12].

3. Model Setup

In this section, we model the emergence of the Lightning Network as a (bond) percolation process. We consider N agents, who are all able to reciprocally transfer Bitcoins using the main blockchain and – if economically convenient – to open a channel on the Lightning Network and transact "off chain". We introduce the node capacity (or *wealth*) w_i of node i, a random variable extracted from a pdf $\Pi(w)$, which is proportional to the maximum amount of Bitcoins that node i can lock in a Lightning channel it partakes in. We will consider two explicit examples for the wealth distribution (uniform and exponential) in the following, with qualitatively similar results.

Two nodes are more likely to open a Lightning channel if they expect to submit a large number of transactions over a given period of time. Therefore, we introduce for each node i a quantity ℓ_i that represents its "activity" in terms of average number of transactions node i sends through each channel in the network. The average number of transaction is also a random variable extracted from the discrete distribution $\hat{\Pi}(\ell)$ over non-negative integers. We also include the costs associated with transacting over one of the two networks (main blockchain only or blockchain and Lightning). These costs

$*$Data taken from `https://1ml.com/statistics` and `https://bitcoinfees.info` .

can be fixed per transaction (*base fee*) or can be calculated as a percentage of the value transferred (*fee rate*).

- c, *Lightning channel maintenance/usage base fee:*
  Using the LN channel provided by an operator or other users to transfer coins carries an associated LN fee. Opening a channel has also maintenance costs (fee setup, market and nodes monitoring, connections) and costs related to locking Bitcoins and providing liquidity in the channel.

- ϕ, *main blockchain rate fee:*
  We assume that a fraction ϕ of the value transferred in each transaction needs to be paid by the sender to have it included in blocks and validated by miners.

The probability of opening a new LN channel between two nodes p_{ij}^{LN} can be modelled as a function of (i) the costs associated with opening the LN channel (if the costs are significantly smaller than using the Bitcoin blockchain, there is an incentive for the users towards opening the channel), (ii) users' affinity (the more likely are users to transact over a period of time τ, the higher the benefits of opening a channel), (iii) the wealth of the nodes (nodes wishing to open a LN channel have to lock a minimal amount of Bitcoins on the main blockchain as collateral).

The growth of the Lightning Network can be modelled as a bond percolation process on a set of N nodes representing Bitcoin users. The edges then represent new Lightning channels being opened. In particular, we construct the bond percolation model considering fitness-dependent networks [43–46]. In fitness models, the network topology is determined by (i) an *attachment kernel* $f(x, y)$, describing the probability that a node with fitness x will connect to a node with fitness y, and (ii) the distribution of fitness $\rho(x)$ across nodes.

The network we consider has a fixed number of nodes N – corresponding to all Bitcoin users that may decide to switch to the LN – and is *sparse*, i.e. the number M of edges is $M \ll N^2$. If we consider node i and j having fitness x_i and x_j respectively, a LN channel, i.e. an edge between them, is added with probability

$$p_{ij}^{\mathrm{LN}} = f(x_i, x_j) \sim \mathcal{O}(1/N) \ . \tag{1}$$

The resulting network is undirected if $f(x, y) = f(y, x)$, which is a sensible requirement: indeed, opening a LN channel between two nodes will require a "symmetric" commitment from both nodes to lock Bitcoins on the main blockchain. In our model, we will consider bond percolation only: number and "state" of the nodes (e.g. occupied/unoccupied or infected/susceptible) will not change.

In the context of the LN network, we define the fitness x_i of node i as the simplest increasing function of both capacity and volume of transactions, i.e.

$$x_i = w_i(\ell_i + 1) \ , \tag{2}$$

where w_i represents the wealth of the node, and $\ell_i + 1 \geq 1$ its "activity" in terms of number of transactions expected to be sent through the channel∗. As in [44], we consider the fitness to be defined in the interval $[0, \infty]$.

Given this definition of the node fitness, the fitness distribution can be calculated from the wealth and activity distributions, $\Pi(w)$ and $\hat{\Pi}(\ell)$ respectively, as

$$\rho(x) = \sum_{\ell \geq 0} \hat{\Pi}(\ell) \int_0^\infty \mathrm{d}w \; \Pi(w) \delta \left(x - w(\ell + 1) \right) \; . \tag{3}$$

From the kernel $f(x, y)$, we can derive the probability that a node with fitness x increases its degree by one as [43]

$$\lambda(x, N) = \frac{1}{N} \frac{\int_0^\infty \mathrm{d}y f(x, y) \rho(y)}{\kappa} \; , \tag{4}$$

which is obtained as the ratio between the number of links connected to a node with fitness x, divided by the total number of links.

We also define $\lambda(x) = N\lambda(x, N)$ and rewrite it as

$$\lambda(x) = \frac{1}{\kappa} \int_0^\infty \mathrm{d}y \; f(x, y) \rho(y) \; , \tag{5}$$

where $N\kappa \sim \mathcal{O}(1)$ is the average degree of the network with N nodes, with

$$\kappa = \int_0^\infty \int_0^\infty \mathrm{d}x \mathrm{d}y \; \rho(x) \rho(y) f(x, y) \; . \tag{6}$$

Note that $\lambda(x)$ clearly satisfies the following normalization condition

$$\int_0^\infty \lambda(x) \rho(x) \mathrm{d}x = 1 \; . \tag{7}$$

The degree distribution $P(k)$ for large N is given by

$$P(k) = \int_0^\infty \mathrm{d}x \; \rho(x) \frac{\mathrm{e}^{-N\kappa\lambda(x)} [N\kappa\lambda(x)]^k}{k!} \; , \tag{8}$$

whose average degree is $\langle k \rangle = N\kappa$, as shown in detail in Appendix A.

In the following, we will assume that the activity distribution is Poisson with average $\bar{n}$, $\hat{\Pi}(\ell) = \exp(-\bar{n}) \bar{n}^\ell / \ell!$, and that the connectivity kernel models the effects of blockchain and LN fees as follows

$$f(x, y) = \frac{\mu}{N} \Theta(x\phi - c) \Theta(y\phi - c) \; , \tag{9}$$

where $\Theta(z)$ is the Heaviside step function†. The interpretation of this kernel is as follows: agent i expects to interact with μ other agents, which we assume for simplicity are chosen randomly.

∗We assume that all nodes are potentially active, $x_i > 0$ for all i.

†We have checked that smoothing $f(x, y)$, e.g. by multiplying the thetas by $xy/(1 + xy)$ or $1 - \exp(-(x + y))$, has a minimal impact on the results.

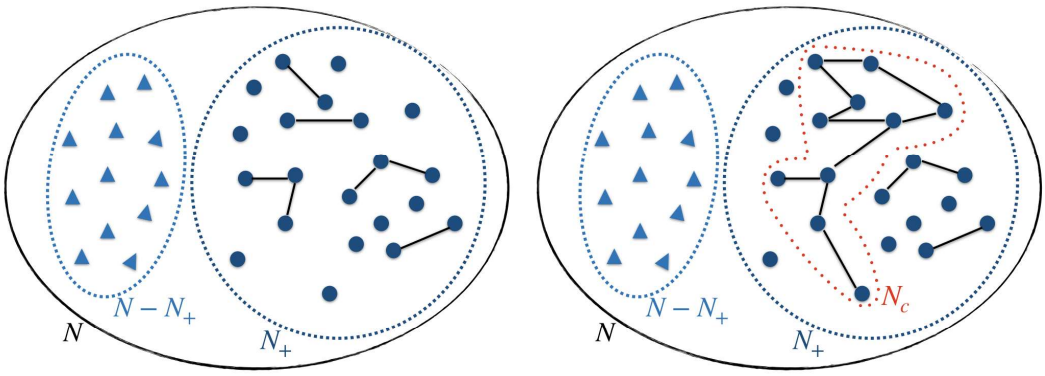


Figure 2. Schematic representation of the emergence of the connected component among fit nodes, below (left) and above (right) the percolation threshold. Light blue triangles represent nodes whose fitness is smaller than c/ϕ, while dark blue circles represent "high-fitness" nodes with $x > c/\phi$. N is the total number of Bitcoin users, N_+ is the fraction of "high-fitness nodes" (see Eq. (10)) and N_C is the fraction of high-fitness nodes belonging to the giant connected component.

The probability of interacting with a given agent j is equal to μ/N for all j. Agent i wishes to transfer an amount $w_i(\ell_i+1)$ (corresponding to ℓ_i+1 transactions of size w_i) to each of them, and is willing to open a Lightning channel if the cost of maintaining it (c) is lower than the cost of transferring the money through the blockchain ($w_i(\ell_i+1)\phi$). The same considerations apply to its counterpart j.

We define

$$N_+ = \sum_{i=1}^{N} \Theta(x_i\phi - c) \tag{10}$$

the number of nodes with "high" fitness, for whom it is economically viable to engage in a LN. Note that N_+ is a random variable, which depends on the realization of the fitnesses. We define the average fraction $f_+ = \langle N_+ \rangle / N$.

The network constructed via the sequential deposition of links (as described above) may undergo a *percolation transition* [42, 43, 47, 48] as a function of f_+, such that – beyond a critical value of f_+ – a giant connected component of N_C nodes emerges, whose fractional average size $S = \langle N_C \rangle / N$ remains finite as $N \to \infty$. We stress that in any fixed instance $N_C \leq N_+$, since some high-fitness nodes may still not engage in LN (see Fig. 2). In our language, this connected component represents the set of nodes that not only do exploit Lightning channels to exchange wealth off-chain between nearest neighbors, but may also transfer wealth to any "distant node", routing the transaction via connected paths. It is therefore of paramount importance to understand under which conditions on the average wealth, average volume of transactions, and routing fees, this transition may happen, and what finite fraction of nodes will it involve.

With the choice of the kernel in (9), the topology of the resulting Lightning Network of N_+ nodes is that of an Erdős-Rényi (E-R) graph with average degree equal to $\mu f_+ \sim \mathcal{O}(1)$. At odds with the standard model of E-R graphs, in our case the size

of the graph N_+ is itself a random variable, which depends on the parameters of the model. In fact, once f_+ has been obtained, the model can be mapped onto a site percolation problem on random networks, where each node is occupied with probability f_+, and the emergence of a viable Lightning Network corresponds to the emergence of a giant component of occupied nodes.

The relevant percolation theory is summarized in Appendix B to make the paper self-contained.

3.1. Uniform wealth distribution

We now take $\Pi(w)$ – the pdf of wealth across nodes – as uniform in the interval $[0, w_0]$. Hence, we have

$$\rho^{(u)}(x) = \sum_{\ell \geq 0} \frac{\mathrm{e}^{-\bar{n}} \bar{n}^\ell}{\ell!} \int_0^{w_0} \frac{\mathrm{d}w}{w_0} \delta\left(x - w(\ell+1)\right) \ , \tag{11}$$

where the superscript $^{(u)}$ refers to uniform wealth distribution. Simplifying we obtain

$$\rho^{(u)}(x) = \frac{1}{w_0} \sum_{\ell \geq \lceil \frac{x}{w_0} - 1 \rceil} \frac{\mathrm{e}^{-\bar{n}} \bar{n}^\ell}{\ell!(\ell+1)} = \frac{1}{w_0 \bar{n}} \left(1 - \frac{\Gamma\left(\lceil \frac{x}{w_0} \rceil, \bar{n} \right)}{\Gamma\left(\lceil \frac{x}{w_0} \rceil \right)} \right) \ , \tag{12}$$

where $\Gamma(a,x) = \int_x^\infty t^{a-1} \mathrm{e}^{-t} \mathrm{d}t$, and $\lceil z \rceil$ denotes the smallest integer larger than z. In this case, it follows from (5) and (9) that

$$\lambda^{(u)}(x) = \frac{1}{f_+^{(u)}} \Theta(x\phi - c) \ , \tag{13}$$

where $f_+^{(u)}$ is the average fraction of high-fitness nodes and is given by

$$f_+^{(u)} = \int_{c/\phi}^\infty \mathrm{d}x \ \rho^{(u)}(x) = \frac{1}{w_0} \sum_{\ell \geq 0} \frac{\mathrm{e}^{-\bar{n}} \bar{n}^\ell}{\ell!} \int_0^{w_0} \mathrm{d}w \ \Theta(w(\ell+1) - c/\phi) \ , \tag{14}$$

which requires $w > c/[(\ell+1)\phi]$, in turn constraining $c/[(\ell+1)\phi] \leq w_0 \to \ell \geq \lceil \frac{c}{w_0 \phi} - 1 \rceil$ (which may also be negative). Therefore

$$f_+^{(u)} = \frac{1}{w_0} \sum_{\ell = \max\left(0, \lceil \frac{c}{w_0 \phi} - 1 \rceil\right)} \frac{\mathrm{e}^{-\bar{n}} \bar{n}^\ell}{\ell!} \int_{\frac{c}{(\ell+1)\phi}}^{w_0} \mathrm{d}w = \Psi(0) - \frac{c}{\phi w_0} \Psi(-1) \ , \tag{15}$$

where

$$\Psi(t) = \sum_{\ell = \max\left(0, \lceil \frac{c}{w_0 \phi} - 1 \rceil\right)} \frac{\mathrm{e}^{-\bar{n}} \bar{n}^\ell}{\ell!} (\ell+1)^t \ . \tag{16}$$

The evaluation of $P^{(u)}(k)$ from (8) requires some care, as $\lambda^{(u)}(x)$ is zero if $x < c/\phi$. Splitting the integration region, we get

$$P^{(u)}(k) = \delta_{k,0} \int_0^{c/\phi} \mathrm{d}x\ \rho^{(u)}(x) + \frac{\mathrm{e}^{-\mu f_+^{(u)}} (\mu f_+^{(u)})^k}{k!} \int_{c/\phi}^{\infty} \mathrm{d}x\ \rho^{(u)}(x)$$
$$= \left(1 - f_+^{(u)}\right) \delta_{k,0} + f_+^{(u)} \frac{\mathrm{e}^{-\mu f_+^{(u)}} (\mu f_+^{(u)})^k}{k!}\ . \tag{17}$$

The interpretation of (17) is quite neat: on average, the network contains a fraction $1 - f_+$ of isolated (low-fitness) nodes, and a fraction f_+ of high-fitness nodes that may (or may not) partake in the LN, establishing sparse random connections with an average of μf_+ other high-fitness nodes. Computing now the generating function (B.1)

$$G_0^{(u)}(s) = 1 - f_+^{(u)} + f_+^{(u)} \exp\left(\mu f_+^{(u)}(s-1)\right)\ , \tag{18}$$

it follows from Eq. (B.2) that

$$G_1^{(u)}(s) = \frac{G_0^{(u)\prime}(s)}{G_0^{(u)\prime}(1)} = \exp\left(\mu f_+^{(u)}(s-1)\right)\ . \tag{19}$$

The general theory (see Appendix B, in particular Eq. (B.20)) then implies that the equation determining $0 < \xi^\star \leq 1$ is

$$\xi^\star = \exp\left[\mu f_+^{(u)}(\xi^\star - 1)\right]\ . \tag{20}$$

The average size of the giant component thus reads from Eq. (B.19)

$$S^{(u)} = (1 - \xi^\star) f_+^{(u)}\ , \tag{21}$$

and the condition in Eq. (B.17) for the giant component to appear (see Fig. 4) is

$$\mu f_+^{(u)} > 1\ . \tag{22}$$

The interpretation of this condition is fairly obvious: the giant connected component can only arise if "fit" nodes open on average more than one channel with other fit nodes.

3.2. Exponential wealth distribution

We now take $\Pi(w)$ to be the exponential pdf with mean w_0. The fitness distribution now becomes

$$\rho^{(e)}(x) = \sum_{\ell \geq 0} \frac{\mathrm{e}^{-\overline{n}} \overline{n}^\ell}{\ell!} \int_0^\infty \frac{\mathrm{d}w}{w_0} \mathrm{e}^{-\frac{w}{w_0}} \delta\left(x - w(\ell+1)\right) = \frac{1}{w_0} \sum_{\ell \geq 0} \frac{\mathrm{e}^{-\overline{n}} \overline{n}^\ell}{\ell!(\ell+1)} \mathrm{e}^{-\frac{x}{w_0(\ell+1)}}\ .$$

As in the uniform wealth case

$$\lambda^{(e)}(x) = \frac{1}{f_+^{(e)}} \Theta(x\phi - c)\ , \tag{23}$$

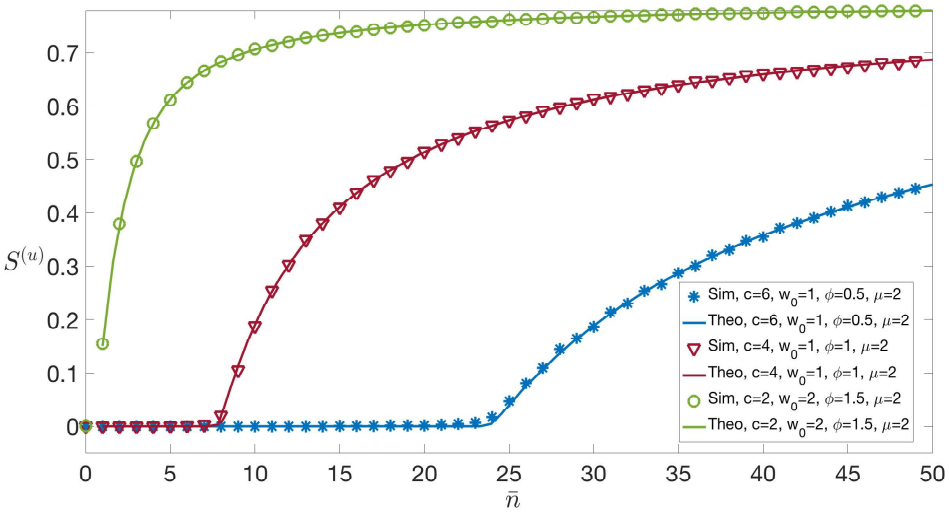


Figure 3. Size of the giant component as a function of $\bar{n}$ for different combinations of the parameters ϕ, c, w_0, μ. Simulations with kernel $f(x, y)$ in Eq. (9) and uniform wealth distribution in the interval $[0, w_0]$. Numerical results have been obtained for a network of $N = 5 \cdot 10^4$ nodes averaging over 5 independent instances. Fixing a certain fraction S of nodes – connected via LN – and increasing the ratio c/ϕ between the Lightning Network and main blockchain fees, we observe that a larger average volume of LN transactions is required to make the system financially sustainable. The transition between $S = 0$ and $S > 0$ happens at a value of $\bar{n}$ that we denote $\bar{n}^\star$.

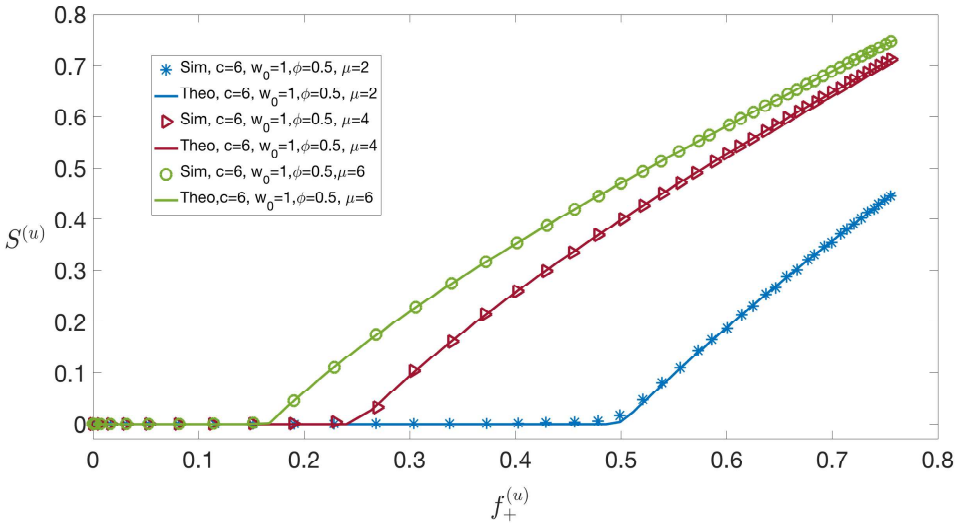


Figure 4. Size of the giant component as a function of $f_+^{(u)}$ (the fraction of high-fitness nodes) varying $\mu = 2, 4, 6$ and fixing $w_0 = 1, \phi = 0.5, c = 6$. Simulations with kernel $f(x, y)$ in Eq. (9) and uniform wealth distribution with parameter w_0. Numerical results have been obtained for a network of $N = 5 \cdot 10^4$ nodes averaging over 5 independent instances. The transition between $S = 0$ and $S > 0$ happens at $1/\mu$ as correctly predicted by the condition in Eq. (22).

where this time $f_+^{(e)}$ reads

$$f_+^{(e)} = \int_{c/\phi}^{\infty} \mathrm{d}x\ \rho^{(e)}(x) = \frac{1}{w_0} \sum_{\ell \geq 0} \frac{\mathrm{e}^{-\bar{n}} \bar{n}^{\ell}}{\ell!} \int_0^{\infty} \mathrm{d}w\ \mathrm{e}^{-w/w_0} \Theta(w(\ell+1) - c/\phi)\ , \tag{24}$$

which requires $\ell \geq \lceil \frac{c}{\phi w} - 1 \rceil$. Therefore,

$$f_+^{(e)} = 1 - \frac{1}{w_0} \int_0^{\infty} \mathrm{d}w\ \mathrm{e}^{-w/w_0} \frac{\Gamma\left(\lfloor \frac{c}{\phi w} \rfloor, \bar{n}\right)}{\Gamma\left(\lfloor \frac{c}{\phi w} \rfloor\right)}\ , \tag{25}$$

where $\lceil z \rceil$ denotes the largest integer smaller than z. As in the uniform-wealth case

$$P^{(e)}(k) = \left(1 - f_+^{(e)}\right) \delta_{k,0} + f_+^{(e)} \frac{\mathrm{e}^{-\mu f_+^{(e)}} (\mu f_+^{(e)})^k}{k!}\ . \tag{26}$$

Now, consider the solution $0 < \eta^\star \leq 1$ of

$$\eta^\star = \exp\left[\mu f_+^{(e)} (\eta^\star - 1)\right]\ . \tag{27}$$

Then, the average size of the giant component reads

$$S^{(e)} = (1 - \eta^\star) f_+^{(e)}\ , \tag{28}$$

and the condition for the giant component to appear reads $\mu f_+^{(e)} > 1$ (see Fig. 5).

4. Numerical Simulations and Results

We present numerical simulations on networks of $N = 5 \cdot 10^4$ nodes, generated by sequential deposition of links with probability as in Eq. (1), using the kernel in Eq. (9). In Fig. 3, where we use a uniform distribution of wealth with average $w_0 = 1$, we plot the average size $S^{(u)}$ of the connected component as a function of $\bar{n}$, the average volume of transactions to be deployed on the LN, for varying values of the fees ratio c/ϕ. Fixing a certain average fraction $S^{(u)}$ of nodes – which can reach each other via a connected LN path – and increasing the ratio c/ϕ between the LN and main-blockchain fees, we observe that a larger average volume of LN transactions is required to make the off-chain network financially sustainable. Increasing the average wealth w_0 would push the curves upwards: as more liquidity becomes available across nodes, more and more players may get involved in the LN for the same level of routing fees. In Fig. 5, we observe qualitatively the same phenomenon, this time for exponential distribution of wealth.

To find the size of the largest connected component, we use a breadth-first search algorithm [49]: starting from a source node s, we label it as belonging to cluster #1. We then explore its neighborhood and assign all nodes reachable from s to cluster #1 as well. The algorithm proceeds recursively until either the whole network has been labelled, or

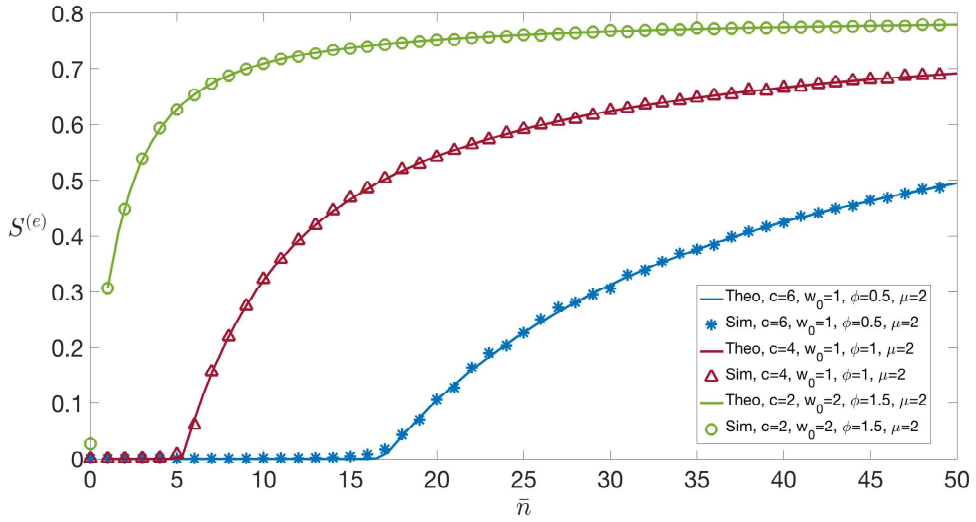


Figure 5. Size of the giant component as a function of $\bar{n}$ for different combinations of the parameters w_0, ϕ, c. Simulations with kernel $f(x, y)$ in Eq. (9) and exponential wealth distribution with average w_0. Numerical results have been obtained for a network of $N = 5 \cdot 10^4$ nodes averaging over 5 independent instances. Increasing w_0 for similar values of the ratio c/ϕ makes the nodes wealthier on average, and therefore more likely to engage in a LN: as a consequence, the average size of the LN-connected component increases for a given value of average volume of transactions. The transition between $S = 0$ and $S > 0$ happens at a value of $\bar{n}$ that we denote $\bar{n}^\star$.

no unlabelled nodes can be further reached. In the latter case, we select another random source among the unlabelled nodes, assign it the label #2, and restart the procedure to find another cluster. At the end, all disjoint clusters have been identified, and their size recorded. In our plots, we monitor the size of the largest cluster.

In Fig. 6, we plot the phase diagram in the (ϕ, c) plane for the uniform wealth distribution model (very similar results are obtained for the exponential wealth distribution, not shown). The colors from blue to yellow represent (from low to high) the values of $\bar{n}^\star$, the minimal average volume of transactions that need to be deployed to make a LN financially viable for a given value of LN and main-blockchain fees, c and ϕ, respectively. We observe a transition between two regimes: one (region **1**) where the LN fees are sufficiently low (compared to main-blockchain fees) that *any* volume of transactions (however low, $\bar{n}^\star = 0$) can be transferred off-chain and still be financially viable, the other (region **2**) where the LN fees are sufficiently high that agents may be discouraged from opening channels and transferring wealth off-chain *unless* there is a minimal volume of transactions to be deployed ($\bar{n}^\star > 0$). The higher the ratio c/ϕ, the less convenient it is to open LN channels for a fixed value of transactional activity.

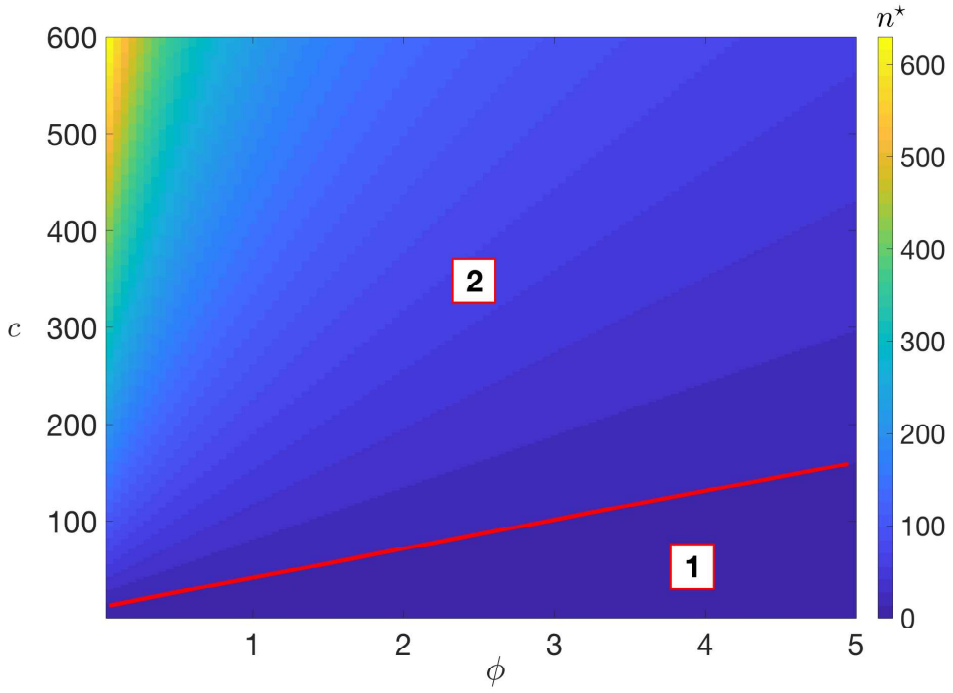


Figure 6. Phase diagram in the (ϕ, c) plane for uniform wealth distribution in the interval $[0, w_0 = 1]$, $\mu = 20$ and kernel $f(x, y)$ (Eq. (9)). The colors from blue to yellow represent (from low to high) the values of the minimal average volume of transactions $\bar{n}^\star$ that need to be deployed to make a LN financially viable. In region (**1**) (low ratio between LN and main-blockchain fees), $\bar{n}^\star = 0$, i.e. for any average volume of transactions, a LN connected component is sustainable. In region (**2**) (larger ratio between Lightning and main blockchain fees), $\bar{n}^\star > 0$, implying that a LN is not financially viable unless the volume of transactions is sufficiently high.

5. Conclusions

In summary, we have presented a simple fitness-based network model for the emergence of a connected set of nodes exchanging wealth off-chain, whose average fractional size S remains finite as $N \to \infty$. The percolation transition resulting from sequential deployment of edges is studied numerically and analytically, as a function of a limited set of parameters that we predict will be in principle possible to infer from data: w_0 (related to the average wealth jointly owned by the agents), $\bar{n}$ (the average volume of transactions that can be handled off-chain), c, ϕ (the fees associated with off-chain and on-chain transactions) and μ (the average number of channel per node). As a matter of fact, different platforms are currently being offered – but only at a test stage – where users can experience the Lighting Network services in a simulated environment. Already at this early stage in the development of a fully operational payment system, some useful data can be gathered: for instance, the platform '1ML'∗ currently aggregates information about $\sim$ 10000 nodes sharing $\sim$ 30000 channels, with an average capacity

∗https://1ml.com/statistics

per node of $\sim$ 1000 USD, and a base fee per transaction of around 0.000072 USD. Similarly, for the Bitcoin blockchain we can gather an estimate of ~ 0.52 USD as base fee per transaction, as well as more accurate figures about number of transaction per day and average transaction values $*$.

The function $f(x, y)$ in Eq. (9) has been selected as the simplest but nontrivial attachment kernel that favors a link (i.e. the opening of a Lightning channel) whenever the fitness of both concurring nodes (in terms of exchangeable wealth and volume of predicted activity) exceeds a financially viable threshold. We have checked that "smoothing" the 0/1- kernel in Eq. (9) has negligible effects on the results, while making the analytical treatment unnecessarily more complicated. Similarly, the model is fairly insensitive to the details of the full probability distribution of wealth that is used, while being flexible enough to generate a desired degree distribution $P(k)$ via a different choice of the attachment kernel $f(x, y)$ [50]. A percolation transition separates a phase where no sustainable LN can be formed, from a phase where the fees being charged, the total available wealth and the average activity conspire to make off-chain payments a viable option for a finite fraction of the network in the limit $N \to \infty$. The transition is elucidated analytically and numerically, with excellent agreement.

In the future, this investigation can be extended in the following ways:

- A mechanism for dynamical update of wealth as more channels are opened and funds are locked may be introduced, to investigate the liquidity constraints of the network in more detail. Dynamically generated wealth inequalities and concentration may be detected by means of centrality measures.
- The resilience of the network can be studied under different types of attacks and compared with available empirical results [6, 10].
- Different choices of the kernel $f(x, y)$ (e.g. non-factorized) may be also explored. This could lead to networks with heterogeneous (heavy-tailed) degree distribution, which seems to be in line with recent empirical studies [6].

Once the development of the Lighting Network technology and implementation will have reached maturity, it will be possible to gather data to calibrate our model, which can serve as a driver for policy changes and as guidance for incentive mechanisms design.

Appendix A. Degree distribution $P(k)$

Following [43], the probability $p_{M,N}(k|x)$ that a node in a large undirected graph with N nodes and $M \ll N^2$ edges has degree k given that its fitness is x follows the recursion

$$p_{M+1,N}(k+1|x) = p_{M,N}(k+1|x)[1 - 2\lambda(x, N)] + 2p_{M,N}(k|x)\lambda(x, N) \ . \tag{A.1}$$

The interpretation is easy: the probability of having a node with degree $k+1$ after an edge addition $(M+1)$ is equal to the probability that the node already had degree $k+1$

$*$Data available at `https://www.blockchain.com/en/charts`.

times the probability that the new edge does not have any of its two terminal points attached to it ($[1-2\lambda(x,N)]$), plus the probability that the node had degree k times the probability that the new edge has one of its two terminal points connected to it ($2\lambda(x,N)$).

Multiplying both sides of Eq. (A.1) by s^k and summing over $k \geq 0$, we obtain the following equation for $F_{M,N}(s|x) = \sum_{k\geq 0} s^k p_{M,N}(k|x)$

$$F_{M+1,N}(s|x) - F_{M,N}(s|x) = 2F_{M,N}(s|x)(s-1)\lambda(x,N) + F_{M+1,N}(0|x) - F_{M,N}(0|x)(1-2\lambda(x,N)) \ . \tag{A.2}$$

For large M, Eq. (A.2) can be rewritten as an ordinary differential equation of the form $\frac{\partial F}{\partial M} = 2(s-1)\lambda(x,N)F + (\frac{\partial F}{\partial M} + 2\lambda(x,N)F)|_{s=0}$, with solution

$$F_{M,N}(s|x) = \mathrm{e}^{\frac{2M}{N}\lambda(x)(s-1)} \ , \tag{A.3}$$

where we recall that we defined $N\lambda(x) = \lambda(x,N)$ and we used the initial condition $F_{0,N}(s|x) = 1$ that follows from the fact that in a network with zero edges, $p_{0,N}(k|x) = \delta_{k,0}$.

Taylor-expanding around $s = 0$ and noting that $2M/N = N\kappa$, we obtain the degree distribution conditional on the fitness of the node x

$$p_{M,N}(k|x) = \frac{\mathrm{e}^{-N\kappa\lambda(x)}\left(N\kappa\lambda(x)\right)^k}{k!} \ . \tag{A.4}$$

Marginalizing with respect to x, we eventually obtain the probability that a node as degree k (irrespective of its fitness) as

$$P(k) = \int_0^\infty \mathrm{d}x \ p_{M,N}(k|x)\rho(x) = \int_0^\infty \mathrm{d}x \ \frac{\mathrm{e}^{-N\kappa\lambda(x)}\left(N\kappa\lambda(x)\right)^k}{k!}\rho(x) \ , \tag{A.5}$$

which correctly implies

$$\langle k \rangle = \sum_{k\geq 0} kP(k) = N\kappa \ , \tag{A.6}$$

using (7).

Appendix B. Giant component

The generating function of the probability that a node has degree k is denoted by

$$G_0(s) = \sum_{k\geq 0} P(k)s^k \ . \tag{B.1}$$

We introduce the generating function $G_1(s)$ of the (normalized) probability that by following a randomly chosen edge we reach a node with degree k

$$G_1(s) = \frac{\sum_{k\geq 1} kP(k)s^{k-1}}{\sum_{k\geq 0} kP(k)} = \frac{G_0'(s)}{G_0'(1)} = \frac{G_0'(s)}{N\kappa} \ . \tag{B.2}$$

This is because the node we reach by following a randomly chosen edge has degree distribution $kP(k)/N\kappa$ rather than just $P(k)$ – since a randomly chosen edge is more likely to lead to a node of higher degree.

We also define the generating function of the number of nodes that can be reached following a randomly chosen edge and that belong to a connected component of size t with size distribution$*$ $\psi(t)$

$$H_1(x) = \sum_{t\geq 1} \psi(t) x^t \ . \tag{B.4}$$

Moreover, we indicate with $H_0(x)$ the generating function of the probability that a randomly chosen node belongs to a connected component of size t

$$H_0(x) = \sum_{t\geq 1} \pi(t) x^t \ . \tag{B.5}$$

Assuming that the typical component sizes are finite and that the chances of a component containing a closed loop of edges are negligible for sufficiently large N, the distribution of components generated by $H_1(x)$ can be obtained as follows [43, 47, 48]. Let us denote by $\zeta(t|k)$ the probability that a node with degree k belongs to a component of size t

$$\zeta(t|k) = \sum_{t_1\geq 1} \cdots \sum_{t_k \geq 1} \delta\left(t-1, \sum_{m=1}^{k} t_m\right) \prod_{m=1}^{k} \psi(t_m) \ , \tag{B.6}$$

where $\delta(a,b)$ is the Kronecker delta. Indeed, the sum of the sizes of the components that can be reached by following the k edges departing from the node must be equal to $t-1$, and each of these sizes is drawn from the distribution $\psi(t)$.

Marginalizing over the degree distribution, we obtain the probability $\pi(t)$ that a randomly chosen node belongs to a component of size t as

$$\pi(t) = \sum_{k\geq 0} P(k)\zeta(t|k) \ . \tag{B.7}$$

Computing $H_0(x)$ from (B.5)

$$\begin{aligned} H_0(x) &= \sum_{t\geq 1} \pi(t) x^t = \sum_{t\geq 1} x^t \sum_{k\geq 0} P(k)\zeta(t|k) \\ &= \sum_{k\geq 0} P(k) \sum_{t\geq 1} x^t \sum_{t_1\geq 1} \cdots \sum_{t_k\geq 1} \delta\left(t-1, \sum_{m=1}^{k} t_m\right) \prod_{m=1}^{k} \psi(t_m) \\ &= x \sum_{k\geq 0} P(k) \sum_{t_1\geq 1} \cdots \sum_{t_k\geq 1} x^{\sum_m t_m} \prod_{m=1}^{k} \psi(t_m) = x \sum_{k\geq 0} P(k) \left[\sum_{t\geq 1} \psi(t) x^t\right]^k = x G_0(H_1(x)) \ , \end{aligned} \tag{B.8}$$

$*$More precisely,

$$H_1(x) = \lim_{N\to\infty} \sum_{t=1}^{N} \psi(t,N) x^t \ , \tag{B.3}$$

where $\psi(t,N)$ is the probability that – in a network with N nodes – by following a randomly chosen link, we reach a component of size $t \leq N$, and similarly for $H_0(x)$ in (B.5).

where we have used (B.1) and (B.4). The calculation for $H_1(x)$ is analogous, with the replacement $P(k) \to \frac{kP(k)}{\sum_{k'} k'P(k')}$. Summarizing, the two equations to be solved together are

$$H_0(x) = xG_0(H_1(x)) \ , \tag{B.9}$$

$$H_1(x) = xG_1(H_1(x)) \ . \tag{B.10}$$

The average size $\langle t \rangle$ of the connected components is given from (B.5) as

$$\langle t \rangle = \sum_{t \geq 1} t\pi(t) = H_0'(1) \ . \tag{B.11}$$

$H_0'(1)$ can be obtained from (B.9) as

$$H_0'(1) = G_0(H_1(1)) + G_0'(H_1(1))H_1'(1) \ . \tag{B.12}$$

Note that from (B.4) it follows that $H_1(1) = 1$ (by normalization of $\Psi(t)$). Similarly, from (B.1), we have that $G_0(1) = 1$ (by normalization of $P(k)$). Eq. (B.12) can be therefore simplified as follows

$$H_0'(1) = 1 + G_0'(1)H_1'(1) \ . \tag{B.13}$$

We can then compute $H_1'(1)$ using (B.10)

$$H_1'(1) = G_1(H_1(1)) + G_1'(H_1(1))H_1'(1) \ . \tag{B.14}$$

As before, we can simplify it using the fact that $H_1(1) = 1$ and that $G_1(1) = \left.\frac{\sum_k kP(k)s^{k-1}}{\sum_k kP(k)}\right|_{s=1} = 1$ (see(B.2)), obtaining:

$$H_1'(1) = 1 + G_1'(1)H_1'(1) \Rightarrow H_1'(1) = \frac{1}{1 - G_1'(1)} \ . \tag{B.15}$$

Substituting (B.15) in (B.13) yields

$$\langle t \rangle = H_0'(1) = 1 + \frac{G_0'(1)}{1 - G_1'(1)} \ , \tag{B.16}$$

which diverges when

$$1 - G_1'(1) = 0 \tag{B.17}$$

or equivalently (using (B.2)) when $G_0''(1) = N\kappa$, signalling the emergence of the giant component.

When the giant component has formed, $H_0(x)$ and $H_1(x)$ (see Eq. (B.3), (B.4), (B.5)) become the sum of two contributions: one where the sum is restricted to components of size $t \sim o(N)$, and the other restricted to (giant) components of size $t \sim \mathcal{O}(N)$. Assuming that there is only one such giant component, Eq. (B.5) for $x = 1$ can then be written as

$$1 = H_0^{(f)}(1) + S \ , \tag{B.18}$$

where $H_0^{(f)}(1)$ (and similarly $H_1^{(f)}(1)$) satisfy the equations (B.9) and (B.10), as they include $\sim o(N)$ contributions for $N \to \infty$ coming from components other than the giant one, whereas $S = N_C/N$ is the fraction of nodes that belong to the giant component. Therefore (from (B.9) and (B.10))

$$S = 1 - G_0(\xi^\star) \ , \tag{B.19}$$

where $\xi^\star$ satisfies

$$\xi^\star = G_1(\xi^\star) \ . \tag{B.20}$$

Acknowledgments

SB and FC acknowledge funding by UCL Centre for Blockchain Technologies as part of the *1st Internal Call for Project Proposals on Distributed Ledger Technologies*. PV acknowledges support from the UKRI Future Leaders Fellowship grant MR/S03174X/1.

References

[1] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer, 2016.

[3] Pedro Franco. *Understanding Bitcoin*. Wiley Online Library, 2014.

[4] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572. IEEE, 2017.

[5] Joseph Poon and Thaddeus Dryja. The Bitcoin lightning network: Scalable off-chain instant payments, `https://lightning.network/lightning-network-paper.pdf`, 2016.

[6] István András Seres, László Gulyás, Dániel A Nagy, and Péter Burcsi. Topological analysis of Bitcoin's lightning network. *arXiv preprint arXiv:1901.04972*, 2019.

[7] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge University Press, 2008.

[8] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378, 2000.

[9] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Resilience of the Internet to random breakdowns. *Physical Review Letters*, 85(21):4626, 2000.

[10] Elias Rohrer, Julian Malliaris, and Florian Tschorsch. Discharged payment channels: Quantifying the lightning network's resilience to topology-based attacks. *arXiv preprint arXiv:1904.10253*, 2019.

[11] Simina Brânzei, Erel Segal-Halevi, and Aviv Zohar. How to charge lightning. *arXiv preprint arXiv:1712.10222*, 2017.

[12] Andreas M Antonopoulos. *Mastering Bitcoin: unlocking digital cryptocurrencies*. O'Reilly Media, 2014.

[13] Mike Orcutt. How secure is blockchain really. *MIT Technology Review*, 2018.

[14] David Easley, Maureen O'Hara, and Soumya Basu. From mining to markets: The evolution of Bitcoin transaction fees. *Journal of Financial Economics*, 134(1):91 – 109, 2019.

[15] Nicolas Houy. The economics of Bitcoin transaction fees. *GATE WP*, 1407, 2014.

[16] Christian Decker and Roger Wattenhofer. Information propagation in the Bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1–10. IEEE, 2013.

[17] Giuseppe Pappalardo, Tiziana Di Matteo, Guido Caldarelli, and Tomaso Aste. Blockchain inefficiency in the Bitcoin peers network. *EPJ Data Science*, 7(1):30, 2018.

[18] Alexandre Bovet, Carlo Campajola, Jorge F Lazo, Francesco Mottes, Iacopo Pozzana, Valerio Restocchi, Pietro Saggese, Nicoló Vallarano, Tiziano Squartini, and Claudio J Tessone. Network-based indicators of Bitcoin bubbles. *arXiv preprint arXiv:1805.04460*, 2018.

[19] Alexandre Bovet, Carlo Campajola, Francesco Mottes, Valerio Restocchi, Nicolo Vallarano, Tiziano Squartini, and Claudio J Tessone. The evolving liaisons between the transaction networks of Bitcoin and its price dynamics. *arXiv preprint arXiv:1907.03577*, 2019.

[20] Matthias Lischke and Benjamin Fabian. Analyzing the Bitcoin network: The first four years. *Future Internet*, 8(1):7, 2016.

[21] Dániel Kondor, Márton Pósfai, István Csabai, and Gábor Vattay. Do the rich get richer? an empirical analysis of the Bitcoin transaction network. *Plos One*, 9(2):e86197, 2014.

[22] Pavel Ciaian, Miroslava Rajcaniova, and d'Artis Kancs. The economics of Bitcoin price formation. *Applied Economics*, 48(19):1799–1815, 2016.

[23] Lin William Cong, Ye Li, and Neng Wang. Tokenomics: Dynamic adoption and valuation. *Columbia Business School Research Paper*, (18-46), 2019.

[24] Silvia Bartolucci and Andrei Kirilenko. A model of the optimal selection of crypto assets. *arXiv preprint arXiv:1906.09632*, 2019.

[25] Laura Alessandretti, Abeer ElBahrawy, Luca Maria Aiello, and Andrea Baronchelli. Anticipating cryptocurrency prices using machine learning. *Complexity*, Article ID 8983590, 2018.

[26] Stanisław Drożdż, Robert Gębarowski, Ludovico Minati, Paweł Oświęcimka, and Marcin Wątorek. Bitcoin market route to maturity? Evidence from return fluctuations, temporal correlations and multiscaling effects. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):071101, 2018.

[27] Stanisław Drożdż, Ludovico Minati, Paweł Oświęcimka, Marek Stanuszek, and Marcin Wątorek. Signatures of crypto-currency market decoupling from the forex. *arXiv preprint arXiv:1906.07834*, 2019.

[28] Higor YD Sigaki, Matjaž Perc, and Haroldo V Ribeiro. Clustering patterns in efficiency and the coming-of-age of the cryptocurrency market. *Scientific Reports*, 9(1):1440, 2019.

[29] Andrew Urquhart. The inefficiency of Bitcoin. *Economics Letters*, 148:80–82, 2016.

[30] Laura Alessandretti, Abeer ElBahrawy, Luca Maria Aiello, and Andrea Baronchelli. Machine learning the cryptocurrency market. *Available at SSRN 3183792*, 2018.

[31] Abeer ElBahrawy, Laura Alessandretti, Anne Kandler, Romualdo Pastor-Satorras, and Andrea Baronchelli. Evolutionary dynamics of the cryptocurrency market. *Royal Society Open Science*, 4(11):170623, 2017.

[32] Luisanna Cocco, Giulio Concas, and Michele Marchesi. Using an artificial financial market for studying a cryptocurrency market. *Journal of Economic Interaction and Coordination*, 12(2):345–365, 2017.

[33] T. Aste. Cryptocurrency market structure: connecting emotions and economics. *Digital Finance*, 1:5–21, 2018.

[34] Jethin Abraham, Daniel Higdon, John Nelson, and Juan Ibarra. Cryptocurrency price prediction using tweet volumes and sentiment analysis. *SMU Data Science Review*, 1(3):1, 2018.

[35] Young Bin Kim, Jun Gi Kim, Wook Kim, Jae Ho Im, Tae Hyeong Kim, Shin Jin Kang, and Chang Hun Kim. Predicting fluctuations in cryptocurrency transactions based on user comments and replies. *PloS One*, 11(8):e0161197, 2016.

[36] Tianyu Ray Li, Anup Chamrajnagar, Xander Fong, Nicholas Rizik, and Feng Fu. Sentiment-based prediction of alternative cryptocurrency price fluctuations using gradient boosting tree model. *Frontiers in Physics*, 7:98, 2019.

[37] Silvia Bartolucci, Giuseppe Destefanis, Marco Ortu, Nicola Uras, Michele Marchesi, and Roberto

Tonelli. The butterfly “affect": Impact of development practices on cryptocurrency prices. 2019.

[38] Ladislav Kristoufek. Bitcoin meets Google trends and Wikipedia: Quantifying the relationship between phenomena of the Internet era. *Scientific Reports*, 3:3415, 2013.

[39] David Garcia, Claudio J Tessone, Pavlin Mavrodiev, and Nicolas Perony. The digital traces of bubbles: feedback cycles between socio-economic signals in the Bitcoin economy. *Journal of the Royal Society Interface*, 11(99):2014.0623, 2014.

[40] Cathy Yi-Hsuan Chen and Christian M Hafner. Sentiment-induced bubbles in the cryptocurrency market. *Journal of Risk and Financial Management*, 12(2):53, 2019.

[41] Aaron Yelowitz and Matthew Wilson. Characteristics of Bitcoin users: an analysis of Google search data. *Applied Economics Letters*, 22(13):1030–1036, 2015.

[42] Duncan S Callaway, Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85(25):5468, 2000.

[43] Konrad Hoppe and Geoff J Rodgers. Percolation on fitness-dependent networks with heterogeneous resilience. *Physical Review E*, 90(1):012815, 2014.

[44] Guido Caldarelli, Andrea Capocci, Paolo De Los Rios, and Miguel A Muñoz. Scale-free networks from varying vertex intrinsic fitness. *Physical Review Letters*, 89(25):258702, 2002.

[45] Vito DP Servedio, Guido Caldarelli, and Paolo Butta. Vertex intrinsic fitness: How to produce arbitrary scale-free networks. *Physical Review E*, 70(5):056126, 2004.

[46] Ginestra Bianconi and Albert-László Barabási. Competition and multiscaling in evolving networks. *EPL (Europhysics Letters)*, 54(4):436, 2001.

[47] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2):026118, 2001.

[48] Mark EJ Newman. Component sizes in networks with arbitrary degree distributions. *Physical Review E*, 76(4):045101, 2007.

[49] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[50] Igor E Smolyarenko, Konrad Hoppe, and Geoff J Rodgers. Network growth model with intrinsic vertex fitness. *Physical Review E*, 88(1):012805, 2013.

Information Systems and e-Business Management
https://doi.org/10.1007/s10257-020-00465-5

ORIGINAL ARTICLE

Laying the foundation for smart contract development: an integrated engineering process model

Christian Sillaber[1] · Bernhard Waltl[2] · Horst Treiblmaier[3] · Ulrich Gallersdörfer[2] · Michael Felderer[1,4]

Received: 25 February 2019 / Revised: 12 October 2019 / Accepted: 18 January 2020

Abstract

Smart contracts are seen as the major building blocks for future autonomous blockchain- and Distributed Ledger Technology (DLT)-based applications. Engineering such contracts for trustless, append-only, and decentralized digital ledgers allows mutually distrustful parties to transform legal requirements into immutable and formalized rules. Previous experience shows this to be a challenging task due to demanding socio-technical ecosystems and the specificities of decentralized ledger technology. In this paper, we therefore develop an integrated process model for engineering DLT-based smart contracts that accounts for the specificities of DLT. This model was iteratively refined with the support of industry experts. The model explicitly accounts for the immutability of the trustless, append-only, and decentralized DLT ecosystem, and thereby overcomes certain limitations of traditional software engineering process models. More specifically, it consists of five successive and closely intertwined phases: conceptualization, implementation, approval, execution, and finalization. For each phase, the respective activities, roles, and artifacts are identified and discussed in detail. Applying such a model when engineering smart contracts will help software engineers and developers to better understand and streamline the engineering process of DLTs in general and blockchain in particular. Furthermore, this model serves as a generic framework which will support application development in all fields in which DLT can be applied.

Keywords Smart contract · Development process model · Software engineering · Blockchain · Distributed ledger technology · Survey · Design science · Trustless append-only decentralized digital ledgers (TADDL)

✉ Michael Felderer
michael.felderer@uibk.ac.at

Extended author information available on the last page of the article

Published online: 06 February 2020

1 Introduction

Within the last couple of years, blockchain technology, or, more generally, distributed ledger technology (DLT), has become a highly popular research topic and been recognized as a potential game-changer for the industry. The pseudonymous author (or group of authors) Satoshi Nakamoto, who mentioned neither the term "blockchain" nor "DLT" in his paper, introduced the cryptocurrency Bitcoin as the first use case of this technology (Nakamoto 2008). The years following this seminal paper were characterized by intense discussions in academic communities specialized in cryptography and dedicated online groups, but the full potential of the technology was not yet properly understood within applied academic and business-oriented communities. This situation changed around the year 2014 when the exchange rate of Bitcoin started to soar and several authors highlighted the potential of the technology for countless use cases beyond cryptocurrencies (Swan 2015; Tapscott and Tapscott 2016). Not surprisingly, various academic communities have now started to rigorously investigate the topic and an ever-increasing number of papers is being published in various academic outlets (Johansen 2018).

The immutable, trustless model of decentralized computation and transaction handling which is provided by the blockchain strives to ensure fairness for all participating actors and has led to an ever-increasing market capitalization of cryptocurrencies in recent years. By June 2018, the capitalization of the five leading cryptocurrencies (Bitcoin, Ethereum, Ripple, Bitcoin Cash, and Litecoin) exceeded 193 billion dollars (Coinmarketcap 2018). However, the functionality of blockchain does not stop at cryptocurrencies. One of the most widely discussed features of blockchain technology is the possibility to create decentralized and self-executing programs: so-called *smart contracts*. Current application scenarios include areas as diverse as supply chain/logistics, finance, tourism, Internet of Things (IoT), project operations, gamification and education (Treiblmaier and Zeinzinger 2018; Tapscott and Tapscott 2016). Numerous organizations are already applying this technology stack to supplement or supplant existing legal and financial transactions (Egelund-Müller et al. 2017; Fanning and Centers2016; Friedlmaier et al. 2016; Clack et al. 2016; Notheisen et al. 2017). The huge monetary values processed through smart contracts and/or represented by cryptocurrencies necessitate structured software development processes and high levels of quality assurance. Incidents such as the DAO attack (Siegel 2016) as well as the King of the Ether incident (KingOfTheEther 2016), aggregating damage in excess of 60 million dollars, illustrate (1) that the current ad-hoc style of engineering is not suitable for such high-value transactions, (2) that existing software engineering approaches do not ensure a sufficient level software quality, and (3) that these approaches are either unsuitable for or misaligned with the specificities of blockchain technology (Atzei et al. 2016). Atzei et al. (2017, p. 182) point out that "a common cause of insecurity of smart contracts is the difficulty of detecting mismatches between their intended behavior and the actual one". Subsequently, Destefanis et al. (2018) encouraged the academic community to further investigate and develop blockchain-oriented software engineering processes; We answer that call.

Traditional software engineering focuses on principles for developing high-quality software systems and maintaining the systems as they evolve in real-world environments (Mens et al. 2010; Edan and Pliskin 2001). Software that does not evolve during its lifetime, however, will not be able to keep up with changing requirements and will become outdated over time. This has profound implications for existing software process models, which must respond to the increasing need for change and evolution by introducing iterative, incremental, and evolutionary approaches (Mens et al. 2010; Boehm and Turner 2005; Beck et al. 2001). In demonstration, software maintenance and evolution have emerged in the last decade as key research fields that explicitly differentiate between the time phases before, during, and after the software is delivered: denoted development time, deployment time, and runtime, respectively (Jacobson et al. 1999). Post-deployment changes are typically realized by returning to regular development activities, which eventually result in a new version of a software product or a patch that is released to replace or enhance the currently running version during scheduled downtimes. This process is structured by change management activities (Stark 2015; Rajlich and Bennett 2000; Bennett and Rajlich 2000). All such changes that occur after the initial development time are impossible, however, in settings where the contracts are published on blockchain and are immutable from that point onward. To clarify, and to counter an often-repeated yet incorrect narrative, DLT systems are only immutable with respect to their specific context and rules. Changes in the execution environment, in the stakeholder agreement (e.g., switching to another technology or invalidating specific entries), or in the usage of specific patterns (e.g., proxy pattern) yield mutability.

In this paper, we systematically develop a smart contract engineering process that clearly outlines its respective elements and artifacts. This process description represents an essential tool for numerous strategic and operational activities since it helps in defining priorities, clarifying risks, and managing expectations and time frames. We further develop a framework that supports stakeholders of smart contract engineering processes in their coordination efforts and which can be used for activities such as project management and legal risk management, complexity and standards management, as well as for security and quality management.

This paper is structured as follows: First, we identify research topics related to DLT technology and give a short introduction to trustless, append-only, decentralized digital ledgers, and to related software engineering process models. Second, we describe our methodological approach consisting of expert interviews and qualitative content analysis. Third, we develop an integrated process model for DLT technology in a stepwise process. Finally, a short discussion and a comparison to conventional engineering processes concludes this paper.

2 Identification of DLT research topics

The immutable nature of smart contracts in trustless, append-only, and decentralized digital ledgers makes the traditional software engineering lifecycle both inappropriate and insufficient (Sillaber and Waltl 2017). Instead, the technical specificities of blockchain technologies demand central consideration as a robust frame of reference

that helps in decomposing its overall complexity and accommodating the new requirements of smart contract engineering. In Table 1 we list several research topics and questions pertaining to a variety of different managerial aspects of software development that need to be systematically addressed. In the domain of *project and legal risk management*, we list topics surrounding the consideration of all relevant issues and requirements prior to the deployment of the smart contract, including cost comparisons, project duration, mitigation of legal and project-related risks, and potential problems arising due to legal requirements related to "anti-money laundering" (AML) and "know your customer" (KYC). When it comes to *complexity and standards management*, questions arise as to which concepts and elements of smarts contracts need to be standardized, which components of contracts should be individualized: considerations include the decomposition of complexity, the impact of smart contracts on negotiation processes, the integration into legacy systems as well as the portability of existing systems on the blockchain, the identification of the best modeling languages and smart contract patterns, and the handling of contracts that are interrelated. Important issues related to *security and quality management* include the testing and validation of smart contracts, the mitigation of risks that might arise from bugs and vulnerabilities not known at the time of development, the auditing of smart contracts, and the security of the underlying platform.

3 Related work

Although the term *blockchain* is relatively new (Swan 2015; Tapscott and Tapscott 2016), its underlying concepts are not. Some of the foundations of this technology, such as Merkle trees, proof of work algorithms, or smart contracts, were already developed decades ago (Narayanan and Clark 2017). Smart contract engineering, therefore, builds on (1) the conceptual foundation of smart contracts, as well as on (2) state-of-the-art software engineering with a focus on blockchain technology. We briefly outline both topics in the following sections.

3.1 Smart contracts in trustless, append-only, decentralized digital ledgers

The term *smart contract* was introduced by Szabo (1997) when he first described how the computer-based execution of contracts between two parties can be secured without requiring a third party for intermediation or confirmation. His original article provides the first description of decentralized smart contracts as computer programs that are executed by all participants. This allows all participating parties, who do not necessarily know or trust each other, to securely transact with each other. The correct execution of these programs is ensured by a so-called consensus protocol (Luu et al. 2016).

The basic technological properties of trustless, append-only, decentralized digital ledgers (TADDL), which includes blockchain technology, are well-studied and described in the literature (e.g., Tschorsch and Scheuermann 2016). Several separate active research streams focusing on specific technological issues of TADDLs can

Table 1 Research topics

Domain	Related questions	Sources
Project and legal risk management	How can all relevant issues and requirements of stakeholders be safeguarded before the smart contract is finalized in the blockchain? What are the costs of smart contracts compared to offline contracts? What is the expected project duration of the implementation? How can well-known risks be mitigated and unknown risks be identified as early as possible? How can legal risks be mitigated? How can challenges from AML and KYC be efficiently addressed?	Wang et al. (2016), Rückeshäuser (2017), Deshpande et al. (2017), Porru et al. (2017), Xu et al. (2017), Böhme et al. (2015), Pesch and Sillaber (2017), Fairfield (2014), Kiviat (2015), Moyano and Ross (2017)
Complexity and standards management	Which concepts and elements of smart contracts need to be standardized across the entire ecosystem or within specific legal environments? Which components of smart contracts should be individualized for each customer? How can the complexity that results from the holistic and comprehensive nature of legal requirements and their translation into smart contracts be decomposed to make it manageable? How can the codification of legal requirements into smart contract code improve the negotiation process between the involved parties? How can smart contracts be integrated into existing IT systems? Which parts of an enterprise IT Architecture can be ported to blockchain? What are the best modeling notations for smart contract development? What are the patterns in smart contracts? How should contracts that have dependencies between them be dealt with?	Seijas et al. (2016), Frantz and Nowostawski (2016), Marjanovic and Milosevic (2001), Clack et al. (2016), Beck and Müller-Bloch (2017), Bartoletti and Pompianu (2017)
Security and quality management	How can smart contracts be tested and validated? How can stakeholders efficiently mitigate risks from bugs and vulnerabilities in smart contracts? How can smart contracts be audited? How secure is the underlying platform?	Delmolino et al. (2016), Clack et al. (2016), Bhargavan et al. (2016), Atzei et al. (2016), Idelberger et al. (2016), Leitner et al. (2007)

be identified, including topics such as anonymity versus pseudonymity, transaction rates, and proof-of-X (Tschorsch and Scheuermann 2016; Anderson et al. 2016). A TADDL is a decentralized virtual state and computing machine that enables several parties to share a common state, the integrity of which is ensured and verified by other participating parties or "volunteers", including, for example, miners (Anderson et al. 2016). Various incentives promote participation in the mining process, most

notably coin rewards. The use of a so-called consensus protocol that is binding for all participating parties and which forms the mechanism through which consensus is achieved within a peer-to-peer network, implies that the data being stored—cryptocurrency asset balances, for instance—are accepted by all participants. With the components provided by the TADDL, complex digital asset transactions and financial instruments can be created (Buterin 2014; Koulu 2016; Anderson et al. 2016). The use of smart contracts executed in a TADDL can be observed in many different domains, ranging from online gambling to fundraising (e.g., Porru et al. 2017; Egelund-Müller et al. 2017; Xu et al. 2017; Böhme et al. 2015; Klöhn et al. 2018).

3.2 Software engineering process models

The IEEE 1074-1995 Standard for Developing Software Lifecycle Processes defines a process as a set of steps that can be executed in a certain predefined, sequential, parallel, or conditional order (IEEE 1995). Software engineering processes are part of the general Software Engineering Body of Knowledge (Bourque and Fairley 2014). Various process models cover the order and frequency of phases in software projects. Those phases typically include planning, analysis, design, implementation, testing, and maintenance. Waterfall models progress sequentially through these phases, whereas iterative models are typified by repeated execution of the waterfall phases, in whole or in part (Braude and Bernstein 2016). Differing from these phase-oriented process models, agile process models are based on the principles of individuals and interaction, working software, customer collaboration, and fast response to change (Beck et al. 2001; Vidgen and Wang 2009; Lee and Xia 2010). A recent trend is to combine phase-oriented with agile process models to obtain hybrid software engineering process models (Kuhrmann et al. 2017).

Modern software engineering approaches rely heavily on the (re-)use of software patterns (Kuhrmann et al. 2017). Patterns are collections of abstract best practices of software code that engineers can easily adapt. These best practices are the result of previous software engineering experience and often allow faster, more secure, and more reliable software development. As industry experience with smart contracts grows, it is very likely that a set of smart contract patterns will emerge in order to foster efficiency and effectiveness in the creation of smart contracts.

In their overview on the current status of research and practice regarding software engineering process models, Fuggetta and Di Nitto (2014) highlight several challenges caused by the Internet as a basic development, execution, distribution, and business infrastructure. They list research issues such as the fading distinction between design, development, and operation, but also highlight topics such as security, privacy, and trust. Blockchain-oriented software engineering has also attracted recent interest. Porru et al. (2017), for example, outline new research directions for blockchain-oriented software engineering processes, which include the areas of collaboration, enhancement of testing and debugging, as well as the creation of software tools for smart contract languages. In this paper, we extend previous research by developing an integrated process model for smart contract engineering.

4 Methodology

We conducted interviews on smart contracts with eleven industry experts. Table 2 gives an overview of the participants, their organizational roles, qualifications, and previous involvement in blockchain projects, structured by blockchain type and use case. The primary goal of the interviews was to get a better understanding of how the study participants develop smart contracts and which processes, artifacts, and tools they apply. We used a Delphi study approach wherein the findings from the first round (interview partners 1–9) were evaluated and refined in the second round, in which interview partners 8–11 participated (Prusty et al. 2017). Interview partners 8 and 9 were part of both rounds and helped to connect the findings by critically commenting on the feedback from the second round. The experts were identified by contacting the respective leaders of the development teams of the 20 largest blockchain projects, as measured by their token market capitalization listed on ICOAlert (www.icoalert.com) as at November 2017, as well as through the authors' personal networks. All potential experts were invited via email and eleven of them agreed to participate in our study. The participants were briefly informed by the researchers about the context, goals, and scope of the study, and the interviews were conducted via video conferencing or in person. Each interview was recorded and transcribed. After the interviews, the resulting process model, as well as the changes resulting from the interview, were sent to the interviewees for further feedback, which was then again incorporated by the researchers.

We used the open questions shown in Table 3 to structure the interviews and frequently applied follow-up questions to clarify specific issues (cf., Chau and Tam 1997). In a first step, the experts were informed about the goals and the procedures of this research project. Most notably, we presented various intermediate versions of the process model that included modifications and extensions based on the findings generated from previous interviews. Next, they discussed the different stages of the processes they use in their companies in a stepwise manner and included their findings in the model. Finally, summaries were produced from the interviews in order to derive concepts and constructs for further model development (Mayring 2014; Lacity and Janson 1994).

5 Process model development

In the following sections we develop the smart contract engineering process in a stepwise manner. First, we discuss the conceptual base and describe the main types of artifacts that emerged from the interviews. Second, we discuss the findings from the qualitative interviews with several software developers. Third, we present various roles, activities, and artifacts and, fourth, we incorporate these components into one integrative model.

5.1 Conceptual base

We followed a design science approach to precisely define the respective steps of the process model (Baskerville et al. 2018; Zakarian and Kusiak 2001). The core concept of design science is the artifact: an object that can be instantiated with physical or social properties. Examples of artifacts can be as diverse as software, models, or norms (Hevner et al. 2004). In their proposed research framework for the conceptualization of design science research within information systems (IS) research, Hevner et al. (2004) propose three integrated dimensions: (1) the environment including people, organizations, and technology, (2) IS research pinpointing the creation and justification of artifacts, and (3) the knowledge base bringing forward foundations and methodologies to be used in the creation and evaluation of artifacts. More specifically, March and Smith (1995) differentiate between four types of artifacts: constructs, models, methods, and instantiations. *Constructs*, which consist of language and vocabulary specifying problems and solutions, form the baseline design science vocabulary. The construct level establishes a common understanding of the involved entities, by identifying those entities, their attributes, and the relationships between them. Furthermore, constructs describe the terms being used and ensure their consistent usage throughout the domain. *Models* are descriptions and representations of real-world phenomena with a focus on utility, not truth (March and Smith 1995). They can therefore be abstract and may represent nothing more than arbitrary aggregations and groupings of instances. In order to be useful, the entities chosen for the model have to be representative of the underlying information system (Wood 2014). The steps needed to execute a specific process are called *methods*, which are procedures for solving problems and developing solutions. Methods are built on constructs and models. They are used to transform constructs and models from one representation into another and consequently operate on models and concepts as input and output parameters. Methods also subsume abstract algorithms and procedures (e.g., human activities) which are part of the overall process. *Instantiations* (i.e., physical assets) are the realizations of artifacts within their respective environments. They constitute the most concrete entities among the four different artifact types and are most suitable for empirical analysis including performance measures in terms of effectiveness and efficiency of the smart contract engineering process. In Table 4 we map the artifacts from Hevner et al. (2004) to the domain of smart contract engineering, with the respective artifacts shown in the left column and their manifestations within the domain in the right column.

5.2 Expert evaluation

In order to create an integrated process model that accounts for the specificities of smart contracts, we thoroughly analyzed the previous experiences from our experts following the guidelines for qualitative research (Mayring 2014) and design science research (Hevner et al. 2004). All but one interviewee (#7) were familiar with existing software engineering process models and confirmed that they actually build their

Table 2 Survey sample

ID	Role	Smart contracts experience	Software engineering experience	Highest degree	Blockchain type/domain
1	Independent developer	6 months	> 5 years	PhD (CS)	Ethereum with a focus on projects related to energy
2	Independent developer	More than 12 months	> 10 years	MSc (CS)	Ethereum with a focus on projects related to gambling
3	Head of the development team	6 months	> 10 years	MBA	Ethereum, hyperledger
4	Senior developer	6 months	> 10 years	PhD (CS)	Ethereum with a focus on prediction markets
5	Head of the development team	6 months	> 5 years	None	Ethereum, IOTA
6	Senior developer	More than 12 months	> 5 years	PhD (physics)	Ethereum with a focus on projects related to gambling
7	Junior developer	4 months	3 years	BSc (CS)	Ethereum with a focus on projects related to finance
8	Senior developer	6 months	> 5 years	None	Ethereum, did not want to disclose
9	Independent developer	8 months	> 5 years	None	Ethereum, hyperledger
10	Independent developer	4 months	> 20 years	PhD (CS)	Ethereum with a focus on projects related to gambling
11	Head of the development team	12 months	> 5 years	None	Ethereum, bitcoin

Table 3 Expert interviews

Professional background
What is your formal education?
What is your experience as a software developer/engineer?
Smart contracts development in general
Briefly describe the kind of smart contracts you develop.
Do you develop smart contracts for public/private or permissioned/permissionless blockchains?
For which blockchains do you develop smart contracts? Ethereum, Neo, Hyperledger, etc.?
Smart contract development process
Please describe how the roles of your smart contract development team are organized
How many team members are involved in the development of smart contracts?
Do you use a modeling approach?
Do you have a structured software development process? Which practices do you adopt?
What type of notation does your team use to document requirements for smart contracts?
Which programming languages and environments do you use when developing smart contracts?
Do you use tools to support the SW engineering? If yes, which one(s)?
Testing in smart contract development processes
Do you think that vulnerability to security incidents (e.g., due to software bugs) is a problem in current smart contract development?
Which parts are the hardest to test?
How often are software testing practices carried out during smart contract development? How often do you conduct the different types of test activities?
Do you automate your testing activities? To what extent? How do you incorporate security testing in this process?

own processes on them according to their needs—including modifications that are needed to account for the specifics of TADDL environments. All interviewees concurred that developing smart contracts is different from developing software in traditional ways and more comparable to developing hardware: "… it is much more like developing hardware that is shipped off to customers—without any chance to fix bugs once it has been sent off". Additionally, we found almost all described projects to be a mix of traditional software engineering and TADDL-specific engineering (cf "non-SC specific development" in Fig. 2). Additionally, all interviewees agreed that the submission of the smart contract to the live TADDL constitutes the most criticial part of the whole development process. For example, one interviewee stated that "… we even follow a paper-based approach where the entire team has to sign off before it is submitted. The entire team has to be present—it is very ceremonial".

Some interviewees had already perceived problems with existing testing approaches in a blockchain-based environment: "It is not possible to test under real-world conditions—we have a Testnet, but can never be sure to have similar conditions as in the real [TADDL] network". There was a general agreement regarding the importance of the analysis, specification, and validation of the implementation against the requirements: "We have tight feedback loops, where both the backend developers as well as the smart contract developers and our customers discuss the requirements and the implementation". Approval of a smart contract is in general

accompanied by extensive documentation: "… we document the entire [approval]. We print out all the relevant documentation and put it into a binder. We have test reports, coverage reports and, most importantly, the signed approval from our clients in there". Another interview partner reported challenges while launching their platform. As demand for the system exceeded expectations, too many server-side transactions were issued in a short period. As transaction costs were not adjusted, *thousands of transactions were* computed in the wrong order and thus failed. The engineers concluded that the calculation of transactions costs needs access to real-time data. Additionally, a queue for to-be-issued transactions was implemented within the platform.

Several major challenges were explicitly mentioned with regards to the availability of testing data from oracles. As some of our interviewees develop smart contracts that are intended (at least in theory) to run forever, they support the idea of a dedicated finalization phase—but were not able to specifically link it to their respective use cases. A comprehensive documentation is therefore important during contract runtime: "We definitely store everything, even after the smart contract is no longer under active development—especially since we do not know when something bad [referring to the DAO hack] might happen". Finally, based on the fact that smart contracts cannot be changed after deployment, a careful monitoring during runtime turns out to be crucial: "We have a dedicated watchdog [i.e., custom piece of software] that tracks the smart contract's spending and alerts us in case something odd happens."

Table 4 Mapping of Artifacts

Artifact	Manifestation within the domain
Construct	Trustless, append-only, decentralized, digital ledgers (TADDL) Cryptocurrency assets (i.e., tokens) Smart contract execution engine Smart contract expression language Actors (e.g., legal party, smart contract engineer, oracle, miner, legal expert) Wallets
Model	Smart contract code, templates, and patterns Transaction schemes The digital representation of assets Consensus and reward algorithms Interactions via transactions, function calls, oracle inputs
Method	Smart contract engineering (sub-)activities Iterations of the engineering process Simulation activities Test methods for smart contracts
Instantiation	An instance of the smart contract engineering process with its activities Operationalized smart contracts [e.g., instances of high-level languages compiled to Ethereum Virtual Machine bytecode (Wood 2014)] Results from smart contract test scenarios (e.g., reports and log files) Results from smart contract executions and simulations (e.g., transactions)

5.3 Roles, activities, and artifacts

To formally and constructively describe an engineering process, the Rational Unified Process (RUP) Framework (Kruchten 2004) can be used as a baseline for adaption since it is document-centric and reflects the smart contract development process. More specifically, the RUP is used to differentiate between three distinctive elements: First, *roles* pertain to individuals or groups performing activities within the process. Such roles, which might include smart contract engineers, software engineers, and legal experts, are responsible for the artifacts that are the outcome of their activities. Second, *activities* summarize a unit of work that must be performed. The outcome is the creation or update of artifacts. Third, *artifacts* denote the input and output of activities. Artifacts are created, modified, and used by the roles during the procedure and are either the final product, parts of it, or intermediate results. Examples of artifacts include concepts, models, source code, smart contract code, or documents such as performance reports.

Figure 1 illustrates the lifecycle of a smart contract. Sillaber and Waltl (2017) have shown that smart contract lifecycles start with *an implementation phase*, during which requirements are transformed into an implementation (Create and Adapt), verified against the requirements, and either approved for release or modified again. Once the smart contract is approved, it is published on the TADDL in the *submission stage*. In this phase, the smart contract is submitted and distributed within the TADDL network. From that point on, every entity with access to the TADDL can retrieve the contract and share it with other nodes. Once the smart contract has been spread throughout the network and is accepted by general consensus (i.e., it persists on the network), reverting or changing it requires—under ideal circumstances—substantial effort. The contract is now ready to be executed. In a paper contract analogy, this would be the signing of the contract, which is one important step to make the contract valid and enforceable. The *execution stage* of smart contracts is performed by miners or other participants of the TADDL, since the smart contract code is now accessible for all participants in the form of bytecode. To execute it, the smart contract is retrieved from the TADDL and carried out by the respective node (*compute*). Based on a given input, the output (e.g., a return value, a state transition, or a set of transactions) of a smart contract is computed, which is then stored and distributed within the network. This is similar to the "closing" of an offline contract. A smart contract can be executed as long as it is active. Its execution is resource-consuming and the nodes contributing computational power for its execution are rewarded according to the distributed ledger's reward scheme. In the *finalization stage,* the smart contract expires. This can happen either because the parties actively declare the smart contract as invalid (e.g., by withdrawing remaining funds or executing an appropriate function) or because of intrinsic conditions that make further executions impossible (e.g., time expiration; inability to pay the fees required for further execution). In this case, the smart contract remains in the TADDL, but can no longer be executed by the nodes. This means that the smart contract is disabled through a conditional exit that prevents future execution. This is akin to the "final" state of a business process, where specific properties (e.g. successful execution, final state) depend on the specific context.

5.4 An integrated smart contract engineering process

Figure 2 combines and summarizes our findings from the literature as well as the expert interviews into a comprehensive framework for the integrated smart contract engineering process. In the *conceptualization phase*, the preliminary scope and the goals of the smart contract are defined. The scope informs all involved parties about what will be, and what will not be, part of the smart contract and can be directly derived from traditional contractual requirements. This process is called requirements elicitation. The problem definition should also state the desired economic outcome(s). After reaching an agreement about the scope, the next step is the conceptual modeling: the transformation of the requirements of all involved parties into a smart contract model. In this phase, the conceptual model is created. The conceptual model defines classes of objects (e.g., wallets) and the desired relations between these objects and outcomes (e.g., transactions). The construction of the conceptual model will most likely uncover incomplete and contradictory aspects of the problem definition. Additionally, the modeling process may raise new questions for the involved parties to answer and resolve through negotiation. In either case, the problem definition should be adjusted.

After the conceptual modeling phase, the *implementation phase* starts. Here, the conceptual model is mapped onto an executable model (e.g., in Ethereum by using Solidity code) as existing smart contract patterns are identified, adapted and combined. It is critical to note that for performance and cost reasons, most business logic to be implemented will be executed outside the smart contract, within a "traditional software" application (termed "non-smart contract code"). The identification of those parts that should be included in the smart contract, and those that should be excluded, requires a thorough analysis of functional requirements as well as non-functional requirements for confidentiality, integrity, availability, scalability or efficiency. This analysis may even be performed as a formal risk analysis. In a banking application, for instance, core-banking functionality may be implemented as a smart contract, whereas data visualization on a traditional software stack.

An executable smart contract is not necessarily immediately correct and has to be reviewed, tested, and verified. Verification and simulation of the smart contract against the scope and stakeholder requirements are necessary to check whether the code contains errors, including programming errors and mal-adjusted parameters. For verification purposes ("Simulation, testing, code review"), various scenario-based executions can be simulated step-by-step in a private blockchain. Apart from verification, validation of the smart contract is also required. During validation, the simulation results of the smart contracts are compared to real-world contract states and stakeholder requirements. New insights may even lead to an adjustment of the problem definition and/or the conceptual model of the smart contract. A simulated

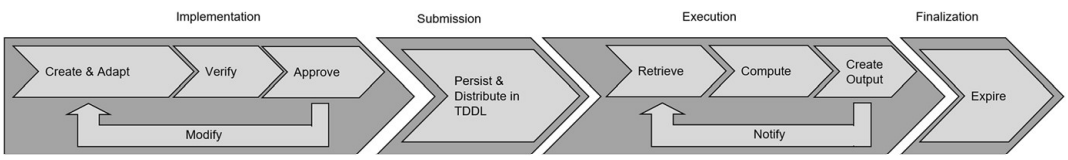


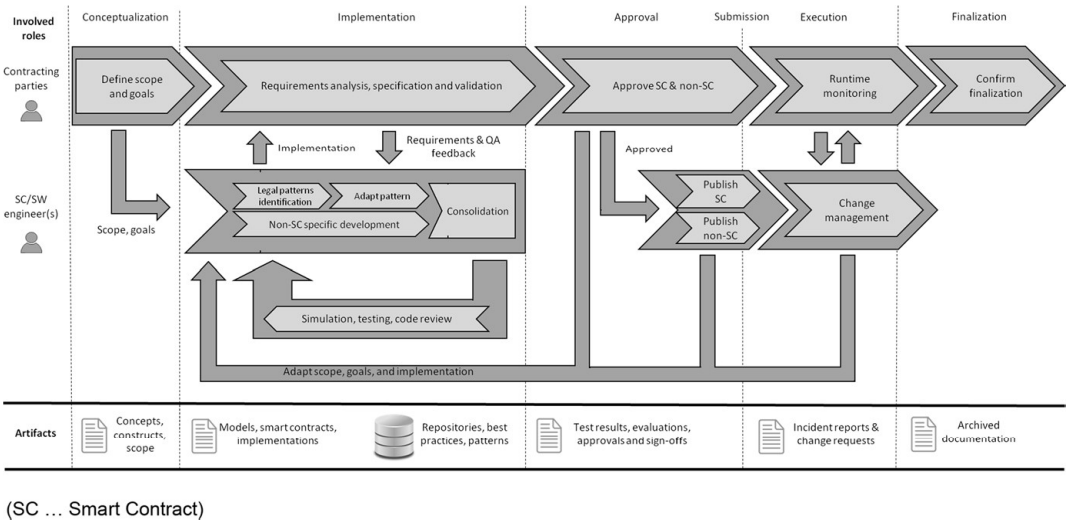
Fig. 1 Simplified lifecycle of a smart contract

smart contract found to be correct after validation is called a validated smart contract after the last round of consolidation.

Starting from the consolidated and validated smart contract, an instance of the smart contract can be frozen and submitted for execution in the live TADDL environment. Finally, in the *approval and execution phases*, the published smart contract is approved and executed in the TADDL and has to be monitored during runtime. In case the smart contract's behavior deviates from the stakeholders' requirements, appropriate change management mechanisms have to be activated: in extreme cases, the deactivation of the smart contract by depleting its funds and the creation of a new smart contract which better meets the stakeholders' requirements. Modifications of non-smart contract components are possible throughout the entire lifecycle of the smart contract. Although the smart contract becomes immutable after it has been submitted to the TADDL environment, the environment itself often provides opportunities to influence the outcome of smart contracts: for example, by influencing the call graph through a function registry or call delegation. The smart contract's runtime behavior is constantly monitored and managed in a change management process. Once the smart contract has reached the end of its life (e.g., by executing the "self-destruction" operation in the Ethereum blockchain), proper finalization can be confirmed in the *finalization phase* by validating whether the desired outcomes have been reached. Figure 2 further shows that feedback between phases is possible and frequently necessary. In practice, many phases will overlap. More specifically, specification, implementation, validation, and verification will go hand in hand once the appropriate tools are available to smart contract developers.

6 Discussion and implications

Smart contracts may well become the backbone of businesses based on blockchain and related technologies (Bailis 2017; Werbach and Cornell 2017). However, prior to the creation of industry-specific solutions, it is prudent to consider the general



(SC … Smart Contract)

Fig. 2 Integrated smart contracts engineering process

characteristics of smart contracts and the roles they play during their lifecycle. In this paper, we therefore present an integrated process model for smart contracts that was developed and iteratively improved using findings from previous research and the feedback of eleven industry experts. This model highlights the important role of smart contracts during their lifecycle and thus supports quality management in software engineering. This is of utmost importance for the business community striving to use blockchain-based solutions, since immutable bugs in smart contracts with no possibility of rectification have been exploited in previous attacks. For example, the controversial hard-fork of the Ethereum blockchain, which basically nullified the effects of malicious transactions, poses an example of how laborious and far-reaching ex-post changes on blockchain can potentially be (Buterin 2016).

Our proposed smart contract engineering process is generic and is applicable to a wide variety of distributed ledger technologies. It is based on traditional software engineering process models and methodologies, such as the waterfall model as well as iterative models that have been successfully applied in a wide variety of use cases, and can be easily integrated with these existing models. For example, one cycle of the implementation phase can be aligned with a Scrum sprint (Schwaber and Beedle 2002). Traditional phase-oriented software engineering process models like the waterfall model typically progress linearly through an analysis, design, implementation, and testing phase. While the analysis, design and implementation phases align with our proposed conceptualization and implementation phases, special care has to be given to the testing phase of smart contracts, as this must be conducted and concluded prior to publishing the contract. Iterative software engineering process models typically iterate sequentially through the aforementioned four phases. The implementation phase proposed in this paper iterates through a pattern selection and adaption, development, consolidation, review, testing, and simulation phase, aligning these process activities with iterative software engineering process models.

Although further refinement of different aspects of the integrated process model may be necessary for specific applications, the integrated model as presented in this paper can immediately be applied in real-world industry settings. It can help smart contract engineers to better understand the strengths and weaknesses of their engineering processes and support them in further optimizing different process activities and artifacts such as software, models, and norms. Additionally, the process model can advance applied smart contract engineering processes and serve as a basis for critically investigating those processes in great detail, which is of practical value for any industry that needs to react fast while at the same time ensuring supreme software quality. Additionally, there are also implications for academic research. The engineering process model is accompanied by directions concerning the involved artifacts, roles, and interdependencies, and thereby lays the foundation for future research. This may include a detailed specification of the different roles and their required profiles.

7 Conclusions, limitations and future research

In this paper, we develop an integrative process model for smart contract engineering and describe its activities, roles, and artifacts. This model lays the foundation for further smart contract development, which has the potential to revolutionize many different industries. We argue that conventional software engineering process models do not provide adequate support for the trustless, append-only, and decentralized environment in which smart contracts are executed. Traditional process models do not account for the immutability of smart contracts after they are submitted, because they assume a (mostly) frictionless transition between software releases and that modifications of existing software releases are possible. Our smart contract engineering process accounts for these peculiarities of blockchain-based software development and consists of five sequential phases: (1) conceptualization, (2) implementation, (3) approval, (4) execution, and (5) finalization. These phases are derived from the properties of the underlying blockchain ecosystem. We propose new directions for smart contract engineering that focus on collaboration among domain experts, testing activities, quality assurance, and specialized workflow tools.

Currently, we see two major limitations of this research that deserve further attention. First, there are no validated measurements for the concepts of the process activities. An attempt was made to use existing process model artifacts and to make as few changes as possible. However, the construct validity of these artifacts cannot currently be guaranteed. Second, due to a lack of established best practices in smart contract engineering, an empirical evaluation of the hypothesized artifacts is not currently feasible.

Future research, therefore, needs to investigate if and how the proposed engineering process model can be tailored to and with different software engineering methodologies (e.g., Scrum, V-model). In this context, research could investigate how the development of smart code can be integrated with the development of traditional software code, as well as how risk analysis can support this integration. Furthermore, it is necessary to integrate this framework with existing work on testing and quality assurance in software engineering. An important aspect here is especially the role of simulation for quality assurance. The behavioral aspects of smart contract engineering have not yet received enough attention. While data is sparse, we have seen DevOps and "full-stack" software engineering behavior with many interviewees and many interesting patterns (e.g., randomness patterns or oracle patterns) that have been adapted from these and related disciplines that warrant future research. Furthermore, there is a pending need to cover the increasing demand for inter- TADDL transactions and developing secure applications that rely on more than one TADDL. Combining various approaches will lead to new insights into how best to cope with the challenges of modern blockchain-based software development and how smart contracts can be used to create viable business models.

Acknowledgements Open access funding provided by University of Innsbruck and Medical University of Innsbruck.

References

Anderson L, Holz R, Ponomarev A, Rimba P, Weber I (2016) New kids on the block: an analysis of modern blockchains. CoRR. arXiv preprint http://arxiv.org/abs/1606.06530

Atzei N, Bartoletti M, Cimoli T (2016) A survey of attacks on Ethereum smart contracts. IACR Cryptology ePrint archive 2016:1007

Atzei N, Bartoletti M, Cimoli T (2017) A survey of attacks on Ethereum smart contracts SoK. In: Proceedings of the international conference on principles of security and trust, Uppsala, Sweden, pp 164–186

Bailis P (2017) Research for practice: cryptocurrencies, blockchains, and smart contracts; hardware for deep learning. Commun ACM 60(5):48–51

Bartoletti M, Pompianu L (2017) An empirical analysis of smart contracts: platforms, applications, and design patterns. In: Proceedings of the international conference on financial cryptography and data security. Springer, Cham, pp 494–509

Baskerville R, Baiyere A, Gregor S, Hevner A, Rossi M (2018) Design science research contributions: finding a balance between artifact and theory. J Assoc Inf Syst 19(5):358–376

Beck R, Müller-Bloch C (2017) Blockchain as radical innovation: a framework for engaging with distributed ledgers as incumbent organization. In: Proceedings of the 50th Hawaii international conference on system sciences, Hawaii, HI, pp 5390–5399

Beck K, Beedle M, Van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, Kern J, Marick B, Martin RC, Mellor S, Schwaber K, Sutherland J, Thomas D (2001) Manifesto for agile software development. http://agilemanifesto.org/. Accessed 20 Apr 2018

Bennett KH, Rajlich VT (2000) Software maintenance and evolution: a roadmap. In: Proceedings of the conference on the future of software engineering, pp 73–87

Bhargavan K, Delignat-Lavaud A, Fournet C, Gollamudi A, Gonthier G, Kobeissi N, Kulatova N, Rastogi A, Sibut-Pinote T, Swamy N, Zanella-Béguelin S (2016) Formal verification of smart contracts: short paper. In: Proceedings of the 2016 ACM workshop on programming languages and analysis for security, pp 91–96

Boehm B, Turner R (2005) Management challenges to implementing agile processes in traditional development organizations. IEEE Softw 22(5):30–39

Böhme R, Christin N, Edelman B, Moore T (2015) Bitcoin: economics, technology, and governance. J Econ Perspect 29(2):213–238

Bourque P, Fairley RE (2014) Guide to the software engineering body of knowledge (swebok (r)): version 3.0. IEEE Computer Society Press, Washington, DC

Braude EJ, Bernstein ME (2016) Software engineering: modern approaches. Waveland Press, Long Grove

Buterin V (2014) A next-generation smart contract and decentralized application platform. White paper. https://github.com/ethereum/wiki/wiki/White-Paper#decentralized-autonomous-organizations. Accessed 10 Jan 2018

Buterin V (2016) Hard fork completed. Ethereum Blog. https://blog.ethereum.org/2016/07/20/hard-fork-completed/. Accessed 17 Dec 2017

Chau PYK, Tam KY (1997) Factors affecting the adoption of open systems: an exploratory study. MIS Q 21(1):1–24

Clack CD, Bakshi VA, Braine L (2016) Smart contract templates: essential requirements and design options. CoRR. arXiv preprint http://arxiv.org/abs/1612.04496

Coinmarketcap (2018) Top 100 cryptocurrencies by market capitalization. https://coinmarketcap.com/. Accessed 17 Apr 2018

Delmolino K, Arnett M, Kosba A, Miller A, Shi E (2016) Step by step towards creating a safe smart contract: lessons and insights from a cryptocurrency lab. In: Clark J, Meiklejohn S, Ryan PYA, Wallach D, Brenner M, Rohloff K (eds) Proceedings of the international conference on financial cryptography and data security, pp 79–94

Deshpande A, Stewart K, Lepetit L, Gunashekar S (2017) Distributed Ledger technologies/blockchain: challenges, opportunities and the prospects for standards. Overview report. The British Standards Institution (BSI). https://www.bsigroup.com/PageFiles/508003/BSI_Blockchain_DLT_Web.pdf. Accessed 1 May 2018

Destefanis G, Marchesi M, Ortu M, Tonelli R, Bracciali A, Hierons R (2018) Smart contracts vulnerabilities: a call for blockchain software engineering? In International workshop on blockchain oriented software engineering (IWBOSE), Campobasso, Italy, pp 19–25

Edan Y, Pliskin N (2001) Transfer of software engineering tools from information systems to production systems. Comput Ind Eng 39(1–2):19–34

Egelund-Müller B, Elsman M, Henglein F, Ross O (2017) Automated execution of financial contracts on blockchains. Bus Inf Syst Eng 59(6):457–467

Fairfield JA (2014) Smart contracts, bitcoin bots, and consumer protection. Wash Lee Law Rev Online 71(2):35–50

Fanning K, Centers DP (2016) Blockchain and its coming impact on financial services. J Corp Account Finance 27(5):53–57

Frantz CK, Nowostawski M (2016) From institutions to code: towards automated generation of smart contracts. In: Proceedings of the IEEE international workshops on foundations and applications of self* systems, pp 210–215

Friedlmaier M, Tumasjan A, Welpe IM (2016) Disrupting industries with blockchain: the industry, venture capital funding, and regional distribution of blockchain ventures. In: Proceedings of the 51st Hawaii international conference on system sciences, Waikoloa, HI, pp 3517–3526

Fuggetta A, Di Nitto E (2014) Software process. In: Proceedings of the conference on future of software engineering, Hyderabad, India, pp 1–12

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. MIS Q 28(1):75–105

Idelberger F, Governatori G, Riveret R, Sartor G (2016) Evaluation of logic-based smart contracts for blockchain systems. In: Alferes J, Bertossi L, Governatori G, Fodor P, Roman D (eds) Rule technologies. Research, tools, and applications. RuleML 2016. Lecture notes in computer science, vol 9718. Springer, Cham, pp 167–183

IEEE (1995) 1074-1995—IEEE standard for developing software life cycle processes. IEEE. https://ieeexplore.ieee.org/document/490501/. Accessed 20 Mar 2018

Jacobson I, Booch G, Rumbaugh J, Rumbaugh J, Booch G (1999) The unified software development process, vol 1. Addison-Wesley, Reading

Johansen S (2018) A comprehensive literature review on the Blockchain as a technological enabler for innovation. Working paper, Mannheim University

KingOfTheEther (2016) Post-mortem investigation. https://www.kingoftheether.com/postmortem.html. Accessed 10 Apr 2018

Kiviat TI (2015) Beyond bitcoin: issues in regulating blockchain transactions. Duke Law J 65:569–608

Klöhn L, Parhofer N, Resas D (2018) Initial coin offerings (ICOs). Z Bankr Bankwirtsch 30(2):89–106

Koulu R (2016) Blockchains and online dispute resolution: smart contracts as an alternative to enforcement. SCRIPTed 13(1):40–69

Kruchten P (2004) The rational unified process: an introduction. Addison-Wesley Professional, Boston

Kuhrmann M, Diebold P, Münch J, Tell P, Garousi V, Felderer M, Trektere K, McCaffery F, Linssen O, Hanser E, Prause CR (2017) Hybrid software and system development in practice: waterfall, scrum, and beyond. In: Bendraou R, Raffo D, LiGuo H, Maggi FM (eds) Proceedings of the 2017 international conference on software and system process, Paris, France, pp 30–39

Lacity MC, Janson MA (1994) Understanding qualitative data: a framework for text analysis methods. J Manag Inf Syst 11(2):137–155

Lee G, Xia W (2010) Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. MIS Q 34(1):87–114

Leitner A, Ciupa I, Oriol M, Meyer B, Fiva A (2007) Contract driven development = test driven development—writing test cases. In: Crnkovic I, Bertolino A (eds) Proceedings of the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering (ESEC-FSE '07). New York, NY, USA, pp 425–434

Luu L, Chu D-H, Olickel H, Saxena P, Hobor A (2016) Making smart contracts smarter. In: Weippl E, Katzenbeisser S, Kruegel C, Myers A, Halevi S (eds) Proceedings of the 2016 ACM Sigsac conference on computer and communications security, pp 254–269

March ST, Smith GF (1995) Design and natural science research on information technology. Decis Support Syst 15(4):251–266
Marjanovic O, Milosevic Z (2001) Towards formal modeling of e-contracts. In: Proceedings of the fifth IEEE international conference on enterprise distributed object computing, pp 59–68
Mayring P (2014) Qualitative content analysis: theoretical foundation, basic procedures and software solution. Dissertation. http://nbn-resolving.de/urn:nbn:de:0168-ssoar-395173. Accessed 1 May 2018
Mens T, Guehéneuc Y-G, Fernández-Ramil J, D'Hondt M (2010) Guest editors' introduction: software evolution. IEEE Softw 27(4):22–25
Moyano JP, Ross O (2017) KYC optimization using distributed ledger technology. Bus Inf Syst Eng 59(6):411–423
Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/en/bitcoin-paper. Accessed 12 Aug 2017
Narayanan A, Clark J (2017) Bitcoin's academic pedigree: the concept of cryptocurrencies is built from forgotten ideas in research literature. ACM Queue 15(4):1–30
Notheisen B, Cholewa JB, Shanmugam AP (2017) Trading real-world assets on blockchain. Bus Inf Syst Eng 59(6):425–440
Pesch PJ, Sillaber C (2017) Distributed Ledger, Joint Control? – Blockchains and the GDPR's Transparency Requirements. Comput Law Rev Int 18(6):166–172. https://doi.org/10.9785/cri-2017-0602t
Porru S, Pinna A, Marchesi M, Tonelli R (2017) Blockchain-oriented software engineering: challenges and new directions. In: Uchitel S, Orso A, Robillard M (eds) Proceedings of the 39th international conference on software engineering companion, Buenos Aires, Argentina, pp 169–171
Prusty SK, Mohapatra PKJ, Mukherjee CK (2017) House of strategy: a model for designing strategies using stakeholders' opinion. Comput Ind Eng 108:39–56
Rajlich VT, Bennett KH (2000) A staged model for the software life cycle. Computer 33(7):66–71
Rückeshäuser N (2017) Do we really want blockchain-based accounting? Decentralized consensus as enabler of management override of internal controls. In: Leimeister JM, Brenner W (eds) Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017), St. Gallen, Switzerland, pp 16–30
Schwaber K, Beedle M (2002) Agile software development with scrum. Prentice Hall, Upper Saddle River
Seijas PL, Thompson SJ, McAdams D (2016) Scripting smart contracts for distributed ledger technology. https://eprint.iacr.org/2016/1156.pdf. Accessed 1 May 2018
Siegel D (2016) Understanding the DAO attack. https://www.coindesk.com/understanding-dao-hack-journalists/. Accessed 10 Apr 2018
Sillaber C, Waltl B (2017) The life cycle of smart contracts in blockchain ecosystems. Datenschutz Datensicherheit DuD 41(8):497–500
Stark J (2015) Product lifecycle management. Springer, London
Swan M (2015) Blockchain: blueprint for a new economy. O'Reilly Media, Sebastopol
Szabo N (1997) The idea of smart contracts. Nick Szabo's papers and concise tutorials. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html. Accessed 1 May 2018
Tapscott D, Tapscott A (2016) Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world. Penguin, New York
Treiblmaier H, Zeinzinger Z (2018) Understanding the blockchain through a gamified experience: a case study from Austria. In: 25th European conference on information systems, June 23–28, Portsmouth: UK
Tschorsch F, Scheuermann B (2016) Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun Surv Tutor (COMST) 18(3):2084–2123
Vidgen R, Wang X (2009) Coevolving systems and the organization of agile software development. Inf Syst Res 20(3):355–376
Wang H, Chen K, Xu D (2016) A maturity model for blockchain adoption. Financ Innov 2(12):1–5
Werbach K, Cornell N (2017) Contracts ex machina. Duke Law J 67(2):313–382
Wood G (2014) Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, pp 1–32
Xu X, Weber I, Staples M, Zhu L, Bosch J, Bass L, Pautasso C, Rimba P (2017) A taxonomy of blockchain-based systems for architecture design. In: Proceedings of the IEEE international conference on software architecture, Gothenburg, Sweden, pp 243–252
Zakarian A, Kusiak A (2001) Process analysis and reengineering. Comput Ind Eng 41(2):135–150

Affiliations

Christian Sillaber[1] · Bernhard Waltl[2] · Horst Treiblmaier[3] · Ulrich Gallersdörfer[2] · Michael Felderer[1,4]

Horst Treiblmaier
horst.treiblmaier@modul.ac.at

[1] University of Innsbruck, Innsbruck, Austria

[2] TU Munich, Munich, Germany

[3] MODUL University Vienna, Am Kahlenberg 1, 1190 Vienna, Austria

[4] Blekinge Institute of Technology, Karlskrona, Sweden

ISSN: 2281-1346

UNIVERSITÀ DI PAVIA
Department of Economics and Management

DEM Working Paper Series

Libra or Librae? Basket based stablecoins to mitigate foreign exchange volatility spillovers

Paolo Giudici
(Università di Pavia)

Thomas Leach
(Università di Pavia)

Paolo Pagnottoni
(Università di Pavia)

 # 183 (02-20)

Via San Felice, 5
I-27100 Pavia

economiaweb.unipv.it

Libra or Librae? Basket based stablecoins to mitigate foreign exchange volatility spillovers*

PAOLO GIUDICI, THOMAS LEACH and PAOLO PAGNOTTONI †

February 5, 2020

ABSTRACT

The paper aims to assess, from an empirical viewpoint, the advantages of a stablecoin whose value is derived from a basket of underlying currencies, against a stablecoin which is pegged to the value of one major currency, such as the dollar. To this aim, we first find the optimal weights of the currencies that can comprise our basket. We then employ volatility spillover decomposition methods to understand which foreign currency mostly drives the others. We then look at how the stability of either stablecoin is affected by currency shocks, by means of VAR models and impulse response functions. Our empirical findings show that our basket based stablecoin is less volatile than all single currencies. This results is fundamental for policy making, and especially for emerging markets with a high level of remittances: a librae (basket based stable coin) can preserve their value during turbolent times better than a libra (single currency based stable coin).

JEL classification: C01, C32, C58, G21, G32
Keywords: Cryptocurrencies; Fintech; Stablecoins; Spillover; Variance decomposition.

*

†University of Pavia, Via San Felice 5, 27100, Italy. ✉: paolo.giudici@unipv.it; thomas.leach01@universitadipavia.it; paolo.pagnottoni@unipv.it

I. Introduction

Carney (2019) posed the question of whether a Synthetic Hegemonic Currency (SHC) would be best provided by the public sector. The rationale would be that a global currency, underpinned by a basket of reserve assets, could better support global outcomes. For example, an SHC could dampen the dominating influence of the US dollar on global trade, it could alleviate spillovers to exchange rates from shocks to the US economy, and trade across countries could become less dependent on the dollar.

This is not the first time that the idea of global currency has been floated publicly. Cooper (1984) advocated for a *radical alternative scheme for the next century: the creation of a common currency for all of the industrial democracies, with a common monetary policy and a joint Bank of Issue to determine that monetary policy.* On the other hand, others have argued in favour of retaining major currencies but with a tighter exchange rate policy among them (Williamson, 1993; McKinnon et al., 1984). Or in maintaining the status quo as suggested by Rogoff (2001).

The revival of discussions concerning an SHC, have somewhat been sparked by the discourse surrounding central bank digital currency (CBDC) and stablecoins. In particular, Facebook announced plans for its own privately issued stablecoin that could emulate some of the characteristics of an SHC. The proposition is to construct a stablecoin that can circulate globally with a value that is derived from an underlying basket of assets comprised of the major currencies.[1] Whilst the exact composition of the underlying basket of assets is yet unspecified, the objective is to devise a digital currency whose exchange rate fluctuations are minimised against several currencies. These plans have been met with resistance from regulators and Facebook itself has repeatedly stated that the Libra stablecoin could be backed by a single currency (the dollar).

Why have regulators reacted with such caution to Facebook's plans to issue a stablecoin? Firstly, as a tech-giant Facebook can push Libra to its vast user-base, approximately 2.41 billion monthly active users.[2] To put this into perspective, currently it is estimated there around 40 million bitcoin wallets and 1 million daily users.[3] [4] Facebook would have to successfully penetrate 2% of its user base to match what is an upper bound on a proxy for the size of bitcoins user base, the most commonly used cryptoasset. Whilst the two assets may serve different purposes, there is potential for Facebook's Libra to rapidly acquire a significant user base transacting in a privately issued global digital currency. This may affect significantly, in particular, private individuals' transfers of money from remittances.

A remittance is a transfer of money made by a foreign worker to an individual in its home country. Remittances are one of the largest capital flows to developing countries. According to the World Bank, in 2018 overall global remittance grew 10% to 689 billion dollars, including 529 billion dollars to low income countries. India consecutively remains the top receiver of remittances, with 80 billion dollars in 2018 (about 3% of India's GDP), followed by China, the Philippines, Mexico and Nigeria.[5] While in the

[1] See https://libra.com

[2] https://newsroom.fb.com/company-info/

[3] The number of bitcoin wallets: https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users/

[4] The number of active wallets: https://coinmetrics.io/.

[5] https://www.knomad.org

past the remittance industry has been dominated by few financial players (such as Western Union), with a high transaction cost, recently many fintech startups (such as TransferWise) have entered the market with competitive offers, opening the door to the possible entrance of bigtechs such as Facebook, with its Libra project.

Against this background, we investigate the consequences of a global SHC ("Librae", in a literal sense), regardless of whether issued by a private company such as Facebook, or by a central bank. In particular, we compare the stability of an SHC to that of a single currency based currency ("Libra", in a literal sense). To this aim we first look at the optimal design of an SHC that is backed by a basket of underlying reference currencies, such as those included in the International Monetary fund Special Drawings Rights (SDRs), and compare the resulting volatility with that of single currencies, from 2002 onwards. We then study the currencies which mostly determine volatility spillovers among exchange rates, using the framework of Diebold and Yılmaz (2014). Based on the previous results, we then proceed to assess, by means of VAR based impulse response functions, the impact that shocks on the driving currencies would carry on the SHC or on single currencies, to understand which stablecoin design (Libra or Librae) better preserves the value of remittances from low income countries. For the optimal construction of a basket of currencies, we follow Hovanov et al. (2004) to compute a minimum variance currency basket using major currencies. We construct a reference basket that contains the Dollar (USD), the Euro (EUR), the Yen (JPY), the Renminbi (CNY) and the Pound Sterling (GBP), the currencies employed for the determination of the IMF's Special Drawing Rights (SDR) basket. The weights are determined applying Markowitz' portfolio allocation algorithm on daily data from January 2002 up until November 2019. We compare the obtained weights with those of the SDR. Our empirical findings show, in particular, that our basket puts less weight in the dollar, and more weight on the Euro and on the Renminbi.

By construction, our basket based currency should be the least varying in comparison to those contained in the basket, and our results confirm this. However, it is of interest to see how the SHC fares against currencies outside of the basket, for example against the currencies of the most important remittance markets. The comparison can answer a very important question, that is: is the exchange rate of the SHC less volatile then the exchange rate of the dollar and, consequently, of a dollar-based stablecoin? To answer this question, we recompute the currency invariant indices with the inclusion of additional currencies, namely the Indian Rupee, the Mexican Peso, the Philippine Peso ad the Nigerian Naira. Our empirical findings show that, overall, the SHC has the lowest volatility and, therefore, remittances converted in SHC best maintain their value. A basket based stablecoin, based on the IMF Special drawing rights, performs almost as well. A dollar based stablecoin, instead, performs worse, with the exception of during the crisis times.

Our volatility spillover decomposition shows that the dollar is the currency that has the largest impact on the others, especially in terms of exporting contagion. As a consequence, a shock on the dollar, expressed by a one standard deviation decrease in its normalised value with respect to the other currencies, causes a shock on all currencies and, through high order contagion, on the dollar itself, leading to a new lower equilibrium. Differently, a shock in the value of the SHC, caused by a shock of a currency in the basket, is offset by

the diversification effect and, therefore, the starting equlibrium is maintained. This implies that remittances converted in basket based stablecoin better maintain their value, with respect to those converted in dollars (or dollar based stable coins).

The rest of the paper is organised as follows, Section II contains a review of the relevant literature, Section III outlines our proposed methodologies, Section IV presents our data and the empirical findings, and finally in Section V we conclude.

II. Literature Review

A. *Stablecoins and e-money*

We take the definition of a 'stablecoin' to be a crypto-asset designed to maintain a stable value relative to another asset (typically a unit of currency or commodity) or a basket of assets (Financial Stability Board, 2019). Bullmann et al. (2019) make the following distinctions between types of stablecoins.

- **Tokenised funds** - denote stablecoins that are a claim on a pool of collateral that consists of funds, including cash, electronic money, commercial bank money or central bank reserve deposits e.g. Tether, Utility Settlement Coin
- **Off-Ledger Collateralised** - stablecoins that are a claim on a pool of collateral that is comprised of various assets e.g. multiple currencies, T-Bills etc
- **On-Ledger Collateralised** - stablecoins that are a claim on a pool of underlying collateral that is held on a blockchain e.g. Dai
- **Algorithmic** - take users expectations into account to stabilise the value of the coin (mostly conceptual) e.g. BasisCoin

At present, tokenised funds and off-ledger collateralised are the most common occurring instances of stablecoins. Libra, would fall into the later as the foundation has plans to invest the funds that are received in return for stablecoins. From herein, we work with two instances of a stablecoin, the Libra, in which single-currency stablecoins are issued in receipt of funds, this is essentially the tokenised funds model. The other instance, Librae, would have its own exchange rate that is backed by a basket of currencies, this would fall into the off-ledger collateralised category as the foundation intends to invest the funds across currencies and potentially in other interest yielding assets.

This is not the first time that electronic money has been on the agenda for central banks and policy makers, after a flurry of innovations in this space, in 1996 and 1998 respectively the BIS and ECB published reports addressing the regulation of e-money and the implications for monetary policy.[6] For various reasons, these forms of e-money never really troubled the concerns of policy makers of the time.[7]. However, discussions around digitised forms of money have reared their head once again.

[6] See European Central Bank (1998); Bank for International Settlements (1996)

[7] For example, see Levene (2006)

B. *Global currencies*

Keynes originally suggested the *bancor* as a unit of account of his proposed International Clearing Union, intended to fix to the dual dollar gold system. His plan for the international monetary system was put up against those of Dexter White. After ongoing negotiations between the United States and the United Kingdom the International Monetary Fund (IMF) was eventually established. The IMF then approved the Special Drawing Rights (SDRs) in 1967. The IMF's issuance of SDRs could be seen as a supranational currency issued by central banks, although the SDR does not fulfil all functions of money. Whilst serving as a store of value and unit of account, SDRs are only used by some central banks and international institutions as a means of exchange to pay each other (Ocampo, 2019). For this, they may not be strictly considered as a"true" global currency.

A boost to the importance of SDRs was given in 2009, when China called for reforms to the international monetary system by adopting the SDR as a reserve asset. Against these developments, Humpage (2009) suggests that while the adoption of the SDR as a reserve asset is technically feasible, it would not reduce the dollar's role any time soon. Many foreign-exchange transactions, even excluding US residents, are denominated and settled in dollars. Producers typically invoice their products in dollars, which keeps their prices in line with their competitors and simplifies cross-border price comparisons among producers (Gopinath et al., 2016).

Given the persistent importance of the US dollar, the question is whether this will remain so under the fintech transformation that is changing the financial world. And, in particular, whether a dollar based stable coin is more likely to be adopted than a basket based one.

C. *Remittances and exchange rates*

A stablecoin backed by a basket of currencies could be an attractive asset for foreign workers that make remittances to families in their home countries. In particular where its value is not directly tied to the domestic currency. Under the status quo, an appreciation in the value of the domestic currency can reduce the remittances ratio because workers want to to keep the additional earning from the appreciation of the currency. On the other hand, workers based in foreign countries, where the value of the domestic currency is declining, may remit money on an urgent basis.

These remittances also have an effect on the receiving countries. One specific challenge for countries that face large inflows of worker remittances could lead to the emergence of "Dutch disease," that is, remittance inflows could result in an appreciation of the equilibrium real exchange rate that would tend to undermine the international competitiveness of domestic production, particularly that of nontraditional exports. BARAJAS et al. (2011) note that reasonable modifications in the modelling of the factors driving remittances, or in the various macroeconomic roles that remittances may play, could moderate or even reverse the expected impact of remittance flows on the equilibrium value of the real exchange rate.

Acosta et al. (2009a) discuss two mechanisms by which this occurs, the first mechanism is demonstrated in the Salter-Swan-Conder-Dornbusch model, whcih points to a "spending effect," by which the increase

in wealth following higher capital inflows from remittances, combined with exogenous tradable prices, causes the prices of nontradable goods and services to rise. These higher prices lead to an expansion in the nontradable sector. By definition, an increase in the price of nontradables relative to the price of tradables translates into real exchange rate appreciation.

The second mechanism, proposed in Acosta et al. (2009b), is that remittances tend to increase household aggregate wealth. An increase in household wealth may lead to a decrease in labor supply as households substitute more leisure for work. A shrinking labor supply, in turn, puts upward pressure on wages. Rising wages raise production costs, and higher production costs can lead to a further contraction of the tradable sector. Both the resource reallocation effects and the labor effects can cause an appreciation of the exchange rate, thereby reducing the international competitiveness of the tradable sector, and may lead to tradable sector contraction, higher wages, and higher production costs.

A basket based currency could dampen some of these effects as it is less susceptible to appreciation and depreciation of the domestic and foreign currencies. However, the effects are likely to be ambiguous and depend on how the stablecoin is used. If it gains acceptability in the home currency this could leads to new episodes of dollarisation, whereas if the currency is only used as a medium of exchange the effect could be negligible.

D. Contribution of the paper

The paper combines the background of the previous streams of literature, namely: the need of a global currency, which is "optimal" in terms of minimum volatility (maximum stability), and resilient to exchange rate shocks; with the emergence of fintech technologies, and of blockchain based stable coins in particular.

Within this background, we contribute to the previous literature, from an economic viewpoint, by answering the following research question: is a basket based stable coin better than a single currency one, in terms of stability?

To answer the previous question, we contribute to the literature, from a methodological viewpoint, with three main innovations: i) we provide a methodology to build a minimum variance basket of currency, statistically deriving the optimal weights; ii) we provide a methodology aimed at assessing contagion spillovers among foreign exchange markets, based on Diebold and Yilmaz variance decomposition model; iii) we provide a VAR based methodology to build impulse response functions aimed at assessing the long run impact of a currency shock on both a basket based and a single currency based stable coin.

III. Methodology

In this section we outline the methodologies employed in our empirical application. Firstly, we describe the optimal control problem which yields to the optimal stablecoin weights. Secondly, we introduce our VAR model and, based on it, we study the spillover effects across the currencies in the basket to determine their interconnectedness and, therefore, to understand which are the most relevant ones in terms of

shock transmission. Thirdly, using again the proposed VAR model, we analyze the impact of shocks in the currencies within the basket on the other currencies and, consequently, on the stablecoins.

A. *Optimal control problem*

We aim to build a basket of predetermined (reference) currencies with optimal weights, namely, weights which minimize the variability of a basket based stablecoin. This translates into an optimal control problem which minimize the variance of the basket constructed with the above mentioned currencies.

Hovanov et al. (2004) show that the values of any given currency depends on the base currency chosen. The latter fact creates ambiguity in evaluating the currency itself and its dynamics. To overcome this issue,Hovanov et al. (2004) proposed a reduced (to the moment t_0) normalized value in exchange (RNVAL) of the i-th currency:

$$\mathrm{RNVAL}_i\,(t/t_0) = \frac{c_{ij}(t)}{\sqrt[n]{\prod_{k=1}^{n} c_{kj}(t)}} \Big/ \frac{c_{ij}\,(t_0)}{\sqrt[n]{\prod_{k=1}^{n} c_{kj}\,(t_0)}} = \sqrt[n]{\prod_{k=1}^{n} \frac{c_{ik}(t)}{c_{ik}\,(t_0)}} \tag{1}$$

By reducing to the moment t_0 and normalizing each currency observation by the geometric average of the other currencies at that specific point in time, the RNVAL allows the computation of a unique optimal, minimum variance currency basket, despite the base currency choice. The minimum variance currency basket is derived by searching the optimal weight vector w^* which solves the following optimal control problem:

$$\mathrm{Min}\left(S^2(w) = \sum_{i,j=1}^{n} w_i w_j \mathrm{cov}(i,j) = \sum_{i=1}^{n} w_i^2 s_i^2 + 2\sum_{i,j=1}^{n} w_i w_j \mathrm{cov}(i,j)\right) \tag{2}$$

subject to

$$\begin{cases} \sum_{i=1}^{n} w_i = 1 \\ \qquad w_i \geq 0 \end{cases}$$

The optimal control problem in Equation (2) yields to the minimum variance weights which enable us to construct the stablecoin value.

B. *VAR models and spillover analysis*

We evaluate spillovers through the methodology by Diebold and Yilmaz (2012). As in their seminal paper, we start from estimating a Vector AutoRegressive (VAR) model, that is :

$$x_t = \sum_{i=1}^{k} \Phi_i x_{t-i} + \varepsilon_t \tag{3}$$

where x_t being the $(n \times 1)$ vector of first differences in RNVALs at time t, Φ_i the $(n \times n)$ VAR parameter matrices, k the autoregressive order, ε_t a zero-mean white noise process having variance-covariance matrix Σ_ε, with n being the number of currencies considered in order to build the basket. Note that the VAR model is built on the variables' first differences, as this ensure the stationarity of the analyzed time series.

The VAR in Equation 3 may also be rewritten in its corresponding vector moving average (VMA) representation, that is

$$x_t = \varepsilon_t + \Psi_1 \varepsilon_{t-1} + \Psi_2 \varepsilon_{t-2} + \cdots \tag{4}$$

where $\Psi_1, \Psi_2, ...$ the $(n \times n)$ are the matrices of VMA coefficients. The VMA coefficients are recursively computed as $\Psi_i = \Phi_1 \Psi_{i-1} + \Phi_2 \Psi_{i-2} + ... + \Phi_i \Psi_1$, having $\Psi_i = 0 \; \forall i < 0$ and $\Psi_1 = I_n$.

As it is widely accepted in the financial econometric literature, the variance decomposition tools are used to evaluate the impact of shocks in one system variable on the others. Strictly speaking, variance decompositions decompose the H-step-ahead error variance in forecasting x_i which is due to shocks to x_j, $\forall j \neq i$ and $\forall i = 1, ..., n$.

In this paper we make use of the KPPS H-step-ahead forecast error variance decompositions, as Diebold and Yilmaz (2012) do. This is because we avoid imposing an a priori ordering exchange rates regarding the influence of shocks across the system variables, as popular techniques like the Cholesky identification scheme do. Indeed, the KPPS H-step-ahead forecast errors have are convenient as they are invariant with respect to the variable ordering.

As already stated, Diebold and Yilmaz (2012) found their methodology on the H-step ahead forecast error variance decomposition. Considering two generic variables x_i and x_j, they define the own variance shares as the proportion of the H-step ahead error variance in predicting x_i due to shocks in x_i itself, $\forall i = 1, ..., n$. On the other hand, the cross variance shares (spillovers) are defined as the H-step ahead error variance in forecasting x_i due to shocks in x_j, $\forall i = 1, \cdots, n$ with $j \neq i$.

In other words, denoting as $\theta_{ij}^g(H)$ the KPPS H-step forecast error variance decompositions, with $h = 1, \cdots, H$, we have:

$$\theta_{ij}^g(H) = \frac{\sigma_{jj}^{-1} \sum_{h=0}^{H-1} (e_i' \Psi_h \Sigma e_j)^2}{\sum_{h=0}^{H-1} (e_i' \Psi_h \Sigma \Psi_h' e_i)} \tag{5}$$

with σ_{jj} being the standard deviation of the innovation for equation j and e_i the selection vector, i.e. a vector having one as i^{th} element and zeros elsewhere. Intuitively, the own variance shares and cross variance shares (spillovers) measure the contribution of each variable to the forecast error variance of itself and the other variables in the system, respectively, thus giving a measure of the importance of each variable in predicting the others.

Note that the row sum of the generalized variance decomposition is not equal to 1, meaning $\sum_{h=0}^{H-1} \theta_{ij}^g(H) \neq$

1. Diebold and Yilmaz (2012) circumvent this problem by normalizing each entry of the variance decomposition matrix by its own row sum, i.e.

$$\tilde{\theta}_{ij}^{g}(H) = \frac{\theta_{ij}^{g}(H)}{\sum_{j=1}^{n} \theta_{ij}^{g}(H)} \tag{6}$$

This tackles the above mentioned issue and yields to $\sum_{j=1}^{n} \tilde{\theta}_{ij}^{g}(H) = 1$, and $\sum_{j,i=1}^{n} \tilde{\theta}_{ij}^{g}(H) = n$.

As a measure of the fraction of forecast error variance coming from spillovers, Diebold and Yilmaz (2012) define the total spillover index (TSI):

$$TSI(H) = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{n} \tilde{\theta}_{ij}^{g}(H)}{\sum_{j,i=1}^{n} \tilde{\theta}_{ij}^{g}(H)} \cdot 100 = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{n} \tilde{\theta}_{ij}^{g}(H)}{n} \cdot 100 \tag{7}$$

Moreover, we also make use of directional spillovers indexes (DSI) to measure, respectively through equations (8) and (9), the spillover from exchange i to all other exchanges J (cfr. Eq. 8) and the spillover from all exchanges J to exchange i (cfr. Eq. 9) as:

$$DSI_{J \leftarrow i}(H) = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{n} \tilde{\theta}_{ji}^{g}(H)}{\sum_{j,i=1}^{n} \tilde{\theta}_{ij}^{g}(H)} \cdot 100 \tag{8}$$

$$DSI_{i \leftarrow J}(H) = \frac{\sum_{\substack{j=1 \\ j \neq i}}^{n} \tilde{\theta}_{ij}^{g}(H)}{\sum_{j,i=1}^{n} \tilde{\theta}_{ij}^{g}(H)} \cdot 100 \tag{9}$$

Directional spillovers may be conceived as providing a decomposition of total spillovers into those coming from - or to - a particular variable. In other words, they measure the fraction of forecast error variance which comes from (or to) one of the variables included in the system - and, hence, the importance of the variable itself in forecasting the others. From the definitions of directional spillover indexes, it is natural to build a net contribution measure, impounded in the net spillover index (NSI) from market i to all other markets J, namely:

$$NSI_i(H) = DSI_{J \leftarrow i}(H) - DSI_{i \leftarrow J}(H) \tag{10}$$

Another very important metric to measure the difference between the gross shocks transmitted from market i to j and gross shocks transmitted from j to i is the net pairwise spillover (NPS), defined as:

$$PNS_{ij}(H) = \left(\frac{\tilde{\theta}^g_{ij}(H)}{\sum_{q=1}^{n} \tilde{\theta}^g_{iq}(H)} - \frac{\tilde{\theta}^g_{ji}(H)}{\sum_{q=1}^{n} \tilde{\theta}^g_{jq}(H)} \right) \cdot 100 \tag{11}$$

All the metrics discussed above are able to yield insights regarding the mechanisms of market exchange spillovers both from a system-wide and a net pairwise point of view. Furthermore, performing the analyses on rolling windows we are able to study the dynamics of spillover indexes over time.

C. *Impulse response functions*

To determine the impact of shocks on the stablecoins we start from estimating a Vector AutoRegressive (VAR) model as the one in Equation (3). Also in this case, the VAR model is built on the variables' first differences to make sure that the assumption regarding the stationarity of the analyzed time series is fulfilled.

From the VAR model in Equation 3, we are able to retrieve impulse response functions. In particular, we look at how negative 1-standard deviation shocks in one currency impact the dynamics of the other currencies in the basket and, thereby, the dynamics of the stablecoin. Finally, in order to determine whether shocks in one currency are permanent, we also evaluate cumulative impulse response functions.

IV. Data and empirical findings

A. *Data*

To test our proposal, we make use of historical data, according to a retrospective analysis. In particular, we use daily foreign exchange rate data over the period 1 January 2002 - 30 November 2019[8]. To build our optimal basket of currencies, we collect data relative to the foreign exchange pairs between the currencies that are included in the IMF's Special Drawings Rights: the US dollar, the Chinese Renmimbi, the Euro, the British pound and the Japanese Yen. According to our research assumption, we will assume that the obtained basket of currencies correspond to a stable coin which can be exchanged and compared with a single currency based stablecoin, for example based on the US dollar. This, in particular, for foreign individuals sending remittances to their home country.

To understand the relative convenience of remittants, to use a basket based coin rather than a dollar based one for example, we have also collected exchange rate data, again over the period 1 January 2002 - 30 November 2019[9], for the most important remittance market currencies (besides China's Renmimbi, already in the basket), namely the Indian Rupee, the Mexican Peso, The Philippines Peso, the Nigerian Naira. Moreover, for what concerns the volatility analysis, we divide the sample into subsets which define the pre-crisis period (2002-2008), crisis period (2009-2011) and post-crisis period (2012-2019).

[8]Data are obtained from *investing.com*

[9]Data are obtained from *investing.com*

Currency	USD	CNY	EUR	GBP	JPY
Optimal Weights	0.21	0.14	0.21	0.21	0.23
IMF SDR Weights	0.42	0.11	0.31	0.08	0.08

Table I
Weights of the currency in the chose basket, according to our methodology (Optimal) and the IMF Special Drawing Rights (IMF SDR)

Finally, for the sake of comparison with a widely known basket-based currency such as the IMF SDR, we also collect data relative to the foreign exchange pair of the dollar with the IMF Special Drawing Rights.

Daily foreign exchange data are then used to compute the reduced normalized values, as illustrated in Section III. In this way, it is possible to analyze the dynamics of exchange rates without imposing any choice of base currency.

B. Optimal basket and stability analysis

First of all, we compute the RNVALs as described in Section III.

The resulting weights are contained, together with those of the IMF SDR, in Table I.

From Table I note that our method yields weights which are quite similar among each other, with the exception of the Chinese Renmimbi. The weights are quite different from the weights of the IMF SDR, which are highly concentrated on the USD dollar. The low weight of the Chinese Renmimbi in our basket can be explained by the fact that the Chinese currency roughly replicates the behaviour of USD. Indeed, in the considered period, it is pegged to it for most of the sample period, although with higher volatility. This makes our method to select a higher quantity of USD rather than CNY, being the former less volatile than the latter. Note also that our method selects a slightly higher portion of JPY compared to the other currencies. This is arguably due to the fact that JPY is the one which is least synchronized with the other currencies in the basket and, therefore, exerts an important diversification effect by reducing the overall volatility of the basket.

To better interpret the results, Figure 1 represents the time series of the Reduced Normalised Values of all considered currencies in the basket, along with our basket based stable coin, in the considered period.

Figure 1 shows the evolution of the RNVALs of the currencies composing the basket during the whole sample period.

From Figure 1 note that, after a first period of small turbulences, the time series start to diverge roughly from the beginning of 2006 onwards. From that point in time onwards, two clusters seem to emerge from the graph: the first one includes USD and CNY, while the second one pertains EUR, GBP and JPY. This is arguably due to the fact that, for many years, the CNY value was pegged to the dollar and, therefore, its dynamics over time shows quite similar patterns to that of the USD. Note that, as expected by construction, the Reduced Normalised Value of the basket based stable coin lies in the middle, ”mediating” between the

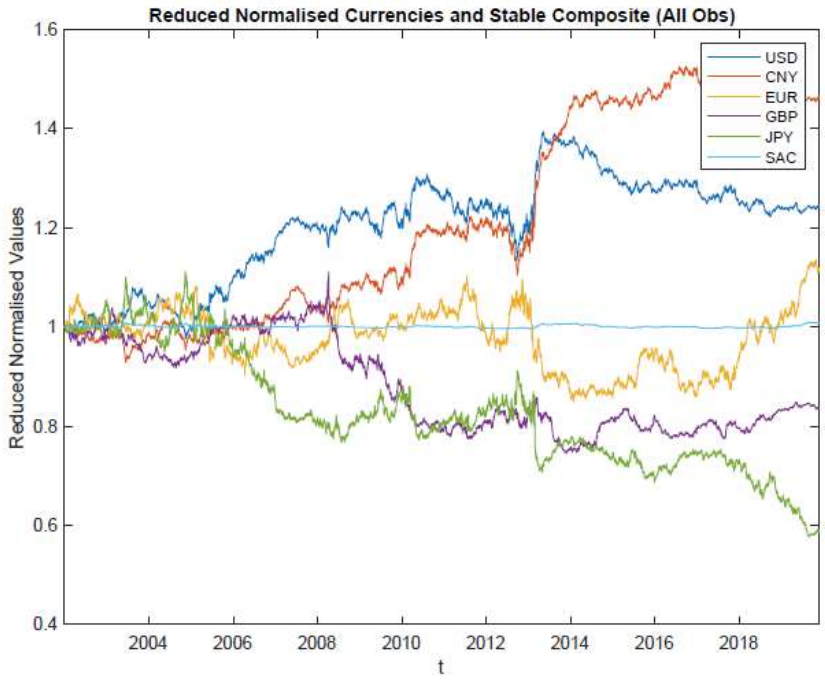


Figure 1
Time evolution of the Reduced Normalised VALue of the basket currencies (USD, CNY, EUR,GBP, JPY), and of the basket based stable coin (SAC)

different currencies, and compensating single deviations with diversification benefits.

For the sake of analyzing the world's emerging market currencies with the highest portions of remittances, we recompute the RNVALs including them. The corresponding graphical representation is contained in Figure 2. In the figure we have included, besides our basket based stable coin, another one that employs the same weights as the Special Drawing Rights.

Figure 2 is consistent with Figure 1, with USD and CNY showing similar patterns over time. All the other currencies seems to belong to another cluster, in the sense that they do not follow an upward trend as the previous ones, but rather fluctuate below the value of 1, with different patterns. The only exception is the Indian rupee (INR), whose value grows over time, although not with the same magnitude as USD and CNY do. Note that both basket based stable coins lie in the middle, similarly as in Figure 1, although their Reduced Normalised value fluctuates. This because the baskets are built using only five currencies, but are normalised with respect to all nine included in Figure 2.

To understand more precisely which stable coin is more stable (Libra: single currency based, or Librae: basket based), Table II presents their volatilities, measured by. their standard deviations, in the considered time period. The table presents also the correlations between the currencies, which help the interpretation of the results.

Table II shows, as far as correlations are concerned, that USD and CNY exhibit relatively strong negative correlations with all others currencies in the basket, but positive between themselves, consistently with what

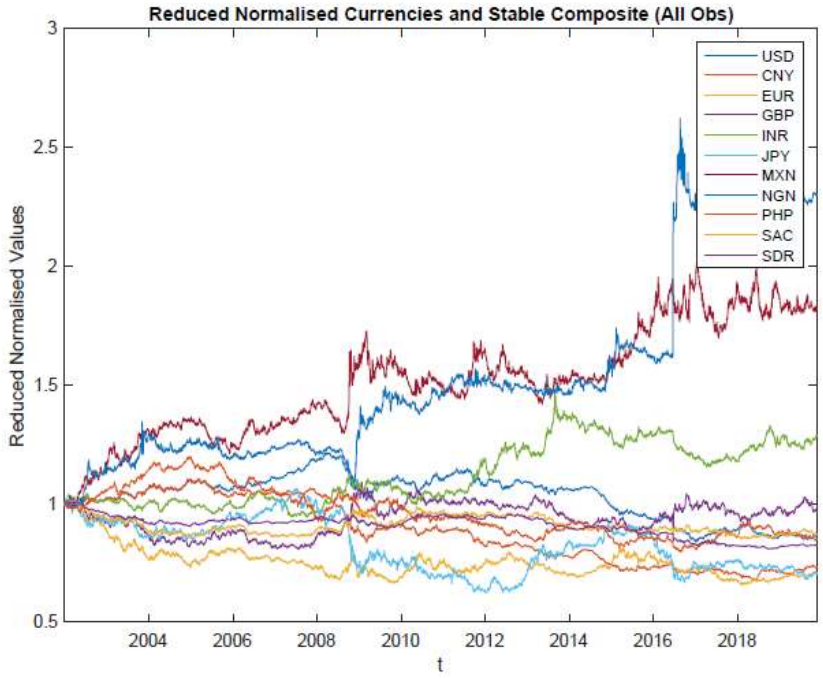


Figure 2
Time evolution of the Reduced Normalised VALue of the basket currencies (USD, CNY, EUR,GBP, JPY), of the considered emerging market currencies (INR, MXN, NGN, PHP) and of the basket based stable coins (SAC, SDR)

	USD	CNY	EUR	GBP	JPY	SAC
USD	1	0.79	-0.48	-0.76	-0.86	0.023
CNY	0.78	1	-0.45	-0.83	-0.86	0.012
EUR	-0.48	-0.45	1	0.2	0.24	0.04
GBP	-0.76	-0.83	0.22	1	0.66	0.027
JPY	-0.86	-0.86	0.24	0.66	1	0.02
SAC	0.02	0.01	0.039	0.027	0.02	1
σ	0.11	0.2	0.06	0.09	0.12	0.003

Table II
Volatility and Correlations between the RNVALs of the basket currencies, and the optimal basket based stable coin.

	USD	CNY	EUR	GBP	INR	JPY	MXN	NGN	PHP	SAC	SDR
σ_{all}	0.09	0.14	0.07	0.06	0.13	0.11	0.22	0.41	0.10	0.04	0.05
σ_{pre}	0.05	0.02	0.04	0.03	0.04	0.07	0.11	0.35	0.03	0.01	0.03
σ_{cri}	0.02	0.04	0.03	0.03	0.12	0.07	0.05	0.04	0.03	0.02	0.02
σ_{post}	0.05	0.06	0.08	0.07	0.04	0.08	0.15	0.09	0.07	0.03	0.02

Table III
Volatility of the RNVALs of the basket currencies, of the emerging market currencies, and of the two basket based stable coins, over the whole period (all), the pre-crisis period (pre), the crisis period (cri) and the post-crisis period (post).

observed in Figure 1. Moreover, one can clearly notice that the EUR acts as a good diversifier, as its pairwise correlations are quite low if compared to those between other currencies. More importantly, from the correlation matrix we can deduce that the stablecoin shows correlations with the other currencies whose values are very close to zero. Low correlations with the other currencies is a clear sign of the goodness of our stablecoin in being isolated with respect to the fiat currencies' dynamics and, therefore, arguably stable. In terms of volatility, the standard deviations show that the most volatile currency is CNY, followed by JPY and USD. Our stablecoin exhibits a standard deviation magnitude which is much lower than those of the other currencies and about ten times lower than that of the least volatile one, namely EUR. This is a clear sign of stability of the proposed stablecoin, as opposed to an hypothetical stablecoin pegged to one single currency.

To determine whether a basket-based stable coin would be a more valuable and more stable alternative than a stablecoin pegged to a single currency, especially for remittances, we can, in analogy with 2, compare the volatility of our stablecoin with that of a SDR based basket, and with the currencies of the most important emerging markets in terms of remittances. Table III contains the comparison, over the whole period and also in three distinct periods, corresponding to the pre-crisis period, the crisis period and the post-crisis period.

From Table III first row, the stablecoin exhibits lower values of volatility, when compared to the other traditional fiat currencies. The other rows in the Table that is always the case, with the exception of the crisis period, in which the USD has a comparable volatility. Indeed, the sovereign crisis in the Euro zone played a role in the devaluation of the currencies pertaining the Euro area. This causes a relatively higher instability of the SAC when compared to the USD. As a consequence, the SDR, whose value is mostly determined by the USD, shows a low volatility as well. For the same reason, and for the persistence of the effects caused by the crisis, the low volatility of the SDR is confirmed during the post-crisis period. Overall, the proposed stablecoin lower volatilities if compared to the single currencies in the basket and to the single emerging market currencies. This can be read as a strength of our stablecoin, as it could function as a better medium of exchange than a country's single currency, in particular as far as remittances are concerned. Note also that the SDR is a valid alternative to our stable coin, possibly easier to implement, from a political consensus viewpoint.

	USD	CNY	EUR	GBP	JPY	FROM
USD	44.94	35.33	13.02	6.67	0.04	11.01
CNY	34.49	49.40	10.76	5.34	0.00	10.12
EUR	15.81	15.22	62.29	6.48	0.19	7.54
GBP	11.4	10.21	6.28	69.58	2.53	6.08
JPY	0.41	0.14	0.01	3.94	95.51	0.90
TO	12.42	12.18	6.01	4.49	0.55	35.66

Table IV
Spillover table

C. Spillover analysis

We now consider spillovers between exchanges, to evaluate the price change connectedness of the currencies that compose the basket, and to understad which is the relative importance of each of the currencies in transmitting shocks. In this way, we are also able to determine which currencies potentially cause strong (or weak) price changes in our proposed stablecoin value.

As far as specifications are concerned, VAR models are built on price changes in reduced normalized values (RNVALs). We use a VAR lag determined by a Bayes-Schwarz information criterion (BIC) that penalizes overparametrization compared to other widely employed information criteria. The optimal number of lag determined by the BIC is 1. We use a $H = 100$ step-ahead forecast horizons for forward iteration of the system. Additionally, dynamic spillovers use a rolling estimation window of length 100 observations.

Firstly, we provide an analysis of unconditional price change spillovers, that are spillovers evaluated on the whole sample period. The results are shown in Table IV.

From Table IV note that there are two currencies which are highly interconnected with the others, meaning USD and CNY, whereas EUR, GBP and in particular JPY are more isolated in terms of return connectedness. Furthermore, the scene appears to be dominated by USD and CNY, whose contributions in terms of price change spillovers towards other currencies are much higher than those of the remaining currencies in the basket.

The analysis of dynamic spillovers is able to clarify the results obtained in the unconditional spillover analysis by means of observing the evolution of spillovers over time. Figure 3 shows the results.

Figure 3 depicts the overall dynamic spillover plotted over the sample period. The overall spillover within the basket ranges from a minimum of 17.87% to a maximum of 80.00%. It seems that the overall spillover follows a generally decreasing trend, as it starts from 54.51% at the beginning of the sample period, while it diminishes to 34.43% at the end of the studied time frame.

Dynamic directional spillovers can shed light on which of the currencies transmit price change spillovers to others and which of them receive price change spillovers from others. We plot from, to, net and pairwise spillovers in Figures 4, 5 and 6, respectively.

From the joint analysis of Figures 4, 5, 6 and **??** we can highlight that that USD is the most influential

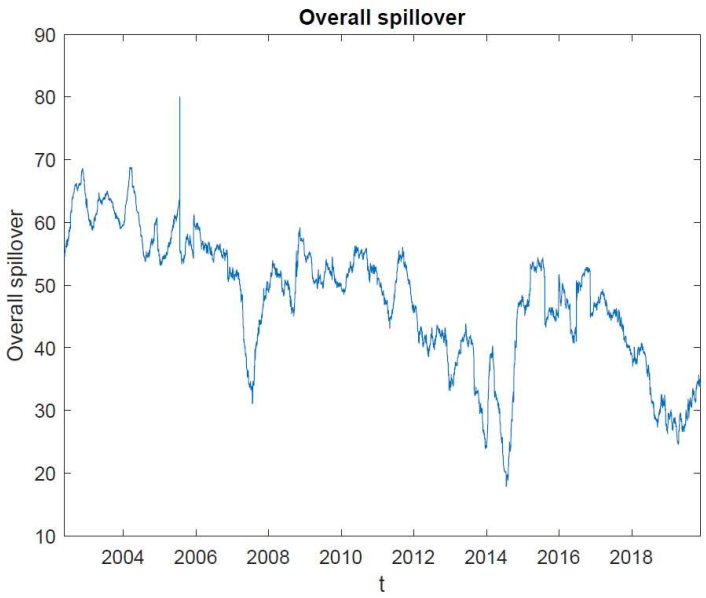


Figure 3
Overall spillovers

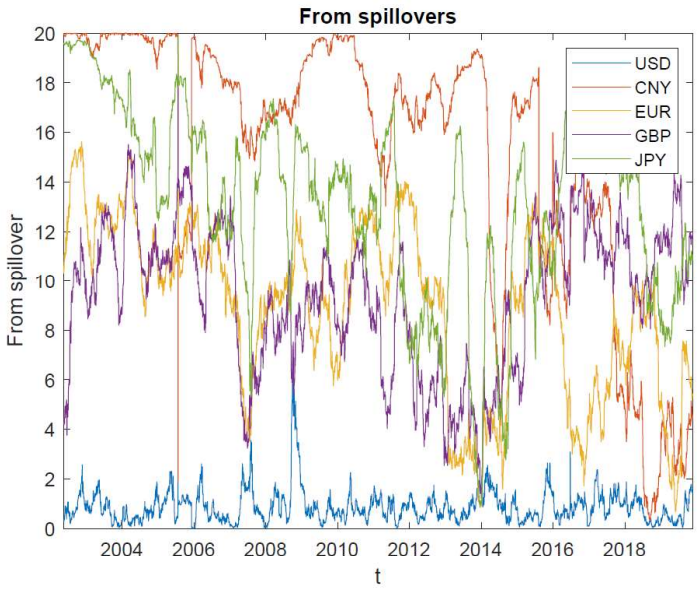


Figure 4
From spillovers

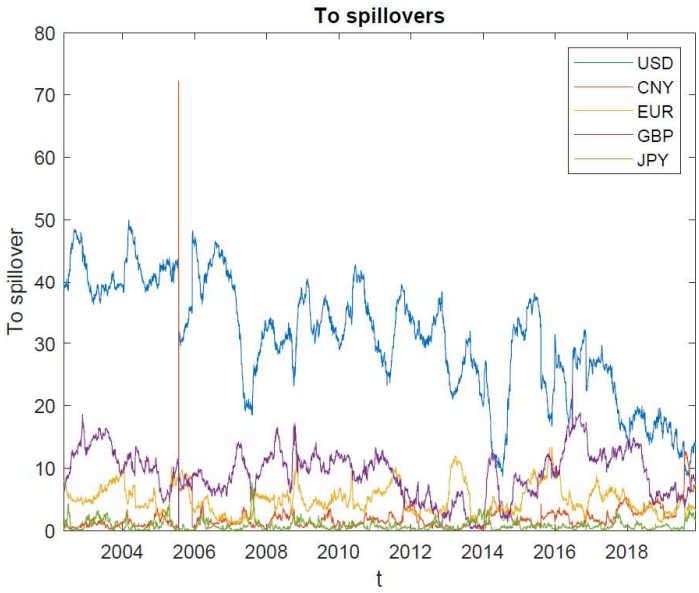


Figure 5
To spillovers

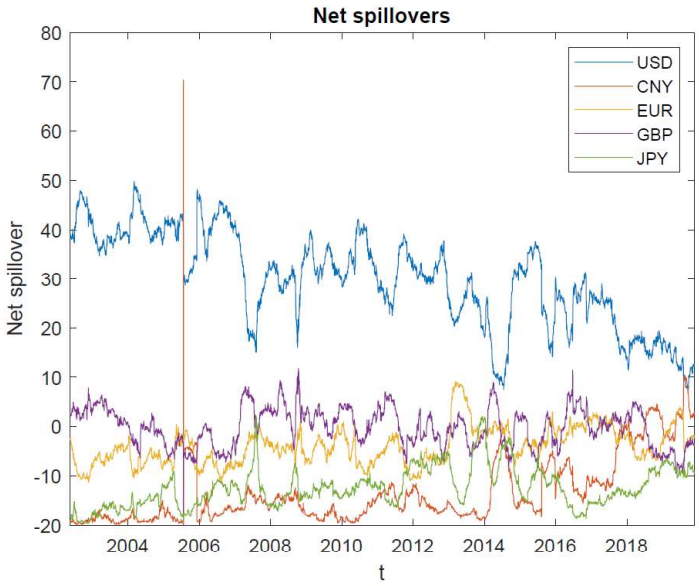


Figure 6
Net spillovers

currencies in terms of return spillovers. Indeed, the magnitude of spillovers received from others is weak compared to that transmitted to others. Moreover, the net spillover dynamics summarizes the dominant position of the USD, being it always positive and taking relatively high values over the sample period. However, the magnitude of spillovers transmitted by USD follows a negative trend over time, meaning the currency is gradually losing its potentiality to contribute to the evolution of the others, perhaps due to the affirmation of emerging economies in the latter period, especially after the 2009 crisis. Despite that, the latter considerations are in line with the full sample results obtained above, which point to the dominance of USD as a spillover transmitting currency.

Differently from what emerged in the full sample analysis, instead, the dynamic analysis shows that CNY is not such a leading currency in transmitting price change shocks. Indeed, the full sample result is arguably driven by a noticeable spike which occurred on 21 July 2005. Indeed, during that day the Chinese Central Bank officially announced the abandonment of the eleven-year-old peg to the dollar and pegged the CNY to a basket of currencies whose composition was not disclosed. This caused a prompt revaluation to CNY 8.11 per USD, as well as to 10.07 CNY per euro. However, the peg to the dollar was reinstituted as the financial crisis strengthened in July 2008. These results indicate that CNY does not particularly contribute to the price change evolution of the other currencies in the basket, although it can exert shocks through sudden policy decisions.

D. Impulse Response to currency shocks

We now apply the impulse response function tools to analyze the impact of currency shocks on the basket. We plot the impulse response functions for some currencies in Figure 7. Specifically, leveraging the results from the previous subsection, we consider USD as shock source, being the most important transmittant of spillovers. In addition, we consider EUR being the second most important residence of foreign remittants. Besides the two single currencies, we consider the to basket based stable coins, the optimal and the SDR one. Figure 7 shows the results of the impulse response analysis.

From Figure 7, we can see the impact of a negative 1-standard deviation shock in the USD on the different currencies. We can clearly notice that the impact of a USD shock is much higher on the dollar itself than on the stablecoin, being the magnitude of the impact lower. This is true also when looking at an IMF SDR versusa stablecoin shock. Moreover, the impact of the latter shock has opposite direction with respect to that on the USD. Given the correlation structure among the currencies, the stablecoin is indeed positively affected by a negative one standard deviation shock in the USD. This yields to the conclusion that a basket-based currency is less influenced by currency shocks than single currencies themselves.

With the aim of evaluating the persistence of currency shocks on the stablecoin value, we plot the cumulative impulse response functions for relevant currencies in Figure 8.

Figure 8 shows that a shock in the USD translates into a permanent effect on the USD itself, on the stablecoin and on the SDR as well. However, the magnitude of the permanent impact is way lower on the stablecoin than in the other two currencies. The same is true when comparing the permanent effects of

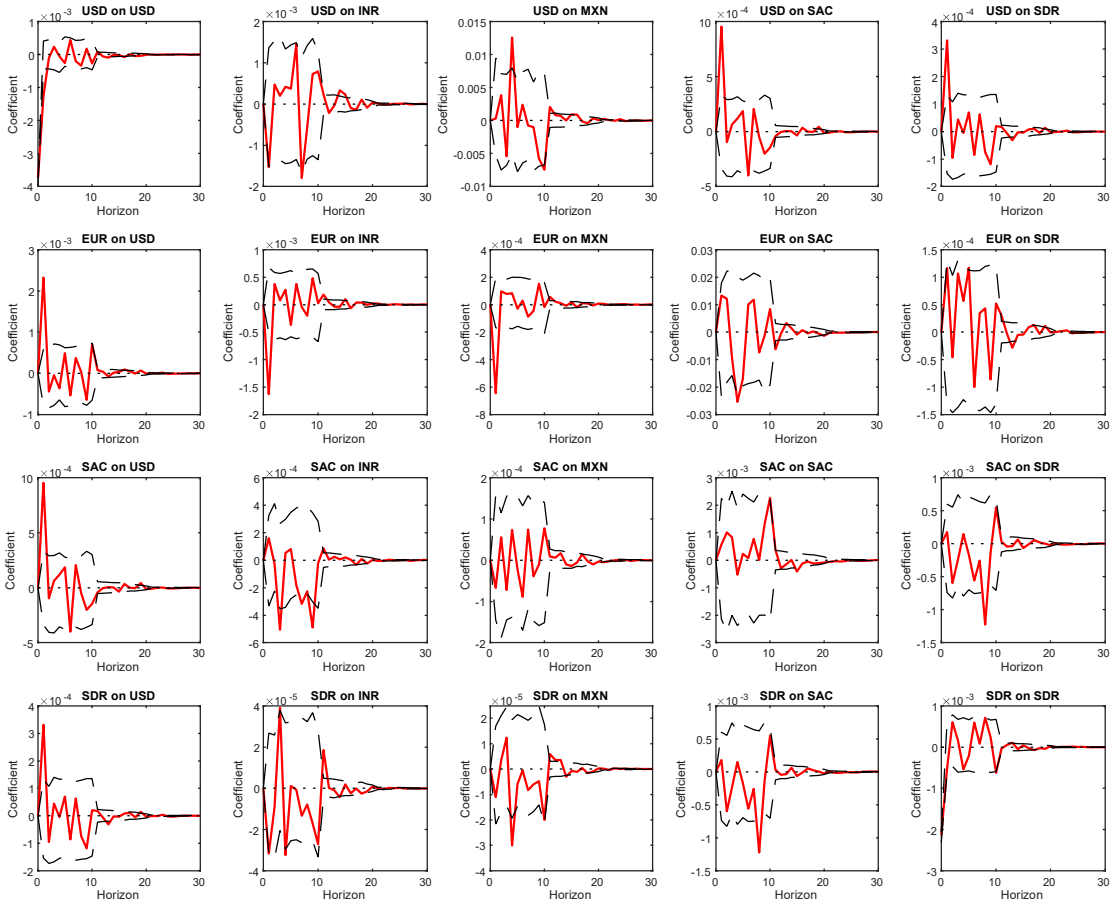


Figure 7
Impulse response functions

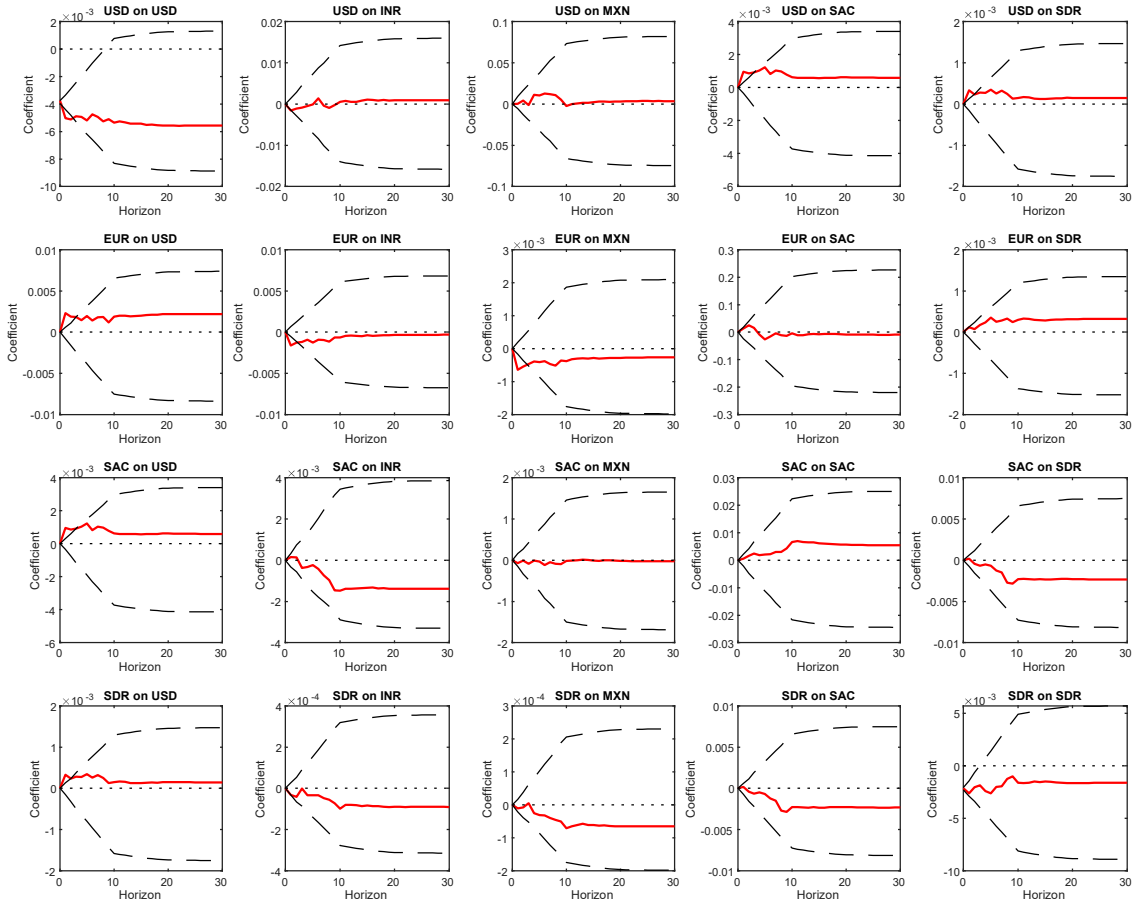


Figure 8
Cumulative impulse response functions

shocks in the EUR on itself, the stablecoin and the SDR. This suggests that single currencies are more prone to be permanently impacted from shocks than basket-based ones, especially if compared to the proposed stablecoin.

V. Conclusion

In the paper we present a methodology to build a basket based stable coin whose weights can maximise stability over a long time period. The weights have been calculated, retrospectively, for the period that follows 2002, and show a distribution more even than the IMF Special Drawing Rights weights.

The proposed stable coin (Librae) appears to be less volatile than single currencies and, therefore, with respect to single currency stable coins (Libra). It can thus constitute a valuable proposal especially for workers who live abroad and make remittances to their own country, a market segment with a high potential of being attracted by payments in stablecoins.

We have also proposed a variance decomposition technique, and an impulse response function analysis, both based on a VAR model, aimed at showing which currencies mostly impact the Foreign Exchange market and whether a single currency or a basket based stablecoin is more resilient to currency shocks. Our results show that the dollar is the currency which mostly impact the market, and that a basket based coin is better than a dollar based one, from a stability and value maintenance viewpoint.

With a basket based stablecoin it is possible to offset the risk of currencies shocks. This is of relevance for different policy purposes and, in particular, for emerging markets and countries having high remittances. Indeed, by holding stablecoins rather than single currencies the risks associated to currency shocks are mitigated and stablecoins holder can count on a currency whose value is less volatile than traditional fiat currencies and, thereby, more reliable. The latter fact has also positive consequences on cross-border payments side, provided that the stability of the stablecoin mitigates the foreign exchange risk, thus contributing to the fact that buyers and sellers give or receive an amount of money whose value is less sensitive to variations over time.

Future research may consider basket that dynamically evolve over time ("AI baskets"), although these are bound to be more difficult to achieve consensus. Furthermore, currency volumes in circulation may be taken to account, along with the technical characteristics of the coins (for example: cybersecurity, redeemability, reliability), from a different, more theoretical, viewpoint.

VI. Acknowledgements

The Authors acknowledge useful discussions and feedback during the IFABS conference in Medellin (Columbia) in which a preliminary version of the methodology contained in this paper was presented. They also acknowledge discussions with Guido Ascari and John Kiff. The work in the paper has received support from the European Union's Horizon 2020 training and innovation programme "FIN-TECH", under the grant

agreement No. 825215 (Topic ICT-35-2018, Type of actions: CSA).

References

Acosta PA, Baerg NR, Mandelman FS. 2009a. Financial development, remittances, and real exchange rate appreciation. *Economic Review* **94**.

Acosta PA, Lartey EK, Mandelman FS. 2009b. Remittances and the dutch disease. *Journal of international economics* **79**: 102–116.

Bank for International Settlements. 1996. Implications for central banks of the development of electronic money.

BARAJAS A, CHAMI R, HAKURA D, MONTIEL P, Tressel T. 2011. Workers' remittances and the equilibrium real exchange rate: Theory and evidence [with comment]. *Economía* **11**: 45–99. ISSN 15297470.
URL `http://www.jstor.org/stable/41343450`

Bullmann D, Klemm J, Pinna A, et al. 2019. In search for stability in crypto-assets: are stablecoins the solution? Technical report, European Central Bank.

Carney M. 2019. The growing challenges for monetary policy in the current international monetary and financial system. In *Remarks at the Jackson Hole Symposium*.

Cooper RN. 1984. A monetary system for the future. *Foreign Aff.* **63**: 166.

Diebold FX, Yilmaz K. 2012. Better to give than to receive: Predictive directional measurement of volatility spillovers. *International Journal of Forecasting* **28**: 57–66.

Diebold FX, Yılmaz K. 2014. On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics* **182**: 119–134.

European Central Bank. 1998. Report on electronic money.

Financial Stability Board. 2019. Regulatory issues of stablecoins.

Gopinath G, Boz E, Casas C, Díez FJ, Gourinchas PO, Plagborg-Møller M. 2016. Dominant currency paradigm. Working Paper 22943, National Bureau of Economic Research.
URL `http://www.nber.org/papers/w22943`

Hovanov NV, Kolari JW, Sokolov MV. 2004. Computing currency invariant indices with an application to minimum variance currency baskets. *Journal of Economic Dynamics and Control* **28**: 1481–1504.

Humpage O. 2009. Will special drawing rights supplant the dollar? `https://voxeu.org/article/sdr-vs-what-it-takes-be-international-reserve-currency`.

Levene T. 2006. 'Cashless society' card that flopped. *The Guardian* .
URL https://www.theguardian.com/money/2006/apr/15/moneysupplement1

McKinnon RI, et al. 1984. An international standard for monetary stabilization .

Ocampo JA. 2019. Is it time for a 'true global currency'? https://www.weforum.org/agenda/2019/04/is-it-time-for-a-true-global-currency.

Rogoff K. 2001. Why not a global currency? *American Economic Review* **91**: 243–247.

Williamson J. 1993. Exchange rate management. *The Economic Journal* **103**: 188–197.

About UCL CBT

The UCL CBT is the first centre globally to actively focus on blockchain-related research on the adoption and integration of Blockchain and Distributed Ledger Technologies into our socio-economic system.

The unique characteristics of the CBT at UCL provides a cross-sectoral platform connecting expertise and drawing knowledge from eight UCL departments centrally in one place. The CBT is a centre of excellence fostering open dialogue between industry players and sharing expertise and resources. It is a neutral think tank providing consultancy services to industry members, dedicated knowledge-transfer activities and cutting-edge in-house solutions.

For engagement outside of the academic world, the CBT's activities have been tailored to industry and policymakers' needs. The UCL CBT draws on its world-leading academic expertise to produce blockchain solutions for industry, start-ups and regulators. With a community of over 180 Research & Industry Associates and Industry Partners, it is the largest Academic Blockchain Centre in the world.

Notable Work

- The CBT released a report on the current adoption of DLT in global physical supply chains. The report featured an analysis of over 100 different projects taking place all over the world in the Grocery, Pharmaceutical and Fashion industries. Access the report here.
- The CBT is leading the Blockchain Technology for Algorithmic Regulation and Compliance (BARAC) project. This is the largest publicly funded blockchain project aimed at the public sector that will be defining feasibility guidelines to policymakers, industry and regulators by identifying problems and associated solutions with a bottom-up approach, built through case studies and proof of concept platforms. For this project, the CBT is partnering with the Financial Conduct Authority and the Singapore Monetary Authority and financial groups and Fintech companies like Banco Santander and R3.
- The CBT is a founding member of the Covid Task Force alongside The International Association for Trusted Blockchain Applications (INATBA) and the European Commission. The task force is convening key players in the global blockchain ecosystem to identify deployable technology solutions that address governmental, social, and commercial challenges caused by COVID. As well as identifying solutions, the Task Force will work to expedite their deployment.
- The CBT successfully funded nine research proposals that investigated topics including stable coin policy, smart contract innovation, blockchain economics and blockchain governance models. Research teams who were funded were made up of individuals from a variety of academic and industry organisations. Learn more about the projects here.
- The CBT launched the Block-Sprint hackathon to promote DLT innovation in the financial services sector. Over 160 individuals took part in the 2019 edition forming teams made up of industry practitioners, academics, and students. Learn about the winners and innovate ideas that were generated in the hackathon here.

About the Discussion Paper Series

The *UCL CBT Discussion Paper is* published on a quarterly basis featuring the latest developments in the blockchain and DLT space. The aim of the discussion paper series is to share recent developments and state-of-the-art solutions on blockchain and DLT of researchers from an interdisciplinary background with the CBT community. All accepted submissions are available in the CBT paper database.

The submissions are circulated among the members of the UCL CBT Editorial Board, led by the Scientific Director so that the results of the research receive prompt and thorough professional scrutiny.

If you are interested in submitting a paper to be included in forthcoming editions, please visit our website here to see what the latest theme and criteria for submission are.

UCL Centre for Blockchain Technologies

http://blockchain.cs.ucl.ac.uk/

UCL Computer Science
Malet Place
London WC1E 6BT
United Kingdom