UCL Centre for Blockchain Technologies

Discussion Paper Series

Q3 2020

Print Edition

Foreword

It is with great pleasure to present this new edition of our discussion paper series, the fourth so far and third of this year.

This time we publish four papers in the print edition and a further two are available online only. We received excellent submissions, and the choice between the print edition and online publication has been challenging, and all the papers deserve significant attention.

We hope our publication of these papers will provide a publicity boost that will increase their impact. These papers cover extremely 'hot' and important topics from technology governance, to taxes and adoption. These are all topics at the centre of the UCL CBT's research activities that we support.

Enjoy your reading

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

November 2020

Acknowledgement

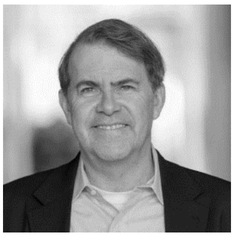
The editorial board wishes to thank Nikhil Vadgama for his high-quality work and strong perseverance, without which this discussion paper series would have never been possible.

Discussion Paper Series Contents

The following are published in this edition:

The following are published online only:

Editorial Board



Discussion Paper #1

An Analysis of Blockchain Adoption in Supply Chains Between 2010 and 2020

Nikhil Vadgama, Paolo Tasca

Centre for Blockchain Technologies

University College London

(United Kingdom)

This version: 30th September 2020

Abstract

In this research, the evolution of Distributed Ledger Technology (DLT) in supply chains has been mapped from the inception of the technology until June 2020, utilising primarily public data sources. Two hundred seventy-one blockchain projects operating in the supply chain have been analysed on parameters such as their inception dates, types of blockchain, stages reached, sectors applied to and type of organisation that founded the project. We confirm generally understood trends in the blockchain market with the creation of projects following the general hype and funding levels in the industry. We observe most activity in the Agriculture/Grocery sector and the Freight/Logistics sector. We see the shift of market interest from primarily private companies (startups) to public companies and consortia and the change in blockchain adoption from Ethereum to Hyperledger. Finally, we observe higher success and lower failure rates for Hyperledger-based projects in comparison to Ethereum-based projects.

Keywords: Blockchain, Supply Chain, Distributed Ledger Technology, Ethereum, Hyperledger, Agriculture, Freight, Logistics

An Analysis of Blockchain Adoption in Supply Chains Between 2010 and 2020

Nikhil Vadgama[1], Paolo Tasca

Centre for Blockchain Technologies
University College London
(United Kingdom)

This version: 30th September 2020

Abstract

In this research, the evolution of Distributed Ledger Technology (DLT) in supply chains has been mapped from the inception of the technology until June 2020, utilising primarily public data sources. Two hundred seventy-one blockchain projects operating in the supply chain have been analysed on parameters such as their inception dates, types of blockchain, stages reached, sectors applied to and type of organisation that founded the project. We confirm generally understood trends in the blockchain market with the creation of projects following the general hype and funding levels in the industry. We observe most activity in the Agriculture/Grocery sector and the Freight/Logistics sector. We see the shift of market interest from primarily private companies (startups) to public companies and consortia and the change in blockchain adoption from Ethereum to Hyperledger. Finally, we observe higher success and lower failure rates for Hyperledger-based projects in comparison to Ethereum-based projects.

Keywords: Blockchain, Supply Chain, Distributed Ledger Technology, Ethereum, Hyperledger, Agriculture, Freight, Logistics

1 Introduction

Distributed Ledger Technology (DLT) promises to disrupt business models, business processes, and aspects of society by creating information systems that are transparent and provide a single point of truth for all members of a network (Pilkington, 2016). As an electronic ledger that has the properties of decentralisation, immutability, cryptography and smart contracts, DLT represents an innovation beyond traditional database technology (Iansiti & Lakhani, 2017).

Since the creation of DLT with its beginnings with Bitcoin in 2008 (Nakamoto, 2008), DLT's impact has begun to move outside of just the financial services domain into other sectors, including that of supply chains (Bünger, 2017; Hughes, Park, Archer-Brown, & Kietzmann, 2019). Within the supply chain domain, it is widely acknowledged by many industry experts that DLT will have a tremendous impact on it, particularly around bringing transparency across various parts of it (O'Marah, 2017; Casey & Wong, 2017).

Supply chains underpin the smooth and timely movement of goods from producer through to the consumer, and with increasing globalisation, this coordination of goods underpins the globalised economy. The supply chain management sector stands at a size of $16 trillion and has large overhead with regards to fraud, errors and administration costs (Boucher, 2017).

[1] Correspondence to Nikhil Vadgama: nikhil.vadgama@ucl.ac.uk

Given the importance of this part of the world economy, it is astonishing that there is still a large degree of manual procedures and processes in operationally complex undertakings. For example, the shipment of refrigerated goods between East Africa and Europe can incorporate as many as 30 different individuals and organisations and involve over 200 different interactions and communications. The cost of processing all the paperwork associated with a shipment can easily be around 15% of the shipment costs (Groenfeldt, 2017).

In light of recent events with the Covid-19 pandemic, the fragility of our current supply chains and globalised trade operations were exposed (Lin & Lanng, 2020). In particular, the importance of bringing supply chain infrastructure up to speed through digitisation is important, and emerging technologies will play a part as the enabling force for economic, business and social transformation (Morkunas, Paschen & Boon, 2019).

The relevance of blockchain for supply chains has already been widely discussed in academic literature (Kshetri, 2018). Much of the literature focuses on Bitcoin and explores potential applications. It does not describe the state of the market and its evolution (Min, 2019). Some studies exist that focus on survey-based methods including Petersen, Hackius, and von See (2018) who survey supply chain professionals on the use of blockchain and Wamba, Queiroz and Trinchera (2020) who survey practitioners to investigate the drivers of blockchain adoption in the supply chain.[2]

This study offers a different perspective by exploring the state of blockchain adoption in supply chains based on publicly available data. As part of the research for this paper, 271 relevant blockchain projects were analysed in the supply chain domain. Analysis of the data gathered through this research supports the narrative of both the general trends observed in the blockchain supply chain domain and concerning project inception dates, types of blockchain utilised, stages projects reached, sectors applied to and type of organisation that founded the project. We confirm generally understood trends in the blockchain market with the creation of projects following the general hype and cryptocurrency market prices and funding levels in the market. We observe most activity happening in the Agriculture/Grocery sector and the Freight/Logistics sector. We see the shift in market interest from primarily private companies (startups) to public companies and consortia and the change in blockchain adoption from Ethereum to Hyperledger and the success and failure rates of projects that adopt these blockchains.

2 Materials and Methods

Information on 271 projects utilising blockchain for supply chain purposes was collected and analysed between the period of May and June 2020. Information was found from a variety of different sources through desktop research primarily of company websites, news articles, and data repositories. Project information found was between 2010 and the first half of 2020. Raw data for 31 different fields were collected. An explanation of some of the more relevant fields for data collection is presented below, and in Appendix 1 (Table 5) and a full example of the cleaned data collected is presented for one particular project in Appendix 2 (Table 6).

The methodological approach taken was first to collect raw data from multiple sources and then to clean it. Thereafter, descriptive statistics were utilised to generate information on overall trends in the data. Finally, inferential statistics were applied to generate insights. Analysis was performed on the parameters of time, stage, blockchain, sector, and organisation type.

There were numerous difficulties in the collection of the dataset. This included the variety of sources that were required in order to gather information, as often data from one source would point to another with each presenting some new information that would be relevant for a particular project. There are also issues on the veracity of the data and whether the information found was completely verifiable. Where possible, every effort was taken to verify claims made by projects and companies through manual cross-validation. This aspect is

[2] In this article we use the term ‘blockchain’ to refer also to the larger family of distributed ledger technologies (DLT), i.e., community consensus-based distributed ledgers where the storage of data is not based on chains of blocks.

discussed further below and in the challenges and the limitations of the research section. With regards to the main fields of analysis and data collected, this is further elucidated below.

Project Organisation Classification: Projects were classified based on the type of organisation that was leading a particular project. Rather than define a project by an organisation, the term project is used as a single organisation could have many different projects, and each project was classified separately. The four types of organisations that were used to classify projects were:

- *Private companies* - labelled as startups as the vast majority were early-stage companies.
- *Public companies* - were those that are listed on public equity markets.
- *Consortia* - were identified by having either a separate legal entity or website and a degree of separation from the individual organisations taking part in the project (i.e. not being an organisation and its clients).
- *Government project* - if a government body led a project.

Stage of development: several different classifications were used to describe the stage a project was at including:

- *Failed* – explicit confirmation was found that the project was abandoned, social network accounts were no longer active, and the website had been taken down or that the project had no update for longer than one year.
- *Development* – the project is in the ideation stage.
- *Pilot* – the project has moved from ideation to trial and proof-of-concept.
- *Production* – the project is successful and being deployed in industry and available for partners and clients to purchase and use.

Sectors: Sectors that were considered for classification of the data were firstly based on the SIC classification system. This was too broad, and so a sector classification based on the natural and obvious sectoral classification of the projects was adopted. Sectors were classified if at least 1% of the data was present; otherwise, these projects were placed in the category *Other*. A *Multiple* classification was used for projects that were not sector-specific. Clearly, identifiable sectors in the data that emerged were:

- *Aerospace and Defence*
- *Agriculture / Grocery*
- *Automotive*
- *Fashion*
- *Finance*
- *Freight / Logistics*
- *Luxury items*
- *Mining*
- *Oil & Gas*
- *Pharma / healthcare*

- *Multiple*
- *Other*

Blockchains - The major blockchains classified include those that were identifiable in at least 1% of the data; otherwise, they were classified as *Other*. In many projects, the blockchain was not disclosed, and these projects had their blockchain categorised as *TBC*. Where a blockchain was utilising an existing codebase and was not completely distinct from it, the blockchain from which the codebase derived was used to classify the project. The group *Agnostic* comprises solutions that were not tied to any particular blockchain. The blockchains classifications used were:

- *Ant Blockchain*
- *Bitcoin*
- *Corda*
- *Ethereum*
- *Hyperledger*
- *Oracle Blockchain*
- *Quorum*
- *VeChain*

- *Agnostic*
- *Other*
- *TBC*

3 Results

We present the results on 271 blockchain projects with respect to the year in which the project was created, the blockchains used, the stage the project has reached, the organisation leading the project and the sector the project was applied in.

Figure 1 shows the number of projects with respect to their founding year. The peak of projects being created is in 2018, with 56.8% of all projects founded in 2017 and 2018 alone. After 2018 we see the number of projects fall. 2020's data is only partial for the year (until June), but already has nearly as many projects as 2015. No projects were discovered that were founded in 2011.

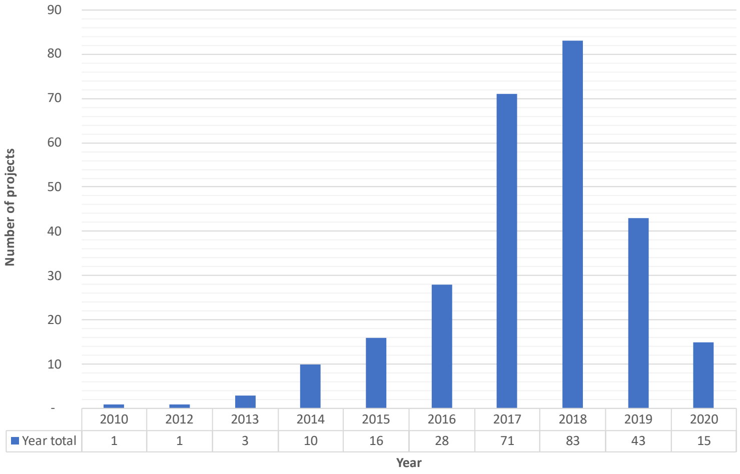


Figure 1: *The number of blockchain projects created in each year*

Figure 2 shows the percentage of projects that use different blockchains based on their founding years. The major blockchains adopted are Ethereum, utilised by 22.9% of all projects and Hyperledger, utilised by 21.4% of all projects. 12.5% of projects are blockchain agnostic.[3] 22.9% of projects do not disclose the blockchain that they use and are in the *TBC* category. Either these projects are still experimenting or deciding which blockchains to use (and in some cases, these projects no longer exist)[4], or they are operating and have not disclosed what

[3] An example project in this category would be Origintrail (https://origintrail.io), who have a protocol that can work with different blockchain solutions.

[4] See for example Resonance (https://www.digicatapult.org.uk/for-startups/success-stories/resonance)

type of blockchain solution they utilise.[5] We also see that projects developed on the Ethereum platform were more prevalent than Hyperledger projects in 2015, 2016, and 2017, whilst this is opposite for projects created in 2018, 2019 and for 2020 so far.

The greatest proportion of Ethereum projects were from companies created in 2017, with 40.3% of all Ethereum based projects created in this year. This is approximately one to two years after the release of Ethereum in July 2015 (Ethereum, 2020), indicating a lag in the creation of projects using this blockchain (as it appears today). On the other hand, the greatest proportion of Hyperledger projects was in 2018 with 37.9% of all projects utilising Hyperledger. Again this also follows a lag of one to two years after the creation of Hyperledger in late 2015, early 2016 (Hyperledger, 2020). Projects in the Agnostic group accounted for the largest percentage of projects founded in 2014, but fluctuate under 20% over other years. Finally, projects in the TBC group are approximately 25-30% of all projects in each of 2016, 2017 and 2018.

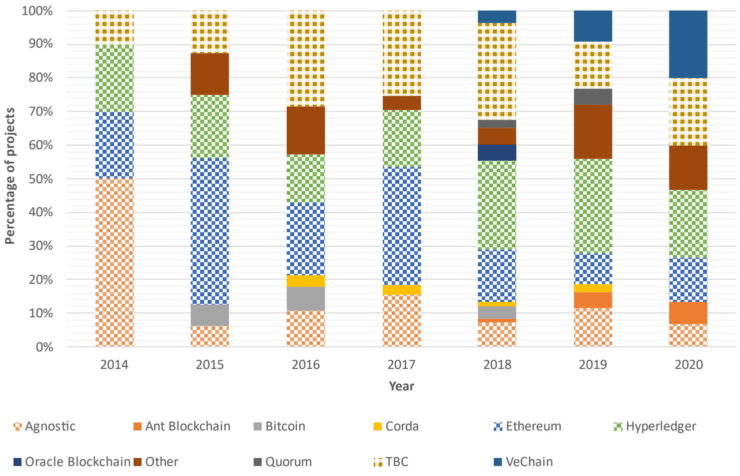


Figure 2: *The percentage of projects using a particular blockchain based on the project's year of creation*

Figure 3 shows the stages that projects have reached based on the year they were created. Out of the entire dataset, 23.2% of projects reached the production stage, 45.4% of projects were in pilot, 21.8% in development and 9.6% were identified as failed. The greatest proportion of projects in production are from 2014, 2015, and 2017 with 50%, 37.5% and 31% of projects respectively. 2016 appears to be an anomaly with only 14.3% of projects from that year reaching production. The greatest proportion of projects in production occurred in 2017. 34.9% of all production projects were created in that year. Many projects also appear stuck in the pilot and development stages, and a minority of projects have also failed. 2017 and 2018 feature the greatest proportion of failed projects, with 34.6% and 46.2% of all failed projects (80.8%) occurring from projects created in these years.

[5] See for example Remedichain https://www.remedichain.org

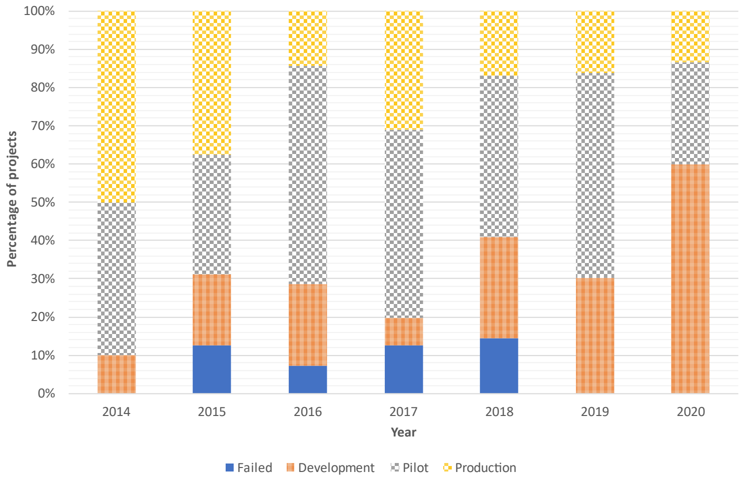


Figure 3: The percentage of projects at various stages based on the project's year of creation

Figure 4 shows the sectors that projects operate in based on their founding year. The top three dominant sectors are Agriculture/Grocery, Freight/Logistics and Multiple sectors at 39.5%, 17% and 12.5% of all projects. Agriculture/Grocery projects account for over 40% of all projects founded in 2014, 2015, 2017, 2018 and 2019. The second most popular sector is Freight/Logistics, which reaches a peak of 35.7% of all projects created in 2016. Finally, 31.8% of all Agriculture/Grocery projects were in 2018 alone, whilst 30.4% of all Freight/Logistics projects were in 2017 alone.

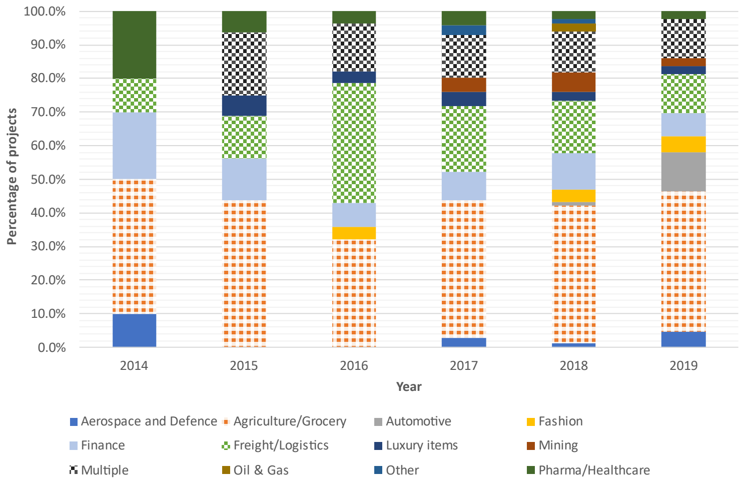


Figure 4: The percentage of projects operating in sectors based on the project's year of creation

Figure 5 shows the lead organisation of a project by the founding year of the project. Startups (private companies) account for 64.2% of the entire dataset, followed by public companies at 17.3%, consortium at 14.8% and finally, government initiatives at 3.7%. Startups account for all projects in 2014 and 2015 and then decline over time, accounting for only 25.6% of all projects created in 2019. Other types of organisations enter into the fray from 2016 onwards. Of all startup projects, 34.5% of them were created in 2017 alone.

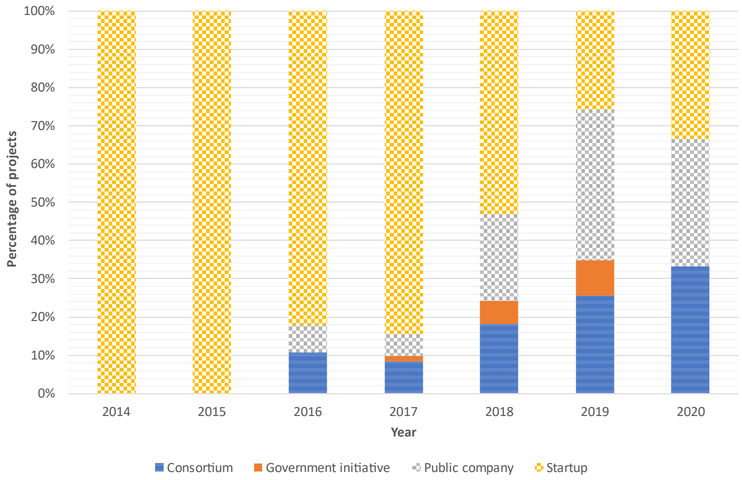


Figure 5: *The percentage of different types of organisation leading projects based on the project's year of creation*

After analysing projects along various dimensions based on the year that they were created, we now analyse all the projects irrespective of time. Figure 6 shows the stages that projects reached based on the applied blockchain solution. Of interest is that Hyperledger projects tend to be more successful on the whole, with 33.3% of projects at the production stage utilising this blockchain. Also of interest is that of those projects identified as having failed, 34.6% of these were utilising the Ethereum platform. Other interesting points to note are that many projects have reached advanced stages without revealing the blockchain technologies that they are working with, which is for example why 16% of projects at the production stage are in the *TBC* category. Of all the projects that utilise Hyperledger, 46.6% of them have reached the pilot phase, 36.2% of them the production phase and only 1.7% have failed. In comparison, to all Ethereum projects, 50% have reached the pilot stage, 21% the production stage and 14.5% have failed.

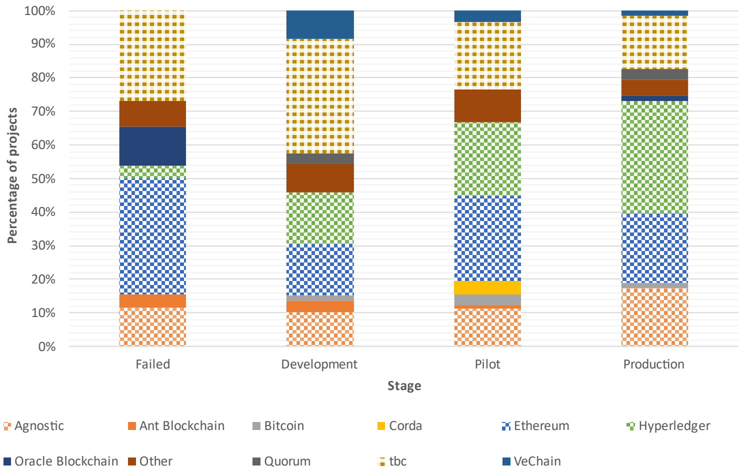


Figure 6: *The percentage of projects using a particular blockchain based on the stages projects reach*

Figure 7 shows the sectors in which the projects are applied in. Sectors which have at least 5% of all projects in the dataset are the Agriculture/Grocery, Freight/Logistics, Multiple and Finance sectors. Looking at the

Agriculture/Grocery sector (with 39.5% of projects overall), 23.4% of all the Agriculture/Grocery projects utilised Ethereum, 18.7% Hyperledger, 8.4% *Agnostic*, 8.4% *Other* and 33.6% were *TBC*. In comparison, for the Freight/Logistics sector (17% of projects overall), 28.3% utilised Ethereum, 10.9% Hyperledger, 13% *Agnostic* and 26.1% *TBC*. For the *Multiple* sectors category, Hyperledger was most popular with 35.3% of projects utilising this blockchain, followed by 23.5% of projects that were *Agnostic* and 8.8% utilising Ethereum. Within the Finance sector (9.2% of projects overall), 24% of projects were in the *TBC* category, 20% utilised Ethereum, and 12% *Agnostic*, Hyperledger and Corda each. Out of all Ethereum-based projects, 40.3% were focused on the Agriculture sector, and 21% in the Freight/Logistics sector. For Hyperledger, 34.5% of projects were in the Agriculture/Grocery sector and 8.6% in the Freight/Logistics sectors. For those projects that did not disclose the blockchains they are working with, 58.1% are focussed on Agriculture.

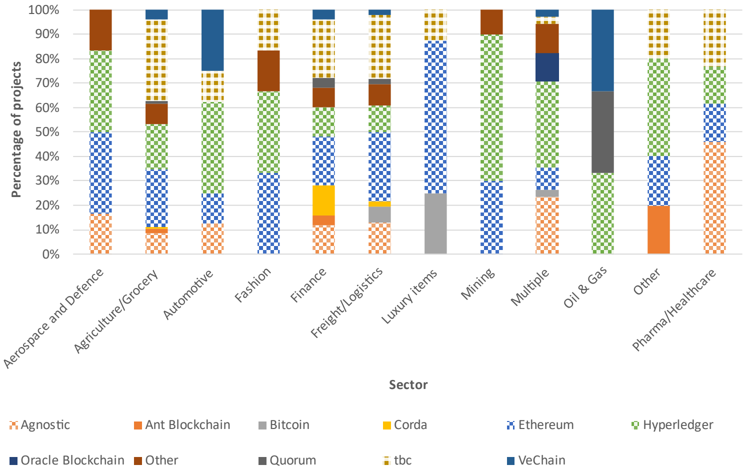


Figure 7: *The percentage of projects using a particular blockchain based on the sector a project operates in*

Figure 8 shows the blockchains used by projects based on the type of lead organisation. Ethereum dominates the startup category, with 32.2% of all startup projects utilising this platform. For consortia and public companies, Hyperledger is the most popular used 35% and 34% respectively. For government initiatives, 50% of projects did not identify the blockchains that they utilised (*TBC*). Out of all Ethereum projects, 90% were utilised by startups. Of all Hyperledger projects, consortia, public companies and startups utilised this blockchain at 24.1%, 27.6% and 44.8%. Overall, startups utilising Ethereum accounted for 20.7% of the dataset as the largest single group.

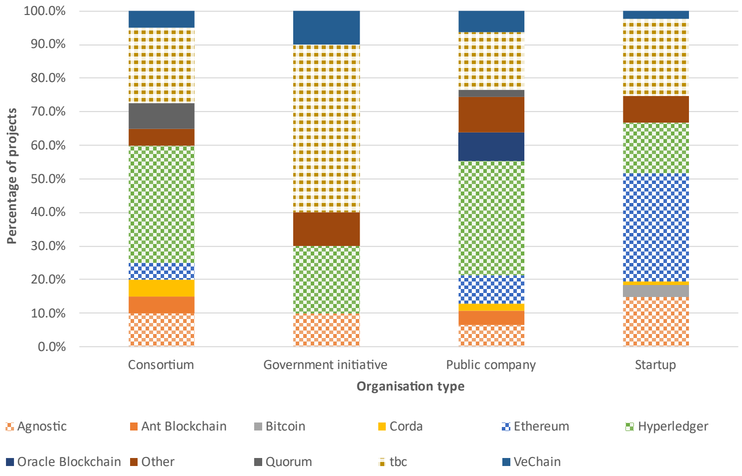


Figure 8: *The percentage of projects using a particular blockchain based on the type of organisation leading the project*

Figure 9 shows the stages reached by different projects based on the type of organisation leading them. Startups - accounting for 64.2% of the entire dataset - have the greatest degree of success in reaching the production stage, with 28.7% of them reaching this stage. Out of all public company projects, 17% of them have reached the production stage, and for government initiatives and consortia, 10% reach this stage each. With regards to failure, startups have the lowest failure rate at 8.6%, followed by consortia and public companies with 10% and 10.6% respectively in their categories. Government initiatives have the highest failure rate at 20% of all projects failing.

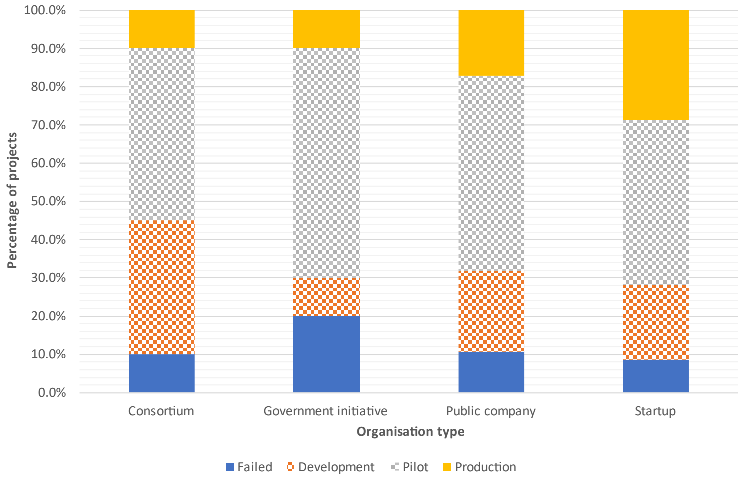


Figure 9: *The percentage of projects at various stages based on the type of organisation leading the project*

Figure 10 shows the projects that operate in different sectors based on the type of organisation leading the project. Agriculture/Grocery projects account for the greatest share in each of the different organisation types with 25%, 60% 31.9% and 43.7% of consortia, government initiatives, public company and startups respectively. Freight/Logistics is the second-largest sector that projects operate in with 22.5%, 20%, 4.3% and 19% of projects in each of the consortia, government initiative, public company and startup categories. Startups are also engaged in the most number of sectors, followed by consortia and public companies. Government initiatives operate in the least number of sectors.

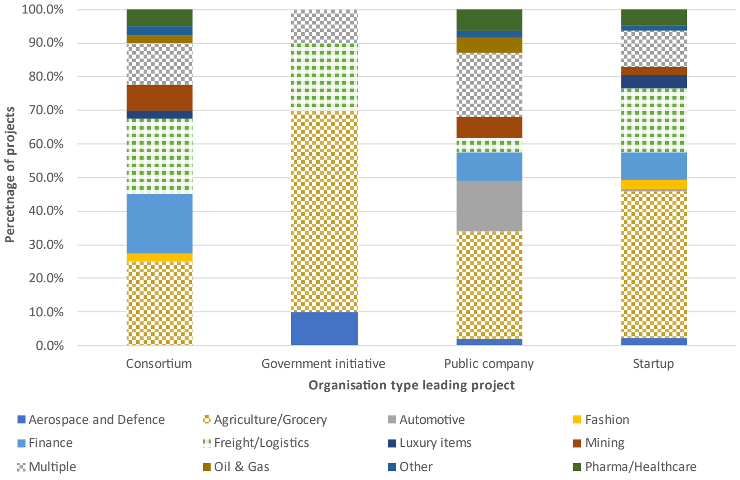


***Figure 10:** The percentage of projects operating in different sectors based on the type of organisation leading the project*

Figure 11 shows the projects at various stages within different sectors. Starting with the most popular sectors of Agriculture/Grocery, Freight/Logistics and Multiple, the percentage of projects within these sectors reaching production are 22.4%, 21.7% and 29.4%, respectively. The sectors with the highest percentage of projects in production are Aerospace and Defense at 33.3%, Oil and Gas at 33.3%, Other at 40% and Pharma/Healthcare at 30.8%. However, it should be noted that the number of projects in these sectors is between 2-5% of the entire dataset. Out of those projects that failed, we observe a failure rate of 14.7% in the *Multiple* category, 10.9% for the Freight/Logistics sector and 8.4% for the Agriculture/Grocery sector.

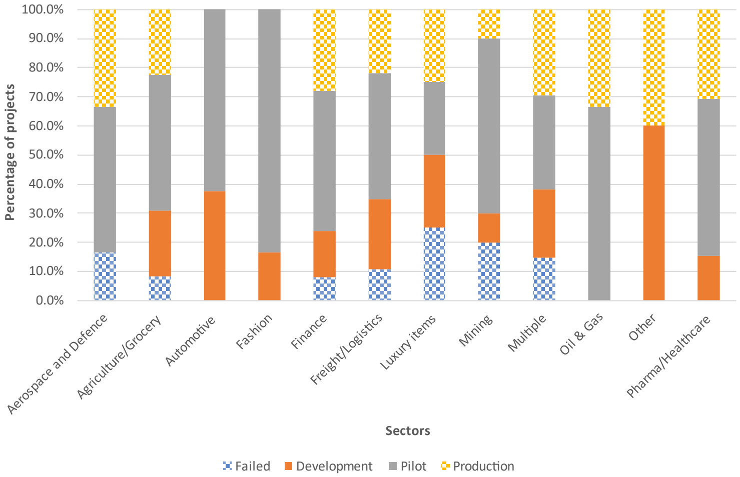


***Figure 11:** The percentage of projects at the stages they have reached within different sectors*

In this last part of the analysis, we look more closely at projects that are utilising the Hyperledger and Ethereum blockchains and also at projects operating in the Agriculture/Grocery and Freight/Logistics sectors. We look at these subsets as these two sectors, and blockchains account for 56.5% and 44.3% of all projects respectively.

Focusing on Ethereum based projects only, Table 1 shows the stages reached by the projects broken down into their leading organisations. Here we see that the vast majority of Ethereum projects are led by startups (90.3%) compared to 64.2% of projects in our entire dataset. With respect to stages, the Ethereum projects failure rate (14.5%) is greater than the entire dataset (9.6%). Finally of all Ethereum projects, 21% reach the production stage compared with 23.2% from the entire dataset.

Organisation Type	**Stage (% of all Ethereum based projects)**					**Organisation total (for all projects)**
	Failed	Development	Pilot	Production	Organisation total	
Consortium	0.0%	1.6%	1.6%	0.0%	3.2%	**14.8%**
Government Initiative	0.0%	0.0%	0.0%	0.0%	0.0%	**3.7%**
Public company	0.0%	1.6%	4.8%	0.0%	6.5%	**17.3%**
Startup	14.5%	11.3%	43.5%	21.0%	90.3%	**64.2%**
Stage total (% of all Ethereum based projects)	14.5%	14.5%	50.0%	21.0%		
Stage total (% of all projects)	**9.6%**	**21.8%**	**45.4%**	**23.2%**		

Table 1: *Ethereum based projects and the stages that they have reached based on the leading organisation of the project*

We repeat the same analysis as above for Hyperledger-based projects only. Table 2 shows the stages reached by the Hyperledger projects broken down into their leading organisations. Differently from Ethereum projects, here we see that the minority of Hyperledger projects are led by startups (44.8%). We also observe that Hyperledger projects are led by consortia and public companies 24.1% and 28%, respectively. This is higher than the dataset average, where we find that consortia and public companies led for 14.8% and 17.3% of all projects respectively. Moreover, we find that of all Hyperledger projects, 36.2% reach the production stage. This percentage is higher than the average production rate of all projects (23.2%). Also, Hyperledger projects' failure rate is very low (1.7%) compared with the average failure rate of all projects (9.6%).

Organisation Type	**Stage (% of all Hyperledger based projects)**					**Organisational total (for all projects)**
	Failed	Development	Pilot	Production	Organisation total	
Consortium	1.7%	1.7%	17.2%	3.4%	24.1%	**14.8%**
Government Initiative	0.0%	0.0%	3.4%	0.0%	3.4%	**3.7%**
Public company	0.0%	5.2%	13.8%	8.6%	27.6%	**17.3%**
Startup	0.0%	8.6%	12.1%	24.1%	44.8%	**64.2%**
Stage total (% of all Hyperledger based projects)	1.7%	15.5%	46.6%	36.2%		
Stage total (% of all projects)	**9.6%**	**21.8%**	**45.4%**	**23.2%**		

Table 2: *Hyperledger based projects and the stages that they have reached based on the leading organisation of the project*

Table 3 compares Hyperledger and Ethereum projects in the Agriculture/Grocery and Freight/Logistics sectors. Here we see that Hyperledger has a greater percentage of projects in production and less failed projects than Ethereum in the Agriculture/Grocery sector. For the Freight/Logistics sector, Ethereum has more production and failed projects than Hyperledger. To summarise, Hyperledger based projects tend to perform better by showing a lower failure rate and a higher production rate not only when compared with Ethereum based projects but also for the entire dataset as well.

Blockchain	**Sector / Stage**	**Failed**	**Development**	**Pilot**	**Production**
Ethereum (% of all Ethereum projects in that sector or stage)	Agriculture/Grocery	12.0%	12.0%	60.0%	16.0%
	Freight/Logistics	15.4%	15.4%	23.1%	46.2%
	Stage total	**14.5%**	**14.5%**	**50.0%**	**21.0%**
Hyperledger (% of all Hyperledger projects in that sector or stage)	Agriculture/Grocery	0.0%	20.0%	45.0%	35.0%
	Freight/Logistics	0.0%	0.0%	80.0%	20.0%
	Stage total	**1.7%**	**15.5%**	**46.6%**	**36.2%**
Percentage of all projects at each stage		**9.6%**	**21.8%**	**45.4%**	**23.2%**

Table 3: *Comparison of projects that use Hyperledger and Ethereum within the Agriculture/Grocery and Freight/Logistics sectors*

Table 4 shows the percentage of projects using either Hyperledger or Ethereum in a particular year and within a particular sector in comparison with the averages for all sectors. Here we can see that for the Agriculture/Grocery sector, Ethereum usage is higher for projects created in 2014, 2015, and 2016 than Ethereum-based projects in all sectors and for Hyperledger projects. For 2017-2020, the proportion of Ethereum projects is lower than Ethereum projects in all sectors. For 2018-2020, the proportion of Hyperledger projects are greater than Ethereum in the Agriculture/Grocery sector. For the Freight/Logistics sector, the proportion of Ethereum projects tend to be in greater than Hyperledger (except in 2014 where they are equal and 2018 where there are more Hyperledger projects).

Sector	Blockchain / Year	2014	2015	2016	2017	2018	2019	2020
Agriculture (% of projects in the sector that use that particular blockchain)	Ethereum	50.0%	71.4%	22.2%	27.6%	14.7%	5.6%	0.0%
	Hyperledger	25.0%	28.6%	22.2%	13.8%	17.6%	22.2%	50.0%
Freight (% of projects in the sector that use that particular blockchain)	Ethereum	0.0%	100.0%	20.0%	50.0%	7.7%	20.0%	0.0%
	Hyperledger	0.0%	0.0%	10.0%	7.1%	23.1%	0.0%	0.0%
All sectors (blockchain used as a percentage of all projects from that year)	Ethereum	20.0%	43.8%	21.4%	35.2%	15.7%	9.3%	13.3%
	Hyperledger	20.0%	18.8%	14.3%	16.9%	26.5%	27.9%	20.0%

Table 4: *Comparison of Ethereum and Hyperledger projects in the Agriculture and Freight and Logistics sectors over time*

4 Discussion

Based on the results, a number of general findings emerge that showcase how blockchain projects are evolving in the supply chain arena. Many of the findings fit industry reporting, news and survey results that are available in the public domain (for example, see Deloitte, 2020). In line with general interest indicators such as google trends (search term "blockchain") (Google, 2020) and cryptocurrency prices (Bitcoin's peak price in late 2017) (CoinMarketCap, 2020), we these patterns of peak interest and price match the pattern of new projects being created. New project formation peaks in 2018 and then drops off in 2019 and 2020. This is most clearly seen in Figure 1, showcasing the number of projects by their year of creation.

As well as the number of projects being created each year, we see the shift in the market from being a startup, and private company dominated to enterprise and consortium dominated. With regards to startups and private companies, as shown in Figure 5, we see the percentage of private company projects trending from 100% in 2014 and gradually decreasing to 25.6% in 2019 and 33.3% for the first half of 2020. This shows that more public companies are coming to the fray to engage in projects, as well as consortia and government initiatives. Public companies account for the majority at 39.5% of all projects in 2019.

Further confirming the movement from private companies to public company led projects is the change in the blockchain platforms utilised. Of identifiable blockchains, Ethereum represents 22.9% of projects in the sample, and Hyperledger represents 21.4% of projects. We see that Ethereum projects account for 43.8%, 21.4% and 35.2% of projects created in 2015, 2016, 2017, and then Hyperledger dominating in 2018, 2019, 2020 with 26.5%, 27.9% and 20% respectively as seen in Figure 2. Part of this switch is, of course, the difference in the inception of the different blockchains. Ethereum was launched in 2015 (Ethereum, 2020), whilst Hyperledger in 2016 (Hyperledger, 2020). The use of these blockchains can be seen in Figure 8. As public companies become dominant in 2018 and 2019, we see that 34% of all public company projects utilise Hyperledger, whilst the 32.2% of all startup projects utilise Ethereum.

One interesting fact point to examine is the number of projects that do not identify the blockchain that they use. As shown in Figure 2, overall, 22.9% of all the projects are in the *TBC* category. Either this is because the projects are still in ideation, or pilot phases where they wish to keep this information proprietary. 72.6% of the projects in the *TBC* category are in the development and pilot stages, more than for Ethereum or Hyperledger (64.5% and 62.1% of those projects). *TBC* projects account for the largest percentage of all projects that are in the development stage at 33.9%, which represents the largest grouping and suggests that these projects are still more in the ideation stage.

Concerning stages, one would expect that projects that have existed for longer (and created earlier) would perhaps have a greater chance of experimenting and getting to the production stage. This is confirmed in Figure 3 where we see that projects created in 2014 and 2015 reached the production stage at 50% and 37.5%, respectively. With respect to failures, 2018 has the highest failure rate, with 14.5% of all projects created in that year having failed. Due to the large number of projects created on the back of hype in the sector, this may indicate these projects were less thought through and therefore had lower chances of success. In particular, the boom in Initial Coin Offerings (ICO) occurred with the peak in cryptocurrency prices and the formation of projects, decreasing the barrier to starting a project as funding was more accessible. Funding peaked for ICO's in 2018 at $12.62 billion (Liu, 2020), after the peak in Bitcoin's price in late 2017 (CoinMarketCap, 2020). The pattern of projects created we observe very much fits the funding cycle for 2017, 2018 and 2019 (Tasca,Vigliotti & Gong, 2018).

Another interesting fact to note is the relatively large number of projects that remain in the development and pilot stages. 67.2% of all projects are in the development and pilot stages, with 27.7% of all projects at these stages founded in the years between 2014 and 2017. There may be two reasons for this. The first could be that projects are particularly complex and take many years to move through stages. This is certainly described in the literature (see, e.g. Iansiti and Lakhani, 2017) which states that it will take years for blockchain as a foundational technology to change the supply chain landscape. The second reason could be that organisations wish to signal that projects are still alive to either not admit failure, or to be ready to revive a project when the time in the market for deployment is right. For projects led by public companies, consortia, and government initiatives, a greater percentage than the dataset average (67.2%) are in the development and pilot stages.

With respect to stages and organisations, we see from Figure 9 that the greatest proportion (28.7%) of startup projects are in the production stage. Government projects had the highest failure rate at 20% of all government projects. This may be due to the fact that government projects may have a greater degree of complexity than private sector projects due to legislative issues and accompanying bureaucracy and this may explain the reason fewer government initiatives have succeeded.

With respect to sectors, as shown in Figure 4, Agriculture/Grocery dominates throughout all years and accounts for 39.5% of the dataset. Food safety is of paramount importance, and so is the ability to track and trace agriculture and grocery products. This may explain why the majority of projects that utilise blockchain are concentrated in this sector. The need for this is brought particularly to light given scandals in recent years where there have been incidents of milk powder contamination (Xiu & Klein, 2010), E-coli outbreaks (Casey & Wong, 2017) and meat substitution (Falkheimer & Heide, 2015).

Freight/Logistics is the second-largest sector accounting for 17% of all projects. As discussed earlier, the complexity of moving products for example from East Africa to Europe required over 200 interactions and involved more than 30 individuals and organisations in the journey and had costs of paperwork exceeding 15% of the entire transportation cost (Bapai, 2017 & Groenfeldt, 2017). This complexity implies there are potentially large efficiency gains that could be made which explains the attention given to these sectors. Agriculture is also the dominant sector amongst organisation types leading projects and is largest for government-led projects, with 60% of these projects taking place in the agriculture sector (above the 39.5% for all projects). Out of all consortia led projects, only 22.5% are in the Freight/Logistics sector, greater than the dataset average (14.8%). This reflects the relative coordination complexity in the sector, implying greater coordination of stakeholders needed, which consortia with their governance structures may be able to facilitate.

Of most interest is the comparison between Ethereum and Hyperledger projects; and the comparison between projects in Agriculture/Grocery and Freight/Logistics sectors.

The results show that Ethereum has a much greater proportion of startups (90% of all projects) than the average across all projects (Table 1). This is very different from Hyperledger, which has a lower proportion in startups (44.8% of all projects), but a relatively higher proportion of consortia and public companies (Table 2). This distinction can be attributed to the different nature of the blockchains. Ethereum is a public blockchain and relatively easy to fundraise for, particularly during the ICO boom as seen in 2017. Hyperledger, on the other hand, is more suited for private usage and therefore fits use by enterprises, or public companies.

On the proportion of projects that have failed compared to those that reached production, Ethereum shows a higher degree of failure and a lower level of production than the average rates of all the projects in the dataset. On the opposite side, Hyperledger projects show a lower failure rate than the average rate of all the projects in the dataset. Moreover, Hyperledger projects show a higher production rate. It is interesting to note why this may be the case, and perhaps this is to do with the nature of blockchain implementation. It is much easier to implement projects amongst an ecosystem if one is the dominant player. Public companies are much more likely to exhibit this behaviour, and certainly, we see this in the case of Hyperledger with 24% of all production projects coming from public companies and 31% of all public companies utilising Hyperledger reaching this stage.

The above results may find an explanation on Wamba, Queiroz and Trinchera's (2020) findings according to which knowledge sharing and trading partner pressure lead to successful outcomes for the adoption of blockchain in supply chains. This supports the assertion that public companies (due to their size and influence) are more likely to be able to create pressure on organisations in their ecosystem to adopt blockchain, thereby potentially leading to greater success for projects. Indeed, the very small number of public companies, consortia or government initiatives leading Ethereum based projects can explain the lower production stage statistics we see for this blockchain.

Finally, Table 4 shows the trends in Hyperledger and Ethereum adoption over time for all projects and the Agriculture/Grocery and Freight/Logistics sectors. As discussed earlier, the shift from Ethereum to Hyperleger can be seen occurring in 2018. This pattern is also seen in the Agriculture/Grocery sector, but not in the Freight/Logistics sector. This may be explained by the fact that Freight/Logistics projects have greater complexity and involve cross border provenance, for instance, requiring the use of public blockchains over private ones, and hence the utilisation of Ethereum here. Indeed, we can observe that Ethereum outperforms Hyperledger in the Freight/Logistics sector because 46% of those projects reached production compared to 20% for Hyperledger.

5 Challenges in the research, limitations and future directions

This research's limitations are primarily based on the difficulty of sourcing full information on the nature of blockchain projects in supply chains. Although there exist some repositories of project information, there are still many more projects that were found by searching online, looking at company websites and examining general press and news reports. It was also difficult to find good quality data on the nature of blockchains in supply chains. This limited the amount of analysis that could be completed. For example, it would be interesting to examine the funding levels for projects. However, within our sample, funding data was only available for 28% of projects and therefore deemed not large enough to draw meaningful conclusions.

Our analysis also focussed on projects with information accessible in English. This precluded many projects that are assumed to be occurring in China. This inference can be made by looking at patent applications. China has accounted for nearly 60% of the total number of blockchain applications submitted by the USA, China, Japan, South Korea and Germany altogether through 2018, with its application total being nearly three times larger than the USA (Chen, 2020). Given this, the fact that only 7.7% of projects in our dataset were operating in China would indicate Chinese projects were most likely underrepresented. This is most likely as Chinese project information is not published in English and therefore, could not be included in this study.

Many projects may not accurately represent their true status with, for example, the large number of projects in the development and pilot stages. Some projects have been at these stages for many years. It is interesting to speculate why they are still active on communications and on their websites (as mentioned in the discussion above). This may undoubtedly be explained as there is a reason to keep a project going on for marketing purposes in case their use case will be useful in the future. Production-level projects may also be overestimated as organisations can simply state they have production-level projects without actually having any other stakeholders utilising their solutions. For example, a startup that builds a blockchain solution that is ready to use can be classified as production-ready, even though there may not be any users.

This research is also a snapshot of the state of the blockchain market historically from today's perspective. This means that although some elements of the market's evolution have been presented, the full extent of all trends in the industry cannot be analysed. For example, some projects in 2013 and 2014 are using the Ethereum blockchain. This is today's snapshot of their behaviour as Ethereun was not launched until the middle of 2015. It was not possible to see what solutions they were using before Ethereum and, when and why they switched technologies.

Finally, it would be interesting to extend the dimensions of this research into other variables if enough information could be found on funding levels, application areas in the supply chain and even to examine if the current pandemic situation has led to more opportunities for implementing DLT based solutions. Furthermore, this research has not examined in detail reasons for why projects have succeeded or failed beyond looking at the statistics and making inferences. It would be interesting to explore several cases with interviews. This may paint a better picture of the market's evolution of blockchain usage over time and enable discussion of best practises leading to more successful project outcomes for the deployment of blockchain in supply chains.

6 Conclusion

In this research, we have begun to map out how blockchain has evolved with respect to its usage in the supply chain sector. Utilising a number of different parameters, we have investigated which sectors have seen projects take place, which blockchains are utilised, what organisations are leading and how successful projects have been. We have observed that the greatest concentration of projects is in the Agriculture/Grocery and Freight/Logistics sectors. We have confirmed market trends that blockchain projects have shifted from being startup (private company) led to public company led and that the most popular blockchain used has changed from Ethereum to Hyperledger. Finally, we see that Hyperledger based projects have a greater success rate and lower failure rate than Ethereum concerning our entire dataset.

7 Appendix 1

Field Name	**Explanation of Field**
Project name	The name of the project, or company if only a single company was leading this project
Website	The website of the project
Type of Organisation behind the Project	The type of project, whether this was: • Startup (or private company) • Government initiative • Public Company • Consortium
Sector of operation	Assessment of most suitable sector for the project
Project location	What country(s) the project is primarily operating in
Region	What region(s) the project is primarily operating in
Year of founding	The year the project was founded
Stage of project	Stage the project has reached: • Failed • Development • Pilot • Production
Organisations Involved in the Project	Other organisations involved in the project if any
Name of DLT utilised	The DLT that was primarily utilised

Table 5: *Explanation of the main fields of data collection.*

8 Appendix 2

Field	**Information**
Project name	Ambrosus
Website	https://ambrosus.com/
Type of Organisation behind the Project	Startup
Sector of operation	Pharma / healthcare
Project location	Switzerland
Region	EMEA
Year of founding	2017
Stage of project	Pilot
Organisations Involved in the Project	Nongshim, UN 10YFP, European Institute of Innovation & Technology
Name of DLT utilised	AMB-NET (Ethereum)

Table 6: *Example of project-specific fields.*

9 References

Bajpai, C.P. (2017). *How IBM And Maersk Will Use The Blockchain To Change The Shipping Industry*. [online] www.nasdaq.com. Available at: https://www.nasdaq.com/articles/how-ibm-and-maersk-will-use-blockchain-change-shipping-industry-2017-03-06 [Accessed 2 Sep. 2020].

Boucher, P. (2017). *How blockchain technology could change our lives*. [online] Available at: https://www.europarl.europa.eu/RegData/etudes/IDAN/2017/581948/EPRS_IDA(2017)581948_EN.pdf.

Bünger, M. (2017). *Blockchain for industrial enterprises: Hype, reality, obstacles and outlook - IoT Agenda*. [online] internetofthingsagenda.techtarget.com. Available at: https://internetofthingsagenda.techtarget.com/blog/IoT-Agenda/Blockchain-for-industrial-enterprises-Hype-reality-obstacles-and-outlook [Accessed 24 Sep. 2020].

Casey, M.J. and Wong, P. (2017). *Global Supply Chains Are About to Get Better, Thanks to Blockchain*. [online] Harvard Business Review. Available at: https://hbr.org/2017/03/global-supply-chains-are-about-to-get-better-thanks-to-blockchain [Accessed 24 Sep. 2020].

Chen, X. (2020). *China dominates blockchain patent applications*. [online] World IP Review. Available at: https://www.worldipreview.com/article/china-dominates-blockchain-patent-applications-as-regulations-are-streamlined [Accessed 24 Sep. 2020].

CoinMarketCap (2020). *Bitcoin*. [online] CoinMarketCap. Available at: https://coinmarketcap.com/currencies/bitcoin/ [Accessed 24 Sep. 2020].

Deloitte (2020). *Deloitte's 2020 Global Blockchain Survey From promise to reality*. [online] Available at: https://www2.deloitte.com/content/dam/insights/us/articles/6608_2020-global-blockchain-survey/DI_CIR%202020%20global%20blockchain%20survey.pdf [Accessed 24 Sep. 2020].

Etheruem (2020). *Ethereum Whitepaper*. [online] ethereum.org. Available at: https://ethereum.org/en/whitepaper/ [Accessed 24 Sep. 2020].

Falkheimer, J. and Heide, M. (2015). Trust and Brand Recovery Campaigns in Crisis: Findus Nordic and the Horsemeat Scandal. *International Journal of Strategic Communication*, 9(2), pp.134–147.

Fosso Wamba, S., Queiroz, M.M. and Trinchera, L. (2020). Dynamics between blockchain adoption determinants and supply chain performance: An empirical investigation. *International Journal of Production Economics*, 229.

Google (2020). *Google Trends*. [online] Google Trends. Available at: https://trends.google.com/trends/explore?date=all&q=blockchain [Accessed 24 Sep. 2020].

Groenfeldt, T. (2017). *IBM And Maersk Apply Blockchain To Container Shipping*. [online] Forbes. Available at: https://www.forbes.com/sites/tomgroenfeldt/2017/03/05/ibm-and-maersk-apply-blockchain-to-container-shipping/#76b588293f05 [Accessed 24 Sep. 2020].

Hughes, A., Park, A., Kietzmann, J. and Archer-Brown, C. (2019). Beyond Bitcoin: What blockchain and distributed ledger technologies mean for firms. *Business Horizons*, 62(3), pp.273–281.

Hyperledger (2020). *About – Hyperledger*. [online] Hyperledger. Available at: https://www.hyperledger.org/about [Accessed 24 Sep. 2020].

Iansiti, M. and Lakhani, K. (2018). *The Truth About Blockchain*. [online] Harvard Business Review. Available at: https://hbr.org/2017/01/the-truth-about-blockchain.

Kshetri, N. (2018). 1 Blockchain’s roles in meeting key supply chain management objectives. *International Journal of Information Management*, 39, pp.80–89.

Lin, J. and Lanng, C. (2020). *Here’s how global supply chains will change after COVID-19*. [online] World Economic Forum. Available at: https://www.weforum.org/agenda/2020/05/this-is-what-global-supply-chains-will-look-like-after-covid-19/ [Accessed 24 Sep. 2020].

Liu, S. (2020). *Amount raised for blockchain ICO projects 2017-2019*. [online] Statista. Available at: https://www.statista.com/statistics/804748/worldwide-amount-crytocurrency-ico-projects/ [Accessed 24 Sep. 2020].

Min, H. (2019). Blockchain technology for enhancing supply chain resilience. *Business Horizons*, 62(1), pp.35–45.

Morkunas, V.J., Paschen, J. and Boon, E. (2019). How blockchain technologies impact your business model. *Business Horizons*, 62(3), pp.295–306.

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [online] Available at: https://bitcoin.org/bitcoin.pdf.

O’Marah, K. (2017). *Blockchain For Supply Chain: Enormous Potential Down The Road*. [online] Forbes. Available at: https://www.forbes.com/sites/kevinomarah/2017/03/09/blockchain-for-supply-chain-enormous-potential-down-the-road/#775af6f73db5 [Accessed 24 Sep. 2020].

Petersen, M., Hackius, N. and von See, B. (2018). Mapping the sea of opportunities: Blockchain in supply chain and logistics. *it - Information Technology*, 60(5–6), pp.263–271.

Pilkington, M. (2016). Blockchain Technology: Principles and Applications. In: F.X. Olleros and M. Zhegu, eds., *Research Handbook on Digital Transformations*. Edward Elgar Publishing, pp.1–39.

Tasca, P., Vigliotti, M.G. and Gong, H. (2018). Risks and challenges of initial coin offerings. *Journal of Digital Banking*, 3(1), pp.81–96.

Xiu, C. and Klein, K.K. (2010). Melamine in milk products in China: Examining the factors that led to deliberate use of the contaminant. *Food Policy*, [online] 35(5), pp.463–470. Available at: https://www.sciencedirect.com/science/article/pii/S0306919210000540.

Discussion Paper #2

Resource-Aware Session Types for Digital Contracts

ANKUSH DAS, Carnegie Mellon University

STEPHANIE BALZER, Carnegie Mellon University

JAN HOFFMANN, Carnegie Mellon University

FRANK PFENNING, Carnegie Mellon University

ISHANI SANTURKAR, Carnegie Mellon University

Abstract

Programming digital contracts comes with unique challenges, which include (i) expressing and enforcing protocols of interaction, (ii) controlling resource usage, and (iii) preventing the duplication or deletion of a contract’s assets. This article presents the design and type-theoretic foundation of Nomos, a programming language for digital contracts that addresses these challenges. To express and enforce protocols, Nomos is based on shared binary session types. To control resource usage, Nomos employs automatic amortized resource analysis. To prevent the duplication or deletion of assets, Nomos uses a linear type system. A monad integrates the effectful session-typed language with a general-purpose functional language. Nomos’ prototype implementation features linear-time type checking and efficient type reconstruction that includes automatic inference of resource bounds via off-the-shelf linear optimization. The effectiveness of the language is evaluated with case studies about implementing common smart contracts such as auctions, elections, and currencies. Nomos is completely formalized, including the type system, a cost semantics, and a transactional semantics to instantiate Nomos contracts on a blockchain. The type soundness proof ensures that protocols are followed at run-time and that types establish sound upper bounds on the resource consumption, ruling out re-entrancy attacks and out-of-gas vulnerabilities.

Resource-Aware Session Types for Digital Contracts

ANKUSH DAS, Carnegie Mellon University
STEPHANIE BALZER, Carnegie Mellon University
JAN HOFFMANN, Carnegie Mellon University
FRANK PFENNING, Carnegie Mellon University
ISHANI SANTURKAR, Carnegie Mellon University

Programming digital contracts comes with unique challenges, which include *(i)* expressing and enforcing protocols of interaction, *(ii)* controlling resource usage, and *(iii)* preventing the duplication or deletion of a contract's assets. This article presents the design and type-theoretic foundation of *Nomos*, a programming language for digital contracts that addresses these challenges. To express and enforce protocols, Nomos is based on *shared binary session types*. To control resource usage, Nomos employs *automatic amortized resource analysis*. To prevent the duplication or deletion of assets, Nomos uses a *linear type system*. A monad integrates the effectful session-typed language with a general-purpose functional language. Nomos' *prototype implementation* features *linear-time type checking* and efficient type reconstruction that includes automatic *inference of resource bounds* via off-the-shelf linear optimization. The effectiveness of the language is evaluated with case studies about implementing common smart contracts such as auctions, elections, and currencies. Nomos is completely formalized, including the type system, a cost semantics, and a transactional semantics to instantiate Nomos contracts on a blockchain. The type soundness proof ensures that protocols are followed at run-time and that types establish sound upper bounds on the resource consumption, ruling out re-entrancy attacks and out-of-gas vulnerabilities.

1 INTRODUCTION

Digital contracts are programs that implement the execution of a contract. With the rise of blockchains and cryptocurrencies such as Bitcoin [Nakamoto 2008], Ethereum [Wood 2014], and Tezos [Goodman 2014], digital contracts have become popular in the form of smart contracts, which provide potentially distrusting parties with programmable money and a distributed consensus mechanism. Smart contracts are used to implement auctions [Auc 2016], investment instruments [Siegel 2016], insurance agreements [Initiative 2008], supply chain management [Law 2017], and mortgage loans [Morabito 2017]. They hold the promise to lower cost, increase fairness, and expand access to the financial infrastructure.

Many of today's prominent smart contract languages suffer from security vulnerabilities, which have severe financial consequences. A well-known example is the attack on The DAO [Siegel 2016], resulting in a $60 million theft by exploiting a contract re-entrancy vulnerability. Smart contract languages have been typically derived from existing general-purpose languages [Auc 2016; Liq 2018; Cachin 2016] and fail to accommodate the domain-specific requirements of digital contracts. These requirements are: *(i)* expressing and enforcing protocols of interaction, *(ii)* controlling resource (or gas) usage, and *(iii)* preventing duplication or deletion of a contract's assets.

This article presents the design, type-theoretic foundation, and implementation of Nomos, a language for digital contracts accommodating these requirements by construction.

To express and enforce the protocols underlying a contract, Nomos is based on *session types* [Caires and Pfenning 2010; Honda 1993; Honda et al. 1998, 2008; Pfenning and Griffith 2015; Toninho et al. 2013; Wadler 2012]. Session types capture the protocols of interactions in the type, rather than the implementation code, and type-checking statically guarantees protocol adherence at run-time. Delimiting the sequences of actions that must be executed

Authors' addresses: Ankush Das, Carnegie Mellon University, ankushd@cs.cmu.edu; Stephanie Balzer, Carnegie Mellon University, balzers@cs.cmu.edu; Jan Hoffmann, Carnegie Mellon University, jhoffmann@cmu.edu; Frank Pfenning, Carnegie Mellon University, fp@cs.cmu.edu; Ishani Santurkar, Carnegie Mellon University, ivs@andrew.cmu.edu.

2018. 2475-1421/2018/1-ART1 $15.00
https://doi.org/

atomically, session types also prevent *re-entrance* into a contract in an inconsistent state. To control resource usage, Nomos employs *automatic amortized resource analysis (AARA)*, a type-based technique for automatically inferring symbolic resource bounds [Carbonneaux et al. 2017; Hoffmann et al. 2011, 2017; Hofmann and Jost 2003; Jost et al. 2010]. AARA is parametric in the cost model, allowing instantiation to track gas usage. As a result, Nomos contracts mitigate denial-of-service attacks without being vulnerable to out-of-gas exceptions. Moreover, resource bounds are integrated with session-typed protocols and enable precise path-sensitive descriptions of cost that avoid gaps between worst-case and average-case cost. To prevent duplication or deletion of assets, Nomos uses a *linear type system* [Girard 1987]. The effectful session-typed language, which implements contract interfaces and contract-to-contract communication, is integrated with a strict, general-purpose functional language using a contextual monad.

Integrating these seemingly disparate approaches (session types, resource analysis, linearity, and functional programming) and combining them with the different roles that arise in a digital contract (contract, asset, transaction) in a way that the result remains consistent, presents unique challenges. For one, both the functional as well as session-typed language use potential annotations to bound the resource consumption, which requires care when functional values are exchanged as messages between processes. For another, prior work on integrating shared and linear session types [Balzer and Pfenning 2017] preclude contracts from persisting their linear assets across transactions, a feature essential to digital contract development; a restriction that we lift in this work. Fundamental is the use of different forms of *typing judgments* for expressions and processes along with *judgmental modes* to distinguish the different roles in a digital contract. The modes are essential in ensuring type safety, as they allow the expression of mode-indexed invariants on the typing contexts and their enforcement by the typing rules.

Nomos is completely formalized, including the type system, a cost semantics, and a transactional semantics to instantiate Nomos contracts on a blockchain. A type soundness proof ensures that protocols are followed at run-time and that types establish sound upper bounds on the resource consumption. Type checking is linear in the size of the program and resource bounds can be efficiently inferred with an off-the-shelf LP solver. Efficient type checking is particularly important if type-checking is part of contract validation and can be used for denial-of-service attacks.

To evaluate Nomos, we implemented a publicly available open-source prototype [Nom 2019] and conducted 8 case studies implementing common smart contracts such as auctions, elections, and currencies. Our experiments show that type-checking overhead is less than 0.7 ms for each contract and bound inference (needed once at deployment) takes less than 10 ms. Moreover, gas bounds are tight for most contracts. To the best of our knowledge, this is the first implementation to integrate shared binary session types into a functional language with support for resource analysis.

To simplify programming and make Nomos accessible to digital contract developers, we incorporated the following design decisions: *(i)* we developed an intuitive surface syntax particularly related to the contextual monad integrating session types into a functional core; *(ii)* we used a bi-directional type checker with a particular focus on improving the quality of error messages whenever a Nomos program fails to typecheck to guide the programmer to locate the source of the error; *(iii)* we used an off-the-shelf LP solver to automatically infer channel modes and potential annotations so that the burden of inference does not fall on the programmer.

Our main technical contributions are:

- design of Nomos, a language that addresses the domain-specific requirements of digital contracts by construction;
- a fine-tuned system of typing judgments (Section 4) that uses *modes* to orchestrate the sound integration of session types (Section 3), functions (Section 5), and resource analysis (Section 6);
- extension of shared session types to store linear assets;
- resource cost amortization by allowing gas storage in internal data structures (Section 6);
- type safety proof of Nomos using a novel asynchronous cost semantics (Section 7);
- an implementation and case study of prominent blockchain applications (Section 8);

- a transactional semantics to deploy and execute Nomos contracts and transactions on a blockchain (Section 9).

In addition, the supplementary material details the technical development, provides additional explanations and provides the full implementation of the blockchain applications.

2 NOMOS BY EXAMPLE

This section provides an overview of the main features of Nomos based on a simple auction contract.

Explicit Protocols of Interaction. Digital contracts, like traditional contracts, follow a *predefined protocol*. For instance, an auction contract distinguishes a bidding phase, where bidders submit their bids, possibly multiple times, from a subsequent collection phase, where the highest bidder receives the lot while all other bidders receive their bids back. In Solidity [Auc 2016], the bidding phase of an auction is typically implemented as the bid function below. This function receives a bid (msg.value) from a bidder (msg.sender) and adds it to the bidder's total previous bids (bidValue).

```
function bid() public payable {
 require (status == running);
 bidder = msg.sender; bid = msg.value;
 bidValue[bidder] = bidValue[bidder] + bid; }
```

To guarantee that a bid can only be placed in the bidding phase, the contract uses the state variable status to track the different phases of a contract. The require statement tests whether the auction is still running and thus accepts bids. It is checked at run-time and aborts the execution if the condition is not met. It is the responsibility of the programmer to define state variables, update them, and introduce corresponding guards.

Rather than burying the contract's interaction protocol in implementation code by means of state variables and run-time checks, Nomos allows the explicit expression and static enforcement of protocols with *session types*. The auction's protocol amounts to the below session type:

$$
\begin{array}{lll}
\mathsf{auction} = \uparrow^{S}_{L} \triangleleft^{22} \oplus \{\textbf{running} : \&\{\textbf{bid} : \mathsf{id} \rightarrow \mathsf{money} \multimap \downarrow^{S}_{L} \mathsf{auction}, & & \%\ \text{recv bid from client} \\
\qquad\qquad\qquad\qquad\qquad\qquad \textbf{cancel} : \triangleright^{21} \downarrow^{S}_{L} \mathsf{auction}\}, & & \%\ \text{client canceled} \\
\qquad\qquad\quad \textbf{ended} : \&\{\textbf{collect} : \mathsf{id} \rightarrow \oplus\{\textbf{won} : \mathsf{lot} \otimes \downarrow^{S}_{L} \mathsf{auction}, & & \%\ \text{client won} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \textbf{lost} : \mathsf{money} \otimes \triangleright^{7} \downarrow^{S}_{L} \mathsf{auction}\}, & & \%\ \text{client lost} \\
\qquad\qquad\qquad\qquad\quad \textbf{cancel} : \triangleright^{21} \downarrow^{S}_{L} \mathsf{auction}\}\} & & \%\ \text{client canceled}
\end{array}
$$

We first focus on how the session type defines the main interactions of a contract with a bidder and ignore the operators $\uparrow^{S}_{L}$, $\downarrow^{S}_{L}$, $\triangleleft$, and $\triangleright$ for now. To distinguish the two main phases an auction can be in, the session type uses an internal choice ($\oplus$), leading the contract to either send the label **running** or **ended**, depending on whether the auction still accepts bids or not, respectively. Dual to an internal choice is an external choice ($\&$), which leaves the choice to the client (i.e., bidder) rather than the provider (i.e., contract). For example, in case the auction is running, the client can choose between placing a bid (label **bid**) or backing out (**cancel**). In the former case, the client indicates their identifier (type id), followed by a payment (type money). Nomos session types allow transfer of both non-linear (e.g., id) and linear assets (e.g., money), using the operators arrow ($\rightarrow$) and ($\multimap$), respectively. Should the auction have ended, the client can choose to check their outcome (label collect) or back out (cancel). In the case of collect, the auction will answer with either **won** or **lost**. In the former case, the auction will send the lot, in the latter case, it will return the client's bid. The linear product ($\otimes$) is dual to $\multimap$ and denotes the transfer of a linear value from the contract to the client. The auction type guarantees that a client cannot collect during the running phase, while they cannot bid during the ended phase.

Nomos uses *shared* session types [Balzer and Pfenning 2017] to guarantee that bidders interact with the auction in mutual exclusion from each other and that the sequences of actions are executed *atomically*. To demarcate the parts of the protocol that become a *critical section*, the above session type uses the $\uparrow^{S}_{L}$ and $\downarrow^{S}_{L}$ modalities. The $\uparrow^{S}_{L}$ modality denotes the beginning of a critical section, the $\downarrow^{S}_{L}$ modality denotes its end. Programmatically, $\uparrow^{S}_{L}$ translates into an *acquire* of the auction session and $\downarrow^{\mathsf{S}}_{\mathsf{L}}$ into its *release*, which is only sound if the protocol behaves like an auction afterwards (*equi-synchronizing* type).

Contracts are implemented by *processes*, revealing the concurrent, message-passing nature of session-typed languages. The process *run* below implements the auction's running phase. Line 2 gives the process' signature, indicating that it offers a shared session of type auction along the channel *sa* and uses a linear hash map b : hashmap$_{\mathsf{id,bid}}$ of bids indexed by id and a linear lot l. The bid session type (line 1) can be queried for the stored identifier and bid value, and is offered by a process (not shown) that internally stores this identifier and money. Line 4 onward list the process body. Line 1 defines session types bid and bids, respectively.

```
 1:  stype bid = &{addr : id ∧ bid, val : money},   stype bids = hashmap$_{\mathsf{id,bid}}$
 2:  (b : bids), (l : lot) ⊢ run :: (sa : auction)            %  syntax for process declaration
 3:     sa ← run ← b l =                                       %  syntax for process definition
 4:        la ← accept sa ;                                    %  accept a client acquire request
 5:        la.running ;                                        %  auction is running
 6:        case la ( bid ⇒ r ← recv la ;                       %  receive identifier r : id
 7:                        m ← recv la ;                       %  receive bid m : money
 8:                        sa ← detach la ;                    %  detach from client
 9:                        b′ ← addbid r ← b m ;               %  store bid internally
10:                        sa ← check ← b′ l                   %  check if threshold reached
11:                 | cancel ⇒ sa ← detach la ;                %  detach from client
12:                          sa ← run ← b l)                   %  recurse
```

The contract process first *accepts* an acquire request by a bidder (line 4) and then sends the message running (line 5), indicating the auction status and waiting for the bidder's choice. Should the bidder choose to make a bid, the process waits to receive the bidder's identifier (line 6) followed by money equivalent to the bidder's bid (line 7). After this linear exchange, the process leaves the critical section by issuing a *detach* (line 8), matching the bidder's release request. Internally, the process stores the pair of the bidder's identifier and bid in the data structure bids (line 9). The ended protocol of the contract is governed by a different process (not shown), responsible for distributing the bids back to the clients. The contract transitions to the ended state when the number of bidders reaches a threshold (stored in auction). This is achieved by the *check* process (line 10) which checks if the threshold has been reached and makes this transition, or calls *run* otherwise. Should the bidder choose to cancel, the contract simply detaches and recurses (lines 11,12).

Re-Entrancy Vulnerabilities. A contract function is re-entrant if, once called by a user, it can potentially be called again before the previous call has completed. As an illustration, consider the following collect function of the auction contract in Solidity where the funds are transferred to the bidder before the hash map is updated to reflect this change.

```
function collect() public payable {
  require (status == ended);
  bidder = msg.sender; bid = bidValue[bidder];
  bidder.send(bid); bidValue[bidder] = 0; }
```

```
function () payable {
  // 'auction' variable stores the
  // address to auction contract
  auction.collect(); }
```

A bidder can now cause re-entrancy by creating a dummy contract with an unnamed *fallback* function (on the right) that calls the auction's collect function. This call is triggered when collect calls send (last line on the left), leading to an infinite recursive call to collect, depleting all funds from the auction. The message-passing framework of session types eliminates this vulnerability. While session types provide multiple clients access to a contract, the acquire-release discipline ensures that clients interact with the contract in mutual exclusion. To attempt re-entrancy, a bidder will need to acquire the auction contract twice without releasing it.

Linear Assets. Nomos integrates a linear type system that tracks the assets stored in a process. The type system enforces that assets are never duplicated, but only exchanged between processes. Moreover, the type system prevents a process from terminating while it holds linear assets. For example, the auction contract treats money and lot as linear assets, which is witnessed by the use of the linear operators $\multimap$ and $\otimes$ for their exchange. In contrast, no provisions to handle assets linearly exist in Solidity, allowing such assets to be created out of thin air, duplicated, or discarded. In the above bid function, for instance, the language does not prevent the programmer from writing $\mathsf{bidValue[bidder] = bid}$ instead, losing the bidder's previous bid.

Resource Cost. Another important aspect of digital contracts is their *resource usage*. On a blockchain, executing a contract function, or *transaction*, requires new blocks to be added to the blockchain. In existing blockchains like Ethereum, this is done by *miners* who charge a fee based on the *gas* usage of the transaction, indicating the cost of its execution. Precisely computing this cost statically is important because the sender of a transaction must pay this fee to the miners along with sending the transaction. If the sender does not pay a sufficient amount, the transaction will be rejected by the miners and the sender's fee is lost!

Nomos uses resource-aware session types [Das et al. 2018b] to statically analyze the resource cost of a transaction. They operate by assigning an initial *potential* to each process. This potential is consumed by each operation that the process executes or can be transferred between processes to share and amortize cost. The cost of each operation is defined by a cost model. If the cost model assigns a cost to each operation as equivalent to their gas cost during execution, the potential consumed during a transaction reflects upper bound on the gas usage.

Resource-aware session types express the potential as part of the session type using the operators $\triangleleft$ and $\triangleright$. The $\triangleleft$ operator prescribes that the client must send potential to the contract, with the amount of potential indicated as a superscript. Dually, $\triangleright$ prescribes that the contract must send potential to the client. In case of the auction contract, we require the client to pay potential for the operations that the contract must execute, both while placing and collecting their bids. If the cost model assigns a cost of 1 to each contract operation, then the maximum cost of an auction session is 22 (taking the max number of operations in all branches). Thus, we require the client to send 22 units of potential at the start of a session using $\triangleleft^{22}$. In the lost branch of the auction type, on the other hand, the contract returns 7 units of potential to the client using $\triangleright^{7}$. This simulates gas usage in smart contracts, where the sender initiates a transaction with some initial gas, and the leftover gas at the end of the transaction is returned to the sender. In contrast to existing smart contract languages like Solidity, which provide no support for analyzing the cost of a transaction, Nomos' type checker has automatically inferred these potential annotations and guarantees that well-typed transactions cannot run out of gas.

Bringing It All Together. Combining all these features soundly in one language is challenging. In Nomos, we achieve this by using different *typing judgments* and *modes*, identifying the role of the process offered along that channel. The mode R denotes *purely linear processes* for linear assets or private data structures, such as b and l in the auction. The modes S and L denote *sharable processes*, i.e., contracts, that are either in their shared or linear phase such as sa and la, respectively. The mode T denotes a *transaction process* that can refer to shared and linear processes and is issued by a user, such as bidder in the auction. The mode assignment carries over into the process typing judgments imposing invariants (Definition 1) that are key to type safety. The mode annotations are automatically inferred by the type checker relieving programmers from this burden.

3 BASE SYSTEM OF SESSION TYPES

Nomos builds on linear session types for message-passing concurrency [Caires and Pfenning 2010; Honda 1993; Honda et al. 1998, 2008; Wadler 2012] and, in particular, on the line of works that have a logical foundation due to the existence of a Curry-Howard correspondence between linear logic and the session-typed π-calculus [Caires and Pfenning 2010; Wadler 2012]. Linear logic [Girard 1987] is a substructural logic that exhibits exchange as the only structural property, with no contraction or weakening. As a result, linear propositions can be viewed as resources

Session Type	**Cont.**	**Process Term**	**Cont.**	**Description**
$c : \oplus\{\ell : A_\ell\}_{\ell \in L}$	$c : A_k$	$c.k \;;\; P$	P	provider sends label k along c
		$\text{case } c\ (\ell \Rightarrow Q_\ell)_{\ell \in L}$	Q_k	client receives label k along c
$c : \&\{\ell : A_\ell\}$	$c : A_k$	$\text{case } c\ (\ell \Rightarrow P_\ell)_{\ell \in L}$	P_k	provider receives label k along c
		$c.k \;;\; Q$	Q	client sends label k along c
$c : A \otimes B$	$c : B$	$\text{send } c\ w \;;\; P$	P	provider sends channel $w : A$ on c
		$y \leftarrow \text{recv } c \;;\; Q_y$	$[w/y]Q_y$	client receives channel $w : A$ on c
$c : A \multimap B$	$c : B$	$y \leftarrow \text{recv } c \;;\; P_y$	$[w/y]P_y$	provider receives chan. $w : A$ on c
		$\text{send } c\ w \;;\; Q$	Q	client sends channel $w : A$ on c
$c : \mathbf{1}$	–	$\text{close } c$	–	provider sends *end* along c
		$\text{wait } c \;;\; Q$	Q	client receives *end* along c

Table 1. Overview of binary session types with their operational description

that must be used *exactly once* in a proof. Under the Curry-Howard correspondence, an intuitionistic linear sequent $A_1, A_2, \ldots, A_n \vdash C$ can be interpreted as the offer of a session C by a process P using the sessions $A_1, A_2, \ldots, A_n$

$$(x_1 : A_1), (x_2 : A_2), \ldots, (x_n : A_n) \vdash P :: (z : C)$$

We label each antecedent as well as the conclusion with the name of the channel along which the session is provided. The x_i's correspond to channels *used by* P, and z is the channel *provided by* P. As is standard, we use the linear context Δ to combine multiple assumptions.

For the typing of processes in Nomos, we extend the above judgment with two additional contexts (Ψ and Γ), a resource annotation q, and a mode m of the offered channel:

$$\Psi \;;\; \Gamma \;;\; \Delta \vdash^{q} P :: (x_m : A)$$

We will gradually introduce each concept in the remainder of this article. For future reference, we show the complete typing rules, with additional contexts, resource annotations, and modes henceforth, but highlight the parts that will be discussed in later sections in blue.

The Curry-Howard correspondence gives each connective of linear logic an interpretation as a session type, as demonstrated by the grammar:

$$A, B \quad ::= \quad \oplus\{\ell : A\}_{\ell \in K} \mid \&\{\ell : A\}_{\ell \in K} \mid A \multimap_m B \mid A \otimes_m B \mid \mathbf{1}$$

Each type prescribes the kind of message that must be sent or received along a channel of that type and at which type the session continues after the exchange. Following previous work on session types [Pfenning and Griffith 2015; Toninho et al. 2013], the process expressions of Nomos are defined as follows.

$$P ::= x.l \;;\; P \mid \text{case } x\ (\ell \Rightarrow P)_{\ell \in K} \mid \text{close } x \mid \text{wait } x \;;\; P \mid \text{send } x\ w \;;\; P \mid y \leftarrow \text{recv } x \;;\; P \mid x \leftarrow y$$

Table 1 provides an overview of the types along with their operational meaning. Because we adopt the intuitionistic version of linear logic, session types are expressed from the point of view of the provider. Table 1 provides the viewpoint of the provider in the first line, and that of the client in the second line for each connective. Columns 1 and 3 describe the session type and process term before the interaction. Similarly, columns 2 and 4 describe the type and term after the interaction. Finally, the last column describes the provider and client action. Figure 1 provides the corresponding typing rules. As illustrations of the statics and semantics, we explain internal choice ($\oplus$) and linear implication ($\multimap$) connectives. The complete formalization is presented in the supplement.

Internal Choice. The linear logic connective $A \oplus B$ has been generalized to n-ary labeled sum $\oplus\{\ell : A_\ell\}_{\ell \in K}$. A process that provides $x : \oplus\{\ell : A_\ell\}_{\ell \in K}$ can send any label $l \in K$ along x and then continues by providing $x : A_l$. The corresponding process term is written as $(x.l \;;\; P)$, where P is the continuation. A client branches on the label

$\boxed{\Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{q} P :: (x_m : A)}$ Process P uses linear channels in Δ and offers type A on channel x

$$\frac{\Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{q} P :: (x_m : A_l) \qquad (l \in K)}{\Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{q} x_m.l \; ; \; P :: (x_m : \oplus\{\ell : A_\ell\}_{\ell \in K})} \oplus R$$

$$\frac{\Psi \; ; \; \Gamma \; ; \; \Delta, (x_m : A_\ell) \vdash^{q} Q_\ell :: (z_k : C) \qquad (\forall \ell \in K)}{\Psi \; ; \; \Gamma \; ; \; \Delta, (x_m : \oplus\{\ell : A_\ell\}_{\ell \in K}) \vdash^{q} \mathsf{case}\ x_m\ (\ell \Rightarrow Q_\ell)_{\ell \in K} :: (z_k : C)} \oplus L$$

$$\frac{\Psi \; ; \; \Gamma \; ; \; \Delta, (y_n : A) \vdash^{q} P :: (x_m : B)}{\Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{q} y_n \leftarrow \mathsf{recv}\ x_m \; ; \; P :: (x_m : A \multimap_n B)} \multimap_n R$$

$$\frac{\Psi \; ; \; \Gamma \; ; \; \Delta, (x_m : B) \vdash^{q} Q :: (z_k : C)}{\Psi \; ; \; \Gamma \; ; \; \Delta, (w_n : A), (x_m : A \multimap_n B) \vdash^{q} \mathsf{send}\ x_m\ w_n \; ; \; Q :: (z_k : C)} \multimap_n L$$

$$\frac{q = 0}{\Psi \; ; \; \Gamma \; ; \; (y_m : A) \vdash^{q} x_m \leftarrow y_m :: (x_m : A)} \mathsf{fwd}$$

Fig. 1. Selected typing rules for process communication

received along x using the term case $x\ (\ell \Rightarrow Q_\ell)_{\ell \in K}$. The typing rules for the provider and client are $\oplus R$ and $\oplus L$, respectively, in Figure 1.

The operational semantics is formalized as a system of *multiset rewriting rules* [Cervesato and Scedrov 2009]. We introduce semantic objects $\mathsf{proc}(c_m, w, P)$ and $\mathsf{msg}(c_m, w, N)$ denoting process P and message N, respectively, being provided along channel c at mode m. The resource annotation w indicates the work performed so far, the discussion of which we defer to Section 6. Communication is *asynchronous*, allowing the sender $(c_m.l \; ; \; P)$ to continue with P without waiting for l to be received. As a technical device to ensure that consecutive messages arrive in the order they were sent, the sender also creates a fresh continuation channel c_m^+ so that the message l is actually represented as $(c_m.l \; ; \; c_m \leftarrow c_m^+)$ (read: send l along c_m and continue as c_m^+):

$$(\oplus S) : \quad \mathsf{proc}(c_m, w, c_m.l \; ; \; P) \;\mapsto\; \mathsf{proc}(c_m^+, w, [c_m^+/c_m]P), \mathsf{msg}(c_m, 0, c_m.l \; ; \; c_m \leftarrow c_m^+)$$

Receiving the message l corresponds to selecting branch Q_l and substituting continuation c^+ for c:

$$(\oplus C) : \quad \mathsf{msg}(c_m, w, c_m.l \; ; \; c_m \leftarrow c_m^+), \mathsf{proc}(d_k, w', \mathsf{case}\ c_m\ (\ell \Rightarrow Q_\ell)_{\ell \in K}) \;\mapsto\; \mathsf{proc}(d_k, w + w', [c_m^+/c_m]Q_l)$$

The message $\mathsf{msg}(c_m, w, c_m.l \; ; \; c_m \leftarrow c_m^+)$ is just a particular form of process, where $c_m \leftarrow c_m^+$ is *forwarding*, which is explained below. Therefore, no separate typing rules for messages are needed; they can be typed as processes [Balzer and Pfenning 2017].

Channel Passing. Nomos allows the exchange of channels over channels, also referred to as higher-order channels. A process providing $A \multimap_n B$ can receive a channel of type A at mode n and then continue with providing B. The provider process term is $(y_n \leftarrow \mathsf{recv}\ x_m \; ; \; P)$, where P is the continuation. The corresponding client sends this channel using $(\mathsf{send}\ x_m\ w_n \; ; \; Q)$. The corresponding typing rules are presented in Figure 1. Operationally, the client creates a message containing the channel:

$$(\multimap_n S) : \mathsf{proc}(d_k, w, \mathsf{send}\ c_m\ e_n \; ; \; P) \mapsto \mathsf{msg}(c_m^+, 0, \mathsf{send}\ c_m\ e_n \; ; \; c_m^+ \leftarrow c_m), \mathsf{proc}(d_k, w, [c_m^+/c_m]P)$$

The provider receives this channel, and substitutes it appropriately.

$$(\multimap_n C) : \mathsf{proc}(c_m, w', x_n \leftarrow \mathsf{recv}\ c_m \; ; \; Q), \mathsf{msg}(c_m^+, w, \mathsf{send}\ c_m\ e_n \; ; \; c_m^+ \leftarrow c_m) \mapsto \mathsf{proc}(c_m^+, w + w', [c_m^+/c_m][e_n/x_n]Q)$$

An important distinction from standard session types is that the $\multimap$ and $\otimes$ types are decorated with the mode m of the channel exchanged. Since modes distinguish the status of the channels in Nomos, this mode decoration is necessary to ensure type safety.

Forwarding. A forwarding process $x_m \leftarrow y_m$ (which provides channel x) identifies channels x and y (both at mode m) so that any further communication along x or y occurs on the unified channel. The typing rule fwd is given in Figure 1 and corresponds to the logical rule of *identity*.

$$\begin{array}{lll} (\mathrm{id}^{+}C): \quad \mathrm{msg}(d_m, w', N), \mathrm{proc}(c_m, w, c_m \leftarrow d_m) & \mapsto & \mathrm{msg}(c_m, w + w', [c_m/d_m]N) \\ (\mathrm{id}^{-}C): \quad \mathrm{proc}(c_m, w, c_m \leftarrow d_m), \mathrm{msg}(e_k, w', N(c_m)) & \mapsto & \mathrm{msg}(e_k, w + w', N(d_m)) \end{array}$$

Operationally, a process $c \leftarrow d$ *forwards* any message N that arrives along d to c and vice versa. Since linearity ensures that every process has a unique client, forwarding results in terminating the forwarding process and corresponding renaming of the channel in the client process.

Process and Type Definitions. Process definitions have the form $\Psi \; ; \; \Gamma \; ; \Delta \vDash^{q} f = P :: (x_m : A)$ where f is the name of the process and P its definition. All definitions are collected in a fixed global signature Σ. We require well-typedness, i.e., $\Psi \; ; \; \Gamma \; ; \Delta \vDash^{q} f = P :: (x_m : A)$ for every definition, which allows the definitions to be mutually recursive. For readability of the examples, we break a definition into two declarations, one providing the type (on the left) and the other the process definition (on the right) binding the variables x_m and those in Ψ, Γ and Δ (omitting their types):

$$\Psi \; ; \; \Gamma \; ; \Delta \vDash^{q} f = P :: (x_m : A) \qquad\qquad x_m \leftarrow f \; \Psi \leftarrow \Gamma \; ; \Delta = P$$

A new instance of a defined process f can be spawned with the expression $x_m \leftarrow f \; \overline{y_1} \leftarrow \overline{y_2} \; ; \; Q$ where $\overline{y_1}$ is a sequence of functional variables matching the antecedents Ψ and $\overline{y_2}$ is a sequence of channels matching the antecedents $\Gamma \; ; \; \Delta$. The newly spawned process will use all variables in $\overline{y_1}$ and channels in $\overline{y_2}$ and provide x_m to the continuation Q. The operational semantics is defined by

$$\begin{array}{l} (\mathrm{def}C): \mathrm{proc}(c_k, w, x_m \leftarrow f \; \overline{d} \leftarrow \overline{e} \; ; \; Q) \;\; \mapsto \\ \qquad\qquad\qquad\qquad \mathrm{proc}(a_m, 0, [a_m/x_m, \overline{d}/\Psi, \overline{e}/\Gamma \; \Delta]P), \mathrm{proc}(c_k, w, [a_m/x_m]Q) \end{array}$$

where a_m is a fresh channel. Here we write $[\overline{d}/\Psi]$ and $[\overline{e}/\Gamma \; \Delta]$ to denote substitution of the variables in $\overline{d}$ and $\overline{e}$ for the corresponding variables in Ψ and $\Gamma \; ; \; \Delta$ respectively in that order.

Sometimes a process invocation is a *tail call*, written without a continuation as $x_m \leftarrow f \; \overline{y_1} \leftarrow \overline{y_2}$. This is a short-hand for $x'_m \leftarrow f \; \overline{y_1} \leftarrow \overline{y_2} \; ; \; x_m \leftarrow x'_m$ for a fresh variable x'_m, that is, we create a fresh channel and immediately identify it with x_m (although it is implemented more efficiently).

Session types can be naturally extended to include recursive types. For this purpose we allow (possibly mutually recursive) type definitions $X = A$ in the signature, where we require A to be *contractive* [Gay and Hole 2005]. This means here that A should not itself be a type name. Our type definitions are *equi-recursive* so we can silently replace X by A during type checking, and no explicit rules for recursive types are needed.

4 SHARING CONTRACTS

Multi-user support is fundamental to digital contract development. Linear session types, as defined in Section 3, unfortunately preclude such sharing because they restrict processes to exactly one client; only one bidder for the auction, for instance (who will always win!). To support multi-user contracts, we base Nomos on *shared* session types [Balzer and Pfenning 2017]. Shared session types impose an acquire-release discipline on shared processes to guarantee that multiple clients interact with a contract in *mutual exclusion* of each other. When a client acquires a shared contract, it obtains a private linear channel along which it can communicate with the contract undisturbed by any other clients. Once the client releases the contract, it loses its private linear channel and only retains a shared reference to the contract.

$$\begin{array}{lcl}
A_{\mathsf{R}} & ::= & \oplus\{\ell : A_{\mathsf{R}}\}_{\ell \in L} \mid \&\{\ell : A_{\mathsf{R}}\}_{\ell \in L} \mid \mathbf{1} \mid A_m \multimap_m A_{\mathsf{R}} \mid A_m \otimes_m A_{\mathsf{R}} \mid \tau \to A_{\mathsf{R}} \mid \tau \wedge A_{\mathsf{R}} \\
A_{\mathsf{L}} & ::= & \oplus\{\ell : A_{\mathsf{L}}\}_{\ell \in L} \mid \&\{\ell : A_{\mathsf{L}}\}_{\ell \in L} \mid \mathbf{1} \mid A_m \multimap_m A_{\mathsf{L}} \mid A_m \otimes_m A_{\mathsf{L}} \mid \tau \to A_{\mathsf{L}} \mid \tau \wedge A_{\mathsf{L}} \mid \downarrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{S}} \\
A_{\mathsf{S}} & ::= & \uparrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{L}} \\
A_{\mathsf{T}} & ::= & A_{\mathsf{R}}
\end{array}$$

Fig. 2. Grammar for shared session types

A key idea of shared session types is to lift the acquire-release discipline to the type level. Generalizing the idea of type *stratification* [Benton 1994; Pfenning and Griffith 2015; Reed 2009], session types are stratified into a linear and shared layer with two *adjoint modalities* going back and forth between them:

$$\begin{array}{lcll}
A_{\mathsf{S}} & ::= & \uparrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{L}} & \text{shared session type} \\
A_{\mathsf{L}} & ::= & \ldots \mid \downarrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{S}} & \text{linear session types}
\end{array}$$

The $\uparrow_{\mathsf{L}}^{\mathsf{S}}$ type modality translates into an *acquire*, while the dual $\downarrow_{\mathsf{L}}^{\mathsf{S}}$ type modality into a *release*. Whereas mutual exclusion is one key ingredient to guarantee session fidelity (a.k.a. type preservation) for shared session types, the other key ingredient is the requirement that a session type is *equi-synchronizing*. A session type is equi-synchronizing if it imposes the invariant on a process to be released back to the same type at which the process was previously acquired. This is also the key behind eliminating *re-entrancy vulnerabilities* since it prevents a user from interrupting an ongoing session in the middle and initiating a new one.

Recall the process typing judgment in Nomos $\Psi\ ;\ \Gamma\ ;\ \Delta \vdash^{q} P :: (x_m : A)$ denoting a process P offering service of type A along channel x at mode m. The contexts Γ and Δ store the shared and linear channels that P can refer to, respectively (Ψ and q are explained later and thus marked in blue in Figure 3). The stratification of channels into layers arises from a difference in structural properties that exist for types at a mode. Shared propositions exhibit weakening, contraction and exchange, thus can be discarded or duplicated, while linear propositions only exhibit exchange.

Allowing Contracts to Rely on Linear Assets. As exemplified by the auction contract, a digital contract typically amounts to a process that is shared at the outset, but oscillates between shared and linear to interact with clients, one at a time. Crucial for this pattern is the ability of a contract to maintain its linear assets (e.g., money or lot for the auction) regardless of its mode. Unfortunately, current shared session types [Balzer and Pfenning 2017] do not allow a shared process to rely on any linear channels, requiring any linear assets to be consumed before becoming shared. This precaution was logically motivated [Pruiksma et al. 2018] and also crucial for type preservation.

A key novelty of our work is to lift this restriction while *maintaining type preservation*. The main concern regarding preservation is to prevent a process from acquiring its client, which would result in a cycle in the linear process tree. To this end, we factorize the process typing judgment according to the *three roles* that arise in digital contract programs: *contracts*, *transactions*, and *linear assets*. Since contracts are shared and thus can oscillate between shared and linear, we get 4 sub-judgments for typing processes, each characterized by the mode of the channel being offered.

Definition 1 (Process Typing). *The judgment $\Psi\ ;\ \Gamma\ ;\ \Delta \vdash^{q} P :: (x_m : A)$ is categorized according to mode m. This factorization imposes certain invariants on the judgment outlined below.* $\mathbf{L}(A)$ *denotes the language generated by the grammar of A.*

(1) If $m = \mathsf{R}$, then (i) Γ is empty, (ii) for all $d_k \in \Delta \implies k = \mathsf{R}$, and (iii) $A \in \mathbf{L}(A_{\mathsf{R}})$.
(2) If $m = \mathsf{S}$, then (i) for all $d_k \in \Delta \implies k = \mathsf{R}$, and (ii) $A \in \mathbf{L}(A_{\mathsf{S}})$.
(3) If $m = \mathsf{L}$, then (i) for all $d_k \in \Delta \implies k = \mathsf{R} \vee k = \mathsf{L}$, and (ii) $A \in \mathbf{L}(A_{\mathsf{L}})$.
(4) If $m = \mathsf{T}$, then $A \in \mathbf{L}(A_{\mathsf{T}})$.

Figure 2 shows the session type grammar in Nomos. The first sub-judgment in Definition 1 is for typing linear assets. These type a purely linear process P using a purely linear context Δ (types belonging to grammar A_{R} in Figure 2) and offering a purely linear type A along channel x_{R}. The mode R of the channel indicates that a purely linear session is offered. The second and third sub-judgments are for typing contracts. The second sub-judgment shows the type of a contract process P using a shared context Γ and a purely linear channel context Δ (judgment Δ $\mathsf{purelin}$) and offering shared type A on the shared channel x_S. Once this shared channel is acquired by a user, the shared process transitions to its linear phase, whose typing is governed by the third sub-judgment. The offered channel transitions to linear mode L, while the linear context may now contain channels at modes R or L. Finally, the fourth typing judgment types a linear process, corresponding to a *transaction* holding access to shared channels Γ and linear channels Δ, and offering at mode T.

This novel factorization upholds preservation while allowing shared contract processes to rely on linear resources. The modes impose the ordering $\mathsf{R} < \mathsf{S} < \mathsf{L} < \mathsf{T}$ among the linear channels in the configuration. A process (offering a channel) at a certain mode is allowed to rely only on processes at the same or lower mode. These are exactly the conditions imposed by Definition 1. This introduces an implicit ordering among the linear processes depending on their mode, thus eliminating cycles in the process tree. Relatedly, shared processes can only refer to shared channels (at mode S) or purely linear channels (at mode R) as exemplified by the judgment Δ $\mathsf{purelin}$ in Figure 3. Formally, Δ $\mathsf{purelin}$ denotes that for all $d_k \in \Delta \implies k = \mathsf{R}$. This ensures that a shared contract must release all processes it has acquired before itself being released. This further enforces an ordering in which the channels are acquired and released, thus *allowing contracts to interact with other contracts without compromising type safety*.

Shared session types introduce new typing rules into our system, concerning the *acquire-release* constructs (see Figure 3). In rule $\uparrow_\mathsf{L}^\mathsf{S} L$, an acquire is applied to the shared channel $x_\mathsf{S} : \uparrow_\mathsf{L}^\mathsf{S} A_\mathsf{L}$ in Γ and yields a linear channel x_L added to Δ when successful. A contract process can *accept* an acquire request along its offering shared channel x_S. After the accept is successful, the shared contract process transitions to its linear phase, now offering along the linear channel x_L (rule $\uparrow_\mathsf{L}^\mathsf{S} R$).

The synchronous dynamics of the *acquire-accept* pair is

$$(\uparrow_\mathsf{L}^\mathsf{S} C) : \mathsf{proc}(a_\mathsf{S}, w', x_\mathsf{L} \leftarrow \mathsf{accept}\ a_\mathsf{S} \ ;\ P_{x_\mathsf{L}}), \mathsf{proc}(c_m, w, x_\mathsf{L} \leftarrow \mathsf{acquire}\ a_\mathsf{S} \ ;\ Q_{x_\mathsf{L}}) \mapsto \\ \mathsf{proc}(a_\mathsf{L}, w', P_{a_\mathsf{L}}), \mathsf{proc}(c_m, w, Q_{a_\mathsf{L}})$$

This rule exploits the invariant that a contract process' providing channel a can come at two different modes, a linear one a_L, and a shared one a_S. The linear channel a_L is substituted for the channel variable x_L occurring in the process terms P and Q.

The dual to acquire-accept is *release-detach*. A client can *release* linear access to a contract process, while the contract process *detaches* from the client. The corresponding typing rules are presented in Figure 3. The effect of releasing the linear channel x_L is that the continuation Q loses access to x_L, while a new reference to x_S is made available in the shared context Γ. The contract, on the other hand, detaches from the client by transitioning its offering channel from linear mode x_L back to the shared mode x_S. Both right rules $\uparrow_\mathsf{L}^\mathsf{S} R$ and $\downarrow_\mathsf{L}^\mathsf{S} R$ require Δ $\mathsf{purelin}$ ensuring that a shared process releases all shared channels before themselves being released. Operationally, the release-detach rule is inverse to the acquire-accept rule.

$$(\downarrow_\mathsf{L}^\mathsf{S} C) : \mathsf{proc}(a_\mathsf{L}, w', x_\mathsf{S} \leftarrow \mathsf{detach}\ a_\mathsf{L} \ ;\ P_{x_\mathsf{S}}), \mathsf{proc}(c_m, w, x_\mathsf{S} \leftarrow \mathsf{release}\ a_\mathsf{L} \ ;\ Q_{x_\mathsf{S}}) \mapsto \\ \mathsf{proc}(a_\mathsf{S}, w', P_{a_\mathsf{S}}), \quad \mathsf{proc}(c_m, w, Q_{a_\mathsf{S}})$$

5 ADDING A FUNCTIONAL LAYER

To support general-purpose programming patterns, Nomos combines linear channels with conventional data structures, such as integers, lists, or dictionaries. To reflect and track different classes of data in the type system, we take inspiration from prior work [Pfenning and Griffith 2015; Toninho et al. 2013] and incorporate processes into a functional core via a *linear contextual monad* that isolates session-based concurrency. To this end, we introduce a separate functional context to the typing of a process. The linear contextual monad encapsulates open concurrent computations, which can be passed in functional computations but also transferred between processes in the form of *higher-order processes*, providing a uniform integration of higher-order functions and processes.

$\boxed{\Psi \,;\, \Gamma \,;\, \Delta \vdash^{q} P :: (x_m : A)}$ Process P uses shared channels in Γ and offers A along x.

$$\frac{\Psi \,;\, \Gamma \,;\, \Delta, (x_{\mathsf{L}} : A_{\mathsf{L}}) \vdash^{q} Q :: (z_m : C)}{\Psi \,;\, \Gamma, (x_{\mathsf{S}} : \uparrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{L}}) \,;\, \Delta \vdash^{q} x_{\mathsf{L}} \leftarrow \mathsf{acquire}\ x_{\mathsf{S}} \,;\, Q :: (z_m : C)} \uparrow_{\mathsf{L}}^{\mathsf{S}} L$$

$$\frac{\Delta\ \mathsf{purelin} \qquad \Psi \,;\, \Gamma \,;\, \Delta \vdash^{q} P :: (x_{\mathsf{L}} : A_{\mathsf{L}})}{\Psi \,;\, \Gamma \,;\, \Delta \vdash^{q} x_{\mathsf{L}} \leftarrow \mathsf{accept}\ x_{\mathsf{S}} \,;\, P :: (x_{\mathsf{S}} : \uparrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{L}})} \uparrow_{\mathsf{L}}^{\mathsf{S}} R$$

$$\frac{\Psi \,;\, \Gamma, (x_{\mathsf{S}} : A_{\mathsf{S}}) \,;\, \Delta \vdash^{q} Q :: (z_m : C)}{\Psi \,;\, \Gamma \,;\, \Delta, (x_{\mathsf{L}} : \downarrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{S}}) \vdash^{q} x_{\mathsf{S}} \leftarrow \mathsf{release}\ x_{\mathsf{L}} \,;\, Q :: (z_m : C)} \downarrow_{\mathsf{L}}^{\mathsf{S}} L$$

$$\frac{\Delta\ \mathsf{purelin} \qquad \Psi \,;\, \Gamma \,;\, \Delta \vdash^{q} P :: (x_{\mathsf{S}} : A_{\mathsf{S}})}{\Psi \,;\, \Gamma \,;\, \Delta \vdash^{q} x_{\mathsf{S}} \leftarrow \mathsf{detach}\ x_{\mathsf{L}} \,;\, P :: (x_{\mathsf{L}} : \downarrow_{\mathsf{L}}^{\mathsf{S}} A_{\mathsf{S}})} \downarrow_{\mathsf{L}}^{\mathsf{S}} R$$

Fig. 3. Typing rules corresponding to the shared layer.

The types are separated into a functional and concurrent part, mutually dependent on each other. The functional types τ are given by the type grammar below.

$$\begin{array}{rcl} \tau & ::= & \tau \to \tau \mid \tau + \tau \mid \tau \times \tau \mid \mathsf{int} \mid \mathsf{bool} \mid L^{q}(\tau) \\ & \mid & \{A_{\mathsf{R}} \leftarrow \overline{A_{\mathsf{R}}}\}_{\mathsf{R}} \mid \{A_{\mathsf{S}} \leftarrow \overline{A_{\mathsf{S}}} \,;\, \overline{A_{\mathsf{R}}}\}_{\mathsf{S}} \mid \{A_{\mathsf{T}} \leftarrow \overline{A_{\mathsf{S}}} \,;\, \overline{A}\}_{\mathsf{T}} \end{array}$$

The types are standard, except for the potential annotation $q \in \mathbb{N}$ in list type $L^{q}(\tau)$, which we explain in Section 6, and the contextual monadic types in the last line, which are the topic of this section. The expressivity of the types and terms in the functional layer are not important for the development in this paper. Thus, we do not formally define functional terms M but assume that they have the expected term formers such as function abstraction and application, type constructors, and pattern matching. We also define a standard type judgment for the functional part of the language.

$\Psi \Vdash^{p} M : \tau$ term M has type τ in functional context Ψ (potential p discussed later)

Contextual Monad. The main novelty in the functional types are the three type formers for contextual monads, denoting the type of a process expression. The type $\{A_{\mathsf{R}} \leftarrow \overline{A_{\mathsf{R}}}\}_{\mathsf{R}}$ denotes a process offering a *purely linear* session type A_{R} and using the purely linear vector of types $\overline{A_{\mathsf{R}}}$. The corresponding introduction form in the functional language is the monadic value constructor $\{c_{\mathsf{R}} \leftarrow P \leftarrow \overline{d_{\mathsf{R}}}\}$, denoting a runnable process offering along channel c_{R} that uses channels $\overline{d_{\mathsf{R}}}$, all at mode R. The corresponding typing rule for the monad is (ignore the blue portions)

$$\frac{\Delta = \overline{d_{\mathsf{R}} : D} \qquad \Psi \,;\, \cdot \,;\, \Delta \vdash^{q} P :: (x_{\mathsf{R}} : A)}{\Psi \Vdash^{q} \{x_{\mathsf{R}} \leftarrow P \leftarrow \overline{d_{\mathsf{R}}}\} : \{A \leftarrow \overline{D}\}_{\mathsf{R}}} \{\}I_{\mathsf{R}}$$

The monadic *bind* operation implements process composition and acts as the elimination form for values of type $\{A_{\mathsf{R}} \leftarrow \overline{A_{\mathsf{R}}}\}_{\mathsf{R}}$. The bind operation, written as $c_{\mathsf{R}} \leftarrow M \leftarrow \overline{d_{\mathsf{R}}} \,;\, Q_c$, composes the process underlying the monadic term M, which offers along channel c_{R} and uses channels $\overline{d_{\mathsf{R}}}$, with Q_c, which uses c_{R}. The typing rule for the monadic bind is rule $\{\}E_{\mathsf{RR}}$ in Figure 4. The linear context is split between the monad M and continuation Q, enforcing linearity. Similarly, the potential in the functional context is split using the sharing judgment ($\curlyvee$), explained in Section 6. The shared context Γ is empty in accordance with the invariants established in Definition 1 *(i)*, since the mode of offered channel x is R. The effect of executing a bind is the spawn of the purely linear process corresponding to the monad M, and the parent process continuing with Q. The corresponding operational semantics

$\boxed{\Psi \; ; \; \Gamma \; ; \; \Delta \overset{q}{\vdash} P :: (x_m : A)}$ Process P uses functional values in Ψ, and provides A along x.

$$\frac{r = p + q \qquad \Delta = \overline{d_{\mathsf{R}} : D} \qquad \Psi \curlyvee (\Psi_1, \Psi_2) \quad \Psi_1 \overset{p}{\Vdash} M : \{A \leftarrow \overline{D}\} \qquad \Psi_2 \; ; \; \cdot \; ; \; \Delta', (x_{\mathsf{R}} : A) \overset{q}{\vdash} Q :: (z_{\mathsf{R}} : C)}{\Psi \; ; \; \cdot \; ; \; \Delta, \Delta' \overset{r}{\vdash} x_{\mathsf{R}} \leftarrow M \leftarrow \overline{d_{\mathsf{R}}} \; ; \; Q :: (z_{\mathsf{R}} : C)} \; \{\}E_{\mathsf{RR}}$$

$$\frac{\Psi, (y : \tau) \; ; \; \Gamma \; ; \; \Delta \overset{q}{\vdash} P :: (x_m : A)}{\Psi \; ; \; \Gamma \; ; \; \Delta \overset{q}{\vdash} y \leftarrow \mathsf{recv} \; x_m \; ; \; P :: (x_m : \tau \to A)} \; \to R$$

$$\frac{r = p + q \qquad \Psi \curlyvee (\Psi_1, \Psi_2) \qquad \Psi_1 \overset{p}{\Vdash} M : \tau \qquad \Psi_2 \; ; \; \Gamma \; ; \; \Delta, (x_m : A) \overset{q}{\vdash} Q :: (z_k : C)}{\Psi \; ; \; \Gamma \; ; \; \Delta, (x_m : \tau \to A) \overset{r}{\vdash} \mathsf{send} \; x_m \; M \; ; \; Q :: (z_k : C)} \; \to L$$

Fig. 4. Typing rules corresponding to the functional layer.

rule (named $\mathsf{spawn}_{\mathsf{RR}}$) is given as follows:

$$\mathsf{proc}(d_{\mathsf{R}}, w, x_{\mathsf{R}} \leftarrow \{x'_{\mathsf{R}} \leftarrow P_{x'_{\mathsf{R}}, \overline{y}} \leftarrow \overline{y}\} \leftarrow \overline{a} \; ; \; Q) \mapsto \mathsf{proc}(c_{\mathsf{R}}, 0, P_{c_{\mathsf{R}}, \overline{a}}), \mathsf{proc}(d_{\mathsf{R}}, w, [c_{\mathsf{R}}/x_{\mathsf{R}}]Q)$$

The above rule spawns the process P offering along a globally fresh channel c_{R}, and using channels $\overline{a}$. The continuation process Q acts as a client for this fresh channel c_{R}. The other two monadic types correspond to spawning a shared process $\{A_{\mathsf{S}} \leftarrow \overline{A_{\mathsf{S}}} \; ; \; \overline{A_{\mathsf{R}}}\}_{\mathsf{S}}$ and a transaction process $\{A_{\mathsf{T}} \leftarrow \overline{A_{\mathsf{S}}} \; ; \; \overline{A}\}_{\mathsf{T}}$ at mode S and T, respectively. Their rules are analogous to $\{\}I_{\mathsf{R}}$ and $\{\}E_{\mathsf{RR}}$.

Value Communication. Communicating a *value* of the functional language along a channel is expressed at the type level by adding the following two session types.

$$A ::= \ldots \mid \tau \to A \mid \tau \wedge A$$

The type $\tau \to A$ prescribes receiving a value of type τ with continuation type A, while its dual $\tau \wedge A$ prescribes sending a value of type τ with continuation A. The corresponding typing rules for arrow ($\to R, \to L$) are given in Figure 4 (rules for $\wedge$ are inverse). Receiving a value adds it to the functional context Ψ, while sending it requires proving that the value has type τ. Semantically, sending a value $M : \tau$ creates a message predicate along a fresh channel c_m^+ containing the value:

$(\to S) : \mathsf{proc}(d_k, w, \mathsf{send} \; c_m \; M \; ; \; P) \mapsto \mathsf{msg}(c_m^+, 0, \mathsf{send} \; c_m \; M \; ; \; c_m^+ \leftarrow c_m), \mathsf{proc}(d_k, w, [c_m^+/c_m]P)$

The recipient process substitutes M for x, and continues to offer along the fresh continuation channel received by the message. This ensures that messages are received in the order they are sent. The rule is formalized below.

$(\to C) : \mathsf{proc}(c_m, w', x \leftarrow \mathsf{recv} \; c_m \; ; \; Q), \mathsf{msg}(c_m^+, w, \mathsf{send} \; c_m \; M \; ; \; c_m^+ \leftarrow c_m) \mapsto$
$\qquad\qquad\qquad \mathsf{proc}(c_m^+, w + w', [c_m^+/c_m][M/x]Q)$

Tracking Linear Assets. As an illustration, consider the type money introduced in the auction example (Section 2). The type is an abstraction over funds stored in a process and is described as

$\mathsf{money} = \&\{\textbf{value} : \mathsf{int} \wedge \mathsf{money},$	% send value
$\quad \textbf{add} : \mathsf{money} \multimap_{\mathsf{R}} \mathsf{money},$	% receive money and add it
$\quad \textbf{subtract} : \mathsf{int} \to \oplus\{\textbf{sufficient} : \mathsf{money} \otimes_{\mathsf{R}} \mathsf{money},$	% receive int, send money
$\qquad \textbf{insufficient} : \mathsf{money}\}$	% funds insufficient to subtract
$\quad \textbf{coins} : \mathsf{list}_{\mathsf{coin}}\}$	% send list of coins

The type supports querying for value, and addition and subtraction. The type also expresses insufficiency of funds in the case of subtraction. The provider process only supplies money to the client if the requested amount is less than the current balance, as depicted in the **sufficient** label. The type is implemented by a *wallet* process that internally stores a linear list of coins and an integer representing its value. Since linearity is only enforced on the list of coins in the linear context, we trust the programmer updates the integer in the functional context correctly during transactions. The process is typed and implemented as (modes of channels l and m is R, skipped in the definition for brevity)

```
 1: $(n : \mathsf{int})\ ;\ (l_{\mathsf{R}} : \mathsf{list}_{\mathsf{coin}}) \vdash \mathit{wallet} :: (m_{\mathsf{R}} : \mathsf{money})$
 2:   $m \leftarrow \mathit{wallet}\ n \leftarrow l =$
 3:     case $m$                                      % case analyze on label received on $m$
 4:       ($\mathbf{value} \Rightarrow$ send $m\ n$ ;                % receive value, send $n$
 5:               $m \leftarrow \mathit{wallet}\ n \leftarrow l$
 6:       | $\mathbf{add} \Rightarrow m' \leftarrow$ recv $m$ ;               % receive $m'$ : money to add
 7:             $m'.\mathbf{value}$ ;                      % query value of $m'$
 8:             $v \leftarrow$ recv $m'$ ;
 9:             $m'.\mathbf{coins}$ ;                      % extract list of coins stored in $m'$
10:             $k \leftarrow \mathit{append} \leftarrow l\ m'$ ;              % append list received to internal list
11:             $m \leftarrow \mathit{wallet}\ (n + v) \leftarrow k$
12:       | $\mathbf{subtract} \Rightarrow n' \leftarrow$ recv $m$ ;          % receive int to subtract
13:                 if $(n' > n)$ then
14:                   $m.\mathbf{insufficient}$ ;         % funds insufficient
15:                   $m \leftarrow \mathit{wallet}\ n \leftarrow l$
16:                 else
17:                   $m.\mathbf{sufficient}$ ;           % funds sufficient
18:                   $l' \leftarrow \mathit{remove}\ n' \leftarrow l$ ;    % remove $n'$ coins from $l$
19:                   $k \leftarrow$ recv $l'$ ;              % and create its own list
20:                   $m' \leftarrow \mathit{wallet}\ n' \leftarrow k$ ;    % new *wallet* process for subtracted funds
21:                   send $m\ m'$ ;                 % send new money channel to client
22:                   $m \leftarrow \mathit{wallet}\ (n - n') \leftarrow l'$
23:       | $\mathbf{coins} \Rightarrow m \leftarrow l$)
```

If the *wallet* process receives the message value, it sends back the integer n, and recurses (lines 4 and 5). If it receives the message add followed by a channel of type money (line 6), it queries the value of the received money m' (line 7), stores it in v (line 8), extracts the coins stored in m' (line 9), and appends them to its internal list of coins (line 10). Similarly, if the *wallet* process receives the message subtract followed by an integer, it compares the requested amount against the stored funds. If the balance is insufficient, it sends the corresponding label, and recurses (lines 14 and 15). Otherwise, it removes n' coins using the *remove* process (line 18), creates a money abstraction using the *wallet* process (line 20), sends it (line 21) and recurses. Finally, if the *wallet* receives the message coins, it simply forwards its internal list along the offered channel.

6 TRACKING RESOURCE USAGE

Resource usage is particularly important in digital contracts: Since multiple parties need to agree on the result of the execution of a contract, the computation is potentially performed multiple times or by a trusted third party. This immediately introduces the need to prevent denial of service attacks and to distribute the cost of the computation among the participating parties.

The predominant approach for smart contracts on blockchains like Ethereum is not to restrict the computation model but to introduce a cost model that defines the *gas* consumption of low level operations. Any transaction with a smart contract needs to be executed and validated before adding it to the global distributed ledger, i.e., blockchain. This validation is performed by *miners*, who charge fees based on the gas consumption of the transaction. This fee has to be estimated and provided by the sender prior to the transaction. If the provided amount does not cover the gas cost, the money falls to the miner, the transaction fails, and the state of the contract is reverted back. Overestimates bear the risk of high losses if the contract has flaws or vulnerabilities.

It is not trivial to decide on the right amount for the fee since the gas cost of the contract does not only depend on the requested transaction but also on the (a priori unknown) state of the blockchain. Thus, precise and static estimation of gas cost facilitates transactions and reduces risks. We discuss our approach of tracking resource usage, both at the functional and process layer.

Functional Layer. Numerous techniques have been proposed to statically derive resource bounds for functional programs [Avanzini et al. 2015; Cicek et al. 2017; Danner et al. 2015; Lago and Gaboardi 2011; Radiček et al. 2017]. In Nomos, we adapt the work on automatic amortized resource analysis (AARA) [Hoffmann et al. 2011; Hofmann and Jost 2003] that has been implemented in Resource Aware ML (RaML) [Hoffmann et al. 2017]. RaML can automatically derive worst-case resource bounds for higher-order polymorphic programs with user-defined inductive types. The derived bounds are multivariate resource polynomials of the size parameters of the arguments. AARA is parametric in the resource metric and can deal with non-monotone resources like memory that can become available during the evaluation.

As an illustration, consider the function *applyInterest* that iterates over a list of balances and applies interest on each element, multiplying them by a constant c. We use *tick* annotations to define the resource usage of an expression in this article. We have annotated the code to count the number of multiplications. The resource usage of an evaluation of *applyInterest* b is $|b|$.

```
let applyInterest balances =
 match balances with
 | [] -> []
 | hd::tl -> tick(1); (c*hd)::(applyInterest tl) (* consume unit potential for tick *)
```

The idea of AARA is to decorate base types with potential annotations that define a potential function as in amortized analysis. The typing rules ensure that the potential before evaluating an expression is sufficient to cover the cost of the evaluation and the potential defined by the return type. This posterior potential can then be used to pay for resource usage in the continuation of the program. For example, we can derive the following resource-annotated type.

$$\mathit{applyInterest} : L^1(\mathsf{int}) \xrightarrow{0/0} L^0(\mathsf{int})$$

The type $L^1(\mathsf{int})$ denotes a list of integers assigning a unit potential to each element in the list. The return value, on the other hand, has no potential. The annotation on the function arrow indicates that we do not need any potential to call the function and that no constant potential is left after the function call has returned.

In a larger program, we might want to call the function *applyInterest* again on the result of a call to the function. In this case, we would need to assign the type $L^1(\mathsf{int})$ to the resulting list and require $L^2(\mathsf{int})$ for the argument. In general, the type for the function can be described with symbolic annotations with linear constraints between them. To derive a worst-case bound for a function the constraints can be solved by an off-the-shelf LP solver, even if the potential functions are polynomial [Hoffmann et al. 2011, 2017].

In Nomos, we simply adopt the standard typing judgment of AARA for functional programs.

$$\Psi \Vdash^{q} M : \tau$$

It states that under the resource-annotated functional context Ψ, with constant potential q, the expression M has the resource-aware type τ.

The operational *cost* semantics is defined by the judgment

$$M \Downarrow V \mid \mu$$

which states that the closed expression M evaluates to the value V with cost μ. The type soundness theorem states that if $\cdot \Vdash^{q} M : \tau$ and $M \Downarrow V \mid \mu$ then $q \geq \mu$.

More details about AARA can be found in the literature [Hoffmann et al. 2017; Hofmann and Jost 2003] and the supplementary material.

Process Layer. To bound the resource usage of a process, Nomos features resource-aware session types [Das et al. 2018b] for work analysis. Resource-aware session types describe resource contracts for inter-process communication. The type system supports amortized analysis by assigning potential to both messages and processes. The derived resource bounds are functions of interactions between processes. As an illustration, consider the following resource-aware list interface from prior work [Das et al. 2018b].

$$\mathsf{list}_A = \oplus\{\mathsf{nil}^0 : \mathbf{1}^0, \mathsf{cons}^1 : A \overset{0}{\otimes} \mathsf{list}_A\}$$

The type prescribes that the provider of a list must send one unit of potential with every cons message that it sends. Dually, a client of this list will receive a unit potential with every cons message. All other type constructors are marked with potential 0, and exchanging the corresponding messages does not lead to transfer of potential.

While resource-aware session types in Nomos are equivalent to the existing formulation [Das et al. 2018b], our version is simpler and more streamlined. Instead of requiring every message to carry a potential (and potentially tagging several messages with 0 potential), we introduce two new type constructors for exchanging potential.

$$A ::= \ldots \mid \triangleright^r A \mid \triangleleft^r A$$

The type $\triangleright^r A$ requires the provider to pay r units of potential which are transferred to the client. Dually, the type $\triangleleft^r A$ requires the client to pay r units of potential that are received by the provider. Thus, the reformulated list type becomes

$$\mathsf{list}_A = \oplus\{\mathsf{nil} : \mathbf{1}, \mathsf{cons} : \triangleright^1 (A \otimes \mathsf{list}_A)\}$$

The reformulation is more compact since we need to account for potential in only the typing rules corresponding to $\triangleright^r A$ and $\triangleleft^r A$.

With all aspects introduced, the process typing judgment

$$\Psi \; ; \; \Gamma \; ; \; \Delta \vDash^{q} P :: (x_m : A)$$

denotes a process P accessing functional variables in Ψ, shared channels in Γ, linear channels in Δ, offers service of type A along channel x at mode m and stores a non-negative constant potential q. Similarly, the expressing typing judgment

$$\Psi \Vdash^{p} M : \tau$$

denotes that expression M has type τ in the presence of functional context Ψ and potential p.

Figure 5 shows the rules that interact with the potential annotations. In the rule $\triangleleft R$, process P storing potential q receives r units along the offered channel $x_m : \triangleleft^r A$ using the *get* construct and the continuation executes with $p = q + r$ units of potential. In the dual rule $\triangleleft L$, a process storing potential $q = p + r$ sends r units along the channel $x_m : \triangleleft^r A$ in Δ using the *pay* construct, and the continuation remains with p units of potential. The typing rules for the dual constructor $\triangleright^r A$ are the exact inverse. Finally, executing the tick (r) construct consumes r potential from the stored process potential q, and the continuation remains with $p = q - r$ units, as described in the tick rule.

The tick construct is used to simulate a cost model in Nomos. If an operation (e.g., sending a message, calling a function, etc.) has a cost of r, this cost is simulated by inserting tick (r) just before the operation. Then, the tick operations are the only ones that cost potential, thus simplifying the type system. These tick operations are automatically inserted by the Nomos type checker, using a predefined cost model that assigns a constant cost to each operation. In addition, our implementation provides some standard cost models, for instance, that assign a unit cost to each internal operation and sending a message.

$\boxed{\Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{q} P :: (x_m : A)}$ Process P has potential q and provides type A along channel x.

$$\frac{p = q + r \qquad \Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{p} P :: (x_m : A)}{\Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{q} \mathsf{get}\ x_m\ \{r\} \; ; \; P :: (x_m : \triangleleft^{r} A)} \triangleleft R \qquad \frac{q = p + r \qquad \Psi \; ; \; \Gamma \; ; \; \Delta, (x_m : A) \vdash^{p} P :: (z_k : C)}{\Psi \; ; \; \Gamma \; ; \; \Delta, (x_m : \triangleleft^{r} A) \vdash^{q} \mathsf{pay}\ x_m\ \{r\} \; ; \; P :: (z_k : C)} \triangleleft L$$

$$\frac{q = p + r \qquad \Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{p} P :: (x_m : A)}{\Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{q} \mathsf{tick}\ (r) \; ; \; P :: (x_m : A)} \ \mathsf{tick}$$

Fig. 5. Selected typing rules corresponding to potential.

Integration. Since both AARA for functional programs and resource-aware session types are based on the integration of the potential method into their type systems, their combination is natural. The two points of integration of the functional and process layer are (i) spawning a process, and (ii) sending/receiving a value from the functional layer. Recall the spawn rule $\{\}E_{\mathsf{RR}}$ from Figure 4. A process storing potential $r = p + q$ can spawn a process corresponding to the monadic expression M, if M needs p units of potential to evaluate, while the continuation needs q units of potential to execute. Moreover, the functional context Ψ is shared in the two premises as Ψ_1 and Ψ_2 using the judgment $\Psi \curlyvee (\Psi_1, \Psi_2)$. This judgment, already explored in prior work [Hoffmann et al. 2017] describes that the base types in Ψ are copied to both Ψ_1 and Ψ_2, but the potential is split up. For instance, $L^{q_1+q_2}(\tau) \curlyvee (L^{q_1}(\tau), L^{q_2}(\tau))$. The rule $\rightarrow L$ in Figure 4 follows a similar pattern. A process Q storing $r = p + q$ potential sends a monadic expression M needing p units of potential to evaluate, and the continuation remains with q units of potential to execute. The p units of potential are consumed to evaluate M to a value before sending since only values are exchanged at runtime. Thus, the combination of the two type systems is smooth, assigning a uniform meaning to potential, both for the functional and process layer. Remarkably, this technical device of exchanging functional values can be used to exchange non-constant potential with messages. For instance, exchanging a list $l : L^{q}(\tau)$ will exchange $q \cdot n$ units of potential where n is the size of the list l.

Operational Cost Semantics. The resource usage of a process (or message) is tracked in semantic objects $\mathsf{proc}(c, w, P)$ and $\mathsf{msg}(c, w, N)$ using the local counters w. This signifies that the process P (or message N) has performed *work* w so far. The rules of semantics that explicitly affect the work counter are

$$\frac{M \Downarrow V \mid \mu}{\mathsf{proc}(c_m, w, P[M]) \mapsto \mathsf{proc}(c_m, w + \mu, P[V])} \ \mathsf{internal}$$

This rule describes that if an expression M evaluates to V with cost μ, then the process $P[M]$ depending on monadic expression M steps to $P[V]$, while the work counter increments by μ, denoting the total number of internal steps taken by the process. At the process layer, the work increments on executing a *tick* operation.

$$\mathsf{proc}(c_m, w, \mathsf{tick}\ (\mu) \; ; \; P) \mapsto \mathsf{proc}(c_m, w + \mu, P)$$

A new process (or message) is spawned with $w = 0$, and a terminating process transfers its work to the corresponding message it interacts with before termination, thus preserving the total work performed by the system.

7 TYPE SOUNDNESS

The main theorems that exhibit the connections between our type system and the operational cost semantics are the usual *type preservation* and *progress*. First, Definition 1 asserts certain invariants on process typing judgment depending on the mode of the channel offered by a process. This mode, remains invariant, as the process evolves. This is ensured by the process typing rules, which remarkably preserve these invariants despite being parametric in the mode.

Lemma 1 (Invariants). *The typing rules on the judgment $\Psi \; ; \; \Gamma \; ; \; \Delta \vdash^{q} (x_m : A)$ preserve the invariants outlined in Definition 1, i.e., if the conclusion satisfies the invariant, so do all the premises.*

***Configuration Typing*.** At run-time, a program evolves into a number of processes and messages, represented by proc and msg predicates. This multiset of predicates is referred to as a *configuration* (abbreviated as Ω).

$$\Omega ::= \cdot \mid \Omega, \mathsf{proc}(c, w, P) \mid \Omega, \mathsf{msg}(c, w, N)$$

A key question is how to type these configurations because a configuration both uses and provides a number of channels. The solution is to have the typing impose a partial order among the processes and messages, requiring the provider of a channel to appear before its client. We stipulate that no two distinct processes or messages in a well-formed configuration provide the same channel c.

The typing judgment for configurations has the form $\Sigma \; ; \; \Gamma_0 \overset{E}{\vDash} \Omega :: (\Gamma \; ; \; \Delta)$ defining a configuration Ω providing shared channels in Γ and linear channels in Δ. Additionally, we need to track the mapping between the shared channels and their linear counterparts offered by a contract process, switching back and forth between them when the channel is acquired or released respectively. This mapping, along with the type of the shared channels, is stored in Γ_0. E is a natural number and stores the sum of the total potential and work as recorded in each process and message. We call E the energy of the configuration. The supplement details the configuration typing rules.

Finally, Σ denotes a signature storing the type and function definitions. A signature is well-formed if *(i)* every type definition $V = A_V$ is *contractive* [Gay and Hole 2005] and *(ii)* every function definition $f = M : \tau$ is well-typed according to the expression typing judgment $\Sigma \; ; \; \cdot \Vdash^{p} M : \tau$. The signature does not contain process definitions; every process is encapsulated inside a function using the contextual monad.

Theorem 1 (Type Preservation).

- *If a closed well-typed expression* $\cdot \Vdash^{q} M : \tau$ *evaluates to a value, i.e.,* $M \Downarrow V \mid \mu$*, then* $q \geq \mu$ *and* $\cdot \Vdash^{q-\mu} V : \tau$.
- *Consider a closed well-formed and well-typed configuration* Ω *such that* $\Sigma \; ; \; \Gamma_0 \overset{E}{\vDash} \Omega :: (\Gamma \; ; \; \Delta)$. *If the configuration takes a step, i.e.* $\Omega \mapsto \Omega'$*, then there exist* Γ_0', Γ' *such that* $\Sigma \; ; \; \Gamma_0' \overset{E}{\vDash} \Omega' :: (\Gamma' \; ; \; \Delta)$*, i.e., the resulting configuration is well-typed. Additionally,* $\Gamma_0 \subseteq \Gamma_0'$ *and* $\Gamma \subseteq \Gamma'$.

The preservation theorem is standard for expressions [Hoffmann et al. 2017]. For processes, we proceed by induction on the operational cost semantics and inversion on the configuration and process typing judgment.

To state progress, we need the notion of a *poised* process [Pfenning and Griffith 2015]. A process $\mathsf{proc}(c_m, w, P)$ is poised if it is trying to receive a message on c_m. Dually, a message $\mathsf{msg}(c_m, w, N)$ is poised if it is sending along c_m. A configuration is poised if every message or process in the configuration is poised. Intuitively, this means that the configuration is trying to interact with the outside world along a channel in Γ or Δ. Additionally, a process can be *blocked* [Balzer and Pfenning 2017] if it is trying to acquire a contract process that has already been acquired by some process. This can lead to the possibility of deadlocks.

Theorem 2 (Progress). *Consider a closed well-formed and well-typed configuration* Ω *such that* $\Gamma_0 \overset{E}{\vDash} \Omega :: (\Gamma \; ; \; \Delta)$. *Either* Ω *is poised, or it can take a step, i.e.,* $\Omega \mapsto \Omega'$*, or some process in* Ω *is blocked along* a_{S} *for some shared channel* a_{S} *and there is a process* $\mathsf{proc}(a_{\mathsf{L}}, w, P) \in \Omega$.

The progress theorem is weaker than that for binary linear session types, where progress guarantees deadlock freedom due to absence of shared channels.

8 IMPLEMENTATION AND EVALUATION

We have developed an open-source prototype implementation [Nom 2019] of Nomos in OCaml. This prototype contains a lexer and parser (929 lines of code), a type checker (2388 lines of code), a pretty printer (451 lines of code), an LP solver interface (915 lines of code) and an interpreter (1286 lines of code) for implementing, type checking and executing Nomos programs. We also describe our efforts to simplify programming and improve accessiblity of Nomos to developers.

Syntax. The lexer and parser for Nomos have been implemented in Menhir [Pottier and Régis-Gianas 2019], an LR(1) parser generator for OCaml. A Nomos program is a list of mutually recursive type and process definitions. To visually separate out functional variables from session-typed channels, we require that shared channels are prefixed by #, while linear channels are prefixed by $. This avoids confusion between the two, both for the programmer and the parser. We also require the programmer to indicate the *mode* of the process being defined: *asset*, *contract* or *transaction*, assigning the respective modes R, S and T to the offered channel. The modes for all other channels are inferred automatically (explained later). The initial potential $\{q\}$ of a process is marked on the turnstile in the declaration. The syntax for definitions is

```
type v = A
proc <mode> f : (x1 : T), ($c2 : A), ... |{q}- ($c : A) = M
```

In the context, T is the functional type for variable $\mathsf{x1}$, while A is the session type for channel $\mathsf{\$c2}$ and M is a functional expression implementing the process. We add syntactic sugar, such as the forms $\mathsf{let}\ \mathsf{x} = \mathsf{M}; \mathsf{P}$ and $\mathsf{if}\ \mathsf{M}\ \mathsf{then}\ \mathsf{P}_1\ \mathsf{else}\ \mathsf{P}_2$, to the process layer to ease programming. Finally, a functional expression can enter the session type monad using $\{\}$, i.e., $\mathsf{M} = \{\mathsf{P}\}$ where P is a session-typed expression.

Type Checking. We implemented a bi-directional [Pierce and Turner 2000] type checker with a specific focus on the quality of error messages, which include, for example, *extent* (source code location) information for each definition and expression. The programmer provides the initial type of each variable and channel in the declaration and the definition is checked against it, while reconstructing the intermediate types. This helps localize the source of a type error as the point where type reconstruction fails. Type equality is restricted to reflexivity (constant time), although we have also implemented the standard co-inductive algorithm [Gay and Hole 2005] which is quadratic in the size of type definitions. For all our examples, the reflexive notion of equality was sufficient. *Type checking is linear time in the size of the program*, which is important in the blockchain domain where type checking can be part of the attack surface.

Potential and Mode Inference. The potential and mode annotations are the most interesting aspects of the Nomos type system. Since modes are associated with each channel, they are tedious to write. Similarly, the exact potential annotations depend on the cost assigned to each operation and is difficult to predict statically. Thus, we implemented an automatic inference algorithm for both these annotations by relying on an off-the-shelf LP solver.

Using ideas from existing techniques for type inference for AARA [Hoffmann et al. 2017; Hofmann and Jost 2003], we reduce the reconstruction of potential annotations to linear optimization. To this end, Nomos' inference engine uses the Coin-Or LP solver. In a Nomos program, the programmer can indicate unknown potential using $*$. Thus, resource-aware session types can be marked with $\triangleright^*$ and $\triangleleft^*$, list types can be marked as $L^*(\tau)$ and process definitions can be marked with $|\{*\}-$ on the turnstile. The mode of all the channels is marked as 'unknown' while parsing.

The inference engine iterates over the program and substitutes the star annotations with potential variables and 'unknown' with mode variables. Then, the bidirectional typing rules are applied, approximately checking the program (modulo potential and mode annotations) while also generating linear constraints for potential annotations (see Figure 4). and mode annotations (see Definition 1 and Figure 3). Finally, these constraints are shipped to the LP solver, which minimizes the value of the potential annotations to achieve tight bounds. The LP solver either returns that the constraints are infeasible, or returns a satisfying assignment, which is then substituted into the program. The final program is pretty printed for the programmer to view and verify the potential and mode annotations.

8.1 Case Studies

We evaluate the design of Nomos by implementing several smart contract applications and discussing the typical issues that arise. All the contracts are implemented and type checked in the prototype implementation and the potential and mode annotations are derived automatically by the inference engine. The cost model used for these examples assigns 1 unit of cost to every atomic internal computation and sending of a message. We show the contract types from the implementation with the following ASCII format: i) `/\` for $\uparrow_{\mathsf{L}}^{\mathsf{S}}$, ii) `\/` for $\downarrow_{\mathsf{L}}^{\mathsf{S}}$, iii) `<{q}|` for $\triangleleft^{q}$, iv) `|{q}>` for $\triangleright^{q}$, v) `^` for $\wedge$, vi) `*[m]` for $\otimes_m$, vii) `-o[m]` for $\multimap_m$.

ERC-20 Token Standard. ERC-20 [ERC 2018] is a technical standard for smart contracts on the Ethereum blockchain that defines a common list of standard functions that a token contract has to implement. The majority of tokens on the Ethereum blockchain are ERC-20 compliant.

The ERC-20 token contract implements the following session type in Nomos:

```
type erc20token = /\ <{11}| &{
  totalSupply : int ^ |{9}> \/ erc20token,
  balanceOf : id -> int ^ |{8}> \/ erc20token,
  transfer : id -> id -> int -> |{0}> \/ erc20token,
  approve : id -> id -> int -> |{6}> \/ erc20token,
  allowance : id -> id -> int ^ |{6}> \/ erc20token }
```

The type ensures that the token implements the protocol underlying the ERC-20 standard. To query the total number of tokens in supply, a client sends the totalSupply label, and the contract sends back an integer. If the contract receives the balanceOf label followed by the owner's identifier, it sends back an integer corresponding to the owner's balance. A balance transfer can be initiated by sending the transfer label to the contract followed by sender's and receiver's identifier, and the amount to be transferred. If the contract receives approve, it receives the two identifiers and the value, and updates the allowance internally. Finally, this allowance can be checked by issuing the allowance label, and sending the owner's and spender's identifier.

A programmer can design their own implementation (contract) of the erc20token session type. In our implementation, we store two hash maps, one for the balance of each account, and one for the allowance between each pair of accounts. The contract relies on custom linear coins that are used exclusively for exchanges among the private accounts. These coins can be minted by a special transaction that can only be issued by the owner of the contract and that creates coins out of thin air (consuming gas to create coins). We use a built-in type to represent a single coin, providing custom functions to *mint* and *burn* a coin. The type for the two hash maps is described below.

```
type balance-map = &{ get-balance : id -> int ^ balance-map,
                   transfer : id -> id -> int -> balance-map}
type allowance-map = &{ get : id -> id -> int ^ allowance-map,
                    set : id -> id -> int -> allowance-map}
```

The type balance−map supports two functionalities: querying the balance value of an account by receiving an id and responding with an int; and allowing a transfer by receiving the sender and receiver ids and the transfer amount. In each case, the type recurses back to balance−map allowing other users to interact with the hash map. The allowance−map type stores the allowances for each pair of accounts, which can be queried and updated using the get and set functionalities. They have a similar communication protocol as the balance−map.

Another implementation can use a different linear type with its own introduction and elimination forms for minting and burning, respectively. Nomos' linear type system enforces that the coins are treated linearly modulo minting and burning.

Hacker Gold (HKG) Token. The HKG token is one particular implementation of the ERC-20 token specification. Recently, a vulnerability was discovered in the HKG token smart contract based on a typographical error leading to a re-issuance of the entire token [HKG 2017]. When updating the receiver's balance during a transfer, instead of writing `balance+=value`, the programmer mistakenly wrote `balance=+value` (semantically meaning balance=value). Moreover, while testing this error was missed, because the first transfer always succeeds (since the two statements are semantically equivalent when balance=0. Nomos' type system would have caught the linearity violation in the latter statement that drops the existing balance in the recipient's account.

Puzzle Contract. This contract, taken from prior work [Luu et al. 2016] rewards users who solve a computational puzzle and submit the solution. The contract allows two functions, one that allows the owner to update the reward, and the other that allows a user to submit their solution and collect the reward.

In Nomos, this contract is implemented to offer the type

```
type puzzle = /\ <{14}| &{
  update : id -> money -o[R] |{0}> \/ puzzle,
  submit : int ^ &{ success : int -> money *[R] |{5}> \/ puzzle,
                   failure : |{9}> \/ puzzle } }
```

The contract still supports the two transactions. To update the reward, it receives the update label and an identifier, verifies that the sender is the owner, receives money from the sender, and acts like a puzzle again. The transaction to submit a solution has a *guard* associated with it. First, the contract sends an integer corresponding to the reward amount, the user then verifies that the reward matches the expected reward (the guard condition). If this check succeeds, the user sends the success label, followed by the solution, receives the winnings, and the session terminates. If the guard fails, the user issues the failure label and immediately terminates the session. Thus, acquire-release discipline along with the guarded session type guarantees that the user submitting the solution receives their expected winnings.

Voting. The voting contract provides a ballot type in an election.

```
type ballot = /\ <{16}| +{
  open : id -> +{ vote : id -> |{0}> \/ ballot,
                 novote : |{9}> \/ ballot },
  closed : id ^ |{13}> \/ ballot }
```

This contract allows voting when the election is **open** by receiving the candidate's *id*. To only allow legitimate voters to cast a ballot and prevent double voting by the same voter, the contract then checks if the voter is eligible to vote. It then replies with **vote** or **novote** depending on their eligibility. Once the election closes (the **closed** label), the contract can be acquired to check the winner of the election. We use two implementations for the contract: the first stores a counter for each candidate that is updated after each vote is cast (voting in Table 2); the second does not use a counter but stores potential inside the vote list that is consumed for counting the votes at the end (voting-aa in Table 2). This stored potential is provided by the voter to amortize the cost of counting. The type above shows the potential annotations corresponding to the latter.

Insurance. Nomos has been carefully designed to allow inter-contract communication without compromising type safety. We illustrate this feature using an insurance contract that processes flight delay insurance claims after verifying them with a trusted third party. The insurer and third party verifier are implemented as separate contracts providing the following session types.

```
type insurer = /\ <{6}| &{
  submit : claim -> +{ success : money *[R] |{0}> \/ insurer,
                    failure : |> \/ insurer } }
type verifier = /\ <{3}| &{
  verify : claim -> +{ valid : |{0}> \/ verifier,
                    invalid : |{0}> \/ verifier } }
```

Contract	LOC	Defs	Procs	T (ms)	Vars	Cons	I (ms)	Gap
auction	176	5	10	0.558	229	730	5.225	3
ERC 20	136	4	2	0.579	161	561	4.317	6
puzzle	108	3	7	0.410	126	389	8.994	8
voting	101	3	6	0.324	109	351	3.664	0
voting-aa	101	3	7	0.346	140	457	3.926	0
insurance	56	3	2	0.299	76	224	8.289	0
escrow	85	2	2	0.404	95	321	3.816	3
bank	147	4	5	0.663	173	561	4.549	0
wallet	30	3	2	0.231	32	102	3.224	0

Table 2. Evaluation of Nomos with Case Studies. LOC = lines of code; Defs = #type definitions; Procs = #process definitions; T (ms) = type checking time in ms; Vars = #potential and mode variables generated during type checking; Cons = #constraints generated during type checking; I (ms) = potential and mode inference time in ms; Gap = maximal gas bound gap.

The insurer type provides the option to **submit** a claim by receiving it and responds with **success** or **failure** depending upon verification of the claim. If the claim is successful, the insurer sends over the reimbursement in the form of money. The verifier type provides the option to **verify** a claim by receiving it and responding with **valid** or **invalid** depending on the validity of the claim.

The insurer, upon receiving a claim, acquires the verifier and sends it the claim details. If the claim is valid, then it responds with **success**, sends the money and detaches from its client. If the claim is invalid, it responds with **failure** and immediately detaches from its client.

8.2 Experimental Evaluation

We implemented 8 case studies in Nomos. We have already discussed the auction (Section 2), ERC 20, puzzle, voting and insurance contracts. The other case studies are:

- An escrow to exchange bonds between two parties.
- A bank account that allows users to create accounts, make deposits and withdrawals and check their balance relying on custom linear coins.
- A wallet allowing users to store money on the blockchain.

Table 2 contains a compilation of our experiments with the case studies and the prototype implementation. The experiments were run on an Intel Core i5 2.7 GHz processor with 16 GB 1867 MHz DDR3 memory. It presents the contract name, its lines of code (LOC), the number of type (Defs) and process definitions (Procs), the type checking time (T (ms)), number of potential and mode variables introduced (Vars), number of potential and mode constraints that were generated while type checking (Cons) and the time the LP solver took to infer their values (I (ms)). The last column describes the maximal gap between the static gas bound inferred and the actual runtime gas cost. It accounts for the difference in the gas cost in different program paths. However, this waste is clearly marked in the program by explicit *tick* instructions so the programmer is aware of this runtime gap, based on the program path executed.

The evaluation shows that the type-checking overhead is less than a millisecond for case studies. This indicates that Nomos is applicable to settings like distributed blockchains in which type checking could add significant overhead and could be part of the attack surface. Type inference is also efficient but an order of magnitude slower than type checking. This is acceptable since inference is only performed once during deployment of the contract. Gas bounds are tight in most cases. Loose gas bounds are caused by conditional branches with different gas cost. In practice, this is not a major concern since the Nomos semantics tracks the exact gas cost, and a user will not be overcharged for their transaction. Moreover, Nomos' type system can be easily modified to only allow contracts with tight bounds.

Our implementation experience revealed that describing the session type of a contract crystallizes the important aspects of its protocol. Often, subtle aspects of a contract are revealed while defining the protocol as a session type. Once the type is defined, the implementation simply *follows* the type protocol. The error messages from the type checker were helpful in ensuring linearity of assets and adherence to the protocol. Using $*$ for potential annotations meant we could remain unaware of the exact gas cost of operations. The syntactic sugar constructs reduced the programming overhead and the size of the contract implementations.

9 BLOCKCHAIN INTEGRATION

To integrate Nomos with a blockchain, we need a mechanism to *(i)* represent the contracts and their addresses in the current blockchain state, *(ii)* create and send transactions to the appropriate addresses, and most importantly, *(iii)* construct the global distributed ledger, which stores the history of all transactions.

Nomos on a Blockchain. We assume a blockchain like Ethereum that contains a set of Nomos contracts $C_1, \ldots, C_n$ together with their type information $\cdot \; ; \; \Gamma^i \; ; \; \Delta^i_{\mathsf{R}} \vdash^{q_i} C_i :: (x^i_{\mathsf{S}} : A^i_{\mathsf{S}})$. The shared context Γ^i types the shared contracts that C_i refers to, and the linear context Δ^i_{R} types the contract's linear assets. The channel name x^i_{S} of a contract is its address and has to be globally unique. We allow contracts to carry potential given by the annotation q_i and the potential defined by the annotations in Δ^i_{R} but the type system could easily be altered to suppress the stored potential.

These contracts form a stuck configuration (a valid virtual blockchain state) typed as

$$\Sigma \; ; \; \Gamma \overset{E}{\vDash} \mathsf{proc}(x^1_{\mathsf{S}}, w_1, C_1) \ldots \mathsf{proc}(x^n_{\mathsf{S}}, w_n, C_n) :: (\Gamma \; ; \; \cdot)$$

where $\Gamma = (x^1_{\mathsf{S}} : A^1_{\mathsf{S}}), \ldots, (x^n_{\mathsf{S}} : A^n_{\mathsf{S}})$ and $E = \Sigma^n_{i=1} q_i + w_i$ is the total energy, that is, the sum of the stored potential and previously performed work. To perform a transaction with a contract, a user submits a transaction script Q (a process) that is well-typed with respect to the existing contracts:

$$\cdot \; ; \; \Gamma \; ; \; \cdot \vdash^{q} Q :: (x_{\mathsf{T}} : \mathbf{1})$$

We mandate that the transaction offers along a channel of type $\mathbf{1}$ and terminates by sending a close message on its offered channel. This approach enables dynamic deadlock detection (explained later) and allows abortion of a transaction if a deadlock is detected. This script process is added to the set of contracts and the new (closed) configuration is typed as

$$\Sigma \; ; \; \Gamma \overset{E+q}{\vDash} \mathsf{proc}(x^1_{\mathsf{S}}, w_1, C_1) \ldots \mathsf{proc}(x_{\mathsf{T}}, 0, Q) :: (\Gamma \; ; \; (x_{\mathsf{T}} : \mathbf{1}))$$

This configuration then steps according to the Nomos semantics, ending with the termination of the script Q, leaving the configuration in a stuck state again to start a new transaction. If type checking were too costly here, that can lead to yet another source of denial-of-service attacks. In Nomos however, type checking is linear time in the size of the script.

A transaction script is connected to the blockchain state using a server process. This process, named bc−server stores the entire transaction history and offers along channel bc : tx_interface where the transaction code is received and relayed to the blockchain state. It is defined as follows.

```
type tx_code = {1}        type tx_queue = list tx_code
stype tx_interface = tx_code ⊸ tx_interface
$(txns : \mathsf{tx\_queue}) \; ; \; \cdot \; ; \; \cdot \vdash^{0} \mathsf{bc{-}server} :: (bc : \mathsf{tx\_interface})$
  $bc \leftarrow \mathsf{bc{-}server}\; txns =$
      $tx \leftarrow \mathsf{recv}\; bc \; ; \; x_{\mathsf{T}} \leftarrow tx \; ; \; \mathsf{wait}\; x_{\mathsf{T}} \; ;$
      $bc \leftarrow \mathsf{bc{-}server}\; (tx :: txns)$
```

The transaction script is packaged as a value of the contextual monadic type introduced in Section 5. For instance, the transaction Q is packaged as $\{x_\mathsf{T} \leftarrow Q\} : \{\mathbf{1}\}$ = tx_code. The bc−server process receives this code, spawns a process corresponding to it and waits for the transaction to terminate (line 5). Note that the transaction is required to terminate with a (close x_T) message which matches with the (wait x_T) being executed by the server, ensuring the execution order of the transactions. Finally, the latest transaction is added to the queue of transactions $txns$: type tx_queue = list tx_code, and the bc−server process recurses.

A transaction can either create new contracts or update the state of existing ones. In the former case, new contracts are added to the blockchain state, making them visible in the type of the configuration for subsequent transactions to access. In the latter case, it *acquires* the contracts it wishes to interact with, followed by an update in the contracts' internal state and *releases* them. Since the contract types are equi-synchronizing, they remain unchanged at the end of transaction execution. This ensures that the subsequent transactions can access the same contracts at the same type. In the future we plan to allow *sub-synchronizing* types that enable a client to release a contract channel not at the same type, but a *subtype*. The subtype can then describe the phase of the contract. For instance, the ended phase of auction contract will be a subtype of the running phase.

Deterministic Execution. Since blockchains rely on consensus among the miners, it is important to ensure deterministic execution of transactions. However, Nomos semantics has one source of non-determinism: the *acquire-accept* rule where an accepting contract latches on to any acquiring transaction. One simple approach to resolve this non-determinism is to determinize the resource scheduler based on some heuristics. Another promising approach is *record-and-replay* [Lidbury and Donaldson 2019; Ronsse and De Bosschere 1999]. The miner records the order in which the contracts are acquired in the ledger, which is then replayed by others to compute the current blockchain state.

Deadlocks. The only language specific reason a transaction can fail is a deadlock in the transaction code. Our progress theorem accounts for this possibility of deadlocks. Since a valid blockchain state represents a stuck configuration of a particular form (only shared contracts in the configuration), we verify at the end of the transaction execution if the new configuration has this form. If not, we conclude that a deadlock occurred during the execution, and we simply abort the whole transaction. We maintain snapshots of the configuration after every transaction execution, so we simply revert to the previous valid blockchain state. It is the user's responsibility to issue a new transaction that does not deadlock. In the future, we also plan to employ deadlock prevention techniques [Balzer et al. 2019] to statically rule out deadlocks.

10 OTHER RELATED WORK

We classify the related work into 3 categories - i) new programming languages for smart contracts, ii) static analysis techniques for existing languages and bytecode, and iii) session-typed and type-based resource analysis systems technically related to Nomos.

Smart Contract Languages. Existing smart contracts on Ethereum are predominantly implemented in Solidity [Auc 2016], a statically typed object-oriented language influenced by Python and Javascript. Languages like Vyper [Vyp 2018] address resource usage by disallowing recursion and infinite-length loops, thus making estimation of gas usage decidable. However, both languages still suffer from re-entrancy vulnerabilities. Bamboo [Bam 2018], on the other hand, makes state transitions explicit and avoids re-entrance by design. In contrast to our work, none of these languages use linear type systems to track assets stored in a contract.

Domain specific languages have also been designed for other blockchains apart from Ethereum. Typecoin [Crary and Sullivan 2015] uses affine logic to solve the peer-to-peer affine commitment problem using a generalization of Bitcoin where transactions deal in types rather than numbers. Although Typecoin does not provide a mechanism for expressing protocols, it also uses a linear type system to prevent resources from being discarded or duplicated.

Rholang [Rho 2018] is formally modeled by the ρ-calculus, a reflective higher-order extension of the π-calculus. Michelson [Mic 2018] is a purely functional stack-based language that has no side effects. However, none of these languages describe and enforce communication protocols statically. Scilla [Sergey et al. 2019] is an intermediate-level language where contracts are structured as communicating automata providing a continuation-passing style computational model to the language semantics. Scilla does not use session types or linearity but features static gas bounds. A difference is that Nomos' bounds are not asymptotic and are proved sound with respect to a cost semantics. The Move programming language from Facebook [Blackshear et al. 2019] is a flexible language based on Rust [Klabnik and Nichols 2018] to implement contracts on the Libra blockchain. Similar to Nomos, it provides the ability to define custom linear types to represent assets. However, it does not provide support to express contract protocols or gas usage.

Static Analysis. Analysis of smart contracts has received substantial attention [Grishchenko et al. 2018; Tikhomirov et al. 2018] recently due to their security vulnerabilities [Atzei et al. 2017; Tsankov et al. 2018]. KEVM [Hildenbrandt et al. 2018] creates a program verifier based on reachability logic that given an EVM program and specification, tries to automatically prove the corresponding reachability theorems. However, the verifier requires significant manual intervention, both in specification and proof construction. Oyente [Luu et al. 2016] is a symbolic execution tool that checks for 4 kinds of security bugs in smart contracts: transaction-order dependence, timestamp dependence, mishandled exceptions and re-entrancy vulnerabilities. MadMax [Grech et al. 2018] automatically detects gas-focused vulnerabilities with high confidence. The analysis is based on a decompiler that extracts control and data flow information from EVM bytecode, and a logic-based analysis specification that produces a high-level program model. Bhargavan et al. [2016] translate Ethereum contracts to F* to prove runtime safety and functional correctness, although they do not support all syntactic features. VeriSol [Lahiri et al. 2018] is a highly-automated formal verifier for Solidity that can produce proofs as well as counterexamples and proves semantic conformance of smart contracts against a state machine model with access-control policy. However, in contrast to Nomos, where guarantees are proved by a soundness proof of the type system, static analysis techniques often do not explore all program paths, can report false positives that need to be manually filtered, and miss bugs due to timeouts and other sources of incompleteness.

Session types and Resource analysis. Session types were introduced by Honda [Honda 1993] as a typed formalism for inter-process dyadic interaction. They have been integrated into a functional language in prior work [Toninho et al. 2013]. However, this integration does not account for resource usage or sharing. Sharing in session types has also been explored in prior work [Balzer and Pfenning 2017], but with the strong restriction that shared processes cannot rely on linear resources that we lift in Nomos. Shared session types were also never integrated with a functional layer or tracked for resource usage. While we consider binary session types that express local interactions, global protocols can be expressed using multi-party session types [Honda et al. 2008; Scalas and Yoshida 2019]. Automatic amortized resource analysis (AARA) has been introduced as a type system to derive linear [Hofmann and Jost 2003] and polynomial bounds [Hoffmann et al. 2017] for functional programming languages. Resource usage has also previously been explored separately for the purely linear process layer [Das et al. 2018a,b], but was never combined with shared session types or integrated with the functional layer.

11 CONCLUSION

We have described the programming language Nomos, its type-theoretic foundation, a prototype implementation and evaluated its feasibility on several real world smart contract applications. Nomos builds on linear logic, shared session types, and automatic amortized resource analysis to address the challenges that programmers are faced with when implementing digital contracts. Our main contributions are the design and implementation of Nomos' multi-layered resource-aware type system and its type soundness proof.

In future work, we plan to explore refinement session types [Das and Pfenning 2020] for expressing and verifying functional correctness of contracts against their specifications. We also plan to target open questions regarding a blockchain integration. These include the exact cost model, fluctuation of gas prices, and potential compilation to a lower-level language. Since Nomos has a concurrent semantics, we also plan to support parallel execution of transactions using speculation techniques [Saraph and Herlihy 2019] and evaluate the corresponding speed-up.

REFERENCES

2016. Solidity by Example. https://solidity.readthedocs.io/en/v0.3.2/solidity-by-example.html. Accessed: 2018-11-04.

2017. Ether.Camp's HKG Token Has A Bug And Needs To Be Reissued. https://www.ethnews.com/ethercamps-hkg-token-has-a-bug-and-needs-to-be-reissued. Accessed: 2019-02-25.

2018. Bamboo. https://github.com/cornellblockchain/bamboo. Accessed: 2018-11-04.

2018. ERC20 Token Standard. https://theethereum.wiki/w/index.php/ERC20_Token_Standard. Accessed: 2018-02-027.

2018. The Michelson Language. https://www.michelson-lang.com/. Accessed: 2018-11-04.

2018. Rholang. https://github.com/rchain/Rholang. Accessed: 2018-11-04.

2018. Vyper. https://vyper.readthedocs.io/en/latest/index.html. Accessed: 2018-11-04.

2018. Welcome to Liquidity's documentation! http://www.liquidity-lang.org/doc/index.html. Accessed: 2018-11-04.

2019. Nomos Implementation. link to repository removed for double blind review. Accessed: 2019-11-11.

Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. 2017. A Survey of Attacks on Ethereum Smart Contracts (SoK). In *Principles of Security and Trust - 6th International Conference, POST 2017*. 164–186. https://doi.org/10.1007/978-3-662-54455-6_8

Martin Avanzini, Ugo Dal Lago, and Georg Moser. 2015. Analysing the Complexity of Functional Programs: Higher-order Meets First-order. In *Proceedings of the 20th ACM SIGPLAN International Conference on Functional Programming (ICFP 2015)*. ACM, New York, NY, USA, 152–164. https://doi.org/10.1145/2784731.2784753

Stephanie Balzer and Frank Pfenning. 2017. Manifest Sharing with Session Types. *Proceedings of the ACM on Programming Languages (PACMPL)* 1, ICFP (2017), 37:1–37:29.

Stephanie Balzer, Bernardo Toninho, and Frank Pfenning. 2019. Manifest Deadlock-Freedom for Shared Session Types. (2019). 28th European Symposium on Programming (to appear).

P. N. Benton. 1994. A Mixed Linear and Non-Linear Logic: Proofs, Terms and Models. In *8th International Workshop on Computer Science Logic (CSL) (Lecture Notes in Computer Science)*, Vol. 933. Springer, 121–135. An extended version appeared as Technical Report UCAM-CL-TR-352, University of Cambridge.

Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Anitha Gollamudi, Georges Gonthier, Nadim Kobeissi, Natalia Kulatova, Aseem Rastogi, Thomas Sibut-Pinote, Nikhil Swamy, and Santiago Zanella-Béguelin. 2016. Formal Verification of Smart Contracts: Short Paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security (PLAS '16)*. ACM, New York, NY, USA, 91–96. https://doi.org/10.1145/2993600.2993611

Sam Blackshear, Evan Cheng, David L Dill, Victor Gao, Ben Maurer, Todd Nowacki, Alistair Pott, Shaz Qadeer, Dario Russi Rain, Stephane Sezer, et al. 2019. Move: A language with programmable resources.

Christian Cachin. 2016. Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, Vol. 310.

Luís Caires and Frank Pfenning. 2010. Session Types as Intuitionistic Linear Propositions. In *21st International Conference on Concurrency Theory (CONCUR)*. Springer, 222–236.

Quentin Carbonneaux, Jan Hoffmann, Thomas Reps, and Zhong Shao. 2017. Automated Resource Analysis with Coq Proof Objects. In *29th International Conference on Computer-Aided Verification (CAV'17)*.

Iliano Cervesato and Andre Scedrov. 2009. Relating state-based and process-based concurrency through linear logic (full-version). *Information and Computation* 207, 10 (2009), 1044 – 1077. https://doi.org/10.1016/j.ic.2008.11.006 Special issue: 13th Workshop on Logic, Language, Information and Computation (WoLLIC 2006).

Ezgi Cicek, Gilles Barthe, Marco Gaboardi, Deepak Garg, and Jan Hoffmann. 2017. Relational Cost Analysis. In *44th Symposium on Principles of Programming Languages (POPL'17)*.

Karl Crary and Michael J. Sullivan. 2015. Peer-to-peer Affine Commitment Using Bitcoin. In *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '15)*. ACM, New York, NY, USA, 479–488. https://doi.org/10.1145/2737924.2737997

Norman Danner, Daniel R. Licata, and Ramyaa Ramyaa. 2015. Denotational Cost Semantics for Functional Languages with Inductive Types. In *Proceedings of the 20th ACM SIGPLAN International Conference on Functional Programming (ICFP 2015)*. ACM, New York, NY, USA, 140–151. https://doi.org/10.1145/2784731.2784749

Ankush Das, Jan Hoffmann, and Frank Pfenning. 2018a. Parallel Complexity Analysis with Temporal Session Types. In *23rd International Conference on Functional Programming (ICFP'18)*.

Ankush Das, Jan Hoffmann, and Frank Pfenning. 2018b. Work Analysis with Resource-Aware Session Types. In *33rd ACM/IEEE Symposium on Logic in Computer Science (LICS'18)*.

Ankush Das and Frank Pfenning. 2020. Session Types with Arithmetic Refinements and Their Application to Work Analysis. arXiv:cs.PL/2001.04439

Simon Gay and Malcolm Hole. 2005. Subtyping for session types in the pi calculus. *Acta Informatica* 42, 2 (01 Nov 2005), 191–225. https://doi.org/10.1007/s00236-005-0177-z

Jean-Yves Girard. 1987. Linear Logic. *Theoretical Computer Science* 50 (1987), 1–102.

L.M Goodman. 2014. Tezos — a self-amending crypto-ledger. https://tezos.com/static/papers/white_paper.pdf.

Neville Grech, Michael Kong, Anton Jurisevic, Lexi Brent, Bernhard Scholz, and Yannis Smaragdakis. 2018. MadMax: Surviving Out-of-gas Conditions in Ethereum Smart Contracts. *Proc. ACM Program. Lang.* 2, OOPSLA, Article 116 (Oct. 2018), 27 pages. https://doi.org/10.1145/3276486

Ilya Grishchenko, Matteo Maffei, and Clara Schneidewind. 2018. Foundations and Tools for the Static Analysis of Ethereum Smart Contracts. In *Computer Aided Verification*, Hana Chockler and Georg Weissenbacher (Eds.). Springer International Publishing, Cham, 51–78.

Everett Hildenbrandt, Manasvi Saxena, Xiaoran Zhu, Nishant Rodrigues, Philip Daian, Dwight Guth, Brandon Moore, Yi Zhang, Daejun Park, Andrei Stefănescu, and Grigore Rosu. 2018. KEVM: A Complete Semantics of the Ethereum Virtual Machine. In *2018 IEEE 31st Computer Security Foundations Symposium*. IEEE, 204–217.

Jan Hoffmann, Klaus Aehlig, and Martin Hofmann. 2011. Multivariate Amortized Resource Analysis. In *38th Symposium on Principles of Programming Languages (POPL'11)*.

Jan Hoffmann, Ankush Das, and Shu-Chun Weng. 2017. Towards Automatic Resource Bound Analysis for OCaml. In *44th Symposium on Principles of Programming Languages (POPL'17)*.

Martin Hofmann and Steffen Jost. 2003. Static Prediction of Heap Space Usage for First-Order Functional Programs. In *30th ACM Symp. on Principles of Prog. Langs. (POPL'03)*.

Kohei Honda. 1993. Types for Dyadic Interaction. In *4th International Conference on Concurrency Theory (CONCUR)*. Springer, 509–523.

Kohei Honda, Vasco T. Vasconcelos, and Makoto Kubo. 1998. Language Primitives and Type Discipline for Structured Communication-Based Programming. In *7th European Symposium on Programming (ESOP)*. Springer, 122–138.

Kohei Honda, Nobuko Yoshida, and Marco Carbone. 2008. Multiparty Asynchronous Session Types. In *35th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*. ACM, 273–284.

Blockchain Insurance Industry Initiative. 2008. B3i. (2008).

Steffen Jost, Kevin Hammond, Hans-Wolfgang Loidl, and Martin Hofmann. 2010. Static Determination of Quantitative Resource Usage for Higher-Order Programs. In *37th ACM Symp. on Principles of Prog. Langs. (POPL'10)*.

Steve Klabnik and Carol Nichols. 2018. *The Rust Programming Language*. No Starch Press, USA.

Ugo Dal Lago and Marco Gaboardi. 2011. Linear Dependent Types and Relative Completeness. In *26th IEEE Symp. on Logic in Computer Science (LICS'11)*.

Shuvendu K. Lahiri, Shuo Chen, Yuepeng Wang, and Isil Dillig. 2018. Formal Specification and Verification of Smart Contracts for Azure Blockchain. *CoRR* abs/1812.08829 (2018). arXiv:1812.08829 http://arxiv.org/abs/1812.08829

Angwei Law. 2017. *Smart contracts and their application in supply chain management*. Ph.D. Dissertation. Massachusetts Institute of Technology.

Christopher Lidbury and Alastair F. Donaldson. 2019. Sparse Record and Replay with Controlled Scheduling. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2019)*. ACM, New York, NY, USA, 576–593. https://doi.org/10.1145/3314221.3314635

Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. 2016. Making Smart Contracts Smarter. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. ACM, New York, NY, USA, 254–269. https://doi.org/10.1145/2976749.2978309

Vincenzo Morabito. 2017. Smart contracts and licensing. In *Business Innovation Through Blockchain*. Springer, 101–124.

Satoshi Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. http://bitcoin.org/bitcoin.pdf.

Frank Pfenning and Dennis Griffith. 2015. Polarized Substructural Session Types. In *18th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*. Springer, 3–22.

Benjamin C. Pierce and David N. Turner. 2000. Local Type Inference. *ACM Trans. Program. Lang. Syst.* 22, 1 (Jan. 2000), 1–44. https://doi.org/10.1145/345099.345100

Francois Pottier and Yann Régis-Gianas. 2019. *Menhir Reference Manual*.

Klaas Pruiksma, William Chargin, Frank Pfenning, and Jason Reed. 2018. *Adjoint Logic*. Technical Report. Carnegie Mellon University.

Ivan Radiček, Gilles Barthe, Marco Gaboardi, Deepak Garg, and Florian Zuleger. 2017. Monadic Refinements for Relational Cost Analysis. *Proc. ACM Program. Lang.* 2, POPL (2017).

Jason Reed. 2009. A Judgmental Deconstruction of Modal Logic. (January 2009). http://www.cs.cmu.edu/~jcreed/papers/jdml.pdf Unpublished manuscript.

Michiel Ronsse and Koen De Bosschere. 1999. RecPlay: A Fully Integrated Practical Record/Replay System. *ACM Trans. Comput. Syst.* 17, 2 (May 1999), 133–152. https://doi.org/10.1145/312203.312214

Vikram Saraph and Maurice Herlihy. 2019. An Empirical Study of Speculative Concurrency in Ethereum Smart Contracts. *CoRR* abs/1901.01376 (2019). arXiv:1901.01376 http://arxiv.org/abs/1901.01376

Alceste Scalas and Nobuko Yoshida. 2019. Less is More: Multiparty Session Types Revisited. *Proc. ACM Program. Lang.* 3, POPL, Article 30 (Jan. 2019), 29 pages. https://doi.org/10.1145/3290343

Ilya Sergey, Vaivaswatha Nagaraj, Jacob Johannsen, Amrit Kumar, Anton Trunov, and Ken Chan Guan Hao. 2019. Safer Smart Contract Programming with Scilla. *Proc. ACM Program. Lang.* 3, OOPSLA, Article 185 (Oct. 2019), 30 pages. https://doi.org/10.1145/3360611

David Siegel. 2016. Understanding The DAO Hack for Journalists. https://medium.com/@pullnews/understanding-the-dao-hack-for-journalists-2312dd43e993.

S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov. 2018. SmartCheck: Static Analysis of Ethereum Smart Contracts. In *2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*. 9–16.

Bernardo Toninho, Luís Caires, and Frank Pfenning. 2013. Higher-Order Processes, Functions, and Sessions: a Monadic Integration. In *22nd European Symposium on Programming (ESOP)*. Springer, 350–369.

Petar Tsankov, Andrei Dan, Dana Drachsler-Cohen, Arthur Gervais, Florian Bünzli, and Martin Vechev. 2018. Securify: Practical Security Analysis of Smart Contracts. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*. Association for Computing Machinery, New York, NY, USA, 67–82. https://doi.org/10.1145/3243734.3243780

Philip Wadler. 2012. Propositions as Sessions. In *17th ACM SIGPLAN International Conference on Functional Programming (ICFP)*. ACM, 273–286.

Gavin Wood. 2014. Ethereum: A secure decentralized transaction ledger. http://gavwood.com/paper.pdf.

Discussion Paper #3

Legal Regulation of Virtual Currencies: Illicit Activities and Current Developments in the realm of Payment Systems

Ilias Ioannou

Centre of Commercial Law Studies, Queen Mary University of London

Abstract

Since Bitcoin was invented a decade ago, the phenomenon of Virtual Currencies has been hailed as an ingenious innovation and decried as the preferred transaction vehicle for illicit actors. Despite the numerous headlines discussing the virtues and vices of virtual currencies, heretofore there has been no comprehensive legal response. Regulators seem to contemplate the wrong question; they wonder whether they should regulate virtual currencies instead of how to regulate them.

The present contribution elaborates on the second question. Starting with a conceptual analysis of virtual currencies and their promising potential, it identifies the financial crime risks posed by the intersection between legitimate and illegitimate users. The research shows that a fragmentary regulation would be ineffective; this promising technology will either be integrated into the lawful economy or it will be exploited by criminals. The paper attempts to fill the regulatory gap by providing a recommendation for the embeddedness of Virtual Currencies into the EU financial system. It achieves that by redirecting regulation towards the uniqueness of their underlying technology.

Keywords: Blockchain, Cryptocurrencies, Directive (EU) 2018/843, E-Money, Financial Crime, Money Laundering, Payment Systems, Virtual currencies

‘Legal Regulation of Virtual Currencies: Illicit Activities and Current Developments in the realm of Payment Systems’

Abstract

Since Bitcoin was invented a decade ago, the phenomenon of Virtual Currencies has been hailed as an ingenious innovation and decried as the preferred transaction vehicle for illicit actors. Despite the numerous headlines discussing the virtues and vices of virtual currencies, heretofore there has been no comprehensive legal response. Regulators seem to contemplate the wrong question; they wonder whether they should regulate virtual currencies instead of how to regulate them.

The present contribution elaborates on the second question. Starting with a conceptual analysis of virtual currencies and their promising potential, it identifies the financial crime risks posed by the intersection between legitimate and illegitimate users. The research shows that a fragmentary regulation would be ineffective; this promising technology will either be integrated into the lawful economy or it will be exploited by criminals. The paper attempts to fill the regulatory gap by providing a recommendation for the embeddedness of Virtual Currencies into the EU financial system. It achieves that by redirecting regulation towards the uniqueness of their underlying technology.

Keywords: Blockchain, Cryptocurrencies, Directive (EU) 2018/843, E-Money, Financial Crime, Money Laundering, Payment Systems, Virtual currencies

Table of Contents

A. Introduction

The era of virtual currencies (VCs) has begun. The shift from centralised to decentralised blockchain-based creations for the transfer of value is ongoing. VCs which operate in so called 'permission-less' blockchains, such as Bitcoin, emerged as viable competitors to central bank's fiat currency, aiming to provide 'a version of electronic cash that would allow online payments to be sent directly from one party to another without going through a financial institution'.[1] Banking institutions, such as J.P. Morgan Chase, are trying to stay in the game by leveraging VCs to provide digitised cross-border payments.[2] Newer initiatives, such as Facebook's Libra, aspire to bring cryptocurrency to the masses by providing 'a reliable digital currency and infrastructure that together can deliver on the promise of the internet of money'.[3]

VCs offer exceptional opportunities for innovation and development in international payment and remittance systems. However, they are also uniquely tailored to facilitate illegitimate activities. Starting with an analysis of the applicability and suitability of the current EU legal concepts, this inquiry will re-examine the statutory framework for the regulation of companies engaging in virtual currency business activity. In the current state of affairs, where the fifth anti-money laundering directive[4] has to be implemented by member states into national law by January 2020, this research could function as an inquiry on its implementation. However, VCs are by nature global coins, hence, a comprehensive solution can only be elaborated at a supranational level, rather than solely through a particular country's regulatory framework.

Following an introductory chapter on the basic notions related to VCs, Part II analyses their promising role in the realm of international payments. Subsequently, the

[1] Satoshi Nakamoto, 'Bitcoin: A Peer to Peer Electronic Cash System' (Bitcoin, 2008) <https://bitcoin.org/bitcoin.pdf> accessed 16 December 2019.
[2] Press Release, 'J.P. Morgan Creates Digital Coin for Payments' (February 2019) <https://www.jpmorgan.com/global/news/digital-coin-payments> accessed 16 December 2019.
[3] Libra Association, 'An introduction to Libra', (Libra, 2019), 4 <https://libra.org/en-US/white-paper/?noredirect=en-US#introduction> accessed 16 December 2019.
[4] Directive (EU) 2018/843 of the European Parliament and of the Council of 30 May 2018 amending Directive (EU) 2015/849 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing, and amending Directives 2009/138/EC and 2013/36/EU [2018] OJ L 156/43 (hereinafter 5AMLD)

illicit activities are identified through a risk analysis method in the context of money laundering, terrorist financing and illicit flow of money. It is argued that the regulation needs to be balanced between the attractiveness of VCs and its utility for the identified financial crime risks. To that effect, Part IV analyses the existing financial crime related regulation and policymaking in the EU. In Part V the shortcomings of the EU anti-money laundering regime are highlighted and concrete recommendations for its improvement are laid out. The article concludes with a comprehensive proposition for the legal regulation of VCs in the EU.

B. Analysis

I. Typology of Virtual Currencies

1.1.Key Definitions

As regulators and legislative bodies around the globe begin to deal with the legal challenges that virtual currencies (VCs) pose to the payment systems, it becomes clear that they lack a common language that accurately describes the different types of VCs.[5] In the EU, the fifth anti-money laundering directive (5AMLD) legally defines VCs as follows:

> ‘A digital representation of value that is not issued or guaranteed by a central bank or a public authority, is not necessarily attached to a legally established currency and does not possess a legal status of currency or money, but is accepted by natural or legal persons as a means of exchange and which can be transferred, stored and traded electronically.’[6]

According to the Financial Action Task Force (FATF) virtual currency is ‘a digital representation of value that can be digitally traded and functions as (1) a medium of exchange; and/or (2) a unit of account; and/or (3) a store of value, but does not have

[5] Robby Houben and Alexander Snyers, ‘Cryptocurrencies and blockchain: Legal context and implications for financial crime, money laundering and tax evasion’ (July 2018) Department for Economic, Scientific and Quality of Life Policies, 20 <https://goo.gl/PSxbzx> accessed 5 April 2020.

[6] See art.1(2)(d) of the 5AMLD, which inserts this definition in the art.3(18) of the Directive (EU) 2015/849 of the European Parliament and of the Council of 20 May 2015 on the prevention of the use of the financial system for the purposes of money laundering or terrorist financing, amending Regulation (EU) No 648/2012 of the European Parliament and of the Council, and repealing Directive 2005/60/EC of the European Parliament and of the Council and Commission Directive 2006/70/EC [2015] OJ L 141/73 (hereinafter 4AMLD).

legal tender status in any jurisdiction'.[7] The legal tender status refers to monetary instruments that, by law, can be used by a debtor of a monetary debt to discharge the debt, without necessarily requiring the creditor to accept the payment.[8]

The term is used in this article broadly to refer to Bitcoin, Ether, Libra and other stablecoins, altcoins,[9] cryptocurrencies, and other virtual tokens that can be used as a means of payment without having a legal tender status. It is ostensibly distinct from 'e-money', which is usually understood as a digital transfer mechanism for fiat currency,[10] and from 'digital currency' which is an overarching term for digital representation of either virtual currency or fiat money.

1.2.Taxonomy

VCs can be classified based on their functionality as tokens, their convertibility, their issuance and administration structure. Regarding their functionality as tokens, virtual[11] or crypto[12] assets have been distinguished by legislative authorities and commentators into three classes:[13] (i) 'exchange' or 'currency' tokens, which are intended and designed to be used as a means of exchange, (ii) 'utility tokens', which, akin to pre-payment vouchers, embody a relationship between the token issuer and the token holder, and (iii) 'security' or 'investment' tokens, which are comparable to

[7] FATF, 'Virtual Currencies: Key definitions and Potential AML/CFT Risks' (June 2014), 4 <https://www.fatf-gafi.org/media/fatf/documents/reports/Virtual-currency-key-definitions-and-potential-aml-cft-risks.pdf> accessed 16 December 2019.

[8] See Commission Recommendation 2010/191/EU on the scope and effects of legal tender of euro banknotes and coins [2010] OJ L 83/70, which refers to three characteristics of legal tender: (1) it is mandatory acceptance; (2) its acceptance at full face value; and (3) its power to discharge payment obligations. Similarly, in English law, the concept of legal tender developed from the law on the performance of debts; See Moss v Hancock (1899) 2 Q.B. 116; See also Thomas H Greco, 'Money: Understanding and creating alternatives to legal tender', (Chelsea Green Publishing 2001) 26.

[9] 'Altcoin' is a combination of the word 'alt' signifying 'alternative' and 'coin' signifying 'cryptocurrency'. Thus, it refers to all cryptocurrencies which are alternatives to the first cryptocurrency, namely all cryptocurrencies which are not Bitcoin.

[10] The so-called 'fiat currency' is any currency that exists and has a nominal value determined by the law establishing the currency. According to FATF, 'Virtual Currencies: Guidance for a risk-based approach' (June 2015) 26, 'fiat currency' is 'the coin and paper money of a country that is designated as legal tender; circulates and is customarily used and accepted as a medium of exchange'. The 5AMLD defines fiat currency more broadly as 'coins and banknotes that are designated as legal tender and electronic money of a country.'

[11] In FATF terminology.

[12] In European Banking Authority (EBA) terminology.

[13] FCA, 'Guidance on Cryptoassets' (July 2019) <https://www.fca.org.uk/publication/policy/ps19-22.pdf> accessed 16 December 2019; Philipp Maume and Mathias Fromberger, 'Regulation of Initial Coin Offerings: Reconciling US and EU Securities Laws' (2019) 19 (2) Chicago Journal of International Law 548, 558; Luka Muller et al, 'Conceptual Framework for Legal and Risk Assessment of Crypto Tokens' (Zurich 2019), 10 <https://www.mme.ch/fileadmin/files/documents/180501_BCP_Framework_for_Assessment_of_Crypto_Tokens_-_Block_2.pdf> accessed 16 December 2019.

traditional securities. The research hereinafter will focus primarily on the first kind of tokens, which are the most relevant to the payment industry.

Based on its convertibility into fiat currency, each VC can be allocated to one of three types:[14] (i) VCs that cannot be purchased or exchanged for legal tender,[15] (ii) VCs that can be obtained with legal tender, but are not convertible back into legal tender[16] and (iii) open or fully convertible VCs that are bidirectional.[17] This categorisation is helpful because it indicates which risks are relevant to each type of token. However, it refers more to the *de facto* convertibility of VCs rather than to a legal assessment of them.[18]

Depending on their issuance and administration pattern, VCs can be either centralised or decentralised.[19] All closed and unidirectional VCs are centralised because they are issued by a central entity.[20] Decentralised VCs are distributed open-source, peer-to-peer and have no central administering authority.[21] This means that the entire system is made up 'of versions of the software that end-users download and run on their personal computers.'[22]

Exchange tokens can be analysed further into cryptocurrencies, stablecoins and e-money tokens. The term cryptocurrency is commonly understood as a specific subcategory of VCs, namely 'decentralised convertible VC that is protected by cryptography'.[23] A stablecoin is a token designed to avoid the volatility inherent in

[14] ECB, 'Virtual Currency Schemes' (2012), 6 <https://www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemes201210en.pdf> accessed 16 December 2019.
[15] For example, coins that are obtained as a reward in a computer game and can only be spent on virtual items therein.
[16] For example, PlayStation Store credit.
[17] For example, Bitcoin.
[18] Anton Didenko and Ross Buckley, 'The evolution of currency: From Cash to Cryptos to Sovereign Digital Currencies', 42 (4) Fordham International Law Journal (2019) 1076; For example, the address 'https://www.g2g.com/' allows online conversion of World of Warcraft in-game gold.
[19] ECB, 'Virtual Currency Schemes – A further analysis' (2015), 6 <https://www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemesen.pdf> accessed 16 December 2019.
[20] FATF, Virtual Currencies 2014 (n 7), 5.
[21] Angela Walch 'The Bitcoin Blockchain as Financial Market Infrastructure: A consideration of Operational Risk' (2015) 18(4) NYU Journal of Legislation & Public Policy, 33.
[22] Shawn Bayern, 'Of Bitcoins, Independently Wealthy Software, and the Zero-Member LLC' (2014) 108 Northwestern University Law Review 1485, 1488 <https://scholarlycommons.law.northwestern.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1009&context=nulr> accessed 16 December 2019.
[23] FATF, Virtual Currencies 2014 (n 7) 5.

other VCs by using a variety of mechanisms,[24] such as fiat/asset-collateralised, crypto-collateralised and non-collateralised stabilisation models.[25] Lastly, e-money tokens are any tokens that meet the definition of electronic money in a given jurisdiction.

1.3.Blockchain System Participants

Various actors are involved in the VC ecosystem. A first player is the issuer of a VC, which can be either the 'coin inventor' or 'coin offeror'.[26] A coin inventor is an individual or an organisation that sets up the technical foundations of a VC and gratuitously distribute it. A coin offeror offers the token at an initial coin offering (ICO),[27] usually against payment to fund the coin's development.

VCs can also be purchased on secondary markets such as VC exchanges. An exchanger is a business engaged in the exchange of VCs for fiat currency, funds, precious metals or other forms of VC and vice versa for a commission.[28] VC exchangers are distinguished between providers who facilitate the exchange between fiat currencies and VCs[29] and pure cryptocurrency exchanges, which accept payments solely in other VCs.[30]

Furthermore, so-called 'trading platforms' are also a possible entry-point to the VC market. Trading platforms provide cryptocurrency users with a platform on which they can directly trade with each other without buying or selling VCs themselves.[31] Some of them are decentralised or peer-to-peer platforms as they are not run by either an entity or a company.[32] Instead, they are operated by software allowing purchasers and vendors to conduct deals online or even locally.[33]

[24] Makiko Mita, Kensuke Ito, Shohei Ohsawa and Hideyuki Tanaka, 'What is Stablecoin?: A Survey on Price Stabilisation Mechanisms for Decentralised Payment Systems' (2019), 2 <arXiv:1906.06037v1> accessed 16 December 2019.

[25] Ben Regnard-Weinrabe, 'Stablecoins' (Harvard Law School Forum on Corporate Governance and Financial Regulation, 10 February 2019) <https://corpgov.law.harvard.edu/2019/02/10/stablecoins/> accessed 16 December 2019.

[26] Houben and Snyers (n 5) 28.

[27] For an overview of the ICO market, see Dirk Zetzsche et al., 'The ICO Gold Rush: It's a Scam, It's a Bubble, It's a Super Challenge for Regulators', (2019) 63 Harvard International Law Journal, 2 <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3072298> accessed 16 December 2019.

[28] FATF, Virtual Currencies 2014 (n 7) 7.

[29] For example, websites (www.bitfinex.com, www.coinbase.com) or kiosks (bitcoin ATMs).

[30] For example, www.binance.com, https://international.bittrex.com/, and https://changelly.com/.

[31] Houben and Snyers (n 5) 27.

[32] Andrew Marshall, 'P2P Cryptocurrency exchanges' (CoinTelegraph, 7 April 2017) <https://cointelegraph.com/explained/p2p-cryptocurrency-exchanges-explained> accessed 16 December 2019.

[33] For example, www.localbitcoins.com.

To purchase, hold and transfer VCs, end-users need a pair of cryptographic keys. The public key is an alphanumeric code which functions as an address, similar to a bank account number. The private key is a similar code, except it is secret since it functions as the owner's 'signature', thereby proving the identity of address owner. The pair of private and public keys allows the user to transfer VCs in the form of transaction outputs/inputs, which are stored in a 'wallet' that is the primary user's interface.[34]

A wallet provider translates an address transaction history into a customer friendly format, enabling end-users to conduct VC transactions.[35] Wallet providers can be distinguished between custodian and non-custodian. Custodian wallet providers run platforms which allow users to create accounts and automatically store their private keys. To perform a transaction, the sender only needs to access credentials similar to an e-banking platform and is not required to remember their private key. In contrast, non-custodian wallet providers facilitate end-users to store their private keys themselves by providing them with a suitable wallet.[36]

Finally, the so-called 'tumbler services',[37] a different type of service provider, amalgamates a VC transaction with others to obscure the nexus between a sender and a recipient.[38] In this way, it becomes unclear whom the user intended the VCs to be directed to. Skillful users may use other tools designed to further enhance online anonymity, such as *Tor*[39] and *Darkwallet*.[40] The risks from these platforms are out of the scope of this research because they are not uniquely tailored to VCs.[41]

[34] Antonopoulos, *Mastering Bitcoin* (2nd Edition O'Reilly 2018) Ch. 5.

[35] Houben and Snyers (n 5) 27.

[36] For example, a hardware wallet (USB device) or a paper wallet (a piece of paper with two QR codes on it).

[37] For example, https://bitblender.io/ or https://bitcoin-laundry.com/

[38] Lars Haffke, Mathias Fromberger and Patrick Zimmermann, 'Virtual Currencies and AML – The Shortcomings of the 5th AML Directive (EU) and How to Address Them' (2019), 7 <papers.ssrn.com/sol3/papers.cfm?abstract_id=3328064;> accessed 16 December 2019.

[39] Tor (www.torproject.org/download/) is a network that permits users to browse the web anonymously. See Husam Al Jwaheri et al., 'Deanonymising Tor hidden service users through Bitcoin Transactions analysis' (2020) 89 Computers & Security <https://www.sciencedirect.com/science/article/pii/S0167404818309908?A> accessed 5 April 2020.

[40] Dark wallet is a digital wallet which promises total anonymity to its user base. See Aaron Van Wirdum, 'CoinJoin's First Steps: How Dark Wallet Paved the Way for A More Private Bitcoin' (Bitcoin Magazine, February 2020) < https://bitcoinmagazine.com/articles/coinjoins-first-steps-how-dark-wallet-paved-the-way-for-a-more-private-bitcoin> accessed 5 April 2020.

[41] Cath Senker, *Cybercrime and the Darknet: Revealing the hidden underworld of the Internet* (Arcturus Publishing 2016); Sesha Kethineni, Ying Han Cao and Casssandra Dodge 'Use of Bitcoin in Darknet Markets: Examining Facilitative Factors on Bitcoin-Related Crimes' (2018) 43(2) American Journal of Criminal Justice, 141.

The conceptual tools elaborated in this chapter will serve as a benchmark throughout this paper and will facilitate the assessment of the adequacy of the existing and upcoming legal framework. Before touching upon the details of how regulation needs to be reoriented, the study will examine the revolutionising potential of VCs in the context of payment systems.

II. The Promising Role of Virtual Currencies in International Payment Systems.

2.1.Introduction

VCs have the potential to improve payment efficiency and reduce transaction costs for payments and international fund transfers. They can function as global currencies but avoid exchange fees and are processed with significantly lower transaction fees than traditional payments.

2.2.The Status Quo

A payment is considered to be 'a gift or a loan of money or any act offered and accepted in performance of a money obligation'.[42] In a simple domestic credit transfer scheme, the Debtor (D) sends a payment order to his Bank (DB) in electronic form, specifying the Beneficiary's account details and the amount to be sent. The DB checks the D's account balance, performs other know-your-customer (KYC) and anti-money laundering (AML) duties, debits the D's account, notifies the beneficiary's corresponding bank (CB) and credits its reserve account with a central bank. The central bank is the central clearing and settlement house. In other words, it exchanges the payment orders between the participating banks and settles the payment through entries in its ledger.

Absent a central settlement and clearing institution, an international funds transfer relies on an interbank correspondent banking system conducted through further intermediary banks. The payment order is processed through intermediary clearing agents before the transaction is processed and settled through nostro and vostro accounts and interbank correspondence between the central clearing houses of each country.[43]

[42] Ewan McKendrick, *Goode on Commercial Law* (5th ed. 2016), 488.

[43] N Asokan, Philippe Janson, Michael Steiner and Michael Waidner 'The state of the art in electronic payment systems' (1997) 30(9) IEEE Computer Society Press 28-29; BG Aksatha and SB Akash 'Nostro and Vostro accounts – effective tool for cross border settlements' (2014) 4(2) ZENITH International Journal of Multidisciplinary Research 37.

The validation of information at each intermediary is timely, costly and the associated transaction risk is concentrated in large clearing houses. According to the Federal Reserve (FED), the risks that may arise include credit or counterparty risk, liquidity risk, the operational risk and the legal risk.[44] The latter unfolds in costly disputes as to whether a payment was made timely[45] or whether a payment can be revoked. [46]

2.3. Fintech Solutions

Fintech, the combination of the words 'finance' and 'technology', is any financial technology infrastructure that offers financial services.[47] In terms of payment systems, there have been remarkable developments for the cross-border transfers of money. For example, *PayPal* is another intermediary between the debtor, creditor and banks which facilitates online payments through its customer-friendly interface. Similarly, *Revolut*, an entirely digital bank with minimal operational expenditure, can hold and exchange with the interbank exchange rate 29 currencies, including 5 VCs. Additionally, *TransferWise* performs international transfers through two local transfers linked together by software. For instance, a customer sends money in Euros to the French *TransferWise* and the equivalent in British pounds is sent from the UK *TransferWise* to the account recipient, without the funds crossing borders. Overall, the payment industry has attracted a spate of fintech start-up businesses, such as *Stripe*, *Moneygram* and *M-Pesa*, which facilitate transactions denominated, primarily, in fiat money. [48]

2.4.Blockchains and Virtual Currencies

The process of modernisation and digital transformation of current payment systems creates fertile ground for the adoption of new technologies, such as blockchains and more generally distributed ledger technologies.[49] A blockchain payment system may be either permission-less or permissioned.

[44] Federal Reserve, 'Policy on Payment System Risk', 2017, 4-5.

[45] *Rekstin v Severo* [1933] 1 KB 47 (CA), *Royal Products Ltd v Midland Bank Ltd* [1981] 2 Lloyd's Rep 194.

[46] *The Laconia* [1977] AC 850 (HL), *The Chikuma* [1981] 1 WLR 314 (HL).

[47] Umar Oseni, *Fintech in Islamic finance theory and practice*. (Routledge, 2019), Ch 11.

[48] Ignacio Mas and Dan Radcliffe, 'Mobile Payments Go Viral: M-PESA in Kenya' (2011) 32 Journal of Financial Transformation 169.

[49] Didenko and Buckley (n 18) 1070.

2.4.1. Permission-less Blockchain Payment Systems

The most famous permission-less blockchain-based payment system is Bitcoin: According to Antonopoulos, 'bitcoin is a collection of concepts and technologies that form the basis of a digital money ecosystem'.[50] A key characteristic of Bitcoin is that it is entirely virtual and that there are no physical or even digital coins per se. The users have their own keys and can participate in the network as full node validators of the transactions. The transactions are peer-to-peer and there is not a central clearing house or point of control. The transfer of bitcoin is represented by a chain of current and past owners and the procedure is the same irrespective of whether the transfer is domestic or international.

The transfer can be initiated by the debtor who creates the transaction, signs with his private key and the public key of the beneficiary. The transactions are broadcasted to all participating nodes that then verify the transaction message validity and forward it to connected nodes. Each node (called 'miner') bundles the transactions into a block and works on solving a difficult mathematical problem known as proof-of-work (PoW).[51] Once a miner solves the PoW, all participating nodes are informed. The nodes check for the validity and accept (or reject) the block. They express their acceptance by working on creating the next block using the encrypted stamp (hash) of the accepted block as the previous block hash. The block is uploaded, and the successful miner is awarded a new batch of bitcoin automatically generated by the software (coinbase transaction) and transaction fees.

As soon as the D hits the send button, the payment becomes irreversible. The transaction will appear in B's wallet as 'unconfirmed' within seconds and will be confirmed after an estimated ten minutes when it will then be added to the blockchain. This timeframe is much quicker than the speed at which traditional payments are processed, especially cross-border ones where several days may be required. The disintermediation of the payment system results in much lower transaction fees and, theoretically, there are no transaction or foreign exchange risks.

[50] Antonopoulos (n 34) 1.

[51] David Orrel and Roman Chlupaty, *The Evolution of Money* (Columbia UP 2016) 199.

On the other side of the spectrum, huge price fluctuations[52] and scalability issues have prevented permission-less VCs from being broadly accepted. As they transform to speculative assets, their utility as decentralised payment mechanisms become secondary. That is one of the reasons for the shift towards stablecoin proposals running in permissioned blockchains, such as Libra in the retail payments context and the JPM coin in financial markets.

2.4.2. Permissioned Blockchain Payment Systems

Permissioned blockchains maintain an access control layer to allow certain actions to be performed solely by authorised participants.[53] In terms of payment systems, a permissioned blockchain could be a peer-to-peer network of participating banks or private institutions. The latter will act as validators of the transactions and not as classical intermediaries by using the proof-of-stake,[54] proof-of-identity[55] or another consensus mechanism. Following network validation, native tokens, which could be stablecoins, are transferred between the counterparties. The difference from a permission-less blockchain is that the counterparties can either be the permissioned entities, in which case the blockchain is used as a settlement mechanism, or 'lightweight clients', whereby the transaction will take place directly between wallet holders and/or end-users but will be validated only by the participating banks or authorised entities that operate as full nodes.[56]

Blockchain-based VCs make possible to transfer value across the globe in a seamless and cost-effective way. The counterparty risk, the liquidity risk, the operational risk as well as the legal ambiguities associated with the current payment infrastructure are eliminated by the instantaneous transfer of ownership. Similarly, the high transaction costs and the geographical limitations of Fintech-based payment systems could be supplanted by fast, scalable and secure global VCs. The promising role of VCs has been acknowledged by major sovereignties and central banks, such as

[52] Gina Pieters and Sofia Vivanco, 'Financial regulations and price inconsistencies across Bitcoin markets' (2017) 39 Information Economics and Policy 3.
[53] Jake Frankenfield, 'Permissioned Blockchain' (Investopedia, August 2019) <https://www.investopedia.com/terms/p/permissioned-blockchains.asp> accessed 16 December 2019.
[54] In proof-of-stake VCs, the creator of the next block is chosen via various combinations of random selection and wealth or age (i.e. the stakes).
[55] Proof of identity/authority is a modified form of proof-of-stake where a validator's identity performs the role of stake.
[56] For a comparison of blockchain protocols see Taskinsoy, 'Facebook's Project Libra' (2019) 14 <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3423453> accessed 16 December 2019.

the Bank of England, the European Central Bank and the People's Bank of China, which are now contemplating central bank versions of virtual currencies initiatives running on permissioned blockchains.

III. Risks Associated with the Use of Virtual Currencies

3.1.Understanding Financial Crime Threats

VCs have the potential to offer benefits to businesses and consumers, but they also have features which make them attractive for abuse, namely, their almost anonymous nature, their global reach and the absence of a central intermediary.

3.1.1. High Level of Anonymity

The foundational cause that provokes illicit activities using VCs is the high level of anonymity associated with them. Anonymity can be construed both as privacy and hidden identity.[57] Anonymity as privacy refers to citizenry concerns regarding their personal data and their freedom to browse the web and make purchases without being tracked. As such, privacy is considered a desirable feature of VCs.[58] Contrarily, anonymity as hidden identity is affiliated with criminals who need it to avoid the authorities and hide their identity.[59] Given that anyone can create as many public addresses as they want without providing any identifying information, VCs enable completely anonymous transfers. [60]

It should be noted, however, that most VCs are not entirely anonymous. Instead, they are pseudonymous, and the owners are identified by their public cryptocurrency address. Consequently, given that the identity of some wallet owners is known, it is possible to use these known addresses in order to track the transactions.[61] By doing so, law enforcement agencies can deploy techniques to expose the identity of the owners of unknown wallets with whom the known wallets transacted. [62] This could create a cascade effect, whereby the increase of identified wallet holders would make it easier

[57] Victor Dostov and Pavel Shust, 'Cryptocurrencies: an unconventional challenge to the AML/CFT regulators?' (2014) 21(3) Journal of Financial Crime 249, 258.

[58] Ibid.

[59] See Gary Marx, 'Identity and Anonymity: Some Conceptual Distinctions and Issues for Research' in J Caplan and J Torpey (eds), *Documenting Individual Identity* (Princeton University Press, 2001) 311.

[60] David Carlisle, 'Virtual Currencies and Financial Crime', RUSI Occasional Paper, (March 2017) 10.

[61] For example, *Chainanalysis Inc*. has developed an adequate software to track Blockchain transactions.

[62] Steven Goldfeder et al., 'When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies' (CoRR 2017), 2-3 <https://arxiv.org/pdf/1708.04748.pdf> accessed 16 December 2019.

to identify the owners of unknown addresses.[63] For this reason, the approach of regulators and legislative authorities should be to gather as much information as possible for the enforcement authorities.

3.1.2. Absence of a Central Intermediary

A second reason that renders decentralised VCs suitable for illicit activities is that the existing financial regulation system relies on regulating intermediaries to control harmful behaviour. For instance, financial institutions are used as regulatory agents and are obliged to perform know-your-customer (KYC) and other customer due diligence (CDD) duties to prevent money laundering.[64] In the context of VCs, miners, exchanges, wallet providers and other system participants are indispensable in enabling blockchain payment networks to function. However, payment systems operating on permission-less blockchain are still decentralised and anyone can participate on a peer-to-peer basis. As a result, law enforcement authorities cannot target a central administrator or clearing house for investigative and law enforcement purposes. Therefore, it is crucial to identify which players in the virtual market should be regulated.

3.1.3. Global Reach

Lastly, a key innovation of VC systems is their worldwide accessibility through the Internet and their ability to transcend national borders instantaneously. However, the inherently cross-border and mobile nature of VCs likewise makes them well suited for transnational crimes. Particularly, some participants of a decentralised VC ecosystem may be located in jurisdictions that do not have effective regulations or adequate AML/CFT controls, thereby leveraging completely anonymous interaction with the system.[65] Similarly, VCs that are deployed in permissioned blockchains could consciously seek out jurisdictions with weak AML controls to engage with money laundering schemes or to avoid the costs of compliance.[66] During subsequent stages, the VCs can be freely transferred across borders between different wallets in any other

[63] Omri Marian, 'A conceptual framework for the regulation of cryptocurrencies' (2015) 82 University of Chicago Law Review Dialogue 53, 63.
[64] Andy Yee, 'Internet Architecture and the Layers Principle: A conceptual framework for regulating Bitcoin' (2014) 3(3) Internet Policy Review 3.
[65] FATF, Virtual Currencies 2014 (n 7) 10.
[66] Haffke et al (n 38) 5.

country. It can thus be concluded that regulation will only be adequate when it is adopted at a supranational or even global level.

3.1.4. Financial Crime Risk Analysis

The aforementioned characteristics not only enable end-users to disguise the source and purpose of money, making VCs appealing for money launderers and terrorist financiers, but they can also act as a payment mechanism between criminals.

In this context, it has been suggested that, before committing a crime, prospective criminals consider both the expected outcome from their criminal action as well as the probability and cost of any penalty that may be levied upon them.[67] Following this, they decide whether it is in their interest to engage in a particular crime.[68] Using VCs as a vehicle for nefarious activity significantly reduces the probability of detection due to the related anonymity. The reduced likelihood of being caught respectively decreases the deterrence factor. Therefore, in the absence of a regulatory response, the introduction of VCs would reasonably be expected to increase the level of criminal activity *ceteris paribus*, because more individuals would choose to engage in it.[69]

Admittedly, such unlawful behaviour also exists in the fiat payment system. Credit cards, e-banking, wire transfers and cash transactions all continue to be exploited for illicit activities.[70] Accordingly, the legal regulation of VCs should not necessarily aim to eliminate the interconnected financial crime risks. If that was the only concern, then a complete prohibition of VCs would be justified.[71] Instead, the regulation should inhibit the additional risks emerging from the specific features of VCs without preventing VCs from achieving their innovative potential. Consequently, it is necessary

[67] This is the classic utility model of criminal behaviour suggested by Gary S Becker at 'Crime and Punishment: An Economic Approach', (1968) 76(2) The Journal of Political Economy 169, 177.

[68] Becker's basic utility model is defined as:

$$E[U] = pU(Y-f) + (1-p) U (Y)$$

Where EU is the expected utility from engaging in criminal behaviour; p is the probability of conviction per offence; Y is the income expected to be generated from an offence; and f is the monetary equivalent of the criminal sanction. See Ibid.

[69] Marian (n 63) 60.

[70] Yaya J. Fanusie and Tom Robinson, 'Bitcoin laundering: an analysis of illicit flows into digital currency services' (2018) Centre on Sanctions & Illicit Finance, 13.

[71] Complete prohibition was the legal approach to VCs in some countries, such as Bangladesh, Bolivia, and Thailand. See Michael Sackheim and Nathan Howell (eds) *The Virtual Currency Regulation Review* (The Law Reviews 2018) 243.

to only target the anonymity, disintermediation and cross-border nature of VCs to the extent that they limit the current level of criminal activity.[72]

3.2.The Use of Virtual Currencies for Illicit Activities

The anonymity risks, in conjunction with the absence of sufficient regulation, instigate the offender’s motive and abet the following types of criminality.

3.2.1. Money Laundering

VCs are notoriously used to launder the proceeds of crime through two recognisable methods. Firstly, dirty fiat currency are converted into VCs and then put through a variety of transfers to obscure the fund’s illegal source.[73] Unlicensed entities, including VC exchanges, peer-to-peer trading platforms and tumbling services can be used to introduce illegal gains into the VC ecosystem as a result of the absence of strict KYC and other suspicious activity reporting (SAR) measures that financial institutions usually implement. Criminal counterparts can then complete the placement stage of money laundering and hide the nexus between the VCs and their origin.[74] A second *modus operandi* involves perpetrators selling illegal goods or services directly in exchange for VCs and subsequently converting those to fiat currency.

Law enforcement is already seeing cases of money laundering exploiting both centralised and decentralised VCs. The most famous case involving a centralised VC is the case of *Liberty Reserve*,[75] which is considered one of the largest online money laundering cases in history.[76] *Liberty Reserve* was an Internet-based payment system which issued its own VC and purposely assisted money laundering among criminals.[77] The VC, liberty dollar, was a bidirectional stablecoin and the transfers were denominated and stored in US dollars.

[72] Marian (n 63) 59.

[73] Carlisle (n 60) 14.

[74] The process of Money Laundering involves three recognisable phases: placement, layering and integration. See Jeffrey Simpser, ‘Money Laundering and asset cloaking techniques’ (2008) Journal of Money Laundering Control 15; Stefan Cassella, ‘Toward a New Model of Money Laundering: Is the ‘Placement, Layering, Integration’ Model Obsolete?’ (2018) 21 Journal of Money Laundering Control 494.

[75] Press Release, US DoJ, ‘Co-founder of Liberty Reserve Pleads Guilty to Money Laundering’ (2013) <http://www.justice.gov/opa/pr/2013/October/13-crm-1163.html> accessed 16 December 2019.

[76] According to Carlisle (n 60) 15, the value of the transactions involved is estimated at 8bn USD.

[77] Lawrence Trautman, ‘Virtual Currencies Bitcoin & What Now After Liberty Reserve, Silk Road, and Mt. Gox?’ (2014) 20(4) Richmond Journal of Law & Technology 87 <http://jolt.richmond.edu/v20i4/article13.pdf> accessed 16 December 2019.

Among decentralised VCs, an important ongoing money laundering case is the indictment of 'BTC-e' and its executive Alexander Vinnik[78] for the alleged crimes of conspiracy, money laundering and operating an unlicensed exchange.[79] The most notorious money laundering case, however, is the *Silk Road*, a cryptomarket[80] designed to enable its users to sell and purchase illegal goods and services. Its creator, Ross Ulbricht, was convicted *inter alia* of money laundering and sentenced to life imprisonment.[81] Other well-known cases include the VC exchanges BitInstant,[82] OKCoin,[83] where hundreds of thousands of US dollars were laundered, and Ripple, where a 700,000 USD civil penalty was levied to Ripple Labs Inc. for failure to maintain an adequate AML programme.[84]

3.2.2. Terrorist Financing

The rise of the Islamic State (IS), also known as Daesh, ISIS or ISIL, has raised concerns that terrorist organisations could deploy VCs to finance deeds of terrorism.[85] Specifically, it has been argued that the aforementioned characteristics of VCs make them 'ideal for terrorism financing'.[86] However, law enforcement actions against terrorist financiers who employ VCs are limited and largely anecdotal. In the US, an adolescent male was sentenced to more than 11 years for using a Twitter account to describe how to support the IS with bitcoin.[87] In Indonesia, authorities have claimed that IS militants have used PayPal and bitcoin for money transfers. However, they did

[78] Indictment *US v BTC-e and Alexander Vinnik*, case 4:19 (July 2019) US District Court Northern District of California <https://www.scribd.com/document/419868940/US-vs-BTC-e-Vinnik> accessed 16 December 2019.
[79] Daniel Kuhn, 'Prosecutors file formal complaint against infamous BTC-e crypto exchange' (CoinDesk, July 2019) <https://www.coindesk.com/formal-complaint-filed-against-infamous-btc-e-exchange> accessed 16 December 2019.
[80] See *infra* 3.2.3.
[81] *U.S. v Ulbricht*, 2014 US District LEXIS 93093.
[82] Jose Pagliery, 'Bitcoin Exchange CEO Arrested for Money Laundering' CNN Tech (January 2014) <https://money.cnn.com/2014/01/27/technology/security/bitcoin-arrest/index.html> accessed 16 December 2019.
[83] Gautham, 'Bitcoin Exchange OKCoin Fined in Money Laundering Case' (newsbtc, 2016) <https://www.newsbtc.com/2016/08/15/china-okcoin-exchange-fined/> accessed 16 December 2019.
[84] Press Release, 'FinCEN Fines Ripple Labs Inc. in First Civil Enforcement Action Against a Virtual Currency Exchanger' (2015) <https://www.fincen.gov/news/news-releases/fincen-fines-ripple-labs-inc-first-civil-enforcement-action-against-virtual> accessed 16 December 2019.
[85] Alan Brill and Lonnie Keene, 'Cryptocurrencies: The Next Generation of Terrorist Financing?' (2014) 6(1) Defence Against Terrorism Review 7.
[86] Iwa Salami 'Terrorism Financing with Virtual Currencies' (2018) 41 (12) Studies in Conflict &Terrorism, 968, 971.
[87] Press Release, US DoJ, 'Virginia man sentenced to more than 11 years for providing material support to ISIL' (2015) <https://www.justice.gov/opa/pr/virginia-man-sentenced-more-11-years-providing-material-support-isil> accessed 16 December 2019.

not share further details.[88] Generally, terrorist actions seem to require simpler forms of funding and terrorist financing through VCs ought to be regarded as a potential risk, rather than a crystallised one.[89]

3.2.3. *Cryptomarkets*

Cryptomarkets, such as the infamous drug marketplaces *Silk Road* and *AlphaBay*, are online platforms designed to facilitate the trade of illicit commodities. They provide their participants with anonymity as they are located on the dark web and they utilise cryptocurrencies for payment. The most tradeable products in cryptomarkets include illegal drugs, stolen information, weapons, pornographic content and other illicit services, such as hacking for hire.[90]

It should be noted however that cryptomarkets anonymity is not achieved by VCs per se. Instead, end-users obfuscate their IP addresses and encipher their communication by operating on the Tor network and utilising encryption software. The role of VCs is restricted as a means of payment. As explained above, the pseudonymous nature of well-known VCs empowers authorities to relate transactions and pinpoint the identity of the owner of an address. This became evident in the *Silk Road* case where the arrest of Ulbricht enabled law enforcement to track illegal transactions and identify a number of participants including his employees,[91] drug traffickers,[92] bitcoin vendors[93] and the FBI agents who initially investigated and blackmailed him.[94]

To achieve complete anonymous transactions, participants in cryptomarkets usually prefer to use altcoins of enhanced privacy, as for example Dash, Monero (XMR), NEO and DarkCoin (DARK).[95] These VCs are not usually available in virtual-to-fiat currency exchanges,[96] rather they are usually used as a peer-to-peer payment

[88] Pete Rizzo, 'Indonesia's AML watchdog links bitcoin to IS' (CoinDesk, January 2019) <https://www.coindesk.com/indonesias-aml-agency-links-bitcoin-islamic-state-terrorism> accessed 16 December 2019.

[89] See Europol, 'Changes in Modus Operandi of Islamic State Terrorist Attacks', (January 2016) 7.

[90] Jake Frankenfield, 'Darknet Market' (Investopedia, February 2018) <https://www.investopedia.com/terms/d/darknet-market-cryptomarket.asp> accessed 16 December 2019.

[91] *US v. Jones, Davis, and Nash*, (2013) <https://www.justice.gov/usao-sdny/pr/manhattan-us-attorney-announces-charges-against-three-individuals-virginia-ireland-and> accessed 16 December 2019.

[92] *US v. Sadler & White* (2013) <http://www.bellevuereporter.com/news/bellevue-couple-charged-with-silk-road-drug-sales/> accessed 16 December 2019.

[93] *US v. Faiella*, 39 F.Supp.3d 544 (S.D.N.Y. 2014).

[94] *US v. Force and Bridges* <https://www.courtlistener.com/docket/4181958/united-states-v-bridges/> accessed 16 December 2019.

[95] Details about these VCs accessible at https://coinmarketcap.com/.

[96] For example, in *Coinbase* none of these VCs are available.

method between illicit actors. However, there are many pure cryptocurrency exchanges which enable the conversion of these VCs to other VCs, thereby enabling cryptomarket vendors to deviously channel their proceeds into the fiat financial system. Thus, it is important to impose appropriate regulatory measures to hinder the flow of tainted VCs into the real economy.

IV. Legal Regulation of Virtual Currencies

4.1.International Initiatives

In various international fora, the rise of VCs has caused concern. In June 2019, the Financial Action Task Force (FATF), adopted an ‘Interpretive Note to its Recommendation 15’ (hereinafter IN) to respond to the increasing use of VCs for money laundering and terrorist financing.[97] The amended recommendations require member countries to ensure that ‘Virtual Asset Service Providers’ (hereinafter VASPs) implement AML and KYC controls and be subject to effective monitoring and reporting obligations. On the same day, FATF also adopted a new guidance for a risk-based approach,[98] which includes a comprehensive recommendation-by-recommendation analysis for the implementation of the preventive measures in the context of VCs.

As stated by FATF, to qualify as a VASP an entity must both act as a business on behalf of customers and actively facilitate VC-related activities.[99] These activities include (i) exchange between VCs and fiat currencies, (ii) exchanges between one or more forms of VCs, (iii) transfer of VCs from one address or account to another, (iv) safekeeping or administration of VCs or instruments enabling control over VCs and (v) provision of services related to an issuer’s offer of a VC.[100]

Almost every blockchain system participant is covered by this definition. Its ambit includes both virtual-to-fiat as well as virtual-to-virtual exchangers. Providers of kiosks, ‘bitcoin ATMs’ and VC brokerage services are also included in the above definition. Moreover, peer-to-peer or decentralised trading platforms may fall under the

[97] Press Release, FATF, ‘Public Statement on Virtual Assets and Related Providers’ <http://www.fatf-gafi.org/publications/fatfrecommendations/documents/public-statement-virtual-assets.html> accessed 16 December 2019.

[98] FATF, ‘Virtual Assets and Virtual Service Providers: Guidance for a risk-based approach’ (June 2019) 13 <https://www.fatf-gafi.org/media/fatf/documents/recommendations/RBA-VA-VASPs.pdf> accessed 16 December 2019.

[99] Ibid 4.

[100] Ibid 13-14.

category of VASP where the platform facilitates the exchange, transfer, or other financial activity involving VCs.

In the context of wallet providers, only custodian wallet providers fall within the limits of part (iv) of the VC-related activities definition, as they safeguard the private key attached to an address.[101] Non-custodian wallet providers do not seem to provide a VC-related activity, because they do not usually have any control over the VCs. Finally, Issuers and other facilitators of ICOs are covered from the element (v). Notably, this definition promotes a 'technological neutrality' by determining whether a person engaged in VC-related activities is a VASP based on whether he is engaged as business or as an end-user, regardless of the technology deployed.[102]

The FATF standards revolve around information sharing, KYC and CDD procedures as well as effective, proportionate and dissuasive sanctions.[103] A significant novelty introduced by the IN is that VASPs ought to be licensed or registered in the jurisdiction where they are created.[104] The IN acknowledges the money laundering and terrorist financing risks emerging from the global nature of VCs and it tables the broadest possible range of international cooperation.[105]

The same approach was adopted by the bank of international settlements (BIS) and the financial stability board (FSB).[106] In particular, bank of international settlements suggested that FATF should foster the global implementation of its standards and 'focus on the point at which a cryptocurrency is exchanged into a sovereign currency' and 'on cryptocurrency infrastructure providers, such as crypto wallets'.[107]

4.2.Developments in the EU

The inherent cross-border nature of VCs renders state-based legislative responses to VCs insufficient. An international Treaty for the regulation of VCs apart

[101] FATF, 'Virtual Assets and Virtual Service Providers' (n 98) 16.

[102] The term technological neutrality is used here with the meaning that the same regulatory principles should apply regardless of the technology used. See Winston Maxwell and Marc Bourreau 'Technology Neutrality in Internet, Telecoms and Data Protection Regulation' (2014) 1 Computer and Telecommunications Law Review 2.

[103] FATF, 'Virtual Assets and Virtual Service Providers' (n 98) 56.

[104] IN (n 98) par. 3.

[105] Ibid par. 8.

[106] FSB, 'Crypto-asset markets: Potential channels for future financial stability implications' (October 2018), 15.

[107] BIS, 'BIS Annual economic report' (2018), 107

from highly unlikely, would also be ineffective. This is because the time-consuming procedures that accompany an international Treaty render the latter unsuitable for regulating a fast-moving industry that orientates its innovation towards evading regulation. Considering that a solution is needed at a supranational level, this chapter analyses the applicability of EU legal frameworks on payment services, electronic money and AML to VCs. It is noted that the second Directive on markets in financial instruments (MiFID)[108] is not contemplated here since 'payment-type crypto assets are unlikely to qualify as financial instruments.'[109]

4.2.1. Payment Services Directives

A key issue for the qualification of VCs in the EU regulation is whether they qualify as a 'fund' as per the payment services directives. That is because if a VC qualifies as a 'fund' under the second payment services directive (PSD2),[110] the provisions of the directive would apply to VC service providers, reducing significantly their suitability for financial crime abuse. According to article 4 par. 25 of the PSD2, the notion of funds is defined as 'banknotes and coins, scriptural money and electronic money'. This is yet problematic because we defined earlier VCs as 'tokens that can be used as a means of payment without having a legal tender status'. Hence, the PSD2 will not apply to classic decentralised VCs that do not qualify as e-money. A different approach suggested by the French banking supervisor (ACPR) arguing that VC exchanges are providing payment services since they receive 'banknotes and coins, scriptural money and electronic money' in exchange for VCs has gained only limited popularity.[111]

[108] Directive 2014/65/EU of the European Parliament and of the Council of 15 May 2014 on markets in financial instruments and amending Directive 2002/92/EC and Directive 2011/61/EU [2014] OJ L 173/349.

[109] ESMA, 'Advice: Initial Coin Offerings and Crypto-Assets' (January 2019) 19. Similarly, given that the article focuses on the payments industry, the discussion regarding the applicability of general property law concepts to VCs is beyond the scope of this study. Cf. *AA v Persons Unknown* [2019] EWHC 3356 (Comm) according to which VCs are property within the meaning of English law.

[110] Directive (EU) 2015/2366 of the European Parliament and of the Council of 25 November 2015 on payment services in the internal market, amending Directives 2002/65/EC, 2009/110/EC and 2013/36/EU and Regulation (EU) No 1093/2010, and repealing Directive 2007/64/EC [2015] OJ L 337/35 (hereinafter PSD2).

[111] Nikolaos Theodorakis, 'The Use of Cryptocurrencies for Illicit Activities and Relevant Legislative Initiatives' (2018) The Art of Crime 12 <https://theartofcrime.gr/the-use-of-cryptocurrencies-for-illicit-activities-and-relevant-legislative-initiatives> accessed 16 December 2019.

Overall, it appears to be a consensus that the PSD2 does not leave room to include VCs in the notion of 'funds'.[112] This is envisaged in the 5AMLD where it is stated that 'VCs should not be confused with the larger concept of funds as defined in point (25) of Article 4 of PSD2'.[113] Intriguingly, this choice has been made out of fear that if VC exchange platforms are subject to licensing and safeguarding requirements, this will further legitimise the use of VCs and create the unwarranted misconception to consumers that VCs are safe products.[114] This circular logic, however, is open to criticism on the basis that regulatory oversight usually urges regulated entities to greater transparency regarding their services and the associated risks.

4.2.2. E-money Directives

As elaborated above, VCs would only constitute regulated 'funds' if they qualify as e-money. The second e-money directive (EMD2) defines 'electronic money' as follows:[115]

> [E]lectronically, including magnetically, stored monetary value as represented by a claim on the issuer which is issued on receipt of funds for the purpose of making payment transactions as defined in Article 4 of [PSD2], and which is accepted by a natural or legal person other than the electronic money issuer.[116]

According to this definition, classic decentralised VCs that operate in permission-less blockchains, such as Bitcoin or Monero, will not qualify as electronic money because they are not represented by a holder's claim on the issuer. It could be proposed that this definition applies to utility tokens, which incarnate a relationship between the token issuer and the token holder.[117] This view, however, neglects to consider that utility tokens are not usually accepted by a person other than the issuer. Even if these tokens are traded in the secondary market, they are still not accepted as a means of payment; rather they are traded for investment objectives.

[112] Sergii Shcherbak, 'How Bitcoin Should be regulated?' (2014), 7(1) European Journal of Legal Studies 45, 61;

[113] 5AMLD, recital 10.

[114] European Banking Authority (EBA), 'Opinion of the EBA on the EU Commission's proposal to bring VC into the scope of Directive 2015/849' (2016) 4-5.

[115] Directive 2009/110/EC of the European Parliament and of the Council of 16 September 2009 on the taking up, pursuit and prudential supervision of the business of electronic money institutions amending Directives 2005/60/EC and 2006/48/EC and repealing Directive 2000/46/EC [2009] OJ L 267/7.

[116] Article 2(2) EMD2.

[117] See, *supra* 1.2.

This paper suggests that certain VCs could potentially qualify as e-money within the range of the EMDs.[118] Specifically, some VCs, such as stablecoins, may not only be accepted as a means of payment by other entities, but also represent a claim on the issuer.[119] For example, Libra will be represented by a claim against the Libra Association.[120] The latter will be the sole issuer of Libra, namely the only party able to create and destroy Libra.[121] Considering that Libra's business proposition 'is to enable a simple global currency'[122] and that, at the time of this writing, the consortium consists of many leading organisations, including Facebook, Spotify, Shopify, Uber, Lyft and Farfetch,[123] it is concluded that Libra will also be accepted by a legal person other than the issuer.[124]

In such cases, stablecoins should qualify as e-money and the issuer of such VCs may require an authorisation as an e-money institution.[125] As a consequence, these VCs would fall into the definition of 'funds' and the VC service providers may require additional licences as 'e-money institutions' under EMD2 and 'payment services providers' under the PSD2. Therefore, wallet providers and validating nodes would perform all the AML/CFT/CDD checks on users in the same way that current regulated digital banks and payment institutions do. As a result, the use of regulated VCs will not exacerbate the aforesaid financial crime risks as compared to fiat money.

4.2.3. Anti-Money Laundering Directives

The majority of VCs, however, do not fall within the scope of the e-money definition. They are unregulated and exploited for illicit activities, such as money laundering and illicit financing. In the EU law, the 4AMLD is the main regulatory response to these activities. It applies to 'obliged entities', such as financial institutions, estate agents, and other legal persons. Generally, it is accepted that VC system

[118] See *contra* Stefaan Loosveld, 'The 5th Anti-Money Laundering Directive: Virtual Currencies and Other Novelties' (2018) 33 Journal of International Banking Law and Regulation 297, 301.

[119] EBA, 'Report with advice for the European Commission on crypto-assets' (January 2019) 13.

[120] Cristian Catalini et al., 'The Libra Reserve', 2 <https://libra.org/en-US/about-currency-reserve/?noredirect=en-US#the_reserve> accessed 16 December 2019.

[121] See Libra Association (n 3) 8.

[122] Ibid 1.

[123] Libra Association (n 3) 4.

[124] Dirk Zetzsche, Ross Buckley and Douglas Warner, (2019) 47 'Regulating Libra: The transformative potential of Facebook's Cryptocurrency and Possible Regulatory Responses' University of New South Wales Law LRS, 17-18.

[125] EBA, 'Report with advice for the European Commission on crypto-assets' (January 2019) 13.

participants are not included in the notion of obliged entities under the (pre-amended) 4AMLD.[126]

By launching the fifth AMLD, EU legislators aimed, for the first time ever, to target directly the financial crime risks emerging from the anonymity attached to VCs.[127] They espouse a 'balance and proportional approach' by empowering competent authorities to monitor the use of VCs through obliged entities.[128] The 5AMLD achieves this by introducing to the 4AMLD two new obliged entities: 'Providers engaged in exchanges between virtual currencies and fiat currencies' and 'custodian wallet providers'.[129] Given that most users buy VCs through exchange platforms and use custodian wallet providers in their payments, they will now be required to verify their identity toward those intermediaries.

The 5AMLD acknowledges, however, that these amendments will not adequately mitigate the anonymity risks attached to VC transactions, since users can also transact without such providers.[130] Therefore, it is envisaged that national financial intelligence units (FIUs) should be able to gather information facilitating them to associate VC addresses to the identity of the owners.[131] In this regard, the Directive leaves open the possibility of creating a central database with the details of all end-users.[132]

V. Critical Analysis of the EU regulatory framework

5.1.The definition of Virtual Currencies

The EU legal definition on VCs explicitly refers that they 'must be accepted by natural or legal persons as a means of exchange'.[133] According to the taxonomy of VCs on the basis of their functionality, only 'exchange' or 'currency tokens' are intended to be used as a means of exchange, while 'utility' or 'investment' tokens provide different

[126] Niels Vandezande, *Virtual Currencies: A Legal Framework* (Cambridge Intersentia 2018), 298; 5AMLD, recital 8.
[127] 5AMLD recital 9.
[128] Ibid recital 8.
[129] Ibid Art. 1(1)(c)(g) and Art. 1(1)(c)(h).
[130] 5AMLD recital 9.
[131] Thomas Frick, 'Virtual and Cryptocurrencies—Regulatory and Anti-Money Laundering Approaches in the EU and in Switzerland' (2019) 20 ERA Forum Journal of the Academy of European Law <https://link.springer.com/article/10.1007/s12027-019-00561-1> accessed 16 December 2019.
[132] 5AMLD Art. 65(1) *in fine*.
[133] See *supra* Part 1.1.

kind of rights to their holders. With the above in mind, investment and utility tokens do not fall within the perimeter of the 5AMLD.

This is problematic for two reasons: First, the wording of the definition is conflicting with Recital 10 of the 5AMLD which suggests that 'the objective of this Directive is to cover all the potential uses of VCs' and not only their use as a medium of exchange.[134] Secondly, all kinds of tokens can be used for money laundering and terrorist financing. Besides, tokens that function as investment instruments can easily be traded for payment-type VCs in a secondary market.

In the author's view, the national legislator would need to make explicit the application of the 5AMLD to VCs that are used for investment or store-of-value purposes.[135] At the EU level, an altered definition should be adopted analogous to FATF's definition on virtual assets. The latter encompasses tokens that 'can be used for payment or investment purposes.'[136] Finally, the wording of the reformed definition should be open-ended in order to cover all possible type of tokens.

5.2.Assessing the Scope of the Regulation

By including virtual-to-fiat exchange services and custodian wallet providers under the scope of the AMLDs, the EU legislator attempted to mitigate the anonymity risks related to the use of VCs. The author opines that the scope of the regulation is still inadequate, and it should be further broadened in order to ameliorate law enforcement's ability to trace transactions.

5.2.1. Virtual Currency Exchanges

Under the 5AMLD, all virtual-to-fiat currency exchanges qualify as obliged entities for AML/CFT purposes as long as they trade at least one VC within the definition of the 5AMLD against fiat currency.[137] As things stand, the Directive will only cover services that exchange 'currency' tokens against fiat currency and not providers engaged in exchange of 'utility' or 'investment' tokens against fiat currency.[138]

[134] 5AMLD, recital 10.

[135] Contrary, the application of the payment services directive and e-money directive is not indicated to this type of tokens.

[136] FATF, June 2019 (n 98) 13.

[137] Houben and Snyers (n 5) 77.

[138] See *supra* Part 5.1.

Furthermore, pure crypto exchanges as set out in Chapter 1.3. remain out of the 5AMLD's perimeter because they have no dealing with fiat currency. For example, a provider that offers a collection of well-known altcoins and accepts payments exclusively in bitcoin, as long as it does not qualify as a custodian wallet provider, is not covered by the 5AMLD. Even though one could argue that virtual-to-virtual exchanges bear no direct relation with the fiat financial system, it is suggested that they can equally be exploited from illicit actors to disguise the source of their VCs as well as be used to facilitate illicit transactions and means of payment.

This poses additional risks in the fast-moving world of VCs, as the network of actors that accept VCs as a means of payment can grow vastly. For instance, in New Zealand paying salaries on VCs has been legalised.[139] If VCs effectively become broadly accepted, the need to exchange VCs for fiat money through an exchanger might diminish over time. Consequently, virtual-to-virtual exchanges could be a focal point of control in monitoring money being transferred within VC networks.

In the light of the above, member states should include pure cryptocurrency exchanges into the list of the obliged entities. Similarly, the EU legislator should overhaul its list of obliged entities pursuant to the FATF's notion of VASP, which ingeniously embraces exchange services between one or more forms of VCs.

5.2.2. Tumbler Services

Even though tumbler services are the entities that, for the most part, amplify the anonymity risks associated with the use of VCs, they are not included as obliged entities in the 5AMLD. Given that tumbler services obscure the chain of transaction on one blockchain at a time, they could not qualify as virtual currency exchangers either. Thus, the article argues that the EU list of obliged entities should be amended to include not only virtual-to-virtual exchanges, but also exchanges within one form of VCs.

5.2.3. Wallet Providers

As stated above, custodian wallet providers have been incorporated in the list of obliged entities under 5AMLD, while non-custodian wallet providers have not been incorporated. The same approach is followed by FATF, since VASPs contain only

[139] Daniel Palmer, 'New Zealand Tax Office Make it Legal to Pay Salaries in Crypto' (CoinDesk, August 2019) <https://www.coindesk.com/new-zealand-tax-office-makes-it-legal-to-pay-salaries-in-crypto> accessed 16 December 2019.

services that have exclusive or independent control of the private key associated with an address.

It could be argued that wallet providers may elude regulation by requiring end-users to enter manually their private key. The question then becomes whether the scope of the 5AMLD should be extended to non-custodian wallet providers.[140] This will result, allegedly, in complete transparency of the virtual transactions and maybe in overregulation of VCs.[141] However, it is not always possible to extend the regulation to non-custodian wallet providers. For example, a hardware wallet provider does not provide a continuous service, but simply a product in which he is not supposed to have any oversight after the transfer.

On this point, the regulation seems to keep up with the current VC market developments, as most wallet providers are custodial. An additional step leading towards stricter regulation could be to further regulate the software non-custodian wallet providers, which provide end-users with software applications and interfaces that allow them both to access the network and save their keys locally. This will obstruct the above-mentioned potential loophole of the regulation.

5.2.4. Trading Platforms

Trading platforms that facilitate peer-to-peer transfer of VCs constitute a blind spot in the EU framework. Insofar they do not provide wallets to their users, trading platforms can function as unregulated online marketplaces where different VC holders can meet and interact directly. This has been acknowledged by FATF, which has incorporated entities that facilitate the transfer of VCs into the definition of VASP.[142]

It has been highlighted, however, that many trading platforms are 'decentralised', thus, it would be very hard to regulate them.[143] Even though the enforcement of the regulation might be complicated, law enforcement agencies should nevertheless have the appropriate legal infrastructure to do so, to the extent they can identify the owner or the operator of such platform . Consequently, trading platforms should be included in the list of obliged entities in the 5AMLD.

[140] Haffke et al (n 38) 13.

[141] Ibid.

[142] FATF, June 2019 (n 98), 16.

[143] IOSCO, 'Issues, Risks and Regulatory Considerations Relating to Crypto-Asset Trading Platforms' (May 2019) 17 <https://www.iosco.org/library/pubdocs/pdf/IOSCOPD627.pdf> accessed 16 December 2019.

5.2.5. *Issuers of Virtual Currencies*

As stated above VC issuers could be divided in coin inventors and coin offerors.[144] It is evident that coin inventors are not obliged entities under 5AMLD. Contrary, coin offerors could be considered 'providers that engage in exchange services between virtual currencies and fiat currencies' as long as they accept payment in fiat money.[145] If pure crypto exchanges become obliged entities according to our proposed alterations,[146] all coin offerors could fall within the scope of the 5AMLD.

One could argue that the suggestion that the EU legislator unwittingly incorporated coin offerors to the definition of VC exchanges is arbitrary. Specifically, the recitals of the Directive do not mention coin offerors; if the legislator wanted to regulate the VC issuers, he would have simply done that explicitly. In the author's view, VC issuers should be regulated with more clarity in the Directive. They could be included as a separable obliged entity within the scope of the 5AMLD in the following form: 'providers engaged in services related to an initial offer of a VC'.[147]

5.3.Regulating end-users: A proposal in Dispute

As we have seen throughout this research, regulation needs to be balanced between the attractiveness of VCs and the financial crime risks identified above. The EU legislator has taken, hitherto, a very soft approach towards unveiling the anonymity of end-users. Users that do not hold their VCs via a custodian wallet provider and do not transact through a regulated virtual exchange can still operate anonymously. The suggested here widening of the scope of the 5AMLD will impede only partially the enhanced anonymity risks since users could still transact peer-to-peer without using any intermediary.

In this respect, it has been suggested by the EU legislator that it might become necessary to regulate end-users by registering their identities.[148] User's registration could take place either voluntarily or mandatorily.[149] A voluntary registration would, however, be ineffective since it would be absurd to trust a money launderer to willingly register his identity. In contrast, a mandatory registration would be very intrusive, and

[144] See *supra* Part 1.3.
[145] Haffke et al (n 38) 19.
[146] See *supra* Part 5.2.1.
[147] Derived from FATF June 2019 (n 98), 16.
[148] 5AMLD Art. 65(1) *in fine*.
[149] Houben and Snyers (n 5) 80.

it might stifle the potential of VCs. Specifically, it would foist on end-users a regulatory burden that they would not have if they had chosen to transact with fiat money. In addition to this, it would be impractical for national FIUs to maintain such central databases. Therefore, applying this regulation directly to end-users is not considered an appropriate and proportional legislative suggestion.

In legal theory it has been proposed to regulate end-users indirectly by holding them vicariously liable for interacting with undesirable VCs.[150] For example, users transacting with a permission-less blockchain-based VC which is well-known for being exploited by illicit actors are ultimately responsible for its value and maintenance. Hence, it might seem reasonable to hold these users vicariously liable for facilitating the unlawful activities stemming from the use of this VC by other users. Imposing such a risk to the end-users would function as a powerful deterrence for end-users to transact with high-risk VCs.[151] However, this could create causation problems, is highly debatable from a fairness perspective and it will undoubtedly impede the adoption of the lawful technology and stifle innovation.

5.4.Introducing a comprehensive approach

For the purpose of pulling more end-users into the light, a more invasive approach towards anonymity is warranted. In this paragraph, it is argued that VCs should be regulated according to their underlying technology.

Permissioned Blockchain based VCs, such as Libra, JPM coin or XRP Ripple, should qualify as ‘licenced VCs’. This would entail the EU financial services law applying to these VCs. As stated above, some of these VCs fulfil the requirements of the definition of e-money.[152] The article suggests that the e-money definition should be revised in order to explicitly include all VCs that operate in a permissioned blockchain payment system. Consequently, issuers of such VCs would require authorisation as e-money institutions. Accordingly, entities which undertake blockchain payment services, such as the execution of payments or money remittance, would fall within the perimeter of the PSD2[153] and they would require appropriate licencing. Hence, the licenced blockchain system participants would perform all the AML/CFT/KYC

[150] P De Filippi and A. Wright, *Blockchain and the Law: The Rule of Code* (Cambridge, Harvard Press 2018), 176.
[151] Ibid.
[152] See *supra* Part 4.2.2.
[153] Payment services are listed exhaustively in Annex I of the PSD2.

obligations in the same way as the fiat payment services providers. This will comprehensively thwart all the identified FC risks since regulated intermediaries will provide monitoring on the transactions, thereby rendering VCs unattractive for illicit behavior.

Permission-less blockchain VCs, such as Bitcoin, Ether or Monero, should qualify as ‘unlicenced VCs’. This implies that the EU financial services law shall not apply to these VCs. The absence of known issuers and central administrators renders the regulation of permission-less blockchain VCs through obliged entities as the only viable solution. The widening of the perimeter of the 5AMLD, by expanding the list of obliged entities, will undoubtedly bring more users’ identities into light. On the other hand, illicit actors and so-called ‘cypherpunks’ are increasingly developing new decentralised methods that improve the privacy of transactions. For example, a John Hopkins University professor has designed Zerocoin, an extension to Bitcoin that provides end-users with anonymity without being a tumbler service.[154] It is expected that without a comprehensive regulatory framework, code developers will constantly be a step ahead of regulation.

In the author’s view, a comprehensive solution that would tackle the anonymity associated with permission-less blockchain-based VCs is two-fold: First, apart from accepting the alterations proposed in Chapter 5.2., all VC service providers should also be licenced in a way that they would be accountable to the authorities. Second, and most important, it should be made mandatory for users to transact only through a licenced intermediary. For example, end-users would only hold and trade VCs lawfully if they do so through a licenced custodian wallet provider. This will allow the supervisory authorities to attach regulation to an identifiable third person. The latter entity could also be held accountable for remedying victims of fraud and other aggrieved users. The regulatory burden will not be imposed to end-users but to businesses, thence the attractiveness of VCs will not be stifled. These provisions could be either inserted as special provisions to the current AMLDs or form part of a framework exclusively for VCs.

A probable counterargument against our regulatory proposal for permission-less blockchain VCs would be that part of the Blockchain’s innovative potential is that

[154] http://zerocoin.org/.

has cut out such 'middleman'. The same argument about the removal of all middleman was repeatedly being made when the Internet first pushed into mainstream consciousness.[155] Nonetheless, despite the fact that the Internet eliminated the need for some middleman, as it gained broad adoption it enabled the development of new intermediaries to which regulation could be attached.[156] A similar pattern is unfolding in the course of blockchain-based systems. It is argued that imposing just one quasi-mandatory intermediary would not restrict the revolutionising potential of permission-less blockchain payment systems since the latter still cut-off a whole chain of banking intermediaries.

Lastly, meta-regulation should not be overlooked. The latter suggests that regulators may seek to induce the regulated entities to develop their own, internal, self-regulatory responses to public problems.[157] In our context, the proposed regulatory imposition will prompt the entities engaging in VC activity to co-regulate themselves and accept in the course of their business only VCs stemming from lawful wallets.[158] Illicit users operating without a licenced intermediary would not be able to transact with lawful users. Consequently, the expected utility from engaging in VC related criminal behaviour will be reduced significantly and the financial crime risks will be mitigated in a virtual ecosystem which would be clearly delineated.

C. **Conclusion**

VCs present a challenge for legislators around the globe. On the one hand, they have the power to enhance or even supplant the traditional payment systems. They could offer a number of potential benefits, and regulators should be careful not to hinder these innovations. On the other hand, VCs create new financial crime risks. Using the classic utility model of criminal behavior as a lynchpin, this inquiry identified the risks

[155] Andrew L Shapiro, 'Digital Middleman and the Architecture of Electronic Commerce' (1998) 24 Ohio Northern University Law Review 795.
[156] P De Filippi and A. Wright, *Blockchain and the Law: The Rule of Code* (Cambridge, Harvard Press 2018), 179.
[157] Cary Coglianese and Evan Mendelson, *The Oxford Handbook of Regulation* (Oxford University Press 2010), 150.
[158] The term co-regulation is preferred by some legal scholars; see Michele Finck, *Blockchain Regulation and Governance in Europe* (Cambridge University Press 2019) 172 who argues that a model of polycentric co-regulation would be a suitable approach for regulating decentralised blockchain ecosystems.

raised by the anonymity, decentralisation and cross-border nature of VCs in the context of money laundering, terrorist financing and illicit flow of money.

The first research question probed the extent at which VCs are regulated under the current EU regulatory framework on e-money, payment services and AML. The main conclusion to be drawn with regard to this question is that *de lege lata* only the AML regime undoubtedly applies to VCs. Contrary, the applicability of the EMD2 and PSD2 to VCs is open to interpretation and should be examined on a case-by-case basis.

The second research question was how regulation needs to be reorientated in order to balance between the attractiveness of VCs and the financial crime risks associated therewith. On this question, the research attempted to introduce a comprehensive solution based on the underlying technology of VCs. With regard to permissioned blockchain-based VCs, the article proposed that the EMD2 and PSD2 are in need of modification to explicitly integrate these VCs under their ambit. In respect of VCs that operate in permission-less blockchains, it is firmly suggested to impose a compulsory intermediary between the end-user and the blockchain.

To that end, the utterly anonymous peer-to-peer transactions that the creators of modern virtual currencies envisioned should be outlawed. The anonymity as privacy that end-users require, can be preserved by applying appropriate additional regulation, such as the General Data Protection Regulation of the EU, to the regulated intermediaries.

Discussion Paper #4

Bitcoin Governance as a Decentralized Financial Market Infrastructure

Hossein Nabilou

University of Luxembourg, Faculty of Law, Economics and Finance

Abstract

Since Bitcoin was invented a decade ago, the phenomenon of Virtual Currencies has been hailed as an ingenious innovation and decried as the preferred transaction vehicle for illicit actors. Despite the numerous headlines discussing the virtues and vices of virtual currencies, heretofore there has been no comprehensive legal response. Regulators seem to contemplate the wrong question; they wonder whether they should regulate virtual currencies instead of how to regulate them.

The present contribution elaborates on the second question. Starting with a conceptual analysis of virtual currencies and their promising potential, it identifies the financial crime risks posed by the intersection between legitimate and illegitimate users. The research shows that a fragmentary regulation would be ineffective; this promising technology will either be integrated into the lawful economy or it will be exploited by criminals. The paper attempts to fill the regulatory gap by providing a recommendation for the embeddedness of Virtual Currencies into the EU financial system. It achieves that by redirecting regulation towards the uniqueness of their underlying technology.

Keywords: Bitcoin, Cryptocurrency, Blockchain, Governance, Censorship resistance

JEL classification: E42, E51, E58, G01; G23; G28; K22; K23, K24

Bitcoin Governance as a Decentralized Financial Market Infrastructure

Hossein Nabilou[*]

Abstract

Bitcoin is the oldest and most widely established cryptocurrency network with the highest market capitalization among all cryptocurrencies. Although bitcoin (with lowercase b) is increasingly viewed as a digital asset belonging to a new asset class, the Bitcoin network (with uppercase B) is a decentralized financial market infrastructure (dFMI) that clears and settles transactions in its native asset without relying on the conventional financial market infrastructures (FMIs). To be a reliable asset class as well as a dFMI, however, Bitcoin needs to have robust governance arrangements; whether such arrangements are built into the protocol (i.e., on-chain governance mechanisms) or relegated to the participants in the Bitcoin network (i.e., off-chain governance mechanisms), or are composed of a combination of both mechanisms (i.e., a hybrid form of governance).

This paper studies Bitcoin governance with a focus on its alleged shortcomings. In so doing, after defining Bitcoin governance and its objectives, the paper puts forward an idiosyncratic governance model whose main objective is to preserve and maximize the main value proposition of Bitcoin, i.e., its censorship-resistant property, which allows participants to transact in an environment with minimum social trust. Therefore, Bitcoin governance, including the processes through which Bitcoin governance crises have been resolved and the standards against which the Bitcoin Improvement Proposals (BIPs) are examined, should be analyzed in light of the prevailing narrative of Bitcoin as a censorship-resistant store of value and payment infrastructure. Within such a special governance model, this paper seeks to identify the potential shortcomings in Bitcoin governance by reference to the major governance crises that posed serious threats to Bitcoin in the last decade. It concludes that the existing governance arrangements in the Bitcoin network have been largely successful in dealing with Bitcoin's major crises that would have otherwise become existential threats to the Bitcoin network.

Keywords: *Bitcoin, Cryptocurrency, Blockchain, Governance, Censorship resistance*

JEL classification: *E42, E51, E58, G01; G23; G28; K22; K23, K24*

[*] Postdoctoral researcher in Banking and Financial Law, University of Luxembourg, Faculty of Law, Economics and Finance. Research Associate at the University College London, UCL Centre for Blockchain Technologies (UCL CBT), and the Amsterdam Center for Law and Economics (ACLE), University of Amsterdam Law School. E-mail: hossein.nabilou@uni.lu
The author is grateful to the participants in the Max Planck Institute Luxembourg for International, European and Regulatory Procedural Law Conference on Blockchain and Procedural Law: Law and Justice in the Age of Disintermediation (15 November 2019) held in Luxembourg for their comments and feedback. All errors are those of the author.

Introduction

In November 2018, the Society for Worldwide Interbank Financial Telecommunication (SWIFT) - a financial market infrastructure (FMI) institution that provides secure messaging for international payments - suspended certain Iranian banks' access (including that of the Central Bank of Iran) to its messaging system. This step was seemingly taken to protect the stability and integrity of the global financial systems.[1] However, the reluctant tone of the announcement could hardly disguise the reality that SWIFT took such an action conceding to the US government push to expand the secondary sanctions on financial messaging services to the Central Bank of Iran. Although the US did not have jurisdiction over SWIFT, which is a cooperative company incorporated under the Belgian law and is owned and controlled by its shareholders (financial institutions), after months of intense negotiations about the US demands and irrespective of the dismay expressed by European officials, SWIFT had to acquiesce.[2] In making SWIFT to yield to US demands, the US government apparently went to such an extent as to threaten the SWIFT's twenty five board members (which include two US bankers from Citi Bank and JPMorgan) with visa bans and asset freezes, and its member banks with charges and fines.[3]

The intrusion of considerations beyond the scope of financial regulation in the operation and risks management of Financial Market Infrastructures (FMIs),[4] was of such proportions that triggered radical proposals for reforming and restructuring the international FMI institutions. The recent calls for establishing international payment rails independent of the US have shown the frustration with the hegemony of a single dominant player having formal (i.e., through extraterritorial application of its laws or through secondary sanctions) and informal dominance over international payment infrastructures.[5] For example, German foreign minister Heiko Maas proposed that Europe could create its own SWIFT rival based on the euro rather than the US dollar (USD).[6] More recently, speculations about Synthetic Hegemonic Currency (SHC), which would be provided by the public sector through a network of central bank digital currencies

[1] Michael Peel, *Swift to comply with US sanctions on Iran in blow to EU*, FINANCIAL TIMES, November 5, 2018.; SWIFT, *Update: Iran Sanctions Agreement*, (17 January 2016). This was not the first time that SWIFT cut access to the Central Bank of Iran (CBI). CBI's access was cut in 2012. *See* SWIFT, SWIFT instructed to disconnect sanctioned Iranian banks following EU Council decision (15 March 2012).

[2] SWIFT, *Compliance: How is SWIFT governed?*, (Undated).

[3] Katrina Manson, *Europe steps up drive to exempt Swift from Iran sanctions*, FINANCIAL TIMES, Octover 9, 2018.

[4] Klaus Löber, *Extraterritorial Application or Regulation in the Area of Financial Market Infrastructure: The Case for Cross-Border Cooperative Oversight*, *in* EUROPEAN FINANCIAL INFRASTRUCTURE IN THE FACE OF NEW CHALLENGES 54, (Franklin Allen, et al. eds., 2019).

[5] Guy Chazan, *Germany calls for global payments system free of US*, FINANCIAL TIMES, August 21, 2018. August 21, 2018.; Yves Mersch, *Strengthening the European financial industry amid disruptive global challenges*, SPEECH BY YVES MERSCH, MEMBER OF THE EXECUTIVE BOARD OF THE ECB, AT THE EUROPEAN INSTITUTE OF FINANCIAL REGULATION (EIFR), PARIS, 3 SEPTEMBER 2018 (September 3, 2018).; JP Koning, *Monetary Exclusion*, AMERICAN INSTITUTE FOR ECONOMIC RESEARCH (July 26, 2018).
For the first practical steps taken at the EU level, *see* European Union External Action - European External Action Service, Implementation of the Joint Comprehensive Plan of Action: Joint Ministerial Statement (September 24, 2018).; *See also* Europe, Iran, and Economic Sovereignty: A New Banking Architecture in Response to US Sanctions. (2018).

[6] Manson, FINANCIAL TIMES, Octover 9, 2018.

(CBDCs),[7] could also be viewed as a mechanism that could - in the long run - lead to decentralization in a multipolar international monetary and financial system.[8] However, it is unlikely that a system, which is based on fiat money, issued and controlled by states, could stand tall against the pressures exerted by one or more groups of hegemonic governments.

These developments have also highlighted the need for a truly decentralized uncensorable FMI on which one or a group of coordinated actors could not exert arbitrary influence. Such a value proposition requires a settlement asset that is denationalized, decentralized (peer-to-peer), divisible, digital, and globally transferable, and that provides certain levels of anonymity to its users. Bitcoin, which is built upon an open-source protocol, a distributed tamper-resistant timestamped globally synchronized ledger, and embeds a native digital asset is an obvious candidate to play such a role, despite its shortcomings in terms of price volatility.[9] In spite of its currently predominant use as a speculative asset, Bitcoin and its underlying technology can be viewed as a new model for a parallel decentralized FMI (dFMI) for clearing and settling obligations in its unanchored native settlement asset (i.e., bitcoin). In addition to clearing and settling its native asset, Bitcoin can be used to transfer the title to tangible or intangible assets on top of the Bitcoin blockchain. This is made possible because Bitcoin's scripting language allows embedding metadata in bitcoin transactions. For example, colored coins allow for recording the creation, ownership, transfer, and tracking of extrinsic digital and physical assets other than bitcoin.[10]

Whether a parallel dFMI relying on a settlement asset other than fiat currencies can alleviate the issues of financial exclusion and payment censorship remains to be seen. In particular, because the reason that the US possesses disproportionate influence over payment infrastructures is not entirely due to the reserve currency status of the USD, which is predominantly used in international FMIs, but also it is because the US has a large and attractive economy the benefits of which are hard to forgo for market participants in the face of a threat of being cut out of the US markets.[11] But it seems that decentralized financial technologies (FinTech), in particular, their structural architecture, which is built upon decentralized or distributed,[12]

[7] For the notion, potential design features, and legal issues concerning CBDCs, *see* Hossein Nabilou, *Testing the Waters of the Rubicon: The European Central Bank and Central Bank Digital Currencies*, JOURNAL OF BANKING REGULATION (2019).; Hossein Nabilou & André Prüm, *Central Banks and Regulation of Cryptocurrencies*, REVIEW OF BANKING & FINANCIAL LAW (FORTHCOMING) (2019).

[8] Mark Carney, The Growing Challenges for Monetary Policy in the current International Monetary and Financial System (Bnak of England 23 August 2019).

[9] For potential shortcomings of Bitcoin businesses to be recognized as a payment institution in terms of compliance with the existing regulations, *see* Hossein Nabilou, *The dark side of licensing cryptocurrency exchanges as payment institutions*, 13 LAW AND FINANCIAL MARKETS REVIEW (2019).

[10] ANDREAS M. ANTONOPOULOS, MASTERING BITCOIN: PROGRAMMING THE OPEN BLOCKCHIAN 278 (O'Reilly Media, Inc., . 2017).

[11] *See* Koning, AMERICAN INSTITUTE FOR ECONOMIC RESEARCH, (July 26, 2018).

[12] For the distinction between decentralized and distributed systems in the context of Bitcoin, *see* William J. Luther & Sean Stein Smith, *Is Bitcoin a Decentralized Payment Mechanism?*, SSRN WORKING PAPER SERIES (2020). Under his definition of decentralization, Bitcoin is best described as a distributed system. This paper uses the terms *decentralized* and *distributed* interchangeably.

consensus-based and censorship-resistant mechanisms without relying on centralized third parties can help shield such infrastructures from undue political influence.

A reflection on the history of Bitcoin shows that censorship-resistance considerations was central to its creation.[13] Although censorship resistance in Bitcoin has been achieved through a combination of technological innovations hardcoded in the Bitcoin network, preserving and maintaining such properties ultimately depend on the participants' consensus (i.e., judgment and discretion) in the Bitcoin network. In other words, preserving and enhancing or otherwise dispensing and undermining the censorship-resistant property of Bitcoin relies on its governance, which encompasses the processes and procedures for effecting changes in the rules governing the Bitcoin network.[14]

This study has been motivated by the recent developments concerning the censorship of the international FMI institutions, which have highlighted the vulnerabilities in the governance of the conventional FMIs. It will explore the governance of Bitcoin in the shadow of its censorship-resistant property and will discuss whether its governance mechanisms have been successful in addressing Bitcoin governance crises. This paper proceeds as follows. *Firstly*, it briefly discusses the objectives of Bitcoin governance by highlighting the differences in the objectives pursued in Bitcoin governance from those pursued in other governance schemes. In so doing, it also briefly compares Bitcoin governance with the conventional governance in constitutional systems, corporate governance and internet governance and highlights the idiosyncrasies in Bitcoin governance objectives. *Secondly*, after briefly sketching the built-in governance mechanisms in Bitcoin, the paper proceeds to identify the potential market failures in its governance. Having discussed the potential market failures and the potential market-driven and decentralized forms of governance mechanisms that would remedy such governance weaknesses in Bitcoin, the paper *finally* concludes that at the time of writing, there has been no serious market failure that could necessitate third-parity intervention in the governance framework of Bitcoin as it has managed to resolve some of its most pressing concerns with relative success in the last decade.

Bitcoin governance: What is it and what is it for?

Bitcoin is a distributed peer-to-peer (P2P) system that brings together a decentralized P2P network (the Bitcoin Protocol), a public transaction ledger (the blockchain), a set of consensus rules for independent transaction validation and native asset issuance, and a mechanism for reaching global consensus on the

[13] Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system (2008).
[14] Such changes may include changes to the size of each block, the number of block rewards, and even smaller changes to unlock or enable new features in Bitcoin such as SegWit activation.

valid chain in a decentralized manner, i.e., the Proof-of-Work (PoW) Algorithm.[15] The popularity of bitcoin as a medium of exchange and a unit of account in the Bitcoin network overshadows its rather complex, innovative and transformative aspects, i.e., establishing the first dFMI. It is important to highlight that the Bitcoin network functions as the infrastructure, while bitcoin (i.e., the token) functions as a medium of exchange in the Bitcoin network. Following the convention in the computer science literature, throughout this paper uppercase-B Bitcoin refers to the network and lowercase-b bitcoin refers to the unit of account. Needless to say, this paper concerns with the governance of Bitcoin as a network or infrastructure.

Governance is a system that shapes coordination between various participants.[16] As such, it refers to the processes that enable an organization to set its objectives, identify the means of achieving them, and monitor the performance of the organization against those objectives.[17] In the traditional corporate law, the main governance objective is to design incentive mechanisms that optimally allocate ownership rights, ownership structures, and define *control*, while aligning the interests of owners (principals) and managers (agents).[18] In the FMI context, governance is defined as "the set of relationships between an FMI's owners, board of directors (or equivalent), management, and other relevant parties, including participants, authorities, and other stakeholders (such as participants' customers, other interdependent FMIs, and the broader market)."[19]

From the above-mentioned definitions of governance, one can surmise that issues relating to *control* takes the center stage in defining governance. The questions of control in corporations often arise when there is a need for change in ownership, business model, structure of the firm, or its long-term strategic goals. In this sense, governance should be separate from the law or regulations applicable to the governance of a firm in that it is *internal* to the firm, whereas the law and regulations are external to the firm or organization.[20]

[15] Nakamoto. 2008.; ANDREAS M. ANTONOPOULOS, MASTERING BITCOIN: PROGRAMMING THE OPEN BLOCKCHAIN 2 (O'Reilly Media, Inc., . 2017).

[16] Philipp Hacker, *Corporate Governance for Complex Cryptocurrencies? A Framework for Stability and Decision Making in Blockchain-Based Organizations*, FORTHCOMING IN: REGULATING BLOCKCHAIN. TECHNO-SOCIAL AND LEGAL CHALLENGES, EDITED BY PHILIPP HACKER, IOANNIS LIANOS, GEORGIOS DIMITROPOULOS, AND STEFAN EICH, OXFORD UNIVERSITY PRESS, 2019, 10 (2018).

[17] Committee on Payment and Settlement Systems & Technical Committee of the International Organization of Securities Commissions, Principles for Financial Market Infrastructures 26 (Bank for International Settlements and International Organization of Securities Commissions April 2012).

[18] Eugene F. Fama & Michael C. Jensen, *Separation of Ownership and Control*, 26 THE JOURNAL OF LAW AND ECONOMICS (1983).; Brian L. Connelly, et al., *Ownership as a Form of Corporate Governance*, 47 JOURNAL OF MANAGEMENT STUDIES (2010).; Michael C. Jensen & William H. Meckling, *Theory of the firm: Managerial behavior, agency costs and ownership structure*, 3 JOURNAL OF FINANCIAL ECONOMICS (1976).; In other words, governance refers to the dynamics of power and influence that shapes decision making within a firm and delineates the rights and responsibilities of various stakeholders towards the firm. *See* Ruth V. Aguilera & Gregory Jackson, *Comparative and International Corporate Governance*, 4 THE ACADEMY OF MANAGEMENT ANNALS, 489-490 (2010).; Ying-Ying Hsieh, et al., *The internal and external governance of blockchain-based organizations: Evidence from cryptocurrencies*, *in* BITCOIN AND BEYOND: CRYPTOCURRENCIES, BLOCKCHAINS, AND GLOBAL GOVERNANCE 48, (Malcolm Campbell-Verduyn ed. 2018).

[19] Systems & Commissions, 26. April 2012.

[20] This important point seems to be overlooked in many discussions regarding Bitcoin governance, *See* Vlad Zamfir, *Against Szabo's Law, For A New Crypto Legal System*, MEDIUM (Janauray 26, 2019).

In the past decade, several critical developments in the Bitcoin ecosystem have thrown governance issues into the spotlight. As such crises could not be resolved simply by relying on the Nakamoto consensus, they highlighted the central role of human discretion in the governance of Bitcoin and other cryptocurrencies.[21] Those incidents were as follows:

1. Integer overflow incident (2010) involving an integer overflow bug that created 184,467 *billion* bitcoins: This bug was discovered on October 15, 2010 at the block 74,638. Once this bug was reported in the Bitcoin Forum, within 3 hours, Satoshi published a new Bitcoin client and rewound the hyper inflated chain. Nearly two hours later Satoshi released version 0.3.1. of the Bitcoin client where the hacked coins were erased. It took 19 hours for the good chain to prevail and become the dominant chain.
2. An erroneous upgrade to Bitcoin protocol and its rollback by coordination between developers and miners:[22] On March 11, 2013, there was an erroneous upgrade to Bitcoin protocol that led to two sets of miners mining legacy protocol and the updated one separately. A chain-split of at least 24 blocks occurred with the new chain having a maximum lead of 13 blocks. Two separate chains were mined for several hours and there has been a successful double-spend. This incident caused bitcoin price to sink by one third. However, the fork was rolled back by coordination between developers and miners who decided to *ignore the longest chain*, an apparent violation of the Nakamoto consensus. This resulted in some transactions being voided,[23] and raised concerns not only about settlement finality, but also about Bitcoin governance and who decides about issues concerning upgrades, and how such critical issues should be managed in the future.
3. The epitome of the governance crisis in Bitcoin was the debate about Bitcoin scaling that reached its zenith in 2017 and led to extremely polarizing controversies in the Bitcoin community.[24] Two main camps emerged on this dividing issue; one supporting vertical scaling solutions or second-layer solutions,[25] the other camp supporting horizontal scaling solutions or increasing the block size.[26] Ultimately, the dispute was settled by hard-forking. This crisis raised questions about

[21] Aaron van Wirdum, *A Primer on Bitcoin Governance, or Why Developers Aren't in Charge of the Protocol*, BITCOIN MAGAZINE Sept. 7, 2016.

[22] BitMEX Research, *A complete history of Bitcoin's consensus forks*, (28 December 2017).; Cryptocurrencies: looking beyond the hype. (2018).

[23] Research, (28 December 2017).; Settlements, 102-103. 2018. *See also* macbook-air, *A successful DOUBLE SPENT US$10000 against OKPAY this morning*, BITCOIN FORUM (March 12, 2013). Available at: http://archive.is/64Rkj

[24] Bitcoin itself can be viewed as an invention that emerged to overcome social scalability problem in the first place. Although the discussion of this paper is limited to technological scalability, the problem of social scalability stands at the core of the scalability issues in Bitcoin. Indeed, the perceived inefficiencies in the PoW can be understood in the balance struck between social scalability and computational scalability. In the Bitcoin Blockchain the latter is sacrificed to improve the former. For more details, *see* Nick Szabo, *Money, blockchains, and social scalability*, UNENUMERATED (February 09, 2017).

[25] ANTONOPOULOS, Mastering Bitcoin: Programming the Open Blockchian 300-321. 2017.

[26] Joseph Poon & Thaddeus Dryja, *The bitcoin lightning network: Scalable off-chain instant payments*, (2016).; Aaron van Wirdum, *The History of Lightning: From Brainstorm to Beta*, BITCOIN MAGAZINE (4 April 2018).; Tom Elvis Jedusor, *Mimblewimble*, (19

transparency, power asymmetry, censorship, and more importantly the very nature and purpose of Bitcoin.

4. The discovery of an inflation bug in the Bitcoin protocol allowing for potential double-spends in September 2018: The manner of handling this bug[27] raised questions about transparency in Bitcoin governance, because the discovery of the bug was publicly disclosed only after the inflation bug was discovered by other non-core developers and after the news made it to the public.

It is asserted that the Bitcoin blockchain by design is a technology that embeds governance and makes the rules-based economic order possible.[28] In the same vein, it is argued that within public blockchains in general, cryptocurrencies enable the creation and execution of rule-based systems that harbinger new institutional forms of economic governance, amounting to a new form of rule-system for economic coordination besides firms, markets, clubs, commons and governments.[29] However, as the above-mentioned instances of Bitcoin crises demonstrate, on-chain governance mechanisms have proved to be insufficient for its long-term viability. In addition to a need for constant intervention for protocol maintenance and improvements by developers, human intervention in the Bitcoin's open or permissionless blockchain has proved that Bitcoin governance is not immune to human discretion[30] and that the code-as-law narrative, put forward by the early proponents of such innovations, is far from accurate. In addition, these developments have put forward a serious governance question about who controls the changes to the Bitcoin network and specifically to the protocol.[31] The current literature on Bitcoin governance is far from thorough in providing answers to such questions.[32] More importantly, as Bitcoin does not entirely rely on on-chain mechanisms to resolve governance issues, the censorship-resistant property of Bitcoin ultimately relies on its off-chain governance that eventually depend on the power dynamics and interactions of the participants in the Bitcoin network. A broken governance framework can easily become prone to censorship due to centralization.

July 2016).; Aaron van Wirdum, *Mimblewimble: How a Stripped-Down Version of Bitcoin Could Improve Privacy, Fungibility and Scalability All at Once*, BITCOIN MAGAZINE (12 August 2016).

[27] *See* BitcoinCore (September 20, 2018), CVE-2018-17144 Full Disclosure (Notice), available at: https://bitcoincore.org/en/2018/09/20/notice/

[28] Sinclair Davidson, et al., *Disrupting governance: The new institutional economics of distributed ledger technology*, SSRN WORKING PAPER SERIES (2016).

[29] Sinclair Davidson, et al., *Economics of blockchain*, SSRN WORKING PAPER SERIES, 6-7 (2016).

[30] Research, (28 December 2017).; Cryptocurrencies: looking beyond the hype. (June 2018).; Aaron van Wirdum, *The Good, the Bad and the Ugly Details of One of Bitcoin's Nastiest Bugs Yet*, BITCOIN MAGAZINE (September 21, 2018).

[31] Although governance in dFMIs, can be in the form of on-chain governance, off-chain governance or a combination of both, most projects opt for a hybrid form of governance. Bitcoin has already embedded certain technical features, such as monetary policy, in its protocol. However, this does not mean that such rules are not subject to change. Therefore, Bitcoin governance, within the meaning of the governance adopted in this paper, remains to be off-chain.

[32] van Wirdum, BITCOIN MAGAZINE. Sept. 7, 2016.; Matthew A. Zook & Joe Blankenship, *New spaces of disruption? The failures of Bitcoin and the rhetorical power of algorithmic governance*, 96 GEOFORUM (2018).; MICHÈLE FINCK, BLOCKCHAIN REGULATION AND GOVERNANCE IN EUROPE (Cambridge University Press. 2019).; Philipp Paech, *The Governance of Blockchain Financial Networks*, 80 THE MODERN LAW REVIEW (2017).; Primavera De Filippi & Benjamin Loveluck, *The invisible politics of Bitcoin: governance crisis of a decentralised infrastructure*, 5 INTERNET POLICY REVIEW (2016).; Davidson, et al., SSRN WORKING PAPER SERIES, (2016).; Hacker, FORTHCOMING IN: REGULATING BLOCKCHAIN. TECHNO-SOCIAL AND LEGAL CHALLENGES, EDITED BY PHILIPP HACKER, IOANNIS LIANOS, GEORGIOS DIMITROPOULOS, AND STEFAN EICH, OXFORD UNIVERSITY PRESS, 2019, (2018).

This is why the importance of governance framework cannot be overstated. Having said so, the next section studies the analogies made between Bitcoin governance, constitutional arrangements, corporate governance, and internet governance, and will show why there is a need for an idiosyncratic governance model for Bitcoin.

Bitcoin governance vs. constitutional, corporate, and internet governance

In the relatively nascent literature on Bitcoin governance, several analogies have been made between Bitcoin governance and the governance in other disciplines spanning from constitutional law (i.e., analogies to separation of powers and checks and balances systems),[33] and corporate governance to internet governance. One such analogy draws parallels between Bitcoin nodes and the executive branch, the miners and the judiciary, the developers and the Senate, and finally the business and infrastructure community and the House of Representatives. In this analogy, the users, who may also be node operators, often use the businesses to interact with the network. However, the analogy to the constitutional checks and balances system remains misleading at best. In the Bitcoin ecosystem, there is neither a clear separation of powers or roles, nor even a clear division of labor. For example, a Bitcoin user can be a developer, may run a fully validating node and at the same time can be a miner or can have other Bitcoin related businesses. The same applies to other participants in the Bitcoin governance. As constitutional checks and balances system is heavily built on the idea of separation of powers, in the absence of such a separation, checks and balances system would at best be dysfunctional and at worst redundant. The second problem with such analogies is that there is no real representation or agency relationships between the user community and developers, miners, or node operators. For example, when a developer writes a piece of code, or otherwise contributes to the protocol, one could hardly imagine that she is acting on behalf of or as an agent to users. Therefore, such analogies fail to convey any meaningful message about Bitcoin governance.

Parallels have also been drawn between Bitcoin governance and corporate governance. This parallelism is either implicit or explicit.[34] Some studies have explicitly advocated the application of corporate governance standards to the governance in cryptocurrency ecosystems.[35] In contrast, other studies do not explicitly refer to corporate governance, however, their treatment of Bitcoin governance, conclusions and policy

[33] Buck Perley, *Crypto-Governance and the Dangers of Faction: Lessons from the 18th Century for designing a decentralized future*, MEDIUM (October 27, 2017).; TwoBitIdiot, *Bitcoin's Constitutional Crisis & Why I Support the UASF*, see id. at (June 21, 2017).; Fred Ehrsam, *Blockchain Governance: Programming Our Future*, see id. at (November 27, 2017).

[34] For an example of explicit application of the corporate governance principles to cryptocurrencies, *see* Hacker, FORTHCOMING IN: REGULATING BLOCKCHAIN. TECHNO-SOCIAL AND LEGAL CHALLENGES, EDITED BY PHILIPP HACKER, IOANNIS LIANOS, GEORGIOS DIMITROPOULOS, AND STEFAN EICH, OXFORD UNIVERSITY PRESS, 2019, (2018).

[35] Id. at.

recommendations seem to be rooted in the corporate governance literature.[36] Further studies have highlighted the role of Bitcoin as FMI and whether Bitcoin is compliant with the current legal and regulatory and governance requirements applicable to conventional FMIs.[37] This latter approach should also be viewed in the shadow of the Bitcoin governance as corporate governance thesis, in which there is an implicit or explicit assumption of agency problems or trust relationships, and the assumption that the Bitcoin blockchain and its function is a (semi-)public function as market infrastructure.

However, a quick overview of the corporate governance literature would clearly demonstrate that the drawing parallels between corporate governance and Bitcoin governance is misleading. Since the seminal work of Berle and Means entitled *the modern corporation and private property*,[38] it is believed that the separation of ownership and control in large corporations results in the managerial autonomy, because diffuse share ownership would prevent shareholders from effectively monitoring the managers. The entire corporate governance literature has been developed on this simple, but powerful insight. This is why shareholders select board members to monitor managerial activities. However, the main difference between such a governance scheme built on the premise of agency relationship and information asymmetry, which could give rise to opportunistic behavior, and Bitcoin governance is that in the Bitcoin ecosystem, there does not seem to be any meaningful separation between ownership and control and hence no agency relationship. Furthermore, Bitcoin blockchain transparency minimizes the information asymmetry between various Bitcoin stakeholders in a way that most of the venues for opportunistic behavior are practically closed. In addition, Bitcoin has successfully decreased the role of intermediaries and gave birth to a trust-minimized ecosystem. Such an achievement is what sets Bitcoin governance apart from conventional corporate governance.

One of the major implications of this analogical reasoning based on the seeming resemblance between Bitcoin governance and corporate governance has been the suggestion that certain fiduciary duties should be imposed on Bitcoin coders or developers,[39] a proposal that can have a significant impact on Bitcoin governance. However, it seems that such a proposal is based on the assumption that there exists an agency

[36] Angela Walch, *In Code(rs) We Trust: Software Developers as Fiduciaries in Public Blockchains*, *in* REGULATING BLOCKCHAIN: TECHNO-SOCIAL AND LEGAL CHALLENGES (Philipp Hacker, et al. eds., Forthcoming 2019).;

[37] Angela Walch, *The bitcoin blockchain as financial market infrastructure: A consideration of operational risk*, 18 N.Y.U. JOURNAL OF LEGISLATION & PUBLIC POLICY (2015).

[38] ADOLF A. BERLE JR. & GARDINER C. MEANS, THE MODERN CORPORATION AND PRIVATE PROPERTY (The Macmillan Company. 1933).

[39] Walch, In Code(rs) We Trust: Software Developers as Fiduciaries in Public Blockchains. Forthcoming 2019;Hacker, FORTHCOMING IN: REGULATING BLOCKCHAIN. TECHNO-SOCIAL AND LEGAL CHALLENGES, EDITED BY PHILIPP HACKER, IOANNIS LIANOS, GEORGIOS DIMITROPOULOS, AND STEFAN EICH, OXFORD UNIVERSITY PRESS, 2019, (2018).; Jack M. Balkin, *Information fiduciaries and the first amendment*, 49 UC DAVIS LAW REVIEW (2016). For an different view, *see* Aaron van Wirdum, *A Primer on Bitcoin Governance, or Why Developers Aren't in Charge of the Protocol*, BITCOINMAGAZINE Sept. 7, 2016.; Jerry Brito & Peter van Valkenburgh, *Writing and publishing code alone cannot be a crime*, COINCENTER.ORG (Octover 29, 2018).; Raina Haque, et al., *Blockchain Development and Fiduciary Duty*, STANFORD JOURNAL OF BLOCKCHAIN LAW & POLICY (2019).

and trust relationship between Bitcoin developers and users and that developers can effect changes either by themselves or on behalf of other network participants. The problem with fiduciary duties in Bitcoin governance is that developers can only *propose* changes to the protocol, whereas the implementation of a Bitcoin Improvement Proposals (BIP) requires a consensus to be reached by other participants, especially Bitcoin users. In this respect, it is not clear how such fiduciary duties are to be designed if the developers do not have the authority or power to implement and impose that implementation on other network participants. In this sense, the developers are said to be one of the least powerful of major Bitcoin stakeholders.[40] Furthermore, the imposition of fiduciary duties should be commensurate to the extent of the agent's authority to exert a meaningful influence on behalf of the principal that could bind her. If no such authority exists, imposition of the duty would go far beyond the long-established legal principle that no one should be held liable for that which is beyond her control. Moreover, if such duties are to be imposed, it is not very obvious to whom those duties should be owed. Ideally, the majority of Bitcoin users should be anonymous, which makes the identification of the persons to whom the duty is owed an arduous task.

A third approach to Bitcoin governance draws parallels between Bitcoin governance and internet governance.[41] The main focus of this approach is to discern whether it is appropriate to follow the internet governance model and apply similar mechanisms to Bitcoin governance. The literature on internet governance has always been a battle ground of two opposing forces: government-centric multilateral governance model as opposed to the private-sector-led multi-stakeholder governance model (or distributed governance model).[42] Indeed, the Bitcoin governance shares many common features with internet governance, in particular in that both governance models deal with the governance in relatively decentralized systems. This indeed makes internet governance the closest model that can be followed in studying Bitcoin governance.

Bitcoin governance is similar to internet governance in that it is concerned with the governance of a distributed system. The emphasis on decentralization in earlier days in internet governance is what makes its especially similar to Bitcoin governance. In this sense, Bitcoin governance has already benefited from the governance mechanisms in internet governance. For example, the process that is used to make updates to Bitcoin protocol follows the Request for Comments (RFC) format created in 1969 for the ARPANET.[43]

[40] Jeffery Atik & George Gerro, *Hard forks on the Bitcoin blockchain: reversible exit, continuing voice*, see id. at, 7 (2018).

[41] De Filippi & Loveluck, INTERNET POLICY REVIEW, (2016).; PRIMAVERA DE FILIPPI & AARON WRIGHT, BLOCKCHAIN AND THE LAW: THE RULE OF CODE (Harvard University Press. 2018).

[42] Who Runs the Internet? The Global Multi-stakeholder Model of Internet Governance. (2017).

[43] In the same vein, the analogies to internet can be useful in analyzing certain aspects of Bitcoin governance, such as the analogy made of Bitcoin layers to the layers of the internet. *See* DE FILIPPI & WRIGHT. 2018.; Lombrozo, BIP 123. https://github.com/bitcoin/bips/blob/master/bip-0123.mediawiki; Aaron van Wirdum, *Why Some Changes to Bitcoin Require Consensus: Bitcoin's 4 Layers*, BITCOIN MAGAZINE (February 26, 2016). By slicing the Bitcoin network to many layers similar to that of the internet, this provides an analytical view for the governance of Bitcoin that is essential to both understanding the Bitcoin network and potential regulatory measures for minimizing the risks stemming from the network. For example, an indirect regulatory

However, as it is well known, the promise of decentralization in the internet largely faded away with the passage of time, and other issues such as access to internet, net neutrality, data protection, and the role of governments in regulating the internet have taken the center stage in its governance.

In addition to the decentralized aspect of the internet and Bitcoin, Bitcoin governance diverges from the internet governance in some other key respects. For example, some of the market failures in the governance of the internet have been *designed out* with various innovations in Bitcoin. In particular, embedding the digitally scarce native asset is one such innovation that motivates the stakeholders, including miners, users and developers to play an active role in securing the Bitcoin network and contributing to the maintenance of the network. In addition, embedding the PoW algorithm in the Bitcoin network shields it against various attacks and spamming activities and indirectly encourages cooperative behavior on the part of the participants in the ecosystem. Furthermore, the direct compensation of miners through block rewards ensures that the miners have enough at stake not only to behave cooperatively to secure the system, but also to contribute to the governance of Bitcoin.[44] Embedding such incentive-compatible mechanisms in the design of the Bitcoin network mitigates the concerns about positive externalities, or potential tragedy of the commons expressed where governance is seen as a public good or commons. In other words, such built-in incentive mechanisms that promote the active role for participants in Bitcoin governance differentiate the governance model of Bitcoin from that of the internet.

To summarize, in spite of the similarities to constitutional, corporate and internet governance, Bitcoin possesses features that differentiate it from all those three governance models. Therefore, as much as enlightening such analogies are, they remain misleading to varying degrees. By highlighting the idiosyncrasies of Bitcoin and its governance mechanisms, the next section argues for an idiosyncratic governance model for Bitcoin that maximizes its unique value proposition and hence benefits the entire population of its constituents and stakeholders.

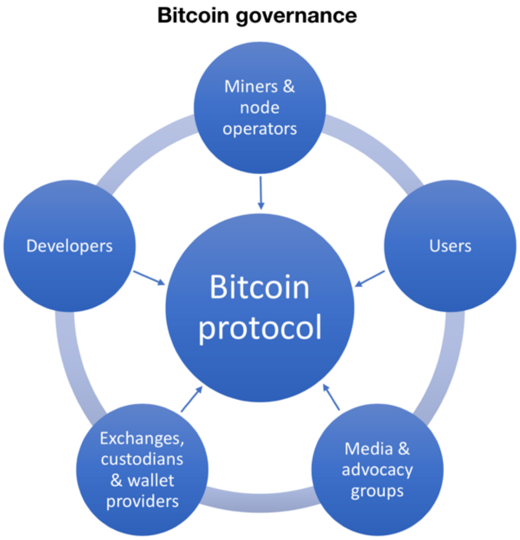
Towards an idiosyncratic governance model for Bitcoin

Economic organizations have various actors, stakeholders or constituents, each with oft-divergent interests. Aside from shareholder value maximization, one of the main objectives of corporate governance is to create an institutional or otherwise streamlined framework for resolution of disputes stemming from those

approach to regulating Bitcoin can be advocated based on such an analogy. For more details on the indirect regulation of Bitcoin, *See* Hossein Nabilou, *How to Regulate Bitcoin? Decentralized Regulation for a Decentralized Cryptocurrency*, INTERNATIONAL JOURNAL OF LAW AND INFORMATION TECHNOLOGY (2019).

[44] The heavy capital investment by miners in the mining infrastructure highly increases their incentives to act cooperatively rather than attack the network. See: Hasu, *No, Concentration Among Miners Isn't Going to Break Bitcoin*, COINDESK (February 20, 2020). In addition, the fact that selling freshly minted bitcoins (block rewards) by miners require 100 confirmations, also provide another incentive-compatible mechanism for securing the system against the miners misbehavior and spending the coins on orphaned blocks.

conflicting interests inside an ecosystem, organization or network. In conventional corporate governance, there are internal actors including owners (shareholders), managers, and the board members, and external actors or constituents such as customers, the media, government, and the broader community.[45] In Bitcoin governance, it is equally important to discern who the main stakeholders of Bitcoin network are and what Bitcoin can uniquely offer to maximize their payoffs. In the Bitcoin ecosystem, various actors such as mining pools, node operators, users, developers, exchanges, custodians and wallet providers, and eventually the media and advocacy groups have their say and they ultimately decide over critical governance issues either by reaching a consensus or by forking. It appears that it is the very unique value proposition of Bitcoin as censorship-resistant store of value and value transfer infrastructure that is likely to maximize value for all Bitcoin stakeholders. This paper analyzes Bitcoin governance in light of this unique value proposition.



The governance framework of an organization or network is a function of governance objectives. Different organizational structures require different sets of governance mechanisms based on their idiosyncratic features. For example, bank governance is often deemed different from the governance of other financial and non-financial institutions due to the idiosyncrasies in banking.[46] Similarly, the governance of dFMIs should be different from the governance models of conventional centralized businesses and organizational structures, mainly due to their decentralization.[47] Given the importance of certain firms such as those

[45] Hsieh, et al., 51. 2018.
[46] John Armour, *Bank Governance*, *in* THE OXFORD HANDBOOK OF CORPORATE LAW AND GOVERNANCE (Jeffrey Gordon & Wolf-George Ringe eds., 2015).; Klaus J. Hopt, *Corporate Governance of Banks after the Financial Crisis*, *in* FINANCIAL REGULATION AND SUPERVISION: A POST-CRISIS ANALYSIS (Eddy Wymeersch, et al. eds., 2012).; Jakob de Haan & Razvan Vlahu, *Coroporate governance of banks: A survery*, 30 JOURNAL OF ECONOMIC SURVEYS (2016).; Kern Alexander, *Bank Corporate Governance: Law and Regulation*, *in* PRINCIPLES OF BANKING REGULATION 128, (2019).
[47] For an attempt to apply the traditional corporate governance to the cryptocurrencies and blockchain-based technologies, *see* Hacker, FORTHCOMING IN: REGULATING BLOCKCHAIN. TECHNO-SOCIAL AND LEGAL CHALLENGES, EDITED BY PHILIPP HACKER, IOANNIS LIANOS, GEORGIOS DIMITROPOULOS, AND STEFAN EICH, OXFORD UNIVERSITY PRESS, 2019, (2018).

providing conventional FMI services, a formal governance framework is mandatory for them. For example, Principle 2 of the Principles for Financial Market Infrastructures states that "[a]n FMI should have governance arrangements that are clear and transparent, promote the safety and efficiency of the FMI, and support the stability of the broader financial system, other relevant public interest considerations, and the objectives of relevant stakeholders." But the question is whether a formal governance framework for FMIs are also suitable for Bitcoin as a dFMI.

In this regard, it is important to make a distinction between permissioned and permissionless blockchains. The governance of a permissioned network, though has its unique challenges, can be straightforward and could be similar to the governance of conventional FMIs, mainly because their value proposition is very similar to that of conventional businesses, and more impotently because there is no need for minimizing trust by relying on decentralized technologies. However, the governance in permissionless decentralized networks can be extremely challenging mainly due to their unique value proposition. In addition, thanks to their decentralization, no entity could be identified to have a meaningful control on the network so that it would be granted specific powers and burdened with equivalent responsibilities in the governance of a distributed network. In this sense, unlike the mainstream perception on Bitcoin governance, this paper argues that Bitcoin's current governance framework is suited for the purpose that it is created to serve, i.e., establishing a censorship-resistant store of value and medium of exchange. In this perspective, the importance of Bitcoin manifests itself in providing optionality for those who do not have access to the conventional financial markets or whose access has been revoked for legitimate or illegitimate reasons.

Censorship resistance as the unique value proposition of Bitcoin is very much reflected in the Bitcoin whitepaper[48] as well as Satoshi Nakamoto's communications with early bitcoin adopters. Given the fate of Bitcoin's predecessors such as Digicash and American Liberty Dollar (ALD),[49] whose centralization was their undoing, the creator or creators of Bitcoin had the understanding that permissionless innovative payment systems have to be decentralized, otherwise those innovations will face the same fate as Bitcoin's ancestors. This is also clear from the chronology of the technological breakthroughs that led to the birth of Bitcoin.[50]

More importantly, censorship-resistant property of Bitcoin is reflected in the design of the Bitcoin network. The clearest manifestation of this property is in the trade-off between efficiency and censorship resistance. Rather than opt for fast and efficient payments, Bitcoin goes a long way to create extreme inefficiencies by

[48] Nakamoto. 2008.

[49] Aaron van Wirdum, *The Genesis Files: How David Chaum's eCash Spawned a Cypherpunk Dream*, BITCOIN MAGAZINE (24 April 2018).; Aaron van Wirdum, *The Genesis Files: If Bitcoin Had a First Draft, Wei Dai's B-Money Was It*, BITCOIN MAGAZINE (15 June 2018).; NICK BILTON, AMERICAN KINGPIN: THE EPIC HUNT FOR THE CRIMINAL MASTERMIND BEHIND THE SILK ROAD (Portfolio/Penguin. 2017).

[50] Arvind Narayanan & Jeremy Clark, *Bitcoin's academic pedigree*, 60 COMMUNICATIONS OF THE ACM (2017).

introducing a distributed ledger that should be maintained, updated and validated by all fully validating nodes, only to make sure that no single or a small group of participants violate the rules of the Bitcoin protocol, modify the ledger arbitrarily or censor other stakeholders from participating in the Bitcoin network. Such a tradeoff has been made because the unique value proposition of Bitcoin and blockchain technology is not to replicate the functions of centralized technologies in a faster or cheaper fashion. Indeed, as blockchain technology is one of the least efficient technologies for payments and data storage, and as virtually everything that can be done on blockchain can also be performed on a network that uses a client-server or master-slave architecture, the very use of blockchain technology creates inefficiencies that are absent in centralized systems.[51] As the client-server architecture uses a centralized coordination mechanism, it is much faster, cheaper and efficient than the systems relying on blockchain technology. However, blockchain technology can be suitable for use-cases such as creating an asset that would give its owner near-full control so that she could hold and transact with the asset using a secure and trust-minimized network.[52] In this perspective, the objective of becoming a fast and efficient global settlement layer is secondary and should yield when conflicted with the objective of censorship resistance.

It is indeed hard not to notice that the censorship-resistant property of Bitcoin drives the entire mechanism designs embedded in the Bitcoin network. Since Bitcoin is designed to operate outside the legal framework (i.e., alegality), it assumes an adversarial environment and prepares to defend itself against various attack vectors using a variety of ex-ante built-in mechanisms within the Bitcoin network rather than rely on the external legal system for ex-post remedies. To this end, the PoW security and consensus mechanism and various other incentive mechanisms are embedded to align the often varied and divergent interests of network participants and discourage uncooperative behavior that could result in attacks on the network.[53]

More importantly, it seems that pursuing censorship resistance is the single use-case that could bring the interests of users, investors, miners, and other actors in the ecosystem together. As other properties of Bitcoin could easily be replicated and conducted more efficiently within the traditional banking, financial and payment institutions, without its censorship-resistant property, Bitcoin would be redundant.[54] In this sense, if Bitcoin network were just a normal payment system without any censorship-resistant property, its native token would have had virtually no value at all. Therefore, compromising this core feature would render the entire network useless. As protecting censorship resistance would entail a decent amount of decentralization, to make the system resilient against the attacks aimed at the central point of failure,

[51] Szabo, UNENUMERATED, (February 09, 2017).; *See also* Gabriel Shapiro, *Tokenizing Corporate Capital Stock*, ZERO_LAW (October 28, 2018).
[52] Shapiro, ZERO_LAW, (October 28, 2018).
[53] De Filippi & Loveluck, INTERNET POLICY REVIEW, 5-6 (2016).
[54] For a similar view, *see* Gabriel Shapiro, *In Defense of Szabo's Law, For a (Mostly) Non-Legal Crypto System: A Lawyer's Response to Vlad Zamfir's "Against Szabo's Law, For A New Crypto Legal System"*, MEDIUM (January 26, 2019).

decentralization and censorship-resistant property go hand in hand and should be given equal weight in Bitcoin governance. Having said so, the ultimate goal of Bitcoin governance should be retaining its censorship-resistant property through boosting its decentralization.

Not only does decentralization makes Bitcoin censorship resistant, but also it ensures that Bitcoin's other important features, including bitcoin issuance or emission rate, cannot be easily manipulated by one or coordinated groups of related parties. If for example, bitcoin issuance would have been subject to change by a few players who could be able to coordinate easily, it would have been unlikely for Bitcoin to retain the value that it has retained so far. As my coauthor and I have argued elsewhere, partly due to decentralized architecture of Bitcoin, which begets potential endogenous information insensitivity, Bitcoin has a good shot at becoming a safe asset.[55] Gaining such a status ultimately depends on the governance of Bitcoin. Meaning that if the governance of Bitcoin makes it malleable to changes that can be easily incorporated into the Bitcoin protocol and allows for information asymmetries, Bitcoin would lose its potential to become endogenously information insensitive.

This paper is written under the assumption that for the foreseeable future, Bitcoin will continue its role as a niche medium of exchange and dFMI. Under this assumption, the greatest value proposition of Bitcoin would be its censorship-resistant property. Therefore, the efficiency and effectiveness of governance arrangements in Bitcoin should be tested against this important criterion. This acknowledgement will constitute a point of departure from the governance models in other types of organizations in that Bitcoin governance should lean towards protecting that censorship resistance even at the expense of creating inefficiencies for other perceived use-cases of Bitcoin. This assumption, in turn, means that the application of the stringent governance standards as applied to conventional organizations and FMIs to Bitcoin is misguided, not only because the application of the centralized governance models to decentralized models can be counterproductive, but also because for a censorship resistant network to remain so, it needs to be regulation-resistant too. In this sense, it appears that the external regulation by governments of the governance of the Bitcoin protocol, even if feasible, is likely to undermine Bitcoin's core value proposition.[56]

Having said so, the question on the failure or the shortcomings of Bitcoin governance should be analyzed in light of the objectives of Bitcoin governance. If Bitcoin governance is to maximize its value as an uncensorable dFMI, one could issue a different verdict on Bitcoin governance compared to the scenario in

[55] Hossein Nabilou & André Prüm, *Ignorance, debt and cryptocurrencies: The old and the new in the law and economics of concurrent currencies*, 5 JOURNAL OF FINANCIAL REGULATION (2019).; *See also* David Andolfatto, *Is Bitcoin a Safe Asset?*, MACROMANIA (March 27, 2016).

[56] Jameson Lopp, *Nobody Understands Bitcoin (And That's OK)*, COINDESK (March 11, 2017).; Jameson Lopp, Who Controls Bitcoin Core? (2018).

which the objective of Bitcoin governance is to maximize its value proposition as a payment mechanism to compete with established payment infrastructures such as Visa, Mastercard or other wholesale and retail payment infrastructures. In the latter case, one could concede that the governance of Bitcoin is broken, because it is unlikely for a decentralized organization to reach the level of efficiency that exists in the centralized systems due to the resources needed to overcome coordination problems in PoW-based decentralized systems. The rest of this paper will gauge the success or failure of Bitcoin governance in light of the fact that it is designed to be a censorship-resistant network.

Is Bitcoin governance broken?

Various governance crises in Bitcoin thus far have highlighted the role of governance and its importance in the Bitcoin ecosystem. One of these crises was the famous scaling crisis that led certain well-known figures in the cryptocurrency ecosystem to declare that Bitcoin governance is broken, and that Bitcoin project should be liquidated.[57] Despite those protestations and doomsday predictions, the Bitcoin network has continued to grow. However, the legitimate question remains to be whether there are deficiencies in Bitcoin governance that could warrant interventions by third parties, such as private sector stakeholders or governments. As the classic reason for third-party intervention in free markets is to establish a case for market failures, the starting point for the investigation is to fathom whether Bitcoin governance works or there are market failures that would necessitate remedial actions.

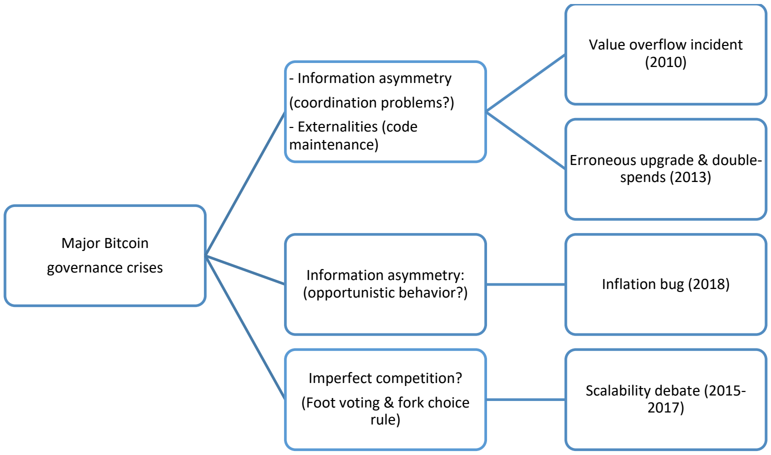
The theory of market failure suggests that markets fail due to externalities, imperfect competition, and imperfect information which gives rise to agency costs and coordination problems. The rest of this paper is to identify the potential market failures in Bitcoin governance by reference to its four major governance crises. These crises include the value overflow incident (2010), the erroneous upgrade and double spend in Bitcoin network (2013), the discovery of an inflation bug (2018), and the Bitcoin scalability debate (2015-2017).

Three such crises highlight the coordination problems arising from imperfect information (i.e., information asymmetry) in the maintenance of the code and dealing with the bugs. Such crises include the value overflow incident (2010), an erroneous upgrade and double spends (2013), and the inflation bug (2018). The first two of these incidents highlight the coordination problems in the maintenance of the code, whereas the latter put the spotlight on the potential opportunistic behavior that such information asymmetries would give rise to in the relationship between developers and users (and even miners). The third most important issue giving rise to market failures originate from the problems associated with imperfect competition. The

[57] Mike Hearn, *The resolution of the Bitcoin experiment*, MEDIUM (January 14, 2016).

fourth major Bitcoin crisis (i.e., scalability debate) highlights the role of competition and fork-choice rule and foot voting in resolving governance disputes in the Bitcoin ecosystem.

Information asymmetry could give rise to either coordination problems that might hinder effective governance and maintenance of Bitcoin or to opportunistic behavior, including adverse selection (such as the classic Lemons problem à l'Akerlof) that might hinder bitcoin adoption.[58] The latter also includes the adverse selection problem that may emerge if certain participants who have private information about the Bitcoin network, such as the knowledge of an existing bug, engage in selling bitcoins to less informed market participants. In this section, the potential opportunistic behavior is studied first and then the coordination problems will be investigated.


Major Bitcoin governance crises
- Information asymmetry (coordination problems?)
- Externalities (code maintenance)
Information asymmetry: (opportunistic behavior?)
Imperfect competition? (Foot voting & fork choice rule)
Value overflow incident (2010)
Erroneous upgrade & double-spends (2013)
Inflation bug (2018)
Scalability debate (2015-2017)

Agency costs and opportunistic behavior in Bitcoin governance

One of the main objectives of the conventional corporate law and governance is to remedy the agency costs in a corporation. In traditional corporate governance, information asymmetry between principals (shareholders) and agents (managers or officers) increases the agency costs, e.g., chances of engaging in opportunistic behavior such as looting[59] by managers. The early literature has recognized this problem by focusing on the separation of ownership and control and highlighting that the diffuse share ownership would eventually lead to managerial autonomy.[60] To mitigate the agency costs, the principals often use various mechanisms. Central to such mechanisms are the board of directors to monitor the performance of those

[58] George A. Akerlof, *The Market for "Lemons": Quality Uncertainty and the Market Mechanism*, 84 THE QUARTERLY JOURNAL OF ECONOMICS (1970).
[59] George A. Akerlof, et al., *Looting: The Economic Underworld of Bankruptcy for Profit*, 1993 BROOKINGS PAPERS ON ECONOMIC ACTIVITY (1993).
[60] BERLE JR. & MEANS. 1933.; John Armour & Jeffrey N Gordon, *The berle-means corporation in the 21st century*, (2009).; Brian R Cheffins, *The Rise and Fall (?) of the Bearle-Means Corporation*, 42 SEATTLE UNIVERSITY LAW REVEIW (2018).

who are in control (management).[61] Combined with various other mechanisms, conventional corporate governance has only been relatively successful in controlling managers and protecting shareholders, especially the minority shareholders.

The question is whether conventional corporate governance paradigm can be applied to Bitcoin. It appears that such a transplant remains dubious at best. *Firstly*, as the agency relationships (both in its legal and economic sense) do not exist between various participants in the Bitcoin network, the traditional mechanisms of corporate governance can hardly be applicable to Bitcoin governance. Agency relationships and the protections that are in place for both principals and agents, such as fiduciary duties and duties of care and loyalty,[62] are traditionally established where there is a relationship of trust between the parties to a relationship. However, central to the idea of Bitcoin is the creation of an alternative decentralized network that could compete with traditional economic organizations by proposing a different form of organizational governance.[63] In other words, the main idea of having Bitcoin in place is to use a trustless or trust-minimized socially scalable system for transacting or reaching consensus in an adversarial environment.[64] This applies in particular to those who run fully validating nodes and hence participate in the Bitcoin network directly and without the intermediary role of third parties. In this sense, Bitcoin has tried to remove the principal-agent problem by replacing humans with machines at the center of the network and moving the human discretion to the periphery.[65]

Secondly, since the Bitcoin network is highly transparent about its protocol, blockchain and methods of effecting a change in the protocol, it is hard to establish a clear-cut informational asymmetry between various participants in the network. Such a difficulty is exacerbated by the problem that various participants in the network can assume different roles. For example, a developer can simultaneously be a user and/or can be employed by a mining company or other Bitcoin businesses. Furthermore, she can operate a fully validating node or can run her own Bitcoin business. This muddies the traditional information asymmetry framework in conventional organizations where there is a relatively clear division of labor and separation of roles and powers that make it possible to define various conflict-of-interest rules for participants.

Finally, as my coauthor and I have argued elsewhere,[66] it appears that in the Bitcoin network, participants are symmetrically informed about all aspects of the Bitcoin network, or where there is a hidden aspect to

[61] However, the management has wielded considerable influence on the board members making such monitoring ineffective. The increasing emphasis on the use of independent board members has been a response to such unfettered influence.
[62] For more details on such duties, *see* Balkin, UC DAVIS LAW REVIEW, (2016).
[63] Davidson, et al., SSRN WORKING PAPER SERIES, (2016).; ARVIND NARAYANAN, et al., BITCOIN AND CRYPTOCURRENCY TECHNOLOGIES: A COMPREHENSIVE INTRODUCTION (Princeton University Press. 2016).
[64] Szabo, UNENUMERATED, (February 09, 2017).
[65] Hsieh, et al., 51. 2018.; Vitalik Buterin, *DAOs, DACs, DAs and More: An Incomplete Terminology Guide*, ETHEREUM BLOG (May 6, 2014).
[66] Nabilou & Prüm, JOURNAL OF FINANCIAL REGULATION, (2019).

the network (e.g., unknown unknowns), that information is hidden symmetrically. To be clear, this does not mean that there is complete information, but there is no *asymmetric* information. One might argue that there are influential figures - constituting a small fraction of participants in the Bitcoin network - whose decisions can affect the network disproportionately, may have some inside information that would create information asymmetry. Although information asymmetries might exist outside the technical environment of the Bitcoin network and in the allegedly closed circles of a few prominent participants such as miners, developers and investors,[67] there are mechanisms embedded in the Bitcoin network that can remedy such informational asymmetries. For example, the mere fact that the decisions made by so-called insiders cannot be imposed on other participants due to the possibility of forking and creating the version of Bitcoin that best serves the interests of the broader community of the users, makes such informational asymmetry largely inconsequential.[68]

Thus far, there is no publicly known evidence that any of the core developers or their close relatives, or those endowed with the private information about potential bugs in the Bitcoin protocol has engaged in opportunistic behavior and abused their special status by trading on such private information. Though the possibility of such a phenomenon cannot be entirely disregarded, the Bitcoin's checks and balances system thus far has largely dealt with this problem. Therefore, as of this writing, there seems to be no need for imposing potential liability rules on Bitcoin network participants above and beyond the existing general liability rules that would nevertheless be applicable to anyone irrespective of being active in the cryptocurrency ecosystem or in the real economy.

Coordination problems in Bitcoin governance

Information asymmetries may give rise to coordination problems among participants in the Bitcoin network, especially among Bitcoin developers, and may eliminate any incentives for their contribution to protocol improvements and maintenance, in particular in times of crises. *First*, the developers are dispersed and relatively decentralized, the coordination for fixing bugs and adding patches can take substantial amount of time. Due to this operational risk, the very existence of Bitcoin could be put at risk. *Second*, the information asymmetry can also be present where there is a major impending event that could fork Bitcoin. Such informational asymmetry can create standoffs of the sort that were seen in the debate about scaling

[67] Hsieh, et al. 2018.

[68] Even if there were consequential information asymmetries in the Bitcoin network, imposition of traditional conflict-of-interest rules, including fiduciary duties on developers could not be justified, mainly because the core development team is not an official designation that could grant rights to or impose responsibilities on the developers. In addition, having such designations as official Bitcoin developer, risks a level of centralization in Bitcoin network that a truly decentralized and censorship-resistant network cannot afford. Furthermore, as mentioned earlier, an additional problem with assigning liability to protocol developers is that they cannot force software changes and updates to other participants in the network. *See* Haque, et al., STANFORD JOURNAL OF BLOCKCHAIN LAW & POLICY, 11 (2019).

Bitcoin. *Third*, information asymmetry in distressed times can also give rise to runs on Bitcoin. If participants in the Bitcoin network would come to the conclusion that there is an epsilon of material information that is unevenly distributed in the network, each and every participant in the network is better off selling her holdings and only afterwards investigating into the nature of the piece of information hidden from her.

The Bitcoin network mitigates some of the coordination problems that exist among users, nodes, miners and other direct participants in the Bitcoin network. For example, the Nakamoto consensus is embedded in the Bitcoin network to make it Byzantine fault tolerant and resilient to sybil attacks, which in itself encourages coordination among network participants. However, the governance of the network, as it is indirect and is exercised at a different level (off-chain), remains to be subject to coordination problems, especially where there would be a need for maintenance, upgrade or otherwise making changes to the protocol, or when there is an urgent need to effectively respond to an attack on the network in a timely manner.

Various governance mechanisms attempt to reduce these coordination problems. For example, making upgrades to the Bitcoin protocol follows the tradition of open source software, i.e., the Request for comments format, created in 1969 for the ARPANET.[69] To further reduce coordination problems, various channels of communication such as BitcoinTalk Forum and Bitcoin Core GitHub repository are set up and relatively informal processes for upgrading Bitcoin's protocol, such BIP, which is a standardized process for proposing, testing and peer review of the new proposals for effecting changes to the protocol, are established.[70] The latter procedure is to ensure that innovation is not hindered while the improvements are implemented through consensus and collaboration.

In the early days of Bitcoin, there was no specific framework for improving Bitcoin protocol. Satoshi created the code and simply made improvements. He used to solicit feedback from the Cryptography Mailing List, and eventually decided to create BitcoinTalk Forum. At the time, Satoshi alone was in charge of effecting any changes or upgrades to Bitcoin. In 2011, Nakamoto left the Bitcoin project and handed it over to Gavin Andresen. As Gavin did not want to accept the responsibility alone, he, in his turn, enlisted four other developers,[71] who became known as Bitcoin Core developers (Core Devs). Bitcoin Core Devs have commit access to the Bitcoin Core Github repository and maintain the Bitcoin codebase. They are the

[69] De Filippi & Wright. 2018.

[70] In addition, there are several mechanisms that are hardcoded to the bitcoin network to reduce or eliminate coordination problems. For example, the programmability of Bitcoin allows several other mechanisms to be embedded into the transactions that could function as a signal by Bitcoin network participants such as miners in crucial decision-making processes (i.e., miner signaling). In this regard, BIP 9 has been introduced to function as a signaling mechanism that allows miners to indicate whether they are ready to implement a particular change. Some BIPs require the signaling of a super majority of miners to be activated. *See* Atik & Gerro, Stanford Journal of Blockchain Law & Policy, 4 (2018).

[71] These core maintainers include Pieter Wuille, Wladimir van der Laan, Gregory Maxwell, and Jeff Garzik.

only developers that have the ability to push live code to the Bitcoin Core client. Therefore, although hundreds of developers have contributed to the code, only a few of them has the commit access to the code base.[72]

Having only a few Core Devs with commit access to the code would give the impression of significant centralization. However, these Core Devs are not unconstrained and the changes to the code follows a process of rough consensus that determines what proposed changes to be merged in the protocol.[73] In deciding whether to merge a proposal or a patch, Bitcoin Core Devs will take the followings into account:

- Whether the patch is in line with the general principles of the project;
- Whether it meets the minimum standards required for inclusion; and
- Whether it aligns with the general consensus of contributors.[74]

Such a broad language grants a great deal of informal powers to Bitcoin Core Devs. In addition, it is even theoretically possible for maintainers to organize a coup to hijack the GitHub repository and censor the dissenting developers, or even highjack the brand name of Bitcoin (core). However, those powers are again constrained by at least two factors. *First*, the fact that Bitcoin developers are not a homogenous pool of developers, which makes it unlikely for them to highjack the project. *Second*, the fact that dissenting developers can always fork the code and shift their work to a different repository that is not controlled or otherwise influenced by Bitcoin Core maintainers.[75] Again, the threat of forks, especially hard forks, to which the developers have expressed aversion, presents a strong check over the potential abuse of powers by developers.

In closing this section, it is important to note that the difficulty in the coordination in Bitcoin should not always be viewed as a negative feature. A truly decentralized cryptocurrency network will inevitably face certain levels of disagreements at every level of coordination. Therefore, slow and laggard nature of decision-making and resolving disputes in a decentralized fashion should not be viewed as a problem for a decentralized network, unless where there is a critical issue that poses an existential threat to the network. A parallel can be made between the evolution of dispute resolution mechanisms in the common law countries through a network of distributed courts as opposed to contriving dispute resolution methods through codification across the board. In the former, although the emergence of standards can take decades or ages, giving the impression of inefficiency, the ultimate outcome often benefits from the dispersed knowledge and experience of the vast number of participants and only afterwards becomes the law, which

[72] SFOX, *Bitcoin Governance: What are BIPs and how do they work?*, MEDIUM (April 16, 2019).
[73] See Contributing to Bitcoin Core, GitHub, available at: https://github.com/bitcoin/bitcoin/blob/master/CONTRIBUTING.md
[74] SFOX, MEDIUM, (April 16, 2019).
[75] Id. at.; *See also* Lopp, Who Controls Bitcoin Core? 2018.; Lopp, COINDESK, (March 11, 2017).

is likely to last for decades or centuries. In contrast, codification either dictates the law or codifies the preferences of limited number of participants into the law, which may entail the creation of economic rents. It seems that the first approach, despite being slow, is more sustainable. By the same token, dispute resolution in Bitcoin would benefit from the slow process of deliberation that distills the decentralized knowledge dispersed among various economic participants to overcome the Hayekian knowledge problem.[76] This may eventually turn the threat of coordination problem to an opportunity for long-term sustainability of decision making and governance for the Bitcoin network. In this sense, it seems that there would be a last-mover advantage in Bitcoin governance, which seems particularly true where there is a need for tweaking the protocol layer rather than higher levels of the Bitcoin network.

The impact of competition on Bitcoin governance

Zooming on various competitive forces in Bitcoin, three forms of competition with Bitcoin Core can be identified: competition between chains, competition between independent implementations, and competition between Bitcoin and other competing software projects (which neither change the consensus rules nor reimplements the codebase).[77] These competitive forces may best be classified as internal and external competition. Internal competition refers to the ability of each individual to fork Bitcoin's codebase and blockchain and create her competing network or chain. External competition, similar to the market for corporate control, refers to the competitive pressures exerted by other cryptocurrency projects that would put substantial pressure on Bitcoin to improve its governance model. Unlike the market for corporate control for certain institutions (e.g., banks) in certain jurisdictions (e.g., Europe) that are somewhat paralyzed by extensive regulatory requirements,[78] Bitcoin faces a cut-throat competition in a market for cryptocurrencies without any entry or exit restrictions and no regulatory barriers to competition.[79]

The most important governance mechanism in Bitcoin is provided by its open-source software that can be tweaked and forked. One method of making a change in Bitcoin is through hard forks. A hard fork is a backward-incompatible change to the rules of the consensus. For example, increasing the block size of

[76] F. A. Hayek, *The Use of Knowledge in Society*, 35 The American Economic Review (1945).

[77] Bitmex Research, Competing with Bitcoin Core (2018).

[78] For the regulatory limitations imposed on the corporate takeover of banks in the EU, *see* Georgina Tsagas, *The Market for Coporate Control in the Banking Industry*, *in* The Law on Corporate Governance in Banks (Iris H-Y Chiu & Michael McKee eds., 2015).

[79] Lawrence H. White, *The market for cryptocurrencies*, 35 CATO Journal (2015).; In addition to such competition, the second dimension of competition that contributes to the healthy level of governance is the competition among each category of participants in the Bitcoin network. For example, miners competing for the block reward and fees, despite being resource intensive, provides a strong security for Bitcoin network. In addition, there is competition among users for block space. In the future, where there would be no block reward to be distributed to the miners, such competition would be essential for the security model of Bitcoin as it would incentivize miners to divert substantial resources to mining bitcoin and, hence securing the network. At this stage, where the transaction fees play a less important role on the security model of Bitcoin, the participation of the majority of users remains passive. This may be attributed to rational ignorance or user apathy, similar to the well-known phenomenon in the voting systems, which would result in citizens not participating in elections.

Bitcoin can be seen as a hard fork, in the sense that if a fully validating node would receive blocks with sizes higher than one megabyte, it will reject the block as it violates the rules of the consensus. Once the block height, where a specific hard fork is scheduled to activate, hits, each individual miner and user can determine which set of rules to follow and enforce.

There may be several methods for resolving a hard fork. First, if one chain accumulates lower amount of hash rate than its competing chain, it would take longer for it to reach the 2016 block cycle for the readjustment of the mining difficulty.[80] However, the chain with higher hash rate would continue building its blocks regularly and accumulating more PoW. In this case, following Nakamoto consensus users opt for following the chain with most accumulated PoW. The second way to solve the hard fork is for the users/full nodes to decide to follow the chain having the lower amount of accumulated PoW. This may render the chain with higher PoW irrelevant. The third method of resolving the hard fork is that two independent networks would run independently using replay protection. Indeed, nothing can prevent the two resulting chains from being mined by miners or validated by full node operators. This can result in two competing chains running in parallel, which could cause a great deal of uncertainty, that often receives lukewarm reception as *the dispute resolution mechanism of the last resort* in the Bitcoin ecosystem.

An extraordinary example of resolving disputes through hard forks in the Bitcoin ecosystem occurred in 2017 that came to be known as scaling crisis of 2017.[81] Bitcoin's block size limit is set at one megabyte by Satoshi Nakamoto in 2010 using a soft fork.[82] As the demand for using Bitcoin increased, so did the competition for block space on the Bitcoin blockchain and consequently the blocks were nearing their full capacity. This higher demand for block space sharply increased transaction fees in the Bitcoin network, which priced out many earlier use cases of Bitcoin, especially those involving micropayments. Such a development increased the demand for increasing the block size, in particular from the Bitcoin business community.

Various proposals were put forward for resolving the problem, ranging from increasing the block size from one megabyte to 2 megabytes or even to 8 megabytes to having a dynamic block size limit for Bitcoin. The first of such proposals were put forward in August 15, 2015 by the introduction of Bitcoin XT that was released by Gavin Andresen and Mike Hearn as a soft fork that could be turned into a hard fork down the

[80] For more technical details, see: ANTONOPOULOS, Mastering Bitcoin: Programming the Open Blockchain 253. 2017.

[81] Bitcoin itself can be viewed as an invention that emerged to overcome social scalability problem in the first place. Although the discussion of this paper is limited to technological scalability, the problem of social scalability stands at the core of the scalability issues in bitcoin. Indeed, the perceived inefficiencies in the PoW can be understood in the balance struck between social scalability and computational scalability. In the Bitcoin Blockchain the latter is sacrificed to improve the former. For more details, see: Szabo, UNENUMERATED, (February 09, 2017).

[82] *See* Eric Lombrozo, *Forks, Signaling, and Activation*, MEDIUM (June 18, 2017).; *See also* https://github.com/bitcoin/bitcoin/commit/f1e1fb4bdef878c8fc1564fa418d44e7541a7e83#diff-118fcbaaba162ba17933c7893247df3aR1422

road. This marked the beginning of a protracted dispute within the Bitcoin community. The opponents, such as Gregory Maxwell and Nick Szabo, argued that increasing the block size would lead to centralization, security risks, and variable and delayed confirmation times. Two main camps emerged on this dividing issue; one supporting vertical scaling solutions or second-layer solutions,[83] the other camp supporting horizontal scaling solutions or increasing the block size.[84] There has been various allegation of censorship by Bitcoin XT supporters claiming that Bitcoin core team had censored them. After much debate, Hearn resigned,[85] and Bitcoin XT had eventually been abandoned. Later projects were launched to increase the block size, such as Bitcoin Unlimited, Bitcoin Classic, and BitPay Core with very limited success. Some intermediate proposals, such as SegWit2X, were put forward, but failed to gather momentum. The protracted *civil war* within the Bitcoin community eventually resulted in the failed SegWit2X, and a hard fork leading to the creation of bitcoin cash (BCH) and subsequent user-activated soft fork (UASF) and the activation of SegWit on the legacy chain.[86]

Scholars have argued that this governance crisis and failure in conflict resolution amount to a fragile decision-making mechanism within the Bitcoin network.[87] However, with the benefit of hindsight and the fact that the narrative of Bitcoin has shifted from being a payment system to becoming a store of value and a digital infrastructure for clearing and settling transactions without being subject to censorship, Bitcoin's adversarial governance mechanisms in fact have contributed to its long-term goal of establishing such an image of Bitcoin as a censorship-resistant network.

In addition to role of the above-mentioned internal competition in Bitcoin governance, Bitcoin faces external competitive forces from both competing cryptocurrencies as well as fiat currencies. As there are virtually no barriers to entry and exit, Bitcoin - to the extent that it is used as a substitute to currencies - competes with currencies in the forex markets. In addition, to the extent that Bitcoin is viewed as a store of value, it competes with traditional commodities used as store of value such as gold. Such competitive forces have indeed huge implications for Bitcoin's governance and pushes Bitcoin to continuously improve itself to stay relevant.[88]

[83] ANTONOPOULOS, Mastering Bitcoin: Programming the Open Blockchian 300-321. 2017.

[84] Poon & Dryja, (2016).; Wirdum, BITCOIN MAGAZINE, (4 April 2018).; Jedusor, (19 July 2016).; Wirdum, BITCOIN MAGAZINE, (12 August 2016).

[85] Hearn, MEDIUM, (January 14, 2016).

[86] Laura Shin, *Will This Battle For The Soul Of Bitcoin Destroy It?*, FORBES Oct. 23, 2017. Similar controversies happened on the Ethereum's blockchain due to the loss of funds associated with DAO project, resulting in a chain split and the creation of Ethereum and Ethereum Classic. *See* Quinn DuPont, *Experiments in algorithmic governance: A history and ethnography of "The DAO," a failed decentralized autonomous organization*, *in* BITCOIN AND BEYOND: CRYPTOCURRENCIES, BLOCKCHAINS, AND GLOBAL GOVERNANCE (Malcolm Campbell-Verduyn ed. 2018).; Report of Investigation Pursuant to Section 21(a) of the Securities Exchange Act of 1934: The DAO. (July 25, 2017).

[87] De Filippi & Loveluck, INTERNET POLICY REVIEW, 9 (2016).

[88] However, one would expect that an easy exit enabled by both internal and external competitive forces in Bitcoin governance would lead to the tendency of having less voice being raised by stakeholders in Bitcoin, as evidenced by some prominent Bitcoin

Externalities and public goods nature of Bitcoin governance

Although there is no general agreement on whether the law and governance belong to the category of public goods,[89] governance exhibits many properties of public goods in so far as it is non-excludible and non-rivalrous. In the same vein, to the extend it is non-excludible, and non-rivalrous,[90] Bitcoin governance possess the properties of public goods. Although bitcoin, as a unit of account in the Bitcoin network, is not a public good as it is both excludible and rivalrous, the Bitcoin network, and in particular, the maintenance and governance of the protocol could be said to be a public good. As the benefits of good governance in Bitcoin are shared by everyone, and the use by one participant, does not decrease such benefits to other participants, Bitcoin network participants face the collective action and free-rider problems. Therefore, it is likely that maintenance and governance of Bitcoin would be under-provided if left to the markets.

Despite public-good designation of Bitcoin governance, upon closer inspection, the main market failure in the Bitcoin governance may be due to classic commons property of Bitcoin governance. The permissionless nature of Bitcoin means that no one can be excluded from either participating in the Bitcoin governance or from consuming the benefits of good governance in Bitcoin, however, as the use of block space by one user reduces the amount of space left to other users, Bitcoin can be said to be a rivalrous good or service. The classic problem of commons is that no one has adequate incentives to contribute to its governance, because the benefits of such contributions cannot be fully captured by the contributors,[91] and the overuse of the common resource often leads to the tragedy of the commons and its eventual depletion.

The market failures due to the commons nature of Bitcoin especially applies to fully validating nodes. Such node operators, which are the watchdogs of miners, may have no incentive to operate a fully validating node in the absence of proportionate reward for their operation due to the free-rider problem. The same applies to developers whose contribution to the code is voluntary, reputation-based or in some instances market driven, i.e., financed by start-ups or corporations. Such collective action problems may be addressed using a variety of mechanisms. Various incentive mechanisms embedded in the Bitcoin network have incentivized participants to contribute to Bitcoin governance and to the maintenance of the code. For

developers leaving the project rather than staying and engaging in the long-term development of the project. For an excellent backgrounder, see ALBERT O. HIRSCHMAN, EXIT, VOICE, AND LOYALTY: RESPONSES TO DECLINE IN FIRMS, ORGANIZATIONS, AND STATES (Harvard University Press. 1970).; Atik & Gerro, STANFORD JOURNAL OF BLOCKCHAIN LAW & POLICY, (2018).

[89] *See* Tyler Cowen, *Law as a Public Good: The Economics of Anarchy*, 8 ECONOMICS AND PHILOSOPHY (1992).; David D. Friedman, *Law as a Private Good: A Response to Tyler Cowen on the Economics of Anarchy*, 10 see id. at (1994).

[90] Commons are goods that are non-excludible in the sense that others cannot be excluded from its consumption, but they are rivalrous, meaning that its consumption by one individual will decrease its availability for others. In contrast, public goods are both non-excludible and non-rivalrous.

[91] The term "tragedy of the commons" coined by Garret Hardin in his famous article in 1968 indicates a source shared by a group of people, in which individuals are granted the right to use that given resource without any cost-efficient way of monitoring or limiting each other's use. This will lead to the depletion of that resource. *See* Garrett Hardin, *The tragedy of the commons*, 162 SCIENCE (1968).; CHARLOTTE HESS & ELINOR OSTROM, UNDERSTANDING KNOWLEDGE AS A COMMONS: FROM THEORY TO PRACTICE 4 (MIT Press. 2007).

example, using bitcoin for payments involves fees that discourage spamming and the overuse of the network. In the future, it is likely that the industry would provide financial incentives for contribution to Bitcoin development. In certain other cryptocurrencies, there have been suggestions to provide subsidies for cryptocurrency developers by taxing miners.[92] Though the dislike for compulsory taxation would make the realization of such proposals in the Bitcoin ecosystem very unlikely, more market-based and voluntary incentive mechanisms are likely to emerge.

Despite the fact that Bitcoin governance is prone to the free-rider problem, thus far, its governance proved to be both effective and relatively successful. This is puzzling as insights from the economic literature, such as public goods, club goods,[93] the tragedy of the commons,[94] and game theory[95] suggest that free-rider problems would result in a failed and broken governance model in Bitcoin. Perhaps the solution to this puzzle should be found in different schools of thought. For example, in contrast to the standard economic theory that predicts the tragedy of the commons and over-exploitation and depletion of the common resource, studies - pioneered by Elinor Ostrom - have found that such mainstream economic thought has not been entirely accurate, and where common resources exist, a variety of bottom-up mechanisms have been emerged to address the issue.[96] Therefore, it is not clear whether the tragedy of the commons is a reality or a myth.

On the other hand, deriving evidence from the pattern of social production in the internet (e.g., Wikipedia), some scholars suggest that the new patterns of production are emerging that are based on different incentive mechanisms than those studied under the classical economic theory. The theory of commons-based social production, put forward by Yochai Benkler, is a prominent example of this approach.[97] Yet another way to explain the success of the Bitcoin community in maintaining the code and resolution of disputes without any resort to external factors or third parties is by reference to the literature on *law without order* that highlights the insignificance of the law and formal mechanisms in social coordination and dispute

[92] Jiang Zhuoer, *Infrastructure Funding Plan for Bitcoin Cash*, MEDIUM (January 22, 2020).

[93] Richard Cornes & Todd Sandler, The Theory of Externalities, Club Goods, and Public Goods (Cambridge University Press Second ed. 1996).

[94] Hardin, SCIENCE, (1968).; ELINOR OSTROM, GOVERNING THE COMMONS: THE EVOLUTION OF INSTITUTIONS FOR COLLECTIVE ACTION (Cambridge university press. 1990).; Elinor Ostrom, *Coping with the tragedies of the commons*, 2 ANNUAL REVIEW OF POLITICAL SCIENCE (1999).; Franciesco Parisi & Ben Depoorter, *Commons and Anticommons*, *in* THE ENCYCLOPEDIA OF PUBLIC CHOICE (Charles K. Rowley & Friedrich Schneider eds., 2004).; Francesco Parisi, et al., *Simultaneous and Sequential Anticommons*, 17 EUROPEAN JOURNAL OF LAW AND ECONOMICS (2004).

[95] Stephen Morris & Hyun Song Shin, *Global Games: Theory and Applications*, *in* ADVANCES IN ECONOMICS AND ECONOMETRICS: THEORY AND APPLICATIONS, EIGHTH WORLD CONGRESS (Lars Peter Hansen, et al. eds., 2003).; Joseph Abadi & Markus Brunnermeier, *Blockchain economics*, NATIONAL BUREAU OF ECONOMIC RESEARCH WORKING PAPER 25407 (2018).

[96] Ostrom, ANNUAL REVIEW OF POLITICAL SCIENCE, (1999).; OSTROM, Governing the commons: The evolution of institutions for collective action. 1990.; Elinor Ostrom, *Beyond Markets and States: Polycentric Governance of Complex Economic Systems*, 100 THE AMERICAN ECONOMIC REVIEW (2010).

[97] YOCHAI BENKLER, THE WEALTH OF NETWORKS: HOW SOCIAL PRODUCTION TRANSFORMS MARKETS AND FREEDOM (Yale University Press. 2006).

resolution, and instead puts the spotlight on the importance of unwritten social norms.[98] Irrespective of the reasons, Bitcoin seems to be like the proverbial Bumblebee that in theory it would not fly, but in practice it does.[99] The above mentioned three non-mainstream theories may be helpful in shedding some light on how Bitcoin governance and several other phenomena that rely on the cooperation and collaboration in distributed systems work and manage to maintain the code and resolve disputes.

Bitcoin governance and downward accountability

It is argued that Bitcoin governance, similar to other governance models, needs legitimacy. Although at the first blush, Bitcoin governance seems to be ambivalent about the concept of legitimacy, this paper argues that legitimacy in Bitcoin governance stems from its downward or market accountability,[100] meaning that the users are afforded with the mechanisms that enable them to exert a significant influence on Bitcoin governance. Among others, users may employ a variety of mechanisms to participate in the governance of the network. *Firstly*, users may threaten not to run the software proposed by developers. *Secondly*, those users running fully validating nodes may threaten not to validate certain blocks broadcast by miners. *Thirdly*, users may vote with their feet.

The importance of network effects in Bitcoin renders foot voting a very powerful mechanism in Bitcoin governance because abandoning the project by its users could amount to its immediate demise. Although foot voting is available to all participants in Bitcoin governance with varying degrees of costs and benefits, it seems it is the cheapest option for users, compared to miners or developers. This enables users to leverage this powerful device in any dispute over Bitcoin governance, ultimately resulting in a market-driven bottom-up decentralized form of governance for Bitcoin.[101]

In addition, users, especially those running fully validating nodes, have considerable leverage against miners, because Bitcoin governance largely relies on the emergent consensus through a network-wide

[98] ROBERT C. ELLICKSON, ORDER WITHOUT LAW: HOW NEIGHBORS SETTLE DISPUTES (Harvard University Press. 1991).

[99] This metaphor is expressed by the former president of the European Central Bank (ECB) about the euro. *See* Mario Draghi, *Speech by Mario Draghi, President of the European Central Bank at the Global Investment Conference in London 26 July 2012*, (2012).

[100] Colin Scott, *Accountability in the Regulatory State*, 27 JOURNAL OF LAW AND SOCIETY (2000).; Colin Scott, *Regulation in the age of governance: the rise of the post-regulatory state*, *in* THE POLITICS OF REGULATION: INSTITUTIONS AND REGULATORY REFORMS FOR THE AGE OF GOVERNANCE (Jacint Jordana & David Levi-Faur eds., 2004).

[101] Given the roles of other stakeholders such as miners and developers, this renders Bitcoin governance to a strong governance framework for dFMIs similar to what is sometimes dubbed as polycentric or networked governance framework in conventional governance studies. For the concept of polycentric governance model, *see* David Zaring, *Financial Regulation's Overlooked Networks*, *in* RECONCEPTUALISING GLOBAL FINANCE AND ITS REGULATION (Ross P. Buckley, et al. eds., 2016);David Zaring, *International law by other means: the twilight existence of international financial regulatory organizations*, 33 TEXAS INTERNATIONAL LAW JOURNAL (1998). ANNE-MARIE SLAUGHTER, A NEW WORLD ORDER (Princeton University Press. 2004).; Anne-Marie Slaughter, *Global government networks, global information agencies, and disaggregated democracy*, 24 MICHIGAN JOURNAL OF INTERNATIONAL LAW 1041(2003).; FINCK, 172-178. 2019.

agreement of rules that are ultimately enforced by the users running full nodes.[102] In Bitcoin, miners have to create blocks according to these rules and submit them to the network of full nodes for validation. Then, full nodes validate the block by downloading the block and verifying if those blocks match the consensus rules of the client. The nodes will not reject any block that is considered valid in terms of the most accumulated PoW. However, if the block does not match the criteria of a valid block or does not have the most accumulated PoW, it will be rejected by the nodes. In sum, miners are bound by the rules of the network and have to implement them, otherwise, those blocks will be rejected by the full nodes, who take on the role of watchdogs, constantly watch miners for their compliance with Bitcoin's consensus rules.[103]

Although miners may use the threat of forks (Miner Activated Soft Forks (MASFs) or hard forks) or they may even use the threat of a 51% attack after the fork to kill the parallel chain as a mechanism to exert influence on Bitcoin governance, ultimately, it is the user community who decides whether to use the fork supported by a subset of miners. For example, in the Bitcoin scaling saga and the activation of the User Activated Soft Fork (UASF), some nodes made a commitment to represent the views of the users as well as some segments of the business community by advocating a soft fork implementation of Segregated Witness (SegWit), in which both SegWit and non-SegWit compliant blocks could be processed.[104] As the majority of the miners did not adopt the SegWit update for a long time after the release of the code, certain Bitcoin users installed a client that threatened to suspend the Nakamoto consensus by ignoring the blocks relayed by the miners refusing the SegWit after a specific date. If this situation would have dragged on, that soft fork would have been resulted in a contentious fork. The mere threat to Bitcoin utility and value from such a contentious fork and hence the miners' business model finally persuaded the miners to stop resisting the SegWit update and acquiesce to the users' intended result.[105] The same applies to the powers of developers who can propose software rule changes. If changes proposed or even implemented by developers would not be supported by significant number of users, those changes are eventually doomed.

The precedent in Bitcoin's history shows that users have decided to even ignore Nakamoto consensus due to the fact that the longer chain or the chain with the most accumulated PoW did not represent the social contract the users were perceived to be parties to.[106] As explained before, this happened in the 2010 integer overflow bug where within 3 hours, Satoshi published a new Bitcoin client and rewound the hyperinflated

[102] Although users' contribution to Bitcoin governance could be hampered by collective action problems, the technological means, which are embedded in the Bitcoin network, such as the ability to signal one's preferences via the network itself, would help alleviate coordination problems.

[103] ecurrencyhodler, *Bitcoin Governance*, MEDIUM (February 5, 2019).

[104] Aaron van Wirdum, *The Latest Twist to the Block Size Debate is Called a "UASF"*, BITCOIN MAGAZINE (March 2, 2017).

[105] Atik & Gerro, STANFORD JOURNAL OF BLOCKCHAIN LAW & POLICY, 7 (2018). *See also* Alyssa Hertig, *UASF Revisited: Will Bitcoin's User Revolt Leave a Lasting Legacy?*, COINDESK (August 3, 2017).

[106] Hasu, et al., A model for Bitcoin's security and the declining block subsidy § 2019 (2019).

chain.[107] Similarly, in 2013 inflation bug incidents, where the 0.7/0.8 consensus bug split the blockchain into two separate chains for several hours.[108] In this case, the incident could only be resolved when developers and the mining pools suspended the fork-choice rule temporarily, by supporting the 0.7 fork and abandoning the 0.8 chain. Although this required some miners to forgo the block rewards form the 0.8 chain, they did so with the expectation that it would eventually maximize the overall value of the network.[109]

Ultimately, the key takeaway in Bitcoin governance is that users are not bound to follow the miners or developers if a majority of the users do not share the same ideas about the future path of the network. In addition, if there is a disagreement over how to maximize network utility, users can suspend Nakamoto consensus and *disempower* miners.[110] This also applies to any attacker to the network, as the users may stop following the chain with the most accumulated PoW, the attacker must take this into account before spending resources on attacking the Bitcoin network.[111]

Conclusion

Bitcoin's main value proposition is its censorship-resistant property, which is backed up by a technological innovation that allows participants to transact in a trust-minimized environment. To retain and maintain such a property, it is highly important for Bitcoin to remain decentralized. Decentralization entails that various participants in the Bitcoin network, in particular users, have an effective voice in Bitcoin governance. Although a variety of stakeholders in the Bitcoin network take part in Bitcoin governance and there is no single or group of homogenous participants that has the final say, it appears that the ultimate decision is principally made by those who can successfully fork Bitcoin and convince the majority of users to shift to the new chain. In this regard, the users of the Bitcoin network seem to possess the ultimate authority to decide which software to install and run or which implementation to follow. The possibility of forking means that unlike many other modes of resolving the societal collective action problems, which ultimately relies on coercion, Bitcoin governance is based on deliberation, persuasion, volition, and choice.

To resolve disputes in the Bitcoin network, unlike other political decision-making processes, in addition to open and free entry and exit, each and every participant can fork the codebase or the blockchain and create her own version of Bitcoin and try to persuade other users to follow her version of the chain or code. By providing for the technical possibility that the new chain maintains the history of the blockchain going back

[107] Charlie Shrem, *Bitcoin's Biggest Hack In History: 184.4 Billion Bitcoin from Thin Air*, HACKERNOON (January 11, 2019).
[108] This incident related to the 0.8 update to the Bitcoind, which was the most popular Bitcoin implementation. The new software had an unintended change to the consensus rules causing the block 225,430 to become incompatible with older clients. For more details, *see* Vitalik Buterin, *Bitcoin Network Shaken by Blockchain Fork*, BITCOIN MAGAZINE (March 13, 2013).
[109] Id. at.
[110] Hasu, et al. 2019.
[111] Id. at.

to the genesis block, as well as the fact that the holders of the legacy coins receive the new coins proportionate to their holdings in the legacy chain, Bitcoin is most open to competition provided by forks.[112] To make a comparison, there is also foot voting in corporate law, however, forking is something more than foot voting. Forking is similar to incorporating a new company with the same brand name and certain tangible and intangible assets without any new capital expenditure, the only limitation being convincing the greater number of participants (good will) to follow the new forked chain. This significantly lifts the barriers to entry and invites even more competition.

There are additional factors that strengthen the position of users as the ultimate decision makers in the Bitcoin network. For example, users may decide not to install and run the particular software developed by developers. Furthermore, they may decide not to validate the blocks broadcast by certain miners. Last but not least, they have their own mechanisms of forking Bitcoin (e.g., UASF). Contrary to the popular belief that the miners and developers control Bitcoin, the history of forks in Bitcoin and the constraints that the developers and miners face in influencing the protocol or users confirm that the ultimate decision makers are users and markets rather than a relatively centralized groups of developers or miners. Give the multiparty decision-making process in Bitcoin as well as its reliance on users at the apex of decision-making processes, Bitcoin governance remains decentralized and it is unlikely that any single actor could have a disproportionate impact over the protocol.

Thus far, Bitcoin governance has worked well in addressing the potential market failures. However, potential future challenges lie ahead, and it is not clear whether Bitcoin's decentralized governance mechanisms would be able to deal with future crises. One of the issues that is likely to give rise to governance crises is the declining block reward or subsidy and its implications for Bitcoin security model.[113] Various proposals or mechanisms for dealing with such an issue have been put forward, such as improving block space, perpetual issuance, crowdfunding, and adapting the supply of the block space.[114] Another important issue would be the discussions about a change in security and consensus mechanisms and potential shift from PoW to proof of stake (PoS) or other mechanisms of securing and reaching consensus in Bitcoin. Governance crises are also likely to emerge if the tamper-resistance property of Bitcoin would be called into question. For example, in the immediate aftermath of the Binance hack in 2019, there have been discussions about Bitcoin blockchain reorganization to reverse transactions and undo the damage. Although such discussions faced immediate and strong resistance from users and developers, which led to

[112] In other words, in addition to various types of foot-voting mechanisms, the Bitcoin network is welcome to forking, despite the natural aversion and dislike expressed by some participants against forking Bitcoin. However, if forking is frequently used and become widespread, it may lead to market fragmentation and present a risk to users, consumers or investors. *See* Decentralised financial technologies: Report on financial stability, regulatory and governance implications. (6 June 2019).
[113] Raphael Auer, *Beyond the doomsday economics of "proof-of-work" in cryptocurrencies*, BIS WORKING PAPERS NO 765 (2019).;
[114] Hasu, et al. 2019.

the concession by miners and exchanges not to pursue such a proposal, such issues are likely to bring the question of governance to the fore again. Only time will tell whether Bitcoin and its governance model can address those critical governance issues.

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