Howell, Bronwyn E.; Potgieter, Petrus H.; Sadowski, Bert M.

Conference Paper

Governance of Blockchain and Distributed Ledger Technology Projects

2nd Europe - Middle East - North African Regional Conference of the International Telecommunications Society (ITS): "Leveraging Technologies For Growth", Aswan, Egypt, 18th-21st February, 2019

Provided in Cooperation with:

International Telecommunications Society (ITS)

Suggested Citation: Howell, Bronwyn E.; Potgieter, Petrus H.; Sadowski, Bert M. (2019) : Governance of Blockchain and Distributed Ledger Technology Projects, 2nd Europe - Middle East - North African Regional Conference of the International Telecommunications Society (ITS): "Leveraging Technologies For Growth", Aswan, Egypt, 18th-21st February, 2019, International Telecommunications Society (ITS), Calgary

This Version is available at:
http://hdl.handle.net/10419/201737

Governance of Blockchain and Distributed Ledger Technology Projects

Bronwyn E. Howell[*], Petrus H. Potgieter[†], Bert M. Sadowski[‡]

Abstract

Blockchains are the most well-known example of a distributed ledger technology (DLT). Unlike classic databases, the ledger is not maintained by any central authority. The integrity of the ledger is maintained automatically by an algorithmic consensus process whereby nodes vote and agree upon the authoritative version. In effect, the consensus algorithm operates in the manner of a decision-making process within a governance system.

The technological characteristics of blockchain systems are well documented (Narayanan, Bonneau, Felton and Miller, 2016). We propose that one of the reasons why it has so far proved very difficult to seed large-scale commercial DLT (blockchain) projects lies in the arena of project ownership and governance. Unlike classic centralised database systems, DLTs have no one central point of "ownership" of any of the system's infrastructure or data.

In this piece of exploratory research, we propose applying theories of club governance to both the technical design and operational development of a range of DLT (blockchain) systems, including (but not necessarily limited to) cryptocurrencies and enterprise applications to explore how they can explain the development of (or lack of development of) sustainable solutions to real business problems. There are many parallels to the governance arrangements observed historically in the origins of complex distributed telecommunications networks.

Keywords: blockchain, distributed ledger, governance, club governance, distributed consensus

1 Introduction

> "Reform is a profoundly political process, not a technical one." Fukuyama (2014, 161)

Blockchains are the first, and most well-known example of a distributed ledger technology (DLT). A distributed ledger (DL) is a database (or file) spread across several nodes or computing devices. Each node in a network has access to (and probably saves) an identical copy of the ledger. Unlike

[*]School of Management, Victoria University of Wellington, bronwyn.howell@vuw.ac.nz

[†]Department of Decision Sciences, University of South Africa, potgiph@unisa.ac.za / php@grensnut.com

[‡]Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology, b.m.sadowski@tue.nl

classic databases, the ledger is not maintained by any central authority. The integrity of the ledger is maintained automatically by an algorithmic consensus process whereby nodes vote and/or agree upon the authoritative version, which is then updated and saved independently on each node. In effect, the consensus algorithm operates in the manner of a decision-making process within a governance system.

Blockchain DLs use a chain of blocks linked to one another and secured using public-key cryptography to provide a secure and valid distributed consensus. A blockchain is usually distributed across and managed by peer-to-peer networks. Its append-only structure only allows data to be added to the database: altering or deleting previously entered data on earlier blocks is impossible. Blockchain technology is therefore well-suited for recording events, managing records, processing transactions, tracing assets, and voting.

The technological characteristics of blockchain systems are well documented (Narayanan, Bonneau, Felten, Miller and Goldfeder, 2016). Considerable faith has been placed in the technology as a means of revolutionising digital transacting (Mulligan, Scott, Warren and Rangaswami, 2018; Crosby, Nachiappan, Pattanayak, Verma and Kalyanaraman, 2015; Czepluch, Lollike and Malone, 2015; Swan, 2015). However to date, outside of the arena of highly-publicised cryptocurrencies such as Bitcoin and Ethereum, few examples exist of the use of the technology to support significant economic activities. Nonetheless, plans for many other blockchain systems have been announced – for example Sovrin for identity management, and Halo for supply chain management.

Comparatively little has however been documented so far about the *governance* of blockchain systems and the commercial activities they support, beyond the algorithmic voting processes via which the nodes agree on the authoritative version of the DL. This paper represents an exploratory endeavour to address this gap.

We begin by reviewing current interpretations of blockchain "ownership" and "governance". Neither the data in the ledgers nor the software governing blockchain operations is claimed to be owned by anyone in particular. Nonetheless ongoing responsibility for the rules governing their ongoing operation must be assumed by someone if they are to be created and operated successfully for commercial endeavours. We propose that the governance of DL systems can be analogised to that of clubs.

While some of the governance rules are embedded in system software, and may be costly and difficult to change, leading to stable ledger content, other rules are embedded in the institutional arrangements linking system participants (club members) outside of the software, and may not be so costly to change, depending upon how key decision-making rights are distributed across them. The stability of a given DL system will depend upon the interaction of the decision rights allocated and exercised within the software with rights allocated and exercised outside of it.

From the theoretical discussion, we develop a framework for examining a given blockchain DL system to identify and evaluate the effectiveness of its governance arrangements given its specific commercial application. We apply the framework in two case studies: the cryptocurrency Bitcoin and the identity management system Sovrin. While both claim to be public blockchains using very similar "proof of work" algorithms to agree ledger content, Sovrin's structure as a "permissioned" blockchain (it poses barriers to entry for node operators (they are required to become Stewards) and requires end-users to have a relationship with node operators separately verifiable from their blockchain relationship) differentiates it from Bitcoin. We suggest that the costs of changing Sovrin's governance arrangements as circumstances change are much lower

than those of Bitcoin. Thus, Bitcoin is less susceptible to successful forking than Sovrin. Sovrin's stability relies more strongly upon the alignment of the interests of its node operators (stewards) than does Bitcoin (miners are node operators), because Sovrin's governance arrangements allow them both greater control over changes to software content and lower costs of co-ordinating successful forking than their Bitcoin comparators.

2 Ownership and governance of distributed ledgers

Who owns and governs a blockchain system? One view (prevailing with the development of cryptocurrencies) is that a specific DL application is "governed by no-one", because it is "owned by no-one" (Sovrin, 2018). Anyone who wishes to use the blockchain application may do so (it is "public"). Unlike web-based applications, any user operating as a node has a copy of both the ledger and the software required to participate in the system (although some classes of user may interact via web pages managed locally by a node operator). Neither the data nor the software are proprietary to a single controlling entity. As the software code is open source, any node operator is free at any time to make changes to the code (which they themselves are using) and institute a new blockchain operating independently of the first (termed "forking"), without facing the disadvantage of a centralised system of not having access to the accumulated historic data. All blocks up to the point of forking are identical in both the original and the new chain.

Although the components of a blockchain are "owned by no-one", it cannot be said that a blockchain system is "governed by no-one". All systems operate within a framework of rules, either derived implicitly from the norms and cultures of the participants or explicitly articulated in formal agreements (such as constitutions and contracts) (Williamson, 1999; 2000). Collectively, these rules comprise the governance arrangements under which systemic interaction takes place. Within them, in order to co-ordinate participants and streamline decision-making, selected groups of individuals are granted superior decision-making rights by assuming those ceded to them by specific subsets of system users. Governance arrangements can emerge endogenously over time (bottom-up) (for example, as has occurred with the constitutional arrangements of nation states), or be imposed from the outset (top-down)(for example, following military conquest, or in the Constitutions and Articles of Incorporation of firms, marketplaces (e.g. stock exchanges, clubs and trusts) (Ostrom, 1990; 2005).

Efficient and effective governance arrangements will specify both the set of rules prevailing for normal transacting, and provisions whereby those rules can be changed in response to changing circumstances. When the provisions for rule changes are explicit, and users have clear means of observing those charged with the responsibility for managing the rules process and holding them to account for their actions (or inactions), then the systems will tend to be more efficient than if the rules and/or the identity of the decision-makers are unclear, those with decision-making rights can exercise them covertly, and there are no clear means of users holding the decision-makers to account or the costs of doing so are so high as to render the probability of occurrence remote (Hansmann, 1996; Cordery & Howell, 2017).

The classic shareholder-owned firm provides an example of one such explicit system, where shareholders give up their rights to make decisions about the day-to-day use of the firm's assets to boards and management to facilitate more efficient firm functioning than if the shareholders themselves were required to undertake co-ordination (Berle and Means, 1937; Williamson, 1985). Other examples include the arrangements pertaining to the management of non-owned

non-rival and non-excludable "public" goods where governments as trustees exercise decision-making rights on behalf of all citizens, and those pertaining to the management of non-rival but excludable "club" goods enjoyed by an identified population (as a subset of general citizenry).

2.1 Distributed ledger systems as clubs

Club theory, proposed initially by Buchanan (1965) in respect of clubs dealing in rival, excludable goods provided and consumed by volunteer-members has been expanded subsequently to take account of the separate dimensions of non-rivalry and non-excludability of the goods provided by the clubs, and the exclusivity of club membership (e.g. Olson, 1989; Comes and Sandler, 1996). Important work by E Ostrom and V Ostrom (Ostrom, 1990; 2005; 2010; Ostrom, 2014) melded the concept of the club with theories of self-organising governance systems, federalism and polycentrism in government, demonstrating that common resources could be managed successfully without government regulation or privatisation, by way of decentralised entities operating as polities using representation relationships (i.e. "membership") rather than contractual assignment of rights proportional to asset ownership to allocate decision-making control.

In this view, a DL system (DLS) is a club with (arguably) open membership. It is a 'public system' as any member of the public agreeing to abide by the rules can join in order to use it. At any point in time, the club membership is defined by those participating in the DLS. The DLS is governed by rules covering both membership and operation. Various classes of membership are usually determined by the nature and form of interactions the members have with it. Operational rules cover how routine operations will occur and how, in the event that conflicts arise that cannot satisfactorily be resolved within the existing rules, the rules can be changed to maintain the integrity of the system and thereby ensure ongoing use by the members (Cordery and Howell, 2017). The consensus arrangements by which the DLS resolves conflicts about the content of the DL constitute but one part of the system's operational rules.

Importantly, decision-making powers attach to and differ by membership status. Axiomatically, users operating nodes that agree the ledger content participate differently in decision-making from end-users participating only by using the DLS to transact with other end-users. The over-arching institutional arrangements under which the DLS operates – including the allocation to membership status and associated decision-making rights – must be decided first by founding members in order for the system to be created. The founding members will determine the original governance rules – both those coded into the software and other non-coded arrangements. They exercise considerable design control. It matters, therefore, whether these individuals exert decision-making influence in either or both of operational and other representative roles. The rules must address the potential for conflicts in these decision-making responsibilities to be resolved.

The founding members assume fiduciary duties to both future members and the DLS club as a whole. To the extent that the initial rules establish hierarchies of membership and allocation of decision-making responsibilities, these can be thought of as defining the club committees and sub-committees, with the founding members being allocated to different roles. Once the DLS becomes operational, the roles may transfer to new members as they join and as the new membership begins exercising its rights. This may include the replication of "branches" or "sub-branches" of the club with their own committees and sub-committees as the number of node operators expands. The DLT rules specifying how these distributed entities are federated

into the overall club governance functions and how decision-making rights and responsibilities are distributed across them are effectively constitutions.

However, there is also a threat that new members are offered discriminatory treatment by incumbent members or are excluded altogether. Rey and Tirole (2007) have shown that incumbent members have an incentive to exploit their monopoly power or restrict entry by new players. Within the blockchain context, that is a massive centralization problem due to the concentration of mining power within a small group of initial members. However, there is a movement from the concept of Proof of Work (PoW) towards the Proof of Stake (PoS) that tries to address this problem by giving greater voting power to those who have a stake in the venture.

For a discussion of blockchains as constitutional entities, where club members are equated to citizens, see Berg, Berg and Novak (2018) and Berg, Novak, Potts and Thomas (2018). Our analysis differs from theirs in that they focus only on the ongoing operation and management of a DLS once it has been created, whereas we examine both instantiation and ongoing operation. We also extend our analysis to include relationships between different classes of member outside of the operation of the ledger – that is, we see the constitutional rules encompassing both software and non-software elements. Their analysis focuses predominantly on the software-mediated elements.

2.2 DLS governing rules and club stability

The initial DLS is offered as a "take it or leave it" package to the first public users – that is, as a 'top-down' imposition as per Ostrom (2009) (e.g. implementation of a new stock exchange). This differs from the voluntary agreement of federated arrangements when extant groups negotiate the rules under which their activities become linked (e.g. when two stock exchanges merge) – that is, bottom-up arrangements enabling large-scale co-operation (Bednar, 2009). In either case, once the rules are agreed, they can change via either gradual reinterpretation of the rights and obligations defined by apparently stable governance rules, or substantive episodic change in the formal structure of the governance rules (analogous respectively to Buchanan's distinction between regular political activity and constitutional moments – Buchanan and Tullock, 1962; Buchanan and Brennan 1985; Congleton, 2014).

Both clubs and political entities will function effectively so long as there are credible commitments by members/citizens to monitoring, enforcement/sanction and conflict resolution within the existing rules (North, 1993; Ostrom, 2005), and the ability to bring about changes to those rules to meet the demands of changing circumstances (Tarko, Schlager and Lutter, 2018). The former are enhanced by rule stability and certainty; yet too much stability can lead to fragility if the governance rules are not well-adapted to new challenges. But allowing for ready change may also lead to instability as competing centres of authority may attempt to devise and impose rules to benefit themselves at the expense of others. The challenge is to find a workable balance between stability and flexibility in the governing arrangements.

As long the original (or extant) arrangements suit the (ever-changing) membership, a DLS will survive – albeit in a dynamically-adapting form as constitutionally defined. However, when an issue arises which cannot be agreed using the current rules, a fork may occur (a 'break-away club' forms). The ongoing success of both clubs/DSLs now depends on the proportion of the members who move to the new DLS or remain with the original one.

On the one hand, the ease with which disaffected node operators can break away provides significant pressure on the existing DLS to be designed for and operate consistently in node operator interests, which may not necessarily be in the interests of end-users. On the other hand, if end-users' interests are compromised, they will not participate in the DLS in the first place. Careful balancing of interests of both node operators and end users is necessary to both attract a critical mass of node operators and user-members and maintain DLS stability. If these are not well-balanced, the DLS will be unstable – that is, prone to either failure (no members) or forking.

2.3 Trading flexibility and stability in a dynamic context

However, to the extent that many of the governance arrangements must be coded into the software in advance of the system beginning operation, DLS design is subject to the bounded rationality of human designers. Arrangements that are satisfactorily balanced at one point in time within one set of wider environmental circumstances may not be optimal if those circumstances change (e.g. substantial changes in electricity prices for node operators). Typically, the more flexible the governance rules are, the more easily they can be altered to take account of the changes and the DLS as an institutional entity will be more stable. However, the co-ordination required to institute the changes can be costly.

A strength of DLS consensus algorithms is that the costs of co-ordinating to change the software-governed outcomes are very high. This leads to high confidence in the integrity of the data held in the ledgers. However, the costs of changing the non-software-governed elements can, but need not, be high. The lower are the costs of co-ordinating the activities of the human entities using the system, the easier it is to institute changes – either to enhance the operation of the existing DLS by a general agreement of all with sufficient powers to amend the software to maintain its stability, or to facilitate a successful fork by persuading a critical mass of users of the existing system to support the break-away rules and system, instead of the original.

If all end users are definitively linked via mechanisms that enable them to be easily identified, then it is cheaper to communicate with them to co-ordinate any action than if they cannot be easily identified. Change of either type (forking; mutually-agreed rule and software changes) is more likely as co-ordination costs are lower. Furthermore, the greater is the extent to which the member activities are linked outside the day-to-day operation of the DL, the cheaper is the cost of co-ordinating activities for (for example), end users to follow the decisions of their relevant node operator when it becomes necessary to decide whether to support a software change for the existing system or to support a fork.

Assume, for example, that the node operators are required by the DLS rules to be identified and known to each other in order to be authorised to act in this capacity. That is, the DLS is a permissioned system. The cost of co-ordinating activity amongst a known number of identified individuals is less than where neither the number nor the identity of the members is known. Both agreed changes maintaining existing DLS integrity and forking will be less costly, suggesting that permissioned systems may be less stable. In the long run, this could have the effect of making it harder for the DLS to attract and retain new node operators, and thus build membership scale quickly. This effect could be overcome in part by adopting rules that make forking more expensive – for example, requiring the payment of a substantial bond (membership fee) that is forfeited if the member initiates or joins a fork.

The requirement for a node operators to develop their own software to interface with the DLS offers only a weak form of a bond against forking once the desired system has been selected from the range available. If a successful fork occurs, then the software will be equally useful on either variant (at least initially). Thus expected DLS scale rather than stability likely has a greater influence on the selection of the system by a given node operator. However, the advantages of a flexible, permissioned system become more evident when certainty about the future environment in which the DLS will operate is lower.

Assume now that a DLS is operating in a volatile environment, where changes in these external circumstances may alter the returns to a subset of node operators such that they may find it desirable to change the existing rules or create/support a fork with rules more conducive to their interests. Such actions will be frustrated by the high costs of identifying the likely disaffected operators and co-ordinating the requisite change. If no one operator once having joined a system can co-ordinate a defection at low cost, the incentives to join in the first instance, and to invest in developing the requisite code for a fork are likely low. In these circumstances, it may be feasible to instigate a new club only if, in the first instance, the rules serve to lower the costs of changing them when circumstances indicate. This could be the case in the early days of developing the business case for a DLS to be used for a specific application or in a specific industry (e.g. identity verification or supply chain management). However, it is not axiomatic that this state will prevail indefinitely. Generally, the more mature is the DL application, the more widely used it is, and the more diverse are the interests of ts user groups, the more costly it will be to gain a consensus on changes to the non-software-mediated governance rules, and the more stable will be the DLS.

We note that, by analogy, the internet did not originate as the public, open entity governed by the cultures, norms and formal arrangements currently prevailing. Rather, it began as a closed, permissioned entity with substantial restrictions placed by its governance arrangements in its users. It was initially a network of peers in government and academia with very narrow, homogeneous research interests in network technology development, beginning in 1969. The governance arrangements expanded gradually to include users with broader, more general research-oriented interests with their own network resources that specifically excluded commercial network operators. While users may have utilised PSTN connections to make their connections, the public telephone companies were unable to participate meaningfully in the internet ecosystem until changes in the governance arrangements in 1995 enabled wide private sector participation (Leiner, Cerf, Clark, Kahn, Kleinrock, Lynch, Postel, Roberts and Wolff, 2009). Over time, as more and more user groups were added, and users became more and more heterogeneous, the changes in governance arrangements became less and less frequent as the costs of co-ordinating their interests in order to institute changes became greater. Substantive changes now occur very rarely indeed, via costly consensus-seeking international processes organised by entities such as the International Telecommunications Union and the Internet Corporation for Assigned Names and Numbers.

Furthermore, as DLs exist to serve applications for communities of specific interest, the extent to which the environments in which they operate are stable or volatile, and hence the costs of co-ordination will vary depending on a wide range of context-specific factors. In part, this explains why, despite the existence of several thousand cryptocurrencies, only a handful are operating at a meaningful scale. Unless nearly all operators are equally affected by the exogenous change in circumstances or an internally-agreed rule change, then even if the high co-ordination costs could be overcome, the proportion of defections to a forked variant will be small and it will

be unlikely to appeal to a significant number of end-users. The greater is the choice of nodes available to an end-user to interact with the original DLS, the lower are the incentives for a single node to defect, unless the choice of end-users to patronise other nodes is also restricted. It now matters how end-users' interaction with the DL is mediated. If their choice of mediating node is restricted to a limited number of operators, then the costs of co-ordinating the migration of end-users' future use from the original DLS to the forked version are substantially reduced compared to the alternative of multiple (or 'free') node choices.

3 Developing an inquiry framework

To analyse a DLS as a club, it is first necessary to identify its membership and the rules governing its operation and governance. As with any club, there may be many different states of membership, defined in the rules. The rules will specify their relationships with each other, along with the various powers each class of member may exercise in both regular operation and in club governance.

3.1 Membership

The fundamental technical entities in any DLS are the nodes on which the DL copies are stored. Each node is managed by human actor, which may be either a real (unique human) or a legal (corporate) person. The human actor makes the decision to join the club, and in doing so agrees to abide by the club rules. Human node operators will subsequently be termed node-members of the DLS club.

Upon joining, node operators must acquire the current version of the DLS operating software from the club's software bank or find equivalent and indistinguishable (from the point of the network) software elsewhere. The software bank is managed by a sub-committee of members, who may or may not fulfil other roles within the club at the point of time the analysis is being undertaken. However, as the origin of any DLS relies upon the development of the relevant software, and no nodes can join the until the software has been 'released' for production, most DLS clubs will begin with a small number of members all of whom have strong stakes in the software development. Members who participate in the development and maintenance of software alone will be termed software-members. At the origin of a DLS, the club is most likely comprised almost exclusively of software-members. Over time, however, as more node-members join, the proportions will change. Software-members play a vital role as they have the knowledge and skills necessary to evaluate the effectiveness of the existing software and implement changes to it – such as those necessary to generate a fork. Members of the software subcommittee therefore exert significant power (control) over the software content and hence the likelihood of forking or changes to the existing software occurring.

Node-members are typically remunerated for holding ledger copies and processing transactions via a combination of payments from the DLS (in redeemable tokens - system currency) upon becoming the 'winner' who first posts the ultimately-agreed block, and from the entities who requested the transaction in the first place. They have strong vested interests in both the system-generated rules for token payments and any other agreements about the setting and collection of transaction fees paid by those using the system.

The development of the software for a DLS and its promotion, are not costless. Software-members may contribute to software developments without being paid, but nonetheless, they face an opportunity cost for the time invested. In effect, they donate that time (albeit that they may expect to be rewarded subsequently via returns from DLS operation as node operators, transaction generators or end-users). However, the DLS may have members who support the club's activities with explicit financial contributions. Members acting in this capacity will subsequently be termed donor-members. The club nature of a DLS is distinguished from a proprietary firm by the fact that these donor-members are not shareholders. They have no defined claim on either the club's assets or any profits made from operating (though of course, like software-members they may obtain benefits in other capacities of interaction in the system).

For a DLS to be operational, it requires two other classes of member. These are

- end user-members, who wish to use the system to undertake transactions with other end users or request information held in the DL, and
- transaction-members, who manage the interfaces via which end-users participate.

Transaction members may hold copies of the DL content, to facilitate raising transactions and answering queries. However, they do not participate in processing the transactions, which is undertaken by the node-members.

In some cases, the transaction-members may also be end-users, interacting with the DL for their own purposes. These users are typically dependent upon using software and/or code provided by the DLS in order to generate transactions to it or queries on it. However, they do not exercise any rights in the development of that software/code, unless they also participate separately as software-members. They must take the code 'as given'. An example is a cryptocurrency wallet, managed by an individual end-user.

In other cases, transaction-members may undertake a vast range of activities separate and distinct from the DL, with a vast range of end-users, as well as posting transactions for node-members to process and queries to be responded to. These transaction-members may generate their own bespoke code and applications (e.g. web pages) that build on code supplied by the DLS, but once again, unless they engage separately as software-members, they exert no influence on the DLS code per se. An example is a currency exchange, which may interact with many different cryptocurrency DLs in addition to banks handing traditional fiat currencies and payment mechanisms. However, to the extent that transaction-members have access to the DLS code, and have the capacity to understand and modify it, they provide an important discipline on the DLS because of their potential to create a fork.

Transaction-members are typically remunerated in fees paid to them by end-users. These may (but do not need to be) determined by club rule processes. However, transaction-members must pay fees to node operators when transactions are successfully completed. These may be encoded within the DLS and paid using system tokens, or 'off-system' via club rules or other 'private' contracts agreed between members. Club rules can be used to outlaw the latter agreements, but enforcement is contingent upon the ability to detect their existence and use. Their primary governance concerns will pertain to the level of, and rules setting, these fees, and rules concerning how they must relate with equivalent and adjacent club members - that is, other transaction members, node-members and end user-members.

End user-members are those who participate in the DLS only to the extent that they are originators or beneficiaries of transactions, or they make inquiries on the ledger. They are the equivalent of

customers of a shareholder-owned firm. They will pay fees to transaction-members for services requested. They may pay these in stocks of system tokens via processes encoded in the DLS, but equally, in other currencies, via arrangements that need not be agreed by or encoded in the DLS. End user-member interests in DLS governance will pertain largely to the rules via which these fees are determined. As for transaction-members, their primary governance concerns will pertain to how they must relate with equivalent and adjacent club members - that is, other user-members and transaction-members.

3.2 Governance

As identified above, a single club member may interact with the DL in many different capacities. The potential overlaps are illustrated in Figure 1. In the early stages of the DLS life cycle, and especially during its development, all roles (except perhaps donor-membership) may overlap, as in the hatched portion of Figure 1. However, as the DLS matures, and especially as it increase its scale of operations, the roles would be expected to gradually separate out (i.e. specialisation, as per Williamson, 1986 emerges). The nature of the separation will now be governed by the rules implicitly embedded in the DLS software and explicitly articulated in its "offline" rules.

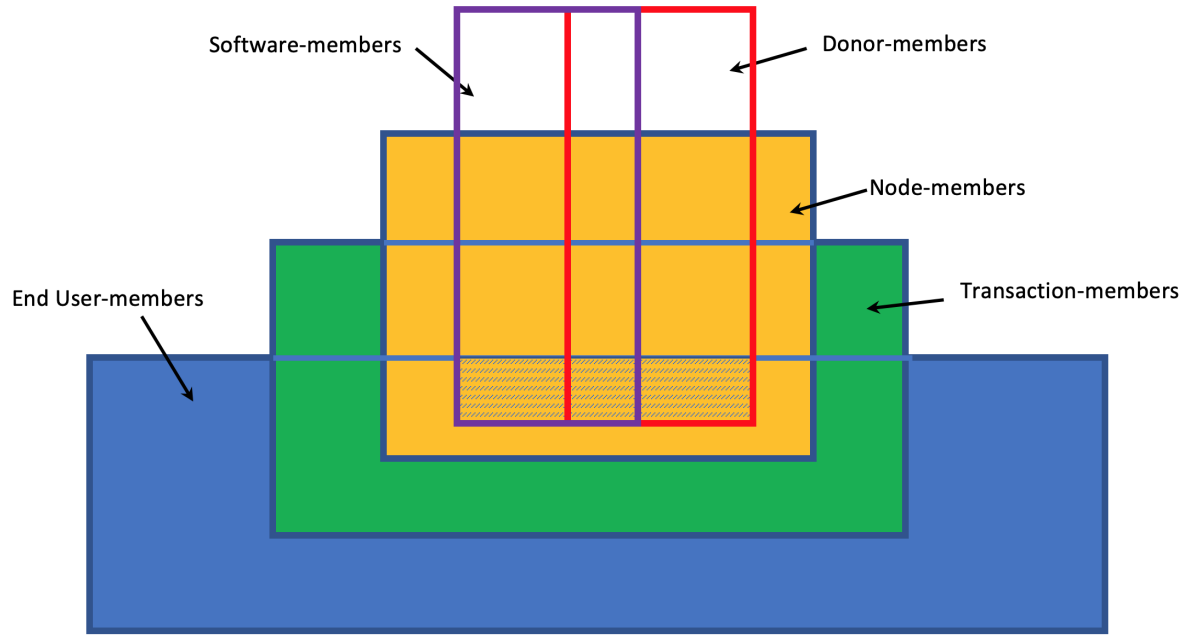

Figure 1: Membership Status

3.2.1 Control

Broadly speaking, Figure 1 identifies a hierarchy in membership status for mature systems. The higher-up in the hierarchy a member sits, the more power potentially is conferred in decision-making in the governance arrangements. End-user members and transaction-members can exert very little formal power via the governance and decision-making processes, as they must 'take as given' the package offered by node members. Their power is confined to 'voting with their

feet' and either choosing not to patronise the DLS, or (to the extent possible given the costs), co-ordinating a successful fork.

The costs of organising a successful fork depend on the extent to which the disaffected transaction-members can ensure that if they leave, end-users will follow them and not defect to transaction-members remaining on the original DLS. This is largely a matter of the design of the contractual relationships between transaction members and end-users. If these allow a transaction-member to limit the extent to which an end-user can patronise other transaction-members, then the costs of co-ordinating to organise a fork will be lower. On the one hand, DLS designers may not want to place many restrictions on these relationships, as reducing the likelihood of forking reinforces system stability. While the power of members higher up the membership hierarchy in Figure 1 is reinforced, it will rarely need to be exercised to change the software and/or other governance rules. On the other hand, as discussed in the preceding theory, if change is anticipated, then it may be necessary to co-ordinate the actions of all members in order to change key elements of the DLS rules (software and other rules) without exposing the DLS to undue risks of forking.

Node-members are pivotal, as the DLS cannot operate without them, but equally, they too may have little choice but to accept a 'take it or leave it' package offered by the founder-members. Once again, they can opt not to join in the first place, or like transaction members, co-ordinate to instigate a successful fork. However, to the extent that they are formally engaged in the process of DLS governance outside of the software channels, they can work constructively with the software and donor members to change the rules in a manner that ensures their ongoing patronage.

By either custom or explicit design, therefore, DLS governance is effectively controlled by a small coalition of software-members, who may also participate as node-members or be closely affiliated with influential node-members (i.e. they form the club committee). In order to motivate their participation, it would be expected they anticipate remuneration from either their node operation activities, or some other arrangement such as an honorarium paid from DLS funds held off the ledger – for example, financial or in-kind contributions (e.g. time, computing resources) made by donor-members. Donor-members without other membership stakes are unlikely to make substantial contributions of this kind unless they too exercise some influence over DLS governance and management – for example, having some powers to appoint or veto candidates to the club committee, or specifying in advance how their donations are to be managed and/or applied – in the same manner as expected by donors to clubs.

Thus, it is more likely that formal articulation of DLS governance arrangements outside of the software itself (e.g. formal agreement of rules, club or trust agreements, etc.) will be necessary the more donor-members there are, and the greater is their contribution of resources towards DLS operation. Sponsorship of these formal arrangements may arise in the event that a group of donor-members form a club to establish a new DLS for a specific purpose (e.g. to serve a trade organisation or similar). Formal governance arrangements may also be necessary for a group of informally-organised software-members wishing to make use of existing entities (e.g. firms, trade organisations) to take an embryonic DLS from test-state to production.

3.2.2 Rule and relationship formalisation

When a DLS is new and small, all members have homogeneous interests, and all are known to each other (i.e. all participate in the club in the same manner, as illustrated by the memberships intersecting in the hatched area of Figure 1), then the need for formal rules articulating the relationships between member groups and how conflicts will be resolved are less necessary. However, as it grows, role specialisation increases and member interests begin to diverge, then rule formalisation becomes more likely to be important. In particular, the allocation of important decision-making powers, processes of appointments to decision-making bodies, the relationships between different member categories and expectations and obligations of the relevant members should be made explicit in order to allow members to make appropriate decisions and enabling them to expect consistent predictable outcomes when interacting with the club.

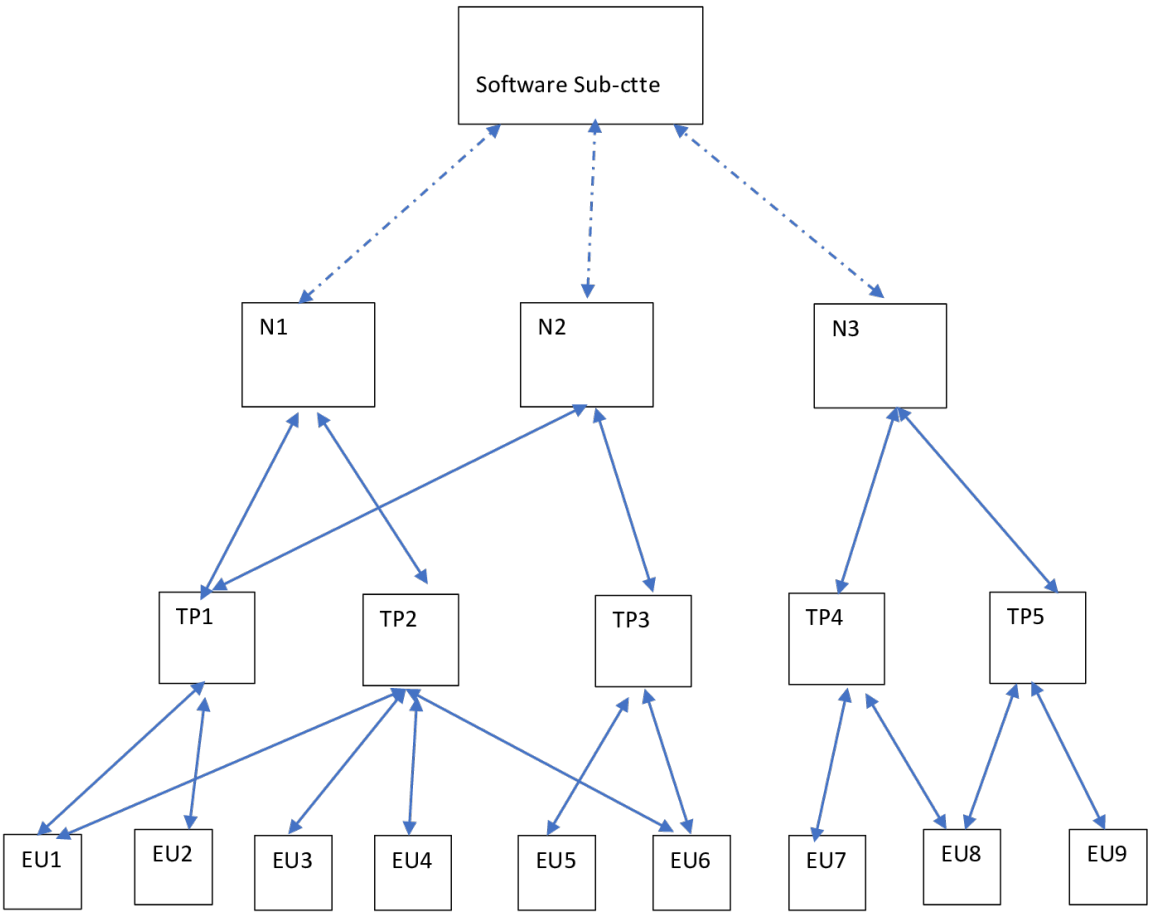

Figure 2: Transacting Relationships

Figure 2 illustrates some potential patterns in relationships between different classes of members in a hypothetical DLS. This can be used to illustrate how different restrictions placed upon the interrelationships of club members affect costs of co-ordination.

For example, node N3 operates in a closed environment with a limited number of transaction-members (TP4 and TP5) who interact with no other node operator. Furthermore, the transaction-members interact with a limited number of end-users (EU7, EU8 and EU9), who do not interact with any other transaction-members who do not operate through node N3. This arrangement could be achieved by having rules restricting interactions to a closed subset of members. That is,

N3 will only accept transactions from transaction members known to or recognised by it, and these members are precluded by software-mediated rules from interacting via any other node-member. Similar obligations can attend the interactions of end user-members with transaction members. In this example, EU8 can interact with any transaction member affiliated with N3 (TP4 or TP5), but EU7 and EU9 may be limited to interacting with TP4 and TP5 respectively.

The N3 limb of Figure 2 is an example of a "permissioned" DLS - each member needs the 'permission' of one higher up the membership tree to interact with the DLS. As these arrangements prevent 'client' transaction- and end user-members from interacting via any other node, considerable power is vested in N3. If it instigates a fork, then it can be sure of maintaining its existing transaction volume at negligible cost. The higher are a node-member's investments in the DLS and its operation, and the greater the share of its remuneration it gets from fees paid by transaction members, as opposed to the DLS, the more likely it is that a node-member will prefer to use the governance rules to restrict transaction-member and end user member choices. If all node operators are comparatively homogeneous in their identities and operations, and the proportion of their remuneration received from payments agreed 'off system' with downstream affiliates (rather than the internal DLS payments), then the more likely it is that a strictly hierarchical system will emerge. Even though the ledger and software are decentralised, each node will operate as the principal of its own federated 'sub-branch' of the DLS club. A commercial analogy is franchisees operating with exclusive territories. As with the franchise system, inducing participation by the node operators is contingent upon these protections. However, unlike franchise systems, the node operator can relatively costlessly exit, taking existing systems and customers along. If the club is to attract node operators in the first place, and remain stable into the future, those members must be protected from the effects of competition emerging from forking ex-members. The governance rules must contain provisions that make forking costly (e.g. very large membership fees, forfeited on forking).

By contrast, nodes N1 and N2 in Figure 2 can interact with any of TP1, TP2 and TP3. It is an example of a "permissionless' or fully public system. If N2 forks, TP1 and TP3 can shift their interaction to N1. Fewer (or no) interaction restrictions lower the likelihood of forking and therefore the risks of joining for a new node. There is less need for governance arrangements to constrain member defection by forking. Indeed, such a system may be able to operate without any special rules governing interaction. Competition within and between members may be satisfactory.

However, in both cases, the more heterogeneous are the node members, the more likely it is that a 'one size fits all' set of rules (especially for remuneration) will be optimal for all member types. Tensions between members are more likely to arise in these circumstances. However, unless members are identifiable and known to each other, and formal channels established for resolving disputes, the costs of achieving a satisfactory resolution are likely so large as to be prohibitive. Change is unlikely to occur, either to the existing rules or via forking. To the extent that these problems can be anticipated, cost-reducing dispute resolution mechanisms may be contained within the DLS governance provisions. If they are not, then arrangements external to it may also facilitate co-ordination - for example, if specific member groups are affiliated in other ways, such as by being members of an industry association. Knowledge of such potentials may alter the strategies by which a specific DLS may seek to expand - for example, by engaging with the aggregating entry directly, or seeking to include it as a member, and thereby relying on its resources to assist in dispute resolution. In this case, it may not be necessary for the DLS to have direct knowledge of the identity of, or direct communication with, individual club

members. Nonetheless, it is noted that the outcomes of such co-ordination may not be aligned with preserving the viability of the DLS unless its governance rules contain means of ensuring the aggregate members are required to prioritise this outcome.

4 Case study: Bitcoin

The Bitcoin blockchain is a distributed ledger (DL) which is used to record transactions in the Bitcoin cryptocurrency. In this paper, however, we consider only the mechanism by which new blocks are added to the ledger rather than the operation of the cryptocurrency which is well described elsewhere, for examply by Böhme, Christin, Edelman and Moore (2015).

4.1 Decription of the Bitcoin protocol

On the most basic level, the Bitcoin "network" consists of a large number of entirely independent computers that exchange messages conforming to certain specifications using the same protocol and each with a copy of the Bitcoin DL. The Internet protocol (IP) addresses of some key servers are published on authoritative websites and it is free to join. These servers can store and distribute the addresses of other servers on the network.

Each server (or, node) can check whether its version of the DL corresponds to those stored by others (up to a certain number of blocks) on the peer-to-peer network but this is really most easily done buy consulting some authoritative website. The basic function of a node is to copy the DL but it can also submit transactions for possible inclusion in the chain using an identity based on a randomly generated public-private key pair. Before considering which transactions (which are actually simply messages in a specific format) are included in the DL, we have to consider the integrity of the system as a whole.

Suppose, as a thought experiment, the Internet were suddenly split into two fully functioning parts. For example, by a single large country detaching itself from the global network. Bitcoin nodes on the detached part of the network might be unable to find some of the servers with IP addresses published on the authoritative websites (if these were available) but as long as some of the authoritative servers are based in the detached portion of the Internet, Bitcoin nodes in the rump would continue to functions as normal. The same would be true for the other portion of the Internet and the two versions of the Bitcoin DL would simply grow differently. For the paranoid, in short, there is no way of knowing that they are operating on the "true" DL.

Nevertheless, Bitcoin has proven quite successful as a payment system and has maintained its integrity and support remarkably well. The main reason for this is the ingenuous design of its proof-of-work system for generating new blocks for inclusion in the ledger.

4.2 Adding new blocks

The work is done by "miners" which are nodes on the network that generate candidates for new blocks. Each of these candidates must contain valid transactions (moderately easily checked by other nodes) and the solution (very easily checked) to a mathematical problems that necessarily involves generating a lot of random candidate solutions, on average. The solution is included in

the candidate block broadcast to the network, as is a transaction that includes awarding a bounty to the miner.

As nodes receive valid candidate blocks from miners, they accept them, add them to their copy of the ledger and rebroadcast them. Subsets of the nodes can at this stage receive and accept different new blocks. This is a dilemma but one that is completely resolved (usually within about an hour) by the nature of the Bitcoin protocol. Whichever new block is accepted by more nodes will tend to be the block that is accepted by miners and that they use to build subsequent blocks. This is entirely consistent with miners' self-interest – they would have no incentive to mine on chains that are likely to be abandoned. With the operation of this majoritarian mechanism, nodes that have accepted a less favoured block will eventually find that the chain is a dead-end and will revert to the surviving chains. This mechanism delivers a consensus that is driven entirely by the self-interest of all the parties and is subject to no prior explicit arrangement.

This majoritarian system of vetting new blocks[1] is the source of fear of the so-called "51% attack" which would consists of putative malicious control of a majority of the mining capacity and the possible introduction of improper blocks (for example, containing invalid transaction messages) that are then included in the DL. Since all nodes are able to relative inexpensively check the validity of blocks, this is extremely unlikely to go unnoticed for very long but it is likely to cause a great deal of confusion and distrust. Nevertheless, since the miners are awarded in Bitcoin, they probably have very little incentive to engage in behaviour that reduces trust in the underlying cryptocurrency.

The governance of the Blockchain ledger is therefore mechanically implied by the protocol, which is the genial invention that engenders great robustness and stability. Nevertheless, this does not exclude the use of explicit agreements among participating nodes (or, indeed natural or juristic persons). Should, for example, a 51% attack introduce and invalid block, there is nothing preventing a large number of stakeholders to agree to make a certain change to the ledger and to restart from that point onward and in a specific way. This could however be costly and disruptive because of the ongoing demand for the DL to record the processing of payments.

4.3 Forking the chain

It has happened on several occasions that a sufficiently large section of Bitcoin users managed to agree to change the protocol that they at a certain point started following new rules ("forking" the blockchain at that point) and that this change has been sufficiently sustainable. Bitcash is one example. At the point of creating Bitcash, anyone with (say) 2.3 Bitcoin would retain that amount but would also have 2.3 units of Bitcash as well, attributed to the same public key identity. Such a fork does not require more than for a viable number of participants to (agree to) do it. In the thought experiment above the splitting of the Internet in two, the fork would have been involuntary.

In August 2010, a notable fork to correct a technical error, took place. A block had been mined that created 184 467 440 737.09551616 units of Bitcoin (van Wirdum, 2016) and sent them to two addresses. The number is remarkable since the Bitcoin protocol only allows for a total of 21 million Bitcoin to ever exist. A bulletin board message on 15 August from "Satoshi Nakamoto" warned

[1] New blocks and who mines them can be observed directly at https://www.blockchain.com/explorer.

> *** WARNING *** We are investigating a problem. DO NOT TRUST ANY TRANSACTIONS THAT HAPPENED AFTER 15.08.2010 17:05 UTC (block 74638) until the issue is resolved.

and the error was reversed by a software update within a few hours. This was the most serious protocol or software error in the history of Bitcoin and it happened when the DL was only two years old. A similar breakdown today would hardly be tolerable in view of the number of transactions per day.

4.4 Informal and formal governance

In addition to the large miners, "a small core of highly skilled developers" (De Filippi and Loveluck, 2016) for Bitcoin Core, the most widely used Bitcoin client, have an outsize influence in practice on the arrangements for this DL. The software was initially published by Satoshi Nakamoto (pseudonym) who also released the Bitcoin founding whitepaper. Development of Bitcoin Core has been funded by MIT Media Lab and others (van Wirdum, 2016). Given the practical dominance of Bitcoin Core software, it would not be entirely out of place to view governance of Bitcoin as identical to the governance of the software project, as a first-order approximation.

This is nevertheless very informal and unstructured, even anarchic, since there is still absolute freedom to fork the open source software project (at the same time as the chain). It would not be incorrect to say that there is no formal governance arrangement for Bitcoin.

4.5 Applying the analytical framework to Bitcoin

Applying Figure 1 to Bitcoin, the following membership stakes are identified:

- software-members are an unidentified person/group of people acting under the soubriquet "Satoshi Nakamoto". It would appear that this group exercises ultimate control over Bitcoin governance;
- MIT Media Lab was an original donor-member, but it is not known whether it or any other unidentified funders continue to contribute actively to Bitcoin governance;
- node members are the miners, who can freely enter and exit of their own accord. There is no explicit relationship between them and any other members;
- transaction members and end user members can freely enter and exit of their own accord. There are no explicit requirements governing their interactions. One entity may participate as all of end user, transaction- and node-members. Overlaps almost certainly occur, given the majority of issued bitcoin are held by small number of system participants

Applying Figure 2, we conclude that in the absence of any apparent rules to the contrary, Bitcoin is a fully-public DLS with no explicit or externally-articulated governance arrangements. All applicable rules are embedded in and executed by the software. Changing the rules is extremely costly - as evidenced by the perpetuation and growth of the DLS despite the absence of substantive changes to the software-based rules since the August 2010 fork. Given the large number of nodes and the comparative inability of member interests to organise successful rule changes or forks where large numbers of members defect, it would seem that Bitcoin members

are heterogeneous, and lack 'off-system' means of cost-reducing co-ordination. The inherent anonymity of Bitcoin members also militates against such actions.

That said, we note that the Bitcoin DLS is underpinned by a relatively simple and reasonably well-understood financial transacting business case. While the distributed ledger component of the system is is novel, the payments processing function is not. It is much easier at the outset to identify the governance requirements for a stable, well-understood system where change in the business model is unlikely to occur. Arguably, the Bitcoin DLS has been remarkably stable because these characteristics have meant that the circumstances of tensions arising between different members or class of members have simply not come about, since 2010 at least. And that the changes made in 2010 were successful was likely in large part attributable to the fact that at that stage, membership was small, much more homogeneous and more likely to be comprised of people known to each other (or at least, a sufficiently large-enough coalition was well-enough known to each other to co-ordinate the forking at a comparatively lower price. To the extent that forking has occurred to start new currencies, it s likely that this has been steered by software=-skilled members, iof not software-members per se.That none of the forked currencies has grown to rival Bitcoin simply serves to reinforce the dominance of the existing arrangements.

5 Case study: Sovrin

Sovrin is a "global public, permissioned identity utility for exchanging identity more securely" (Patel, 2018) based on a distributed ledger overseen by the Sovrin Foundation. It is based on opensource blockchain software and trusted participants that issue and verify identities and other identifying pertinent information about natural and juristic persons. The main aim of the project is to facilitate the reuse of verified information (with the permission of the data subject), to incentivise the release of information and to record the withdrawal of the right to use such information (Ldapwiki, 2018).

5.1 Description of the Sovrin network

Figure 3 shows the Sovrin Goverance Network in which the Sovrin Governance Framework Master Document defines the "constitution" of the Sovrin Network laying down the purpose, core principles and links to other main documents. The Sovrin organisation is formally constituted as a nonprofit organisation incorporated in Utah, USA on February 2 2018. Its purposes include, but are not limited to

(a) To develop, govern and promote an international nonprofit private sector self-sovereign digital identity system based on the Sovrin distributed ledger;
(b) To own, lease, sell, exchange or otherwise deal with all property, real and personal, tangible or intangible, to be used in furtherance of these purposes; and
(c) To engage in any and all lawful activities incidental, useful or necessary to the accomplishment of the above-referenced purposes.[2] While it has no legally-defined members, the term "members" is used "to refer to donors, technology contributors, ledger stewards,

[2]Sovrin Articles of Incorporation, February 2, 2018. https://drive.google.com/file/d/1QC7Ma9DZUiOjY3G4S1URLXD2CBJzvxqw/view

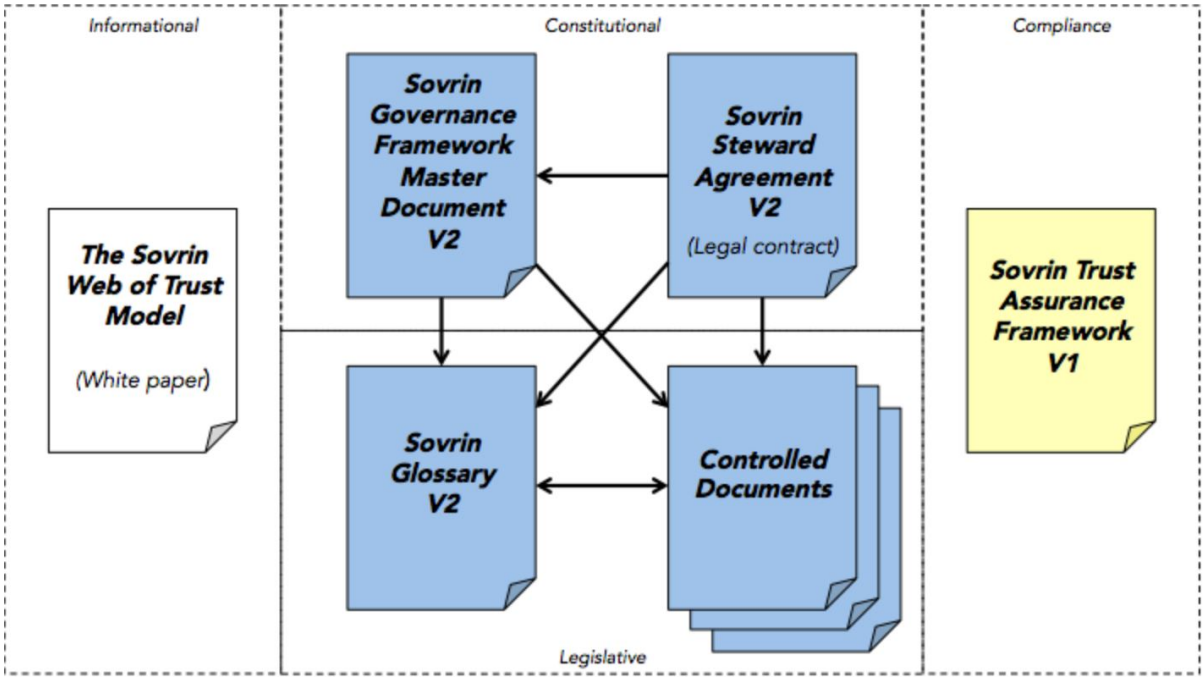

Figure 3: The Sovrin Governance Network

members of Corporation committees or work groups, and other participants in the Sovrin community whose roles may be further defined in the Bylaws, agreements, or other governing documents".[3] Its principal office and mailing address is identified as 151 S 1150 E, Lindon, UT 84042.

The Sovrin Foundation is overseen by a Board of Trustees with no less than three and no more than twenty one members. The original Trustees are identified as Phillip J Windley and Jason A Law, both of Utah, and Drummond S Reed of Washington State. In mid January 2019, it comprised 12 members. A nominations committee selected by the Board identifies eligible nominees, who are elected annually at the Annual Meeting. Trustees are not remunerated, but are (subject to approval by the Board), reimbursed for expenses incurred on behalf of the organisation. The Board appoints an Executive Director to supervise operations. The Chief Executive Officer Heather Dahl is the Executive Director. Along with the CEO, the Chief Financial Officer Roy Avondet, Chief Technology Officer Nathan George and Director of Marketing Helen Garneau comprise the Executive Leadership. Seventeen other staff are identified on the website. The Board also has the power to create Advisory Councils and other committees as required, in addition to the Executive Committee and Finance Committee identified in the Bylaws. In mid-January 2019, a fifteen-member Technical Governance Board, including CTO Nathan George and founder-trustee Jason Law is identified as having been appointed to govern the "technical design, architecture, implementation and operation of the Sovrin Network as a global public utility for self-sovereign identity".

Day-to-day activities are managed by an executive team comprising CEO Heather Dahl, Chief Financial Officer Roy Avondet, Chief Technology Officer Nathan George and Director of Marketing Helen Garneau. CEO Heather Dahl is one of the twelve Board Members. Seventeen staff are identified on the website. The Board

[3]Sovrin Bylaws, Jan 31 2018 https://drive.google.com/file/d/1kkuiEp0vA620ydcAND9pIY_hLHVjHKsG/view

Central to the mode of governance within the Sovrin network is the Sovrin Steward Agreement document specifying the legal obligations, liabilities, etc. for Stewards and the Sovrin Foundation[4]. The agreement, governed by the law of the State of Delaware,contains explicit provisions to be followed in the event of a dispute between the Stewards and Sovrin.

The trusted participants acting as node members in the project are called Stewards. In mid January 2019, forty-eight are identified on the Sovrin website.[5] They include banks, telecommunications companies and universities as well as IT companies such as Cisco and IBM. On the Sovrin network, it is these Stewards that approve transactions for inclusion in the ledger and that, in fact, submit transactions for inclusion (Tobin, 2018). There is a strong emphasis on anonymous identity in ledger records (as actual users can create unique reference numbers for each relationship) as well as zero-knowledge proofs where a user can, for example, use the system to prove that they are over 18 without revealing their actual age – based on the trusted information embedded in the system. This feature is unique to the Sovrin Identity Network (SIDN) which allows to create self-sovereign identity (SSI) for end user members (Muhle et. al 2018).

The software used has been part of The Linux Foundation's *Hyperledger* project since 2017 under the name Hyperledger Indy.[6] Like the Bitcoin software, it is opensource. Sovrin facilitates engagement of individuals in the Hyperledger Indy project, including direct links on its website to the weekly Indy Group calls, Chat room and Mailing List (https://sovrin.org/developers/).

No sensitive data is stored in the DL at all – only the Stewards' identifying information as well as pointers to the end-users' data are included in the ledger. Stewards act as validators (similar to Bitcoin miners) as well as clients (who submit transactions).

5.2 Adding to the ledger

The validation of information submitted to the Sovrin network is entirely the work of the Stewards. This is strikingly dissimilar to the case of Bitcoin where anyone may start mining. One cannot be a Steward without formally entering into an agreement with the Foundation. The validity of transaction is only vouched for by other Stewards and cannot necessarily (as in the case of Bitcoin) be easily checked by anyone with access to the ledger.

5.3 Forking the chain

It is definitely technically possible for a subset of Stewards to agree to defect with a current copy of the ledger but this act itself would deprive them of the governance arrangements embodied by the Sovrin Foundation.

The consensus algorithm in Sovrin is called *plenum*, an enhancement of the redundant byzantine fault tolerance algorithm.[7] In most general terms, this is a voting algorithm that executes very quickly and resolves faults possibly introduced by errant nodes.

[4]https://sovrin.org/library/steward-agreement/
[5]https://sovrin.org/stewards/
[6]https://github.com/hyperledger/indy-sdk/blob/rc/doc/getting-started/getting-started.md
[7]https://github.com/hyperledger/indy-plenum/wiki

5.4 Informal and formal governance

Essentially all governance in the case of Sovrin is formal. There is however a certain devolution of power as regards the Stewards' dealing with individuals.

5.5 Applying the analytical framework to Bitcoin

Applying Figures 1 and 2 to Sovrin, the following membership stakes and interactions are identified:

- The three founder-trustees appear to have acted as initial donor-members. At least one of these has a technical background, so appears to have been an original software-member. As Sovrin has expanded, the Technical Governance Board appears to have assumed responsibility for software oversight. This includes "reviewing the technical qualifications of the Steward candidates to ensure they meet the requirements and principles of the Sovrin Trust Framework" - suggesting a degree of central control over not just the blockchain application but also the applications used by node-member stewards in their engagement with transaction-members and end user-members (the Sovrin Steward Agreement 3b obliges the Stewards to "only run software code that has been approved by the Sovrin Foundation as referenced in the Sovrin Governance Framework").
- Stewards as node-members are central. Entry is strictly controlled by Sovrin, and steward identities are known to all other members. Only stewards can raise transactions on the ledger (although other entities may read data from it), so they fulfil the role of transaction-members as well. The reliability of the identification system relies upon the fact that the stewards themselves can be trusted by other stewards and users of identity information. The steward agreement does not mention any membership payments, but there is likely some considerable expense involved in satisfying the Foundation that admission as a steward will not bring undue risk to the DLS. The more arduous are these processes, the more costly they are for the firms concerned. Only firms who are serious about belonging will put in the effort; if they walk away from the arrangements (i.e. instigate a fork) this cost is truly sunk. This means the number of nodes is likely to be much smaller than a public system like Bitcoin, but it also means that the stewards are both known and can be pursued for any costs caused by seceding.
- End user-members may have many different identities on the ledger, and be effectively anonymous on it, but each has to raise identity transactions via a Steward who is prepared to vouch for their identity when posting to the ledger. Hence end user-members cannot be anonymous in respect of at least that Steward. The knowledge comes from interactions the stewards have with individuals in other ways, for example as customers of a financial institution or telecommunications provider. Stewards thus provide the bedrock of trust in the identities lodged on the DLS. User-member participation will not be possible without an of-system interaction with at least one steward.

The Sovrin system is in a very early stage of development. It is not clear yet how payments for services undertaken will be made. However, Sovrin does embody a token system to reward end-user and steward/node participation.

In contrast to Bitcoin, Sovrin is a new application with a business case that is yet to be fully

proven in any context - either physical or virtual. It is clear that the current Sovrin Framework is a 'work in progress', though one which has moved from the truly novel innovation to small-scale testing phase. Arguably, the stewards who have joined so far are participating as a means of learning how they can make use of the system as it scales up rather than to achieve an already clearly-understood outcome. It is not clear at this point what the future might bring for this venture. However, from our analysis, the much tighter control exerted from the centre, with clearly-articulated responsibilities, rules and disputes resolution processes binding all parties is consistent with a system where future developments are uncertain. The strict governance rules allow for greater flexibility in the direction that can be taken in incorporating rules into the software. While current abilities for Stewards to formally influence directions are less than for the software and donor members of the system, they exert significant commercial power as they mediate the relationships with end users. This close economic co-dependence between them and the Sovrin Foundation affords them a degree of influence in the governance of their DLS far greater than that of the Bitcoin miners over their DLS.

6 Discussion of the case studies

Ultimately, in the case of Bitcoin, the validity of the entire blockchain can be checked by any interested party. This includes that veracity of the solutions to the mining problems. It also includes checking that the cryptographic signatures of the individual transactions are valid. The only thing that is not possible to check is whether all other nodes might not have conspired to pick certain valid mine blocks rather than others but the Bitcoin protocol (involving free entry) makes it rather intuitively unlikely that this would take place.

It has been suggested by Vitalik Buterin, creator of Ethereum, that the degree of (de)centralisation of a network can be examined along three axes (Siriwardena, 2017).

1. Architecture (de)centralisation – what is the physical nature of the system and how robust is it?
2. Political (de)centralisation – how is membership of and participation in the system governed?
3. Logical (de)centralisation – how flexible are interfaces and data structures in the system?

Considering Bitcoin and Sovrin, we suggest that the decentralisation of the two systems can be categorised as follows.

Degree of centralisation	Bitcoin	Sovrin
Architectural	Low	Medium
Political	Low	High
Logical	High	High

Centralisation is linked to governance, a topic to be explored further in future work.

7 Conclusion

The authors have investigated distributed ledger (DL) governance in the context of the theory of clubs. The incentives to passively join and to take part in operating the consensus mechanism of the DL can be understood using this theory. The examples of Bitcoin and Sovrin illustrate how formal and informal arrangements operate – either through formal agreement or through technical arrangements embedded in the software.

We note that DL systems tend to be effectively controlled by a small coalition of software-members, who may also participate as node-members or be closely affiliated with influential node-members. In order to motivate their participation, it would be expected they anticipate remuneration from either their node operation activities, or some other arrangement such as an honorarium paid from contributions made by donor-members. The more donor members there are, the more likely it is that formal articulation of DLS governance arrangements outside of the software itself would be required.

References

Berg, Alastair, Berg, Chris, & Novak, Mikayla. 2018a. Blockchains and Constitutional Catallaxy. Available at SSRN 3295477.

Berg, Chris, Novak, Mikayla, Potts, Jason, & Thomas, Stuart J. 2018b. From Industry Associations to Ecosystem Associations: Blockchain, Interest Groups and Public Choice. Interest Groups and Public Choice (November 16, 2018).

Berle, Adolph, & Means, Gardiner. 1932. Private property and the modern corporation. New York: Mac-millan.

Buchanan, James M. 1962. Predictability: The criterion of monetary constitutions. In search of a monetary constitution, 155–83.

Buchanan, James M. 1965. An economic theory of clubs. Economica, 32(125), 1–14.

Buchanan, James M. 1987. The constitution of economic policy. The American economic review, 77(3), 243–250.

Böhme, Rainer, Christin, Nicolas, Edelman, Benjamin, & Moore, Tyler. 2015. Bitcoin: Economics, Technology, and Governance. Journal of Economic Perspectives, 29(2), 213–38. http://www.aeaweb.org/articles?id=10.1257/jep.29.2.213

Cordery, Carolyn, & Howell, Bronwyn. 2017. Ownership, Control, Agency And Residual Claims In Healthcare: Insights On Cooperatives And Non-profit Organizations. Annals of Public and Cooperative Economics, 88(3), 403–424.

Cornes, Richard, & Sandler, Todd. 1996. The theory of externalities, public goods, and club goods. Cambridge University Press.

Crosby, Michael, Nachiappan, Pattanayak, Pradhan, Verma, Sanjeev, & Kalyanaraman, Vignesh. 2015. Blockchain technology: beyond Bitcoin. Sutardja Center for Entrepreneurship & Technology. http://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf

Czepluch, Jacob Stenum, Lollike, Nikolaj Zangenberg, & Malone, Simon Oliver. 2015. The use of block chain technology in different application domains. The IT University of Copenhagen, Copenhagen.

De Filippi, Primavera, & Loveluck, Benjamin. 2016. The invisible politics of Bitcoin: governance crisis of a decentralised infrastructure. Internet Policy Review, 5(3).

Elinor, Ostrom. 1990. Governing the commons: the evolution of institutions for collective action.

Foroglou, George, & Tsilidou, Anna-Lali. 2015. Further applications of the blockchain. Columbia University PhD in Sustainable Development, 10.

Fukuyama, Francis. 2014. Political Order and Political Decay: From the Industrial Revolution to the Globalization of Democracy. New York: Farrar, Straus and Giroux, 455–466.

Hansmann, Henry. 1996. The changing roles of public, private, and nonprofit enterprise in education, health care, and other human services. Pages 245–276 of: Individual and social responsibility: Child care, education, medical care, and long-term care in America. University of Chicago Press.

Krecké, Elisabeth. 2004. 14 The emergence of private lawmaking on the Internet. Markets, Information and Communication: Austrian Perspectives on the Internet Economy, 289.

Ldapwiki. 2018. Sovrin. Retrieved on 2019-01-13. https://ldapwiki.com/wiki/Sovrin

Leiner, Barry M, Cerf, Vinton G, Clark, David D, Kahn, Robert E, Kleinrock, Leonard, Lynch, Daniel C, Postel, Jon, Roberts, Larry G, & Wolff, Stephen. 2009. A brief history of the Internet. ACM SIGCOMM Computer Communication Review, 39(5), 22–31.

Mattila, Juri. 2016. The blockchain phenomenon. Berkeley Roundtable of the International Economy.

Mazieres, David. 2015. The stellar consensus protocol: A federated model for internet-level consensus. Stellar Development Foundation.

Mulligan, CJ, Scott, Z, Warren, S, & Rangaswami, JP. 2018. Blockchain Beyond the Hype. In: World Economic Forum. http://www3.weforum.org/docs/48423_Whether_Blockchain_WP.pdf. Accessed, vol. 2.

Narayanan, Arvind, & Clark, Jeremy. 2017. Bitcoin's academic pedigree. Communications of the ACM, 60(12), 36–45.

Narayanan, Arvind, Bonneau, Joseph, Felten, Edward, Miller, Andrew, & Goldfeder, Steven. 2016. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press.

Olson, Mancur. 1989. Collective Action. London: Palgrave Macmillan UK. Pages 61–69. https://doi.org/10.1007/978-1-349-20313-0_5

Ostrom, Elinor. 2005. Understanding institutional diversity. Princeton University Press.

Ostrom, Elinor. 2010. Beyond markets and states: polycentric governance of complex economic systems. American economic review, 100(3), 641–72.

Ostrom, Vincent. 2014. Polycentrictiy: The Structural Basis of Self-Governing Systems. Choice, Rules and Collective Action: The Ostrom's on the Study of Institutions and Governance, 45.

Patel, Milan. 2018. IBM Blockchain Trusted Identity: Sovrin Steward closed beta offering. Retrieved on 2019-01-13. https://www.ibm.com/blogs/blockchain/2018/08/ibm-blockchain-trusted-identity-sovrin-steward-closed-beta-offering/

Reijers, Wessel, O'Brolcháin, Fiachra, & Haynes, Paul. 2016. Governance in blockchain technologies & social contract theories. Ledger, 1, 134–151.

Swan, Melanie. 2015. Blockchain: Blueprint for a new economy. O'Reilly Media, Inc.

Szabo, Nick. 1997. Formalizing and securing relationships on public networks. First Monday, 2(9).

Tarko, Vlad, Schlager, Edella, & Lutter, Mark. 2018. The Faustian Bargain: Power-Sharing, Constitutions, and the Practice of Polycentricity in Governance. Governing Complexity: Analyzing and Applying Polycentricity, eds. William A. Blomquist, Dustin Garrick and Andreas Thiel (Cambridge University Press, Forthcoming).

Tobin, Andrew. 2018. Sovrin: What Goes on the Ledger? Retrieved on 2019-01-11. https://sovrin.org/wp-content/uploads/2018/10/What-Goes-On-The-Ledger.pdf

van Wirdum, Aaron. 2016 (Apr). Who Funds Bitcoin Core Development? How the Industry Supports Bitcoin's 'Reference Client'. https://bitcoinmagazine.com/articles/who-funds-bitcoin-core-development-how-the-industry-supports-bitcoin-s-reference-client-1459967859/

Williamson, OE. 1985. 1985: The economic institutions of capitalism. Firms, markets, relational contracting. New York: Free Press.

Williamson, Oliver E. 1999. Strategy research: governance and competence perspectives. Strategic management journal, 20(12), 1087–1108.

Williamson, Oliver E. 2000. The new institutional economics: taking stock, looking ahead. Journal of economic literature, 38(3), 595–613.

Windley, Phillip J. 2016. How Sovrin Works. Retrieved on 2019-01-11. https://sovrin.org/wp-content/uploads/2018/03/How-Sovrin-Works.pdf