Towards a Design Philosophy for Interoperable Blockchain Systems

(Extended Abstract)

Thomas Hardjono Alexander Lipton Alex "Sandy" Pentland
MIT Connection Science

Massachusetts Institute of Technology
Cambridge, MA, USA

hardjono@mit.edu alexander.lipton@gmail.com sandy@media.mit.edu

July 30, 2018

Abstract

In this paper we discuss a design philosophy for interoperable blockchain systems, using the design philosophy of the Internet architecture as the basis to identify key design principles. We recast some of the challenges faced in the design of the Internet architecture to that of the design of an interoperable blockchain architecture. We emphasize interoperability as a crucial requirement for the survivability of blockchain autonomous systems.

Contents

1 Introduction **2**

2 The Design Philosophy of the Internet **3**
2.1 Fundamental Goals 3
2.2 The End-to-End Principle 4
2.3 The Autonomous Systems Paradigm 5

3 Interoperable Blockchains: Towards a Design Philosophy **7**
3.1 Survivability 8
3.2 Variety of service types 11
3.3 Variety of blockchain systems 12
3.4 Reachability 13
3.5 Interconnecting Values 14
3.6 Moveable Smart Contracts for Survivability 15
3.7 Cryptographic Survivability 15

4 Interoperability Design in Tradecoin **16**
4.1 The Tradecoin Autonomous System 17
4.2 The Role of Gateways 18
4.3 Reachability 20
4.4 Inter-Domain Transaction Mediation 20
4.5 Peering Agreements for Blockchain Systems 22

5 Discussion: Survivable Digital Currency and CBDC **22**

6 Conclusions **23**

1 Introduction

The goal of this paper is to bring to the forefront the notion of *interoperability* for blockchain systems, using lessons learned from the three decades of the development of the Internet. Our overall goal is to develop a design philosophy for an interoperable blockchain architecture, and to identify some design principles that promote interoperability.

Currently there is considerable interest (real and hype) in blockchain systems as a promising technology for the future infrastructure of a global value-exchange network – or what some refer to as the "Internet of value". The original blockchain idea of Haber and Stornetta [1, 2] is now a fundamental construct within most blockchain systems, starting with the Bitcoin system which first adopted the idea and deployed it in a digital currency context.

Many parallels have been made between blockchain systems and the Internet. However, many comparisons often fail to understand the fundamental goals of the Internet architecture as promoted and led by DARPA, and thus fail to fully appreciate how these goals have shaped the Internet to achieve its success as we see it today. There was a pressing need in the Cold War period of the 1960s and 1970s to develop a new communications network architecture that did not previously exist, one that would allow communications to survive in the face of attacks. In Section 2 we review and discuss these goals.

We argue in this paper that if blockchain technology seeks to be a fundamental component of the future global distributed network of commerce and value, then its architecture must also satisfy the same fundamental goals of the Internet architecture. In Section 3 we attempt to re-cast some of these fundamental goals of the Internet to the current context of blockchain technology. Among others, we look for some design principles for blockchain technology that should remain true across any blockchain system implementation.

This is followed by Section 4 in which we discuss the interoperability design aspects of the MIT Tradecoin project, mapping much of the Internet's constructs to that of the blockchain architecture. In Section 5 we briefly discuss the current emerging interest for central bank digital currencies, and pose questions regarding the survivability of those digital currencies.

2 The Design Philosophy of the Internet

In considering the future direction for blockchain systems generally, it is useful to recall and understand goals of the Internet architecture as defined in the early 1970s as a project funded by DARPA. The definition of the Internet as view in the late 1980s is the following: it is "a packet switched communications facility in which a number of distinguishable networks are connected together using packet switched communications processors called gateways which implement a store and forward packet-forwarding algorithm" [3, 4].

2.1 Fundamental Goals

It is important to remember that the design of the ARPANET and the Internet favored military values (e.g. survivability, flexibility, and high performance) over commercial goals (e.g. low cost, simplicity, or consumer appeal) [5], which in turn has affected how the Internet has evolved and has been used. This emphasis was understandable given the Cold War backdrop to the packet-switching discourse throughout the 1960s and 1970s.

The DARPA view at the time was that there are seven (7) goals of the Internet architecture, with the first three being fundamental to the design, and the remaining four being second level goals. The following are the fundamental goals of the Internet in the order of importance [3, 4]:

1. *Survivability*: Internet communications must continue despite loss of networks or gateways.

   This is the most important goal of the Internet, especially if it was to be the blueprint for military packet switched communications facilities. This meant that if two entities are communicating over the Internet, and some failure causes the Internet to be temporarily disrupted and reconfigured to reconstitute the service, then the entities communicating should be able to continue without having to reestablish or reset the high level state of their conversation. Therefore to achieve this goal, the state information which describes the on-going conversation must be protected. But more importantly, in practice this explicitly meant that it is acceptable to lose the state information associated with an entity if, at the same time, the entity itself is lost. This notion of state of conversation is related to the end-to-end principle discussed below.

2. *Variety of service types*: The Internet must support multiple types of communications service.

   What was meant by "multiple types" is that at the transport level the Internet architecture should support different types of services distinguished by differing requirements for speed, latency and reliability. Indeed it was this goal that resulted in the separation into two layers of the TCP layer and IP layer, and the use of bytes (not packets) at the TCP layer for flow control and acknowledgement.

3. *Variety of networks*: The Internet must accommodate a variety of networks.

   The Internet architecture must be able to incorporate and utilize a wide variety of network technologies, including military and commercial facilities.

The remaining four goals of the Internet architecture are: (4) distributed management of resources, (5) cost effectiveness, (6) ease of attaching hosts, and (7) accountability in resource usage. Over the ensuing three decades these second level goals have been addressed in different ways. For example, accountability in resource usage evolved from the use of rudimentary management information bases (MIB) [6, 7] into the current sophisticated traffic management protocols and tools. Cost effectiveness was always an important aspect of the business model for consumer ISPs and corporate networks.

2.2 The End-to-End Principle

One of the critical debates throughout the development of the Internet architecture in the 1980s was in regards to the placement of functions that dealt with reliability of message delivery (e.g. duplicate message detection, message sequencing, guaranteed message delivery, encryption). In essence the argument revolved around the amount of effort put into reliability measures within the data communication system, and was seen as an engineering trade-off based on performance. That is, how much low-level function (for reliability) needed to be implemented by the networks versus implementation by the applications at the endpoints.

The line of reasoning against low-level function implementation in the network became known as the *end-to-end argument* or principle. The basic argument is as follows: a lower level subsystem that supports a distributed application may be wasting its effort in providing a function that must be implemented at the application level anyway [8]. Thus, for example, for duplicate message suppression the task must be accomplished by the application itself seeing that the application is most knowledgeable as to how to detect its own duplicate messages.

Another case in point relates to data encryption. If encryption/decryption was to be performed by the network, then the network and its data tranmission systems must be trusted to securely manage the required encryption keys. Also, when data enters the network (to be encrypted there) the data will be in plaintext and therefore susceptible to theft and attacks. Finally, the recipient application of the encrypted messaged will still need to verify the source-authenticity of the message. The application will still need to perform key management. As such, the best place to perform data encryption/decryption is the application endpoints –

there is no need for the communication subsystem to provide for automatic encryption of all traffic. That is, encryption is an end-to-end function.

The end-to-end principle was a fundamental design principle of the security architecture of the Internet. Among others, it influenced the direction of the subsequent security features of the Internet, including the the development of the IP-security sublayer [9] and its attendant key management function [10, 11]. Today the entire Virtual Private Network (VPN) subsegment of the networking industry started based on this end-to-end principle. (The global VPN market alone is forecasted to reach 70 billion dollars in the next few years). The current day-to-day usage of the Secure Sockets Layer (TLS) [12] to protect HTTP web-traffic (i.e. browsers) is also built on the premise that client-server data encryption is an end-to-end function performed by the browser (client) and by the HTTP server.

2.3 The Autonomous Systems Paradigm

Another key concept in the development of the Internet is that of *autonomous systems* (or *routing domains*) as a connectivity unit that provide scale-up capabilities. The notion of autonomous systems provides a way to *hierarchically aggregate routing information*, such that the distribution of routing information itself becomes a manageable task. This division into domains provides independence for each domain owner/operator to employ the routing mechanisms of its choice. IP packet routing inside an autonomous system is therefore referred to as *intra-domain* routing, while routing between (across) autonomous systems is referred to as *inter-domain* routing. The common goal of many providers of routing services (consumer ISPs, backbone ISPs and participating corporations) is that of supporting different types of services (in the sense of speed, latency and reliability).

In the case of intra-domain routing the aim is to share best-route information among routers using an intra-domain routing protocol (e.g. distance vector such as RIP [13], or link-state such as OSPF [14]). The routing protocol of choice must address numerous issues, including possible loops and imbalances in traffic distribution. Today routers are typically owned and operated by the legal owner of the autonomous system (e.g. ISP or corporation). These owners then enter into *peering* agreements with each other in order to achieve end-to-end reachability of destinations across multiple hops of domains. The primary revenue model in the ISP industry revolves around different tiers of services appropriate to different groups of customers.

There are several important points regarding autonomous systems paradigm and the positive impact this paradigm has had on the development of the Internet for the past four decades:

- *Autonomous systems paradigm leads to scale and reach*:
  The autonomous system paradigm, the connectionless routing model and the distributed network topology of the Internet allows each unit (the AS) to solve performance issues locally. This in turn promotes service scale in the sense of throughput (end-to-end) and reach (the large numbers of connected endpoints). As such, it is important to see the global Internet today a connected set of "islands" of autonomous system, stitched together through peering agreements.

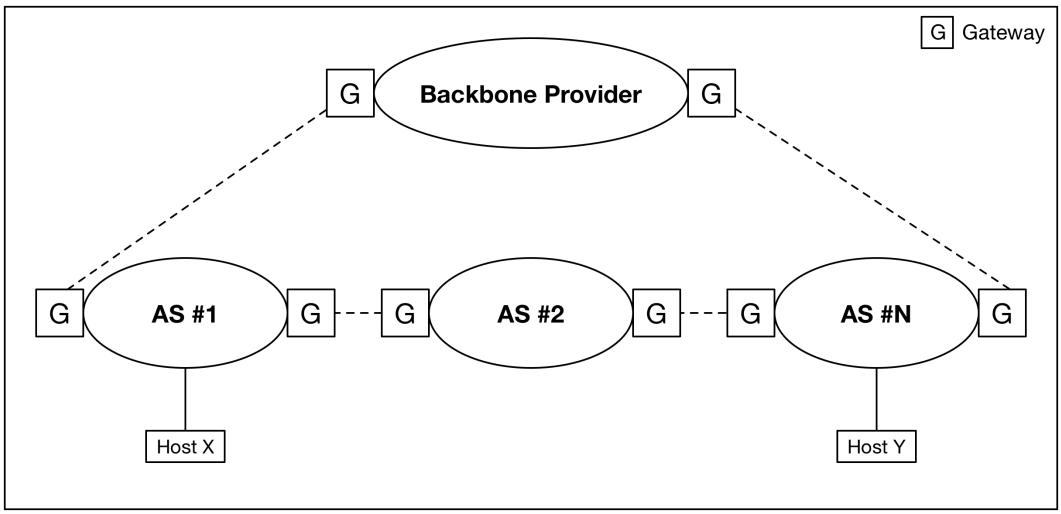

Figure 1: Autonomous Systems as a set of networks and gateways (after [4])

- *Domain-level control with distributed topology*:
  Each autonomous system typically possesses multiple routers operating the same intra-domain routing protocol. The availability of multiple routers implies availability of multiple routing paths through the domain. Despite this distributed network topology, these routers are centrally controlled (e.g. by the network administrator of the domain). The autonomous system as a control-unit provides manageability, visibility and peering capabilities centrally administered by the owner of the domain.

- *Each entity is uniquely identifiable in its domain*:
  All routers (and other devices, such as bridges and switches) in an autonomous system are uniquely identifiable and visible to the network operator. This is a precondition of routing. The identifiability and visibility of devices in a domain is usually limited to that domain. Entities outside the domain may not even be aware of the existence individual routers in the domain.

- *Autonomous system reachability*: Autonomous systems interact with each other through special kinds of routers (called "Gateways") that are designed and configured for cross domain packet routing. These operate specific kinds of protocols (such as an exterior Border Gateway Protocol [15, 16]), which provides transfer of packets across domains. For various reasons (including privacy and security) these exterior-facing gateway protocols advertise only *reachability* status information regarding routers and hosts in the domain. They do not make the routing conditions in the domain and other domain performance/throughput information visible to external entities (i.e. other autonomous systems). This limited visibility prevents external domains from performing traffic shaping that may adversely impact a given autonomous system (e.g. driving too much traffic through a given domain).

- *Autonomous systems are owned and operated by legal entities*:

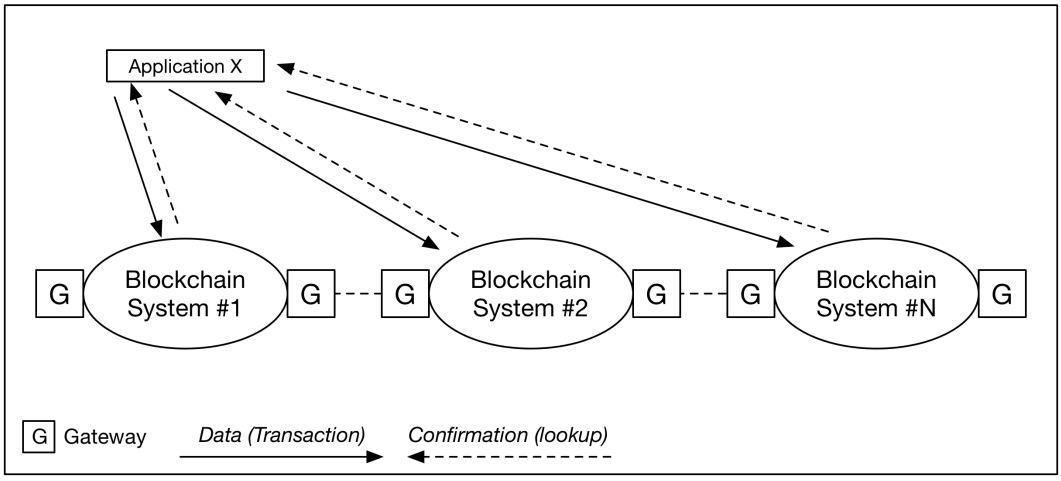

Figure 2: Example of the reliability of a simple transaction

All routing autonomous systems (routing domains) today are owned, operated and controlled by known entities. Internet Service Providers (ISPs) provide their *Autonomous System Numbers* (ASNs) and routing prefixes to *Internet Routing Registries* (IRRs). IRRs can be used by ISPs to develop routing plans. An example of an IRR is the American Registry for Internet Numbers (ARIN) [17], which is one of several IRRs around the world.

It is worthwhile pointing out that the pioneers of the Internet understood the importance of identity, and certainly did not forget it – as is often popularly claimed today by some. Indeed several groups in the IETF discussed the various issues around digital identity, digital certificates and PKI (e.g. SPKI [18], X.509 [19], PGP [20]) on an on-going for several years. It was clear even in the mid-1990s that user (human) identity was a function to be provided at layers above the IP routing layer and was a construct extraneous to routing packets.

3 Interoperable Blockchains: Towards a Design Philosophy

In this section we use the fundamental goals of the Internet architecture in the context of interoperable blockchain systems. In the 1980s there was a clear realization that it was necessary to incorporate the then existing network architectures if Internet was to be useful in a practical sense. These networks represent administrative boundaries of control [3]. Today we are seeing a similar situation, in which multiple blockchain designs are being proposed, each having different degrees of technological maturity.

Different organizations and consortiums (e.g. R3/Corda [21], EEA [22]) are developing different blockchain technologies. Additionally, several dozen digital currencies are operating today, and several digital currency exchanges have emerged. A critical aspect of these proposals is their need to address the fundamental question of *privacy* of transacting parties. Some designs (e.g. Hyperledger Fabric [23]) recognize the importance of privacy in transactions, and address these question through a permissioned design. Others are proposing

to retain a permissionless model, but use advanced cryptographic techniques (e.g. zero-knowledge proofs, homomorphic encryption, etc.) for transaction privacy as a "layer" atop a permissionless blockchain.

We believe the issue of survivability of blockchain systems to be paramount and a precondition to designing for aspects of privacy. Interoperability is key to survivability. As such, we believe that interoperability across blockchain systems will be a core requirement – both at the mechanical level and the value level – if blockchain systems and technologies are to become the fundamental infrastructure components of future global commerce.

In this section we identify and discuss some of the challenges to blockchain interoperability, using the Internet architecture as a guide and using the fundamental goals as the basis for developing a design philosophy for interoperable blockchains. We offer the following definition of an "interoperable blockchain architecture" using the NIST definition of "blockchain" (p.50 of [24]):

> *An interoperable blockchain architecture is a composition of distinguishable blockchain systems, each representing a unique distributed data ledger, where atomic transaction execution may span multiple heterogeneous blockchain systems, and where data recorded in one blockchain is reachable, verifiable and referenceable by another possibly foreign transaction in a semantically compatible manner.*

3.1 Survivability

The popular idea that a blockchain system can withstand a concerted cyberattack simply because it consists of physically distributed nodes is an untested and unproven proposition. The possible types of attacks to a blockchain system has been discussed elsewhere, and consists of a broad spectrum. These range from classic network-level attacks (e.g. network partitions, denial of services, DDOS, etc.) to more sophisticated attacks targeting the particular constructs (eg. consensus implementation [25, 26, 27]), to targeting specific implementations of mining nodes (e.g. code vulnerabilities; viruses).

In the Internet architecture, survivability meant that communications must continue despite loss of networks and gateways. In practical engineering terms, this meant the use of the packet-switching model as a realization of the *connectionless* routing paradigm. In short, the connectionless paradigm gave the freedom for each autonomous system (routing domain) to choose the best-effort delivery path through the domain and to choose the best next peered domain to which to direct traffic. Additionally, the connectionless paradigm meant that applications at both end-points of the connection did not need to be aware of routing paths and that they could continue their high level conversation as soon as the first IP packet was delivered to the receiving end-point. The idea is that the routing domains will dynamically reconstitute any failed segment of the end-to-end path from origin to destination.

This connectionless paradigm was in stark contrast to the 100-year old telecommunications infrastructure that was *connection-oriented*. In the connection-oriented paradigm, a fixed path in the telecommunications network must be setup between the caller's telephone to the receiver's telephone before any voice transmission could be carried-out. Once a path has been created, that path would be retained (fixed) throughout the duration of the call.

Any temporary failure in any segment of the fixed path in the telecommunications network resulted in the complete tear-down of the high level conversation. This would then require the caller to restart the conversation again. Consequently, attacking any one segment of a connection-oriented communications yielded the desired result (disconnection of the high level conversation). This weakness of the connection-oriented paradigm was one of the motivating reasons in the 1960s for interest in packet-switching models of communications [5].

For blockchain systems we re-interpret the term "survivability" in the sense of the completion of an application-level transaction independent of any specific blockchain autonomous system (other than the one to which the application is locally connected intra-domain). Completion here means from its transmission by an end-user application (or by a smart contract on an origin blockchain) to its confirmation on a single blockchain system or multiple blockchain systems. The application-level transaction may be composed of multiple ledger-level transactions (sub-transaction) and which may be intended for multiple distinct blockchain systems (e.g. sub-transaction for asset transfer, simultaneously with sub-transaction for payments and sub-transaction for taxes). Thus the notion of packets routing through multiple domains being opaque to the communications application (e.g. email applications, browsers) is re-cast to the notion of *sub-transactions confirmed on a spread of blockchain systems generally being opaque to the user application.*

Thus, the challenge of reliability and "best effort delivery" becomes the challenge of ensuring that an application-level transaction is completed within reasonable time, possibly independent of the actual blockchains where different ledger-level sub-transactions are finally completed and confirmed. Thus, the notion of "connectionless" here becomes one in which in a two party application-level engagement both parties should not care which specific blockchain systems confirmed their sub-transactions, so long as all confirmed sub-transactions add up to a confirmed application-level transaction that satisfies both parties.

To illustrate the challenges of survivability, we start with the simplest case in which an application sends a "data" transaction (signed hash value) to a blockchain for the purpose of recording it on the ledger of the blockchain. We ignore for the moment the dichotomy of permissionless and permissioned blockchains and ignore the specific logic syntax of the blockchain. Here the application does not care *which* blockchain records the data so long as once it is recorded the application (and other entities) can later find the transaction/block and verify the data has been recorded. Some form of confirmation must be made available by the blockchain to the transmitting application (e.g. it shows up on a confirmed block).

Figure 2 illustrates the scenario. The application transmits data-bytes (hash) to a blockchain system No. 1, and waits for confirmation (of the successful recording to the ledger) to become available. After waiting for some predetermined time unsuccessfully (i.e. time-out), the application transmits the same data-bytes to a different blockchain system No. 2. The application continues this process until it is able to obtain the desired confirmation. Thus for the application "survivability" means that the simple transaction has been successfully confirmed on some blockchain (i.e. on all nodes of that ledger), even if soon after the blockchain ceases being able to process future transactions due to attacks. The side effect maybe that the same transaction is confirmed independently on multiple blockchain systems, but the application does not care about this possible side effect so long as "the transaction

got through".

Although Figure 2 may appear overly simplistic and inefficient, it brings forth a number of questions which echo those posed in the early days of the Internet architecture development:

- *Application degree of awareness*: To what degree must an application be aware of the internal constructs of a blockchain system in order to interact with it and make use of the blockchain.

  As a point of comparison, an email client application today is not aware of constructs of packets, MPDUs, routing and so on. It interacts with mail-server according to a high-level protocol (e.g. POP3, IMAP, SMTP) and a well-defined API.

- *Placement of functions dealing with reliability*: What is the correct notion of "reliability" in the context of interoperable blockchain systems and where should the function of reliability be placed. That is, should the function of re-transmitting the same data-bytes (transaction) be part of the application, part of the blockchain system or part of a yet to be defined "middle layer".

  In the case of the Internet architecture, reliability of transmission is provided by the TCP protocol, which has a number of flow control features that "hides" reliability issues from the higher level applications.

  In the case of blockchain systems, could special types of nodes (e.g. Endorsers in Fabric [23]) provide a means to express reliability functions in a blockchain system to which the application's own reliability function can rely on – including duplicate transaction suppression at the application. That is, could these special nodes become the "middle layer" supporting reliability, based on known commitment-protocols (e.g. 2-Phase Commit in classical database systems) that allows the blockchain system to "pre-commit" (i.e. propose Read/Write sets) to the external application before writing immutably on its ledger.

- *Semantic type of blockchain*: What mechanism is needed to communicate to an external application the semantic type of the ledger-level transaction supported by a given blockchain system. For example, a blockchain system for payments is different from one for recording assets, and furthermore different payments blockchains around the world may be implemented differently. Merely publishing an application-level API does not guarantee interoperability at the blockchain ledger level and does not necessarily provide referenceability of confirmed local transactions by foreign transactions.

- *Distinguishability of blockchain systems*: For an interoperable blockchain architecture, how does an application distinguish among blockchain systems (even if they have compatible semantics) and at what level should an application be aware. Assuming the existence of multiple blockchain systems that can serve the need of the application in Figure 2, how does the application distinguish between these blockchain systems.

- *Objective benchmarks for speed and performance*: How do external entities obtain information about the current performance/throughput of a blockchain system and what measure can be used to compare across systems.

One of the key considerations in the design of the Internet architecture is the real possibility in the case of emergencies for private networks to be temporarily placed under government control for the purposes of government/military communications. The interoperability of networks was therefore crucial in answering this need. Similarly, today the question applies to blockchain systems. In the case of emergencies could independent and/or private blockchain systems be temporarily placed under government control such that relevant transactions (e.g. central bank transactions) can continue to flow. The interoperability of blockchain systems is crucial in answering this future need.

3.2 Variety of service types

The second goal of the Internet architecture was the support for different types of services, distinguished by different speeds, latency and reliability. The bi-directional reliable data delivery model was suitable for a variety of "applications" on the Internet but each application required different speeds and bandwidth consumptions (e.g. remote login; file transfer, etc). This understanding led to the realization early in the design of the Internet that more than one transport service would be needed and that the architecture must support simultaneously transports wishing to tailor reliability, delay or bandwidth usage. This resulted in the separation of TCP (that provided reliable sequenced data stream) from the IP protocol that provided "best effort" delivery using the common building block of the *datagram*. The User Datagram Protocol (UDP) [28] was created to address the need for certain applications that wished to trade reliability for direct access to the datagram construct.

For interoperable blockchain systems, we re-interpret the goal of supporting a variety of services to supporting the following variety of transaction aspects: (i) speed and achieved-majority of confirmation of a given system; (ii) the directionality of transactions; (iii) the strength of consensus:

- *Speed and achieved majority*: The speed (or "throughput") of a blockchain system refers to the confirmation speed, based on the population size of the participating nodes and other factors.
- *Directionality of ledger-transactions*: The directionality refers to whether the transmitting application is acting alone or in a request-response mode of engagement with a second party (or a group).
  - *Uni-directional transactions*: A transaction is uni-directional if the transmitting application does not intend it for any specific entity and no response from a peer application is expected. An example would be a simple asset registry blockchain which records a hash of the digital scan of certificates (e.g. equity shares, land deeds, etc).
  - *Bi-directional transactions*: A transaction is bi-directional if the transmitting application intends that transaction for a peer application and expects a response-transaction from that peer.

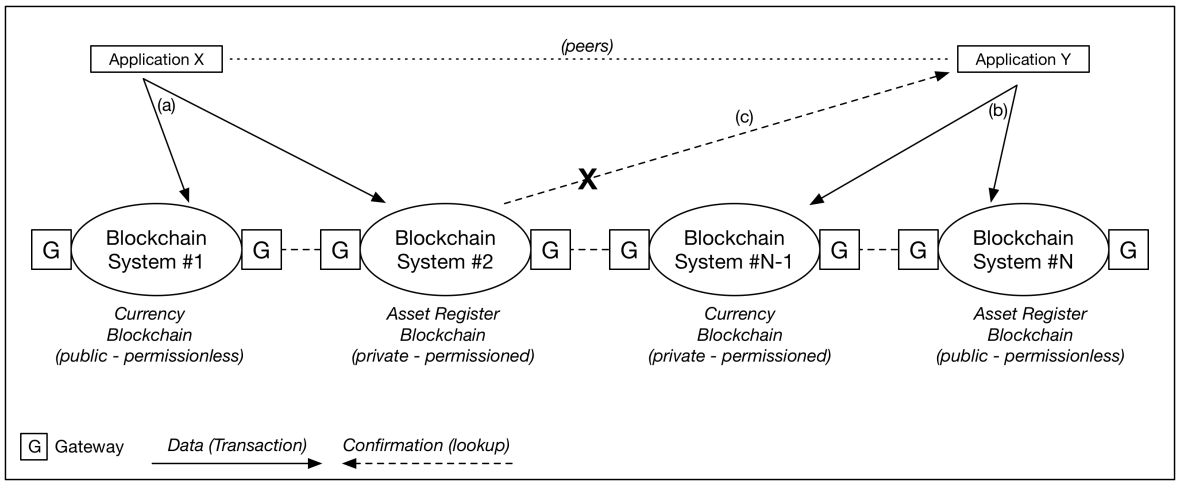

Figure 3: Example of peer applications using different variety of blockchains systems

- *Strength of consensus*: An important consideration for users and applications seeking to refer to (and therefore rely on) data recorded on a ledger within a blockchain system is the size of the population of nodes (i.e. entities contributing to the consensus) at any given moment and whether this information is obtainable. Obtaining this information maybe challenging in systems where nodes are either anonymous, or perhaps unobtainable by external entities in the case of permissioned systems.

  In the case of smart contracts, there is also the question regarding the *provenance* and source-authenticy of the (externally-sourced) data being used by a smart contract to compute an outcome.

3.3 Variety of blockchain systems

The third fundamental goal of the Internet architecture was to support a variety of networks, which included networks employing different transmission technologies (e.g. X.25, SNA, etc.), local networks and long-haul networks, and networks operated/owned by different legal entities. The *minimum assumption* of the Internet architecture – which is core to the success of the Internet as an interoperable system of networks – was that each network must be able to transport a *datagram* as the lowest unit common denominator. Furthermore, this was to be performed "best effort" – namely with reasonable reliability, but not perfect reliability.

From a blockchain interoperability perspective, one possible re-interpretation of this original problem is as follows: how can multiple types of blockchain systems support the completion of a bi-directional transaction between two applications, involving computational resources across blockchain systems where some maybe operated (or owned) by different entities and where each uses a different permissions regime. Figure 3 illustrates with a simplistic example, again aimed at drawing out questions about design.

In Figure 3 applications X and Y are each employing different blockchain systems relating to currency/payments and asset ownership. Each blockchain system implements a different

semantic logic and each operates under a different permissioning regime. When application X seeks to interact with foreign application Y, each may not have sufficient privileges to read from the permissioned blockchain where their previous transactions have been confirmed. Thus, when application X wishes to transfer (to Y) asset "ownership" (e.g. land deed) currently in blockchain system No. 2 (permissioned), application Y has no way to validate the ownership of the asset. This is because the foreign application Y does not have authorization to read from the ledger in blockchain system No. 2. This problem is further compounded in the case of smart contracts that incorporate parts of the business logic of the applications.

As such, this dilemma raises several questions, including one pertaining to the minimum assumption for interoperable blockchain systems:

- *Minimal assumption*: What is the minimal assumption for interoperable blockchain systems with regards to the notion of transaction units. In other words, what is the "datagram equivalent" of transactions – namely the transaction unit that is semantically understandable (processable) by multiple different blockchain systems.

- *Degrees of permissionability*: Currently the permissionless/permissioned distinction refers to the degree to which users can participate in the system [24]. Interoperability across permissioned blockchains poses additional questions with regards to how data recorded on the ledger can be referenced (referred to or "pointed to") by transactions in a foreign domain (i.e. another blockchain system).

- *Degrees of anonymity*: There are at least two (2) degrees of anonymity that is relevant to blockchain systems. The first pertains to the anonymity of end-users (i.e. identity-anonymity [29, 30, 31, 32]) and the second to that of the nodes participating in processing transactions (e.g. nodes participating in a given consensus instance). Combinations are possible, such as where a permissioned system may require all consensus nodes to be strongly authenticated and identified, but allows for end-users to remain permissionless (and even unidentified/unauthenticated).

3.4 Reachability

Peering agreements between autonomous systems on the Internet allows a patchwork of islands of autonomous systems to collectively provide routability of packets from its ingress point to its destination. Gateway routers (entities labelled G in Figure 1) do not export full routing information or paths to entities external to the autonomous system. Instead, gateway routers/devices only provide *reachability* information regarding local hosts or other autonomous systems accessible through its domain. This model is partly driven by business survivability, where competing ISPs (under a peering agreement) may purposely "push" (offload) traffic towards competitor ISPs, using traffic shaping tools and algorithms.

A model akin to this gateway approach may be suitable for the interoperability of blockchain systems. This is particularly important when one or more of the blockchains operate as a private/permissioned autonomous system. Using the example in Figure 3, the gateways in blockchain system No. 2 (permissioned) can act as a proxy for the permissioned blockchain. When a non-permissioned application Y seeks to obtain information regarding confirmed

transactions in blockchain No. 2, it must present a *delegated* authorization from application X who has access privileges to blockchain No. 2. We discuss delegation in Section 4.2.

3.5 Interconnecting Values

The architecture of the Internet was a messaging delivery architecture that separated the mechanical transmission of packets from the *value* of the information contained in the packets. The notion of priority packets was supported, but it was primarily intended for control data versus content payload data. As such, the notion of value was external to the Internet. Today this model remains also true for the majority blockchain systems. For example, the Bitcoin system [33] does not bind the BTC currency denomination to any real-word asset, and a such the notion of value is an external one ("in the eye of the beholder").

For specific families of applications, such as currency and financial applications, the ability to transfer value from one system to another is paramount and indeed the sole purpose of those applications and systems. Thus for a semantically homogenous or near-homogenous network of blockchain systems (e.g. payments) the challenges of value-transferral becomes more manageable.

A promising direction in this respect is the *Inter-Ledger Protocol* (ILP) proposal [34] which puts forward a packet format and per-hop transferal protocol to transmit value (payments) from a sender to a receiver over a network of currency blockchain systems. The end-to-end behavior of ILP is reminiscent of the Resource Reservation Protocol (RSVP) [35] in which a bi-directional path is "reserved" from the origin to destination, and where the path needs to exist only for a short duration of time. To do this the RSVP protocol reserves in/out interfaces (and other computational resources) at each router per-hop from the origin to destination.

In the ILP model the sender and receiver of the payment are assumed to be on distinct blockchain systems. The ILP architecture employs a value *connector* at the application level between two blockchain segments. Thus at each hop through the path from sender to receiver there may be a per-hop connector deployed. The function of the connector entity is to perform value-conversion from one currency to another. The connector behaves very similarly to a currency-exchange, and therefore a connector entity must have sufficient reserves of "foreign" currencies (for each currency it supports) in order to participate in the path being formed from the sender to the receiver. The connector model also mimics the behavior of routers and gateways in dealing with overload to their interfaces. In the case of ILP a connector becomes overloaded when most or all of its pair-wise denominations have been used or "reserved" in one or more payment paths through that connector. In this case an overloaded connector can simply reject new requests until some of its open paths have been closed and its resources (denominations) have become freed-up (settled).

It is important to note that the connectors in ILP represents the value-points which are external to the blockchain systems involved. That is, the notion of value remains separated from the transaction mechanics of the underlying blockchain systems.

3.6 Moveable Smart Contracts for Survivability

For many observers and users today of the nascent blockchain technology, the potentially revolutionizing aspect of blockchain technology is not so much the immutability of the ledger but instead the notion of *smart contracts*, which are invokable executable-code present on the P2P nodes of the blockchain system [36]. The concept of smart contracts resembles "stored procedures" (in classic database systems), but in the case of smart contracts they would reside on many or all nodes of the blockchain. However, in the case of unresponsive or unreachable blockchain systems (e.g. one under attack and therefore has a degraded throughput), the resident smart contracts may not be invokable or may not be able to complete/terminate due to severe resource shortages.

The following are some challenges related to survivability of the smart contracts feature when a blockchain system is under attack:

- Is there a *minimal common syntax* for smart contracts that allow a smart contract to be copied ("moved") from nodes in one blockchain system to nodes at a different blockchain system, where execution at the new blockchain system yields identical or semantically-equivalent output (both in the technical sense and legal sense).

- Should the physical location of the actual execution (i.e. which blockchain system) of smart contracts be important (or even be opaque) to applications.

- How can an application waiting too long (timing out) for a smart contract in one blockchain system trigger or initiate the moving (copying or migration) of the smart contract to a new blockchain system. Should this moving/copying be an automated process by nodes of a blockchain system when they detect that the system is under duress from attacks.

- How do nodes in the P2P network of a blockchain system *know* the system (parts of it or its entirety) is under direct attack or under other subliminal manipulation.

3.7 Cryptographic Survivability

An important consideration with regards to blockchain-based proposals is the complexity of the "cryptographic schemes" underlying some proposed blockchain systems. Many cryptographic schemes are composed of multiple cryptographic building blocks. In each scheme some components maybe well understood, field deployed and even standardized. However, other components may represent new ideas that have never been field-tested or withstand theoretical or practical attacks. The point here is that the development of cryptographic technology is a Darwinian evolutionary process, in which a successful attack on one design becomes lessons learned for improvements for the next design, and that this Darwinian process takes time (years or even decades).

A good case in point is the evolutionary process undergone by symmetric ciphers (block ciphers) for the past three decades (e.g. DES, 3DES, AES, etc). Symmetric ciphers are a crucial component not only for national defense cyber-infrastructures, but also of the global

commercial banking industry. Similarly, weaknesses found in some asymmetric ciphers (e.g. RSA) often results in recommended key lengths being extended (e.g. NIST SP800-175B).

Today some commercial blockchain systems are proposing to use complex schemes (e.g. Zero-Knowledge Proofs (ZKP), homomorphic encryption, etc), many of which are untested large scale in the field and perhaps at research stage at best. It remains to be seen how attacks on these schemes impact the blockchain systems which employ them.

4 Interoperability Design in Tradecoin

The MIT Tradecoin project [37] has a number of objectives, one core goal being the development of a "blueprint" model for interoperable blockchain systems which can be applied to multiple use cases. Some uses cases which have been identified are: (i) a reserve digital currency shared by a number of geopolitically diverse small countries, as a means to provide local financial stability [38]; (ii) a digital currency operating for a narrow-bank that can provide relative stability during financially volatile periods [39]; (iii) farmers with crop assets that wish to combine their assets to achieve better market presence; (iv) logistics chains with many cooperating companies holding assets that will be combined into a final product or service. As a blueprint, the Tradecoin interoperability model must be independent of any specific blockchain implementation.

In this section we discuss various aspects of the Tradecoin interoperability model, following the notion of autonomous systems as developed in the Internet architecture. Attention is given specifically to cross-domain transactions and the role of "gateways" (special nodes or computers) that support interoperability across blockchain autonomous systems.

Although this section focuses on the technical aspects of gateways, it is generally understood that interoperability needs to occur both at the technical (mechanical) level and at the "value" level:

- *Mechanical level interoperability*: This layer encompasses the computer and network systems (hardware and software) that implement the technical blockchain constructs as well as the communications constructs. This layer contains protocols, cryptography, encryption, signing, identities (identifiers), operational governance rules, consensus algorithms, transactions, probes and so on.

- *Value level interoperability*: This layer is external to the blockchain system and encompasses constructs that accord value as perceived in the human world. Humans, societies, real assets, fiat currencies, liquidity, legal regimes and regulations all contribute to form the notion of "value" as attached to (bound to) the constructs (e.g. coins, tokens) that circulate in the blockchain system, and which are in-turn implemented by systems and subsystems at the mechanical level. Included also is the notion of legal governance rules which support humans in making decisions regarding the operations of a given blockchain as an autonomous system.

We believe this general two-level view is consistent with the end-to-end principle of the Internet architecture because it places the human semantics (value) and social interactions

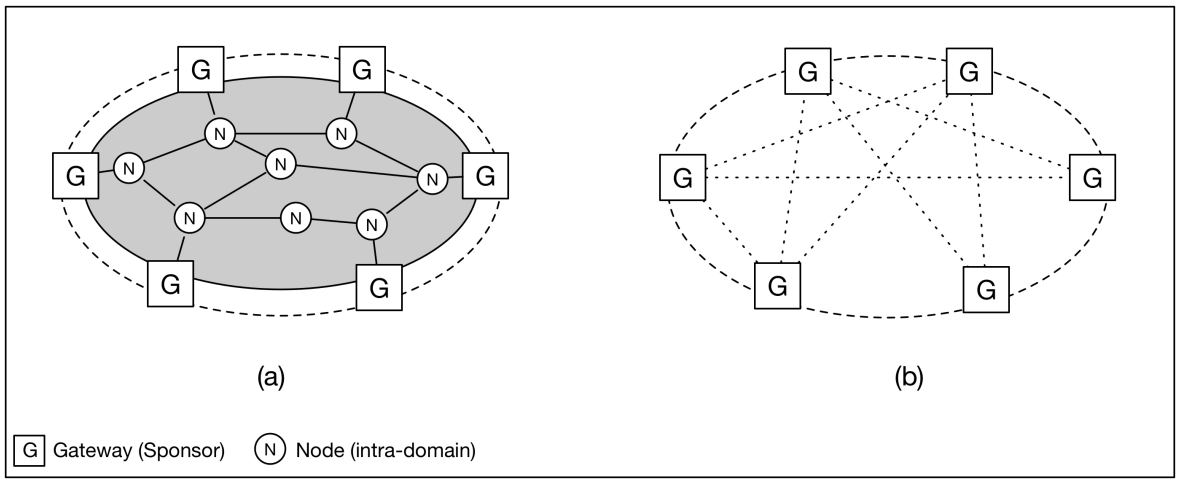

Figure 4: Tradecoin model: (a) Blockchain as Autonomous System, and (b) Gateways

at the end-points of (i.e. outside) the mechanical systems. Interoperability at the mechanical level is necessary for interoperability at the value level but does not guarantee it. Human agreements (i.e. legal contracts) must be used at the value level to provide semantically compatible meanings to the constructs (e.g. coins, tokens) that circulate in the blockchain system. Thus, *semantic interoperability* is required not only at the mechanical level but also at the value level.

The mechanical level plays a crucial role in providing technological solutions that can help humans in quantifying risk through the use of a more measurable *technical trust* [40, 41]. In most cases technical-trust is obtained through a combination of demonstrably strong cryptographic algorithms, proper key management, tamper-resistant hardware (to a specific measure of cost) and *roots of trust* that combine trustworthy computing principles with *legal trust* (e.g. contracts that binds the root of trust with legally enforceable obligations and warranties).

Legal trust is the bridge between the mechanical level and the value level. That is, technical-trust and legal-trust support *business trust* (at the value level) by supporting real-world participants in quantifying and managing risks associated with transactions occurring at the mechanical level. Standardization of technologies that implement technical trust promotes the standardization of legal contracts – also known as legal trust frameworks – which in turn reduces the overall business cost of operating autonomous systems.

4.1 The Tradecoin Autonomous System

The Tradecoin model views each blockchain system as being a fully fledged autonomous system in the sense of the Internet architecture. The basic design of the blockchain autonomous system is shown in Figure 4(a). The design consists of entities which operate intra-domain (e.g. P2P nodes) and entities that operate inter-domain.

- *Intra-domain entities*: The entities dealing with intra-domain transactions are the *peer-to-peer nodes* (N). This includes nodes that participate in consensus computations (e.g.

full mining nodes in Bitcoin [33]), nodes that "orchestrate" consensus computations (e.g. Orderers and Endorsers in Fabric [23]), and nodes which perform validations only (e.g. Validators in Ripple [42]). For simplicity we assume that a blockchain autonomous system contains only one ledger, and possibly multiple ledgers if they are tightly coupled with identical semantics and under the same domain.

- *Inter-domain entities*: The entities dealing specifically with inter-domain transactions are denoted as *Gateways* and shown as G in in Figure 4(b). Depending on implementation, gateways may operate collectively as a group or they may operate loosely-coupled.

The *boundary* (perimeter) of the autonomous system is largely determined by (i) the degree of identity-anonymity and authentication of the intra-domain entities such as nodes (i.e. to each other in the same domain); and (ii) degree of permissionability of the blockchain autonomous system as seen by foreign entities (see Section 3.3). Thus, unlike classical corporate network topologies which define a clear (defensive) physical network perimeter, in blockchain autonomous systems the "perimeter" is defined by the participation of the entities in an intra-domain arrangement. These intra-domain entities need not necessarily be located in the same physical proximity, but in some cases may be required to enter into a legal agreement (e.g. system rules of operation). Depending on the use case, these entities may be owned by a single organization (e.g. private corporation), or be jointly-owned by a multiple organizations (e.g. consortium).

In defining permissionability the Tradecoin interoperability model recognizes a number of permissioning configurations. The first two pertain to the P2P nodes of the blockchain system, while the remaining two pertain to the end-users and applications:

- *Node-permissioned*: In this permissioning configuration, nodes must be permissioned to participate in one or more aspects of the operation of the blockchain system. Using the Hyperledger Fabric [23] as an example, an instance of a Fabric blockchain may require all the Orderers and Endorsers nodes to be authenticated and authorized to operate.

- *Consensus-permissioned*: In this permissioning configuration only the nodes that participate directly in consensus algorithm computations need to be authenticated and authorized.

- *User write-permissioned*: This permissioning configuration pertains to the end-users and their applications. In a write-permissioned configuration, the user/application must be authenticated and authorized to transmit a ledger-modifying transaction to the blockchain.

- *User read-permissioned*: In a read-permissioned configuration, the user/application must be authenticated and authorized to read the contents of the ledger of the blockchain system.

4.2 The Role of Gateways

The set of gateways G collectively provides at least three (3) types of functional support to the entities in the blockchain autonomous system:

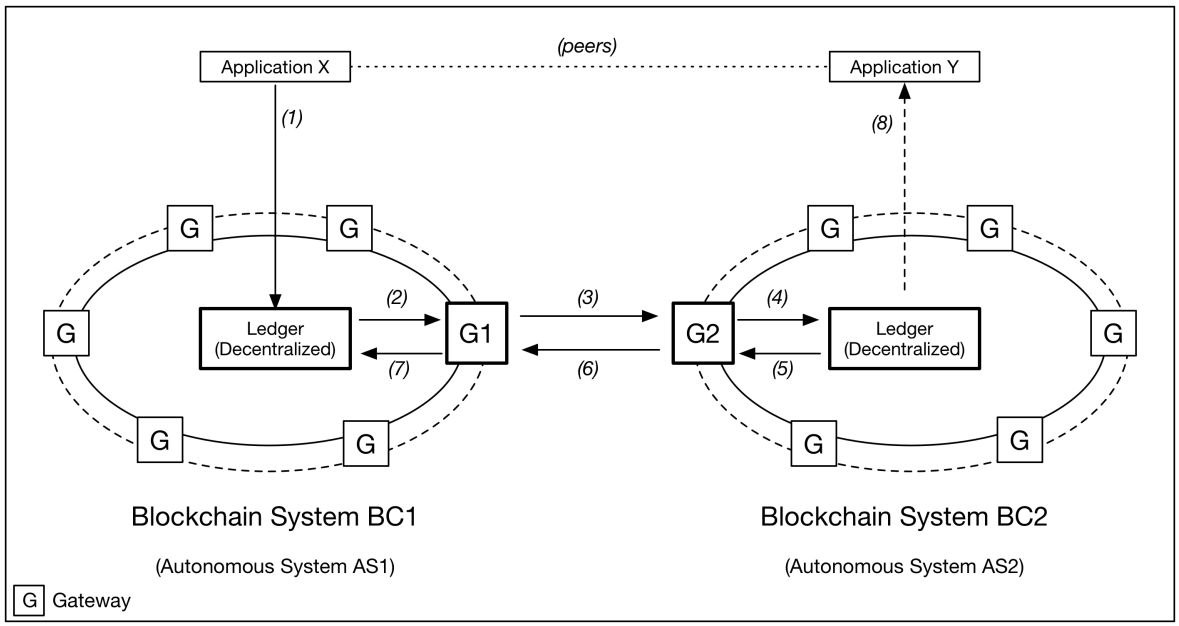

Figure 5: Set of Gateways for Reachability and Transaction Mediation

- *Value stability*: For implementations that exchange value-carrying constructs (e.g. digital coins, assets), the gateways collectively act to provide stability at the value level.

  Thus, for example, in the case of a Narrow Bank scenario [39] which distinguishes asset-contributing entities (called "sponsors") from users, each sponsor may own and operate equal numbers of nodes and gateways within the shared blockchain system. The gateways may collectively implement an asset-tracking blockchain, while the remaining intra-domain nodes implement the local digital currency of the Narrow Bank. The value-translation between the coins circulating in the currency blockchain (intra-domain) against other foreign denominations is performed by the gateways, which may hold a basket of real-world assets (e.g. basket of oil, gold, commodities, greenback, etc) to back the intra-domain digital currency.

- *Reachability*: The gateways collectively support reachability to data intra-domain on ledger of the blockchain. They support cross-domain lookup mechanisms from foreign entities (e.g. nodes in other blockchains) seeking to locate data pertaining to transactions confirmed on the local ledger. That is, the gateways provide reachability so that internal confirmed transaction can be meaningfully referenced outside the blockchain system.

- *Transaction mediation*: The gateways provide a mediation function for cross-domain transactions involving two or more (permissioned) blockchain systems, where transaction data in ledgers may be considered as private and sensitive information.

4.3 Reachability

An interoperable architecture for blockchain systems must allow not only for entities to be uniquely identifiable and authenticated, but also for transactions on the ledger to be uniquely *referenceable* across domains regardless of permissioning configurations:

- *Endpoint identifier resolutions*: Gateways provide a perimeter/boundary for permissioned blockchain systems for the purposes of naming/identifier resolution (i.e. namespace management). Thus when a transaction-identifier is externally referenced and resolved to a ledger-entry inside a permissioned blockchain system, the gateways "intercept" that resolution request and act as (defensive) *end-points* for these external requests.

- *Identifier masking for data privacy*: Gateways provide "masking" (re-naming) of transaction identifiers in local ledgers inside the blockchain autonomous system. This may involve mapping different identifiers for the same transaction data, where one identifier is used for intra-domain referencing and another is used for cross-domain referencing. The gateways also play a role in filtering information leaving the boundary of a blockchain system, limiting privacy leakage.

  Note that this mapping idea itself is not new and is deployed today at a global scale in Internet addressing (e.g. IPv4/IPv6 address mapping, NAT address traversal [43]).

- *Identifier referencing and de-referencing service*: Gateways may collectively implement a transaction-identifier resolution service, in a similar sense to the hierarchical arrangement of the Domain Name Service (DNS and DNSSEC) [44, 45].

With regards to addressability, one promising approach is that of the Inter-Ledger Protocol [34] (v1.0.0) which proposes the use of an addressing scheme that allows ledgers and nested-ledgers to be identified, and which proposes a global allocation scheme for these addresses.

4.4 Inter-Domain Transaction Mediation

Gateways in a blockchain autonomous system may collectively provide a mechanism to proactively *mediate* cross-domain transactions involving two blockchain systems as distinct autonomous systems. The gateways in the respective systems must interact to facilitate the atomic and correct recording of cross-ledger transactions.

Figure 5 illustrates the simple use-case discussed earlier. Here, a user with Application X has his or her asset-ownership (e.g. land title deed) recorded on the ledger inside blockchain BC1. The user wishes to transfer legal ownership of the asset to a different user running Application Y, and to have the asset recoded authoritatively on the ledger inside blockchain BC2. Both blockchain BC1 and BC2 are permissioned/private blockchain systems. Thus, none of the gateways in BC1 can directly read/write to BC2, and none of the gateways in BC2 can directly read/write to BC1.

The term "authoritative" here means that (i) henceforth any external de-referencing of the asset identifier must resolve to blockchain BC2; and (ii) the ledger in BC1 must be

"marked" for that asset such that it henceforth points to BC2 for any local look-ups for that asset transaction data.

The Tradecoin interoperability model proposes the use of the classical cross-domain *delegation* paradigm that is well-established in several sectors of industry (e.g. cross-domain directory services [46, 47, 48]). Without going into details, in Figure 5 the set of gateways in BC1 and BC2 respectively act as delegates to the users/applications involved. Thus, the entities intra-domain within BC1 and BC2 must trust their gateways to create a temporary *bridge* that allows their gateways to be synchronized temporarily until both ledgers (in BC1 and BC2 respectively) have completed recording the asset-transfer transaction.

Since blockchains BC1 and BC2 are permissioned and one side cannot see the ledger at the other side, the gateways of each blockchain must "vouch" that the transaction have been confirmed on the respective ledgers. That is, the gateways must issue legally-binding signed assertions that makes them liable for misreporting (intentionally or otherwise).

The notion of gateway signatures can be expanded. For example, it can be issued by one gateway only, by a subset of a group of gateways (fixed subset or dynamically chosen), by a threshold of any N out of M gateways, or it can even be a collective *group signature* [49] of all gateways in the blockchain system. Each of these gateway signatures must also be recorded on the respective ledgers.

There are several desirable features of the gateway-mediated approach outline above:

- *Verifiability of local confirmations*: Both the transmitter and recipient applications must be able to independently verify that the transaction was confirmed on their respective blockchains, with sufficient data to allow post-event auditing.

- *Legally binding signatures of gateways*: Delegated gateways must have signatures that are binding, independent of how the gateway(s) was chosen to be the delegate for the given cross-domain transaction. Thus, part of the peering agreement between two blockchains autonomous systems must include legal contracts covering digital signatures from their respective gateways.

- *Multiple delegation paths*: There must be multiple reliable "paths" (i.e. set of respective gateways) between blockchains BC1 and BC2. Thus, looking at Figure 5 there must be multiple paths from BC1 and BC2, and from BC2 to BC1. Any gateway in BC1 must be able to "pair" with any gateway in BC2.

  The gateways themselves must never become a hindrance to completing the a cross-domain transaction. Attacking gateways must not yield better results (to the attacker) compared to attacking the P2P nodes intra-domain in the blockchain system.

- *Global resolution to the correct authoritative blockchain*: Any external entities seeking to lookup/resolve an identifier (e.g. linked to the asset) must always resolve to the correct authoritative blockchain system. In order words, in Figure 5 after the cross-domain transaction has completed, subsequent lookups of the asset must always resolve to BC2 (or the gateways of BC2).

- *Identifiability and authenticity of gateways*: In order for gateways to act as a delegate for the user/application, they must be identifiable (i.e. non-anonymous) both internally

(intra-domain) and externally (inter-domain). Gateways must be able to mutually authenticate each other without any ambiguity as to their identity, legal ownership or the "home" blockchain autonomous system which they exclusively represent.

4.5 Peering Agreements for Blockchain Systems

A key aspect in the Internet architecture that promotes and expands the interconnectivity of the autonomous systems is the *peering* agreements between these systems. In the context of IP routing, peering is voluntary and occurs typically between Internet Service Providers (ISPs). The peering agreements are contracts that define the various interconnection mechanical aspects (e.g. traffic bandwidth, protocols, etc) as well as fees ("settlements") and possible penalties. Peering is made possible by the standardization of cross-domain routing protocols (e.g. BGP [15])

Historically, a peering arrangement can be considered "public" in the sense that there is a "group contract" among a group of ISPs that allow any group-member to transit traffic through another member. Peering agreements can also be "private" in that it is entered into by two ISPs, providing mutually better service levels to both parties. Peering agreements provides the ISPs with the correct incentive structure for them to operate their autonomous system as a business.

For the interoperability of blockchain systems, a notion similar to peering and peering-agreements must be developed that: (i) define the semantic compatibility required for two blockchains to exchange cross-domain transactions; (ii) specifies the cross-domain protocols required; (iii) specifies the delegation and technical-trust mechanisms to be used; and (iv) define the legal agreements (e.g. service levels, fees, penalties, liabilities, warranties) for peering.

It is important to note that in the Tradecoin interoperability model, the gateways of a blockchain system represents the *peering-points* of the system. The peering agreement can be established either with another blockchain system (private bilateral), with a group of blockchain systems (private multi-lateral), or with an open "exchange" system (public peering).

5 Discussion: Survivable Digital Currency and CBDC

Several governments around the world have recently indicated their interest in using digital currency technology as the basis for a future reserve currency, also referred to as *Central Bank Digital Currency* (CBDC) or *Sovereign Money* [50]. Some examples include Switzerland [51], China [52] and Russia [53].

Although digital currencies may enhance economic strengths of certain nations, the move to a digital currency comes with its own set of challenges. As nations increasingly rely on digital infrastructures, those infrastructures – if not designed and operated correctly – may bring a different set of liabilities.

In this section we pose a number of questions concerning the survivability of digital currencies and CBDCs, particularly in the face future sophisticated cyber-attacks on the monetary

electronic infrastructure. Instead of focusing on the survival of one blockchain system, we discuss *communities* of blockchain autonomous systems as the basic unit of concern, recognizing that there will be many communities with varying sizes, technical capabilities and varying instruments of value.

- *Independence of communities of blockchain systems*: Assuming that blockchain systems deployments evolve organically in a similar manner to the Internet, what would be the most useful composition of "communities" of blockchain autonomous systems from the point of view of currency survivability? A key aspect is the ability of a community to continue functioning economically while the monetary flows into (out of) that community are temporarily disrupted.

  Today the Internet operates not only at the level of local ISPs, but also across long-haul physical networks coast to coast, and overseas. Connectivity of the Internet within the US and Canada will most likely continue to operate even if overseas connectivity was lost. Similarly, loss of coast to coast IP connectivity will still allow local ISPs to offer services to its local physically connected communities. Multiple reliable IP traffic paths coast to coast ensures that disruptions from attacks have minimal effects.

- *Survivable peering models*: What kinds of peering agreements need to be developed for blockchain autonomous systems within a community to enhance the operational survival chances of that community. Furthermore, should *super-peering* agreements be developed for cross-community engagements that come into effect in emergency situations.

- *Self-protecting blockchain autonomous systems*: How can advanced artificial intelligence (AI) and machine learning (ML) technologies be used to enhance the protection of communities of blockchain autonomous systems? Distributed AI/ML tools can be used to analyze community behaviors on the transactional level and provide insight into anomalies that may indicate unauthorized attempts to alter or influence monetary flows within a given community. The predictive capabilities of these tools could be used to support the enhanced shaping of monetary-flows in anticipation of emergent attacks.

- *Cross-community recovery from attacks*: How can communities of blockchain autonomous systems re-establish transactional connectivity automatically and organically (at the mechanical-level and value-level) after they have experienced "isolation" due to successful attacks? Furthermore, how can "old infrastructure" (e.g. transactional systems, interbank networks, paper cash, etc) be used to boot-up communities into a stable state (preferably into the same pre-attack state).

6 Conclusions

The fundamental goals underlying the Internet architecture has played a key role in determining the interoperability of the various networks and service types, which together compose

the Internet as we know it today. Interoperability is key to survivability. A number of design principles emerged from the evolution of internet routing in the 1970s and 1980s, which ensured the scalable operation of the Internet over the last three decades.

We believe that a similar design philosophy is needed for interoperable blockchain systems.

The recognition that a blockchain system is an autonomous system is an important starting point that allows notions such as reachability, referencing of transaction data in ledgers, scalability and other aspects to be understood more meaningfully – beyond the current notion of throughput ("scale"), which is often the sole measure of performance used with regards to many blockchain systems today.

Furthermore, interoperability forces a deeper re-thinking into how permissioned and permissionless blockchain systems can interoperate without a third party (such as an exchange). A key aspect is the semantic interoperability at the value level and at the mechanical level. Interoperability at the mechanical level is necessary for interoperability at the value level but does not guarantee it. The mechanical level plays a crucial role in providing technological solutions that can help humans in quantifying risk through the use of a more measurable notion of technical-trust. Human agreements (i.e. legal contracts) must be used at the value level to provide semantically compatible meanings to the constructs (e.g. coins, tokens) that circulate in the blockchain system.

References

[1] S. Haber and W. Stornetta, "How to Time-Stamp a Digital Document," in *Advances in Cryptology - CRYPTO'90 (LNCS 537)*, 1991, pp. 437–455.

[2] D. Bayer, S. Haber, and W. Stornetta, "Improving the efficiency and reliability of digital time-stamping," in *Sequences II: Methods in Communication, Security and Computer Science*, R. Capocelli, A. DeSantis, and U. Vaccaro, Eds. Springer, 1993, pp. 329–334.

[3] D. Clark, "The Design Philosophy of the DARPA Internet Protocols," *ACM Computer Communication Review – Proc SIGCOMM 88*, vol. 18, no. 4, pp. 106–114, August 1988.

[4] V. G. Cerf and R. E. Khan, "A Protocol for Packet Network Intercommunication," *IEEE Transactions on Communications*, vol. 22, pp. 637–648, 1974.

[5] J. Abbate, *Inventing the Internet*. MIT Press, 1999.

[6] K. McCloghrie and M. Rose, "Management information base for network management of tcp/ip-based internets," August 1988, RFC1066. [Online]. Available: http://tools.ietf.org/rfc/rfc1066.txt

[7] ——, "Management information base for network management of tcp/ip-based internets," May 1990, RFC1156. [Online]. Available: http://tools.ietf.org/rfc/rfc1156.txt

[8] J. Saltzer, D. Reed, and D. Clark, "End-to-End Arguments in System Design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, November 1984.

[9] S. Kent and R. Atkinson, "Security architecture for the internet protocol," November 1998, RFC2401. [Online]. Available: http://tools.ietf.org/rfc/rfc2401.txt

[10] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet security association and key management protocol (isakmp)," November 1998, RFC2408. [Online]. Available: http://tools.ietf.org/rfc/rfc2408.txt

[11] D. Harkins and D. Carrel, "The internet key exchange (ike)," November 1998, RFC2409. [Online]. Available: http://tools.ietf.org/rfc/rfc2409.txt

[12] T. Dierks and C. Allen, "The tls protocol version 1.0," January 1999, RFC2246. [Online]. Available: http://tools.ietf.org/rfc/rfc2246.txt

[13] G. Malkin, "Rip version 2," November 1998, RFC2453. [Online]. Available: http://tools.ietf.org/rfc/rfc2453.txt

[14] J. Moy, "Ospf version 2," April 1998, RFC2328. [Online]. Available: http://tools.ietf.org/rfc/rfc2328.txt

[15] K. Lougheed and Y. Rekhter, "Border gateway protocol (bgp)," June 1989, RFC1105. [Online]. Available: http://tools.ietf.org/rfc/rfc1105.txt

[16] Y. Rekhter, T. Li, and S. Hares, "A border gateway protocol 4 (bgp-4)," January 2006, RFC4271. [Online]. Available: http://tools.ietf.org/rfc/rfc4271.txt

[17] ARIN, "American Registry for Internet Numbers – Autonomous System Numbers (asn.txt)," 2018, https://www.arin.net.

[18] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "SPKI certificate theory," September 1999, RFC2693. [Online]. Available: http://tools.ietf.org/rfc/rfc2693.txt

[19] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 public key infrastructure certificate and crl profile," January 1999, RFC2459. [Online]. Available: http://tools.ietf.org/rfc/rfc2459.txt

[20] D. Atkins, W. Stallings, and P. Zimmermann, "PGP message exchange formats," August 1996, RFC1991. [Online]. Available: http://tools.ietf.org/rfc/rfc1991.txt

[21] R3CEV, "R3," 2018, https://www.r3.com.

[22] EEA, "Enterprise Ethereum Alliance," 2018, https://entethalliance.org.

[23] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S. W. Cocco, and J. Yellick, "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, ser. EuroSys '18. New York, NY, USA: ACM, 2018, pp. 30:1–30:15.

[24] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain Technology Overview," NIST Draft NISTIR 8202, January 2018, available on `https://csrc.nist.gov`,.

[25] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security - 18th International Conference, FC 2014*, March 2014, pp. 436–454.

[26] J. A. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, 2015, pp. 281–310.

[27] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun, "Is bitcoin a decentralized currency?" *IEEE Security & Privacy*, vol. 12, no. 3, pp. 54–60, 2014.

[28] J. Postel, "User datagram protocol," August 1980, RFC0768. [Online]. Available: http://tools.ietf.org/rfc/rfc0768.txt

[29] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–88, February 1981.

[30] S. Brands, "Untraceable off-line cash in wallets with observers," in *CRYPTO'93 Proceedings of the 13th Annual International Cryptology*. Springer-Verlag, 1993, pp. 302–318.

[31] J. Camenisch and E. Van Herreweghen, "Design and implementation of the Idemix anonymous credential system," in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 21–30.

[32] T. Hardjono and N. Smith, "Cloud-Based Commissioning of Constrained Devices Using Permissioned Blockchains," in *Proceedings of the 2Nd ACM International Workshop on IoT Privacy, Trust, and Security (IoTPTS 2016)*. ACM, 2016, pp. 29–36.

[33] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[34] S. Thomas, E. Schwartz, and A. Hope-Bailie, "The Interledger Protocol," Internet Engineering Task Force, draft-thomas-interledger-00, July 2016, https://tools.ietf.org/html/draft-thomas-interledger-00.

[35] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (rsvp) – version 1 functional specification," September 1997, RFC2205. [Online]. Available: http://tools.ietf.org/rfc/rfc2205.txt

[36] Norton Rose Fulbright, "Can smart contracts be legally binding contracts," Norton Rose Fulbright, Report, November 2016, http://www.nortonrosefulbright.com/knowledge/publications/144559/can-smart-contracts-be-legally-binding-contracts.

[37] A. Lipton, T. Hardjono, and A. Pentland, "Digital Trade Coin (DTC): Towards a more stable digital currency (accepted for publication)," *Journal of the Royal Society Open Science (RSOS)*, 2018.

[38] A. Lipton and A. Pentland, "Breaking the Bank," *Scientific American*, vol. 318, no. 1, pp. 26–31, 2018.

[39] A. Lipton, A. Pentland, and T. Hardjono, "Narrow banks and fiat-backed digital coins," *CAPCO Journal of Financial Transformation*, vol. 47, no. 1, pp. 101–116, April 2018.

[40] Trusted Computing Group, "TPM Main – Part 1 Design Principles – Specification Version 1.2," Trusted Computing Group, TCG Published Specification, October 2003, http://www.trustedcomputinggroup.org/ resources/ tpm_main_specification.

[41] T. Hardjono and N. Smith (Eds), "TCG Infrastructure Reference Architecture for Interoperability (Part 1) – Specification Version 1.0 Rev 1.0," June 2005, http://www.trustedcomputinggroup.org/ resources.

[42] D. Schwartz, N. Youngs, and A. Britto, "The Ripple Protocol Consensus Algorithm," Ripple Inc., Ripple Labs. Report 2014, 2014.

[43] P. Srisuresh and M. Holdrege, "Ip network address translator (nat) terminology and considerations," August 1999, RFC2663. [Online]. Available: http://tools.ietf.org/rfc/rfc2663.txt

[44] P. Mockapetris, "Domain Names: Concepts and Facilities," November 1983, RFC0882. [Online]. Available: http://tools.ietf.org/rfc/rfc0882.txt

[45] D. E. 3rd, "Domain name system security extensions," March 1999, RFC2535. [Online]. Available: http://tools.ietf.org/rfc/rfc2535.txt

[46] J. Kohl and C. Neuman, "The kerberos network authentication service (v5)," September 1993, RFC1510. [Online]. Available: http://tools.ietf.org/rfc/rfc1510.txt

[47] Microsoft Corporation, "Microsoft Kerberos Protocol Extensions," Microsoft Corporation, MS-KILE Specification v20140502, May 2014.

[48] ——, "Microsoft Privilege Attribute Certificate Data Structure," Microsoft Corporation, MS-PAC Specification v20140502, May 2014.

[49] G. Ateniese and G. Tsudik, "Some open issues and new directions in group signature schemes," in *Financial Cryptography - Third International Conference FC '99 (LNCS 1648)*, February 1999, pp. 196–211.

[50] A. Sheng and X. Geng, "A digital currency should be adopted as the world's leading reserve currency," March 2018, https://www.weforum.org/agenda/2018/04/from-dollar-to-e-sdr.

[51] Vollgeld-Initiative, "Sovereign Money Initiative," November 2017, https://www.vollgeld-initiative.ch.

[52] W. Knight, "China's Central Bank has begun cautiously testing a digital currency," June 2017, https://www.technologyreview.com/s/608088/chinas-central-bank-has-begun-cautiously-testing-a-digital-currency/.

[53] K. Galouchko and A. Baraulina, "Russia at odds over cryptocurrencies as central bank digs in," January 2018, https://www.bloomberg.com/news/articles/2018-01-11/russia-at-odds-over-cryptocurrency-trade-as-central-bank-digs-in.