Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: May 11, 2022

T. Hardjono
MIT
M. Hargreaves
Quant Network
N. Smith
Intel
V. Ramakrishna
IBM

November 7, 2021

Interoperability Architecture for DLT Gateways
draft-hardjono-blockchain-interop-arch-03

Abstract

With the increasing interest in the potential use of blockchains and decentralized ledger technology (DLT) networksorks for virtual asset management, there is a need for these networks to have interoperability to support applications and services built atop these networks. An interoperability architecture for DLT networks is therefore needed in order to permit the secure flow of digital assets different DLT networks, satisfying the properties of transfer atomicity, consistency and durability. The architecture must recognize that there are different DLT networks and that the interior constructs in these networks maybe incompatible with one another. This document proposes an interoperability architecture based on DLT Gateways, which are points of interconnection between networks. Among others, the gateways implement one or more protocols for the transfer (or exchange) digital assets between DLT networks. A gateway belonging to a DLT network peers with another gateway belonging to a different DLT network to perform the asset transfer between the two networks.

Status of This Memo

This Internet-Draft will expire on May 11, 2022.

Table of Contents

1. Introduction 3
2. Terminology 3
3. Assumptions and Principles 6
   3.1. Design Principles 6
   3.2. Operational Assumptions 7
4. Interoperability Modes 7
5. Architecture 9
   5.1. Goal of Architecture 9
   5.2. Overview of Asset Transfer 10
   5.3. Desirable Properties of Asset Transfer 10
   5.4. Event log-data, crash recovery and backup gateways 11
   5.5. Overview of the Phases in Asset Transfer 12
6. Pre-transfer Verification of Asset and Identities (Phase 1) . 13
7. Evidence of asset locking or escrow (Phase 2) 15
8. Transfer Commitment (Phase 3) 17
9. Related Open Issues 19
   9.1. Global identification of blockchain systems and public-
        keys 19
   9.2. Discovery of gateways in DLT Networks 20
   9.3. Remote gateway discovery 20
   9.4. Commitment protocols and forms of commitment evidence . . 20
10. Security Considerations 21
11. Policy Considerations 21
12. References 22
   12.1. Normative References 22
   12.2. Informative References 22
Authors' Addresses 24

1. Introduction

Currently there is little technical interoperability between decentralized ledger technology (DLT) networks. This results in the difficulty in transferring or exchanging virtual (digital) assets from one DLT network to another directly.

The existing solutions involve a third party that mediates the transfer. This mediating third party is typically an asset-exchange entity (i.e. crypto-exchange) operating in a centralized hub-spoke fashion. This reliance on a third party results not only in delays in transfers, but also in the need for asset owner to have a business relationship (e.g. open accounts) at the mediating third party. Many of these solutions centralize control at the hands of the mediating party, thereby diminishing the autonomy of blockchains and DLT networks, and limits their scalability.

This document proposes an interoperability architecture based on DLT Gateways, which are points of interconnection between networks. There are several services that may be offered by a DLT gateway, one of which being the direct transfer of a digital asset from one DLT network to another via pairs of gateways without a mediating third party. A given DLT network may have one or more gateways to perform a unidirectional direct transfer of digital assets to another DLT network possessing one or more compatible gateway. Similar to the notion of border gateways in interdomain routing (e.g. running the BGPv4 protocol), a DLT gateway belonging to an origin DLT network is said to peer with another gateway is a destination DLT network. Both gateways must implement an asset transfer protocol that must satisfy certain security, privacy and atomicity requirements.

The purpose of this architecture document is to provide technical framework within which to define the required properties of a DLT gateway that supports an atomic asset transfer protocol, such as ODAP [ODAP]. These properties include the security, reliability and data privacy of digital asset transfers between pairs of gateways belonging to differing DLT networks.

2. Terminology

There following are some terminology used in the current document. We borrow terminology from NIST and ISO as much as possible, introducing new terms only when needed:

o Blockchain system: Blockchains are distributed digital ledgers of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one (making it tamper evident) after validation and undergoing a

consensus decision. As new blocks are added, older blocks become more difficult to modify (creating tamper resistance). New blocks are replicated across copies of the ledger within the network, and any conflicts are resolved automatically using established rules [NIST].

- Distributed ledger technology (DLT) system: Technology that enables the operation and use of distributed ledgers, where the ledger is shared (replicated) across a set of DLT nodes and synchronized between the DLT nodes using a consensus mechanism [ISO].

- DLT Network: A generic term for blockchain systems.

- Resource Domain: Resource Domain: The collection of resources and entities participating within a blockchain or DLT network. The domain denotes an boundary for permissible or authorized actions on resources.

- Interior Resources: The various interior protocols, data structures and cryptographic constructs that are a core part of a blockchain or DLT network. Examples of interior resources include the ledger (blocks of confirmed transaction data), public keys on the ledger, consensus protocol, incentive mechanisms, transaction propagation networks, etc.

- Exterior Resources: The various resources that are outside a blockchain or DLT network, and are not part of the operations of the network. Examples include data located at third parties such as asset registries, ledgers of other DLT network, PKI infrastructures, etc.

- Nodes: The nodes of the blockchain or DLT system which form the peer-to-peer network, which collectively maintain the shared ledger in the system by following a consensus algorithm.

- DLT Gateway: a DLT gateway is the collection of services, controlled by one legal entity, which connects to a minimum of one DLT network to provide read and write access to the ledger of that DLT network. A DLT gateway implements an atomic digital asset transfer protocol, such as ODAP [ODAP], via a DLT-neutral data formats and local storage logs. A gateway does not implement an interior consensus protocol.

- DLT address: This is the public-key of an entity as known within a DLT network or blockchain system, employed to transact on the DLT network and recorded on the ledger of the DLT network. Also referred to as the transaction public key.

- Entity public-key pair: This the private-public key pairs of an entity used for interactions outside the DLT network (e.g. TLS 1.3). The term is used to distinguish this public-key from the blockchain address.

- Asset transfer protocol: The gateway-to-gateway technical protocol used by two gateways to perform a unidirectional transfer of a virtual (digital_ asset.

- Asset profile: The prospectus of a regulated asset that includes information and resources describing the virtual asset. This includes, among others, the asset name/code, issuing authority, denomination, jurisdiction, and the URLs and mechanisms to validate the information. The asset profile is independent from the specific instantiation of the asset (on a DLT network or otherwise) and independent from its instance-ownership information.

- Virtual Asset: A virtual asset is a digital representation of value that can be digitally traded, or transferred as defined by the FATF [FATF]. We use the term interchangeably with ?digital asset?.

- Virtual Asset Service Provider (VASP): Legal entity handling virtual assets as defined by the FATF [FATF].

- Originator: Person or organization seeking the transfer of virtual asset to a beneficiary

- Beneficiary: Person or organization receiving the transferred virtual asset from an originator.

- Travel Rule information: Data regarding the VASPs, originators and beneficiaries involved in an asset transfer, as defined by the FATF [FATF] and as required by the jurisdiction of operations of the VASPs.

- Gateway device identity: The identity of the device implementing the gateway functions. The term is used in the sense of IDevID (IEEE 802.1AR) or EK/AIK (in TPM1.2 and TPM2.0) [IDevID].

- Gateway owner: The VASP who legally owns and operates a gateway within a DLT network.

- Asset locking or escrow: The conditional mechanism used within a DLT network to make an asset temporarily unavailable for use by its owner. The condition of the asset release can be based on a duration of time (e.g. hash time locks) or other parameters.

- Gateway crash recovery: The local process by which a crashed gateway (i.e. device or system fault) is returned back into a consistent and operational state, ready to resume the asset transfer protocol with the peer gateway prior to the crash event.

Further terminology definitions can be found in [NIST] and [ISO]. The term 'blockchain' and 'distributed ledger technology' (DLT) are used interchangeably in this document.

3. Assumptions and Principles

The following assumptions and principles underlie the design of the current gateway architecture, and correspond to the design principles of the Internet architecture.

3.1. Design Principles

- Opaque DLT resources: The interior resources of each DLT network is assumed to be opaque to (hidden from) external entities. Any resources to be made accessible to an external entity must be made explicitly accessible by a gateway with proper authorization.

- Externalization of value: The gateway protocol is agnostic (oblivious) to the economic or monetary value of the virtual asset being transferred.

The opaque resources principle permits the interoperability architecture to be applied in cases where one (or both) DLT networks are permissioned (private). It is the analog of the autonomous systems principle in IP networking [Clar88], where interior routes in local subnets are not visible to other external networks.

The value-externalization principle permits asset transfer protocols to be designed for efficiency, security and reliability - independent of the changes in the perceived economic value of the virtual asset. It is the analog of the end-to-end principle in the Internet architecture [SRC84], where contextual information (economic value) is placed at the endpoints of the transaction. In the case of a transfer of virtual assets, the originator and beneficiary at the respective DLT networks are assumed to have a common agreement regarding the economic value of the asset. This context of the economic meaning of the value of the asset is assumed to exists at the end-points, namely at the originator and beneficiary.

3.2. Operational Assumptions

The following conditions are assumed to have occurred, leading to the invocation of the asset transfer protocol between two gateways:

- Application layer transfer request: The transfer request from an originator in the origin DLT network is assumed to have occurred prior to the execution of the asset transfer protocol.

- Identification of originator and beneficiary: The originator and beneficiary are assumed to have been identified and that consent has been obtained from both parties regarding the asset transfer.

- Identification of origin and destination DLT networks: The origin and destination DLT networks is assumed to have been identified.

- Selection of gateway: The two gateway at the origin and destination DLT networks respectively is assumed to have been identified and selected.

- Identification of gateway-owners (VASP): The VASP operating the gateway are assumed to have been identified and their status verified [FATF].

4. Interoperability Modes

Before delving into the architecture, it would be instructive to survey the different modes (or categories) of operations that necessitate interoperability between two blockchain/DLT network, virtual asset transfer being one such category.

We can reason about this in terms of the interdependencies between business processes in two independent systems. In one category, a ledger state update in one system depends on an update in the other. In other words, a write operation must be performed on both ledgers to maintain integrity of the collective system; if either ledger lies in a blockchain system or DLT network, a new block is also added. From this, one can infer that both writes, or ledger state updates, must occur atomically (either both happen or neither does) across both systems despite their independence and lack of a central coordinator.

The category of atomic writes can further be classified into asset transfers and asset exchanges. In the former, a virtual asset is expunged in one system while atomically being recreated in the other; the owner and recipient need only have accounts in their respective DLT networks. In the latter, two virtual assets are exchanged in two distinct networks atomically; they simply switch ownership without

their profile records leaving their respective systems' ledgers. In this scenario, both owners must have accounts in both networks.

Moving back up the categorization hierarchy, we can identify a different category in which a ledger state update in one system depends on already recorded state in the ledger of another. In other words, a write operation must be performed in one ledger after reading state from another. The use case under this category can be termed data transfer or data sharing, where the advancement of a business workflow in one system depends on the advancement of a different workflow in another without requiring an atomic operation across the two systems. Here, it is useful to distinguish data from asset; the former can be copied in and across systems without losing its integrity whereas the latter must have an unambiguous ownership record at all times.

```
                      Cross DLT-System Dependency
                                  |
                                  |
                +------------------------------------+
                |                                    |
                |                                    |
     +---------------------+              +---------------------+
     |    Bidirectional    |              |   Unidirectional    |
     | (Write <--> Write)  |              |  (Read --> Write)   |
     +---------------------+              +---------------------+
                |                                    |
                |                                    |
      +-------------------+                          |
      |                   |                          |
      |                   |                          |
+-----------------+  +-----------------+    +------------------+
| Asset Transfer  |  | Asset Exchange  |    |  Data Transfer   |
+-----------------+  +-----------------+    +------------------+
```

Figure 1

Though the rest of this document focuses on a gateway architecture to facilitate virtual asset transfers, the same architecture can also be used for asset exchanges and data transfers. Interested readers can find out more about cross DLT-system asset exchanges by referring to literature on Hashed Time Lock Contracts [HTLC21], on cross network data transfers [Abebe19][Abebe21], and on ledger state views and addresses [DLVIEW].

5. Architecture

5.1. Goal of Architecture

The goal of the interoperability architecture is to permit two (2) Gateways belonging to distinct DLT networks to conduct a virtual asset transfer between them, in a secure and non-repudiable manner while ensuring the asset does not exist simultaneously on both networks (double-spend problem).

The virtual asset as understood by the two gateway is a digital representation of value, expressed in an standard digital format in a way meaningful to the gateway syntactically and semantically.

The syntactic representation of the virtual asset between the two gateways need not bear any resemblance to the syntactic asset representation within their respective DLT networks.

The architecture recognizes that there are different DLT networks currently in operation and evolving, and that in many cases the interior technical constructs in these DLT networks maybe incompatible with one another.

The architecture therefore assumes that certain types of computer systems (i.e. gateway) will be equipped with an asset transfer protocol and with other relevant resources that permits greater interoperability across these DLT networks.

The resources within a DLT network (e.g. ledgers, public-keys, consensus protocols, etc.) are assumed to be opaque to external entities in order to permit a resilient and scalable protocol design that is not dependent on the interior constructs of particular blockchain system or DLT network. This ensures that the virtual asset transfer protocol between gateways is not conditioned or dependent on these local technical constructs. The role of a gateway therefore is also to mask (hide) the complexity of the interior constructs of the DLT network that it represents. Overall this approach ensures that a given DLT network operates as a true autonomous system.

The current architecture focuses on unidirectional asset transfers, although the building blocks in this architecture can be used to support protocols for bidirectional transfers (conditional two unidirectional transfers), atomic asset exchanges and data transfers.

For simplicity the current architecture employs two (2) gateways in the respective DLT networks, but collective multi-gateway transfers (i.e. multiple gateways at each side) [HS2019] may be developed based

on the building blocks and constructs identified in the current architecture.

5.2. Overview of Asset Transfer

An asset transfer between two DLT networks is carried out by two (2) gateway in a direct interaction (unmediated), where the gateway represents the two respective DLT networks.

A successful transfer results in the asset being extinguished (deleted) or marked on the origin ledger by the origin-gateway, and for the asset to be introduced by the destination-gateway into the destination ledger.

The mechanism to extinguish or introduce an asset from/into a ledger is dependent on the specific blockchain or DLT network. The task of the respective gateway is to implement the relevant mechanism to modify the ledger of their corresponding DLT networks in such a way that together the two DLT networks maintain consistency from the asset perspective, while observing the design principles of the architecture.

An asset transfer protocol that can satisfy the properties of atomicity and consistency in the case of two private DLT networks should also satisfy the same properties in the case when one or both are public.

5.3. Desirable Properties of Asset Transfer

The desirable features of asset transfers between two gateway include, but not limited, to the following:

- Atomicity: Transfer must either commit or entirely fail (failure means no change to asset ownership).

- Consistency: Transfer (commit or fail) always leaves the ledgers of both DLT networks to be in a consistent state (asset located in the ledger of one DLT network only).

- Isolation: While transfer occurring, asset ownership cannot be modified (no double-spend).

- Durability: Once a transfer has been committed, must remain so regardless of gateway crashes.

- Verifiable by authorized third parties: With proper authorization to access relevant interior resources, third party entities must be able at any time to perform audit-validation of the two

respective ledgers for asset transfers across the corresponding DLT networks.

- Containment of side-effects: Any effects due to errors or security/integrity breaches in a DLT network during an asset transfer must be contained within that network.

An implementation of the asset transfer protocol should satisfy these properties, independent of whether the implementation employs stateful messaging or stateless messaging between the two gateways.

5.4. Event log-data, crash recovery and backup gateways

Implementations of gateway should maintain event logs and checkpoints for the purpose of gateway crash recovery. The log-data generated by a gateway should be considered as an interior resource accessible to other authorized gateways within the same DLT network.

Mechanisms used to select or elect a gateway in a DLT network for a given asset transfer could be extended to include the selection of a backup gateways. The primary gateway and the backup gateway may or may not belong to the same owner (VASP).

Some DLT networks may utilize the ledger itself as means to retain the log-data, allowing other nodes in the DLT network to have visibility and access to the gateway log-data. Other DLT networks may employ off-chain storage that is accessible to all gateway in the same authorization domain. In such cases, to provide event-sequencing integrity the gateway may store a hash of the log- data on the ledger of the DLT network prior to writing the log-data to the off-chain storage.

The mechanism used to provide gateway crash-recovery is dependent on the DLT network and the gateway implementation. For interoperability purposes the information contained in the log and the format of the log-data should be standardized, permitting vendors of gateway products to reduce development costs over time. Similarly, in order to ensure a high degree of interoperability across crash-recovery protocol implementations [BCH21], a standardized interface (e.g. REST APIs) should be defined for read/ write access to the log-storage. The interface should hide the details of the log-storage from the gateway itself, and it should be independent of the gateway recovery strategy (e.g. self-healing, primary-backup, etc.).

The resumption of an interrupted transfer (e.g. due to gateway crash, network failure, etc.) should take into consideration the aspects of secure channel establishment and the aspects of the transfer protocol resumption. In some cases, a new secure channel (e.g. TLS session)

must be established between the two gateways, within which the asset transfer protocol could be continued from the last checkpoint prior to the interruption. However, in other cases both the secure channel and the transfer protocol may need to be started completely afresh (no resumption).

The log-data collected by a gateway acts also as a checkpoint mechanism to assist the recovered (or backup) gateway in continuing the transfer. The point at which to re-start the transfer protocol flow is dependent on the implementation of the gateway recovery strategy. Some owners (VASPs) of gateways may choose to start afresh the transfer of the asset, and not to resume partially completed transfers (e.g. for easier legal audit purposes).

5.5. Overview of the Phases in Asset Transfer

The interaction between two gateways in an asset transfer is summarized in Figure 2, where the origin DLT network is DLN1 and the destination network is DLN2. The gateways are denoted as G1 and G2 respectively.

```
            Originator                                   Beneficiary
                |                                             |
         +-------------+                               +-------------+
         |   Client    |                               |   Client    |
         | Application |                               | Application |
         |    (App1)   |                               |    (App2)   |
         +-------------+                               +-------------+
                |                                             |
                |                  (Phases)                   |
                V                                             V
         +-------------+       |<-----(1)----->|       +-------------+
         | DLT Network |  +----+               +----+  | DLT Network |
         |     DLN1    |  |Gate|               |Gate|  |     DLN2    |
         |             |--|way |<-----(2)----->|way |--|             |
         | +---------+ |  | G1 |               | G2 |  | +---------+ |
         | |Ledger L1| |  +----+               +----+  | |Ledger L2| |
         | +---------+ |       |<-----(3)----->|       | +---------+ |
         +-------------+                               +-------------+
```

Figure 2

The phases are summarized as follows.

- Phase 0: Initiation of transfer at the application layer. The two applications utilized by the originator and beneficiary is assumed to interact as part of the asset transfer. In this phase, the applications App1 and App2 may establish some context information (e.g. Session-ID) that will be made available to their respective gateways G1 and G2. The legal verification of the identities of the Originator and Beneficiary may occur in this phase [FATF]. This phase is outside the scope of the current architecture.

- Phase 1: Pre-transfer Verification of Asset and Identities. In this phase the gateways G1 and G2 must mutually identify themselves and authenticate that both possess gateway-capabilities. Gateway G1 must communicate to G2 the asset-profile of the asset to be transferred, while G2 must validate that it has the ability to support this type of asset in the ledger of its DLT network.

- Phase 2: Evidence of asset locking or escrow. In this phase, gateway G1 must provide gateway G2 with sufficient evidence that the asset on DLN1 is in a locked state (or escrowed) under the control of G1 on ledger L1, and safe from double-spend by its current owner (the originator).

- Phase 3: Transfer commitment. In this phase gateways G1 and G2 commit to the unidirectional asset transfer.

These phases will be further discussed below.

6. Pre-transfer Verification of Asset and Identities (Phase 1)

The primary purpose of the first phase is to verify the various information relating to the asset to be transferred, the correct identities of the originator and beneficiary (as provided by the respective applications), the identity and legal status of the entities (VASPs) who own and operate the gateways, the type of the DLT network, and network parameters, and the device-identities of the gateways.

```
      Orig   L1          G1                    G2           L2  Benef
        |     |            |                    |            |     |
        |---request------->|                    |            |     |
        |     |            |                    |            |     |
      ..|.....|............|....................|............|.....|..
        |     |            |       Phase 1      |            |     |
        |     |            |                    |            |     |
        |     |       (1.1)|<-----VASP id------>|            |     |
        |     |            |                    |            |     |
        |     |            |                    |            |     |
        |     |       (1.2)|<--Asset Profile--->|            |     |
        |     |            |                    |            |     |
        |     |            |                    |            |     |
        |     |       (1.3)|<--Orig/Benef id--->|            |     |
        |     |            |                    |            |     |
      ..|.....|............|....................|............|.....|..
        |     |            |                    |            |     |
```

Figure 3

This phase starts with the assumption that in DLN1 the gateway to process the asset transfer to DLN2 has been selected (namely gateway G1). It also assumes that the destination DLN2 has been identified where the beneficiary address is located, and that gateway G2 in DLN2 has been identified that will peer with G1 to perform the transfer.

There are several steps that may occur in Phase 1:

- Secure channel establishment between G1 and G2: This includes the mutual verification of the gateway device identities and the exchange of the relevant parameters for secure channel establishment. In cases where device attestation [RATS] is required, the mutual attestation protocol must occur between G1 and G2 prior to proceeding to the next phase.

- Mutual device attestations: In cases where device attestation [RATS] is required, each gateway must yield attestation evidence to the other regarding its configuration. A gateway may take on the role as a attestation verifier, or it may rely on an external verifier to appraise the received evidence.

- Validation of the gateway ownership: There must be a means for gateway G1 and G2 to verify their respective ownerships (i.e. VASP1 owning G1 and VASP2 owning G2 respectively). Examples of ownership verification mechanism include the chaining of the gateway-device X.509 certificate up to the VASP entity certificate, directories of gateways and VASPs, and others.

- Validation of VASP status: In some jurisdictions limitations may be placed for regulated VASPs to transact only with other similarly regulated VASPs. Examples of mechanisms used to validate a VASP legal status include VASP directories, Extended Validation (EV) X.509 certificates for VASPs, and others.

- Identification and validation of asset profile: Both gateways must agree on the type of asset being transferred based on the profile of the asset. Gateway G1 must communicate the asset-profile identification to gateway G2, who in turn must validate both the legal status of the asset as well as the technical capability of DLN2 to digitally represent the asset type within its ledger L2. The policies governing DLT network DLN2 with regards to permissible incoming assets must be enforced by G2.

- Exchange of Travel Rule information and validation: In jurisdictions where the Travel Rule policies regarding originator and beneficiary information is enforced [FATF], the owners of gateways G1 and G2 must comply to the Travel Rule. Mechanisms must be used to permit gateways G1 and G2 to make available originator/beneficiary information to one another in such away that the Travel Rule information can be logged as part of the asset transfer history.

- Negotiation of asset transfer protocol parameters: Gateway G1 and G2 must agree on the parameters to be employed within the asset transfer protocol. Examples include endpoints definitions for resources, type of commitment flows (e.g. 2PC or 3PC), lock-time durations, and others [ODAP].

7. Evidence of asset locking or escrow (Phase 2)

The asset transfer protocol can commence when both gateways G1 and G2 have completed the verifications in Phase 1.

The steps of Phase 2 are summarized in Figure 4, and broadly consists of the following:

- Commencement (2.1): Gateway G1 indicates the start of the asset transfer protocol by sending a transfer-commence message to gateway G2. Among others, the message must include a cryptographic hash of the information agreed-upon in Phase 1 (e.g. asset profile, gateway identities, originator/beneficiary public keys, etc.).

- Acknowledgement (2.2): The gateway G2 must send an explicit acknowledgement of the receipt of the commence message, which

should include a hash of commencement message (2.1) and other relevant session parameters.

- G1 lock/escrow asset (2.3): Gateway G1 proceeds to lock or escrow the asset belonging to the originator on ledger L1. This prevents other transactions from changing the state of the asset in L1 until such time the lock by G1 is finalized or released. A time-lock or escrow may also be employed. The mode of the escrow may depend on the fundamental ledger architecture of the respective DLN1 and DLN2 in question (e.g. account-based, UTXO, or other).

- G2 logs incoming asset (2.4): Gateway G2 correspondingly writes a non-committing log (passive transaction) on ledger L2 indicating an imminent arrival of the asset to L2. This may act as a notification for the beneficiary regarding the asset transfer.

- Lock Evidence (2.5): Gateway G1 sends a digitally signed evidence regarding the lock (escrow) performed by G1 on the asset on ledger L1. The signature by G1 is performed using its entity public-key pair. This signifies that G1 (i.e. its owner VASP) is legally standing behind its assertion regarding the lock/escrow on the asset performed by G1.

- Evidence receipt (2.6): If gateway G2 accepts the evidence, G2 then responds with a digitally signed receipt message which includes a hash of the previous lock-evidence message. Otherwise, if G2 declines the evidence then G2 can ignore the transfer and let it time-out (i.e. transfer failed). The signature by G2 is performed using its entity public-key pair.

```
      Orig  L1          G1                    G2          L2  Benef
       |     |            |      (Phase 1)     |            |     |
       |     |            |                    |            |     |
     ..|.....|............|....................|............|.....|..
       |     |            |       Phase 2      |            |     |
       |     |            |                    |            |     |
       |     |       (2.1)|-----Commence------>|            |     |
       |     |            |                    |            |     |
       |     |            |<-------ACK---------|(2.2)       |     |
       |     |            |                    |            |     |
       |     |            |                    |            |     |
       |     |<---Lock----|(2.3)               |            |     |
       |     |            |               (2.4)|----Log---->|     |
       |     |            |                    |            |     |
       |     |            |                    |            |     |
       |     |       (2.5)|---Lock Evidence--->|            |     |
       |     |            |                    |            |     |
       |     |            |<-----Receipt-------|(2.6)       |     |
       |     |            |                    |            |     |
     ..|.....|............|....................|............|.....|..
       |     |            |                    |            |     |
```

Figure 4

The precise form of the evidence in step 2.5 is dependent on the type of ledger technology in DLN1, and must be previously agreed upon between G1 and G2 in Phase 1.

The purpose of this evidence is for dispute resolution between G1 and G2 (i.e. the VASP entities who own and operate G1 and G2 respectively) in the case that double-spend is later detected.

The gateway G2 must return a digitally signed receipt to G1 of this evidence in order to cover G1 (exculpatory proof) in the case of later denial by G2.

8. Transfer Commitment (Phase 3)

In Phase 3 the gateways G1 and G2 finalizes to the asset transfer by performing a commitment protocol (e.g. 2PC or 3PC) as a process (sub-protocol) embedded within the overall asset transfer protocol.

Upon receiving the evidence-receipt message in the previous phase, G1 begins the commitment (see Figure 5):

- Commit-prepare (3.1): Gateway G1 indicates to G2 to prepare for the commitment of the transfer. This message must include a hash of the previous messages (message 2.5 and 2.6).

- Ack-prepare (3.2): Gateway G2 acknowledges the commit-prepare message.

- Lock-final (3.3): Gateway G1 issues a lock-finalization transaction or escrow finalization on ledger L1 that signals the permanent extinguishment of the asset from DLN1. This transaction must include a hash reference to the lock transaction on L1 previously in step (2.3). This indicates that the asset is no longer associated with the public-key of its previous owner (originator) and that the asset instance is no longer recognized on the ledger L1.

- Commit-final (3.4): Gateway G1 indicates to G2 that G1 has performed a local lock/escrow finalization on L1. This message must be digitally signed by G1 and should include the block number and transaction number (of the confirmed block) on ledger L1.

- Asset-create (3.5): Gateway G2 issues a transaction on ledger L2 to create (re-generate) the asset, associated with the public-key of the beneficiary. This transaction must include a hash of the previous message (3.4) and hash reference to the log-incoming transaction on L2 previously in step (2.4). These hash references connects the newly re-generated asset with the overall transfer event originating from gateway G1.

- Ack-final (3.6): Gateway G2 indicates to G1 that G2 has performed an asset-regeneration transaction on L2. This message must be digitally signed by G2 and should include the block number and transaction number (of the confirmed block) on ledger L2.

- Location-record (3.7): Gateway G1 has the option to record the block number and transaction number (as reported by G2 in the previous step) to ledger L1. This transaction should include a hash reference to the confirmed lock-finalization transaction on L1 from step 3.3. This information may aid in future search, audit and accountability purposes from a legal perspective.

- Transfer complete (3.8): Gateway G1 must explicitly close the asset transfer session with gateway G2. This allows both sides to close down the secure channel established in Phase 1.

```
      Orig  L1          G1                   G2          L2  Benef
      |     |            |      (Phase 2)     |            |     |
      |     |            |                    |            |     |
    ..|.....|............|....................|............|.....|..
      |     |            |       Phase 3      |            |     |
      |     |            |                    |            |     |
      |     |       (3.1)|--Commit Prepare--->|            |     |
      |     |            |                    |            |     |
      |     |            |<-----ACK-Prep------|(3.2)       |     |
      |     |            |                    |            |     |
      |     |            |                    |            |     |
      |     |<--Final----|(3.3)               |            |     |
      |     |            |                    |            |     |
      |     |       (3.4)|----Commit Final--->|            |     |
      |     |            |                    |            |     |
      |     |            |               (3.5)|---Create-->|     |
      |     |            |                    |            |     |
      |     |            |<-----ACK-Final-----|(3.6)       |     |
      |     |            |                    |            |     |
      |     |            |                    |            |     |
      |     |<--Record---|(3.7)               |            |     |
      |     |            |                    |            |     |
      |     |       (3.8)|---Complete/End---->|            |     |
      |     |            |                    |            |     |
    ..|.....|............|....................|............|.....|..
      |     |            |                    |            |     |
```

Figure 5

9. Related Open Issues

There are a number of open issues that are related to the asset transfer protocol between gateways. Some of the issues are due to the fact that blockchain technology is relatively new, and that technical constructs designed for interoperability have yet to be addressed. Some of the issues are due to the nascency of the virtual asset industry and lack of conventions, and therefore require industry collaboration to determine the standard conventions.

9.1. Global identification of blockchain systems and public-keys

There is currently no standard nomenclature to identify a DLT network in a globally unique manner. The analog to this is the AS-numbers associated with IP routing autonomous systems.

Furthermore, an address (public-key) may not be unique to one DLT network. An entity (e.g. user) may in fact employ the same public-key simultaneously at multiple distinct DLT networks. Thus, there is no convention today with regards to the application of a key within a given DLT network (comparable to the principal/ domain convention in Internet host naming).

However, in order to perform an asset transfer from one DLT network to another, there needs to be mechanism that resolves the beneficiary identifier (as known to the originator) to the correct public-key and DLT network as intended by the originator.

9.2. Discovery of gateways in DLT Networks

A given DLT network must possess the capability to select or designate gateway that will perform an asset transfer.

A number of DLT networks already employ consensus mechanisms that elect a gateway to perform the transaction processing (e.g. proof of stake in Ethereum). The same consensus mechanisms may be used to elect the gateway (e.g. out of a pool of available gateways in the DLT network).

9.3. Remote gateway discovery

Related to the ability to discover other DLT networks globally is the ability to discover the remote gateways for these other DLT networks. A discovery mechanism for external entities (e.g. for gateway G1) to look for gateways (e.g. remote gateway G2) is required in order for gateways to quickly and efficiently peer without human intervention. The discovery mechanism may employ the available information at gateway G1, such as the originator/beneficiary public keys, the VASPs (owners of the gateways) and other parameters.

Other approaches may also be employed, such as incorporating the gateway identities within a VASP's configuration file (e.g. at a well-known location), and within a global directory of regulated VASPs. Approaches similar to the DNS infrastructure may provide an alternative architecture for solving this problem.

9.4. Commitment protocols and forms of commitment evidence

Within Phase 2, the gateways must implement one (or more) transactional commitment protocols that permit the coordination between two gateways, and the final commitment of the asset transfer.

The choice of the commitment protocol (type/version) and the corresponding commitment evidence must be negotiated between the gateways during Phase 1.

For example, in Phase 2 and Phase 3 discussed above the gateways G1 and G2 may implement the classic 2 Phase Commit (2PC) protocol [Gray81] as a means to ensure efficient and non-disputable commitments to the asset transfer.

Historically, transactional commitment protocols employ locking mechanisms to prevent update conflicts on the data item in question. When used within the context of virtual asset transfers across DLT networks, the fact that an asset has been locked by G1 (as the 2PC coordinator) must be communicated to G2 (as the 2PC participant) in an indisputable manner.

The exact form of this evidence of asset-locking must be standardized (for the given transactional commitment protocol) to eliminate any ambiguity.

10. Security Considerations

As a DLT network hold an increasing number of virtual assets, it may become attractive to attackers seeking to compromise the cryptographic keys of the entities, services and its end-users.

Gateways are of particular interest to attackers because they enable the transferal of virtual assets to external DLT networks, which may or may not be regulated. As such, hardening technologies and tamper-resistant crypto-processors (e.g. TPM, SGX) should be used for implementations of gateways [HS19].

11. Policy Considerations

Virtual asset transfers must be policy-driven in the sense that it must observe and enforce the policies defined for the DLT network. Resources that make-up a DLT network are owned and operated by entities (e.g. legal persons or organizations), and these entities typically operate within regulatory jurisdictions [FATF]. It is the responsibility of these entities to translate regulatory policies into functions on DLT networks that comply to the relevant regulatory policies.

At the application layer, asset transfers must take into consideration the legal status of assets and incorporate relevant asset-related policies into their business logic. These policies must permeate down to the gateways that implement the functions of asset transaction processing.

The smart contract abstraction, based on replicated shared code/state on the ledger [Herl19], must additionally incorporate the notion of policy into the abstraction.

12. References

12.1. Normative References

[BCH21] Belchior, R., Correia, M., and T. Hardjono, "DLT Gateway Crash Recovery Mechanism, IETF, draft-belchior-gateway-recovery-01.", March 2021, <https://datatracker.ietf.org/doc/draft-belchior-gateway-recovery/>.

[DLVIEW] Ramakrishna, V., Pandit, V., Nishad, S., Narayanam, K., and D. Vinayagamurthy , "Views and View Addresses for Blockchain/DLT Interoperability, IETF Draft", November 2021.

[FATF] FATF, "International Standards on Combating Money Laundering and the Financing of Terrorism and Proliferation - FATF Revision of Recommendation 15 (Updated June 2021)", October 2018, <http://www.fatf-gafi.org/publications/fatfrecommendations/documents/fatf-recommendations.html>.

[ISO] ISO, "Blockchain and distributed ledger technologies-Vocabulary (ISO:22739:2020)", July 2020, <https://www.iso.org>.

[NIST] Yaga, D., Mell, P., Roby, N., and K. Scarfone, "NIST Blockchain Technology Overview (NISTR-8202)", October 2018, <https://doi.org/10.6028/NIST.IR.8202>.

[ODAP] Hargreaves, M. and T. Hardjono, "Open Digital Asset Protocol, IETF, draft-hargreaves-odap-01.", November 2020, <https://datatracker.ietf.org/doc/draft-hargreaves-odap/>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

12.2. Informative References

[ABCH20] Ankenbrand, T., Bieri, D., Cortivo, R., Hoehener, J., and T. Hardjono, "Proposal for a Comprehensive Crypto Asset Taxonomy", May 2020, <https://arxiv.org/abs/2007.11877>.

[Abebe19] Abebe, E., Behl, D., Govindarajan, C., Hu, Y., Karunamoorthy, D., Novotny, P., Pandit, V., Ramakrishna, V., and C. Vecchiola, "Enabling Enterprise Blockchain Interoperability with Trusted Data Transfer (Middleware 2019, Industry Track)", December 2019, <https://arxiv.org/abs/1911.01064>.

[Abebe21] Abebe, E., Hu, Y., Irvin, A., Karunamoorthy, D., Pandit, V., Ramakrishna, V., and J. Yu, "Verifiable Observation of Permissioned Ledgers (ICBC2021)", May 2021, <https://arxiv.org/abs/2012.07339>.

[BVGC20] Belchior, R., Vasconcelos, A., Guerreiro, S., and M. Correia, "A Survey on Blockchain Interoperability: Past, Present, and Future Trends", May 2020, <https://arxiv.org/abs/2005.14282v2>.

[Clar88] Clark, D., "The Design Philosophy of the DARPA Internet Protocols, ACM Computer Communication Review, Proc SIGCOMM 88, vol. 18, no. 4, pp. 106-114", August 1988.

[Gray81] Gray, J., "The Transaction Concept: Virtues and Limitations, in VLDB Proceedings of the 7th International Conference, Cannes, France, September 1981, pp. 144-154", September 1981.

[Herl19] Herlihy, M., "Blockchains From a Distributed Computing Perspective, Communications of the ACM, vol. 62, no. 2, pp. 78-85", February 2019, <https://doi.org/10.1145/3209623>.

[HLP19] Hardjono, T., Lipton, A., and A. Pentland, "Towards and Interoperability Architecture for Blockchain Autonomous Systems, IEEE Transactions on Engineering Management", June 2019, <https://doi:10.1109/TEM.2019.2920154>.

[HS2019] Hardjono, T. and N. Smith, "Decentralized Trusted Computing Base for Blockchain Infrastructure Security, Frontiers Journal, Special Issue on Blockchain Technology, Vol. 2, No. 24", December 2019, <https://doi.org/10.3389/fbloc.2019.00024>.

[IDevID] Richardson, M. and J. Yang, "A Taxonomy of operational security of manufacturer installed keys and anchors. IETF draft-richardson-t2trg-idevid-considerations-01", August 2020, <https://tools.ietf.org/html/draft-richardson-t2trg-idevid-considerations-01>.

[SRC84] Saltzer, J., Reed, D., and D. Clark, "End-to-End Arguments in System Design, ACM Transactions on Computer Systems, vol. 2, no. 4, pp. 277-288", November 1984.

Authors' Addresses

Thomas Hardjono
MIT

Email: hardjono@mit.edu

Martin Hargreaves
Quant Network

Email: martin.hargreaves@quant.network

Ned Smith
Intel

Email: ned.smith@intel.com

Venkatraman Ramakrishna
IBM

Email: vramakr2@in.ibm.com