Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 30, 2021

T. Hardjono
MIT
M. Hargreaves
Quant Network
N. Smith
Intel

October 27, 2020

An Interoperability Architecture for Blockchain Gateways
draft-hardjono-blockchain-interop-arch-01

Abstract

With the increasing interest in the potential use of blockchain systems for virtual assets, there is a need for these assets to have mobility across blockchain systems. An interoperability architecture for blockchain systems is needed in order to permit the secure flow of virtual assets between blockchain systems, satisfying the properties of transfer atomicity, consistency and durability. The architecture must recognize that there are different blockchain systems, and that the interior constructs in these blockchains maybe incompatible with one another. Gateway nodes perform the transfer of virtual assets between blockchain systems while masking the complexity of the interior constructs of the blockchain that they represent.

Table of Contents

1. Introduction

Currently there is little technical interoperability between blockchain systems. This results in the difficulty in transferring or migrating virtual assets associated with a public-key (address) in one blockchain system to another blockchain system.

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 delays in transfers, but also in the need for key-holders to open accounts at the third party entity. It diminishes the autonomy of a blockchain system and limits its scalability.

This document describes an architecture for blockchain gateways that perform the unidirectional transfer of virtual assets between two autonomous blockchain systems which employ the gateways.

The purpose of this architecture document is to provide technical framework within which to discuss the various aspects of a transfer between two gateways, including security aspects and transfer commitment aspects.

2. Terminology

There following are some terminology used in the current document:

- Blockchain Domain: The collection of resources and entities participating within a blockchain system.

- Interior Resources: The various interior protocols, data structures and cryptographic constructs that are a core part of a blockchain system. 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 system, and are not part of the operations of a blockchain. Examples include data located at third parties such as asset registries, ledgers of other blockchains, PKI infrastructures, etc.

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

- Gateway nodes: The nodes of the blockchain system that are functionally capable of acting as a gateway in an asset transfer. Gateway nodes implement the gateway-to-gateway asset transfer protocol. Being a node on the blockchain system, a gateway has read/write access to the interior resources (e.g. ledger) of the blockchain. It participates in the consensus mechanism deployed for the blockchain system. Depending on the blockchain system implementation, some or all of the nodes may be gateway-capable.

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

- Entity public-key pair: This the private-public key pairs of an entity used for interactions outside the blockchain system (e.g. TL1.3). 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 gateway nodes to perform a unidirectional transfer of a virtual asset.

- Asset profile: The prospectus of a regulated asset that includes information and resources describing the virtual asset. This includes 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 blockchain 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].

- 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 node within a blockchain system.

- Passive transaction: A transaction aimed at recording some state metadata information on the ledger that does not affect assets recorded on the ledger. A passive transaction can be self-addressed (or has null as destination address) and can be used to signal implicitly to other nodes regarding an state-change of the metadata pertaining to an entity or an asset on the ledger.

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 interoperability architecture, and correspond to the design principles of the Internet architecture.

3.1. Design Principles

- Opaque blockchain resources: The interior resources of each blockchain system 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 node 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) blockchain systems 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 autonomous systems.

The value-externalization principle permits asset transfer protocols to be designed for efficiency, speed 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 blockchain systems are assumed to have a common agreement regarding the economic value of the asset.

3.2. Operational Assumptions

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

- Application layer transfer request: The transfer request from an originator in the origin blockchain 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 blockchain: The origin and destination blockchain systems is assumed to have been identified.
- Selection of gateway nodes: The two gateway nodes at the origin and destination blockchain systems respectively is assumed to have been selected.
- Identification of gateway-node owners (VASP): The VASP operating the gateway nodes are assumed to have been identified and their legal status verified.

4. Architecture

4.1. Goal of Architecture

The goal of the interoperability architecture is to permit two (2) gateway nodes belonging to distinct blockchain systems 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 blockchains (double-spend problem).

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

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

The architecture therefore assumes that certain types of nodes (gateway nodes) will be equipped with an asset transfer protocol and other relevant resources that permits greater interoperability across these incompatible blockchain systems.

The resources within a blockchain system (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 systems. 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 blockchain system that it represents. Overall this approach ensures that a given blockchain system 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).

For simplicity the current architecture employs two (2) gateway nodes in the respective blockchains, but collective multi-node transfers (i.e. multiple nodes at each side) [HS2019] may be developed based on the building blocks and constructs identified in the current architecture.

4.2. Overview of Asset Transfer

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

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 system. The task of the respective gateway is to implement the relevant mechanism to modify the ledger of their corresponding blockchain system in such a way that together the two blockchains 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 blockchain systems, should also do in the case when one or both are public blockchain systems.

4.3. Desirable Properties of Asset Transfer

The desirable features of asset transfers between two gateway nodes 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 both blockchains in a consistent state (asset located in one blockchain 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 blockchain systems.
- Containment of side-effects: Any effects due to errors or security/integrity breaches in a blockchain system during an asset transfer must be contained within that blockchain.

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.

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

Implementations of gateway nodes 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 gateway nodes within the same blockchain system

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

Some blockchain systems may utilize the ledger itself as means to retain the log-data, allowing other nodes in the blockchain to have visibility and access to the gateway log-data. In these cases, the gateway node may employ its transaction-signing key pair to issue a passive transaction (e.g. self-addressed, no asset) on the ledger, incorporating details of the transfer event.

Other blockchain systems may employ off-chain storage that is accessible to all gateway nodes in the blockchain domain. In such cases, to provide event-sequencing integrity the gateway may store a hash of the log-data on the ledger of the blockchain (passive transaction) prior to writing the log-data to the off-chain storage.

The mechanism used to provide gateway crash-recovery is dependent on the blockchain system 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.

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 with the backup gateway node, 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 must be started completely afresh (no resumption).

The log-data collected by a gateway node acts also as a checkpoint mechanism to assist the backup gateway node in continuing the transfer. The point at which to re-start the transfer protocol flow is dependent on the implementation of the gateway. Some owners (VASPs) of gateway nodes may choose to start afresh the transfer of the asset, and not to resume partially completed transfers.

4.5. Overview of the Phases in Asset Transfer

The interaction between two gateways in an asset transfer is summarized in Figure 1, where the origin blockchain is B1 and the destination blockchain is B2. The gateways are denoted as G1 and G2 respectively.

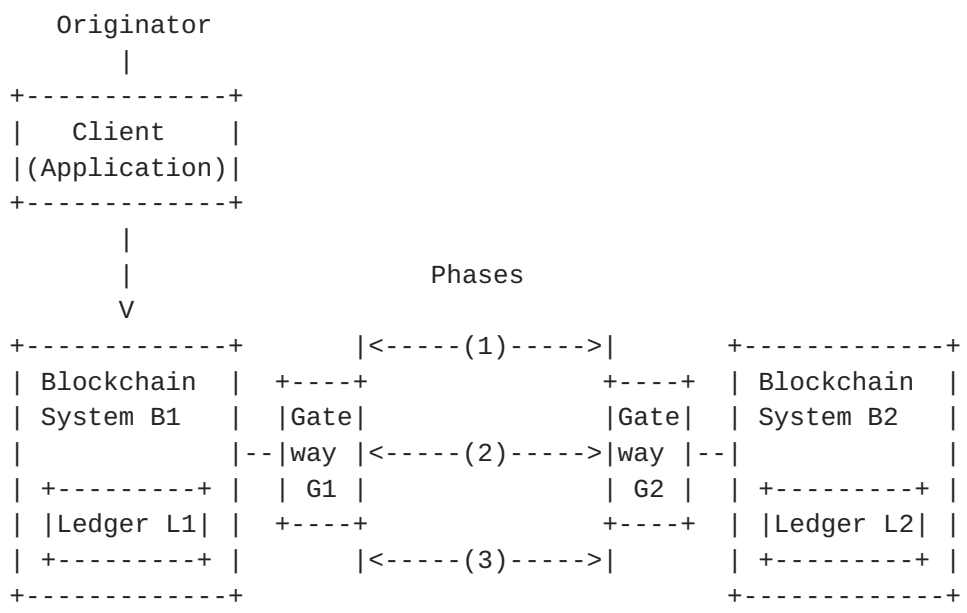

Figure 1

The three phases are summarized as follows.

- 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, and that consent have been obtained from the beneficiary regarding accepting the transfer.

- Phase 2: Evidence of asset locking or escrow. In this phase, gateway G1 must provide gateway G2 with sufficient evidence that the asset on blockchain B1 is in a locked state (or escrowed) under the control of G1 on ledger L1, and safe from double-spend on the part of its current owner (the originator).

- Phase 3: Transfer commitment. In this phase gateways G1 and G2 commits to the transaction

These phases will be further discussed below.

5. 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 identities of the originator and beneficiary, the identity and legal status of the entities (VASPs) who own and operate the gateways, and the device-identities of the gateways.

This phase starts with the assumption that in blockchain B1 the gateway to process the asset transfer to B2 has been selected (namely gateway G1). It also assumes that the destination blockchain B2 has been identified where the beneficiary address is located, and that gateway G2 in blockchain B2 has been identified that will peer with G1 to perform the transfer.

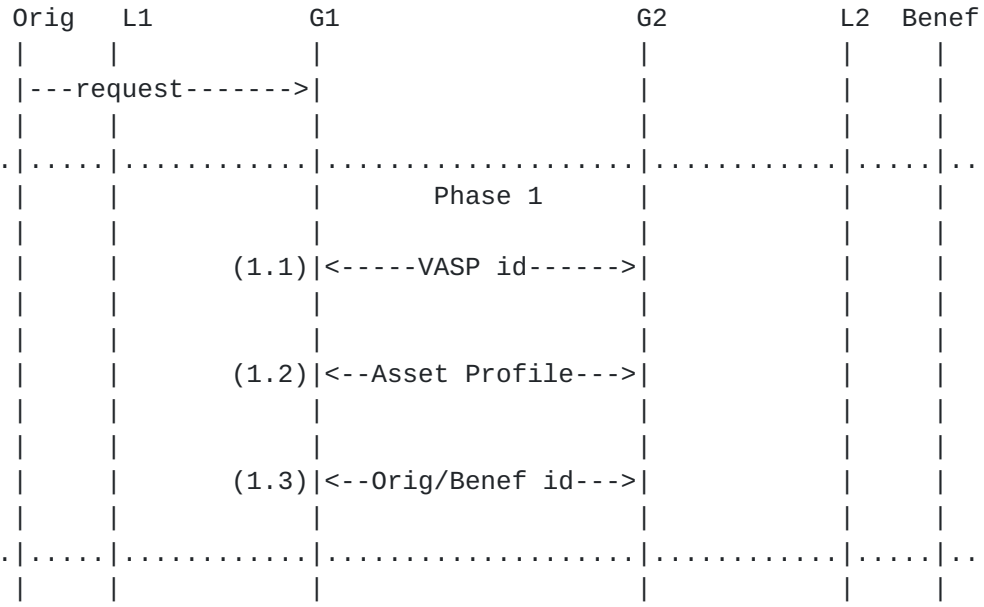

Figure 2

There are several steps that may occur in Phase 1 (see Figure 2):

- 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.

- 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.

- Delivery and validation of asset profile: Gateway G1 must deliver to G2 the asset-profile for the virtual asset to be transferred. Gateway G2 must validate the authenticity of the statements (claims) found in the asset profile. The policies governing blockchain B2 with regards to permissible incoming assets must be enforced by G2.

- Exchange of Travel Rule information: In jurisdictions where the Travel Rule policies regarding originator and beneficiary information is enforced, the gateways G1 and G2 must exchange this information [FATF].

- o 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].

6. 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 is shown in Figure 3, 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).

- Acknowledgement (2.2): The gateway G2 must send an explicit acknowledgement of the commence message, which should include a hash of commencement message 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. The lock may take the form of passive transaction that signals to other nodes that the asset is temporarily inaccessible. This signals to other nodes in the blockchain system to ignore other transactions

pertaining to the asset until such time the lock by G1 is finalized or released.

- G2 log incoming (2.4): Gateway G2 correspondingly writes a 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.

- 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).

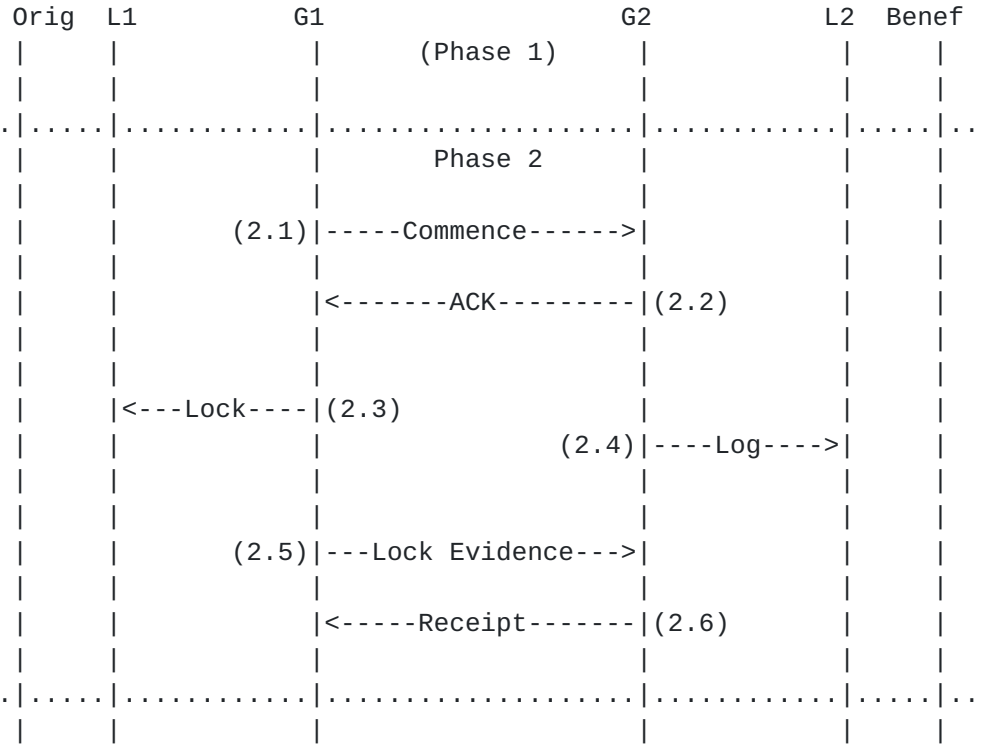

Figure 3

The precise form of the evidence in step 2.5 is dependent on the blockchain system in B1, 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. 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.

7. 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 4):

- 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 (passive) on ledger L1 that signals the permanent extinguishment of the asset. 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-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 issues a transaction on ledger L2 to create 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 created 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-creation 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, using a passive transaction. 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.

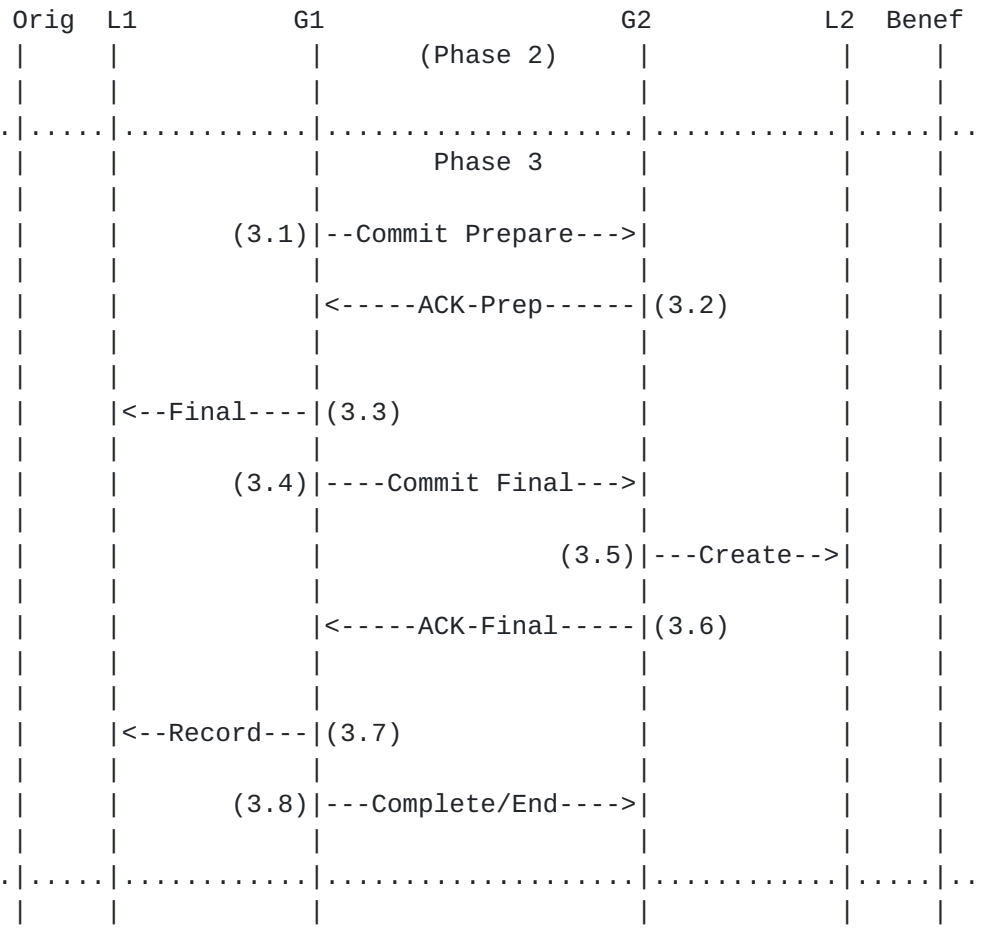

Figure 4

8. Related Open Issues

There are a number of open issues that are related to the asset transfer protocol between gateway nodes. 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 these.

8.1. Global identification of blockchain systems and public-keys

There is currently no standard nomenclature to identify blockchain systems 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 blockchain system. An entity (e.g. user) may in fact employ the same public-key at multiple distinct blockchain systems simultaneously.

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

8.2. Selection of gateways nodes within a blockchain system

A given blockchain system must possess the capability to select or designate gateway nodes that will perform an asset transfer across blockchain systems.

A number of blockchain systems already employ consensus mechanisms that elect a node to perform the transaction processing (e.g. proof of stake in Ethereum). The same consensus mechanisms may be used to elect the gateway node.

However, there are some blockchain systems that do not elect a single node and which employ a race-to-process strategy (e.g. proof of work in Bitcoin). Since the winner of the proof of work can be any node in the blockchain system, this implies that all the nodes in these types of blockchains must be gateway-capable.

8.3. Commitment protocols and forms of commitment evidence

Within Phase 2, the gateway nodes 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 blockchain systems, 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.

9. Security Considerations

Although the current interoperability architecture for blockchain gateways assumes the externalization of the value of assets, as a blockchain system holds an increasing number of virtual assets it becomes attractive to attackers seeking to obtain cryptographic keys of its nodes and its end-users.

Gateway nodes are of particular interest to attackers because they enable the transferal of virtual assets to external blockchain systems, 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].

10. Policy Considerations

Virtual asset transfers must be policy-driven in the sense that it must observe and enforce the policies defined for the blockchain domain. Resources that make-up a blockchain systems 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 blockchain systems 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 nodes 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.

11. References

11.1. Normative References

[FATF] FATF, "International Standards on Combating Money Laundering and the Financing of Terrorism and Proliferation - FATF Revision of Recommendation 15", 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, October 2020, IETF, draft-hargreaves-odap-00.", October 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>.

11.2. 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>.

[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