Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 6 June 2023

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

4 January 2023

Secure Asset Transfer (SAT) Interoperability Architecture
draft-hardjono-sat-architecture-02

Abstract

This document proposes an interoperability architecture for the secure transfer of assets between two networks or systems based on the gateway model.

Table of Contents

1. Introduction

This document proposes an interoperability architecture based on gateways, which are points of interconnection between networks or systems.

There are several services that may be offered by a gateway, one of which being the direct transfer of a digital asset from one network to another via pairs of gateways without a mediating third party.

A given network or system may have one or more gateways to perform a unidirectional direct transfer of digital assets to another network possessing one or more compatible gateway.

Both gateways must implement a secure 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 gateway that supports the secure asset transfer protocol.

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:

* Asset network (system): The network or system where a digital asset is utilized.

* Asset Transfer Protocol: The protocol used to transfer (move) a digital asset from one network to another using gateways.

* Origin network: The current network where the digital asset is located.

* Destination network: The network to which a digital asset is to be transferred.

* Resource Domain: The collection of resources and entities participating within an asset network. The domain denotes a 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 an asset network or system.

* Exterior Resources: The various protocols, data structures and cryptographic constructs that are outside of (external to) the network or system.

* Gateway: The collection of services which connects to a minimum of one network or system, and which implements the secure asset transfer protocol.

* Entity public-key pair: This the private-public key pairs of an entity, where the public-key is available and verifiable outside the network. Among others, it may be utilized for interactions

other entities from outside the network. The term is used to distinguish this public-key from other key-pairs belonging to the same entity, but which is only available within the (private) network.

* Originator: Person or organization in an origin network seeking the transfer of a digital asset to a beneficiary located in a remote network.

* Beneficiary: Person or organization in an destination network seeking to receive the transfer of a digital asset to from an originator located in a remote network.

* 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 entity that owns and operates a gateway within a network.

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 network resources: The interior resources of each 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 asset transfer protocol is agnostic (oblivious) to the economic or monetary value (if any) of the digital asset being transferred.

The opaque resources principle permits the architecture to be applied in cases where one (or both) networks are private (closed membership). 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 an asset transfer protocol to be designed for efficiency, security and reliability -- independent of the changes in the perceived economic value of the digital asset. It is the analog of the end-to-end principle in the Internet architecture [SRC84], where contextual information is placed at the endpoints of the transfer.

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 network is assumed to have occurred prior to the execution of the asset transfer protocol by the gateways.

* 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 asset networks: The origin and destination networks is assumed to have been identified.

* Selection of gateway: The two corresponding gateways at the origin and destination networks is assumed to have been identified and selected.

3.3. Assumptions Regarding Gateway Operators

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

* Identification of gateway-owners: The owners of the two corresponding gateways are assumed to have been identified and their ownership status verified.

* Gateway liabilities: Gateways and gateway-operators are assumed to take on legal and financial liability for their transactions, and gateways are assumed to operate under a well-defined legal framework (e.g. contractual relationship). Furthermore, the legal framework is assumed to be supported by compatible legislation in the relevant jurisdictions where the gateways are operating.

* Gateway message signatures: All messages between gateways are assumed to be signed and verified (e.g. X.509).

* Transitory ownership of asset by gateway: Assets being transferred via SAT will be technically be owned by gateway in transit and gateways are liable for them while they have ownership.

* Network data: Gateways are assumed to have mechanisms in place to trust data returned from their local networks. This will depend on the technical architecture and capabilities of each specific network.

* Gateways are trusted: The gateways are assumed to be trusted to carry-out all the stages of the protocol described in this architecture.

4. Gateway Interoperability Modes

The current interoperability architecture based on gateways recognizes several types of transfer flows:

* Asset transfer: This refers to the transfer of a digital asset from the origin network to a destination network, where a successful asset transfer causes the asset to be extinguished in the origin network and be created (generated) at the destination network.

* Data sharing (transfer): This refers to the sharing of data by a network with an authorized external entity, which itself may be a network, in such a way that the source and veracity of the data can be verified by that external party. This data can consist of information about the state or properties of a digital asset or the state of a business workflow governing one or more digital assets. The data sharing mode of interoperability addresses the use-cases where a state update in one network depends on the existence of state information recorded in a different network. In general, this mode can be used to sync state across independent ledgers in a controllable manner, create a trustworthy and secure data pipeline between private distributed ledger networks and systems, and link business workflows and smart contracts in different networks without depending on a trusted third party to act as conduit.

* Asset exchange (swap): This refers to the case where two users are present in two networks, and they perform concurrent and atomic swaps of two assets in the two corresponding networks, without transferring the assets outside the networks. The gateways aid in coordinating the messages pertaining to the swap.

The remainder of this architecture document will focus on the asset transfer flows.

5. Architecture

5.1. Goal of Architecture

The goal of the interoperability architecture is to permit two (2) gateways belonging to distinct networks to conduct a transfer of digital assets transfer between them, in a secure, atomic and verifiable manner.

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

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

The architecture therefore assumes that in addition to implementing the bilateral secure asset transfer protocol, a gateway has the role of making opaque (i.e. hiding) the constructs that are local and specific to its network.

Overall this approach ensures a high degree of interoperability across these networks, where each network can operate as a true autonomous system. Additionally, this approach permits each network to evolve its interior technology implementations without affecting other (external) networks.

The current architecture focuses on unidirectional asset transfers, although the building blocks in this architecture can be used to support protocols for bidirectional transfers.

For simplicity the current architecture employs two (2) gateways per transfer as the basic building block, with one gateway in the origin and destination networks respectively. However, the architecture seeks to be extensible to address future cases involving multiple gateways at both sides.

5.2. Overview of Asset Transfer

An asset transfer between two networks is performed using a secure asset transfer protocol implemented by the gateways in the respective networks. The two gateways implement the protocol in a direct interaction (unmediated).

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

The secure asset transfer protocol provides a coordination between the two gateways through the various message flows in the protocol that is communicated over a secure channel.

The protocol implements a commitment mechanism between the two gateways to ensure that the relevant properties atomicity, consistency, isolation, and durability is achieved in the transfer.

The mechanism to extinguish or introduce an asset from/into a network by its gateway is dependent on the specific network and outside the scope of the current work.

As part of the commitment mechanism, the sender gateway in the origin network must deliver proof to the received gateway in the destination network that asset in question has been extinguished (deleted) in the origin network.

Similarly, the receiver gateway in the destination network must deliver proof to the sender gateway in the origin network that the asset has been generated in the destination network.

These two tasks must be performed in a synchronized fashion between the two gateways, and the commitment mechanism must provide evidence of the asset transfer that is verifiable by a third party.

5.3. Desirable Properties of Asset Transfer

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

* Atomicity: A transfer must either commit or entirely fail (failure means no change to asset state).

* Consistency: A transfer (commit or fail) always leaves the networks in a consistent state (i.e. the asset is located in one network only at any time).

* Isolation: While the transfer is occurring, the asset state cannot be modified in the origin network.

* Durability: Once a transfer has been committed by both gateways, it must remain so regardless of subsequent gateway crashes.

* Verifiable by authorized third parties: The proof that the asset has been extinguished in the origin network, and the proof that the asset has been generated in the destination network must be verifiable by an authorized third party.

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 a 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 network.

The mechanism used to provide gateway crash-recovery is dependent on the specific network. For interoperability purposes the information contained in the log and the format of the log-data should be standardized.

The resumption of an interrupted transfer session (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) may need to be established between the two gateways, before a resumption of the transfer can begin.

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.

5.5. Overview of the Stages in Asset Transfer

The interaction between two gateways in the secure asset transfer protocol is summarized in Figure 1, where the origin network is NW1 and the destination network is NW2. T he gateways are denoted as G1 and G2 respectively.

```
            Originator                                   Beneficiary
                |                                             |
         +-------------+                               +-------------+
         |   Client    |                               |   Client    |
         | Application |                               | Application |
         |    (App1)   |                               |    (App2)   |
         +-------------+                               +-------------+
                |                                             |
                |                  (Stages)                   |
                V                                             V
         +-------------+       |<-----(1)----->|       +-------------+
         |    Network  |  +----+               +----+  |   Network   |
         |     NW1     |  |Gate|               |Gate|  |   NW2       |
         |             |--|way |<-----(2)----->|way |--|             |
         | +---------+ |  | G1 |               | G2 |  | +---------+ |
         | |  State  | |  +----+               +----+  | |   State | |
         | | Data DB1| |  +----+               +----+  | | Data DB2| |
         | +---------+ |       |<-----(3)----->|       | +---------+ |
         +-------------+                               +-------------+
```

Figure 1

The stages are summarized as follows.

* Stage 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 stage, 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 stages [FATF]. This stage is outside the scope of the current architecture.

* Stage 1: Pre-transfer Verification of Asset and Identities. In this stage the gateways G1 and G2 must perform mutual identification and authentication. Gateway G1 must communicate to G2 the type/information of the asset to be transferred, while G2 must validate that it has the ability to support this type of asset in its network.

* Stage 2: Evidence of asset locking or escrow. In this stage, gateway G1 must provide gateway G2 with sufficient evidence that the asset on its network NW1 is in a locked state (or escrowed) under the control of G1).

* Stage 3: Transfer commitment. In this stage gateways G1 and G2 commit to the unidirectional asset transfer using a 3PC (3-phase commit) subprotocol.

These transfer stages will be further discussed below.

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

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

```
      Orig  DB1          G1                   G2          DB2  Benef
      |     |            |                    |           |     |
      |---request------->|                    |           |     |
      |     |            |                    |           |     |
    ..|.....|............|....................|...........|.....|..
      |     |            |       Stage 1      |           |     |
      |     |            |                    |           |     |
      |     |       (1.1)|<-----Owner id----->|           |     |
      |     |            |                    |           |     |
      |     |            |                    |           |     |
      |     |       (1.2)|<--Asset Profile--->|           |     |
      |     |            |                    |           |     |
      |     |            |                    |           |     |
      |     |       (1.3)|<--Orig/Benef id--->|           |     |
      |     |            |                    |           |     |
    ..|.....|............|....................|...........|.....|..
      |     |            |                    |           |     |
```

Figure 2

This stage starts with the assumption that in network NW1 the gateway to process the asset transfer has been selected (namely gateway G1). It also assumes that the destination network NW2 has been identified where the beneficiary is located, and that gateway G2 in NW2 has been identified.

There are several steps that may occur in Stage 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 stage.

* 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. entities owning G1 and G2 respectively). Examples of ownership verification mechanism include X.509 certificates, directories of gateways and owners, and others.

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

* Identification and validation of type/asset profile: Both gateways must agree on the type of asset being transferred based on the published 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 its network to accept the type of asset. The policies governing network NW2 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 a 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 [SAT].

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

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

The steps of Stage 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 Stage 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 establish a lock or escrow the asset belonging to the originator. This prevents other local transactions in NW1 from changing the state of the asset until such time the lock by G1 is finalized or released. A time-lock or escrow may also be employed.

* G2 logs incoming asset (2.4): Gateway G2 logs a notification to its local state data DB2 that the arrival of the asset is imminent. This may also act as a notification for the beneficiary regarding incoming the asset transfer.

* Lock Evidence (2.5): Gateway G1 sends a digitally signed evidence regarding the locked (escrowed) state on the asset in network NW1. The signature by G1 is performed using its entity public-key pair. This signifies that G1 (i.e. its owner) is legally standing behind its assertion regarding the locked/escrowed state on the asset.

* 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 DB1           G1                   G2           DB2  Benef
      |     |            |      (Stage 1)     |            |     |
      |     |            |                    |            |     |
    ..|.....|............|....................|............|.....|..
      |     |            |       Stage 2      |            |     |
      |     |            |                    |            |     |
      |     |       (2.1)|-----Commence------>|            |     |
      |     |            |                    |            |     |
      |     |            |<-------ACK---------|(2.2)       |     |
      |     |            |                    |            |     |
      |     |            |                    |            |     |
      |     |<---Lock----|(2.3)               |            |     |
      |     |            |               (2.4)|----Log---->|     |
      |     |            |                    |            |     |
      |     |            |                    |            |     |
      |     |       (2.5)|---Lock Evidence--->|            |     |
      |     |            |                    |            |     |
      |     |            |<-----Receipt-------|(2.6)       |     |
      |     |            |                    |            |     |
    ..|.....|............|....................|............|.....|..
      |     |            |                    |            |     |
```

Figure 3

The precise form of the evidence in step 2.5 is dependent on the type of network NW1, and must be previously agreed upon between G1 and G2 in Stage 1.

The purpose of this evidence is for dispute resolution between G1 and G2 (i.e. the entities who own and operate G1 and G2 respectively) in the case that asset state inconsistencies in NW1 and NW2 are discovered later.

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 (Stage 3)

In Stage 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 stage, 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 updates its local state data to indicate lock-finalization or escrow finalization on the asset in network NW1. This signals the permanent extinguishment of the asset from network NW1. This state data must include a hash reference to the lock transaction previously in step (2.3). This indicates that the asset is no longer associated with its previous owner (originator) and that the asset instance is no longer recognized within network NW1.

* Commit-final (3.4): Gateway G1 indicates to G2 that G1 has performed a local lock/escrow finalization in NW1. This message must be digitally signed by G1.

* Asset-create (3.5): Gateway G2 issues a local state update in network NW2 to create (re-generate) the asset, associated with the beneficiary. This transaction must include a hash of the previous message (3.4) and hash reference to the log-incoming data previously in step (2.4). These hash references connect the newly re-generated asset with the overall transfer event originating from gateway G1.

* Ack-final (3.6): Gateway G2 indicates to gateway G1 that gateway G2 has performed an asset-regeneration in network NW2. This message must be digitally signed by G2.

* Location-record (3.7): Gateway G1 has the option to record the locally the information provided by G2 in the previous step. This record should include a hash reference to the confirmed lock-finalization data 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 Stage 1.

```
      Orig DB1          G1                   G2           DB2  Benef
      |     |            |      (Stage 2)     |            |     |
      |     |            |                    |            |     |
    ..|.....|............|....................|............|.....|..
      |     |            |       Stage 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 4

9. Commitment sub-protocol

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

In the case that there are multiple commitment subprotocols supported by the gateways, the choice of the sub-protocol (type/version) and the corresponding commitment evidence must be negotiated between the gateways during Stage 1.

For example, in Stage 2 and Stage 3 discussed above the gateways G1 and G2 may implement the classic 2-Phase or 3-Phase Commit (2PC or 3PC) sub-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 digital asset transfers across networks, the fact that an asset has been locked in NW1 must be communicated to G2 (as the 3PC participant) in an indisputable manner.

Similarly, G2 must return evidence to G1 that the asset has been re-generated in NW2.

These evidences must be verifiable by an authorized third party, in the case that disputes occur (post event) or where legal audit is required on the asset transfer.

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 an asset network holds an increasing number of digital 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 digital assets to external 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

Digital asset transfers must be policy-driven in the sense that it must observe and enforce the policies defined for the network. Resources that make-up a 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 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.

12. References

12.1. Normative References

[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>.

[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>.

[SAT] Hargreaves, M., Hardjono, T., and R. Belchior, "Secure Asset Transfer Protocol, IETF, draft-hargreaves-sat-core-00.", 5 May 2022, <https://datatracker.ietf.org/doc/draft-hargreaves-sat-core/>.

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>.

[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/>.

[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.

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

[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