Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 25 August 2026

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

21 February 2026

Secure Asset Transfer (SAT) Interoperability Architecture
draft-ietf-satp-architecture-09

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 asset 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 asset 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.

The peer gateways must implement a secure asset transfer protocol that must satisfy certain security, privacy, atomicity and liveliness 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. How to read this document

The current interoperability architecture document provides background concepts pertaining to the unidirectional transfer of a digital asset from one asset-network to another. This includes concepts around commitment agreement mechanisms or subprotocols, such as the 2-Phase Commit (2PC), which is at the heart of a unidirectional transfer.

Readers who are new to asset transfer scenarios in the various industries (e.g. banking, real-world assets, supply chains, etc.) are directed to the SAT use-cases document [CASE].

Readers who seek detailed technical information regarding the SAT asset transfer protocol should read the current document before proceeding to the SAT-Core document [CORE].

3. Terminology

The current architcture specification borrows existing terminology from NIST [NISTBC] and ISO [ISOBC]. New terms have been introduced notably in relation to the use of the gateways paradigm and commitment subprotocols.

* Asset-network (system): The network or system where an asset is digitally represented (e.g., as a token). The remainder of the document uses the term "network" to mean asset-systems or asset-networks (including their ledgers where applicable).

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

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

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

* Two-Party Commitment Agreement: The current interoperability architecture utilizes the classic two-party commitment agreement strategy borrowed from distributed database transactions management [Gray81]. The basic idea is for the two parties (two gateways) to follow a sequence of predefined steps, where at each step the parties communicate acknowledgement to proceed to the next step. This progressive strategy leads to the final step, also known as FINAL COMMIT, where both parties agree to a permanent change of state in their backend systems (e.g., databases, blockchains, etc.). This progressive strategy is often referred to in technical terms as the 2-Phase commitment (2PC) agreement subprotocol.

* 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 with external entities (e.g. communications) located outside the network. The term is used to distinguish this public-key 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 the DevID (IEEE 802.1AR) [DevID] or the EK/AIK keys (in TPM1.2 and TPM2.0) [TPMdevID].

* Gateway owner: The entity that owns and operates a gateway within a network.

* Application Context-ID: The relevant identifier used by originator’s application and the beneficiary’s application to identify the context of the asset transfer at the gateway level. The context identifier may also be used to bind the application to selected gateway for the given transfer instance, identified by a Session-ID.

* Gateway Session-ID: This is the identifier used between the sender gateway G1 and the recipient gateway G2 to identify the specific transfer instance between them. The Session-ID value MUST be included in all messages between the peer gateways in a transfer instance.

4. 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.

4.1. 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.

4.2. Operational Assumptions

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

* Application level context establishment: The transfer request from an Originator utilizing an application (App1) in the origin network is assumed to have occurred, and that some context-identifier has subsequently been derived by the respective applications (App1 and App2). Furthermore, this context-identifier is assumed to have been delivered by each application to its corresponding gateway, permiting each gateway to internally bind the transfer session-identifier to that context-identifier.

* Identification of asset to be transferred: The applications at the originator and the beneficiary are assumed to have identified the digital asset to be transferred.

* 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.

4.3. 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 are performing digital signatures on messages. As such, gateway operators are assumed to take on the relevant liabilities for signing the messages.

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

* Transitory control of asset by gateway: An asset being transferred via SAT will technically be controlled by gateway throughout the transfer duration to ensure the state of the asset is not modified by another entity. Gateway owners are liable for the asset throughout this duration.

* 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 [HS19].

5. 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 (burned) in the origin network and be regenerated (minted) at the destination network [HLP19].

* Data Sharing: This refers to the transfer of data only under authorization, in such a way that the data can be verified by a third party [Abebe19]. The data sharing mode addresses the use-cases where the state update in one network or system depends on the existence of state information recorded in a different network or system [DLVIEW].

* 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 (i.e. across) the networks. The gateways aid in coordinating the messages pertaining to the swap.

The current SATP architecture can be extended to address the use-cases pertaining to asset exchanges (swap) between two entities, both of which are assumed to be present in the origin and destination asset networks. Similarly, the SATP architecture can be utilized for the data transfer mode to report the state of an asset within a private network, where the gateway acts as an intermediary. However, the asset exchange and data transfer mode will be addressed in future specifications.

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

6. Architecture

6.1. 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 gateways 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.

The abstract construct of the gateway is used to represent endpoints that implement the asset transfer protocol interactions and the business logic that coordinates the transfer protocol steps until completion satisfying the ACID properties and ensuring liveliness of the protocol interactions. In classical distributed databases, this business logic is often referred to as the transaction manager or the transaction coordinator. This architecture specification does not prescribe any implementations of gateways.

6.2. 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 [CORE]. The two gateways implement the protocol in a direct interaction (unmediated).

A successful transfer results in the asset being extinguished (burned) at the origin network, and for the asset to be regenerated (minted) at 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 two-party commitment agremeent mechanism (subprotocol) between the two gateways to ensure that the relevant properties atomicity, consistency, isolation, and durability (ACID) are achieved in the transfer.

The mechanism to extinguish (burn) or regenerate (mint) an asset from/into a network by its gateway is dependent on the specific network and is outside the scope of the current architecture specification. Similarly, the mechanisms used to provide cryptographic proofs that an asset has been burned or minted in a given network is also network-specific and therefore out of scope.

As part of the two-party commitment agremeent mechanism, the sender gateway in the origin network delivers a signed assertion to the receiver gateway at the destination network which states that asset in question has been extinguished (burned) from the origin network.

Similarly, the receiver gateway at the destination network in return delivers a signed assertion to the sender gateway at the origin network which states that the asset has been regenerated (minted) in the destination network.

These two tasks are be performed in a synchronized fashion between the two gateways, following the steps of the two-party commitment agremeent mechanism. Adherence to the steps provides sufficient evidence of the asset transfer that is verifiable by an authorized third party.

The messages exchanged between the gateways within the two-party commitment agreement subprotocol are digitally signed by the sender and transmitted over a secure TLS session. Each message is cryptographically bound to the previous message (i.e., includes hash of previous message) to ensure the integrity of the step-by-step flows in the commitment agreement subprotocol. Each message carries a message-type, Session-ID and timestamp, thereby allowing both parties to perform the correct sequencing of these messages.

The use of digital signatures on the messages exchanged by the gateways in commitment agreement subprotocol strengthens the security properties of the two-party commitment agreement implementation, and assists in dispute resolutions in the case where one party (or both) acts in a dishonest way.

6.3. Desirable Properties of Asset Transfer

The desirable features of asset transfers between two gateways include, but are 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) MUST 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 MUST NOT 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.

* Liveliness and safety: The asset transfer protocol results in both gateways reaching a non-blocking state (commits or aborts) satisfying the above ACID properties.

* 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 MUST satisfy these properties, independent of whether the implementation employs stateful messaging or stateless messaging between the two gateways.

Performing an asset transfer safely and securely is not simply a matter of communicating desire or intent between two systems represented by gateways, though such communication is a necessary part of asset transfer. The systems, or at least their gateway proxies, must be interoperable in order to transfer assets among themselves, but such interoperability imposes strictly more demands on systems managing digital assets, especially systems that are built on distributed ledgers, compared to conventional communication interoperability.

Communication interoperability, which is concerned with syntax and semantics of information geared towards producing a common understanding (or knowledge reconciliation) among systems, is insufficient to fulfill an asset transfer that requires systems to carry out state updates in concert with each other. But communication, or messaging standards, play a necessary and complementary role to asset transfer protocols. An exemplar of this is ISO 20022, which is a comprehensive global standard for financial messaging that specifies message syntax for common actions occurring in financial business processes, including payments, credit card transactions, securities settlements, funds, and trade [ISO20022]. This standard provides the tools to model business processes from basic logical building blocks and schemas to construct messages using common formats like XML, JSON, and ASN.1.

As discussed later, such messaging standards are useful to communicate information about the states of processes and digital assets across systems, to make requests, and to convey intent. They therefore play a necessary and complementary role in asset transfer protocols. However they are by themselves insufficient to ensure the ACID and verifiability properties described earlier. Another way to think about the relationship between messaging standards like ISO 20022 and asset transfer protocols is that the former is concerned with the "what" of cross-system interoperability whereas the latter is concerned with the "how". Both kinds of protocols treat systems as black boxes, but asset transfer protocols must place some responsibility, and depend, on systems to drive a protocol instance to successful conclusion.

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

Implementations of a gateway MUST maintain event logs and checkpoints for the purpose of gateway crash recovery. The log-data generated by a gateway represents an interior resource that coule be made accessible to other authorized gateways within the same asset 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 [CRASH].

The log-data generated by SATP gateways in the context of asset transfers enables multiple implementations of gateways to serve a given asset network. For example, an asset transfer session interrupted by a crashed gateway could be resumed by another gateway if the log-data was accessible to the second gateway and was in a standardized format.

The resumption of an interrupted transfer session (e.g. due to gateway crash, network failure, etc.) need to 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.

The log-data semantics and syntax for SATP gateway crash management are for future work, and could be developed based on existing log-data standards such as Syslog [RFC5424] and [RFC3164].

6.5. 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. The gateways are denoted as G1 and G2 respectively.

```
            Originator                                   Beneficiary
                |                                             |
         +-------------+                               +-------------+
         |   Client    |                               |   Client    |
         | Application |                               | Application |
         |    (App1)   |                               |    (App2)   |
         +-------------+                               +-------------+
                |                                             |
                |                                             |
                |-----------+                     +-----------|
                |           |  |<--(Stage 0)-->|  |           |
                V           |  |               |  |           V
         +-------------+    V  |               |  V    +-------------+
         |    Network  |  +----+<-- Stage 1 -->+----+  |   Network   |
         |     NW1     |  |Gate|               |Gate|  |     NW2     |
         |             |--|way |               |way |--|             |
         | +---------+ |  | G1 |<-- Stage 2 -->| G2 |  | +---------+ |
         | |  State  | |  +----+               +----+  | |   State | |
         | | Data DB1| |       |               |       | | Data DB2| |
         | +---------+ |       |<-- Stage 3--->|       | +---------+ |
         +-------------+                               +-------------+
```

Figure 1

The stages are summarized as follows.

* Stage 0: Pre-transfer Verification and Context Establishment. The two applications utilized by the originator and beneficiary are assumed to interact as part of the asset transfer. In this stage, the applications App1 and App2 may establish some shared transfer context information (e.g. Context-ID) at the application level that will be made available to their respective gateways G1 and G2. The verification of the identities of the Originator and Beneficiary may occur in this stage [FATF].

* Stage 1: Transfer Initiation and Commencement. In this stage gateways G1 and G2 exchange information (claims) regarding the asset to be transferred, the identity information of the Originator and Beneficiary and other information regarding relevant actors (e.g. gateway owners/operators). The main task in this stage is for both gateways to finalize the parameters previously negotiated in Stage 0 and to agree to commence the transfer.

* Stage 2: Lock Assertion and Receipt. In this stage, gateway G1 provides gateway G2 with a signed assertion that the asset in NW1 has been immobilized and under the control on G1. A signed assertion is needed because NW1 may be a private or closed network, and therefore the state-database (ledger) in NW1 is not readable by external entities including by G2. This means that gateway G1 needs to make an explicit signed assertion about the state in NW1. Note that the owner/operator of G1 takes on liability in signing this assertion.

* Stage 3: Commitment Preparation and Finalization. In this stage gateways G1 and G2 commit to the unidirectional asset transfer using a 2PC (2-phase commitment agreement) subprotocol.

These stages will be further discussed below.

7. Pre-transfer Verification and Context Establishment (Stage-0)

Stage 0 refers to the various verification tasks related to the asset and the actors involved in a transfer instance, and which need be carried-out prior to commencing the transfer in Stage 1.

Several tasks need to be conducted as part of the pre-transfer stage:

* Application level ContextID establishment: The application (App1) used by the originator and the application (App2) used by the beneficiary MUST establish a transfer context identifier (contextID) to uniquely identity the transfer at the application level.

* Identification of the asset in the origin network: The specific asset in the origin network NW1 need to be located and identified, and its ownership MUST be verified by the sender gateway G1. A gateway MUST NOT transfer assets whose ownership is unverified. Examples of identification syntax for digital assets can be found in [ISO20022] or ITIN [ITIN].

* Verification of the class or type of asset: The receiving gateway G2 MUST verify the class or type of asset that is to be transferred by gateway G1 in network NW1. This is to ensure that the asset type/class conforms to the governing policies in the destination network NW2. Additionally, gateway G2 must ensure that network NW2 can technologically receive (mint) the asset of that given type/class. Asset schema definitons, asset profiles and token metadata may assist in this vertification process.

* Validation of asset ownership status: The gateway G1 in the origin network NW1 MUST validate the ownership of the asset to be transferred prior to beginning the transfer. This is ensure that the asset to be transferred to an external network NW2 is owned by the originator who is requesting the transfer.

* Authorization to transfer: The gateway G1 MUST obtain authorization from the owner of the asset (originator) to perform the transfer to the beneficiary in network NW2. Similarly, the gateway G2 serving network NW2 MUST obtain authorization from the beneficiary to receive the transfer and assign the asset to the beneficiary in NW2.

* Exchange of Travel Rule information: The Travel Rule [FATF] pertains to the information regarding the owner of the asset (originator) in NW1 and the intended recipient (beneficiary) in NW2 of the transfer. In some jurisdictons, the information about the originator and the beneficiary must be exchanged (transmitted respectively) prior to the transfer of the asset.

* Validation of the gateway ownership: The gateways G1 and G2 need a mechanism to enable them to validate the identity of the owner (operator) of the gateways respectively, and the fact of the legal ownership of the gateway. Examples of ownership verification mechanism include X.509 certificates, directories of gateways and owners, and others.

* Mutual device attestations: In cases where device attestation [RATS] is required, each gateway need to 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 [HS19].

* Negotiation of transfer protocol and network parameters: Gateway G1 and G2 need to agree on the parameters to be employed within the transfer instance. Examples include endpoints definitions for resources, duration of time-outs of messages, type of commitment agreement subprotocol (e.g. 2PC), signature algorithms, average lock-time durations in their respective networks, and others.

The current specification seeks to reuse as much as possible the existing standards related to digital assets. We seek to rely on existing messaging standards like ISO 20022 [ISO20022] or ITIN [ITIN] for gateway ownership validation, owner status validation, asset profile identification, and communication of travel rule and transfer context information. For identification of digital assets maintained by distributed ledgers or blockchain systems, we can also rely on standards like ITIN [ITIN].

8. Transfer Initiation and Commencement (Stage-1)

In Stage 1 the sender gateway (G1) and the receiver gateway (G2) explicitly accept the parameters of the transfer which were negotiated in the pre-transfer stage (Stage 0).

This explicit acceptance of the parameters takes the form of gateway G1 sending a signed Transfer Proposal message containing a Transfer Initiation Claims set (namely the parameters agreed upon in Stage 0), and for the gateway G2 to respond with a signed Proposal Receipt message which carries a hash of the proposal message.

```
       App1  DB1          G1                      G2          DB2    App2
        |     |            |                      |            |     |
        |     |            |      (Stage 0)       |            |     |
        |     |            |                      |            |     |
        |<------------ (transfer context establishment) ------------>|
        |     |            |                      |            |     |
        |---request------->|                      |<------request----|
        |     |            |                      |            |     |
        |     |            |<-Param Negotiation-->|            |     |
        |     |            |                      |            |     |
      ..|.....|............|......................|............|.....|..
        |     |            |       Stage 1        |            |     |
        |     |            |                      |            |     |
        |     |       (1.1)|---Transf. Proposal-->|            |     |
        |     |            |                      |            |     |
        |     |            |                      |            |     |
        |     |       (1.2)|<--Proposal Receipt---|            |     |
        |     |            |                      |            |     |
        |     |            |                      |            |     |
        |     |       (1.3)|---Transf. Commence-->|            |     |
        |     |            |                      |            |     |
        |     |            |                      |            |     |
        |     |       (1.4)|<--- ACK Commence ----|            |     |
        |     |            |                      |            |     |
      ..|.....|............|......................|............|.....|..
        |     |            |                      |            |     |
```

Figure 2

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.

* Transfer Proposal message (1.1): Gateway G1 sends a signed transfer proposal message that contains the Transfer Initiation Claims to gateway G2. The claims carry the parameters negotiated in Stage 0 (pre-transfer negotiations).

* Proposal Receipt (1.2): The gateway G2 indicates acceptance of the parameters in the Transfer Initiation Claims by way of sending a signed Proposal Receipt message to G1. If gateway G2 decides not to accept parameters in the Transfer Initiation Claims, then G2 can send an abort message to G1, or simply ignore the message (time-out).

* Transfer Commence message (1.3): Once gateway G1 receives the signed Proposal Receipt from gateway G2, gateway G1 is ready to signal the commencement of the asset transfer. This is done by gateway G1 sending a signed Transfer Commence message to G2.

* Commence Acknowledgement message (1.4): Gateway G2 accepts the formal commencement of the transfer by responding with a signed Commence ACK message.

It is important to note the logical separation between the transfer proposal/receipt messages from the commencement messages. This separation allows the gateways to decline to proceed during the proposal finalization (1.1 and 1.2), prior to starting the commitment agreement subprotocol (2PC) which formally begins at the Commence messages (1.3 and 1.4).

This logical separation is useful because in some implementations the decision to start the commencement (1.3 and 1.4) implies that the gateways and network have sufficient resource to complete the transfer. Gateways that experience extreme loads may use this separation to slightly delay the commencement until their loads subsides.

Note that some implementations may choose to enable a multi-round interactions for steps 1.1 and 1.2.

9. Asset Lock Assertion and Receipt (Stage 2)

In this stage, gateway G1 issues a signed assertion that the asset in origin network NW1 has been immobilized and under the control of G1.

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

* G1 Lock/Escrow Asset (2.1): 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.

* Lock Assertion (2.2): Gateway G1 sends a digitally signed assertion regarding the locked (escrowed or immobilized) state on the asset in network NW1. The signature by G1 is performed using its entity public-key pair. This signature signifies that G1 (i.e. its owner/operator) is standing behind its statement regarding the locked/escrowed state on the asset.

* G2 Logs Lock-Assertion Information (2.3): Gateway G2 logs/records a copy of the signed lock-assertion message received in Step 2.4 to its local state data DB2. G2 may also notify the fact of the lock-assertion to all members of network NW2.

* Lock-Assertion Receipt (2.4): If gateway G2 accepts the signed assertion from G1, then G2 responds with a digitally signed receipt message which includes a hash of the previous lock-assertion message. The signature by G2 is performed using its entity public-key pair. Otherwise, if G2 declines accepting the assertion then G2 can simply ignore the transfer and let the session time-out (i.e. transfer attempt has failed).

```
   Orig DB1            G1                    G2             DB2  Benef
    |     |            |      (Stage 1)      |              |     |
    |     |            |                     |              |     |
  ..|.....|............|.....................|..............|.....|..
    |     |            |       Stage 2       |              |     |
    |     |            |                     |              |     |
    |     |<---Lock----|(2.1)                |              |     |
    |     |            |                     |              |     |
    |     |       (2.2)|--Lock-Assertion---->|              |     |
    |     |            |                     |              |     |
    |     |            |                (2.3)|---Record---->|     |
    |     |            |                     |              |     |
    |     |            |                     |              |     |
    |     |            |<-----Receipt--------|(2.4)         |     |
    |     |            |                     |              |     |
  ..|.....|............|.....................|..............|.....|..
    |     |            |                     |              |     |
```

Figure 3

The purpose of the signed lock-assertion 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 returns a digitally signed receipt to G1 regarding the earlier signed lock-assertion in order to cover G1 (exculpatory proof) in the case of later denial by G2.

10. Commitment Preparation and Finalization (Stage 3)

In Stage 3 the gateways G1 and G2 commit to the asset transfer by making permanent the changes they made to the respective asset networks. The previous signed receipt message (2.4) from gateway G2 to gateway G1 signals the start of the commitment subprotocol in Stage 3.

Upon receiving the signed receipt message from G2 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 hashes of the previous messages (message 2.2 and 2.4).

* Temporary asset mint (3.2): Gateway G2 creates (mints) an equivalent asset in NW2 assigned to itself as the owner. This step can be reversed (i.e. asset destroyed) in the case of the failure in the commitment steps because G2 is still the owner of the asset in NW2.

* Commit-ready (3.3): Gateway G2 sends a commit-ready message to G1 indicating that it is ready to carry-out the last steps of the commitment agreement subprotocol. Note that that the entire asset transfer session can be aborted before this step without affecting the asset state in the respective networks.

* Asset burn (3.4): Gateway G1 extinguishes (burns) the asset in network NW1 which it has locked since Step 2.3.

* Commit-final assertion (3.5): Gateway G1 indicates to G2 that G1 has performed the extinguishment of the asset in NW1.

* Asset-assignment (3.6): Gateway G2 assigns the minted asset (which it has been self-holding since Step 3.2) to the Beneficiary.

* ACK-final receipt (3.7): Gateway G2 sends a signed assertion that it has assigned the asset to the intended Beneficiary.

* Record receipt (3.8): Gateway G1 logs/records a copy of the signed receipt message to its local state data DB1. G1 may also notify the fact of the signed receipt to all members of network NW1.

* Transfer complete (3.9): Gateway G1 terminates the asset transfer session with gateway G2. This allows both sides to close down the secure TLS channel established earlier in Stage 1.

```
         Orig DB1           G1                   G2          DB2  Benef
         |     |             |      (Stage 2)     |            |     |
         |     |             |                    |            |     |
       ..|.....|.............|....................|............|.....|..
         |     |             |       Stage 3      |            |     |
         |     |             |                    |            |     |
         |     |        (3.1)|--Commit Prepare--->|            |     |
         |     |             |                    |            |     |
         |     |             |               (3.2)|----Mint--->|     |
         |     |             |                    |            |     |
         |     |             |<--Commit Ready ----|(3.3)       |     |
         |     |             |                    |            |     |
         |     |             |                    |            |     |
         |     |<---Burn-----|(3.4)               |            |     |
         |     |             |                    |            |     |
         |     |        (3.5)|-Commit Final Asrt->|            |     |
         |     |             |                    |            |     |
         |     |             |                    |            |     |
         |     |             |               (3.6)|---Assign-->|     |
         |     |             |                    |            |     |
         |     |             |<-----ACK Final-----|(3.7)       |     |
         |     |             |                    |            |     |
         |     |             |                    |            |     |
         |     |<---Record---|(3.8)               |            |     |
         |     |             |                    |            |     |
         |     |        (3.9)|-----Completed----->|            |     |
         |     |             |                    |            |     |
       ..|.....|.............|....................|............|.....|..
         |     |             |                    |            |     |
```

Figure 4

11. The Commitment Agreement Sub-protocol

Within Stage 3, the gateways implement a transactional commitment agreement sub-protocols that permit the coordination between two gateways, and the final commitment of the unidirectional transfer of the asset.

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 0.

For example, in Stage 2 and Stage 3 discussed above the gateways G1 and G2 may implement the classic commitment aggrement subprotocol [Gray81] as a means to ensure efficient and non-disputable commitments to the asset transfer.

Historically, transactional commitment agreement protocols in distributed databases 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 needs to be communicated via an assertion to G2 in an indisputable manner.

Similarly, G2 here returns a signed assertion to G1 that the asset has been regenerated (minted) in NW2.

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

The assertions (claims) signed by the gateways MUST use a standard format, such as the Secure Assertions Markup Language (SAML) [SAML2.0] or JSON Web-Tokens (JWT) [RFC7519]. The current interoperability architecture consider assertions formats as out of scope, and seeks to make use of these existing standards. The choice of the format for signed assertions must be agreed upon by peer gateways G1 and G2 within Stage-0.

12. 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].

The SAT protocol faces challenges with regards to the confidentiality of a transfer between gateways, and the potential issue related to a denial-of-service (and resource waste) when either gateway is not compliant with the protocol.

For confidentiality of a transfer, the secure asset transfer protocol MUST utilize a TLS1.3 secure channel established between the sender gateway (G1) and the receiver gateway (G2). The two gateways establish this secure channel at the start of Stage 1 before they can proceed to execute the asset transfer protocol. This includes both gateways verifying all the relevant parameters required for their TLS session (e.g. correct TLS endpoints, certificate validation, identity validation, etc.).

There are several challenges that may arise when gateways are not compliant with the SAT protocol. Some of these are described below.

12.1. Multiple intentional aborts by the sender gateway

A dishonest sender gateway G1 may purposely fail to continue the protocol run at certain crucial points. One such crucial point is in Stage-3, where the gateway G1 is expected to transmit the Commit-Final Assertion message (3.5). If the gateway G1 intentionally fails to transmit this message, gateway G2 may conclude that the message has been lost and may proceed to reverse the temporary hold it has previously created (temporary asset mint in message 3.2). Although this dishonest behavior by G1 does not cause asset damage to G2 or NW2, it may exhaust computing resources at gateway G2. If network NW2 incurs transaction fees, such a reversal may be costly for gateway G2.

12.2. Multiple intentional aborts by the receiver gateway

In a similar manner, a receiver gateway G2 may also purposely fail to continue the protocol run at certain crucial points. One such point is the Commit-Ready message (3.3) that it should transmit to G1 after receiving the commit prepare message (3.1) from G1. In this case, gateway G1 may conclude that the message is lost and simply abort the protocol run.

Another possible denial-of-service attack could arise when G2 purposefully fails to send a Lock-Assertion-Receipt (2.4), thereby forcing G1 to reverse its lock that was performed earlier (2.1).

12.3. Failure to transmit ACK-Final Receipt

Another possible point of attack by a dishonest gateway G2 may occur by the gateway intentionally failing to transmit the ACK-Final-Receipt (3.7) in response to the Commit-Final Assertion message (3.5) from gateway G1. Here, the sender gateway G1 may conclude that the message is lost and will assume that the transaction has reached completion in network NW2. The sender gateway G1 has retained the previous Lock-Assertion Receipt (2.4) in Stage-2 that was signed by G2, indicating that the gateway G2 has accepted the responsibility of ensuring that the asset-assignment (3.6) by G2 will be correctly executed. Failure by G2 to complete this task may become a liability for the owner of gateway G2.

In general, it is recommended that multiple redundant gateways be utilized within a network to mitigate a single gateway's malicious behavior. Furthermore, there are gateway recovery and failover mechanisms that have been defined in [CRASH].

12.4. Failure to extinguish asset

Another potential attack may come from a dishonest gateway G1 who intentionally fails to extinguish the asset in network NW1 (in step 3.4). This means G1 is henceforth in control of the asset belonging to the originator.

This This type of denial-of-service could be considered a network-specific limitation because it implies that G1 was able to perform a lock on the asset in network NW1 on behalf of an asset-owner without accountability. Several asset networks currently support solutions to this problem by way of introducing a temporary third-party trusted holder (custodian) in the network for the duration of a transaction.

This denial-of-service is out of scope for the current architecture specification because it represents a weakness on the part of the network NW1.

12.5. Identity impersonations

Another vector of attack may involve a gateway that impersonates an asset holder in a given network. For example, a gateway G1 may pretend to be the owner of an asset (originator) in network NW1 and proceed to transfer it to a beneficiary located in network NW2.

The verification of the identity of the originator and beneficiary MUST be performed as part of the set-up stage (Stage 0) as described above.

The identity verification includes that of the owner of gateways G1 and G2 respectively. Standard protocols for federated identity management already exist and have wide deployment.

13. Policy Considerations

Digital asset transfers are policy-driven in the sense that it must observe and enforce the policies defined each of the respective networks. Resources that make-up a network are owned and operated by entities (e.g. typically persons or organizations), and these entities typically operate within regulatory jurisdictions. 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 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.

14. Compatibility Considerations

The current architecture for the secure transfer of assets between two networks is designed to be agnostic to the identification and the format of the asset within the origin and destination networks. This enables the architecture to be compatible with a broad range of assets identification schemes (e.g. ITIN [ITIN]), asset description syntaxes and business processes such as ISO 20022. A key part of Stage 0 is to enable the gateways in the respective networks to facilitate the exchange information about the asset state, its identification model, and formats. This facilitation enables gateways and networks to determine whether they can process incoming transfers prior to commencing Stage 1.

15. IANA Considerations

This document has no IANA actions.

16. References

16.1. Normative References

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

[ISO20022] ISO, "Universal Financial Industry Message Scheme (ISO 20022).", July 2023, <https://www.iso20022.org>.

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

[ITIN] ITSA, "International Token Identification Number.", July 2023, <https://my.itsa.global/what-we-do>.

[NISTBC] 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>.

16.2. Informative References

[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>.

[CASE] Ramakrishna, V., Hardjono, T., and P. Liu, "IETF Secure Asset Transfer (SAT) Use-Cases. IETF draft-ietf-satp-usecases-06", 31 July 2025, <https://datatracker.ietf.org/doc/draft-ietf-satp-usecases/>.

[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.

[CORE] Hargreaves, M., Hardjono, T., Belchior, R., Ramakrishna, V., and A. Chiriac, "IETF Secure Asset Transfer Protocol (SATP) Core. IETF draft-ietf-satp-core-12.", 2 November 2025, <https://datatracker.ietf.org/doc/draft-ietf-satp-core/>.>.

[CRASH] Belchior, R., Vasconcelos, A., Correia, M., and T. Hardjono, "Hermes: Fault-tolerant middleware for blockchain interoperability. Future Generation Computer Systems, Vol. 129, pp. 236-251.", April 2022, <https://doi.org/10.1016/j.future.2021.11.004>.

[DevID] IEEE, "802.1AR: Secure Device Identity", August 2018, <https://doi.org/10.1109/IEEESTD.2018.8423794>.

[DLVIEW] Ramakrishna, V., Pandit, V., Nishad, S., Narayanam, K., and D. Vinayagamurthy, "Views and View Addresses for Blockchain/DLT Interoperability, IETF draft-ramakrishna-satp-views-addresses-06", 15 September 2025.

[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.

[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>.

[HS19] 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>.

[RATS] Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote ATtestation procedureS (RATS) Architecture", January 2023, <https://datatracker.ietf.org/doc/rfc9334/>.

[RFC3164] Lonvick, C., "The BSD syslog Protocol", August 2001, <https://datatracker.ietf.org/doc/rfc3164/>.

[RFC5424] Gerhards, H., "The Syslog Protocol", March 2009, <https://datatracker.ietf.org/doc/rfc5424/>.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", May 2015, <https://datatracker.ietf.org/doc/rfc7519/>.

[SAML2.0] Ragouzis, N., Hughes, J., Philpott, R., Maler, E., Madsen, P., and T. Scavo, "Security Assertion Markup Language (SAML) V2.0 Technical Overview", March 2008, <https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html>.

[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.

[TPMdevID] TCG, "TPM 2.0 Keys for Device Identity and Attestation", 8 October 2021, <https://trustedcomputinggroup.org/resource/tpm-2-0-keys-for-device-identity-and-attestation/>.

Appendix A. Acknowledgments

The authors would like to thank the following people for their input and support:

Andre Augusto, Denis Avrilionis, Rafael Belchior, Carsten Bormann, Sandip Chakraborty, Shiping Chen, Alexandru Chiriac, Claire Facer, Martin Gfeller, Bishakh Ghosh, Wes Hardaker, David Millman, Paul

Hoffman, Russ Housley, Nick Kerrigan, Peter Liu, Krishnasuri Narayanam, Chris Ostrowski, Vinayaka Pandit, Luke Riley, John Robotham, Orie Steele, Peter Somogyvari, Mike Truter, Gilbert Verdian, Dhinakaran Vinayagamurthy, Paul Wouters, Qin Wang, Weijia Zhang.

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