Network Working Group
Internet-Draft
Intended status: Informational
Expires: 26 February 2026

D. Avrilionis
Compellio S.A.
T. Hardjono
MIT

25 August 2025

SATP Setup Stage
draft-avrilionis-satp-setup-stage-03

Abstract

SATP Core defines an unidirectional transfer of assets in three stages, namely the Transfer Initiation stage (Stage-1), the Lock-Assertion stage (Stage-2) and the Commitment Establishment stage (Stage-3). This document defines the Setup Phase, often called "Stage-0", prior to the execution of SATP Core. During Setup, the two Gateways that would participate in the asset transfer are bound together via a "transfer context". The transfer context conveys information regarding the assets to be exchanged. Gateway can perform any kind of negotiation based on that transfer context, before entering into SATP Core.

Editorial Note

Discussion of this draft takes place on the SATP mailing list (sat@ietf.org), which has its home page at https://datatracker.ietf.org/wg/satp/about/.

Status of This Memo

Table of Contents

1. Introduction

SATP, in its specification in SATP Core[1], defines a three-stage protocol for unidirectional transfer of assets between two Networks. The protocol is executed by two Gateways, according to the flow defined in section 5 of SATP Core. As stated in SATP Core, "the interactions between the peer gateways prior to transfer initiation stage (Stage-1) is referred to as the setup stage (Stage-0)". Although the Setup stage is outside the scope of SATP Core, it plays an important role in the overall consistency of the protocol. For example, during setup, gateways can access important information based on the nature of the assets to be transferred, including technical, business, or compliance data. This document defines the sequence of interactions during Setup. These interactions among applications, gateways and networks lead to the binding of the two gateways that will then perform a specific transfer instance as defined in SATP Core.

Setup is triggered by the Client Application that originates the transfer sequence via a call to Gateway's G1 (via API1). The transfer context associated with the transfer instance conveys all required information about the assets to be transferred.

2. Stage 0 flows

The diagram below presents the sequence of actions during the Setup stage.

App1 the "Originator Application" (driven by Alice) and App2 the "Beneficiary Application" (driven by Bob) must interact with each other in Stage-0 before execution of SATP Core. The sequence of interactions is as follows:

1. App1 (the Originator App) requests to gateway G1 (Originator Gateway) the creation of a transferContext (TC) related to one of more Tokenized Asset Records (TARs - see [3]). We assume App1 can access the definition of TARs, for example stored in an external Registry (accessing TAR definitions stored in a Registry by an App is outside the scope of this document. We assume such operations occurred before the execution of the Setup stage).

2. App1 receives the transferContext ID (TCID) from the Originator Gateway G1.

3. App1 calls the relevant endpoint(s) in Network NW1 in order to bind the Digitized Asset Records (DARs - see [3]) with the specific transferContext ID. Associating the transferContext with specific DARs in a Network is important, especially in case of competing transfer instances: the Network must associate the DARs with the correct transferContext ID to maintain consistency of locking and thus avoid deadlocks / starvation, for example, when different SATP transfer instances compete for the transfer of the same DARs.

4. App1 sends (propagates) the transferContext ID to App2.

5. App2 acknowledges propagation of the specific transferContext ID (via a return message to App1).

6. App2 calls the relevant endpoin(s) in Network NW2 in order to prepare creation of DARs. This step aims at preparing NW2 for the transfer. For example, ensure collateralisation of assets, or perform compliance verifications before the effective transfer.

7. App2 calls the gateway G2 to propagate the transferContext ID.

8. Gateway G2 calls G1 (via API2) to perform binding of the two gateways for the specific Transfer Context ID.

9. At any moment after binding G2 with the Originator Gateway G1, App1 can start the execution of the SATP Core transfer instance.

At the end of the execution of the transfer instance, the Apps can call their respective gateways (via API1) to receive an update on the completion status of the transfer (messages 10 to 13). The completion status can be either "commit" or "rollback".

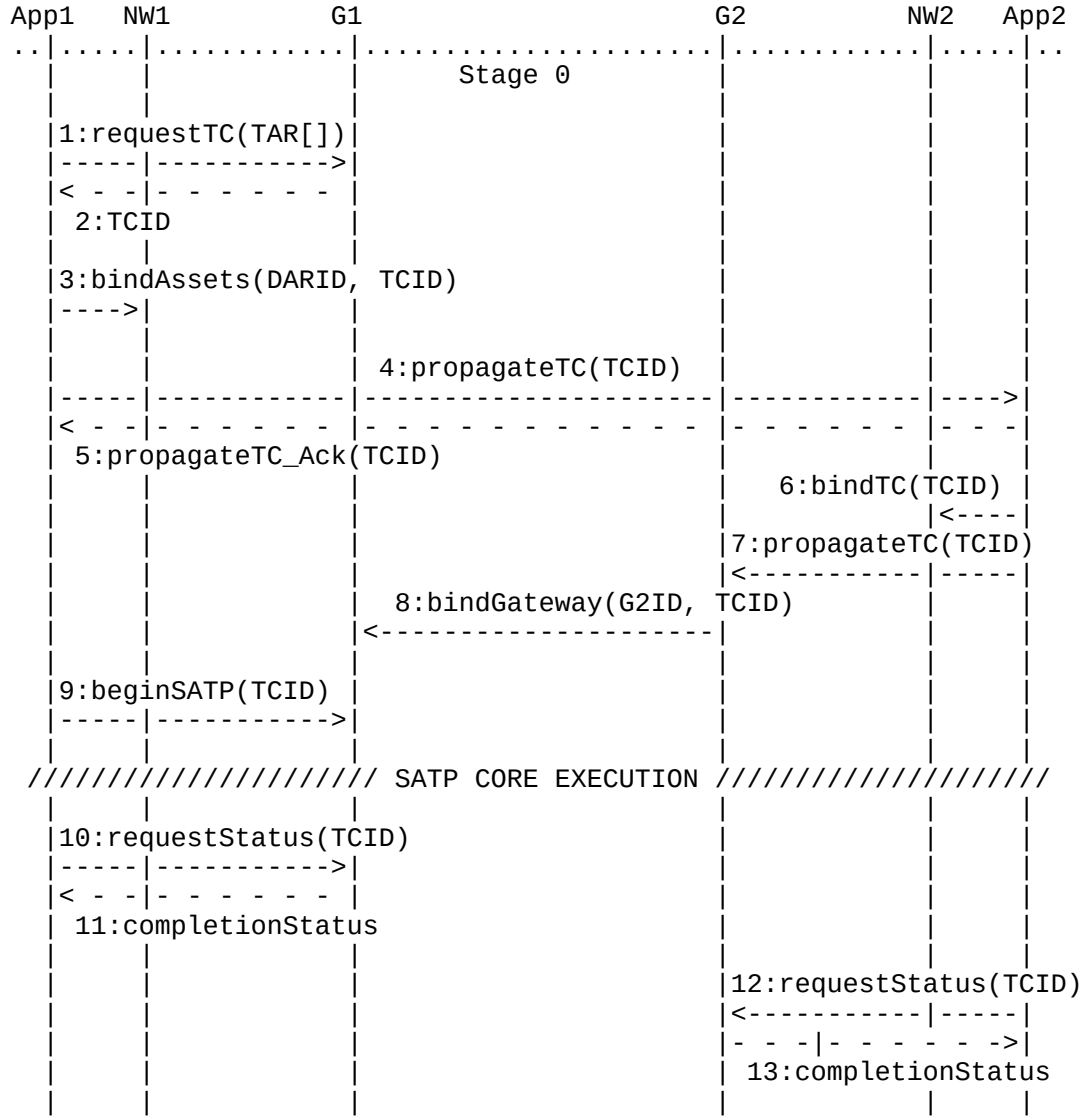

Figure 1: Stage 0 flow

3. Stage 0 pre-conditions

To summarise, at the end of the Setup stage, the following conditions shall be met. Only if all these pre-conditions are true the SATP Core transfer instance can be executed (note: completion can be either commit or rollback).

* The Originator Application (a.k.a. Client App A1, a.k.a. Alice) has obtained a transfer context, i.e. information that unambiguously defines a specific asset transfer instance related to sending a well-defined set of assets (referenced by TARs) from a given Originator application to a given Beneficiary application.

* NW1 is aware that this well-defined set of managed assets (owned by the originator) is going to be transferred for the given transfer context, and this at any moment in the future, based on G1 initiative.

* G1 and G2 are bound for the given transfer context TC

* The Beneficiary Application (a.k.a. Client App A2, a.k.a. Bob) knows that it will receive a well-defined set of assets for the given transfer context TC

* NW2 is aware that a well-defined set of assets will be received for a given beneficiary and for the given transfer context TC

4. Annex: Transfer Context Example

Below we give an example of a transfer context. The following can be noted:

* The identifier of the transfer context contains a reference to the originator gateway address.

* There might be several sessions associated with a transfer context. For example, in case of recovery from a crash, a new session might be created for the specific transfer context (the example shows a transfer context with two previous sessions in history).

* tarIDs refer to TAR that are stored in Registries and can be accessed by Gateways via API3.

* In order not to lock assets forever a specific transfer expiration time limit can be set.

* Network endpoints are important for traceability reasons so it is clear what execution unit in the networks managed the lock/ extinguishing (NW1) and creation (NW2) operations for the specific transfer context.


```
{
   "transferContextID": "urn:satp:transferContext:0x517BBF0c9B71f64b5807f644E1F1bacD3Afb3
ec2.0xf3beac30c498d9e26865f34fcaa57dbb935b0d74",
   "transferStatus": "inProgress",
   "activeSessionID": "urn:satp:session:0x517BBF0c9B71f64b5807f644E1F1bacD3Afb3ec2:0xf3be
ac30c498d9e26865f34fcaa57dbb935b0d74:0x06012c8cf97BEaD5deAe237070F9587f8E7A266d",
   "sessionHistory": [
      "urn:satp:session:0x517BBF0c9B71f64b5807f644E1F1bacD3Afb3ec2:0xf3beac30c498d9e26865
f34fcaa57dbb935b0d74:0x06012c8cf97BEaD5deAe237070F9587f8E7A2777",
      "urn:satp:session:0x517BBF0c9B71f64b5807f644E1F1bacD3Afb3ec2:0xf3beac30c498d9e26865
f34fcaa57dbb935b0d74:0x06012c8cf97BEaD5deAe237070F9587f8E7A2888"
   ],
   "egress": {
      "gw1ID": "urn:satp:gateway:0x517BBF0c9B71f64b5807f644E1F1bacD3Afb3ec2",
      "nw1ID": "urn:satp:network:0xa9c28ce2141b56c474f1dc504bee9b010000008900000000000000
0000000012",
      "nw1EndPoint": "0x6e329e9c3653a0ab99e806e2267e8e4dff4d3fbb",
      "assets": [
         {
            "assetID": "urn:tar:eip155.137:0x897374acc81080d4fb818ee8c5f8822d937d8912",
            "transferExpiration": "2023-07-26T20:20:39+00:00"
         }
      ]
   },
   "ingress": {
      "gw2ID": "urn:satp:gateway:0xc24854b8457710f98ca254308b65e18b12ca0504",
      "nw2ID": "0xa9c28ce2141b56c474f1dc504bee9b010000000100000000000000000000000012",
      "nw2EndPoint": "0x0debadd980c7eec904cf57e105e3993688107aaf"
   }
}
```


Figure 2: Transfer Context example

5. References

[1] Hardjono, T., "SATP Core", Work in Progress, Internet-Draft, draft-ietf-satp-core, 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-satp-core>.

[2] Hardjono, T., "SATP Architecture", Work in Progress, Internet-Draft, draft-ietf-satp-architecture, 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-satp-architecture>.

[3] Avrilionis, D. and T. Hardjono, "Asset Schema Architecture for Asset Exchange", Work in Progress, Internet-Draft, draft-avrilionis-satp-asset-schema-architecture, 2024, <https://datatracker.ietf.org/doc/html/draft-avrilionis-satp-asset-schema-architecture>.

Authors' Addresses

Denis Avrilionis
Compellio S.A.
Email: denis@compell.io

Thomas Hardjono
MIT
Email: hardjono@mit.edu