INTERNATIONAL TRANSACTION LOG

DRAFT
TECHNICAL DESIGN SPECIFICATION (Version 1.0, Draft #5)

Non-paper

7 June, 2004

Revision History

Date	Version	Description	Author
01/03/2004	1.0	Draft #1, Partial (Issuance and Data Model)	Andrew Howard
31/03/2004	1.0	Draft #2, Partial	Andrew Howard
20/04/2004	1.0	Draft #3	Andrew Howard
25/05/2004	1.0	Draft #4	Andrew Howard
06/07/2004	1.0	Draft #5	Andrew Howard

Table of Contents

List of Figures

1. Introduction

1.1 Purpose

This document contains the technical design specifications for the International Transaction Log (ITL) to be developed and implemented by the Secretariat to the United Nations Framework Convention on Climate Change (UNFCCC). The purpose of the ITL is to monitor the validity of the transactions conducted by registries established by Parties under the mechanisms defined in Articles 6, 12 and 17 of the Kyoto Protocol and the modalities for the accounting of assigned amounts under Article 7.4 of the Kyoto Protocol.

The design of the ITL provides for the complementary functioning of supplementary transaction logs (STLs) developed by groups of Parties under the Kyoto Protocol. Such STLs are to conduct additional activities in relation to the transactions of those Parties under the Kyoto Protocol and under regional trading schemes. This complementary functioning is designed to avoid the duplication of validity checks and ensure consistent results between transaction logs. It further serves to integrate electronic communications between the relevant registries.

At time of writing, the only STL undergoing development is the Community Independent Transaction Log (CITL) for the European Union emissions trading scheme. This is being developed under Article 20 of EU Directive 2003/87/EC.

1.2 Intended Audience

This document is intended primarily for the technical experts involved in development and implementation of the ITL. It will also be of relevance to experts developing STLs and registries.

1.3 Scope

This document details the technical design of the ITL. It does not address the design of STLs. However, where applicable, specifications are provided for the ITL side of the data processing, relevant to STLs and the transmission and receipt of electronic communications to and from STLs.

Furthermore, this document does not address the design of registries. It relies upon the technical specifications of the data exchange standards for details relating to the ITL processing of data relevant to registries and for transmission and receipt of electronic communications to and from registries.

1.4 Definitions, Acronyms, Abbreviations, and Terminology

See Section 1.4 and Annex A of the Data Exchange Standards for Registry Systems under the Kyoto Protocol: Technical Specification (Version 1.0, Draft #5, as referred to below).

1.5 Derivation Documents

- Data Exchange Standards for Registry Systems under the Kyoto Protocol: Functional Specifications (Version 1.0)
  à http://unfccc.int/sessions/workshop/281103/documents.html

- Data Exchange Standards for Registry Systems under the Kyoto Protocol: Technical Specification (Version 1.0, Draft #5)

à http://unfccc.int/sessions/workshop/150604/index.html

- Decisions 15-18/CP.7 on the mechanisms under the Kyoto Protocol
  à Document FCCC/CP/2001/13/Add.2
  à http://unfccc.int/resource/docs/cop7/13a02.pdf
- Decision 19/CP.7 containing general requirements for the ITL and registries and modalities for the accounting of assigned amounts under the Kyoto Protocol
  à Document FCCC/CP/2001/13/Add.2
  à http://unfccc.int/resource/docs/cop7/13a02.pdf

- Decision 24/CP.8 containing general design requirements for the data exchange standards
  à Document FCCC/CP/2002/7/Add.3
  à http://unfccc.int/resource/docs/cop8/07a03.pdf

- Decision 19/CP.9 on the modalities and procedures for afforestation and reforestation project activities under the clean development mechanism in the first commitment period of the Kyoto Protocol
  à Document FCCC/CP/2003/6/Add.2
  à http://unfccc.int/resource/docs/cop9/06a02.pdf

2. Assumptions and Standards

2.1 Assumptions

This document is based upon the derivation documents specified in Section 1.5. In particular, it is based upon the constraints and specifications contained in the Technical Specification for Data Exchange Standards. This document assumes that the ITL and an STL would operate as companion applications. Critical assumptions from this document include:

- Communication between the ITL and registries will occur over a hardware VPN;

- Communication between the ITL and registries will occur in real time to the extent possible;

- The ITL will have a Communications Hub that, where applicable, routes communications to the relevant STL;

- The ITL will contain information on CDM Projects provided by the CDM Executive Board and on Joint Implementation Projects (JI Projects) provided by the Article 6 Supervisory Committee; and

- The ITL will contain information from the compilation and accounting database maintained by the UNFCCC Secretariat (for example, on allowable Issuance levels and the eligibility of Parties to participate in the mechanisms under the Kyoto Protocol.riat.

2.2 Standards

The ITL design and development will be based upon the following standards:

- SOAP
  http://www.w3.org/TR/2000/NOTE-SOAP-20000508

- XML
  http://www.w3.org/TR/2000/REC-xml-20001006

- IDEF1X Database Standards
  http://www.itl.nist.gov/fipspubs/idef1x.doc

- WSDL
  http://www.w3.org/TR/wsdl

3. General Requirements

The principal functionality of the ITL is to route and verify transactions that are received from national registries and the CDM Registry. These communications must be secure and processed as real-time transactions. The functional requirements for data exchange standards specify the use of Web services sending encrypted messages over the Internet. Communications must be protected from modification or interception in transit. Additionally, all messages from registries must be authenticated.

Communications can be initiated by either a registry or the ITL with an immediate response expected. These communications are transported through the use of Web services. Actions executed by registry Web services may be asynchronous and processed in real time or can be processed as jobs. The only immediate action that a Web service must respond to is an acknowledgement that the message was received and passed checks for authentication, and that the XML message format meets specifications. Calls to ITL Web services must be processed as soon as possible so that messages can be passed back to the Initiating Registry or forwarded on to another registry without delay. While registries must process a request from the ITL and send a responding message back within a 24-hour period, the ITL must process requests in a first-come, first-served fashion. If a message contains transaction information that involves a Party for which an STL has been established, the contents of that message are forwarded to the relevant STL for further processing.

The ITL also contains functionality to ensure the accuracy of data maintained by each registry and to ensure the consistency of transaction and unit information between the ITL and a registry. The process of comparing registry and ITL data and correcting data inconsistencies is called "reconciliation." The Communications Hub of the ITL may contain additional Web services for supplementary programs. Where these independent Web services are necessary, the ITL Communications Hub will route messages directly to the STL upon receipt. The only checking performed will be to ensure that the message received is well formed and accurately identifies the recipient.

3.1 Technical Architecture Overview

To provide this functionality, the ITL requires:

- Web services and Simple Object Access Protocol (SOAP) for the transport and delivery of XML messages;

- Use of a hardware-based Virtual Private Network for IP authentication and decryption of messages; and

- Digital Certificates to ensure authentication.

Figure 3.1 provides an overview of how messages are sent from Transferring registries to the ITL. The ITL's VPN checks authentication; thereafter, the Communications Hub Web service receives and logs the incoming message. The ITL processes the transaction and routes the message on to an STL, if appropriate. Specifications for the technical architecture of the ITL are contained in Section 4.

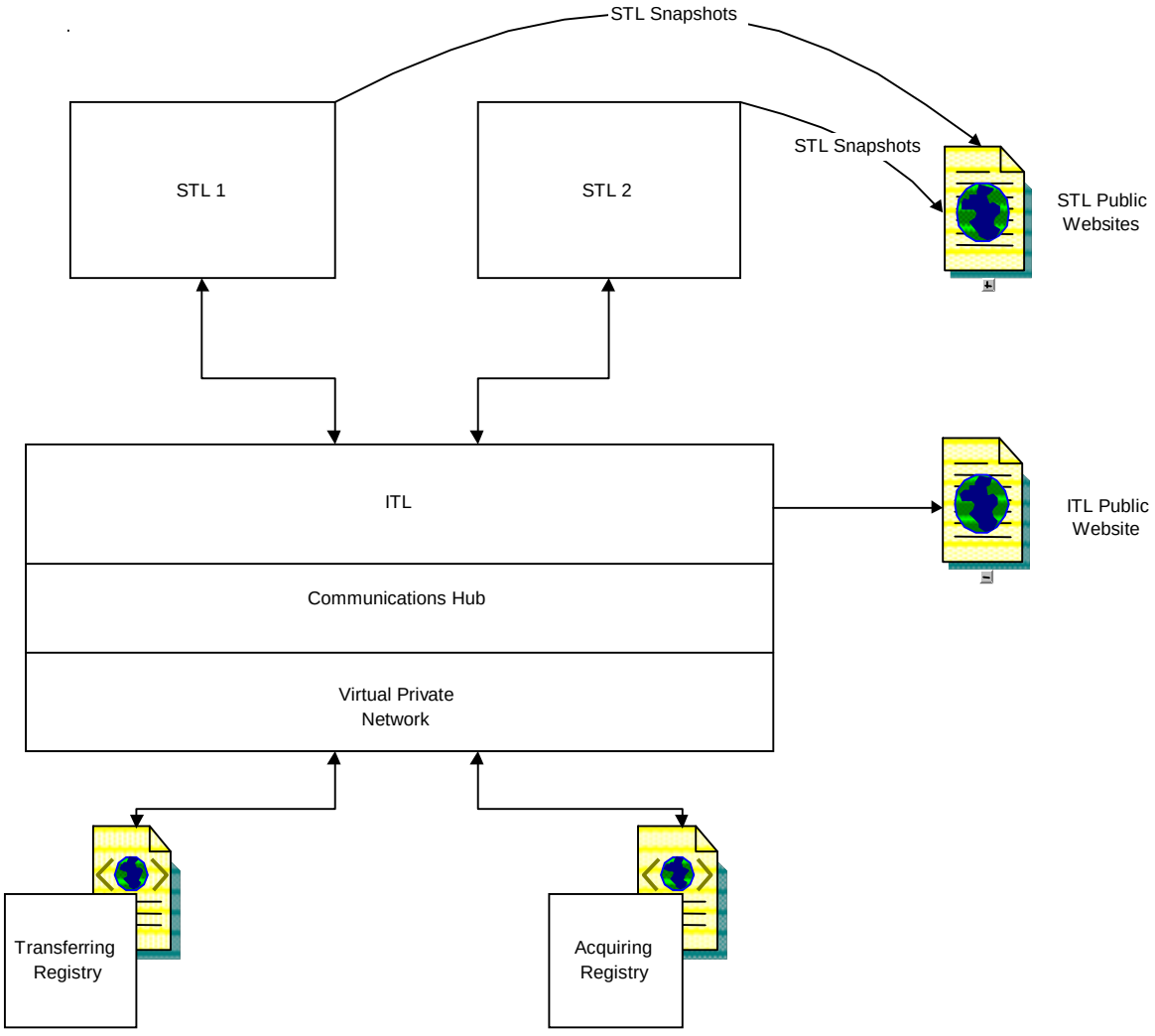
3.2 Test and Acceptance Environment

The ITL will provide both test and production environments for the entire application and metadata schemas supporting the capability for registries to test their communications components prior to going into production.

3.3 Message Exchange and Transaction Verification

The messages that are transported to and from the ITL must follow the data exchange standards that are the basis for the communication with the national registries of Annex I Parties, the CDM Registry, and STLs. These standards provide for the implementation of a common data transfer format and common functionality to ensure accurate, transparent and efficient data transfer between registries and the ITL.

Figure 3.1: Overview of Technical Architecture



Transactions occurring between two Parties for which no STLs are applicable are not routed through an STL. Instead, they are sent directly to the other registry.

For transactions between two Parties for which an STL is applicable, the results of the checks performed by the STL are communicated back to the ITL. If no issues or discrepancies are identified by the STL, the transaction proceeds to the Acquiring Registry. If the STL discovers a discrepancy preventing the transaction from being processed further, the ITL is given this information and a message is sent to the Transferring Registry that the transaction cannot be completed due to a specific error.

3.4 Communications Hub and Message Queue

The Communications Hub hosts a message queue which processes all incoming messages. The purpose of the queue is to receive and store messages and to provide sufficient scalability and throughput during peak processing time. When messages are retrieved for processing they are checked for their timestamp of arrival. If this timestamp exceeds 24 hours from the current time, the message is rejected as having expired and the registry is notified that the message is invalid.

There are three queues servicing incoming messages from registries: a transaction queue, an account management queue and a request for information queue.

3.4.1 Transaction Queue

The transaction queue manages high priority transaction proposals and subsequent notifications. This queue has the highest priority and is checked first for incoming messages.

3.4.2 STL Account Management Queue

STL Web service messages containing information on account data, including on installations and designated contacts and representatives, are held in a separate account management queue before being directed to the appropriate STL. The ITL does not record or track any information contained in such account management messages and only performs the role of routing the messages onward.

3.4.3 Information Request Queue

All other messages, which include reconciliation responses, transaction status requests, and other administrative processes are managed by this lowest priority queue.

3.5 Database Model

Data for the ITL will be maintained in a secure, normalised Oracle database containing all relevant tables to hold all the data for supporting logs, registry holdings, transaction history, and reconciliation history. The entity relationship diagrams and data dictionary in Annex B and Annex C define the application schema.

The database is presented as five major submodels: Registry, Transaction Process, Reconciliation, Projects, and System Data. Each submodel represents key relationships around a set of primary entities. The submodels have dependencies on other submodels.

3.5.1 Registry Submodel

The Registry Submodel contains information pertaining to the operations of each registry. Some elements of this information are input from the compilation and accounting database maintained byt the UNFCCC Secretariat. The tables track and record the following information:

- Registry Web service URL and port for both test and production environments;
- Operational status and status history;
- Eligibility status of Parties to participate in the mechanisms under the Kyoto Protocol;

- Registry contact information and the type of relationship to, or responsibility a person has, for a registry;

- Allowable Issuance quantities or other unit restrictions; and

- Current unit holdings in each registry, by account type.

See Annex B, Figure B3 for detailed information on the Registry Submodel.

3.5.2 Transaction Process Submodel

Any transaction received by the ITL follows a sequence of processes, which are recorded in tables within the Transaction Process Submodel. These tables track and record the following information:

- Receipt of the message from a registry;

- Storage of incoming message;

- Logging of transaction and unit serial blocks in the transaction;

- Tracking of transaction status as various checks are performed;

- Recording of the appropriate responses to checks as applicable to each unit block;

- Identification of units in an ongoing transaction as "unavailable" until a transaction is finalised;

- Units that are currently in an inconsistent state as identified by a reconciliation process;

- Replacement of tCERs or lCERs;

- Expiry Date Changes for tCERs and lCERs; and

- Routing of transaction to an STL for further processing (if transaction involves party in a supplementary program.

See Annex B, Figure B4 for detailed information on the Transaction Process Submodel.

3.5.3 Reconciliation Submodel

Reconciliation occurs as a scheduled job determined by the ITL Administrator, as negotiated with each registry, or as requested by an STL. The reconciliation tables track and record the following information:

- Each instance in which the ITL requests reconciliation information from a registry;

- The date and time (DateTime) each reconciliation stage occurs, along with the status during each stage of processing;

- Logging of inconsistent unit blocks conflicting with information held in the Registry Unit Holdings table;

- Recording of response codes identifying the errors with the inconsistent blocks.

See Annex B, Figure B5 for detailed information on the Reconciliation Submodel.

3.5.4 System Data Submodel

Other tables record and retrieve system data that support the major processes. These tables track and record the following information:

- All inserts, updates, and deletes for all primary transaction tables (exclusive of the transaction and reconciliation logs);

- Current version number for the ITL Administration Application (ITL AA) and ITL Technical Specification for Data Exchange Standards (DES);

- All checks and the appropriate response codes associated with an error or successful check; and

- ITL system parameters that are used to record system defaults, toggles, and various other parameters.

See Annex B, Figure B6 for detailed information regarding these tables.

3.5.5 Project and Notification Submodel

Periodically, it will be necessary for the ITL to notify a registry of actions that need to be taken on units. Examples include Carry-over or Cancellation of a unit at the end of a Commitment Period, or notification from the ITL than an tCER or lCER will expired in 30 days. The ITL is also responsible for notifying registries regarding necessary actions stemming from CDM Project actions by the CDM Executive Board.

The notification tables track the following information:

- CDM and JI Projects approved under the Kyoto Protocol;

- The content of each notification message;

- The Project, if any, related to reason for the the notification; and

- The registries that receive each notification and the number of units or specific unit blocks, if any, each registry must act upon.

See Annex B, Figure B7 for detailed information regarding these tables.

3.5.6 Database Authorisations

The database will maintain several types of accounts to provide access to ITL data. The following figure identifies the various access account types needed, and describes the level of authority and role for each.

Figure 3.2: Database Accounts

Account Type	Privileges	Role
Database Administrator	DBA	Has authority to perform backup and recovery, manage jobs, and manage all other physical database operations.
ITL System Administrator	Read, Write, Delete on all tables in both application and metadata schemas	Has authority to manage logs, review ongoing transactions, and resolve discrepancies. This role may require authority to modify data due to reconciliation or discrepancy corrections.
Public Web service	Read, Write to log tables	Logs all incoming and outgoing messages.
Private Functions	Read, Write, and Delete to all application data	Processes all transaction data.
Report	Read selected data	Web query services and public ITL website.
Auditor	Read all data	Read-only access to all data.

3.6 General Flow of a Transaction Message

All messages received by the ITL follow a set of standard processing procedures. Figure 3.3 shows the general outline of these steps.

1. The incoming message is checked for authentication. These checks are performed by the VPN which first blocks any message from an unidentified sender by examining the digital certificate for authentication.

2. If the authentication check passes, the message must match the WSDL structure as specified by the Web service on the ITL. The message must contain version identifiers which are consistent with the current version.

3. If the Web service accepts the message, it is saved to an incoming message file and is briefly placed in a message queue to be processed as quickly as possible. Each message from the message queue in time-received order and checked to determine that it is less than 24 hours old. If the message is valid, it is processed against a set of message validity, registry, data integrity and message sequence checks. These checks determine whether the registries are eligible and operating, and whether the contents of the message meet the minimum requirements to continue further processing. Each check returns a specific

response code if an error is identified. These checks and response codes are described in Section 5.4.

4. After these checks have been performed, the name of the message file is recorded to a message log (which keeps a record of every incoming and outgoing message). Multiple XML files that are received can be zipped and stored in master files.

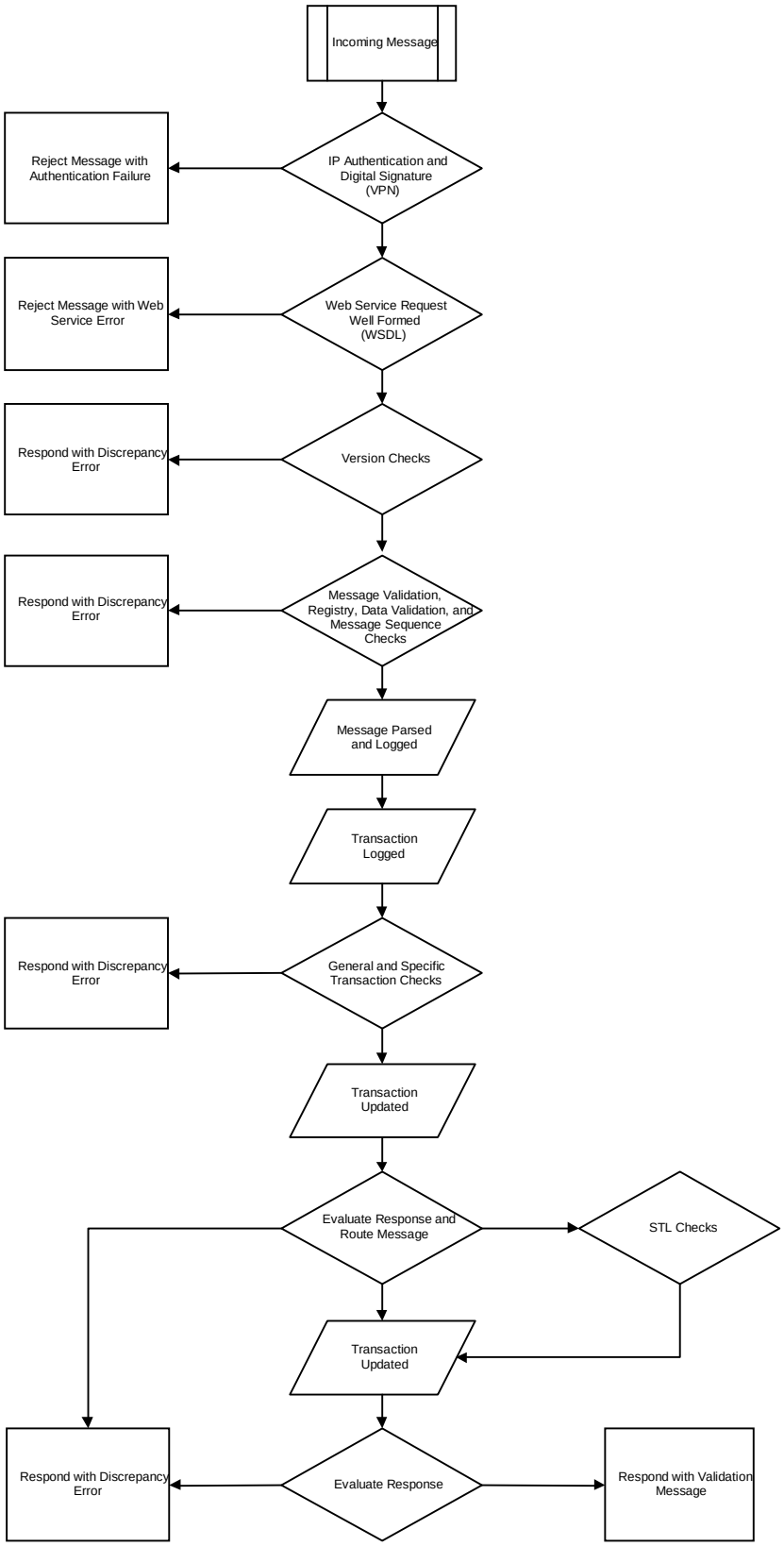
5. If at any point during these checks an error is discovered, an HTTP SOAP response is sent back to the registry indicating a failure and the reason for the failure.

6. Otherwise, the transaction is evaluated against general transaction checks and checks that are appropriate for the specific transaction type identified in the message. Each check returns a specific response code if an error is identified. These checks and response codes are described in Section 5.4.

7. The ITL evaluates the results of all transaction checks. If all prior checks have been performed without error by teh ITL, and if the HTTP SOAP request involves a registry participating in a supplementary program, the contents of the request are forwarded to the appropriate STL for further processing. If the ITL finds discrepancies with the transaction during the transaction checking phases, the message will not be forwarded to the STL, but will be returned to the registry containing the appropriate response codes identifying what checks failed and the reason.

8. Throughout this process, the ITL records the transaction, the transaction status and response codes relating to the transaction. If there is an unhandled exception in the processing of the message, any changes made to the database for this transaction will be rolled back. Prior to the first database change after the message is retrieved from the queue, a new database transaction session will be started. Once the message has been evaluated, all the changes made during the session will be committed at once. If the commitment point is never reached, the changes will be rolled back.

400 **Figure 3.3: General Flow of a Transaction Message through ITL Processing**
401
402



403
404

4. Technical Architecture Specification

The Technical Architecture specification provides information on the physical arrangement (topology) of the ITL and its connection to national registries and STLs. It also includes information on operational support and test environments.

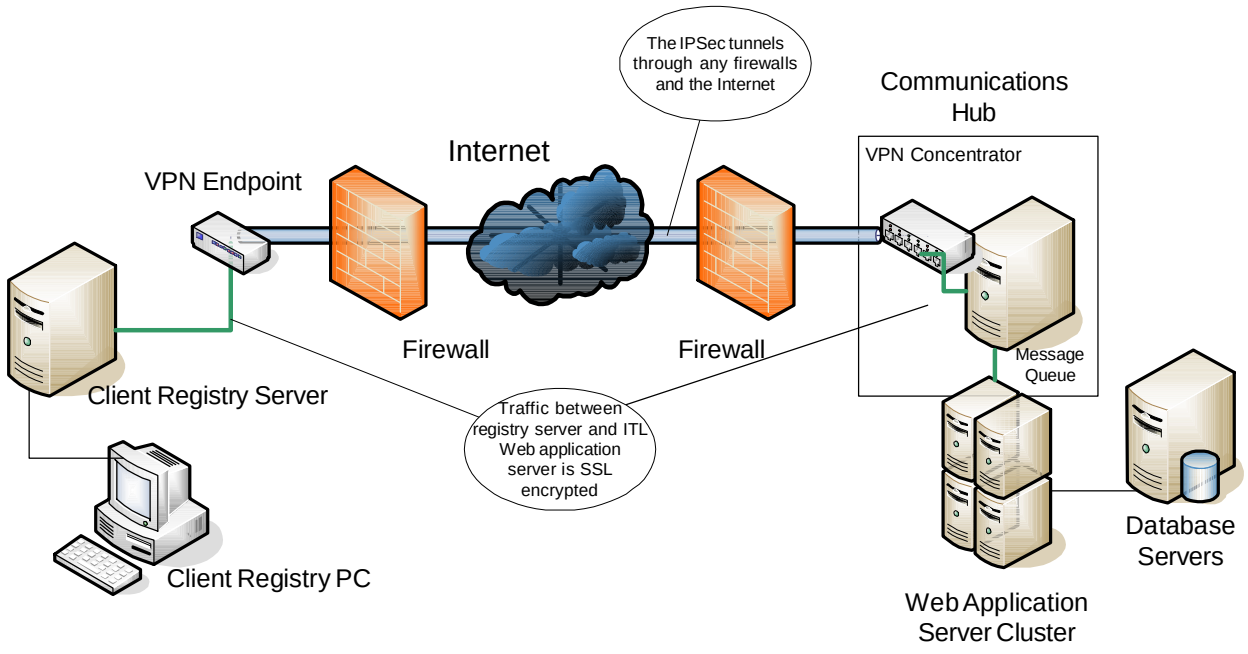
A registry will send communications to and receive them from the ITL through the use of messages over the Internet using SOAP. Communications that occur between the two are bi-directional, meaning that either a registry or the ITL can initiate a Web service request to the other and anticipate a response.

4.1 Technical Architecture

This section identifies the physical technical architecture in which the transporting of messages can occur in a secure and robust fashion.

The ITL will send communications to and receive communications from registries through the use of SOAP messages. These messages are transferred via secure transmissions as outlined below. The principle components in this architecture are a Communications Hub (CH), the ITL and a connection to an STL. The CH and ITL will accept communications from and initiate communications to connected client registries using the Internet. All communications between the ITL and client registries will be protected using standard VPN technology as they travel over the Internet. Figure 4.1 provides a graphical overview of the system.

Figure 4.1: Architecture Overview



4.2 Application Servers

The ITL shall consist of two primary systems: a Communications Hub server cluster that accepts incoming Web service requests, and a Web application server cluster that will process these requests, log them to disk, and then communicate the requests to a database server for further processing.

4.2.1 Web Application Server

The Web application servers will be low- to mid-range servers, as a clustering configuration does not require high-end scalable servers. If more processing power is required, additional servers are added instead of replacing or upgrading current servers. The initial Web cluster shall consist of two Web servers configured as follows:

- Intel XEON 3.0 GHz or better dual processor configuration;
- 1 GB RAM, with capacity for expansion to 8 GB RAM;
- 4 available drive bays;
- Hardware RAID1 (mirrored) drives for operating system. Drive capacity of mirror should be minimum 18 GB; and
- Hardware RAID1 (mirrored) drives for log storage. Drive capacity of mirror should be minimum 73 GB.

4.2.2 Communications Hub and Queue Server

The Communications Hub server will be a low-range server that can be clustered if needed. The server is only required to store queue data. All three queues at peak periods are estimated to need no more than 2 to 3 GB of storage space. The initial queue server shall be configured as follows:

- Intel XEON 3.0 GHz or better dual processor configuration;
- 1 GB RAM;
- Hardware RAID1 (mirrored) drives for operating system. Drive capacity of mirror should be minimum 6 GB; and
- Hardware RAID1 (mirrored) drives for queue and file storage. Drive capacity of mirror should be minimum 40 GB.

4.2.3 Test/Production Ports

The Communications Hub allows access to two versions of the application available to the client registries. The first service for production operation will run on the standard HTTP/HTTPS ports and support a copy of the current production version of the application. A second service for testing will run on non-standard ports, and provide a proving ground service. This second service will allow client registries to connect and run trial transactions against the ITL test database for verification of their compliance with the DES.

4.3 Hardware Load Balancing

Estimates for the ITL indicate that large amounts of traffic will access the site in compressed timeframes. A majority of the traffic will occur at the end of a compliance period with concentrated requests occurring in one 24-hour period. This usage pattern requires a robust solution for managing traffic flow to the Web application cluster. There are several methods for directing this traffic flow. The first involves Round Robin DNS, where the DNS server will successively give out the IP address of the next server in the cluster as requests come in. The second involves using clustering software either at the operating system level, or at the application server level. In both of these methods, an application server or operating system failure can potentially disrupt the functionality of the cluster. To mitigate this, traffic flow to the Web cluster should be managed by a dedicated hardware device. This device should communicate with nodes in the cluster and intelligently direct traffic to the least loaded node. An example of an acceptable device to manage the cluster traffic is Cisco's LocalDirector. Information on the LocalDirector is available at: http://www.cisco.com/warp/public/cc/pd/cxsr/400/index.shtml.

4.4 Database Server

The database server will be a mid-range server capable of processing and storing the required data. The database is estimated to hold 100 GB of data at launch with an expected growth of 15 percent per year. Consequently, the system should be upgradable to meet unexpected loads, especially with regard to CPU and memory requirements. The initial configuration follows:

- Intel XEON 3.0 GHz or better dual processor configuration;
- 4 GB RAM, with capacity for expansion to 8 GB RAM;
- 4 drive bays;
- Hardware RAID1 (mirrored) drives for operating system. Drive capacity of mirror should be minimum 18 GB;
- Hardware RAID1 (mirrored) drives for archive logs. Drive capacity of mirror should be minimum 18 GB;
- Hardware RAID10 (mirrored, striped) drives for database data files and system logging in an external array. Array capacity should be minimum 100 GB. If possible, disks should be arranged through multiple controllers for maximum performance; and
- An external drive array with the same capacity as the RAID10 array, unmirrored, for backup snapshots.

The need for large and expanding amounts of reliable, mirrored data may necessitate the use of a network-attached storage device, such as a NetApp FAS200 series, or EMC NetWin appliance. These and similar products can significantly reduce system downtime for backups and maintenance, as well as provide effectively limitless disk expansion possibilities. Many are also operating system-independent.

A test environment for registry testing will also be made available. The test system is expected to be available during business hours, under moderate stress, and to contain a reasonable subset of data from the production server. It will not contain any non-reproducible transactions. The

performance of the server will depend on the amount of testing taking place at any given time. The additional test database system will be configured as follows:

- Intel XEON 3.0 GHz or better dual processor configuration;
- 1 GB RAM, with capacity for expansion to 8 GB RAM;
- 4 available drive bays;
- Hardware RAID1 (mirrored) drives for operating system. Drive capacity of mirror should be minimum 18 GB;
- Hardware RAID5 (striped) drives for database files and system logging in an external array. Array capacity should be minimum 100 GB. If possible, disks should be arranged through multiple controllers for maximum performance; and
- An external drive array with the same capacity as the RAID10 array, unmirrored, for backup snapshots.

Websites for NAS devices can be found at the following Web addresses:

- FAS200: http://www.netapp.com/products/filer/fas200_ds.html
- EMC NetWin: http://www.emc.com/products/networking/servers/netwin/index.jsp.

4.5 VPN Equipment

The VPN equipment used in the ITL VPN network will need to support between 25 and 50 site-to-site VPN connections. This requires adequately scalable hardware at the headend ITL location, and also compatible hardware at the client registry locations.

Equipment at the ITL and client registry locations must support the following features:

- A site-to-site VPN connection;
- Encryption of data using the 3DES encryption algorithm;
- Data hashing using 160bit SHA-HMAC for data integrity;
- Identification of itself either through unique pre-shared keys, or by use of digital certificates verifiable by an external Certificate Authority (CA);
- In some cases, Split Tunneling capability may be required; and
- In some cases, NAT capabilities may be required.

4.5.1 VPN Equipment for the ITL

VPN equipment for the ITL will maintain between 25 and 50 VPN connections in a hub-to-spoke topology. As such, all IPSec functionality will be implemented at the hardware level. The ITL VPN device should be able to maintain throughput without degradation for all active site-to-site

connections up to the limit of the incoming ISP bandwidth. An example of VPN equipment adequate for ITL VPN connectivity includes the Cisco PIX line of firewall/VPN devices. Information on the Cisco PIX firewalls is available at: http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/.

4.5.2 VPN Equipment for Client Registries

VPN equipment at the client registries should be dedicated devices that can reliably terminate the VPN connection to the ITL as well as maintain acceptable performance levels. It is likely that, for maximum compatibility, the client registries should adopt a hardware solution from the same vendor as the ITL, and run the same code revisions. Although IPSec is an RFC-based industry standard, there are many areas of the RFC that leave mechanisms open to interpretation. This can lead to incompatibilities between vendor implementations that undermine or disrupt the VPN functionality. An example of VPN equipment adequate for client registry VPN connectivity is the Cisco PIX line of firewall/VPN devices. Information on the Cisco PIX firewalls is available at: http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/.

4.6 Secure Transmission

The following section describes the physical arrangement needed to support the secure transmission of SOAP messages being sent to and from the ITL and national registries.

4.6.1 IPSec VPN

All communications between the client registries and the ITL will be confidential and the integrity of data will be maintained while in transit over the Internet. In order to accomplish this, a site-to-site VPN infrastructure will be set up between the ITL and all client registries. A hardware-based VPN concentrator will be installed at the ITL and configured for site-to-site IPSec VPN connections for each client registry. The IPSec VPN will provide for site-to-site authentication, data integrity, and data encryption.

4.6.1.1 Site-to-Site Authentication

IPSec VPN configurations provide for authentication at the two endpoints of the VPN connection. The ITL will identify and authenticate the remote client registries via the IPSec VPN connection using a digital certificate provided by a trusted CA.

4.6.1.2 Data Integrity

IPSec VPN ensures the data integrity of all communications passing through the VPN tunnel and over the Internet. Each packet of traffic is hashed and signed, using the authentication information that was used to establish the VPN connection. If any data traffic packets are modified in transit from one VPN endpoint to the other, the hashes and signatures will not match the data in the packets, and they will be discarded and resent by the originator.

4.6.1.3 Encryption

IPSec VPN ensures data confidentiality by encrypting the data in data packets, or the entire packet as required using Triple DES (3DES). This encryption addresses only the network traffic itself, not the application-level SOAP communications.

4.6.2 Secure Socket Layer (SSL)

SSL will be used for all communications between client registry and ITL application servers. SSL provides application server-to-application server authentication as well as data encryption. Since the IPSec VPN provides only site-to-site authentication, the SSL will authenticate the specific client registry communications to the ITL (in the case where multiple registries are hosted on the same site). Additionally, SSL protects any communications that may pass over networks at the client registry site before transiting the VPN to the ITL through an additional layer of encryption.

4.6.3 Certificate Authority

SSL requires the use of a trusted CA in order to realize the full benefit of positive authentication and secure encryption. Trusted CA services are provided commercially by several vendors, such as Verisign and Thawte. These vendors verify identity and issue certificates which can be used to positively identify an organization and encrypt data communications between the organization and other certificate holders. These vendors are already widely used and trusted worldwide, with a large percentage of online transactions via SSL using their certificates.

Due to the number of registry end points and size of the VPN, a third-party managed CA will be used to facilitate deployment of the VPN.

4.6.4 STL and ITL Transmissions

Communications between the STL and ITL function essentially the same as communications between the ITL and client registries. Both ITL locations will be linked by IPSec VPN, and all communications between the two sites will be secured.

4.7. ITL Regression Testing Environment

The ITL will on occasion undergo upgrades, optimizations, or patching. As this occurs, it is imperative that the ITL continue to process requests from registries running older versions of the Data Exchange Standards. In order to guarantee this, functional regression testing must occur offline in a laboratory environment before any changes are made to the production ITL environment. This test environment only needs to replicate the client registry and ITL application servers themselves, as the VPN infrastructure should at all times be transparent to the functioning of the client registry-to-ITL communications. The testing environment should consist of a basic network, one Web application server, and a lesser version of the database server.

A comprehensive set of functions or scenarios which constitute a fully functional ITL will be established. Methods for testing these scenarios will also be developed. Any proposed change to the system must successfully pass each test in succession before that change can be propagated to the production system. If a change does not pass each test, it cannot be moved into production. These tests should be designed in such a way as to provide stable and responsive system interaction to the user or client system.

4.8. Operations

Specific operational needs for this system will be dependent largely on the final choice of software architecture. However, operational targets do not vary greatly. To remain operational and make recovery from problems as easy as possible, the systems must be backed up regularly, and the database must be maintained in an effective state.

4.8.1 Operational Backups

Every production system must be backed up regularly to alternate media. This may include tapes that are rotated offsite, swappable external disk drives, or a combination of the two. Backups should include both system and database files. Additionally, Oracle must be run in ARCHIVELOG mode for maximum uptime, and the archives generated must themselves be stored between full database backups or complete hot-backup operations. A weekly full backup with incremental system backups should be sufficient. The full backups should include the quiesced database. The database can be quiesced, the datafiles copied to a more suitable location, and the database restarted before the backups, however. Exports are not a sufficient means of database backup.

Between backups, the archive destination must be monitored closely to make sure it does not fill up. When the archive destination is approaching full capacity, the archives must be backed up and deleted, or moved to a location with more space. This will ensure that the database will not stop while waiting for archive space to free up. The database must also be monitored regularly to check for areas of performance concern. These may include problems such as "hot spots" on disks, processes that are consuming too many resources or obtaining and not freeing locks, or inadequate space available for certain operations. Monthly database analysis of tables and indices may prove to be invaluable in maintaining optimum performance.

Additionally, occasional exports of data in the database can ease maintenance in some circumstances. They provide a means of transferring data to other systems for examination without affecting the production installation.

4.8.2 Disaster Recovery

In the event of a disaster, recovery will depend largely on the state of the system backups. With Oracle running in ARCHIVELOG mode, it will be possible to recover the database up to the last committed transaction before the failure, provided a usable full backup and all intermediate archive logs are available. For this reason, all backup tapes must be rotated offsite on a periodic basis. The proposed setup will entail making database copies to local disks first, and then backing up those disks to tape.

This setup has the added benefit of allowing the most recent backup tapes to be sent offsite, thus allowing for optimum recovery of the most recent backups. In the event of a hardware-based, non-catastrophic system failure, the most recent backups and archive logs will exist locally on disk. The server hardware can be replaced quickly, and the backups restored with minimal downtime. In the event of a total system loss, the most recent backups can be returned from offsite storage and applied to a new system in a different location. Archive logs must also be present at the offsite location to permit up-to-failure recovery.

A formal disaster recovery plan must be implemented based on the final configuration of the system. Without absolute redundancy in the network, system, storage, and external hardware, no system configuration will be available without downtime, but good backups are very important.

5. Transaction Processing

5.1 Scope of Technical Design Specification for Data Exchange Processes

This section of the Technical Design Specification addresses the design and development of all functionality necessary to support validation of transactions by the ITL and all other data exchange between registries, the ITL, and an STL.

This section contains design specifications on the following functional processes:

- Issuance;
- Conversion;
- External Transfers;
- Internal Transfers (Cancellation and Retirement);
- Carry-over;
- Replacement; and
- Expiry Date Change.

5.2 Design Elements

The design of the data exchange processes for the ITL is presented in this document through the following elements:

- A relational database model;
- A logical data dictionary;
- Process flow diagrams; and
- Web service and function definitions.

5.2.1 Database Model

An overview of the database model is presented in Section 3 above, in the entity-relationship diagrams in Annex B and in the data dictionary in Annex C. The database is a core design element and is referenced throughout the checks and function descriptions.

5.2.2 UML and Process Flow Diagrams

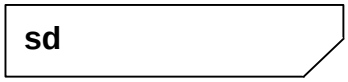
This design documentation utilises the activity and process flow diagrams based on Unified Modeling Language (UML). These diagrams represent the functionality within the ITL which is necessary to review, process, respond to, and store all HTTP SOAP requests. The process flow diagrams are high-level representations which are designed to capture process logic. Within each diagram, specific functions (representing specific programming logic) are identified. These diagrams are annotated with text to help non-technical readers interpret them more easily.

The data process diagrams show business logic and decision structures. Some decisions can be functionally simple or very complex. When the functionality is complex, the box for that component on the behaviour diagram will contain a "ref" symbol to indicate that a detailed sub-diagram is provided. See Figure 5.1, Key to UML Diagrams, for reference.

For the purposes of this technical design document, the UML diagrams in Section 5.5.2 focus on those Web services and functions in the Communications Hub and ITL "swim lanes."

Any references to the functions identified in the registry swim lanes can be found in the Data Exchange Standards for Registry Systems under the Kyoto Protocol.

Figure 5.1: Key to UML Diagrams

UML Element	Description
Actors & swim lanes 	At the top of each diagram, the participants in the process are represented by a word preceded with a colon (:). Actions involving a participant are presented in the "swim lane" which is directly underneath the box, and represented by a dashed vertical line.
	This symbol indicates that the diagram is a sequence diagram. The symbol is followed by the name of the process.
	This symbol indicates that there is a secondary sub-diagram for the component which provides additional detail of the functionality.

5.2.3 Functions and Objects

An object is an entity in which various programming functions can be performed relating to a task. Each function requires specific information to be provided to perform its task ("inputs") and returns specific information after its operations are performed ("outputs"). In all cases, functions generally have objects as inputs and outputs. Although not all objects are identified in these technical specifications, several key objects have been specified and are referenced in these functions. See Annex E for details.

These technical design specifications define the inputs, database interactions, and outputs for each process. The document describes in specific data terms the programming logic that should be implemented for all functions required by the ITL.

Technical information for each function, including required inputs and outputs is included in Annex E.

5.3 Summary of Transaction Types

5.3.1 Issuance (Transaction Type 1)

The Issuance of AAUs is undertaken by a Party in an account in its national registry on the basis of its assigned amount (which is in turn calculated on the basis of greenhouse gas emissions during the base year). The Issuance of RMUs is undertaken by a Party in its national registry on the basis of its net removals of greenhouse gases through LULUCF activities. The Issuance of CERs, tCERs, or lCERs into a pending account is undertaken by the CDM Executive Board, in the CDM Registry, on the basis of certified reductions in greenhouse gas emissions or certified

removals of greenhouse gases through a CDM project activity. The validity of such Issuance is checked by the ITL.

5.3.2 Conversion (Transaction Type 2)

The Conversion of AAUs and RMUs to ERUs is undertaken by a Party in an account in its national registry. AAUs and RMUs are converted to ERUs on the basis of verified reductions in greenhouse gas emissions and verified removals of greenhouse gases through a JI Project. The validity of such Conversion is checked by the ITL.

5.3.3 External Transfer Process (Transaction Type 3)

The External transfer of AAUs, RMUs, ERUs, CERs, tCERs and lCERs to another registry is undertaken by a Party, an entity, or the CDM Executive Board, on the basis of the amount proposed by the transferor. The validity of such External transfers is checked by the ITL.

5.3.4 Internal Transfers Involving Cancellations and Retirements (Transaction Types 4 and 5)

The Internal transfer of AAUs, RMUs, ERUs, CERs, tCERs or lCERs to Voluntary, Net Source, Non-compliance or Excess Issuance Cancellation accounts is undertaken by a Party, an entity, or the CDM Executive Board, on the basis of the amounts proposed by the transferor. The Internal transfer of these units to a Retirement account is undertaken by a Party or an entity, on the basis of the amounts proposed by the transferor. The validity of such Internal transfers is checked by the ITL.

5.3.5 Replacement of tCERs and lCERs (Transaction Type 6)

The Replacement of tCERs or lCERs occurs through the internal transfer of AAUs, RMUs, ERUs, CERs, tCERs or lCERs to a Replacement account and is undertaken by a Party or an entity, on the basis of the amounts proposed by the transferor. The validity of such Replacement is checked by the ITL.

5.3.6 Carry-overs (Transaction Type 7)

The Carry-over of AAUs, ERUs and CERs is undertaken by a Party in an account in its national registry, on the basis of the amount of units in holding accounts (i.e., units that have not been cancelled or retired for that Commitment Period) after expiration of the additional period for fulfilling commitments (the "true-up period"). The units remain in the same account and the serial numbers remain unchanged. The effect of the Carry-over transaction is to give recognition, both within the registry and the ITL, to the validity of the units in the next Commitment Period. Any units in holding accounts that are not carried over in this manner are to be cancelled. The validity of such Carry-over is checked by the ITL.

5.3.7 Expiry Date Change (Transaction Type 8)

An Expiry Date Change is undertaken by a Party for tCERs and lCERs. For tCERs, this transaction may be necessary to change the expiry date of tCERs issued prior to 2006 if the end of the second Commitment Period is determined not to be 31 December 2017 (the expiry date with which tCERs would initially be issued). For lCERs, this transaction will occur when the Executive Board approves the extension of lCERs for a Project for an additional period. The ITL ensures that these expiry date updates are consistent with the Project Approvals, and updates the tCER and lCER expiry dates in the ITL database.

5.3.8 Internal Transfers and Other Supplementary Transactions Routed to STL (Transaction Type 10)

The validity of Internal transfers of AAUs, RMUs, ERUs, CERs, tCERS or lCERs among holding accounts is not checked by the ITL. For these transactions, the ITL conducts general transaction checks necessary to mark the blocks as unavailable due to a pending transaction and splits unit blocks as necessary. The ITL records the results of this basic step and routes them to the relevant STL for further evaluation against STL rules and requirements.

5.4 Transaction Message Checks

As a message is received and processed, it is checked at various levels. Preliminary checks that fail return a failure response immediately back to the registry via a SOAP response. Other checks that may involve interaction with the database take longer to respond to and are not returned immediately. The action taken to respond to failure of these checks depends on the point in the process where the message failed. Failures due to transaction checks are returned in the ResponseObject in an HTTP SOAP response initiated by the ITL to the originating registry. Figure 5.2 below summarizes the check categories, the types of responses, and check actions take during processing.

Figure 5.2: Check Categories

Category	Response Code Range	Category Description	Action Upon Failure
Version and Authentication	1000 - 1299	Checks to authenticate sender and to validate version of DES during preliminary processing.	Message returned with response codes or HTTP Soap Error. Message not placed into message queue (unless only a minor version inconsistency is identified).
Message Viability	1300 - 1399	Checks to determine whether the message is viable when processed from the queue.	Message returned with response codes. Message not logged in the Transaction_Log table.
Registry Validation	1500 - 1599	Checks to validate status of registry during queue processing.	Message returned with response codes. Message not logged in the Transaction_Log table.
Data Integrity	2000 - 2999	Basic checks of data content including numeric ranges and validity of codes during queue processing.	Message returned with response codes. Message not logged in the Transaction_Log table.
Message Sequence for Registry Messages	3000 - 3499	Checks to validate message order and transaction status.	Message returned with response codes. Message not logged in the Transaction_Log table.
Message Sequence for STL Messages	3500 - 3999	Checks to validate message order and transaction status.	Message returned with response codes. Message not logged in the Transaction_Log table.

(cont.)

Figure 5.2: Check Categories (cont.)

Category	Response Code Range	Category Description	Action Upon Failure
General Transaction Checks	4000 - 4999	Checks applicable to all transactions involving unit blocks. These checks are applicable to all transactions	Message returned with response codes and transaction status. Message logged in the Transaction_Log table.
Transaction-specific Checks	5000 - 5899	Kyoto Protocol transaction checks specific to designated transaction types.	Message returned with response codes and transaction status. Message logged in the Transaction_Log table.
Registry Messages	5900 - 5999	Response codes generated by registries.	Response codes sent with transactions to other parties.

5.4.1 Check Phase

Messages will be implemented in phases consistent with the deployment of the ITL and the timing of the relevant requirements. Phase 1 indicates that the check must be implemented as a core requirement; Phase 2 indicates that the check is a Kyoto Protocol requirement that must be implemented in the timeframe of ITL operation prior to Commitment Period 1; Phase 3 indicates a check that can be implemented after the beginning of Commitment Period 1. In the following section, each check is described along with the appropriate response code number that is returned to the registry if the check fails, and the phase in which the check will be enabled is provided. The detailed specification for these checks can be found in Annex F: List of Transaction Checks. A description of all response codes can be found in the DES document, Annex C.

5.4.2 Version and Authentication Checks

Version and authentication checks are performed within the Communications Hub as preliminary checks upon receipt of the HTTP SOAP request and do not involve any interaction with the ITL database. Failures due to authentication checks and poorly formed XML content are returned as HTTP SOAP errors. If these checks are passed, the message is placed in the message queue for processing. See Figure 5.14, Preliminary Processing, for an activity diagram showing the flow of a message.

Figure 5.3: Version and Authentication Checks

Response Code	Check Name	Check Description	Phase
SOAP error	Certificate Check	If certificate is not recognized, message is rejected.	1
SOAP error	SOAP Identifier	Initiating Registry must be consistent with sender of SOAP message.	1

(cont.)

Figure 5.3: Version and Authentication Checks (cont.)

Response Code	Check Name	Check Description	Phase
SOAP error	WSDL Check	Message must conform to WSDL.	1
1031	Major Version	If Major Version number in transaction message does not match Major Version number for DES.	1
1032	Minor Version	If Minor Version number in transaction message does not match Minor Version number for DES.	1

5.4.3 Message Viability Checks

Messages are placed in one of three different queues and are processed on a first-come, first-served basis. The time in which the message is added into the queue becomes the official timestamp in which the ITL acknowledges receipt of the message. However, should the ITL database be unavailable for an extended period of time due to hardware failure, messages remain in the queue until they can be processed. Viability checks determine whether the message from the queue should be processed. See Figure 5.15, Queue Processing Checks, for further details on queue processing.

Figure 5.4: Message Viability Checks

Response Code	Check Name	Check Description	Phase
1301	Message Age	Message must be processed within 24 hours of submission.	1

5.4.4 Registry Validation Checks

After the message has been retrieved from the message queue and the location of the message file has been written to the message log, the ITL performs checks to determine if the registries involved in the transaction are identifiable and eligible to participate. See Figure 5.15, Queue Processing Checks, for further details on queue processing.

Figure 5.5: Registry Checks

Response Code	Check Name	Check Description	Phase
1501	Identify Registry	Must be contained in Registry table.	1
1503	Initiating Registry Available for Transactions	Initiating Registry status code must be equal to zero.	1
1504	Acquiring Registry Available for Transactions	Acquiring Registry status code must be equal to zero.	1

5.4.5 Data Integrity Checks for Transactions

This category of checks is performed by the Data_Integrity_Checks function to identify incoming messages containing data that fail basic data integrity checks. If any data in a message fail these checks, the message is returned to the sender with an appropriate response code. The message is not logged in the Transaction_Log table and is not processed further. Additionally, all data integrity checks are critical checks in that if they result in failure, no further checks should be processed. See Figure 5.15, Queue Processing Checks, for further details on queue processing.

Figure 5.6: Summary of Data Integrity Checks

Response Code	Check Name	Check Description	Phase
2001	Transaction Mask	Transaction ID must be comprised of a valid registry code followed by numeric values.	1
2003	Transaction Status Code	Transaction status code must be valid.	1
2004	Transaction Type Code	Transaction type must be (1) Issuance, (2) Conversion, (3) External, (4) Cancellation, (5) Retirement, (6) Replacement, (7) Carry-over, (8) Expiry Date Change, or (10) Internal/Supplementary Program.	1

(cont.)

Figure 5.6: Summary of Data Integrity Checks (cont.)

Response Code	Check Name	Check Description	Phase
2005	ERU, CER, tCER, lCER Check	If the unit is a CER, an ICER or tCER, an ERU, or an ERU converted from an RMU, a Project Identifier must be present.	1
2006	ERU Track Check	If a unit is an ERU, a valid Track must be present.	1
2007	Supplementary Transaction Type Checks	The Supplementary Transaction Type Code must be blank or valid.	1
2080	Initiating Account Identifier	Initiating Account Identifier must be greater than zero.	1
2082	Acquiring Account Identifier	Acquiring Account Identifier must be greater than zero.	1
2083	Account Type Code	Account Type must be valid.	1
2084	Unit Serial Range	The Unit Serial end block must be greater than or equal to the Unit Serial begin block.	1
2085	Unit Type Code	Unit Type must be valid.	1
2086	Unit Serial	Unit Serial start block and Unit Serial end block must have a value greater than zero.	1
2087	Supplementary Unit Type Code	Supplementary Unit Type Code must be valid.	1
2089	Notification ID/Registry	The Notification ID must be one that was sent to the registry.	1
2090	LULUCF Code	LULUCF activity code must be present in LULUCF Activity Code table.	1
2091	Commitment Period	Original or applicable Commitment Period must be less than 10.	1
2096	Track Code	Track must be blank, 1 or 2.	1
2098	Transaction Status DateTime	Transaction Status DateTime must be between 1/1/05 and current date.	1
2100	RMU Check	If a unit is an RMU, a valid LULUCF code must also be present for the unit block.	1

5.4.6 Message Sequence Checks for Transactions from Registries

After the data in the message have been checked, the ITL performs checks to ensure that the message received has been submitted in the proper sequence, including whether process status is consistent and appropriate. See Figure 5.15, Queue Processing Checks, for further details on queue processing.

Figure 5.7: Sequence Checks for Transactions from Registries

Response Code	Check Name	Check Description	Phase
3001	Transaction Identifier Does Not Exist	Transaction ID does not exist and transaction status = "Terminated," "Completed," or "Accepted."	1
3002	Transaction Status Out of Sequence for Prior Completed Status	Transaction status = "Completed" and prior transaction status = "Completed."	1
3003	Non-external Accepted Status	Check for transaction status = "Accepted" for Non-external transaction.	1
3004	Transaction Status Not Valid	Transaction status = "Checked (No Discrepancy)," "Checked (Discrepancy)," "STL Checked (No Discrepancy)," or "STL Checked (Discrepancy)."	1
3005	Transaction Status Out of Sequence for Prior STL Discrepancy Status	Transaction status = "Completed" and prior transaction status = "STL Checked (Discrepancy)."	1
3006	Transaction Status Out of Sequence for Prior Cancelled Status	Transaction status = "Completed" and prior transaction status = "Cancelled."	1
3007	Transaction Status Out of Sequence for Prior Terminated Status	Transaction status = "Completed" and prior transaction status = "Terminated."	1
3008	Transaction Status Out of Sequence for Prior Checked Discrepancy Status	Transaction status = "Completed" and prior transaction status = "Checked (Discrepancy)."	1
3009	Transaction ID Not Unique	Transaction status = "Proposed" and Transaction ID already exists.	1

5.4.7 Message Sequence Checks for Transactions from STLs

If a message has been received from an STL, the ITL checks to determine if the order of messages and statuses is consistent and appropriate. See Figure 5.15, Queue Processing Checks, for further details on queue processing.

Figure 5.8: Sequence Checks for STL Messages

Response Code	Check Name	Check Description	Phase
3501	Transaction Status Not Compatible with STL	Transaction status <> "STL Checked (No Discrepancy)" or "STL Checked (Discrepancy)."	2
3502	Transaction ID Presence	Transaction ID must exist in Transaction_Log table.	2

5.4.8 General Transaction Checks

The ITL performs this category of checks for all transaction messages involving existing unit blocks. Issuance transactions do not undergo General Transaction checks. See Figure 5.17, Validate Proposal, for further details on how transactions are checked.

Figure 5.9: General Transaction Checks

Response Code	Check Name	Check Description	Phase
4001	Units are Unavailable	All units identified in the transaction must be available for transaction (i.e., not involved in another ongoing transaction.)	1
4002	Units are Cancelled	Cancelled units cannot be the subject of further transactions.	1
4003	Units are Retired	Retired units cannot be the subject of further transactions.	1
4004	Units Have Inconsistencies	Units identified in the transaction cannot have existing reconciliation inconsistencies.	1
4005	Registry Holds Units	Units identified in transaction must be held by Initiating Registry.	1
4006	Current Commitment Period	Any unit in a proposed transaction must have an applicable Commitment Period identifier consistent with the current Commitment Period (including its true-up period) or for the next Commitment Period.	1
4007	Frozen Units	All units in the transaction must not be frozen due to a determination about a Project by the CDM Executive Board	1

5.4.9 Transaction-specific Checks

The ITL performs this category of checks on all Kyoto transactions for the specified transaction types. See Figure 5.17, Validate Proposal, for further details on how transactions are checked.

Figure 5.10: Transaction-specific Checks

Response Code	Check Name	Applies to Transaction Type	Check Description	Phase
5001	AAU Issuance Quantity	Issuance	The number of AAUs issued must not exceed the allowable quantity for the Commitment Period (provided by the C&A database).	2
5002	RMU Issuance Quantity	Issuance	The number of RMUs issued for each activity type must not exceed the allowable quantity (provided by the C&A database).	2
5003	RMU Issuance Timing	Issuance	RMUs cannot be issued before the end of the Commitment Period, unless annual Issuance option is selected.	3
5004	Consistency of Unit Type Issued for a Project	Issuance	Choice of issuing tCERs or lCERs must be consistent with previous issuances for the project.	2
5005	Issuing tCERs or lCERs for Project Type	Issuance	tCERs and lCERs must have a LULUCF activity identifier of 1.	2
5006	Expiry Date	Issuance	tCERs and lCERs must have an expiry date.	2
5007	tCERs Expiry Date	Issuance	Expiry date for tCERs must be end of second Commitment Period.	2
5008	lCERs Expiry Date	Issuance	Expiry date for lCERs must be consistent with end of the crediting period of the Project.	2
5009	CDM Registry Issuance of lCERs, tCERs and CERs	Issuance	Only CDM Registry may issue CERs, tCERs and lCERs.	2
5010	CDM Registry Issuance of Other Unit Types	Issuance	CDM Registry cannot issue unit types other than CERs, lCERs, and tCERs.	2
5012	CER Issuance Amount for Project	Issuance	CER Issuance for each CDM Project by CDM Registry must not exceed amount specified by CDM Executive Board.	2
5013	tCER Issuance Amount for Project	Issuance	tCER Issuance for each CDM Project by CDM Registry must not exceed amount specified by CDM Executive Board.	2

(cont.)

Figure 5.10: Transaction-specific Checks (cont.)

Response Code	Check Name	Applies to Transaction Type	Check Description	Phase
5014	lCER Issuance Amount for Project	Issuance	lCER Issuance for each CDM Project by CDM Registry must not exceed amount specified by CDM Executive Board.	2
5015	Issued Serial Numbers	Issuance	The serial numbers issued must be unique.	1
5030	Expired tCERs and ICERs Trade Restriction	Conversion, External, Retirement, Carry-over, Replacement	If a tCER or an lCER has expired, it may not be involved in any transaction, except an Internal transfer to a Type 3 Voluntary Cancellation account.	2
5031	Replacement Unit Trade Restriction	Conversion, External, Cancellation, Retirement, Carry-over, Replacement	A unit used to replace another unit may not be involved in any other transaction.	2
5032	Flagged Unit Check	Conversion, External, Retirement, Carry-over, Replacement, Expiry Date Change	If the replacement of lCERs from a CDM Project has been notified, lCERs from the Project may only be transferred to Replacement or Cancellation accounts.	2
5033	Unit Block Attributes	Retirement, Cancellation, Replacement of External Transfers	All attributes of a unit block must be consistent with ITL unit block attributes for Retirement, Cancellation, Replacement and External transactions.	2
5050	Units Available for Carry-over	Carry-over	Units identified in serial block must be specified as being available to be carried over.	3
5051	RMU Carry-over	Carry-over	RMUs may not be carried over.	3
5052	ERU Carry-over	Carry-over	ERUs converted from RMUs may not be carried over.	3
5053	lCER or tCER Carry-over	Carry-over	tCERs or lCERs may not be carried over.	3
5054	ERU Carry-over Limit	Carry-over	Total number of ERUs for Carry-over must not exceed limit.	3

(cont.)

Figure 5.10: Transaction-specific Checks (cont.)

Response Code	Check Name	Applies to Transaction Type	Check Description	Phase
5055	CER Carry-over Limit	Carry-over	Total quantity of CERs to be carried over must not exceed limit.	3
5056	Unit Block Attributes for Carry-over	Carry-over	All attributes except for Applicable Commitment Period of a unit block must be consistent with ITL unit block attributes for Carry-over transactions.	1
5101	Conversion Eligibility (Track 1)	Conversion	If the unit is a Track 1 ERU, then the Party must meet all 6 eligibility criteria.	2
5102	Conversion Eligibility (Track 2)	Conversion	If the unit is a Track 2 ERU, then the Party must meet eligibility criteria 1, 2 and 4.	2
5103	Conversion Unit Type	Conversion	Units for conversion must be AAUs or RMUs.	2
5104	Originating Registry for Conversion	Conversion	Units for conversion must be issued by Initiating Registry.	2
5105	Proposing Registry	Conversion	The Initiating Registry converting AAUs or RMUs must be a national registry.	2
5106	Unit Block Attributes for Conversion	Conversion	All attributes of a unit block, except for the Project ID and the Unit Type, must be consistent with ITL unit block attributes for Conversion transactions.	2
5107	ERU Conversion Quantity	Conversion	ERU Conversion for each Track 2 JI Project must not exceed quantity specified by the Article 6 Supervisory Committee.	2
5150	CER, tCER and lCER Retirement Limit	Retirement	Total quantity of tCERs and lCERs retired must not exceed limit.	2
5151	CER, tCER and lCER Retirement Eligibility	Retirement	If the unit is a CER, tCER, or lCER, the Party must meet all 6 eligibility criteria.	2
5200	External Transfers to CDM Registry	External	CDM Registry can only receive transfers to Cancellation accounts.	2

(cont.)

1044 **Figure 5.10: Transaction-specific Checks (cont.)**
1045

Response Code	Check Name	Applies to Transaction Type	Check Description	Phase
5202	Acquiring Registry Eligibility for External Transfers	External	The Acquiring Registry must meet all 6 eligibility criteria.	2
5211	Commitment Period Reserve Level	External	The proposed External transfer must not result in the total registry holdings of all units to fall below the CPR level for the Transferring Registry.	2
5301	Number of Replacement Units	Replacement	The number of units replaced must equal the number of replacing units.	2
5302	One-To-Many Replacement Units	Replacement	A transaction may not contain many-to-many relationships between replaced and replacing blocks.	2
5303	Location of Replaced tCERs	Replacement	Only tCERs that are held in a Retirement account or a Replacement account may be replaced.	2
5304	Replacement Registry	Replacement	The registry holding the units to be replaced and the replacing units must be the same.	2
5305	Replacement Account Type	Replacement	Only tCER Replacement accounts and lCER Replacement accounts may acquire units in a Replacement transaction.	2
5306	Location of Replaced lCERs	Replacement	lCERs that are held in Cancellation accounts may not be replaced.	2
5308	lCER Replacement Units (Due to Expiry)	Replacement	lCER Replacement accounts (unit expiry) may not acquire tCERs or lCERs.	2
5309	lCER Replacement Units (Due to lack of certification or Reversal in Storage)	Replacement	lCER Replacement accounts (lack of certification or Reversal in Storage) may not acquire tCERs and may not acquire lCERs with a Project Identifier other than that specified in the replacement notification.	2
5310	Multiple Replacement	Replacement	A unit may be replaced only once.	2
5311	tCER Replacement Units (Due to Expiry)	Replacement	tCER Replacement accounts (unit expiry) may not acquire lCERs.	2
5401	tCER or lCER Cancellation Purposes	Cancellation	tCERs and lCERs may not be transferred to type 1, 2 or 4 Cancellation accounts.	2

(cont.)

1046

Figure 5.10: Transaction-specific Checks (cont.)

Response Code	Check Name	Applies to Transaction Type	Check Description	Phase
5450	Units for Expiry Date Change	Expiry Date Change	The units for Expiry Date Change must be tCERs or lCERs.	2
5451	New tCER Expiry Date	Expiry Date Change	The new tCER expiry date must be consistent with the end date of the second Commitment Period.	2
5452	New lCER Expiry Date	Expiry Date Change	The new lCER expiry date must be consistent with the end date of the renewed crediting period for the Project.	2
5453	Unit Block Attributes	Expiry Date Change	All attributes of a unit block except for the Expiry Date must be consistent with ITL unit block attributes for Expiry Date Change transactions.	2

5.4.10 Registry Messages

These responses may be returned by an Acquiring Registry in response to a proposed external transaction, if the overall status of the transaction from the Acquiring Registry is "Rejected." If the transaction status is "Accepted," no response codes will be sent with the message from the Acquiring Registry.

Figure 5.11: Registry Messages

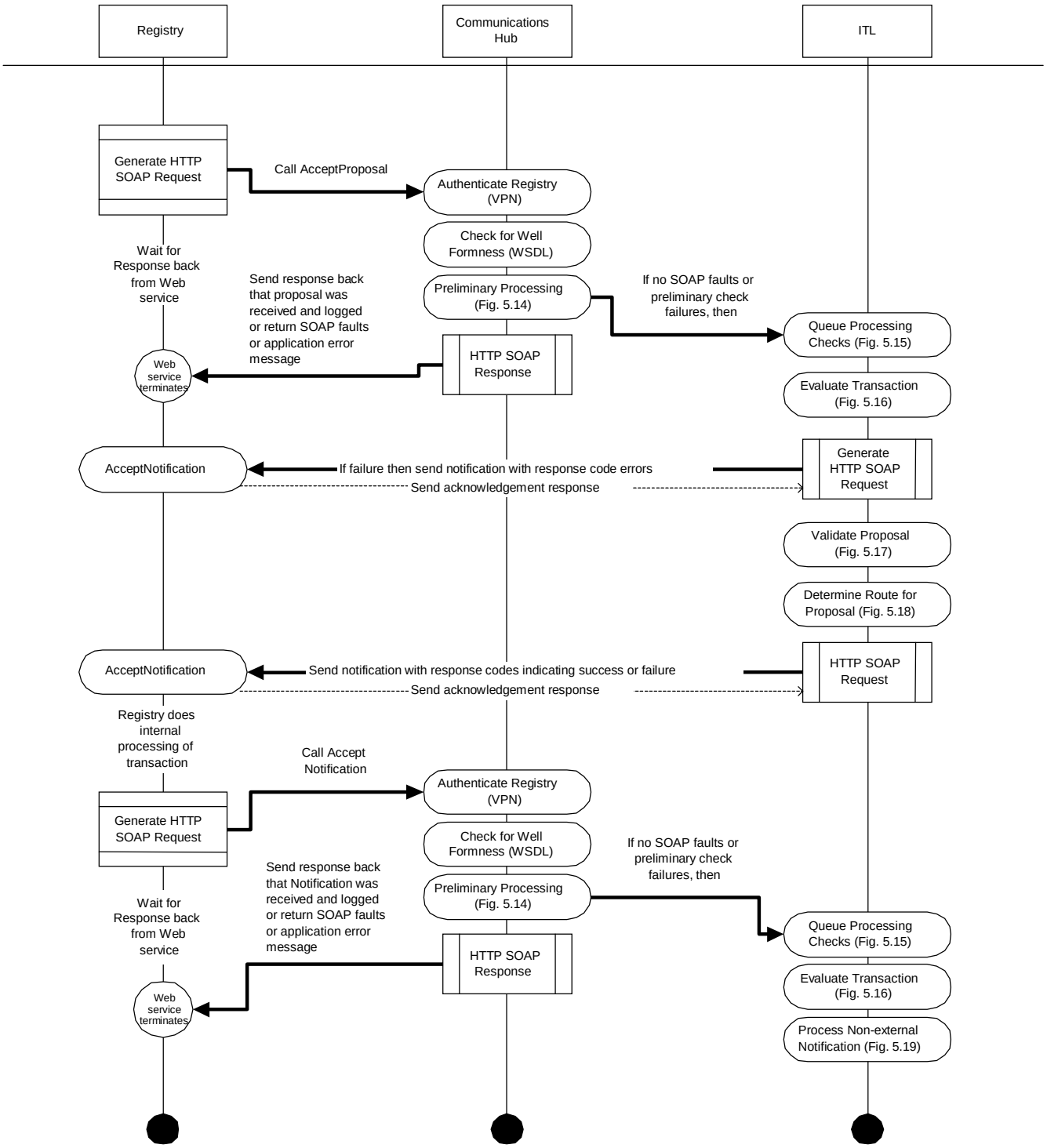
Response Code	Response Description
5902	Acquiring account does not exist.
5903	Acquiring account is not eligible to receive units.
5904	Transaction inconsistent with Party policy.
5905	Transaction rejected by account holder.
5906	Account has been closed.

5.5 Activity Diagrams

The following activity diagrams provide an overview of how the HTTP SOAP request is processed for a transaction. The first diagram describes transactions between a single registry and the ITL. This diagram is applicable to the Issuance, Conversion, Cancellation, Retirement, Carry-over, Replacement, and Expiry Date Change. The second diagram describes an External transaction between two registries and the ITL. These activity diagrams do not show interaction between the ITL and an STL.

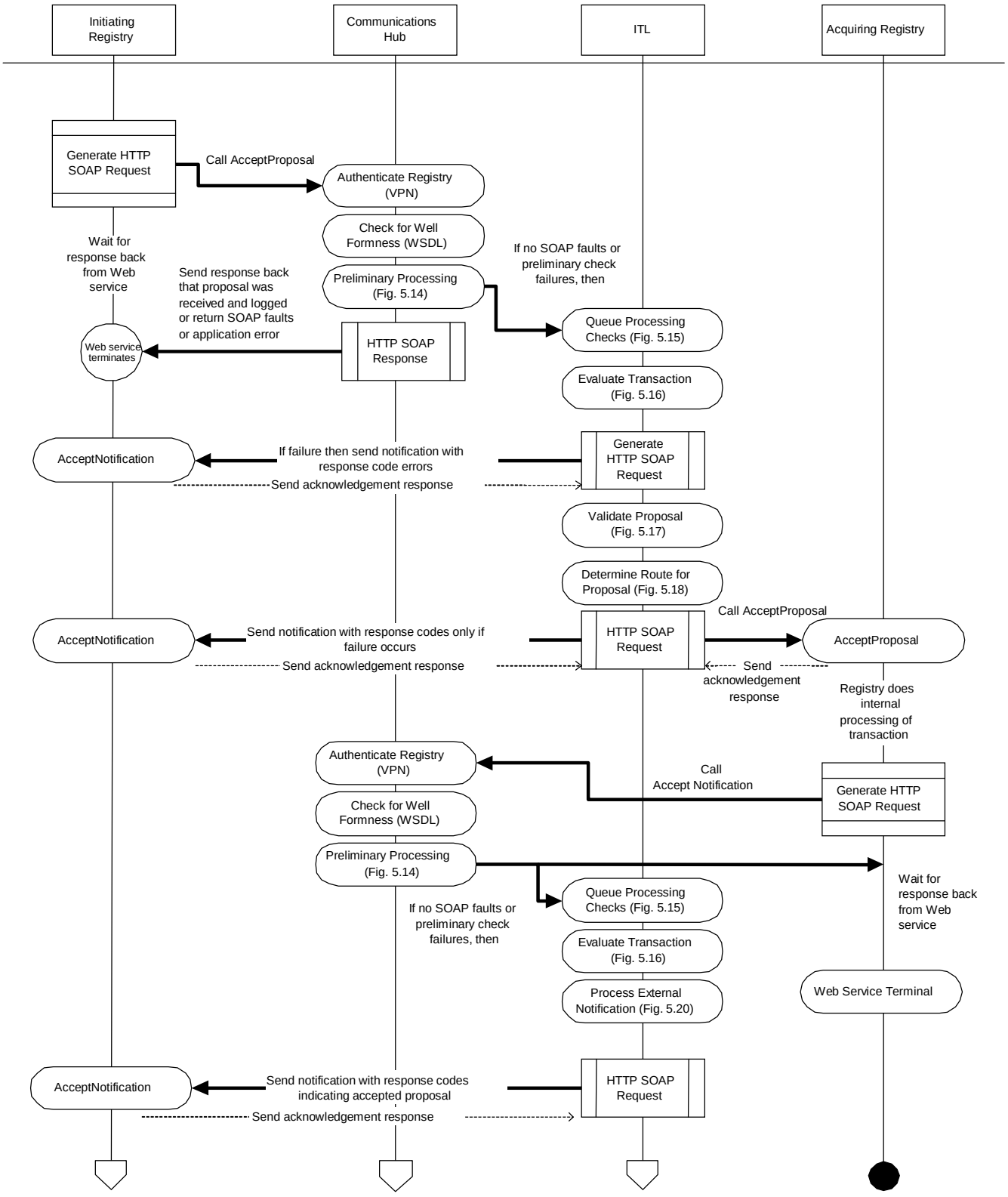
5.5.1 Basic Transaction Activity Diagram

Figure 5.12: Simple (Non-STL) Transaction Process



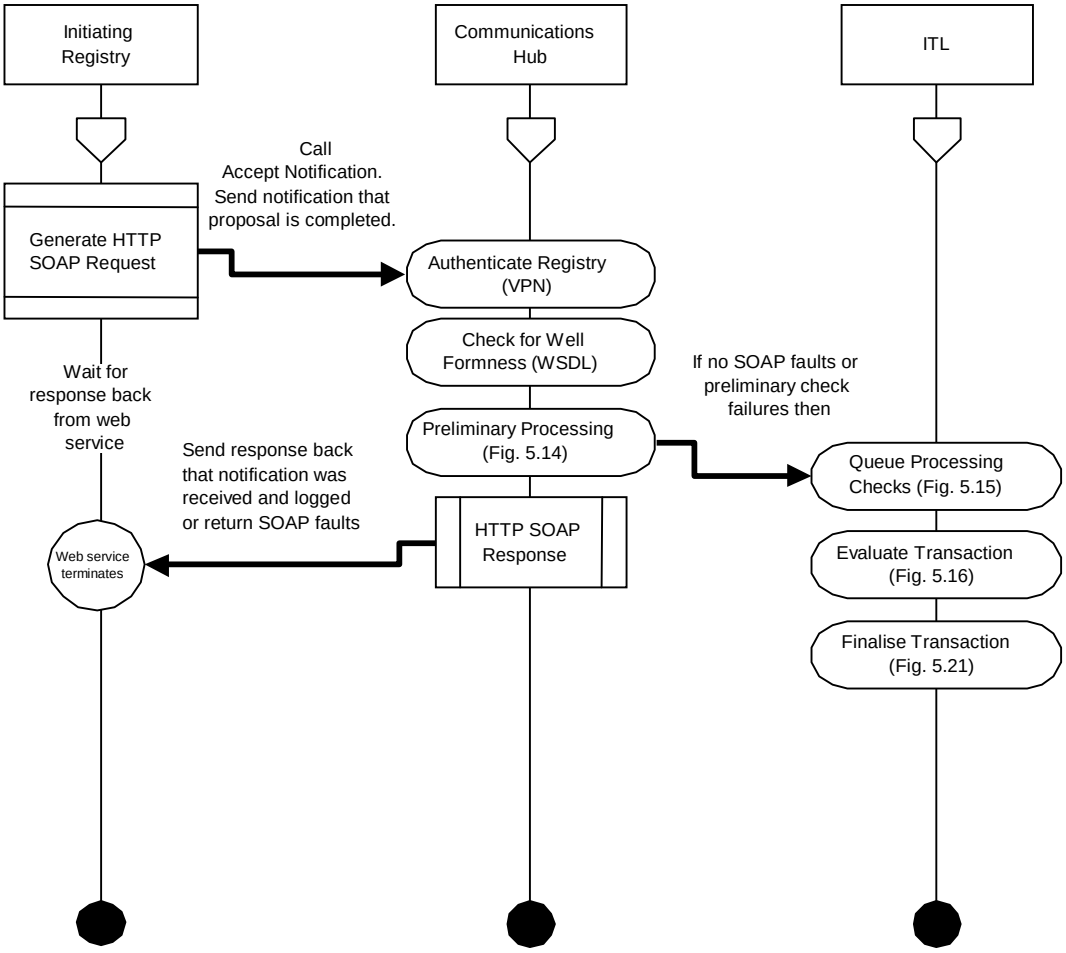
5.5.2 External Transfer Activity Diagram

Figure 5.13: External (Non-STL) Transaction Process



(cont.)

1087 **Figure 5.13 External (Non-STL) Transaction Process (cont.)**
1088



1089

5.6 Transaction Flow Diagrams

For all types of transactions, the processing of a transaction proceeds through a standard sequence of steps. The following process flow diagrams supplement the activity diagrams in Section 5.5. Boxes or diamonds that contain a ":" prefix indicate a function which can be found in detail in Annex E. Boxes with the "ref" symbol in the corner indicate that supplemental sequence diagrams are provided.

Figure 5.14: Preliminary Processing

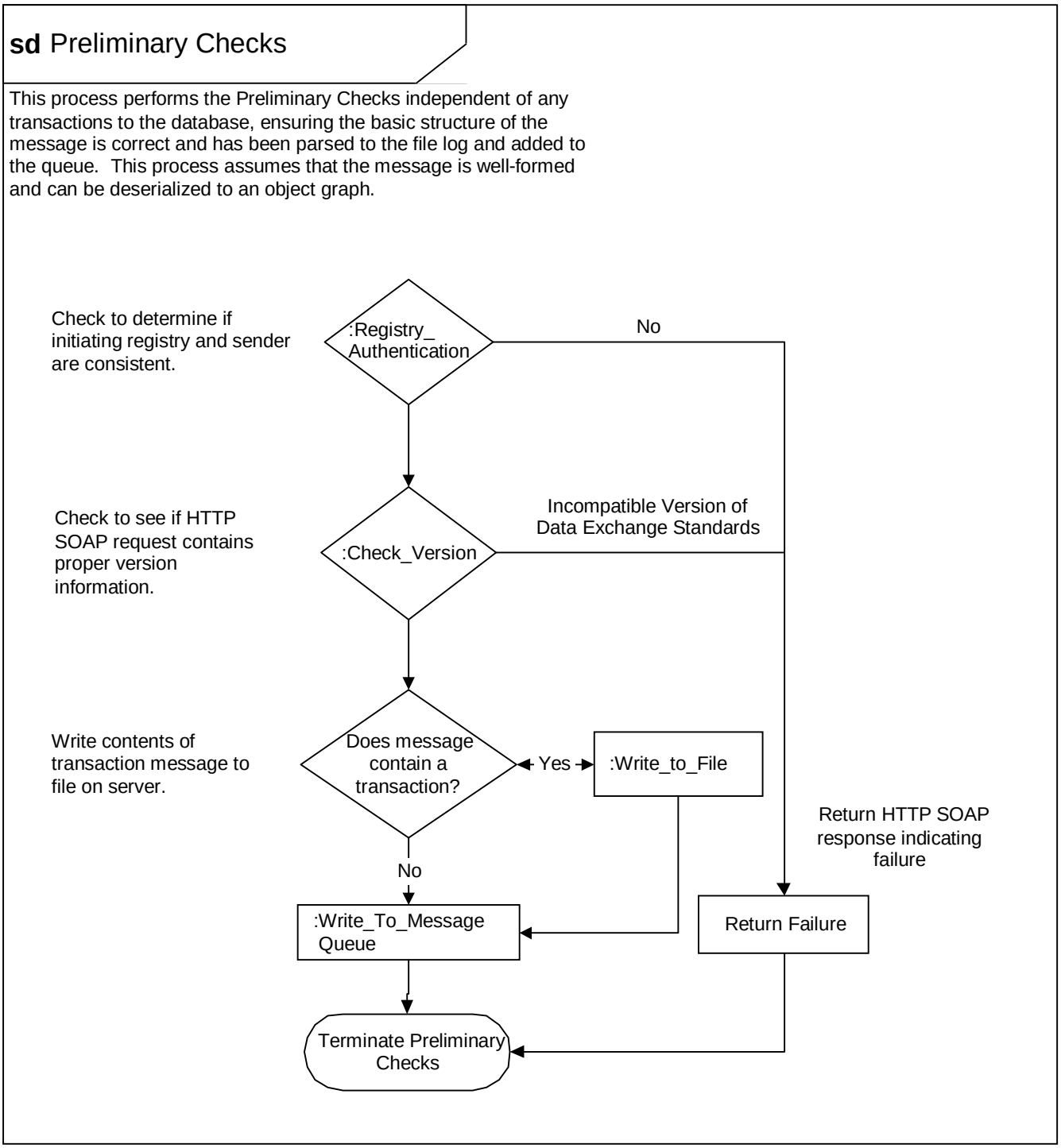


Figure 5.15: Queue Processing Checks

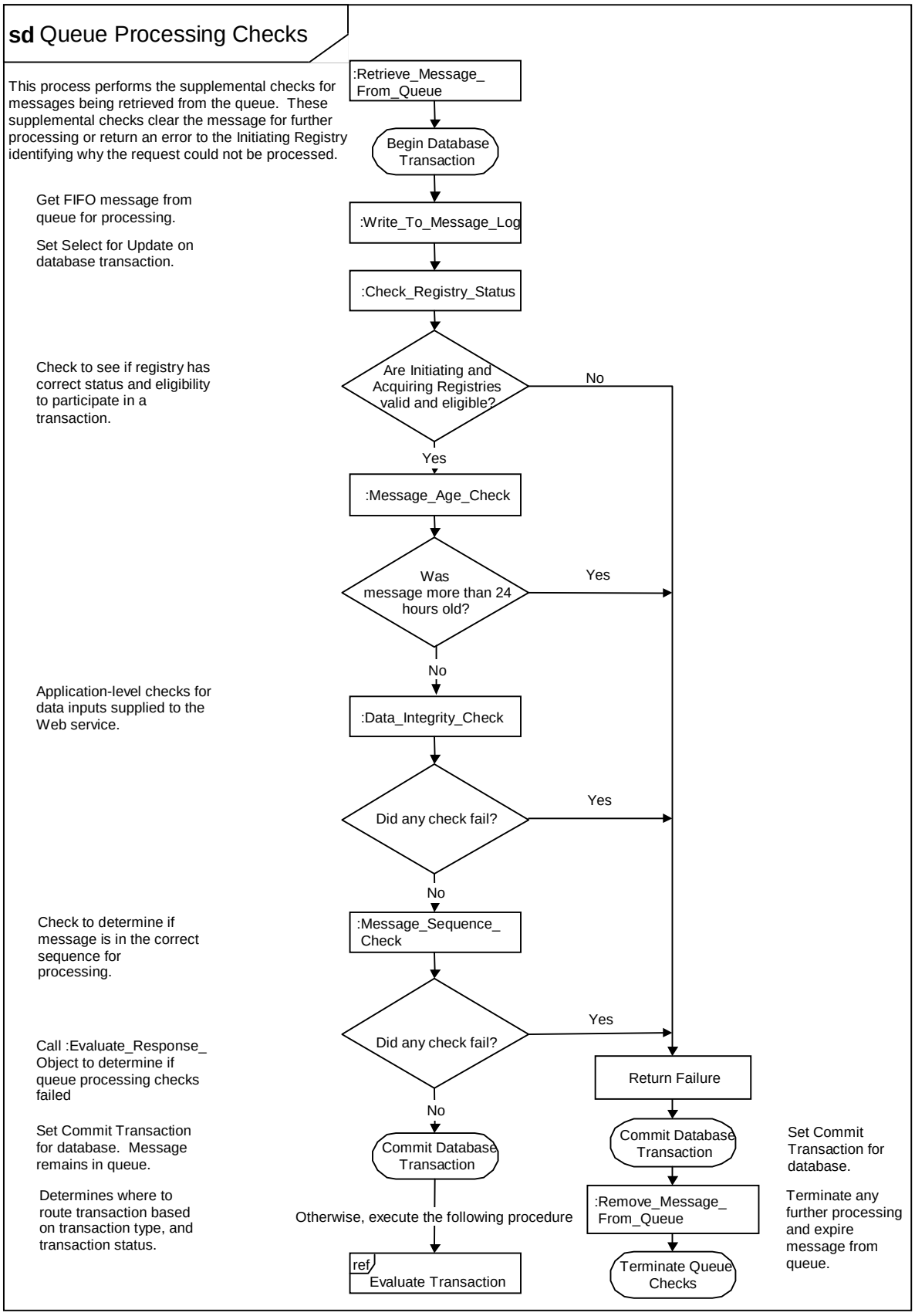
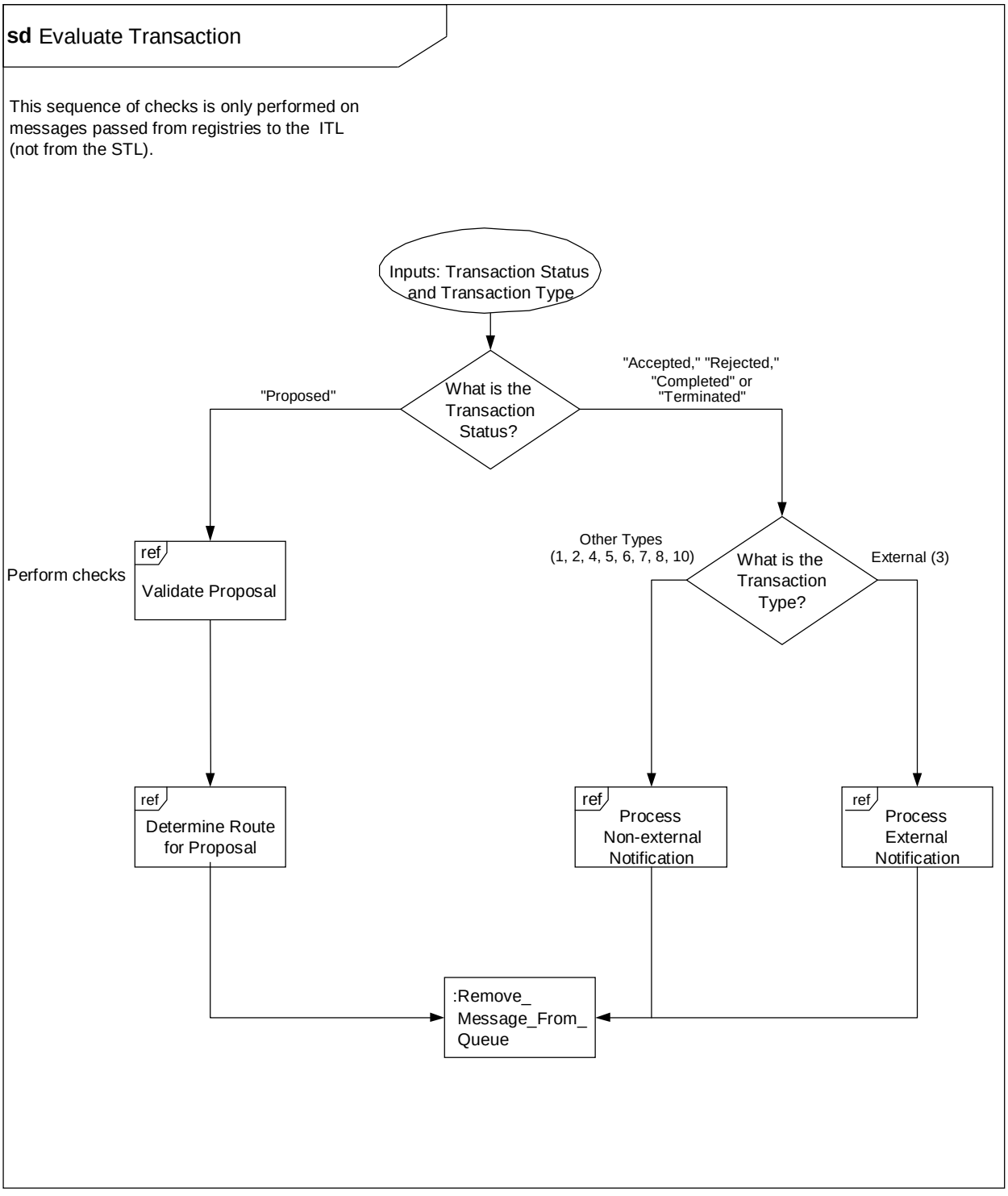
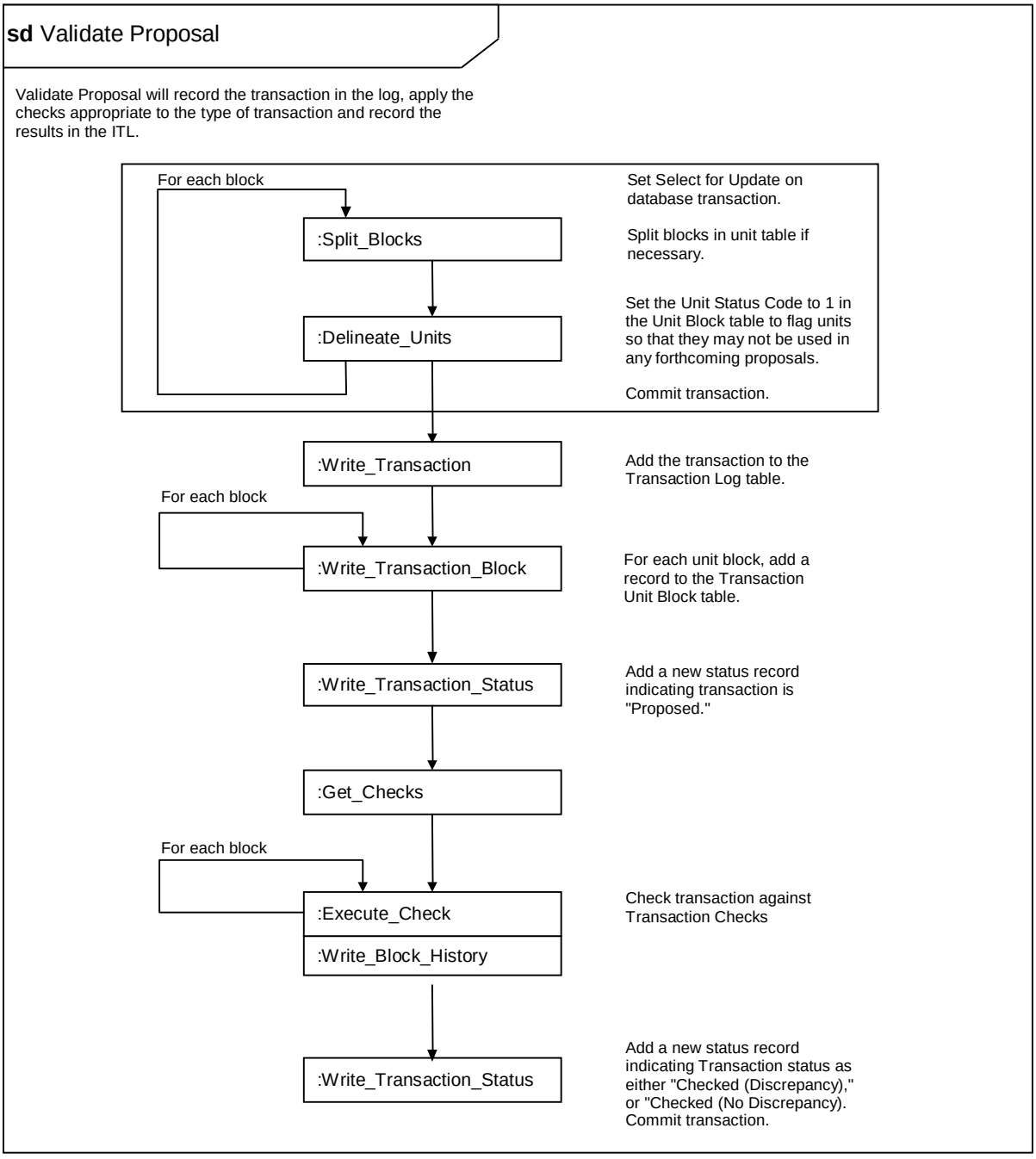


Figure 5.16: Evaluate Transaction



1108 **Figure 5.17: Validate Proposal**

1109



1110

Figure 5.18: Determine Route for Proposal

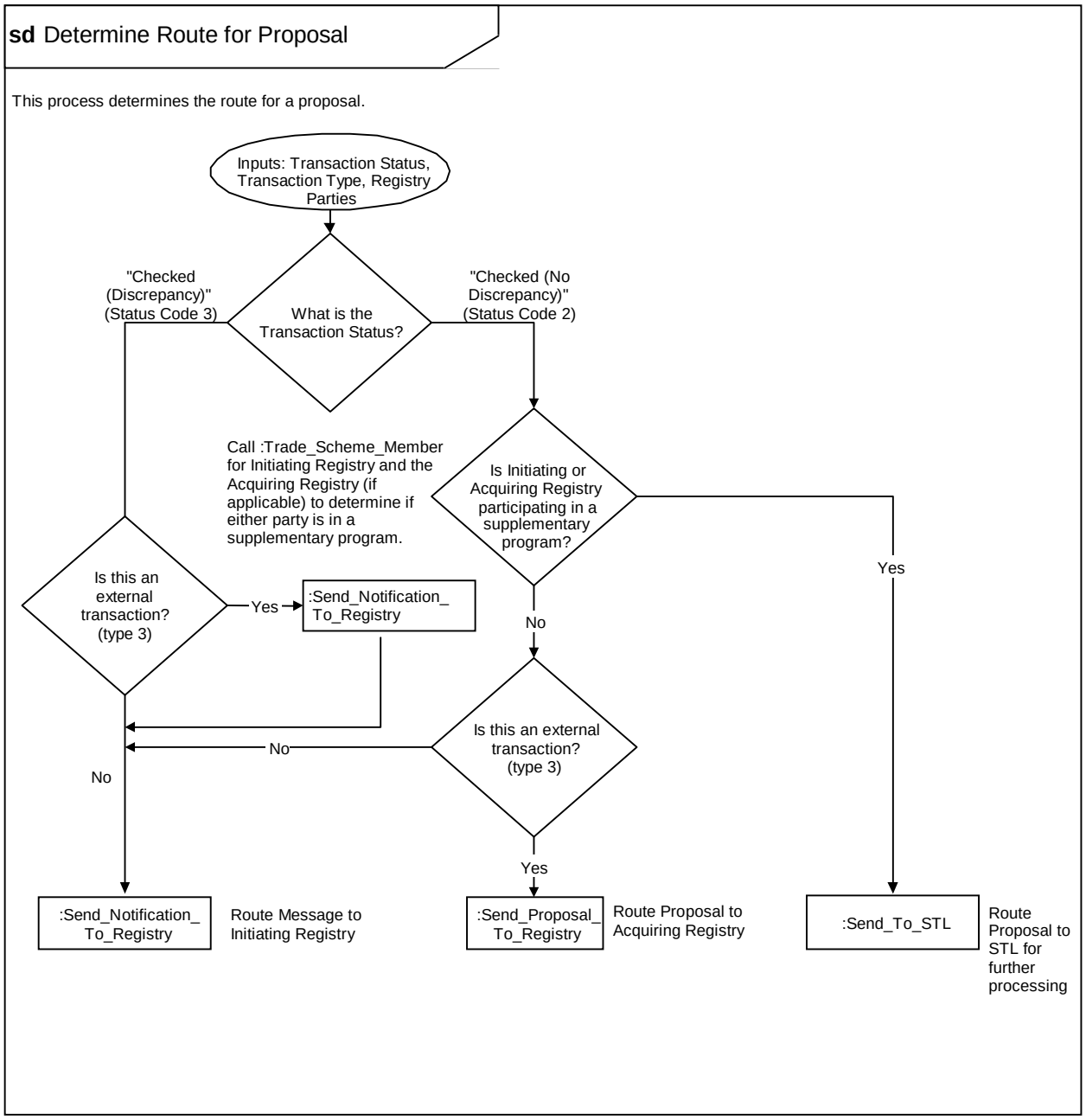


Figure 5.19: Process Non-external Transaction Notification

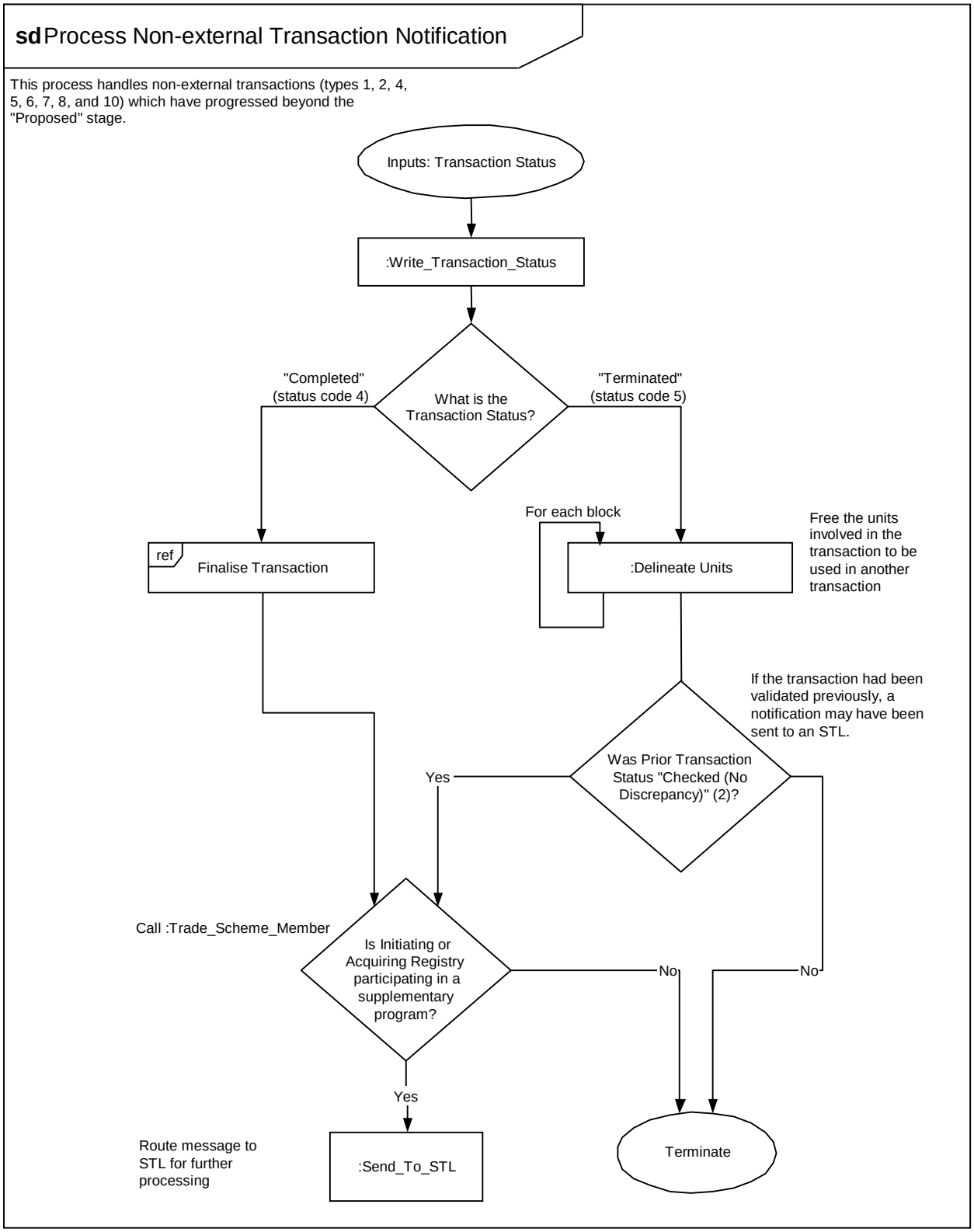


Figure 5.20: Process External Transaction Notification

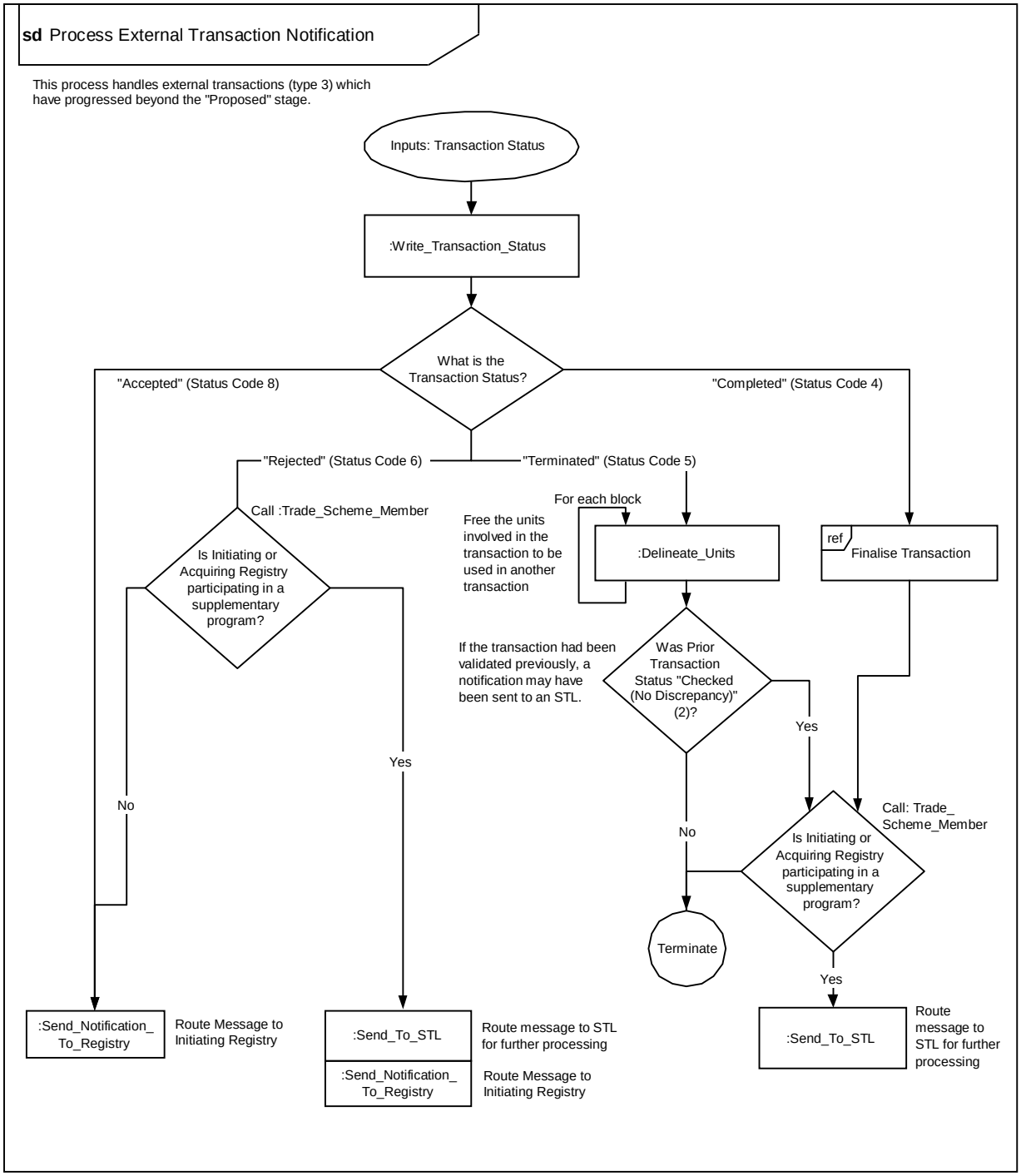


Figure 5.21: Finalise Transaction

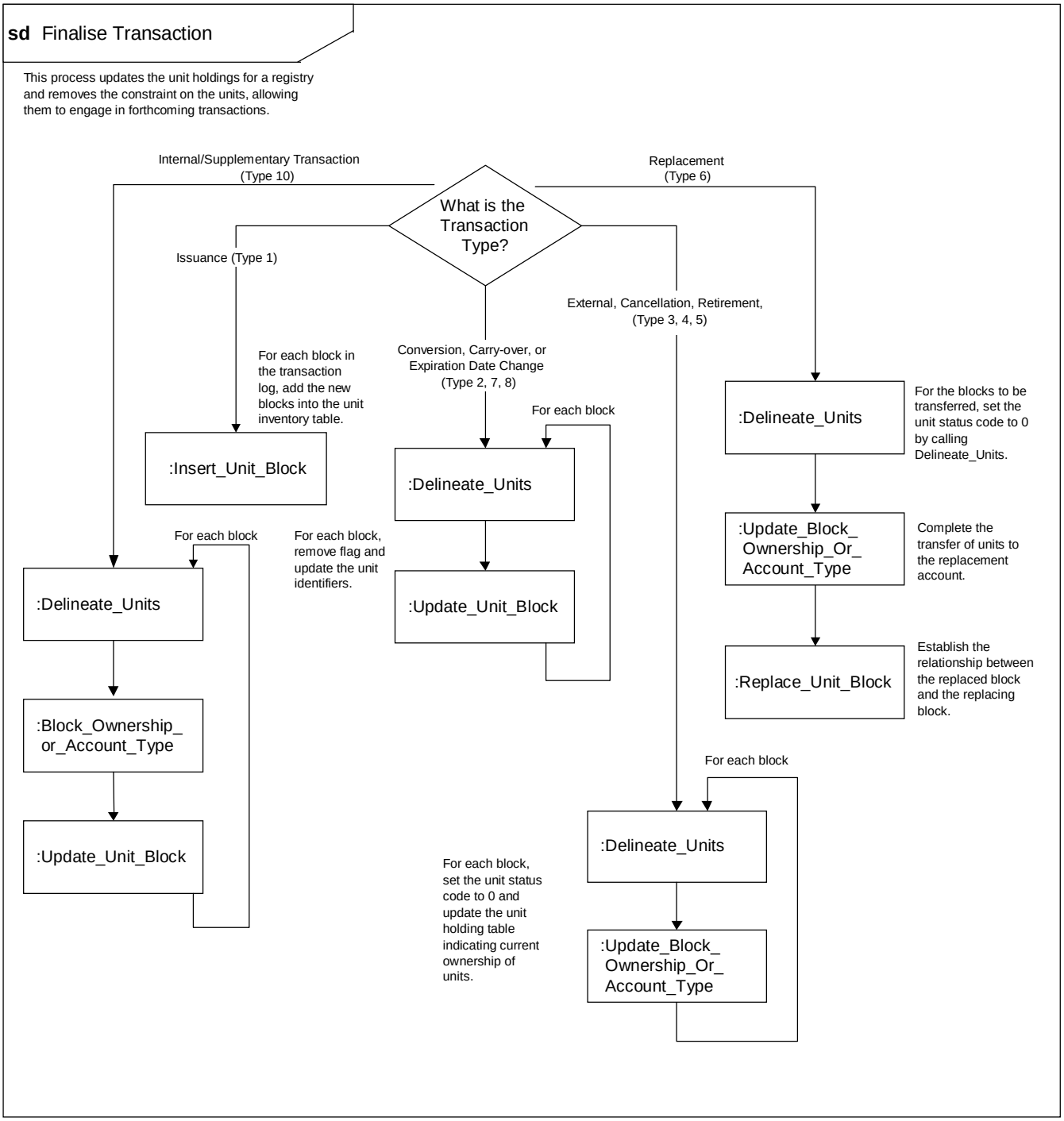
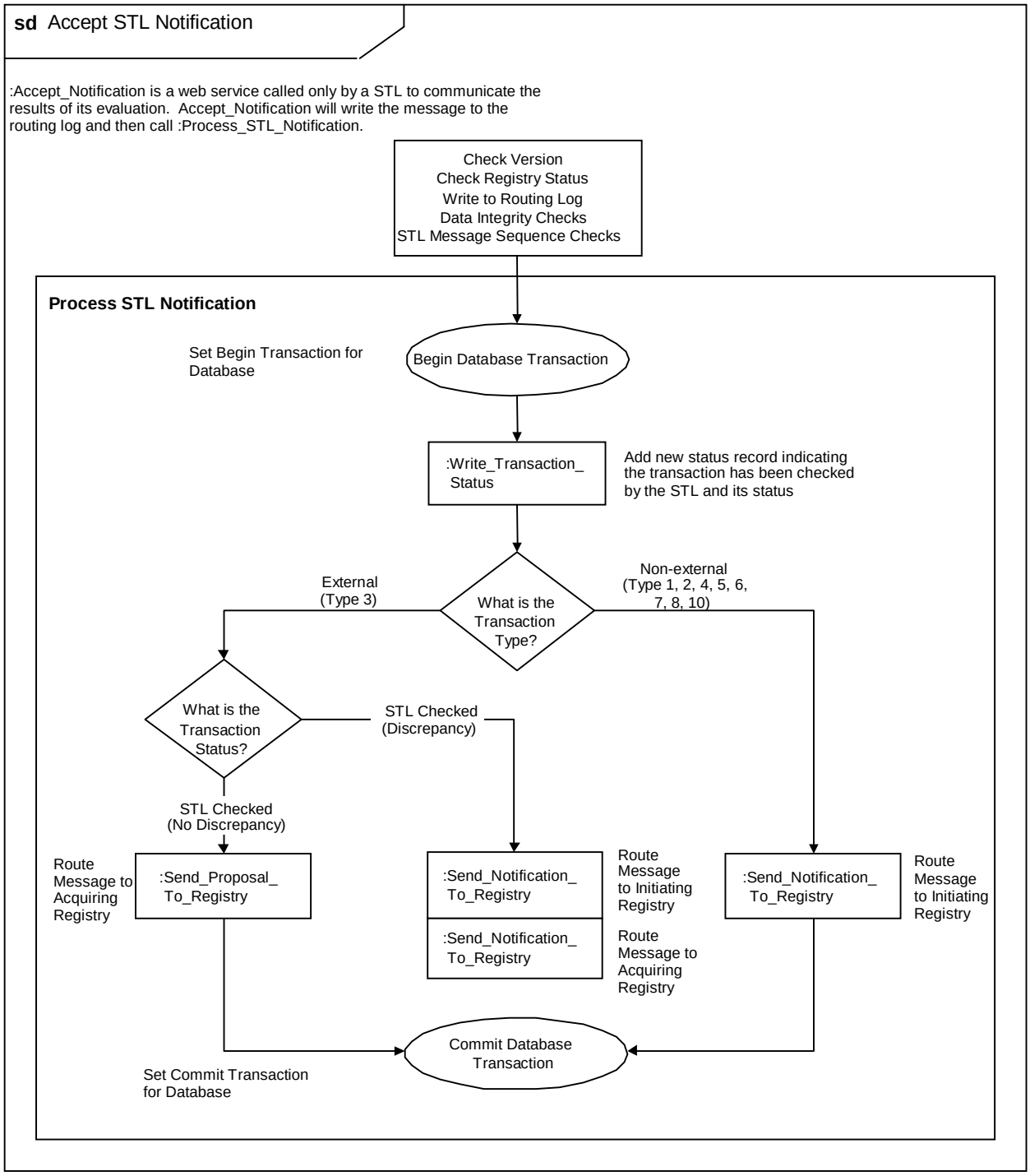


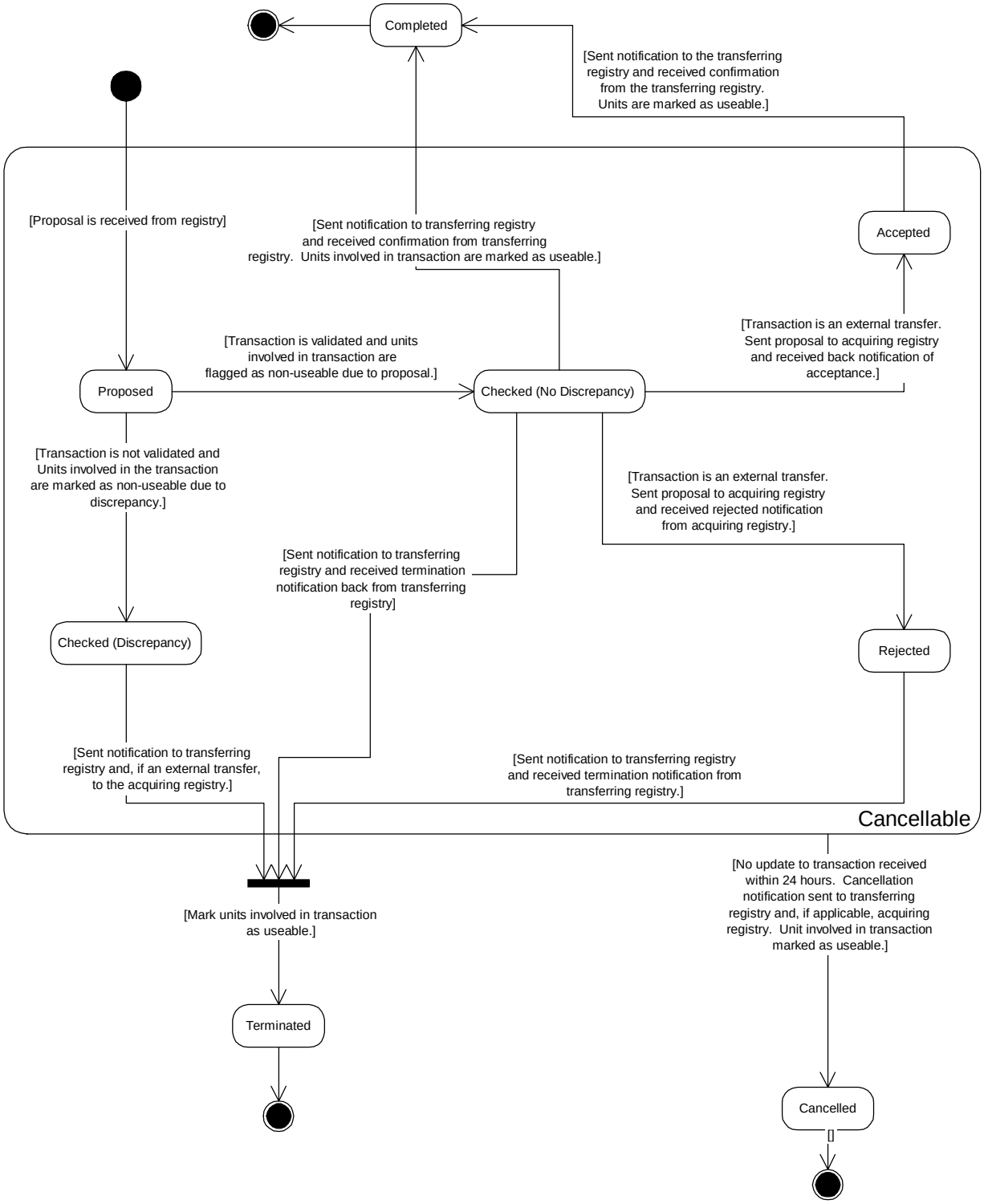
Figure 5.22: Accept STL Notification



5.7 Transaction State Diagram

The following state diagram describes the possible state change of a transaction on the ITL. The state of a transaction changes as checks or actions process the transaction through the sequence of steps described in Figures 5.12 and 5.13.

1135 **Figure 5.23: Transaction State Diagram**
1136



1137
1138

6. Reconciliation Process

The ITL reconciles data on unit holdings in registries periodically on the basis of a data snapshot at a specified future time. A reconciliation action is completed when no inconsistencies are discovered or when any discovered inconsistencies have been resolved through a manual intervention process.

For supplementary program registries, the STL may reconcile data on unit holdings in accounts. For such STLs, a reconciliation action is completed when no inconsistencies are discovered or when any discovered inconsistencies have been resolved through a manual intervention.

An STL may request that the ITL initiate a reconciliation action. A reconciliation action may be initiated for any stage; however, only a successful unit count by account type (ITL) or by account (STL) will result in the release of inconsistent blocks.

6.1 Reconciliation Snapshot Data

The reconciliation of data in a registry with the ITL and any STL occurs at a particular point in time. To simplify this process, it is recommended that the ITL, registry and the STL (if applicable), record a snapshot of the Unit Block table to perform the analyses and comparisons for each reconciliation action. The time of the action will either be negotiated as a regular, scheduled action at a time convenient to the registry, or will be initiated by the ITL or STL in response to the particular situation or corrected problem from a previous reconciliation.

The structure of the snapshot for a registry will contain the following information:

- Reconciliation ID
- Snapshot DateTime
- Account Type
- Account Number
- Unit Type
- Originating Registry
- Start Block
- End Block

The structure of the snapshot for the ITL will be:

- Reconciliation ID
- Snapshot DateTime
- Holding Registry
- Account Type
- Unit Type
- Originating Registry
- Start Block
- End Block

The structure of the snapshot for the STL checking account/unit holdings is recommended to be:

- Reconciliation ID
- Snapshot DateTime

- Holding Registry
- Account ID
- Unit Type
- Originating Registry
- Start Block
- End Block

Within the ITL, the snapshot will be stored in an Oracle table. If necessary to ensure performance, this table would be located on a separate instance of Oracle and on a separate server. All data would be stored until a successful reconciliation is completed, or for a defined period of time following the reconciliation snapshot date, whichever occurs later.

6.2 Reconciliation Message Checks

When the ITL receives reconciliation messages from registries in response to its reconciliation request, the following types of checks are performed on the messages. These are followed by the reconciliation evaluation itself, which is detailed in the Reconciliation Process flow diagrams and functions described in Annex E.

Figure 6.1: Reconciliation Check Categories

Category	Response Code Range	Category Description	Action
Version and Authentication	1000 - 1299	Checks to validate version of DES.	Message returned with response codes. Message not placed into message queue.
Message Validity	1300 - 1399	Checks for message validity.	Message returned with response codes. Message not placed into message queue.
Registry Validation	1500 - 1599	Checks to validate status of registry.	Message returned with response codes. Message not logged in Reconciliation_Log table.
Reconciliation Data Integrity	6000 - 6299	Basic checks of data content including numeric ranges and validity of codes.	Message returned with response codes. Message not logged in Reconciliation_Log table.
Reconciliation Message Sequence	6300 - 6399	Checks to validate message order and reconciliation status.	Message returned with response codes. Message not logged in Reconciliation_Log table.
Other Reconciliation Checks	6400 - 6500	Basic reconciliation checks.	Message returned with response codes and transaction status. Message logged in Reconciliation_Log table.

6.2.1 Version and Authentication Checks for Reconciliation

Preliminary checks, including version and authentication checks, are performed upon receipt of the HTTP SOAP request from a registry and do not involve any interaction with the ITL database. If these checks are passed, the message is placed in the message queue for processing. Failures

due to authentication and poorly formed XML content are returned as HTTP SOAP fault errors. Failures due to any reconciliation check are returned in the ResponseObject in an HTTP SOAP response. See Figure 5.3.

6.2.2 Message Validity Checks for Reconciliation

As with transactions, all messages from the queue are checked to determine if they are more than 24 hours old. See Figure 5.3.

6.2.3 Registry Validation Checks for Reconciliation

When the message has been retrieved from the message queue and recorded in the message log, checks are performed to determine if the registries involved in the transaction are identifiable and eligible to participate. See Figure 5.5.

Figure 6.2: Additional Registry Checks for Reconciliation

Response Code	Check Name	Check Description
1510	Registry Available for Reconciliation Action	Initiating Registry status code must be equal to zero or 1.

6.2.4 Data Integrity Checks for Reconciliation

This category of checks is performed by the Reconciliation_Data_Integrity_Check function to identify incoming messages containing data that fail basic data integrity checks. If any data in a message fail these checks, the message is returned to the sender with an appropriate response code. The message is not logged in the Reconciliation_Log table and is not processed further. Data integrity checks are critical checks in that if they result in failure, no further checks should be processed.

Note that as part of reconciliation, transactions and unit blocks are passed into the ITL, but those items are minimally checked by the data integrity checks. If there is a problem with the format of a transaction or a unit block, the reconciliation process will identify and log those items as the source of an inconsistency.

Figure 6.3: Summary of Reconciliation Data Integrity Checks

Response Code	Check Name	Check Description
6202	Reconciliation Mask	Reconciliation ID must be alpha (3) + numeric (15).
6203	Reconciliation Status Validity	Reconciliation Status must be valid.
6204	Reconciliation Snapshot DateTime	Reconciliation Snapshot DateTime must be between 10/1/04 and current date + 30 days.

(cont.)

Figure 6.3: Summary of Reconciliation Data Integrity Checks (cont.)

Response Code	Check Name	Check Description
6205	Account Type Validity	Account Type must be valid.
6206	Unit Type Validity	Unit Type must be valid.
6207	Supplementary Unit Type Validity	Supplementary Unit Type must be valid.
6208	Reconciliation Phase	Reconciliation Phase must be valid.

6.2.5 Message Sequence Checks for Reconciliation Messages Received from Registries

After the data in the message have been checked, the ITL performs checks to ensure that the message received from a registry has been submitted in the proper sequence.

Figure 6.4: Sequence Checks for Registry Messages

Response Code	Check Name	Check Description
6301	Reconciliation ID Does Not Exist	Reconciliation ID must exist in the Reconciliation_Log table.
6302	Reconciliation Status Not Valid	Out of sequence reconciliation status sent by registry is invalid.
6303	Reconciliation Status Out of Sequence	Incoming reconciliation status should be the same as the reconciliation sequence recorded at the ITL.
6304	Reconciliation Snapshot DateTime	The reconciliation snapshot DateTime must be consistent with the DateTime of the reconciliation DateTime.

6.2.6 Message Sequence Checks for Reconciliation Messages Received from STL

After the data in the message have been checked, the ITL performs checks to ensure that the message received from an STL has been submitted in the proper sequence.

Figure 6.5: Sequence Checks for STL Messages

Response Code	Check Name	Check Description
6311	Reconciliation ID does not exist.	Reconciliation ID must exist in the Reconciliation_Log table unless the incoming reconciliation status is "Initiated."
6312	Reconciliation Status Not Valid	Reconciliation status sent by the STL must a valid STL status.
6313	Reconciliation Status of "STL Totals Inconsistent" is Out of Sequence	If the incoming reconciliation status is "STL Totals Inconsistent," the previously recorded status at the ITL must be "Validated."
6314	Reconciliation Status of "STL Unit Blocks Inconsistent" Out of Sequence	If the Reconciliation Phase = 1 and the incoming reconciliation status is "STL Unit Blocks Inconsistent," the previously recorded status at the ITL must be "STL Totals Inconsistent."
6315	Reconciliation Message Out of Sequence	If incoming reconciliation status is "STL Validated," prior reconciliation status must be "Validated," "STL Totals Inconsistent," or "STL Unit Blocks Inconsistent."
6316	Reconciliation Message Out of Sequence	If incoming reconciliation status is "STL Complete with Manual Intervention," prior reconciliation status must be "STL Totals Inconsistent," or "STL Unit Blocks Inconsistent."

6.2.7 Other Reconciliation Responses

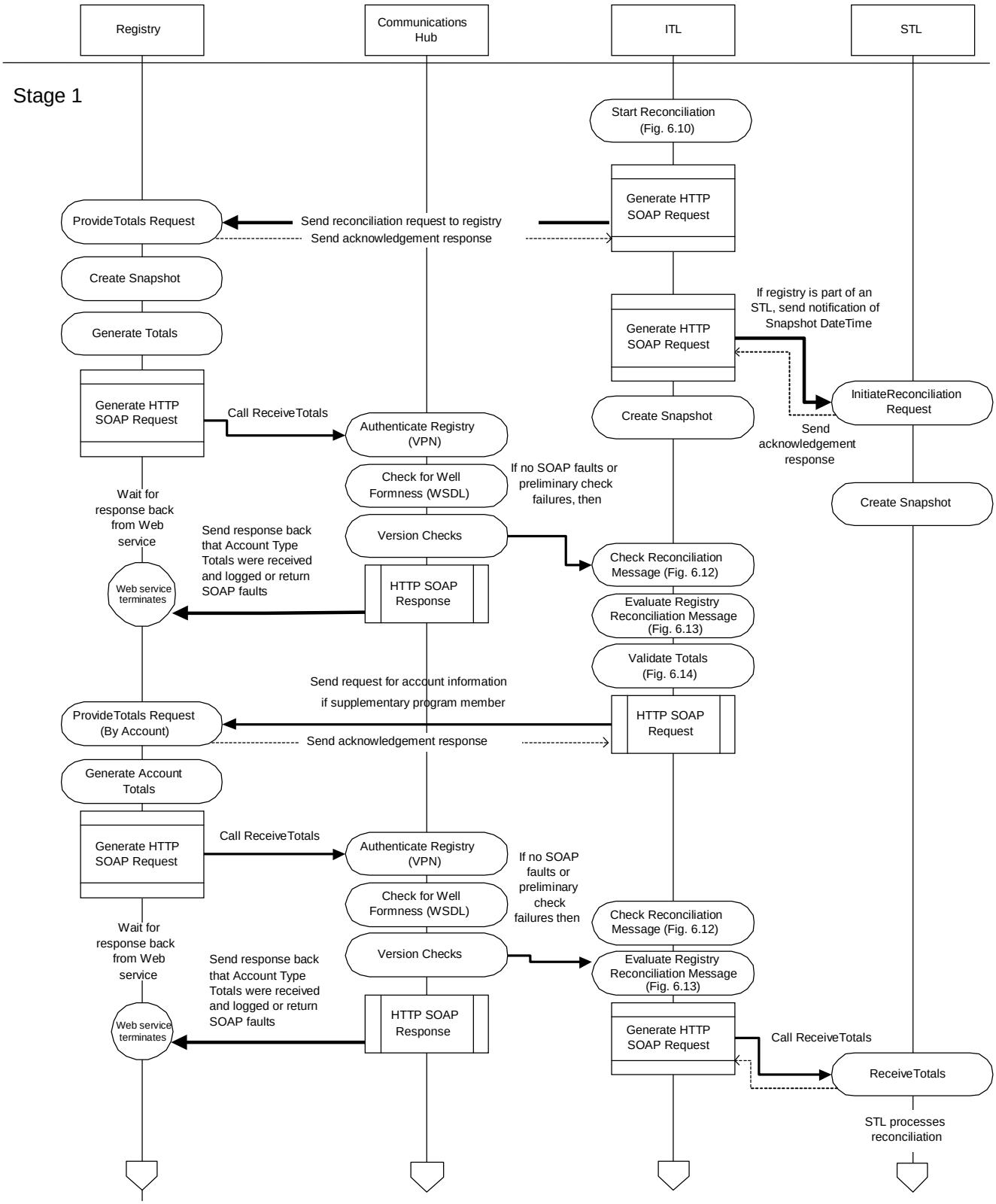
The following response codes provide information about the results of the reconciliation analyses to the registry or STL.

Figure 6.6: Other Reconciliation Checks and Messages

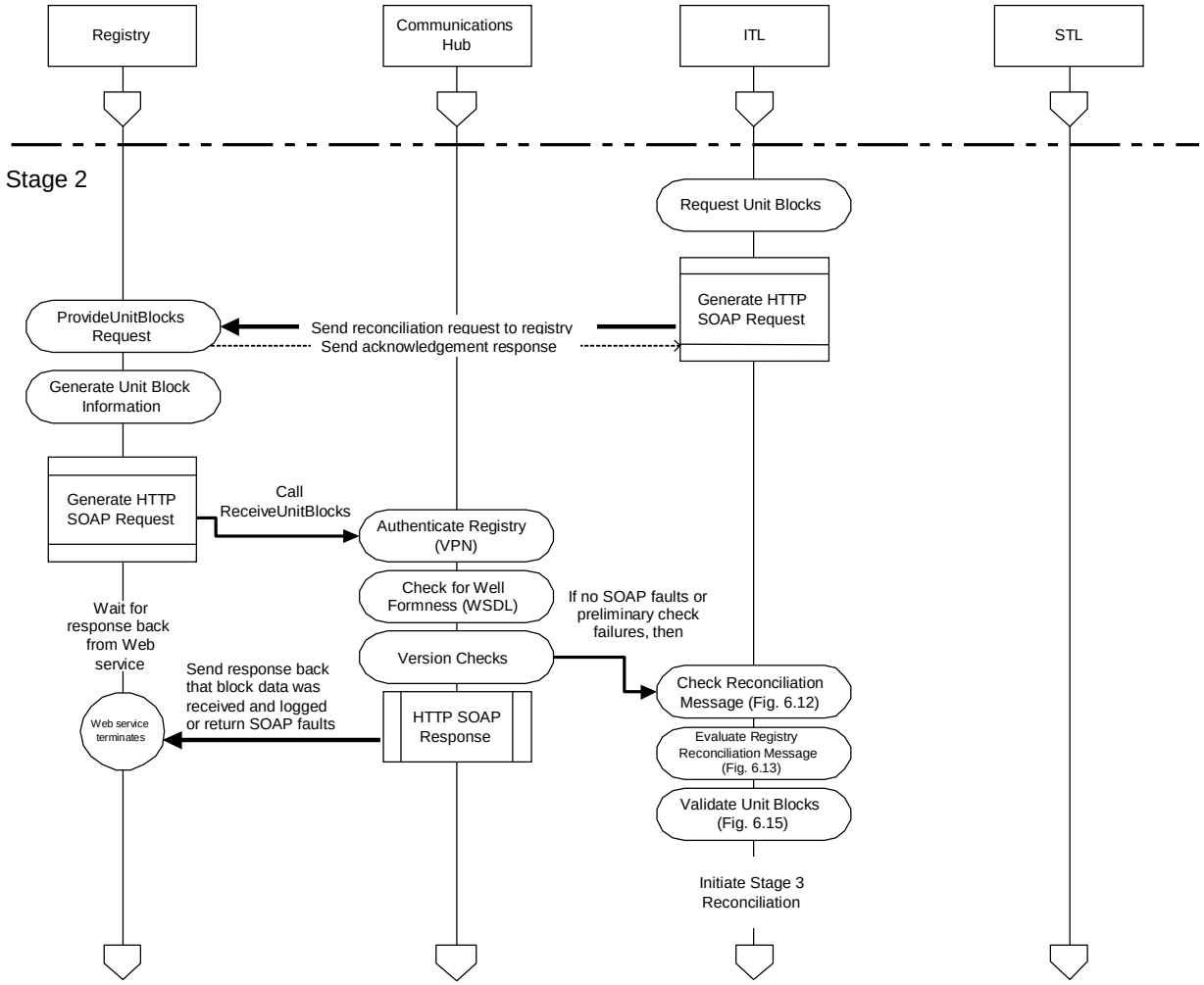
Response Code	Check Name	Check Description
6010	Account Type/Unit Type Totals	There is an inconsistency between the registry and the ITL in the unit block totals for the following totals by account type and unit type.
6420	Account Type/Unit Type Unit Blocks	The registry and ITL unit blocks for each specified account type and unit type must be consistent.
6430	Account Type/Unit Type Unit Blocks Unexpected Consistency	If the totals have failed and the unit blocks pass, an inconsistency should be found.
6440	Snapshot DateTime Validity	The DateTime for reconciliation proposed by the STL must be in the future.
6450	Ongoing Reconciliation	A reconciliation action cannot be initiated at this registry because there is already an ongoing reconciliation action.
6600	Successful Reconciliation of Totals	A reconciliation has been completed with a successful reconciliation of unit totals.

6.3 Activity Diagrams

Figure 6.7: Reconciliation Process Flow Stage 1 - Validate Account Totals

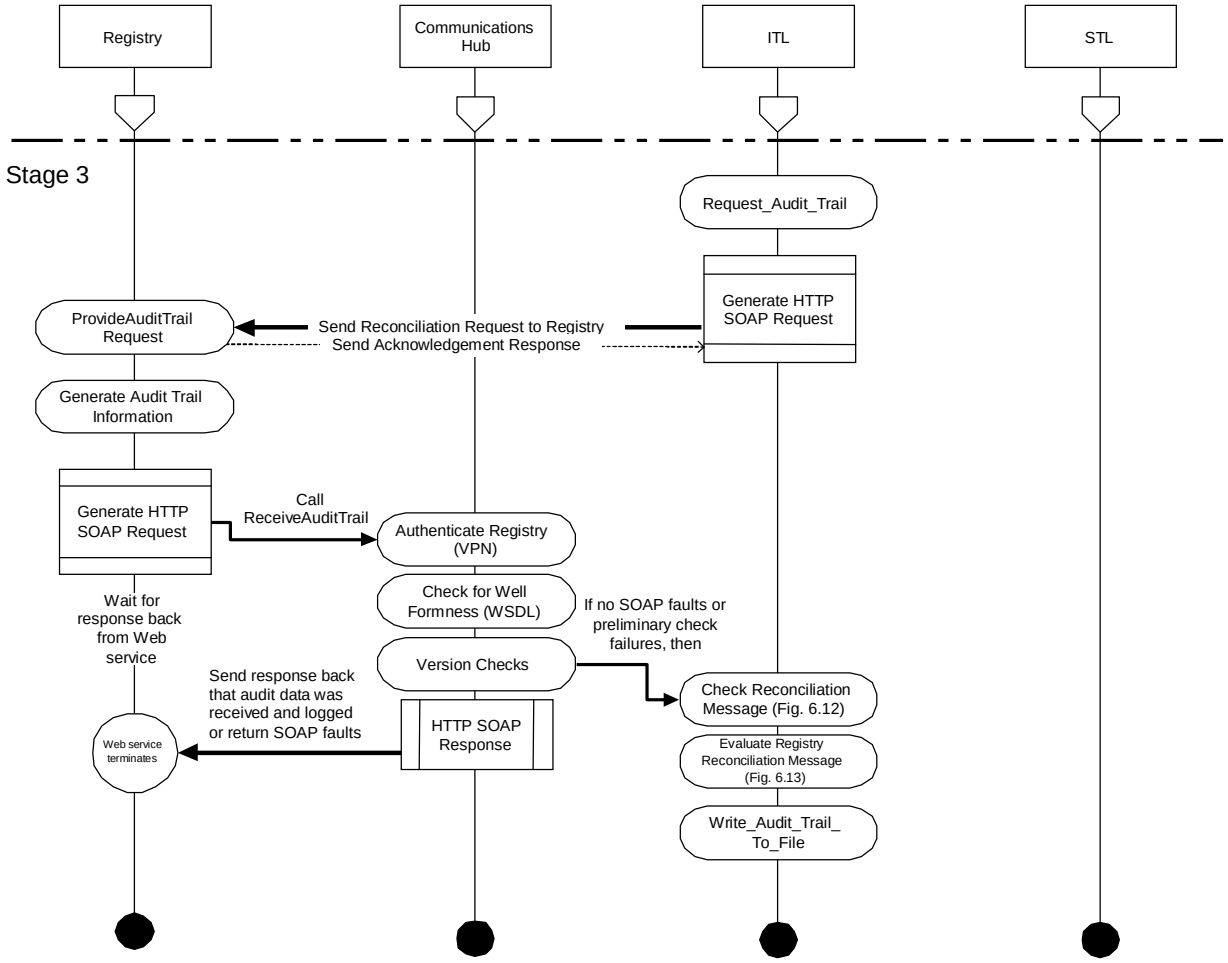


1288 **Figure 6.8: Reconciliation Process Flow Stage 2 - Validate Unit Blocks**
1289



1290

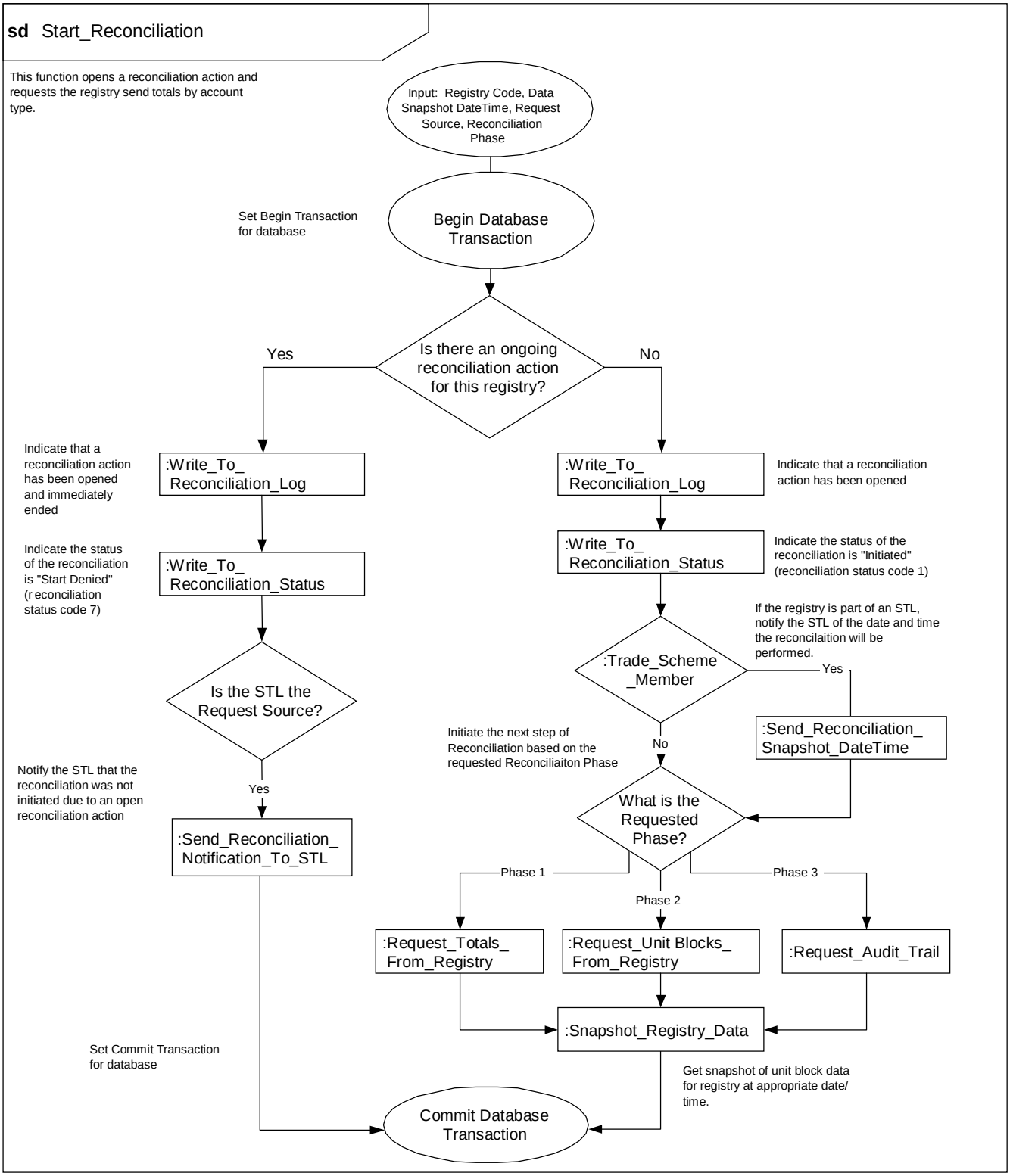
1291 **Figure 6.9: Reconciliation Process Flow Stage 3 - Review Audit Logs**
1292



1293
1294

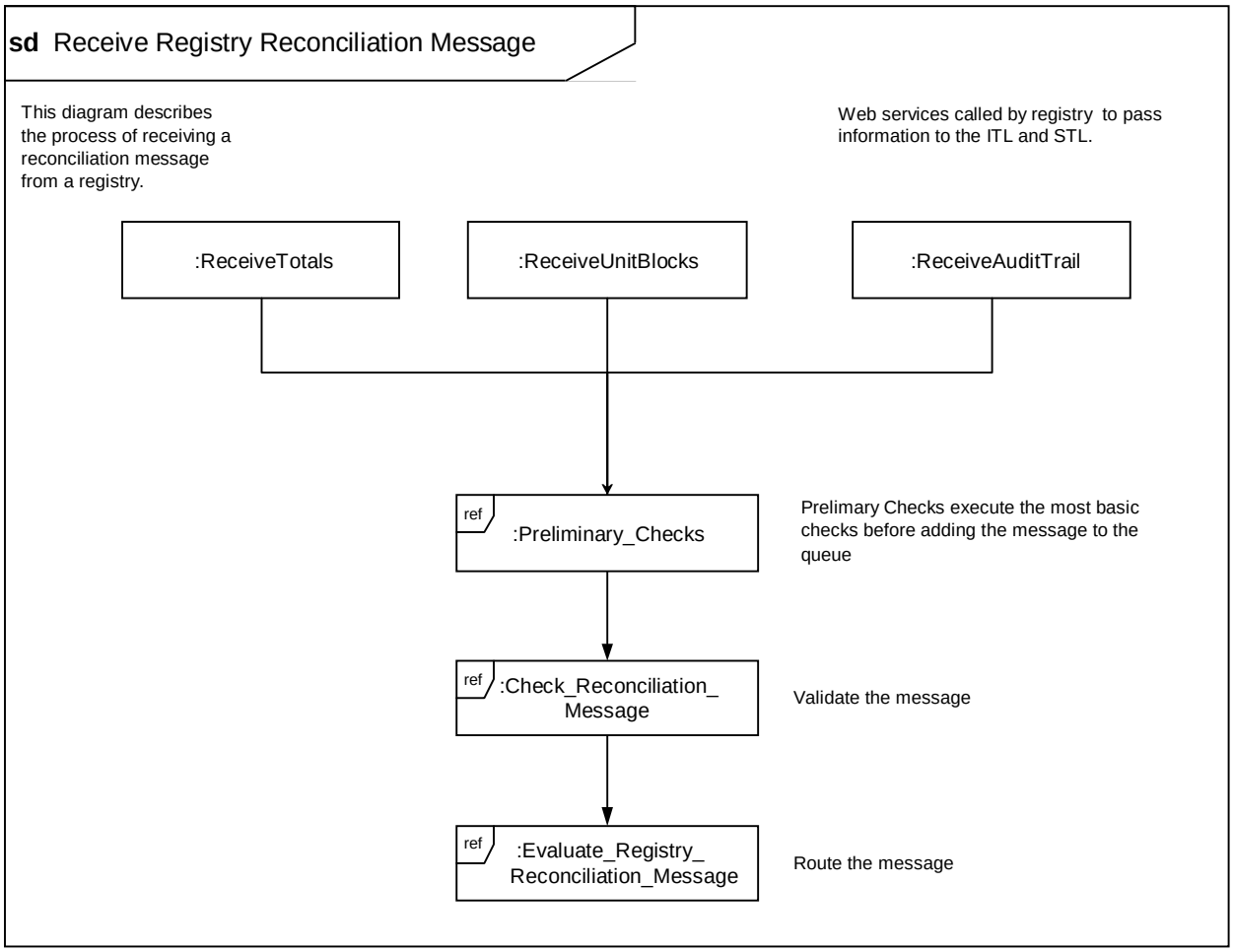
6.4 Reconciliation Processing Flow Diagrams

Figure 6.10: Start_Reconciliation



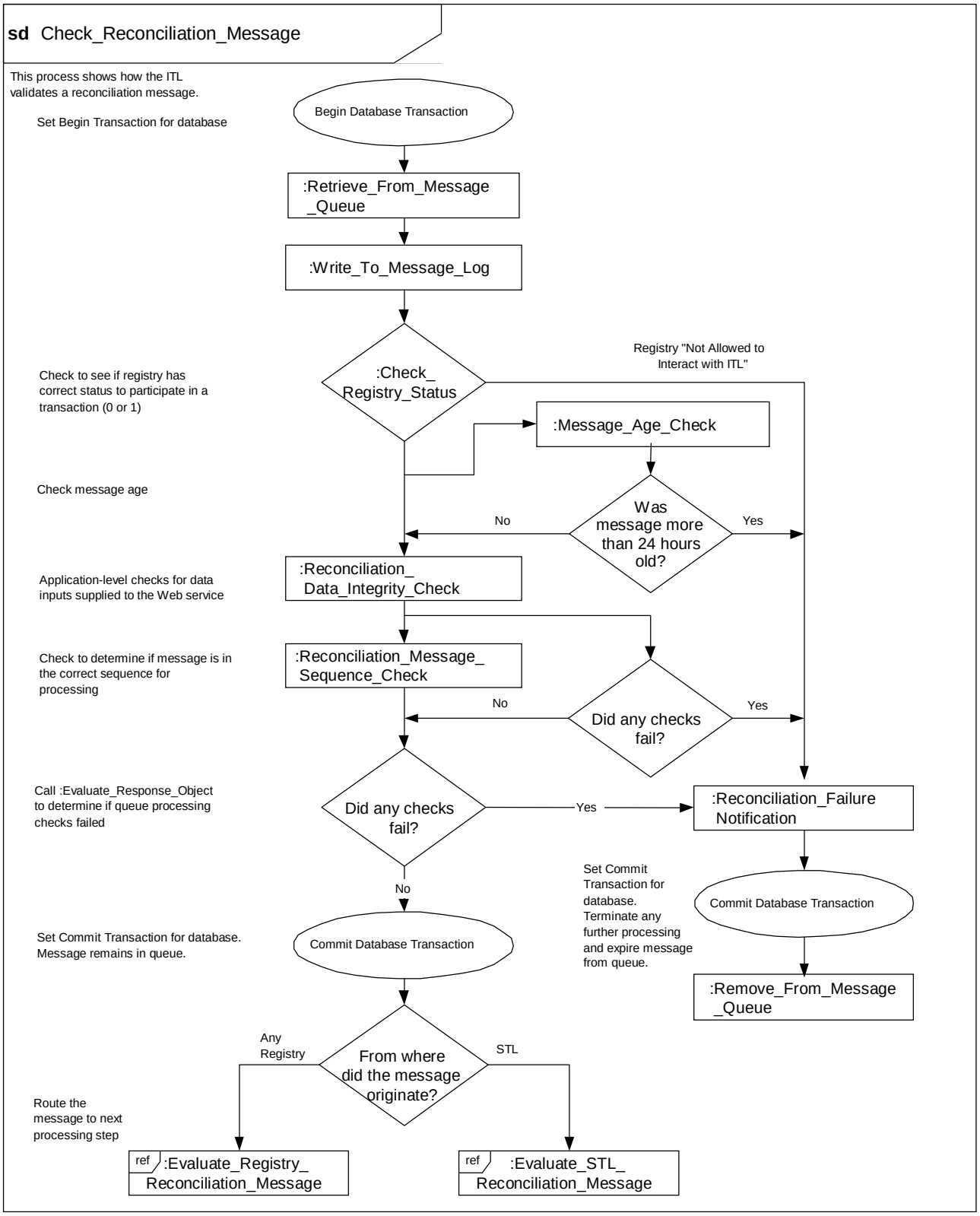
1301 **Figure 6.11: Receive Registry Reconciliation Message**

1302



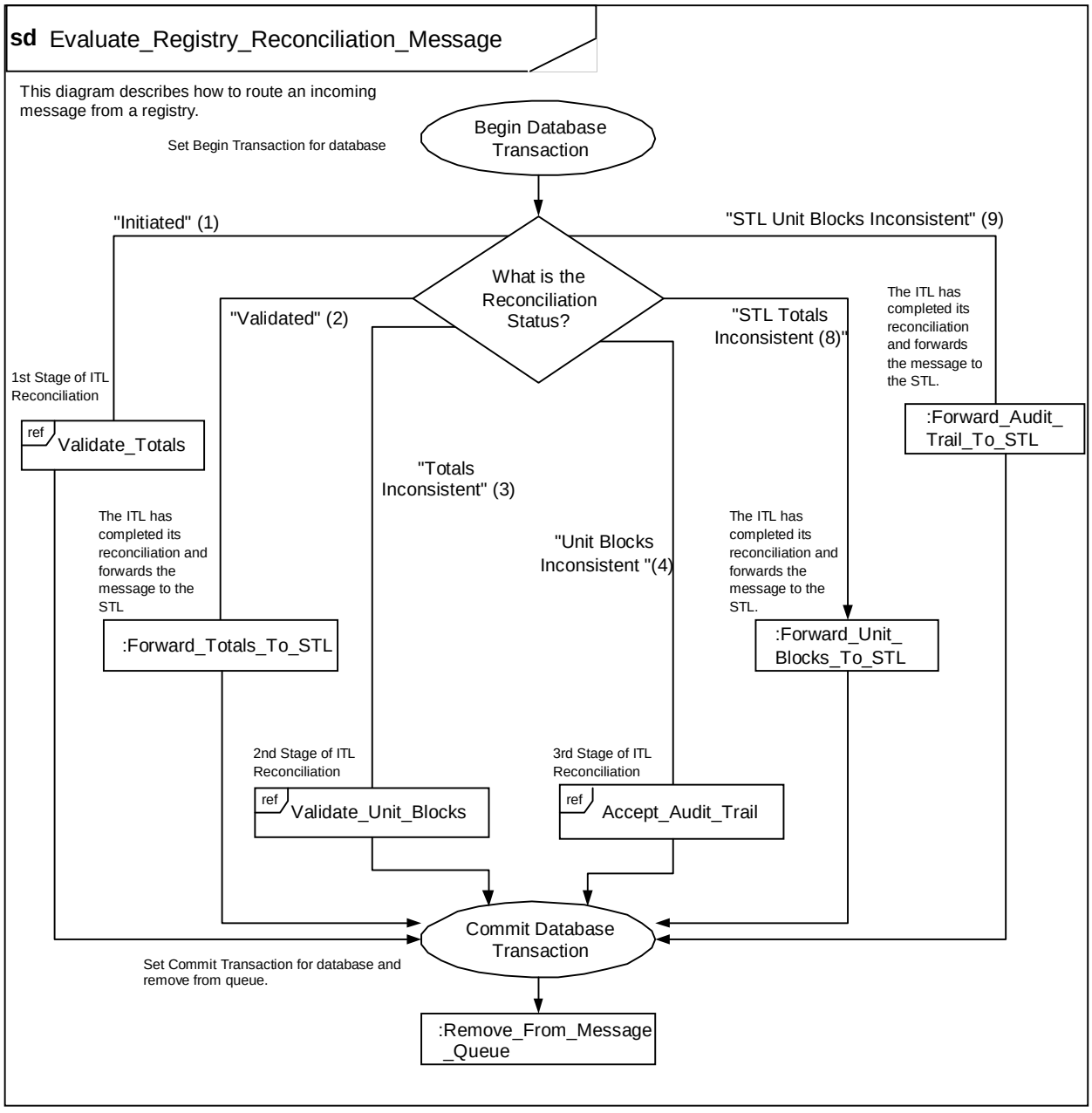
1303

Figure 6.12: Check Reconciliation Message



1307 **Figure 6.13: Evaluate Registry Reconciliation Message**

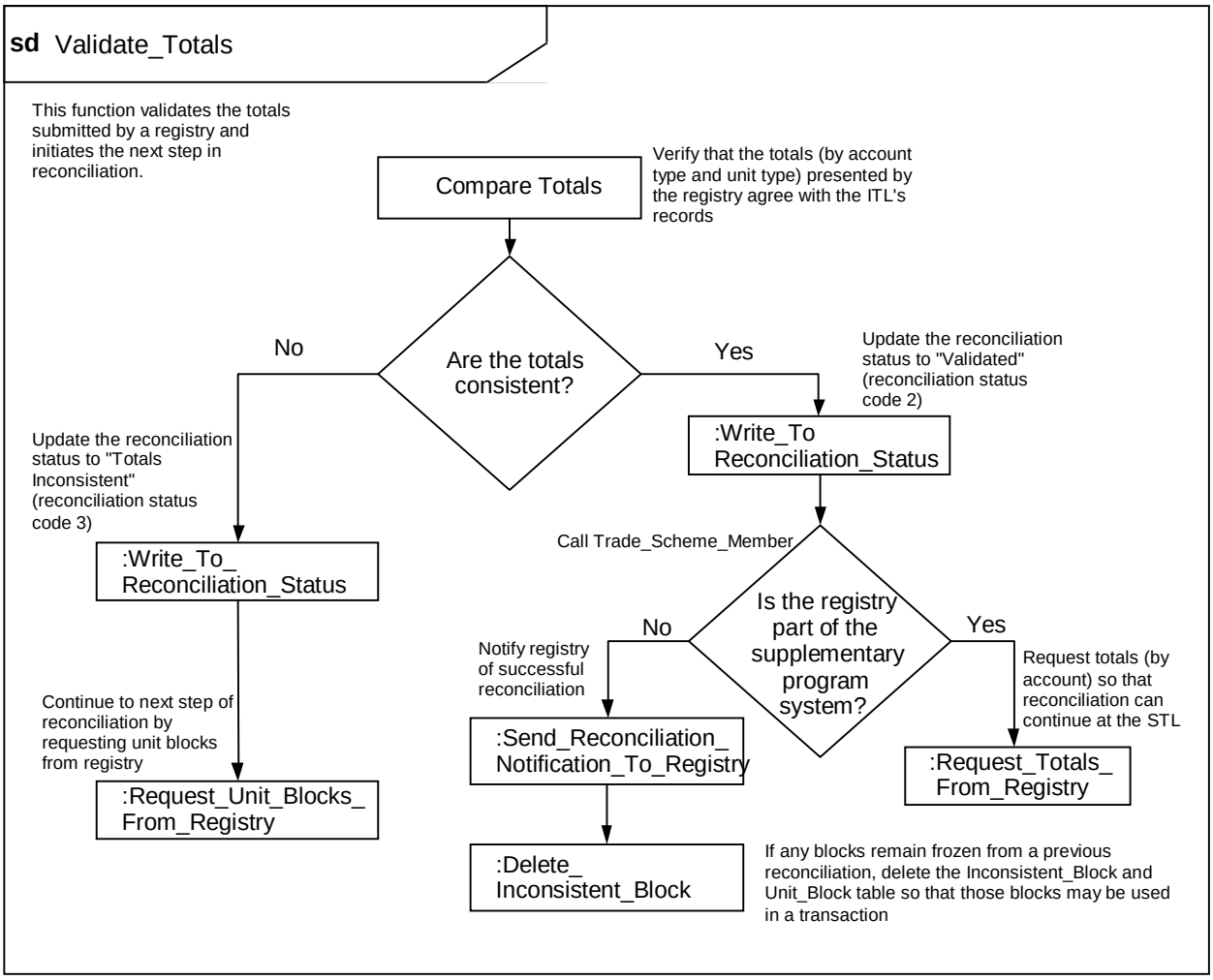
1308



1309

1310 **Figure 6.14: Validate Totals**

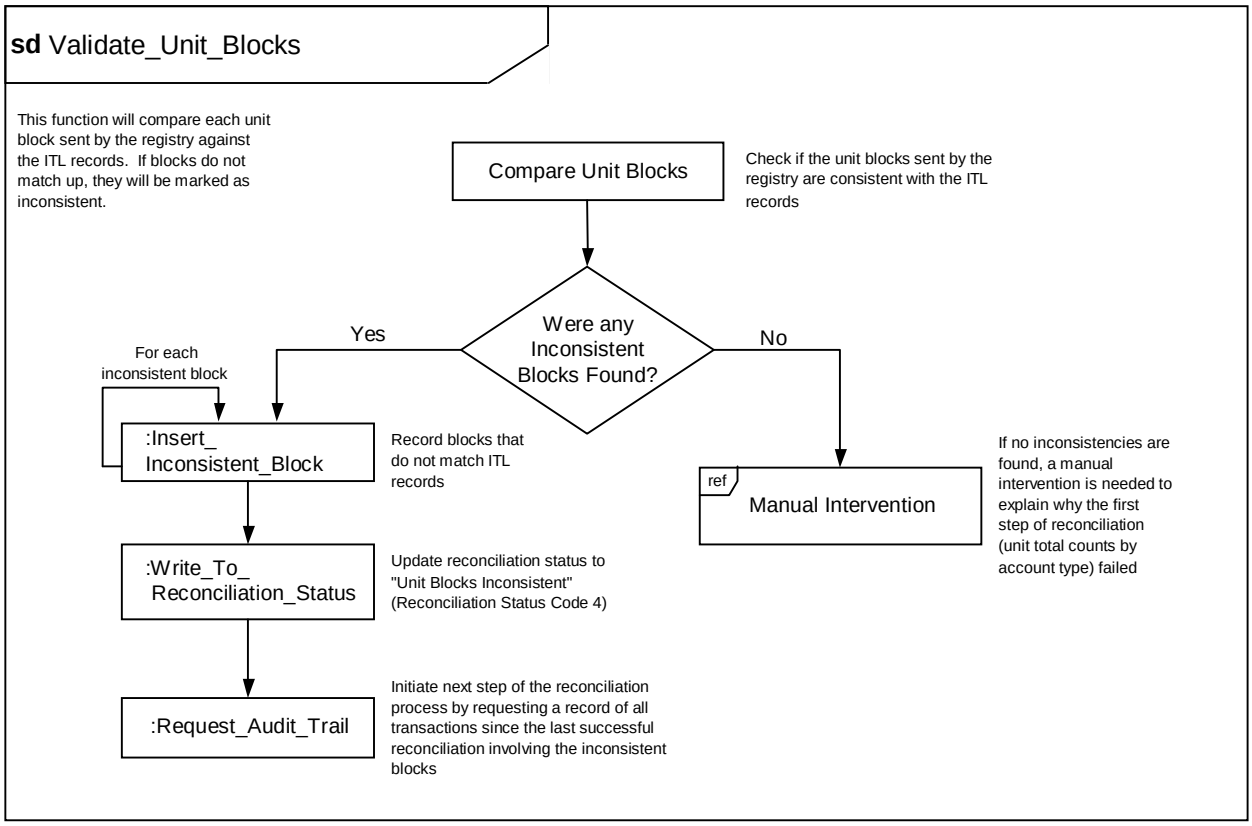
1311



1312

1313 **Figure 6.15: Validate Unit Blocks**

1314



1315

Figure 6.16: Receive Audit Trail

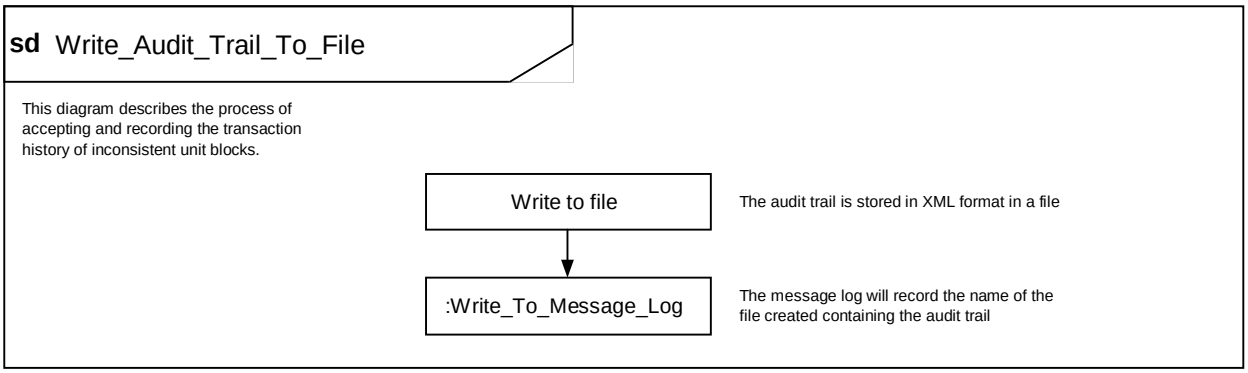
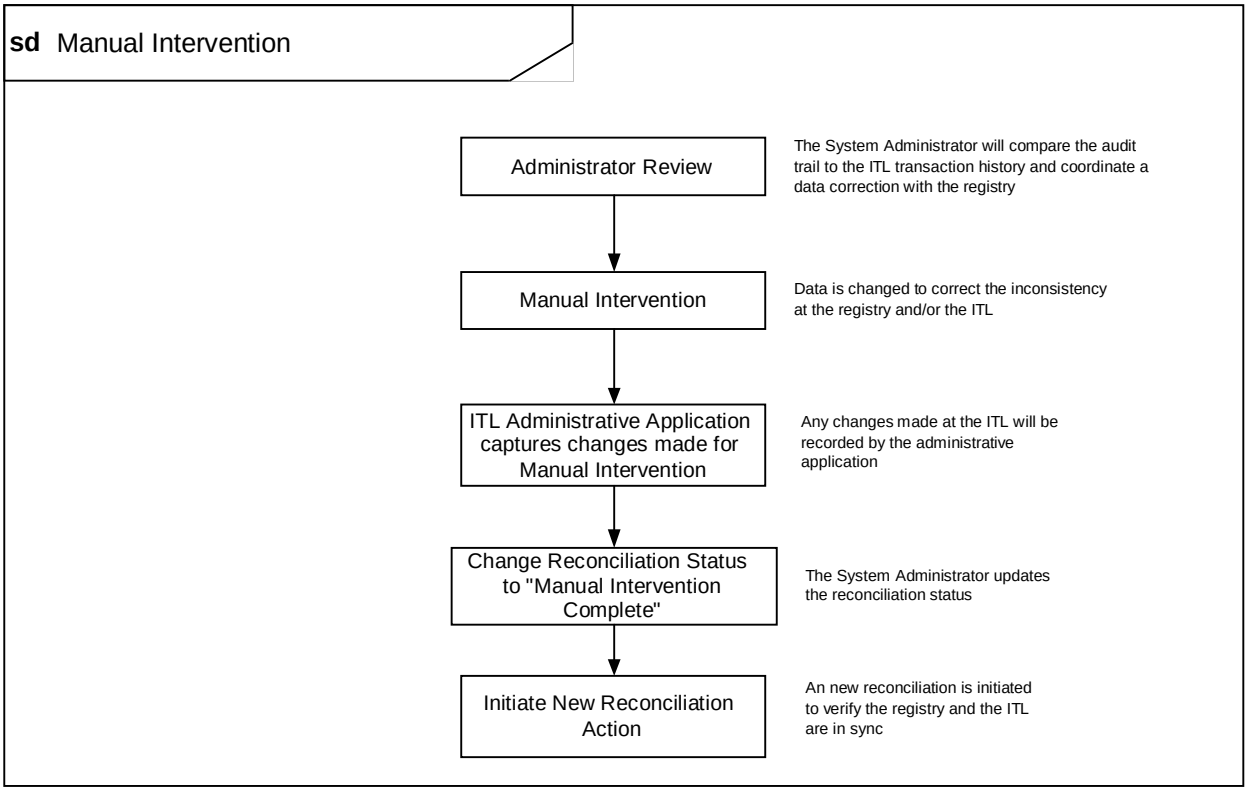
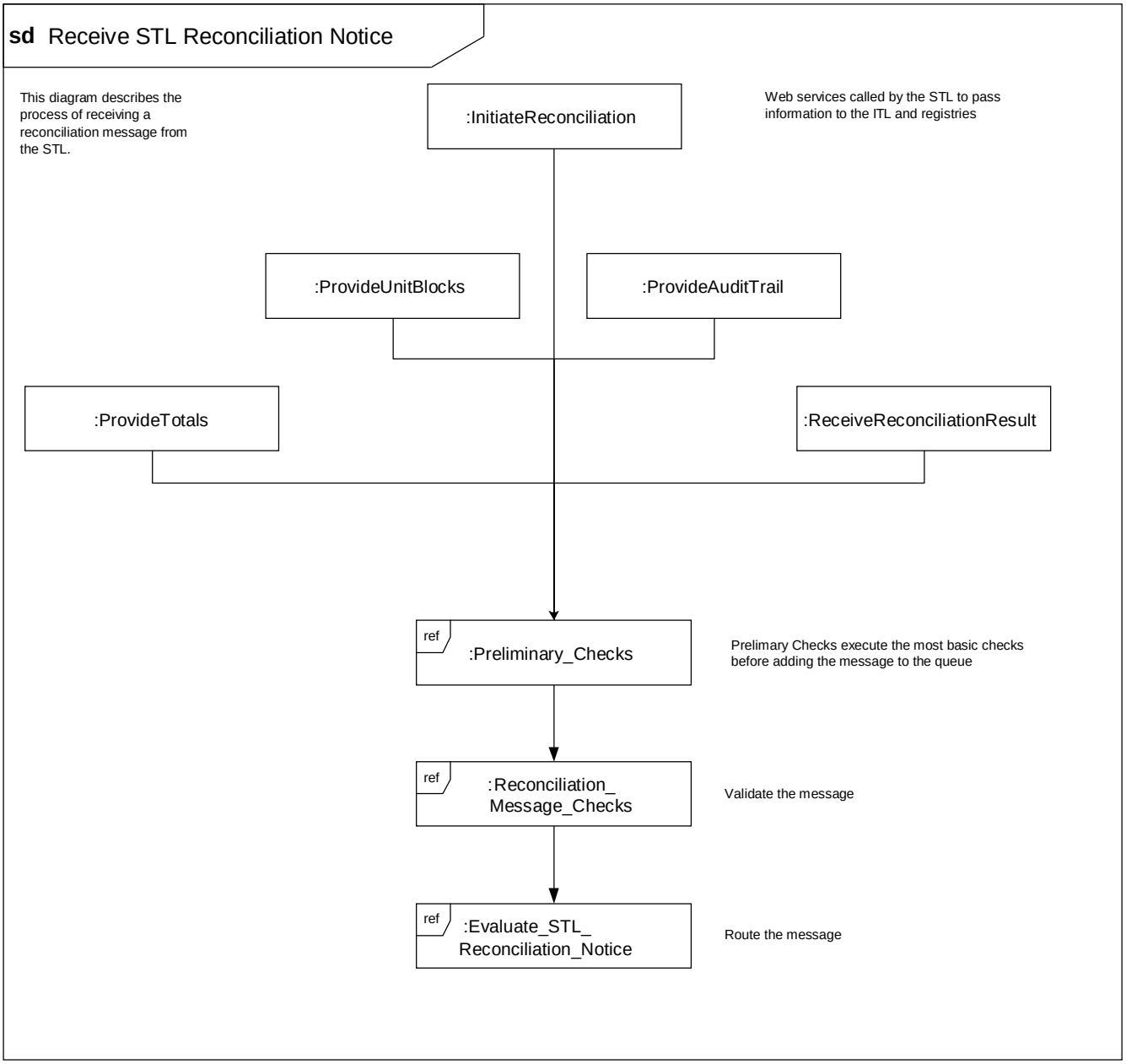


Figure 6.17: Manual Intervention



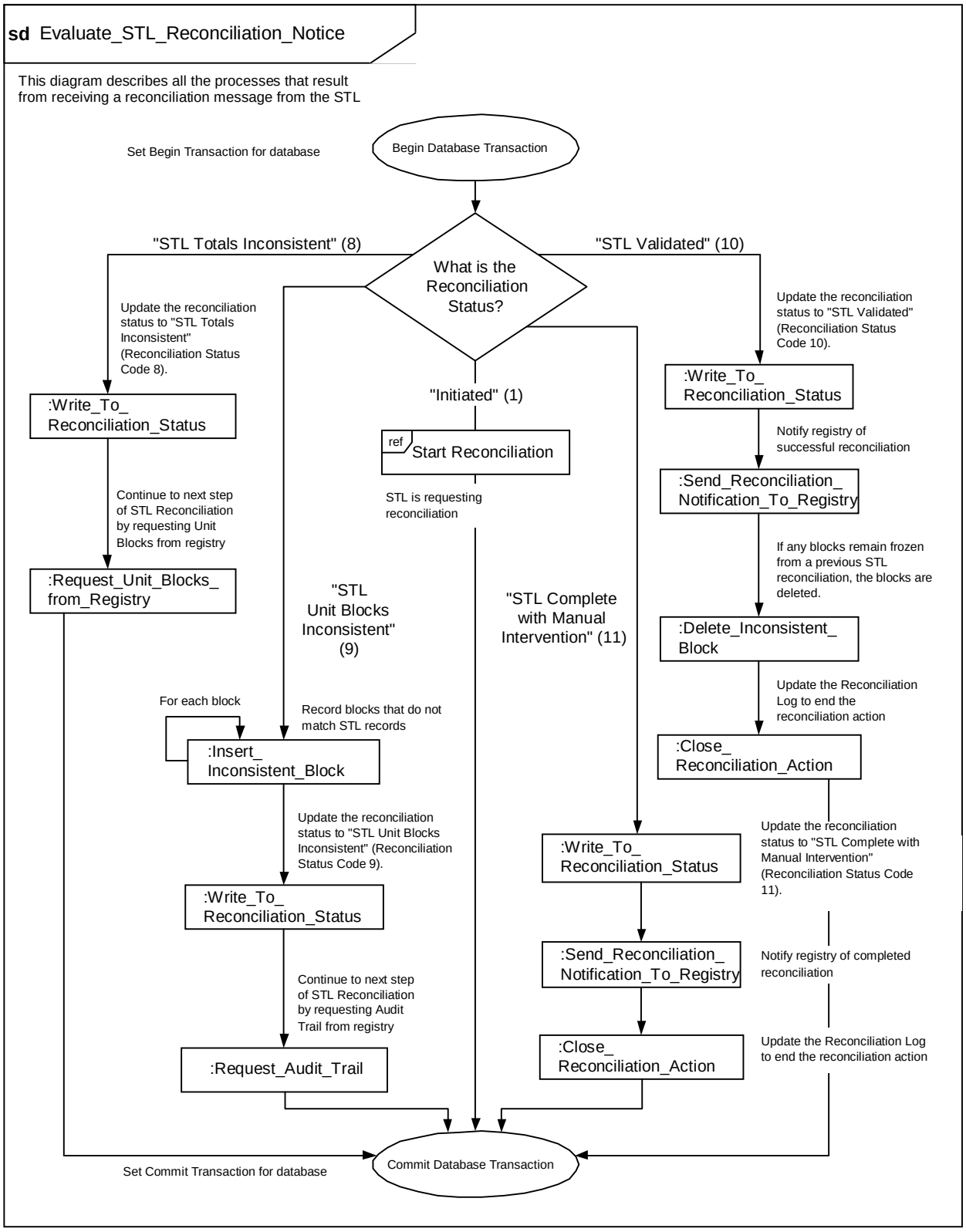
1324 **Figure 6.18: Receive STL Reconciliation Notice**

1325



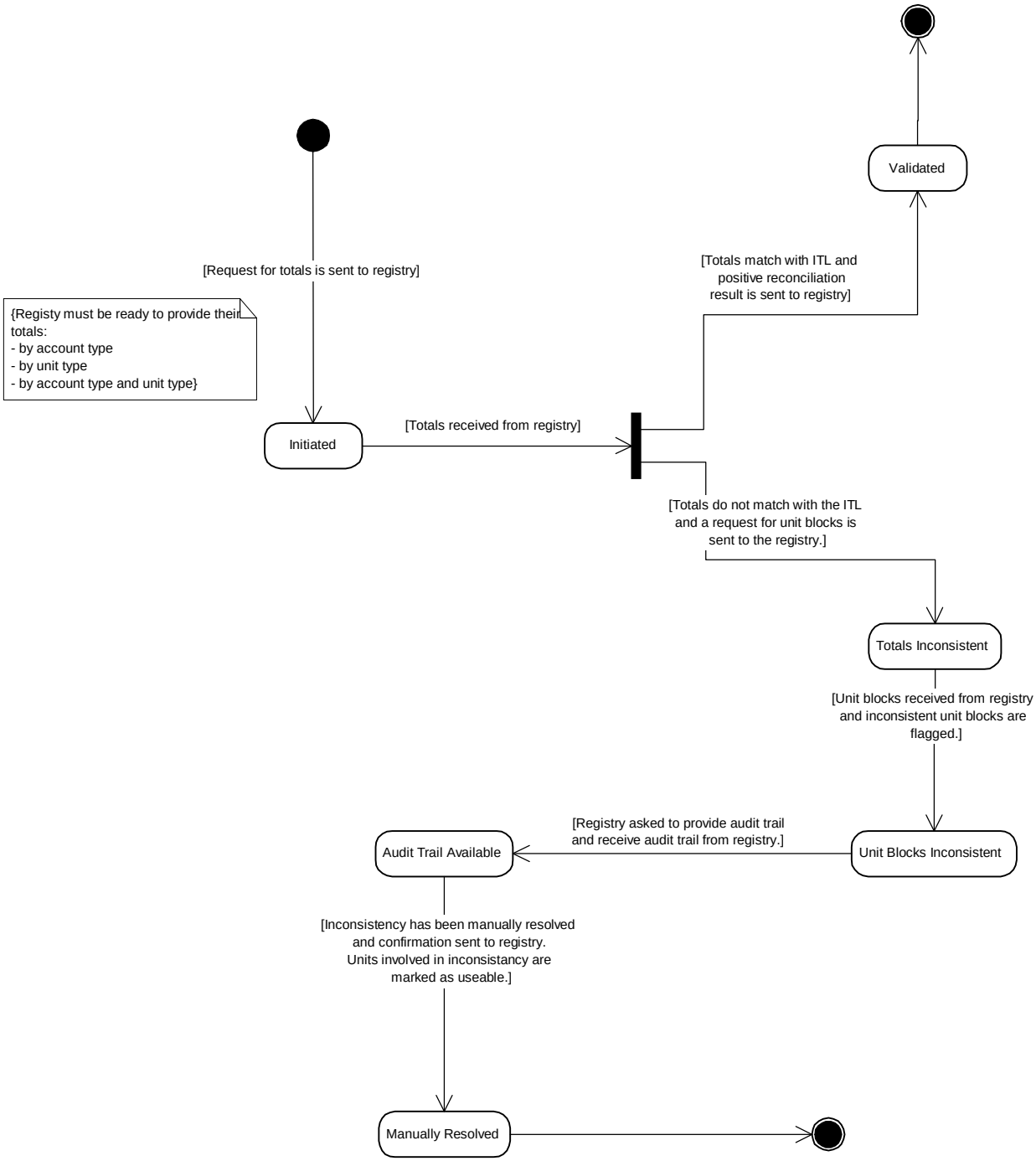
1326

Figure 6.19: Evaluate STL Reconciliation Notice



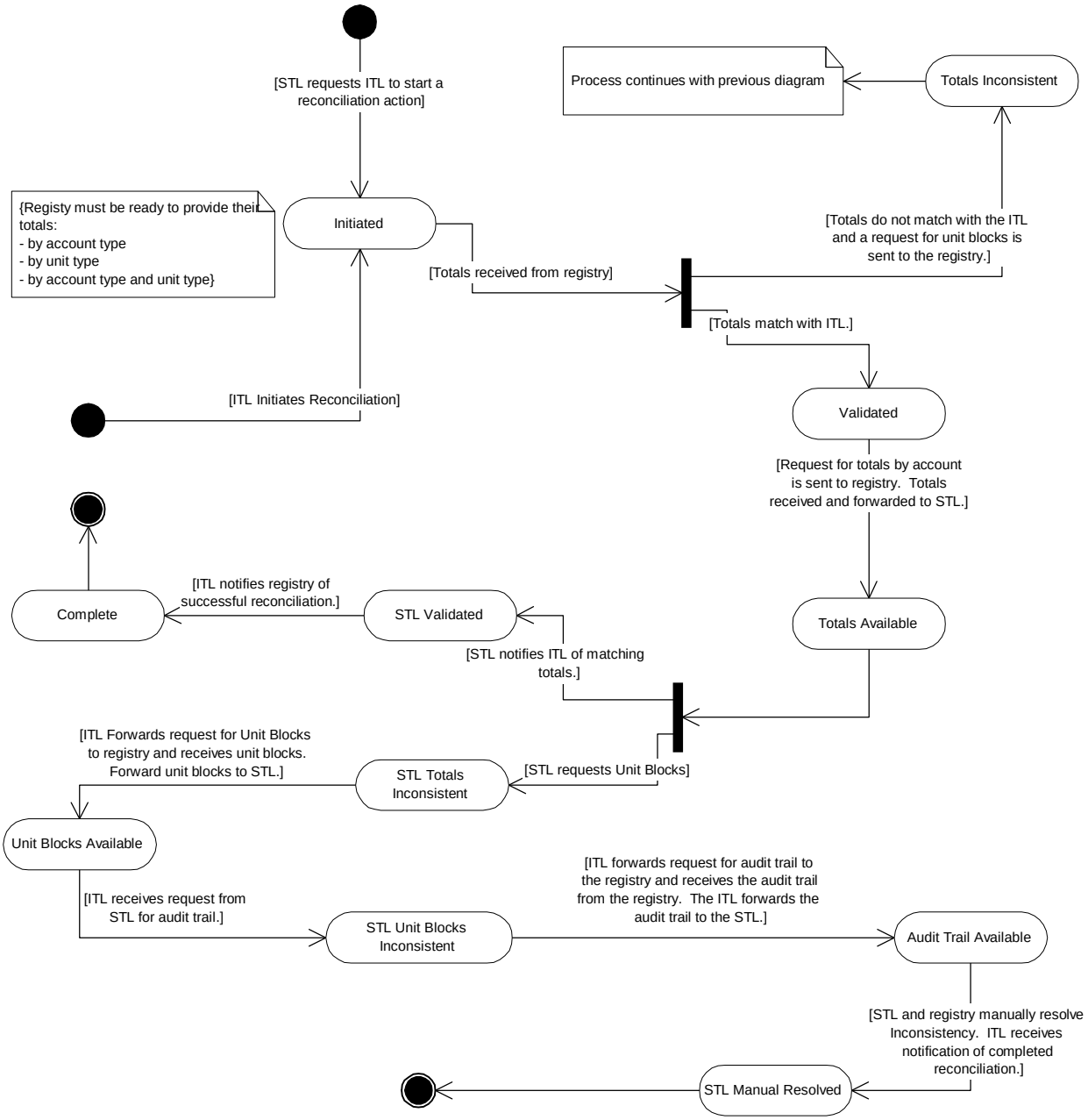
6.5 Reconciliation State Diagrams

Figure 6:20: Registry - ITL Reconciliation State



1335 **Figure 6.21: ITL - STL Reconciliation State**

1336



1337

7. Administrative Processes

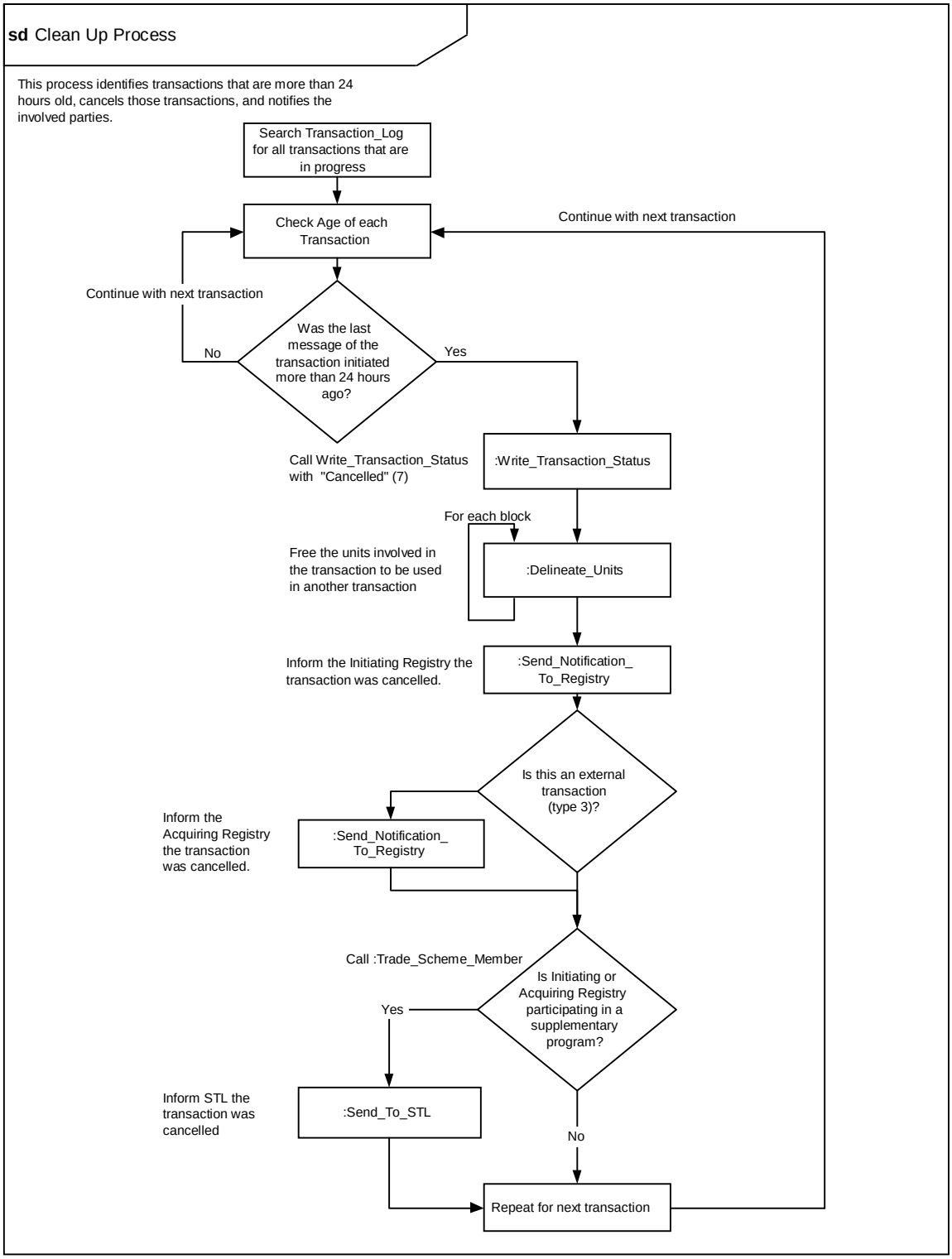
7.1 Transaction Status

The ITL will make available to registries the current status of all transactions. The GetTransactionStatus Web service will provide the current status recorded at the ITL to a requesting registry. The registry will initiate the Web service on the ITL and pass it a transaction ID. The ITL will return the status of the transaction and the date and time the status was last updated. These requests are not recorded or tracked by the ITL. See Get_Transaction_Status in Annex E.

7.2 Transaction Cleanup Process

In order to maintain data integrity and to ensure that registries adhere to established timing requirements, on a periodic basis the ITL identifies transactions that are in progress and for which a message has not been received within 24 hours. This check shall be performed once an hour. The ITL cancels these transactions. After the transaction is cancelled, the unit status is modified such that they are available to be involved in another transaction. Notification is sent to the registries involved in the transaction through the AcceptNotification Web service. The system administrators of the registries should review the notification, investigate the reason for the lack of communication, and reinitiate the transaction as a new transaction, if appropriate. See Figure 7.1 and the Transaction_Cleanup in Annex E.

Figure 7.1: Transaction Cleanup Processes



7.3 tCER and lCER Management

7.3.1 Identify Expired Units

Every 24 hours the ITL will execute a job to identify any tCERs or lCERs that will expire within 30 days. After identifying these units, the ITL will notify each registry that holds one of the units that the units must be cancelled or replaced within 30 days. See CER_Expired_Checks in Annex E.

If tCERs or lCERs expire, a report within the ITL administrative application will highlight any registries that hold expired units. The ITL administrator will then notify the Executive Board that the registry has failed to take appropriate steps to deal with an expired tCER or lCER.

7.3.2 Lack of Certification Report Notice

If the persons responsible for a project have not submitted a certification report, the CDM Executive Board may request that the ITL halt trading (except cancellation) of all units associated with that project. If this occurs, the ITL administrator will initiate a job that will notify all registries holding affected units. The notifications will be sent through the AcceptMessage Web service method at the registry. See Lack_of_Certification_Report in Annex E.

7.3.3 Reversal of Storage Notice

If a reversal in the storage of greenhouse gasses occurs at a project, the CDM Executive Board will notify the ITL administrator who will initiate a job that will determine and then notify registries of the actions they must take. The job will temporarily suspend trading of all units associated with the project and then calculate how many units each registry must replace. Each registry must replace the same percentage of their holdings (excluding cancelled or previously replaced units) as the percentage of the reduction in storage. Each affected registry will be notified through the AcceptMessage Web service method. The message will alert each registry to the number of units it must replace. See Reversal_of_Storage in Annex E.

The registry will then initiate replacement transactions until the appropriate number of CERs have been replaced. The replacement transaction submitted by the registry must reference the identifier of the notification sent by the ITL so the ITL can track when the registry completed replacement.

Another job will run daily and will check for non-replacement by each registry with the Reversal of Storage Notification. The job will search the notification table for open notifications of this type and then evaluate each registry's response by searching for transactions that reference the applicable Notification ID. Each registry has 30 days to complete replacements. See Reversal_of_Storage_Replacement in Annex E.

After 30 days, the ITL Administrator will send a report to the Executive Board regarding the action.

7.4 Outstanding Unit Identification

After the end of a commitment period, the ITL administrator will initiate a job that identifies all units for the prior Commitment Period that have not been retired, cancelled, or carried-over. For each registry found to be holding such a unit, the job will send a notification through the AcceptMessage Web service at that registry. The message to the registry will indicate that it must cancel or carry-over these units within 30 days. A report within the ITL AA will identify those

registries that have not dealt will all applicable units within 30 days. See Outstanding_Unit_Cleanup in Annex E.

7.5 Registry Time Synchronization

In order to maintain consistent system time between the registries and the ITL, the ITL checks the system time of each registry on a periodic basis. If the time is found to be unsynchronized by a specified amount, a message is sent to the system administrator of that registry. In order to accommodate this function each registry must make available a ProvideTime function which is used by the ITL to retrieve the current time of the registry.

Registries must implement the ProvideTime public web service method for the ITL to call. The ITL will compare the time this function returns with the official system time. Detailed specifications for the ProvideTime method are in Annex D to the DES.

The ITL will log the time synchronization result in the System Log and contact the registry manager using a manual process or through a general message if a time problem is identified. See Time_Sync in Annex E.