DATA EXCHANGE STANDARDS FOR REGISTRY SYSTEMS UNDER THE KYOTO PROTOCOL

DRAFT

TECHNICAL DESIGN SPECIFICATION (Version 1.0, Draft #5)

Non-paper

8 June, 2004

Revision History

Date	Version	Description	Author
31/10/2003	1.0	Draft #1	Andrew Howard
14/11/2003	1.0	Draft #2	Andrew Howard
21/11/2003	1.0	Draft #3	Andrew Howard
27/05/2004	1.0	Draft #4	Andrew Howard
08/06/2004	1.0	Draft #5	Andrew Howard

[This page intentionally left blank.]

Table of Contents

List of Figures

1. Introduction

1.1 Purpose

This document contains technical specifications for data exchange between registries and the Independent Transaction Log (ITL) under the Kyoto Protocol. This exchange of data forms the technical basis for transactions under the mechanisms defined in Articles 6, 12 and 17 of the Kyoto Protocol and the modalities for the accounting of assigned amounts (to demonstrate compliance with emission targets) under Article 7.4 of the Kyoto Protocol.

These technical specifications contain full information on *how* the data exchange requirements are to be implemented. They are based on the functional specifications for data exchange, which define in broader terms *what* data are exchanged and *by whom*. The technical specifications are necessary to ensure that the registries and the ITL employ consistent data exchange and messaging functionality.

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 functionality 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 greenhouse gas emissions trading scheme. This is being developed under Article 20 of EU Directive 2003/87/EC.

1.2 Intended Audience

This document is to guide technical experts in the design, development and implementation of communication functionality in registries and the ITL.

1.3 Scope

The data exchange standards define how data are to be exchanged between national registries, the CDM Registry and the ITL under the Kyoto Protocol, as well as any STLs established. The technical specification includes the communication protocols to be used and a messaging architecture that includes an overall design for message management, message content, and data transfer formats. It defines in detail the specific data elements to be exchanged between registry systems to support designated functionality throughout the process.

The diagram in Figure 1-1 demonstrates how both national registries and CDM will both send and receive messages enabling two-way communications exchanges to the ITL through a communications hub.

This technical specification includes:

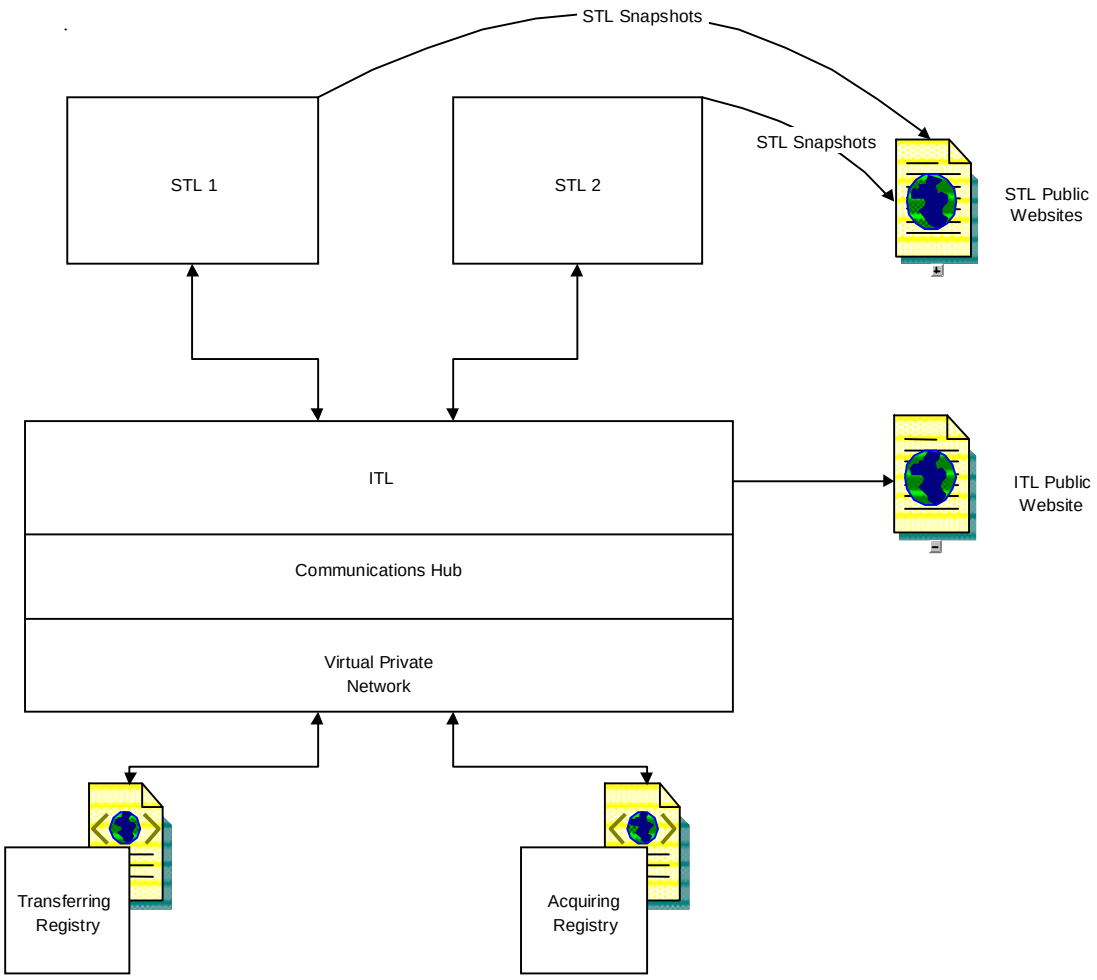
Section 2 Assumptions, Constraints and Requirements

Facts and constraints identified in the functional specifications and held to be true for the technical specification to be valid.

Section 3 Data Exchange Mechanism Specifications

Specifications relating to registration, authentication and communication protocols required.

Figure 1.1: Communication Via the Data Exchange Standards



Section 4 Unit Transactions

Specifications for message exchange relating to unit transactions.

Section 5 Reconciliation

Specifications relating to the process of reconciliation.

Section 6 Administrative Processes

Specifications regarding notifications to registries, time checks and message cancellations.

Section 7 Data Logging Specifications

Specifications for retaining records, utilizing internal logs and communicating transaction data for reconciliation.

Section 8 Change Management Specifications

Specifications to manage and distribute changes in the data content, messages, or sequence of information exchange to accommodate new requirements or changes in requirements.

Section 9 Initialisation of Registry

Specifications on the start up processes a registry will be required to complete before initiating communication and data exchange with the ITL.

Annex A **Glossary of Terms**

This annex provides definitions, acronyms and abbreviations relevant to this document.

Annex B **Web Service Operations and Functions for Transaction Processing**

This annex contains the detailed specifications for the Web services and programming functions which receive and/or generate transaction messages.

Annex C **Web Services Operations and Functions for Reconciliation**

This annex contains the detailed specifications for the Web services and programming functions relating to reconciliation.

Annex D **Web Service Operations and Functions for Administrative Processes**

This annex contains the detailed specifications for the Web services and programming functions which receive and/or generate administrative messages.

Annex E **List of Checks and Response Codes for Transaction Processing**

This annex identifies the categories of transaction responses and provides a numeric list of responses.

Annex F **Definition of Identifiers**

This annex provides detailed specifications and rules for creating and using identifiers for entities for which information is exchanged.

Annex G **List of Codes**

This annex identifies the codes which are used to represent a variety of categories, types, and statuses which may be contained in messages.

Annex H **Test Protocols for Data Exchange Specification Implementation**

This annex addresses the test requirements for verifying conformance with the Data Exchange Specifications Version 1.0.

Annex I **Messaging Service Specification**

This annex provides information on the required XML message structure.

Annex J **QA Checklist by Requirement**

This annex lists the requirements in the "Data Exchange Standards for Registry Systems under the Kyoto Protocol: Functional Specification, Draft version <7.0>" and cross references the sections of the Technical Specifications which address them.

Annex K **Descriptive Langage (WSDL) Documentation**

This annex provides the WSDLs for the Web services required for message exchange between a registry and the ITL.

1.4 Definitions, Acronyms, Abbreviations and Terminology

See the glossary in Annex A for definitions, acronyms and abbreviations relating to the Kyoto Protocol and related policy documents defining how the Protocol is to be implemented. Note in particular that the term "registries" refers to both national registries and the CDM Registry. "Registry systems" refers to both registries and the ITL.

The terms listed below are used in these technical specifications. This list is intended to promote a common understanding of terminology which is critical to understanding and interpreting the technical specifications, and to ensure that developers and policy analysts use a common vocabulary for describing and discussing the specifications for the data exchange. These definitions are specific to these technical specifications and are not generic.

Process: The business area or category of interaction between registries and the ITL. The primary processes are unit issuance, unit conversion, external transfers, internal transfers (including cancellations and retirements), unit carry-overs, replacement and Expiry Date Change, and reconciliation. In addition, there is an ITL Administration process which addresses the need to manage message exchange failures, project approval, and manual intervention relating to reconciliation processes.

Transaction: The term transaction is used to describe a unique operation on a unit or block of units. A transaction is comprised of a series of actions related to a specific process. Each "transaction" is processed in stages and results in the return of a message to the registry identifying subsequent information on the transaction.

A resubmission of the same information following a rejection or other failure is a new transaction.

Stage: The stage of a transaction defines where in the process of information exchange a particular message or evaluation occurs. A stage ends and a new stage begins when a message has been successfully transmitted and received by either a registry or the ITL or when the last step of a process occurs.

Transaction Status: A transaction must have one of the following statuses: proposed, checked (no discrepancy), checked (discrepancy), accepted, completed, cancelled, terminated, rejected, STL checked (no discrepancy), and STL checked (discrepancy).

Message: A message is a communication between the ITL and a registry. It includes all information exchanged, including transaction data, requests for logs and responses. Messages are transported though HTTP SOAP requests.

Acknowledgement: An acknowledgement is the communication that is returned by a web service (located at either the ITL or at a registry) that a message has been successfully received and the transmission was successful. These are HTTP SOAP responses. The acknowledgement occurs before the message is evaluated in any way other than format checks and minimum version requirements.

Notification: A notification is a communication to a registry from the ITL about a required or recommended action involving unit transactions.

Unavailable Status: Units which are involved in transactions that have been proposed, received by the ITL, and are waiting for a response from either another registry (for an external transfer) or the proposing registry are "unavailable" for other transfer. These units are flagged as unavailable. Similarly, a unit involved in either a discrepancy or inconsistency is marked as unavailable until the inconsistency is resolved or the transaction involving the discrepancy is complete.

Response: A response is the information sent following the processing of a proposed transaction. Typically the response includes the transaction ID, an indicator that the proposed

transaction was successful or unsuccessful, and, if unsuccessful, the response code(s) providing the reason for the failure. These response codes are delivered via a HTTP SOAP request.

Invalidation: An invalidation is a finding by the ITL that a message does not conform to the messaging requirements (including data formats, identifiers, etc.) in this Technical Specification.

Discrepancy: A discrepancy is a finding by the ITL that a proposed transaction does not conform to agreed upon transaction rules.

Inconsistency: An inconsistency is a finding by the ITL that the unit information provided to the registry as part of the periodic data reconciliation process differs from the information retained by the ITL.

Cancellation: Cancellation is the action taken by the ITL for a proposed transaction when no response has been received from a registry within 24 hours.

Termination: Termination is the action taken by a registry to end a proposed transaction which has been determined to be invalid, for which a discrepancy has been identified, or for which the allowable response time has lapsed.

Finalisation: Finalisation is the action taken by a registry to complete a transaction which has been validated by the ITL.

Major Version Number: A major version number is the number assigned to the Technical Specification for Data Exchange Standards for purposes of identifying a specific set of technical requirements. The major version number changes only when a change in the Technical Specifications requires programming changes in a registry.

Minor Version Number: A minor version number is the number assigned to identify version changes in the Technical Specifications for Data Exchange which do not require programming changes within registries. These changes may involve response code table updates, for example.

Component: A component is a group of programming functions that perform related tasks.

Web Service: A Web service is a group of operations that perform communication tasks to and from a registry and the ITL.

Function: A function is a section of programming code which performs a specific task.

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

- 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

1.6 Multiple Language Support

With the exception of the country codes which utilize the alpha codes in ISO3166, all message content exchanged is represented as numeric values. The numeric codes are listed in Annex G. Therefore, the content of all messages is independent of a specific language.

1.7 Validity of Data

The non-functional requirements for registries and the ITL require accuracy and data integrity. These requirements are addressed throughout these Technical Specifications, including in particular, the requirements for data elements and message content. The reconciliation process also provides assurance that these non-functional requirements will be met.

2. Assumptions, Constraints, Standards and Requirements

This Technical Specification is based upon the derivation documents specified in Section 1.5. In particular, it is based upon the constraints and requirements contained in the Functional Specifications for the Data Exchange Standard. A detailed cross reference of these technical specifications and the specific requirements is included in Annex J.

2.1 Standards

The ITL will contain information on party eligibility provided by the Secretariat.

These data exchange standards utilize the following standards:

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

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

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

3. Data Exchange Mechanism and Specifications

3.1 General Requirements

Communications between the registries and the ITL must be secure and processed as real-time transactions. The functional requirements for data exchange specify the use of TCP/IP connections using encrypted messages over the Internet. Communications must be protected from modification or interception in transit. Users must be authenticated to ensure their identity and associated permissions. Communications will be initiated by either registries or the ITL and an immediate response will be expected.

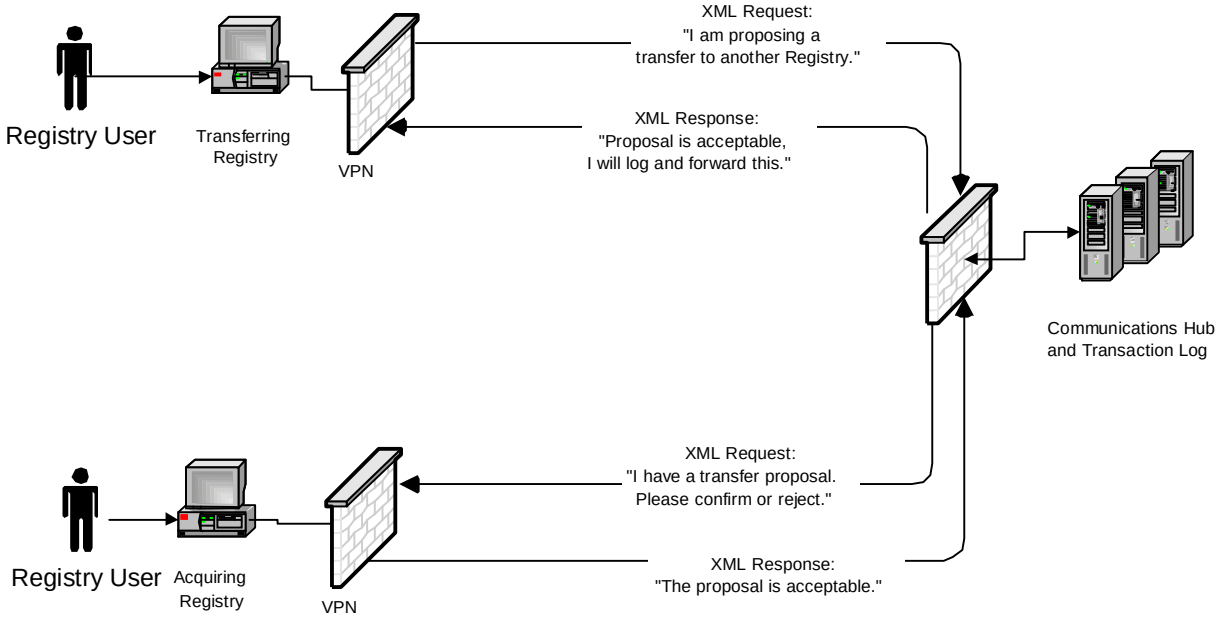
To provide this functionality, the registries and ITL shall utilize a consistent and coordinated set of technical solutions. The technical specifications require:

- Web services using Simple Object Access Protocol (SOAP);
- Virtual private network (VPN);
- XML formats adhering to the described standards in Annex I;
- Digital signature authentication; and
- Network time protocols.

Each of these technologies, with the possible exception of the VPN requirement, is platform and language independent. As discussed below, the hardware specifications for the VPN will take into account cost, interoperability and existing registry hardware, to the extent feasible.

Figure 3.1 provides a diagram of the basic architecture of the data exchange mechanism required for communications between registries and the ITL.

Figure 3.1: Data Exchange Architecture



3.2 Communications Specifications

All registries and the ITL shall use Web services to support the sending and receiving of messages. Web services enable disparate applications running on different machines to easily exchange data with one another without requiring additional proprietary third-party software or hardware. Web services depend upon a standard XML messaging systems and SOAP and therefore are not tied to any one operating system or programming language. Based on common industry standards and existing technology such as XML and HTTP, web services costs very little to deploy. Any information that is exchanged to and from both registries and the ITL shall be through the use of XML exchanged via SOAP. For technical specifications on the construct of these documents, see Annexes I and K.

SOAP is one of several an XML-based protocols for exchanging information between computers and is widely used in the internet community. Since SOAP runs primarily on top of HTTP and XML, all communications are encrypted using Secure Socket Layer (SSL).

Both the ITL and all registries shall be available for requests via the Internet.
The technical specification for the functionality of these Web services are defined in the Web services and functions specified in Annexes B, C, and D.

3.3 Data Transfer Security

3.3.1 Virtual Private Network

All communications to and from registries and the ITL shall be protected using hardware-based virtual private network (VPN) technology. VPN technologies provide the ability to "tunnel" through the Internet from one point to another, protecting all communications. Prior to the creation of a VPN tunnel, a digital certificate is issued to a prospective client end-point, allowing the client to provide proof of identity. The client installs the certificate into their VPN end-point. The client initiates the connection and is authenticated by the VPN server. Using digital certificates, the VPN server accesses a central authority (CA) to negotiate authentication credentials. During the tunnel creation process, encryption is negotiated, ensuring that all communications through the tunnel are protected.

The ITL shall be located on an Internet-connected network protected by a hardware-based firewall. The firewall shall be configured with rules such that only "registered" clients can attempt to make connections to the VPN server. Client registries must have fixed public IP addresses. Access to the ITL is contingent on communications originating from known public IP addresses. Client registries shall implement hardware-based VPN end-points for use in connecting to the system. These VPN end-points shall be configured with the appropriate credentials as provided by the ITL administrators. The client VPN end-points shall be configured to maintain the VPN tunnel permanently, in order to allow reliable, two-way, real-time communication between the ITL and a client registry at all times.

3.3.2 Client VPN Specifications

VPN equipment at the client registries shall be dedicated devices that can reliably terminate the VPN connection to the ITL as well as maintain acceptable performance levels. The recommended VPN equipment adequate for a client registry VPN connectivity is a Cisco PIX firewall/VPN device. Information on the Cisco PIX firewall is available at http://www.cisco.com/warp/public/cc/pd/fw/sqfw5DD/.

3.3.3 IPSec VPN

In addition to the site-to-site VPN infrastructure, the use of IPSec VPN will provide for site-to-site authentication, data integrity, and data encryption. IPSec VPN configurations provide for authentication between two end-points in a VPN connection. The ITL will identify and authenticate the remote client via the IPSec connection using a digital certificate provided by a trusted certificate authority (CA).

IPSec also ensures data integrity of all communications passed through the VPN tunnel. Packets of data are hashed and signed using the authentication information established by the

VPN. Data confidentiality is also ensured by IPSec encrypting the data using Triple DES (3DES). This encryption addresses only the network traffic itself, not the application level SOAP communications.

3.3.4 SSL

SSL shall be used for all communications between the registry and the Communications Hub. SSL provides application server-to-application server authentication as well as data encryption. Since IPSec VPN provides only site-to-site authentication, a method is required to authenticate the actual registry communications to the ITL. Additionally, SSL protects any communications that may pass over the networks at the registry site before transport through the VPN on to the ITL.

3.4 The Communications Hub and Message Queue

The security layer and supporting hardware and software between the VPN and ITL database is the Communications Hub. The Communications Hub receives and logs all messages passed through the VPN. 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 scalability during peak transaction times.

3.5 Data Transfer Format Specifications

All message packages must utilize XML and conform to the standards in Annex I. WSDL specifications for these XML messages are defined in Annex K.

3.6 Certificate Authority

SSL requires the use of a trusted certificate authority (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.

3.7 User Accounts

The ITL VPN shall register and maintain user IDs and passwords for users who are logging in directly to the ITL's web application. A user account is valid for an indefinite period of time. The ITL may revoke or replace a user's registration or password if there is a suspected breach of security or rules of behaviour by a user.

3.8 Time Validation Specifications

To ensure that transaction rules are accurately and consistently applied to all proposed transactions, the ITL and registries shall use a consistent convention for recording time, and shall also utilize time synchronization practices and procedures to ensure accurate logging and sequencing of all transactions. Accurate and consistent time clocks are essential to the reconciliation process.

All dates and times shall be recorded as Greenwich Mean Time (GMT).

Time information shall be submitted as a date, hour, minute and second in the format: YYYY–MM–DD HH:MN:SS

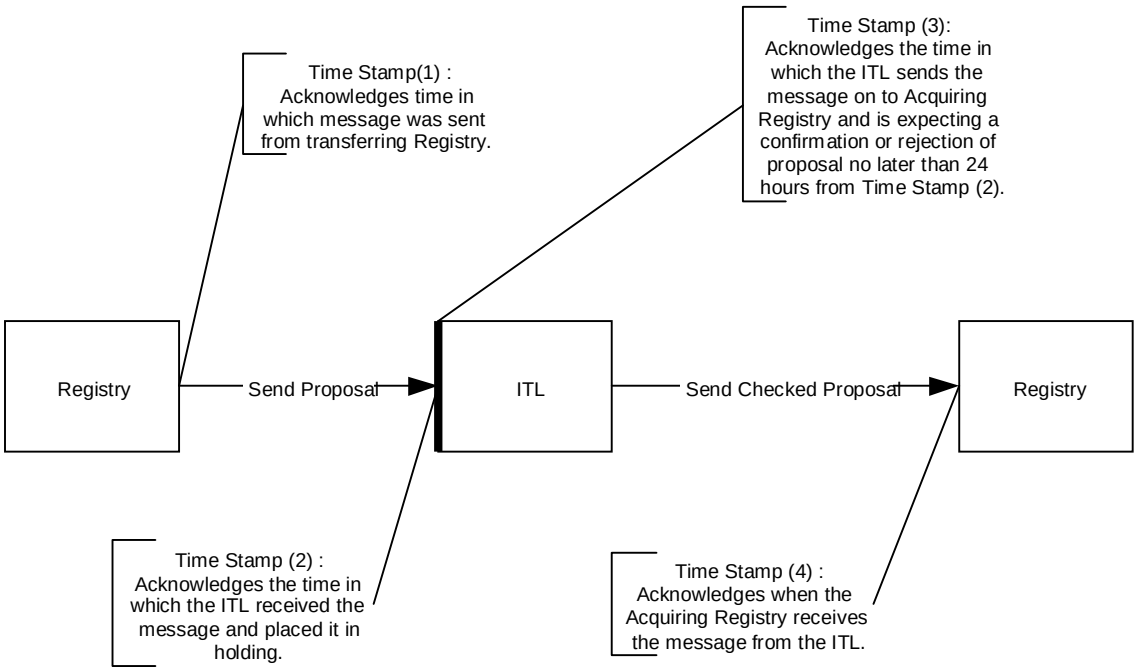
All registries and the ITL shall use Network Time Protocol (NTP) version 3 or better software to synchronize their clocks with well known public Stratum 2 time servers. NTP ensures that both the ITL and registries maintain consistent and accurate times. NTP software has been

ported to all major computing platforms, and in some cases is bundled with the operating system. Information on NTP is found at http://www.ntp.org/. A list of well known international Stratum 2 NTP time servers is maintained at http://www.eecis.udel.edu/~mills/ntp/clock2a.html. A list of NTP ports to various computing platforms and other resources is found at http://www.ntp.org/links.html. Any NTP version 3 or better client is acceptable. Registries should configure their clients to contact two or more public NTP servers that are close to them in terms of traffic routing on the Internet.

As part of the ITL administrative functions described in Section 6 and Annex D, the ITL shall perform a time check service for registries in which the ITL can request the time. The ITL is responsible for recording the time in which each message is routed through its Communications Hub and enters the message queue.

Figure 3.2 outlines the sequence in which the time stamp of a message is evaluated. Sixty seconds is the recommended allowable time between Request Time Stamp (1) and Response Time Stamp (2). All Web services requests must receive an appropriate response, although this does not mean that any one or all transactions requested in the message is complete. An Acquiring Registry shall complete and send to the ITL a confirmation of the transaction as soon as possible and no later than 24 hours from Time Stamp (2). If a confirmation is not received by the ITL within 24 hours, the transaction is no longer valid and will be cancelled by the ITL.

Figure 3.2: Process Time Stamps



3.9 Message Time-to-Live

Messages should be allowed a minimum of sixty seconds in which to respond to the requesting Web service. In most cases, the time in which it takes to validate the digital signature, user account and password and verify that the message was a well-formed XML document should not exceed sixty seconds. A registry may elect to exceed this limit and accept messages which exceed this timeframe.

4. Unit Transactions

4.1 Unit Transaction Types

This section of the Technical Specification addresses the messages and content requirements necessary to support submissions of unit transactions by registries and the validation of those transactions by the ITL. The unit transactions either involve the transfer of ownership of a unit, a change in a attribute of a unit, or the replacement of an lCER or tCER. This section will describe the data exchange flow, the responsibilities of registries, and the responsibilities of the ITL in order to complete a unit transaction.

The following unit transactions are described:

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

4.1.1 Issuance

The issuance of AAUs is undertaken by a Party 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 removals of greenhouse gases through LULUCF activities. The issuance of lCERs, tCERs, and CERs into a pending account is undertaken by the CDM Executive Board, in the CDM registry, on the basis of verified and certified reductions in greenhouse gas emissions or removals of greenhouse gases from the atmosphere through a CDM project activity. Issuance of such units is monitored and validated by the ITL.

The issuance process for AAUs, RMUs, lCERs, tCERs, and CERs follows the single registry model of data exchange described in Section 4.3.

4.1.2 Conversion

The conversion of AAUs and RMUs to ERUs is undertaken by a Party in an account in its national registry. AAUs are converted to ERUs in its national registry on the basis of verified reductions in emissions through a joint implementation project. RMUs are converted to ERUs on the basis of verified removals of greenhouse gases through a joint implementation (JI) project. Conversion of such units is monitored by the ITL.

The conversion process of AAUs and RMUs to ERUs follows the single registry model of data exchange described in Section 4.3.

4.1.3 External Transfer

The external transfer of AAUs, RMUs, ERUs, tCERs, lCERs, and CERs 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 external transfer of such units is monitored and validated by the ITL.

The external transfer process for AAUs, RMUs, ERUs, tCERs, lCERs, and CERs follows the multiple registry model of data exchange described in Section 4.4.

4.1.4 Cancellation

The internal transfer of AAUs, RMUs, tCERs, lCERs, and CERs to a cancellation account is undertaken by a Party, an entity or the CDM Executive Board, on the basis of the amounts proposed by the transferor.

Although the ITL will notify the registry about units which must be carried over or cancelled at the end of a commitment period, it is not necessary to include the Notification ID viewed from the ITL for subsequent cancellation transactions.

The internal transfer of AAUs, RMUs, ERUs, tCERs, lCERs, and CERs follows the single registry model of data exchange described in Section 4.3.

4.1.5 Retirement

The internal transfer of units to a retirement account is undertaken by a Party or an entity, on the basis of the amounts proposed by the transferor. The internal transfer of such units is monitored and validated by the ITL.

The retirement of AAUs, RMUs, ERUs, tCERs, lCERs, and CERs follows the single registry model of data exchange described in Section 4.3.

4.1.6 Carry-over Process

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 must be cancelled. The carry-over of units is monitored and validated by the ITL.

Although the ITL will notify the registry about units which must be carried over or cancelled at the end of a commitment period, it is not necessary to include the Notification ID received from the ITL for subsequent carry-over transactions.

The carry-over of AAUs, ERUs and CERs follows the single registry model of data exchange described in Section 4.3.

4.1.7 Replacement

The replacement of tCERs and 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.

For replacements required by a Reversal of Storage action or a Non-submission of Certification action by the CDM Executive Board, the registry must include the Notification ID associated with the replacement transaction. This Notification ID is used to determine if the replacement should be considered when the ITL performs a follow-up evaluation to assess non-replacement.

The replacement of tCERs and lCERs follows the single registry model of data exchange described in Section 4.3.

4.1.8 Expiry Date Change

The change in the expiry date is undertaken by a party for tCERs and lCERs. For tCERs, this transaction may be necessary to change the expiry date of tCERs issued with a date other than the end of the second commitment period. 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 changes are consistent with the project approvals and updates the tCER and lCER expiry dates in the ITL database.

Although the registry will be notified by the ITL when these units are about to expire, it is not necessary to include the Notification ID received from the ITL in the subsequent Expiry Date Change transaction.

The expiry date update transaction follows the single registry model of data exchange described in Section 4.3.

4.1.9 Internal Transfers and other Transactions Routed to a STL

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.

Supplemental transactions follow either the single registry model or the multiple registry model. If an STL wishes to institute a supplemental transaction to be routed through the ITL, the STL must coordinate the development of this transaction with the ITL administrator.

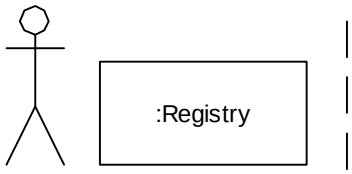
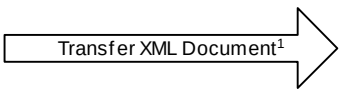
4.2 Description of Data Exchange Flow

The data exchange flow for each unit transaction type follows one of two models: the single registry model or the multiple registry model. Most of the transaction types follow the single registry model. External transactions, and some supplemental transactions not described in this document, follow the multiple registry model. This single registry model is described first, followed by the data exchange model for multiple registries engaged in an external transaction. The sections for both models contain the following subsections:

- UML Behaviour Diagram; and
- Stage Table.

The UML Behaviour Diagram is a high level representation of processes which is designed to capture the participants in each process and the order in which the components are used. Within the diagram, specific technical "components" (representing specific programming logic) are defined. The Behaviour Diagram is based on the standards for the Unified Modeling Language (UML), with text annotations to help non-technical readers interpret it more easily. These diagrams include the following symbols and conventions.

671 **Figure 4.1: Key to UML Diagram**
672

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 participant's icon or 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.
	This symbol indicates that the process supports alternative outcomes in the prior step. Within an alternative, there may be a second alternative scenario, equivalent to programs which contain nested "if…then" statements. In the issuance process, for example, the issuance is either accepted (Result = Success) or a discrepancy is identified (Result = Failed). If successful, the registry can either confirm the issuance or terminate the issuance.
	This symbol indicates that the process within the box will only be executed if a certain condition is met.
	This symbol indicates that the process in the box is repeated a number of times. For example in the Time Synchronization process, the processes within the box are executed once for each registry that interacts with the ITL.
	This symbol represents a message containing an XML document, its transfer and the "acknowledgement" of its receipt. The message is "sent" from one component and "received" by another component, as indicated in footnote (x).
Boxes with dotted outlines	These boxes represent a component or area of functionality necessary to the process, but which does not have specifically defined input or output parameters used for messaging. These components could be defined and implemented by developers in many different ways.
Boxes with solid outlines	These boxes represent a component which performs a specific task necessary to the process. These components either receive or produce the information which is used for messaging.

The Stage Table represents the sequence of events in terms of the "stage" and its relation to the "status" of a transaction or reconciliation action. The stage of a transaction defines where in the process of information exchange a particular message or evaluation occurs. A stage ends and a new stage begins when a message has been successfully transmitted and received by either a registry or the ITL or when the last step of a process occurs. The order in which each defined stage occurs may vary based on the specific process and based on the results of the ITL validation process. The numbers assigned to stages should not be used as an indicator of acceptable stage sequences.

Figure 4.2: Key to Stages

Processes	Stage Code	Stage	Description
Issuance, Conversion, External Transfers, Internal Transfers, Carry-over, Replacement, Expiry Date Change	P	Proposed	Proposal issued from Transferring Registry.
	TR	ITL Review	Proposal evaluated at ITL.
	RR	Registry Review	Proposal evaluated at acquiring registry.
	RA	Registry Accepted	Acquiring Registry has accepted transaction.
	TA	ITL Accepted	ITL has received the evaluation result (accepted or terminated) from the acquiring registry.
	RC	Registry Complete	Registry has completed the transaction.
	TC	ITL Complete	ITL has completed the transaction.

For transaction processes, the stage codes are not submitted in the XML message and are only represented in this table for clarity.

This technical specification describes in general terms the programming logic that should be implemented at the registries and the ITL to establish reliable communications. A list of functions needed to implement this specification is included for each model. Technical information for transaction functions, including required inputs, outputs, and responses, is included in Annex B.

The results of a transaction evaluation conducted by the ITL or an acquiring registry are returned in the XML document in the form of Response codes. Response codes and corresponding checks are grouped by the category of check can be found in Annex E.

4.3 Single Registry Model

The single registry model for transactions applies to the following transaction types:

- Issuance;
- Conversion;
- Cancellation (internal transfer);
- Retirement (internal transfer);
- Carry-over;
- Replacement; and
- Expiry Date Change.

The following steps apply to all the above transactions and describe the sequence of messages necessary to complete the transaction. This description assumes that the transaction is not sent to an STL.

Step 1 - Proposal

The registry sends a proposal for a transaction to the ITL using the AcceptProposal Web service method on the ITL. The proposal contains the transaction type, the units involved in the transaction, and, if appropriate, the transferring and acquiring account and Notification ID information.

Step 2 – ITL Review

The Communications Hub receives the proposal and, once the incoming message is verified to be well formed and authentic, it places the message in a queue for processing. Messages are processed from the queue in the order received. The ITL validates the transaction against the business rules for the appropriate transaction type. If a discrepancy is found, the ITL notifies the registry of the requirement(s) the transaction proposal did not meet. The units involved in the transaction cannot be used in another transaction until the registry sends a termination request.

If the transaction meets all requirements, the ITL records the transaction as pending and marks the units involved in the transaction as unavailable to any other transaction.

The ITL sends the results of the verification, whether a discrepancy was found or not, to the registry via the AcceptNotification Web service method that registries are required to implement. If the ITL identified one or more discrepancies, response codes will be included in the message to indicate net use of the discrepancies.

Step 3 – Registry Complete/Registry Terminate

Once the registry processes the ITL notification it must complete the transaction, either by finalizing it (if no discrepancy was found) or by terminating it (if a discrepancy was found). The registry calls the AcceptNotfication Web service method on the Communications Hub to complete the transaction.

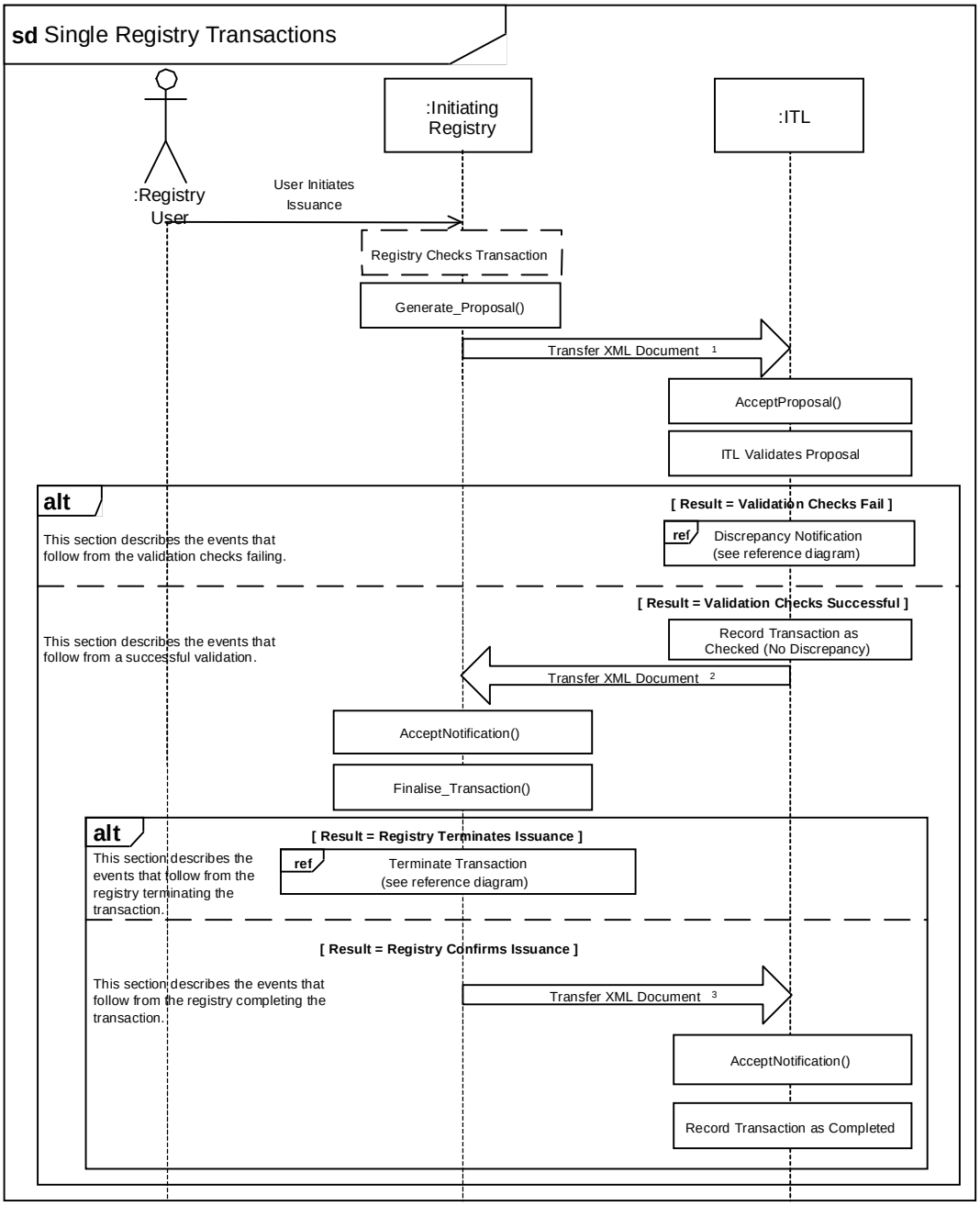
Step 4 – ITL Complete

If the registry requested the ITL to finalize the transaction, the ITL updates its records for the units in the transaction as appropriate for the transaction type. The units are now free to be used in any other transaction. If the registry requested the ITL to terminate the transaction, the ITL will mark the transaction as terminated. The units that had been part of the transaction are now free to be used in another transaction.

The transaction has now been officially recorded at the ITL.

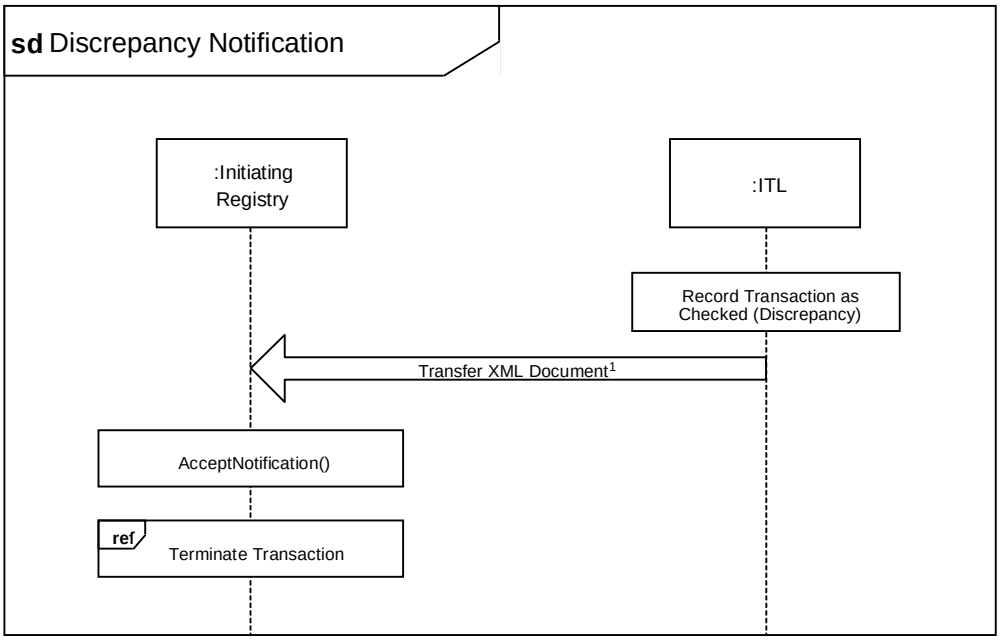
4.3.1 Single Registry Behaviour Diagrams

Figure 4.3: Single Registry Transaction Behaviour Diagram



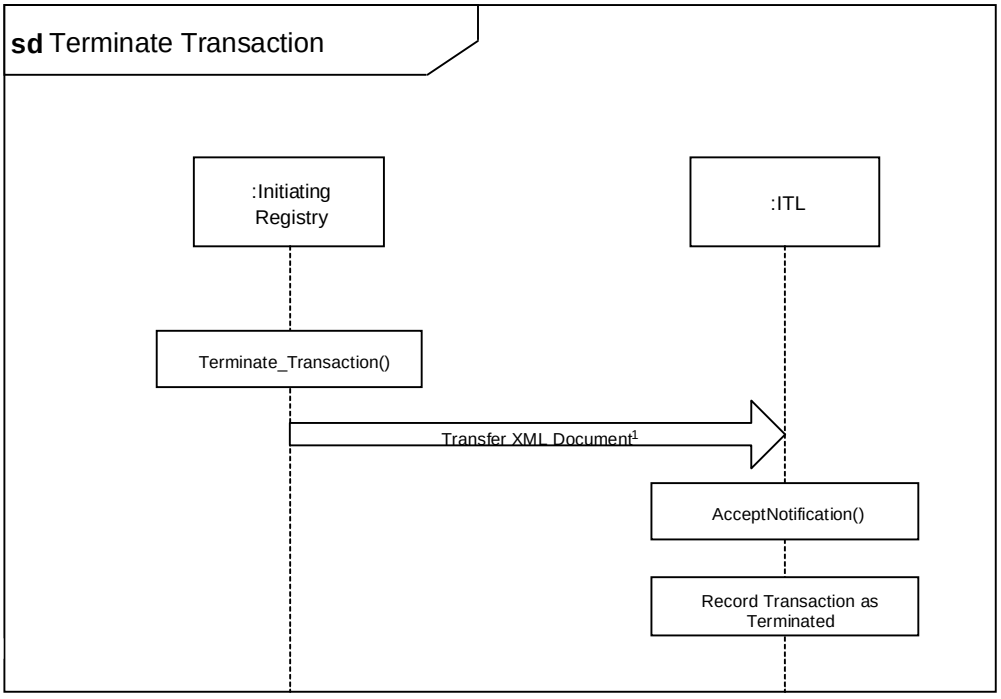
1. Function Generate_Proposal creates an XML document that proposes a transaction and sends the document to the AcceptProposal Web service on the ITL.
2. The ITL creates an XML document to inform the registry that the transaction was successfully validated and sends the document to the AcceptNotification Web service on the registry.
3. Function Finalise_Transaction creates an XML document to inform the ITL that the transaction is complete and sends the document to the AcceptNotification Web service.

761 **Figure 4.4: Discrepancy Notification Sequence Diagram**
762



1. In order to inform the Initiating Registry of a discrepancy the ITL prepares and sends an XML document to the AcceptNotification Web service on the registry.

763
764
765 **Figure 4.5: Terminate Transaction Sequence Diagram**
766



1. The Terminate_Transaction function on the registry creates and sends an XML document to the AcceptNotification Web service on the ITL to inform the ITL that the transaction was terminated.

767

4.3.2 Single Registry Transactions Stage Table

Figure 4.6 describes each stage of the process for single registry transactions. For example, the first row of the table should be read as follows: When the stage is "Proposed," the stage ended with "Message 1" containing the transaction status of "Proposed." The message is sent to the "ITL" and generated by the "registry."

Figure 4.6: Single Registry Stage Table

Stage	Stage Name	Stage Ends With	Transaction Status	Sent To	Generated By
P	Proposed	Message 1	Proposed	ITL	Registry
TR	ITL Review	Message 2	Checked (Discrepancy) or Checked (No Discrepancy)	Registry	ITL
RC	Registry Complete	Message 3	Completed or Terminated	ITL	Registry
TC	ITL Complete	No Message	Completed or Terminated	--	--

4.4 Multiple Registry Transaction Model

The multiple registry model for transactions applies to External Transactions in which units are transferred to a different registry. The initial steps are similar to the single registry model, but require the additional step of forwarding the proposal to the Acquiring Registry. The following steps describe the sequence of messages necessary to complete an external transfer:

Step 1 - Proposal

The registry sends a proposal for a transaction to the ITL using the AcceptProposal Web service method on the Communications Hub. The proposal contains the transaction type, the units involved in the transaction, and, if appropriate, the transferring and acquiring account types.

Step 2 – ITL Review

The Communications Hub receives the proposal and, once the incoming message is verified to be well formed and authentic, it places the message in a queue for processing. Messages are processed from the queue in the order received. The ITL validates the transaction against the business rules for external transactions. If the transaction meets all requirements, the ITL records the transaction as pending and marks the units involved in the transaction as unavailable to any other transaction. It then forwards the proposal to the acquiring registry.

If a discrepancy is found, the ITL will notify the Initiating Registry of the requirement(s) the transaction proposal did not meet. The ITL will also notify the acquiring registry that the transaction was not completed. The units involved in the transaction cannot be used in another transaction until the Initiating Registry sends a termination request.

Step 3 – Registry Review

The ITL forwards the transaction proposal to the Acquiring Registry by calling the AcceptProposal Web service method on the Acquiring Registry. The Acquiring Registry evaluates the proposal and either accepts or rejects it. In either case, the Acquiring Registry calls the AcceptNotification Web service method on the ITL to inform the ITL of its evaluation result.

Step 4 – ITL Relay

The ITL updates the transaction status with the result of the Acquiring Registry evaluation and notification and forwards the evaluation result to the initiating registry. The ITL calls the AcceptNotification Web service method on the Initiating Registry.

Step 5 – Registry Complete

The registry completes the transaction. The registry calls the AcceptNotification Web service method on the ITL to inform the ITL when it has finished updating its records.

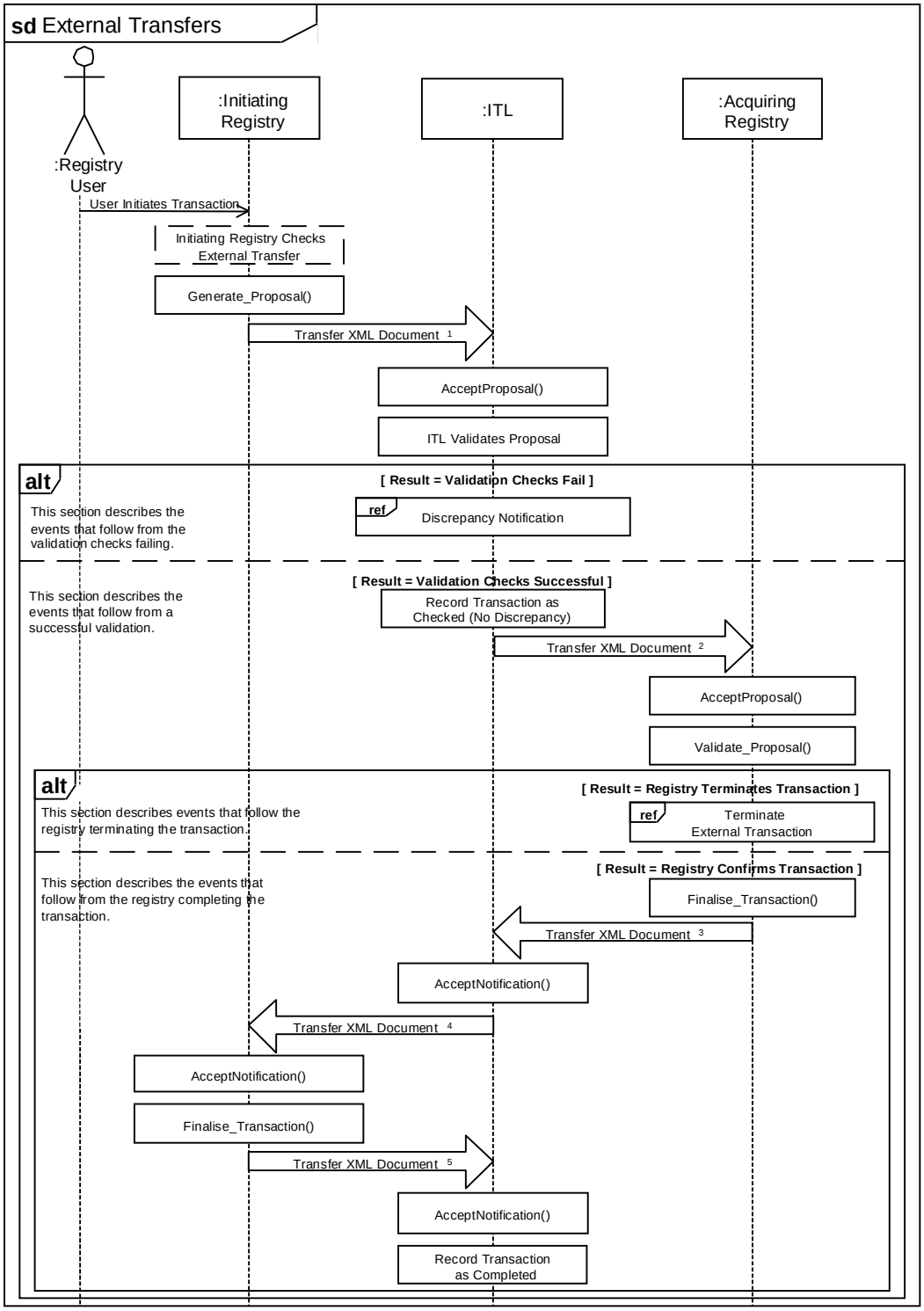
Step 6 – ITL Complete

The ITL completes the transaction. The ITL updates its records for the units in the transaction. The units are now free to be used in any other transaction. The units that had been part of the transaction are now free to be used in another transaction.

The transaction has now been officially recorded at the ITL.

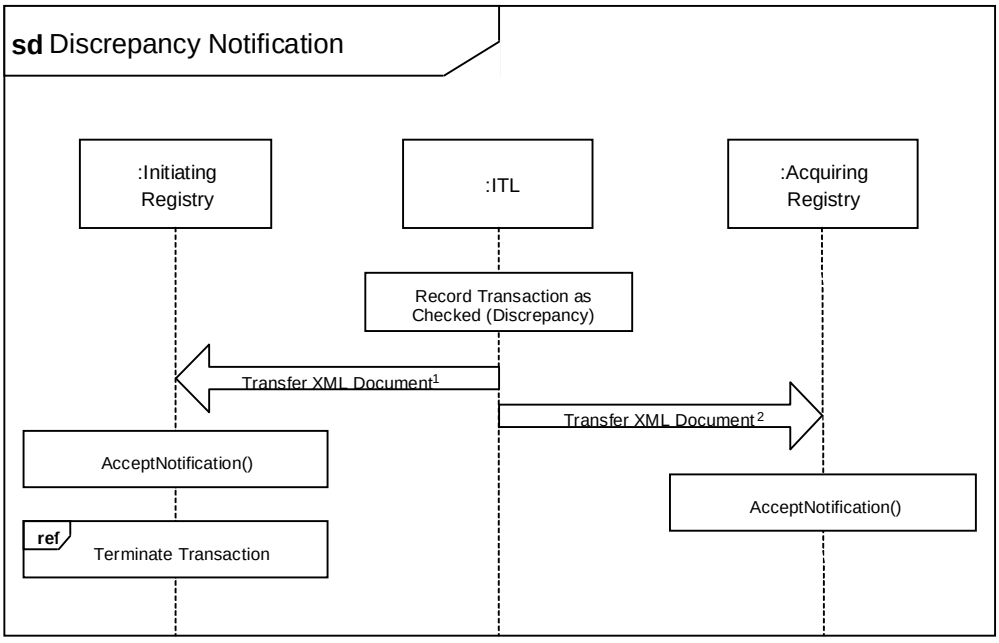
4.4.1 UML Behaviour Diagram for Multiple Registry Transactions

Figure 4.7: External Transfer Behaviour Diagram



1. Function Generate_Proposal creates an XML document that proposes a transfer and sends the document to the AcceptProposal Web service on the ITL.
2. The ITL creates an XML document to pass the proposed transfer on to the AcceptProposal Web service on the Acquiring Registry.
3. Function Finalise_Transaction on the Acquiring Registry creates an XML document to inform AcceptNotification on the ITL that transfer was confirmed.
4. AcceptNotification on the ITL creates an XML document to inform the AcceptNotification Web service on the initiating registry that the Acquiring Registry confirmed the transfer.
5. Function Finalise_Transaction on the Initiating Registry creates an XML document to inform the ITL through the AcceptNotification Web service that transfer was confirmed.

845 **Figure 4.8: Discrepancy Notification Sequence Diagram**
846

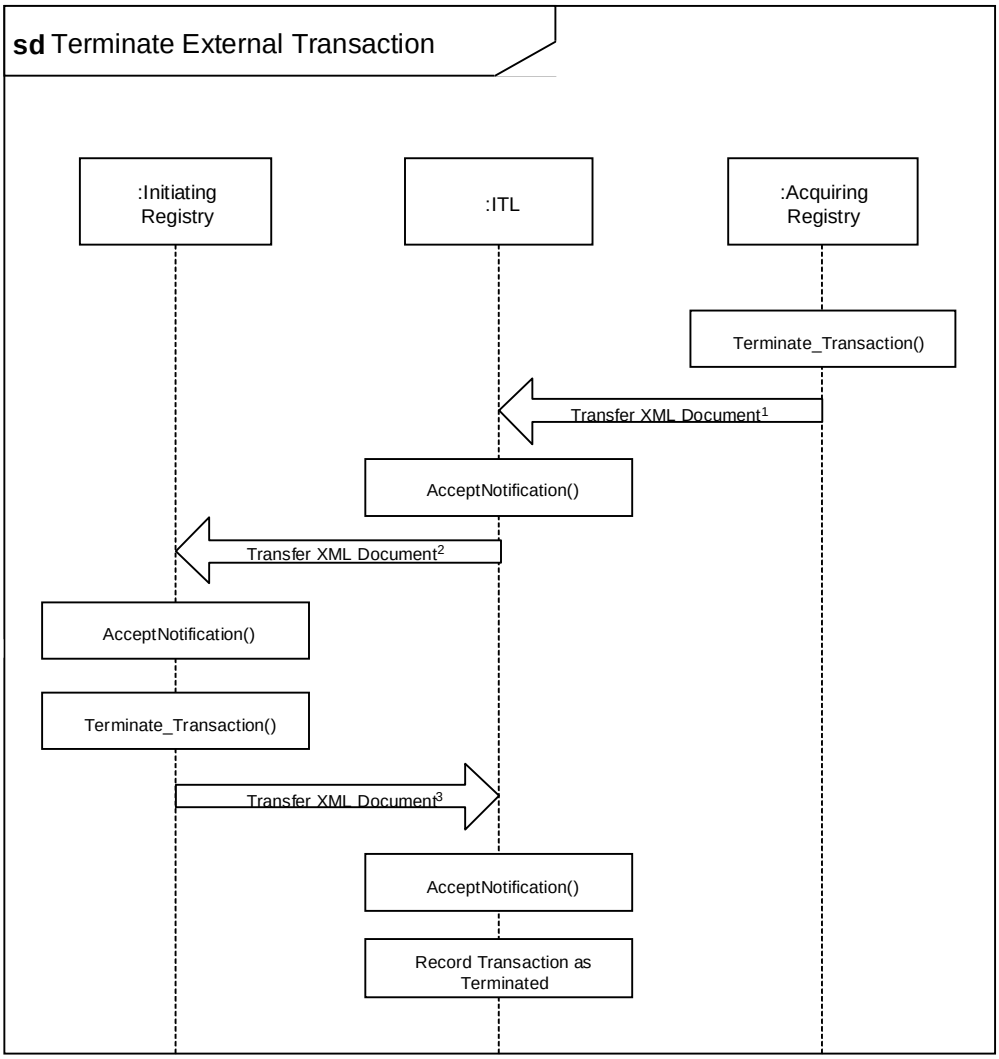


1. In order to inform the Initiating Registry of a discrepancy the ITL prepares and sends an XML document to the AcceptNotification Web service on the registry.

2. In order to inform the Acquiring Registry of a discrepancy, the ITL prepares and sends an XML document to the AcceptNotification Web service on the Acquiring Registry.

847
848

849 **Figure 4.9: Terminate External Transaction Sequence Diagram**
850



1. The Terminate_Transaction function on the Acquiring Registry sends an XML document to the AcceptNotification Web service on the ITL to inform the ITL of the terminated transaction.

2. The ITL sends an XML document to the AcceptNotification Web service on the Initiating Registry to inform the registry that that Acquiring Registry terminated the transaction.

3. The Terminate_Transaction function on the Initiating Registry creates and sends an XML document to the AcceptNotification Web service on the ITL to inform the ITL that the Initiating Registry terminated the transaction.

851

4.4.2 Stage Table for Multiple Registry Transactions

The stage table for Multiple Registry Transactions (such as External Transfer) contains three different scenarios. The stages for scenario three do not include the acquiring registry.

This table describes each stage of the process. For example, the first row of the table should be read as follows: When the stage is "Proposed," the stage ended with "Message 1" containing the transaction status of "Proposed." The message is sent to the "ITL" and generated by the "Transferring Registry."

Figure 4.10: External Transfer Stage Table

Scenario #1	Stage	Stage Name	Stage Ends With	Transaction Status	Sent To	Generated By
Checks (No Discrepancy) Acquiring Registry Accepts	**P**	Proposal	Message 1	Proposed	ITL	Transferring Registry
	TR	ITL Review	Message 2	Checked (No Discrepancy)	Acquiring Registry	ITL
	RR	Registry Review	Message 3	Accepted	ITL	Acquiring Registry
	TA	ITL Accepted	Message 4	Accepted	Transferring Registry	ITL
	RC	Registry Complete	Message 5	Completed	ITL	Transferring Registry
	TC	ITL Complete	No message	Completed	--	--
Scenario #2	**Stage**	**Stage Name**	**Stage Ends With**	**Transaction Status**	**Sent To**	**Generated By**
Checks (No Discrepancy) Acquiring Registry Terminates	**P**	Proposal	Message 1	Proposed	ITL	Transferring Registry
	TR	ITL Review	Message 2	Checked (No Discrepancy)	Acquiring Registry	ITL
	RR	Registry Review	Message 3	Rejected	ITL	Acquiring Registry
	TA	ITL Accepted	Message 4	Rejected	Transferring Registry	ITL
	RC	Registry Complete	Message 5	Terminated	ITL	Transferring Registry
	TC	ITL Complete	No message	Terminated	--	--
Scenario #3	**Stage**	**Stage Name**	**Stage Ends With**	**Transaction Status**	**Sent To**	**Generated By**
Checks (Discrepancy)	**P**	Proposal	Message 1	Proposed	ITL	Transferring Registry
	TR	ITL Review	Message 2	Checked (Discrepancy)	Transferring Registry	ITL
	RC	Registry Complete	Message 3	Terminated	ITL	Transferring Registry
	TC	ITL Complete	No message	Terminated	--	--

4.5 List of Functions for Transaction Data Exchange

4.5.1 Registry Web Services and Functions

In order to participate in data exchange with the ITL, registries must precisely implement Web services that the ITL can use to send it information. The following table shows the Web services methods registries are required to expose for unit transactions. Detailed technical information about the specifications for these Web service methods are in Annex B.

Figure 4.11: Registry Public Web Service Methods

Public Web Service Method	Page
AcceptNotification	B-5
AcceptProposal	B-6

In addition to the above Web service methods that the registry must precisely implement so that they may be used by the ITL, the registry must have capabilities to build transactions, validate transactions, and log transactions. The following functions implement those responsibilities. Note that these functions are not exposed to the public, so they provide more flexibility in how they are implemented.

Figure 4.12: Registry Internal Functions

Private Function	Page
Check_Version	B-7
Data_Integrity_Checks	B-8
Finalise_Transaction	B-9
Generate_Proposal	B-10
Preliminary_Checks	B-11
Update_Units	B-12
Validate_Proposal	B-13
Write_To_File	B-14
Write_To_Message_Log	B-15
Write_Transaction	B-16
Write_Transaction_Block	B-17
Write_Transaction_Status	B-18

4.5.2 ITL Web Services and Functions

Like the registries, the ITL must precisely implement public Web services called AcceptNotification and AcceptProposal to be used by registries in data exchange. Detailed technical information about the specifications for these Web service methods are in Annex E to the ITL Technical Specifications and Annexes I and K to this document.

The ITL also contains extensive functionality for checking and logging data. Detailed technical information about the specifications for these Web service methods are in Annex F of the ITL Technical Specifications and in Annex K to this document.

4.6 Validation Checks and Response Codes

The ITL executes numerous checks on all transactions to assure the authenticity of a message, the format of the message, the sequence of the message, and the validity of the unit transaction. The following categories of checks are performed on the ITL. The list of specific checks and associated response codes is included in Annex E.

It is recommended that registries implement similar checks to reduce the number of discrepancies identified by the ITL.

4.6.1 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. 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 errors. Failures due to transaction checks are returned in the ResponseObject in an HTTP SOAP response initiated by the ITL to the originating registry.

4.6.2 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 such time in which they can be processed. These checks determine whether the message from the queue is still viable and can be processed.

4.6.3 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.

4.6.4 Data Integrity Checks for Transactions

This category of checks is performed by the ITL's Data_Integrity_Checks function to identify whether incoming messages contain data that do not meet 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 ITL's Transaction Log table and is not processed further. All data integrity checks are critical checks; if they result in failure, no further checks are processed.

4.6.5 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.

4.6.6 General Transaction Checks

The ITL performs this category of checks for all transaction messages involving unit blocks.

4.6.7 Transaction-specific Checks

The ITL performs this category of checks on all Kyoto transactions for the specified transaction types.

5. Reconciliation Process

5.1 Reconciliation Process Flow

The data on unit holdings in registries and the ITL are reconciled on a periodic basis on the basis of a data snapshot at a specified time. The snapshot taken should treat proposed transactions (in any status prior to "Completed") as if the transaction had not yet occurred. All unit type and account types for unit blocks held by the registry should be totalled for purposes of reconciliation as if they had not been changed or transferred by any ongoing transactions. This approach is necessary to ensure consistency of totals and unit blocks with the ITL, which will not commit changes in ownership or unit block types (or other attributes) until the message with the transaction status of "Completed" is received from the Initiating Registry. It is recommended that registries delay committing database transactions for proposed transactions and sending the messages for "Completed" transactions for a short period of time surrounding the reconciliation snapshot date and time. The amount of time recommended will vary based on message processing time and is within the discretion of the Registry Manager. The ITL will not change the processing of messages to avoid possible inconsistency. A prolonged period of registry non-operation or suspension of transactions for reconciliation purposes is not foreseen.

A reconciliation action is completed when no inconsistencies are discovered or when any discovered inconsistencies have been resolved. The reconciliation process is implemented in three stages in which three types of data are requested:

- Stage 1 – Validate Account Totals
- Stage 2 – Validate Unit Blocks
- Stage 3 – Review Audit Logs

Stage 1 – Validate Account Totals

1. The ITL requests unit holding totals by account type and unit type from registry. To do so the ITL calls the ProvideTotals Web service method on the registry.

2. The registry receives request from ITL, snapshots the data at the designated time, compiles the totals, and calls ReceiveTotals Web service on the ITL.

3. ITL performs preliminary checks on the message and adds the message to the processing queue. When the ITL removes the message from the queue it performs registry validation, reconciliation data integrity and message sequence checks. Failure results in rejection of message without data recording or further processing.

4. If all the checks are passed, the ITL compares the totals sent by the registry with its own records and determines a result.

5. The ITL records the new status of the reconciliation action. If the status is "Validated" (2), the ITL checks if the party is in a supplementary program. If it is, the ITL initiates STL Reconciliation processing. If the party is not in a supplementary program, the ITL sends notification to the registry that the reconciliation completed successfully. The ITL also removes the freeze flag from any units that remain flagged from a previous reconciliation action at that registry. If the new status is "Inconsistent Totals" (3), the ITL requests the registry to send unit block details.

Stage 2 – Validate Unit Blocks

1. The ITL calls the ProvideUnitBlocks Web service method on the registry to request that the registry send unit blocks. This request may be limited to unit blocks for a specific unit type-account type combination that failed the totals check.

2. The registry sends its unit block inventory to the ITL by calling the ReceiveUnitBlocks Web service method.

3. ITL performs preliminary checks on the message and adds the message to the processing queue. When the ITL removes the message from the queue it performs registry validation, reconciliation data integrity and message sequence checks. Failure results in rejection of message without data recording or further processing.

4. If all the checks are passed, the ITL compares each unit block sent by the registry against the ITL records. If blocks do not match, they are marked as inconsistent.

5. If no inconsistent blocks are found, a manual intervention is triggered to explain why the first step of reconciliation failed. Otherwise, the ITL requests the registry to send a transaction history since the last reconciliation for each inconsistent block.

Stage 3 – Review Audit Logs

1. The ITL calls the ProvideAuditTrail Web service method on the registry to request the audit trail for each inconsistent block.

2. The registry sends the audit trail to the ITL by calling the ReceiveAuditTrail Web service method.

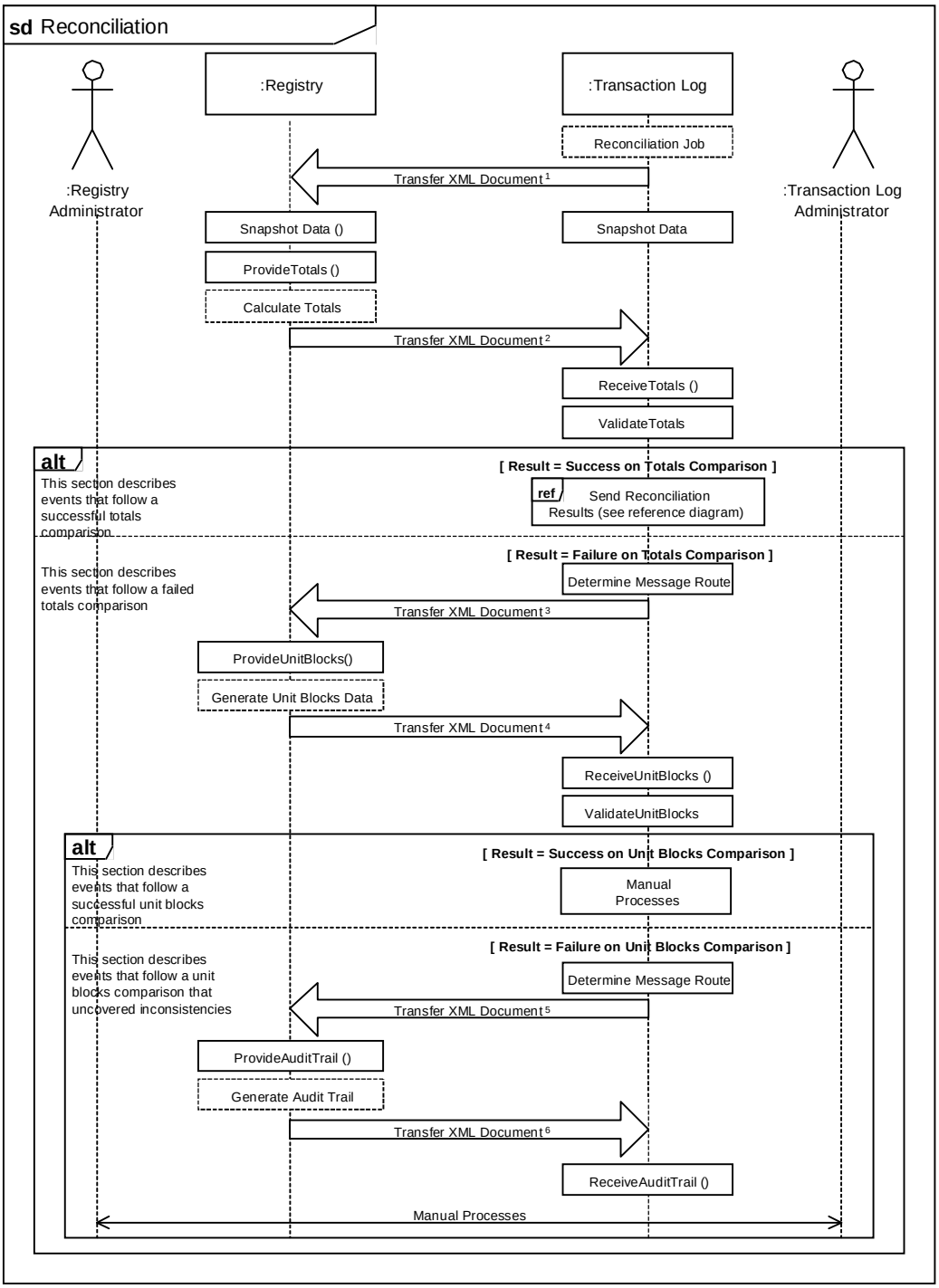
3. The ITL performs preliminary checks on the message and adds the message to the processing queue. When the ITL removes the message from the queue it performs registry validation, data integrity and message sequence checks. Failure results in rejection of message without data recording or further processing.

4. If all the checks are passed the ITL writes the audit trail to a flat file and stores the location in the ITL's Message Log table.

5. The ITL and registry administrators research and correct the cause of the inconsistency. Once corrected a new reconciliation is immediately initiated by the ITL administrator.

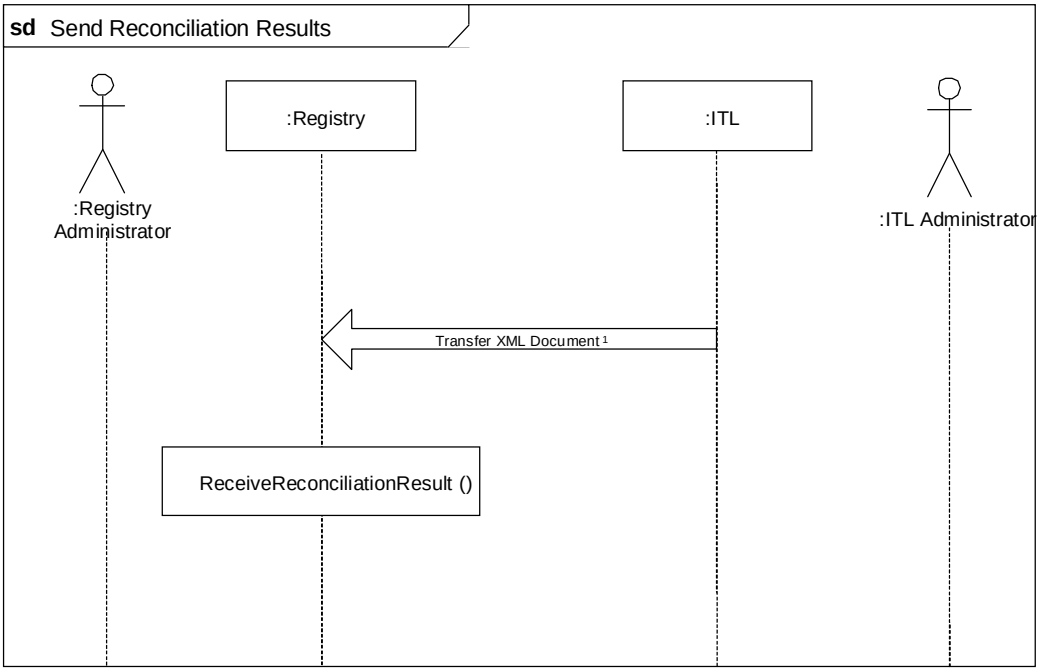
5.2 Reconciliation Behaviour Diagrams

Figure 5.1: Reconciliation Behaviour Diagram



1. The ITL creates an XML document and sends it to the ProvideTotals Web service on the registry.
2. The registry sends an XML document to the ReceiveTotals Web service on the ITL.
3. The ITL sends an XML document to the ProvideUnitBlocks Web service on the registry.
4. The registry sends an XML document to the ReceiveUnitBlocks Web service on the ITL.
5. The ITL sends an XML document to the ProvideAuditTrail Web service on the registry.
6. The registry sends an XML document to the ReceiveAuditTrail Web service on the ITL.

Figure 5.2: Send Reconciliation Results Behaviour Diagram



1. The UpdateReconciliation function on the ITL sends an XML document to the ReceiveReconciliationResult Web service on the registry to inform the registry that Reconciliation is complete.

5.3 Reconciliation Stage Tables

The Stage Table represents the sequence of events in terms of the "stage" and its relation to the "status" of a reconciliation action. The stage defines where in the process of information exchange a particular message or evaluation occurs. A stage ends and a new stage begins when a message has been successfully transmitted and received by either a registry or the ITL or when the last step of a process occurs. The order in which each defined stage occurs may vary based on the specific process and based on the results of the ITL validation process. The numbers assigned to stages should not be used as an indicator of acceptable stage sequences.

Figure 5.3: Reconciliation Stage 1 - Validate Account Totals

Stage 1	Stage Name	Stage Ends With	Reconciliation Status	Sent To	Generated From
Supplementary Program, Validated Totals at ITL and STL	Request	Message 1	"Initiated"	Registry	ITL
	Totals Sent	Message 2	"Initiated"	ITL	Registry
	Totals Evaluated	Message 3	"Validated"	Registry	ITL
	Totals By Account Sent	Message 4	"Validated"	ITL	Registry
	Totals By Account Sent	Message 5	"Validated"	STL	ITL
	Totals Evaluated by STL	Message 6	"STL Validated"	ITL	STL
	Totals Evaluated by STL	Message 7	"STL Validated"	Registry	ITL
	Reconciliation Complete	No message	"STL Validated"	--	--

Figure 5.4: Reconciliation Stage 2 - Validate Unit Blocks

Stage 2	Stage Name	Stage Ends With	Reconciliation Status	Sent To	Generated From
Supplementary Program, Inconsistent Totals at ITL	Request	Message 1	"Initiated"	Registry	ITL
	Totals Sent	Message 2	"Initiated"	ITL	Registry
	Totals Evaluated	Message 3	"Totals Inconsistent"	Registry	ITL
	Unit Blocks Sent	Message 4	"Totals Inconsistent"	ITL	Registry
	Unit Blocks Evaluated	Message 5	"Unit Blocks Inconsistent"	Registry	ITL
	Audit Trail Sent	Message 6	"Unit Blocks Inconsistent"	ITL	Registry
	Manual Intervention	Message 7	"Complete with Manual Intervention"	Registry	ITL
	Reconciliation Complete	No Message	"Complete with Manual Intervention"	--	--
	New Reconciliation action will be initiated by ITL				

Figure 5.5: Reconciliation Stage 3 - Review Audit Logs

Stage 3	Stage Name	Stage Ends With	Reconciliation Status	Sent To	Generated From
Supplementary Program, Validated Totals at ITL, Inconsistent Totals at STL	Request	Message 1	"Initiated"	Registry	ITL
	Totals Sent	Message 2	"Initiated"	ITL	Registry
	Totals Evaluated	Message 3	"Validated"	Registry	ITL
	Totals By Account Sent	Message 4	"Validated"	ITL	Registry
	Totals By Account Sent Relay	Message 5	"Validated"	STL	ITL
	Totals Evaluated By STL	Message 6	"STL Totals Inconsistent"	ITL	STL
	Totals Evaluated By STL Relay	Message 7	"STL Totals Inconsistent"	Registry	ITL
	Unit Blocks Sent to STL	Message 8	"STL Totals Inconsistent"	ITL	Registry
	Unit Blocks Sent to STL relay	Message 9	"STL Totals Inconsistent"	STL	ITL
	STL Unit Blocks Inconsistent	Message 10	"STL Unit Blocks Inconsistent"	ITL	STL
	STL Unit Blocks Inconsistent Relay	Message 11	"STL Unit Blocks Inconsistent"	Registry	ITL
	Audit Trail Sent to STL	Message 12	"STL Unit Blocks Inconsistent"	ITL	Registry
	Audit Trail Sent to STL Relay	Message 13	"STL Unit Blocks Inconsistent"	STL	ITL
	Manual Intervention	Message 14	"STL Complete with Manual Intervention"	ITL	STL
	Manual Intervention	Message 15	"STL Complete with Manual Intervention"	Registry	ITL
	Reconciliation Complete	No Message	"STL Complete with Manual Intervention"	--	--
	New reconciliation action will be requested by STL				

5.4 List of Functions for Reconciliation Process

5.4.1 Registry Functions

In order to participate in reconciliation with the ITL, registries must precisely implement Web services that the ITL can use to send it information. The following table shows the Web services methods registries are required to expose for reconciliation. Detailed technical information about the specifications for these Web service methods are in Annex C.

Figure 5.6: Registry Public Web Service Methods

Public Web Service Method	Page
ProvideAuditTrail	C-7
ProvideTotals	C-7
ProvideUnitBlocks	C-8
ReceiveReconciliaitonResult	C-9

In addition to the above Web service methods that the registry must precisely implement so that they may be used by the ITL, the registry must have additional capabilities to build transactions, validate transactions, and log transactions. The following functions implement those responsibilities. Note that these functions are not exposed to the public, so they provide more flexibility in how they are implemented.

Figure 5.7: Registry Internal Functions

Private Function	Page
Close_Reconciliation_Action	C-4
Start Reconciliation	C-10
Write_To_Reconciliation_Log	C-11
Write_To_Reconciliation_Status	C-12

5.4.2 ITL Functions

Like the registries, the ITL must precisely implement public Web services called ReceiveTotals, ReceiveUnitBlocks, and ReceiveAuditTrail to be used by registries in data exchange. The following tables list the public Web service methods provided by the ITL for reconciliation. Detailed technical information about the specifications for these Web services can be found in the ITL.

The ITL also contains functionality for recording data. Detailed technical information about the specifications for these functions can be found in the ITL Technical Specification.

5.5 Reconciliation Checks and Responses

When the ITL receives reconciliation messages from registries in response to its reconciliation request, the following types of checks are performed on the messages.

Figure 5.8: 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 ITL 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 ITL 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 ITL Reconciliation Log table.
Other Reconciliation Checks and Messages	6400 - 6500	Basic reconciliation checks.	Message returned with response codes and transaction status. Message logged in ITL Reconciliation Log table.

5.5.1 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.

5.5.2 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.

Figure 5.9: 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.
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.
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 5.9: Additional Registry Checks for Reconciliation (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.
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.
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."
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.

1149
1150

5.5.3 Data Integrity Checks for Reconciliation

This category of checks is performed by the ITL to identify whether incoming messages contain data not meeting 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 and is not processed further. Data integrity checks are critical checks and if they result in failure, no further checks are 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 5.10: 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 a valid code.
6204	Reconciliation Snapshot DateTime	Reconciliation Snapshot DateTime must be between 10/1/04 and current date + 30 days.
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.

5.5.4 Message Sequence Checks for Reconciliation Messages Received from Registries

Figure 5.11: 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 snapshot datetime of the reconciliation datetime.

5.5.5 Other Reconciliation Checks and Messages

The following messages may be generated by the ITL as part of the reconciliation analysis.

Figure 5.12: 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.
6600	Successful Reconciliation of Totals	A reconciliation action has been successful and is complete.

6. ITL Administrative Functions

The ITL has four types of administrative Web service functions that involve registries:

- Notifications. The ITL sends Notifications to the AcceptMessage Web service on the registry. This is a simple message, for which no specific response is required. The messages may require a registry to submit transactions described in Section 4.

- General Messages. The ITL may also send general messages to a registry through the AcceptMessage Web service.

- Transaction Status Information. The registry may at any time submit a request to the GetTransactionStatus Web service on the ITL Communications Hub to provide the status of a specific transaction. This is a synchronous communication and the ITL provides the current status of the transaction as an immediate response.

- Time Synchronization. The ITL may at any time submit a request to the Provide Time Web service at a Registry to provide the time. This is a synchronous communication and the registry must provide the time as an immediate response.

6.1 Notifications

The ITL performs these administrative functions on either a periodic basis or as initiated by the ITL Administrator to evaluate data and inform the registries of specific actions required. Each of these functions may result in a notification to one or more registries regarding actions that must be taken by a registry or the status of an action about which the ITL has previously informed the registry. Each of these notifications are associated with a Notification Type code defined in Annex G.

These administrative functions include:

- Twenty-four Hour Transaction Clean-up. This function cancels transactions which are more than 24 hours old and requests that registries terminate the transaction. (Notification Code 1)

- Outstanding Units at end of Commitment Period. This function identifies units which have not been cancelled or carried-over and requests that the registry cancel or carry them over. (Notification Code 2)

- tCER or lCER Expiration. This function identifies tCERs and lCERs that will expire within 30 days and notifies the registry. (Notification codes 9 and 10)

- lCER Transfer Ineligibility Due to Lack of Certification Report. This function undertakes actions for Projects, for which no certification report has been submitted, by making lCERs from the Project ineligible for transfer and informing registries of the requirement to replace these lCERs. (Notification Codes 3, 4, 11)

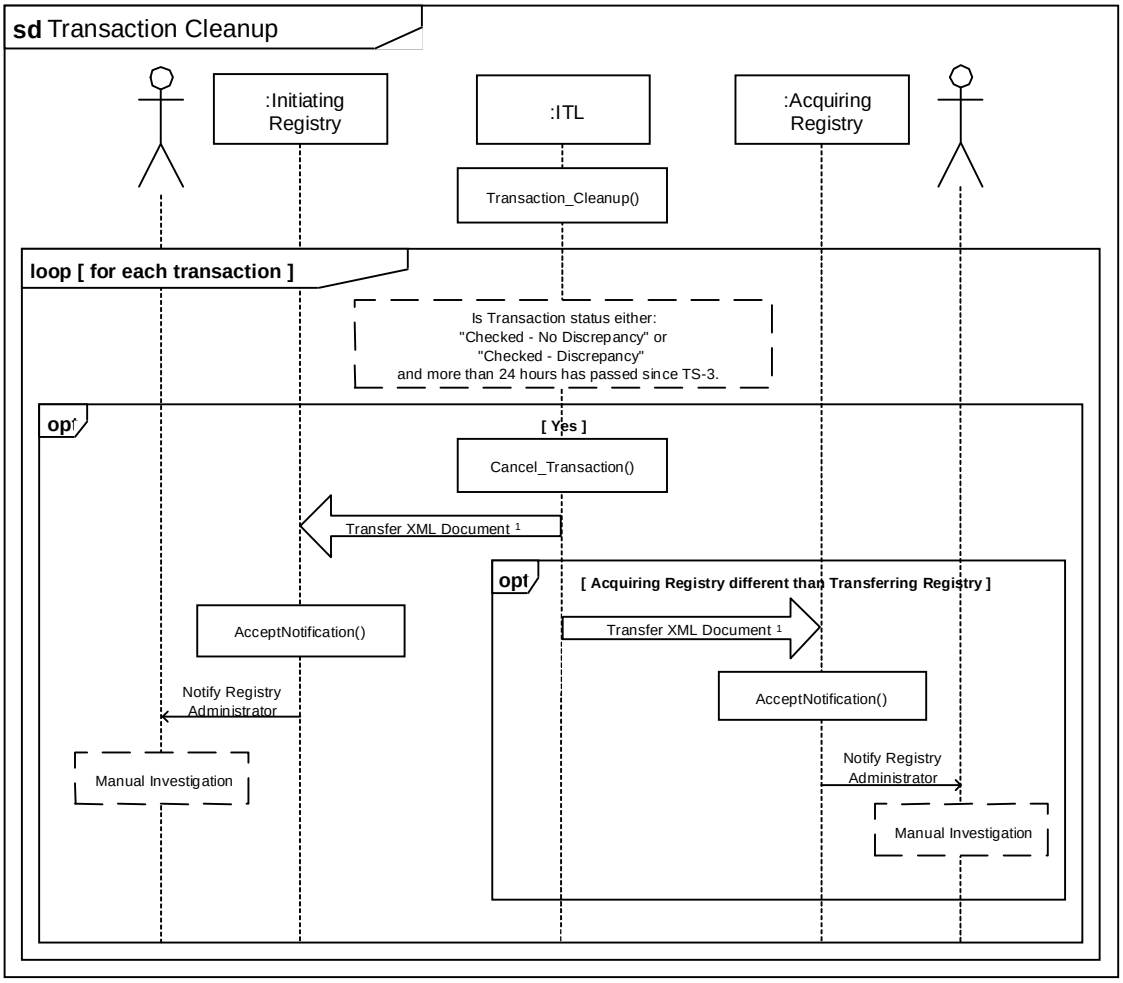
- lCER Transfer Ineligibility Due to Reversal of Storage. This function undertakes actions for Projects, for which a reversal in storage has occurred, by making lCERs from the Project ineligible for transfer and informing registries about the quantity of lCERs which must be replaced. (Notification Code 7)

- Restoration of lCER Transfer Eligibility. This function tabulates the units which have been replaced due to a reversal of storage action, informs the registry of the status, and makes the remaining units for that registry eligible again for transfer when the registry has completed replacement. (Notification Codes 8 and 12)

6.1.1 Transaction Clean-up

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. 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.

Figure 6.1: Transaction Clean-up Diagram

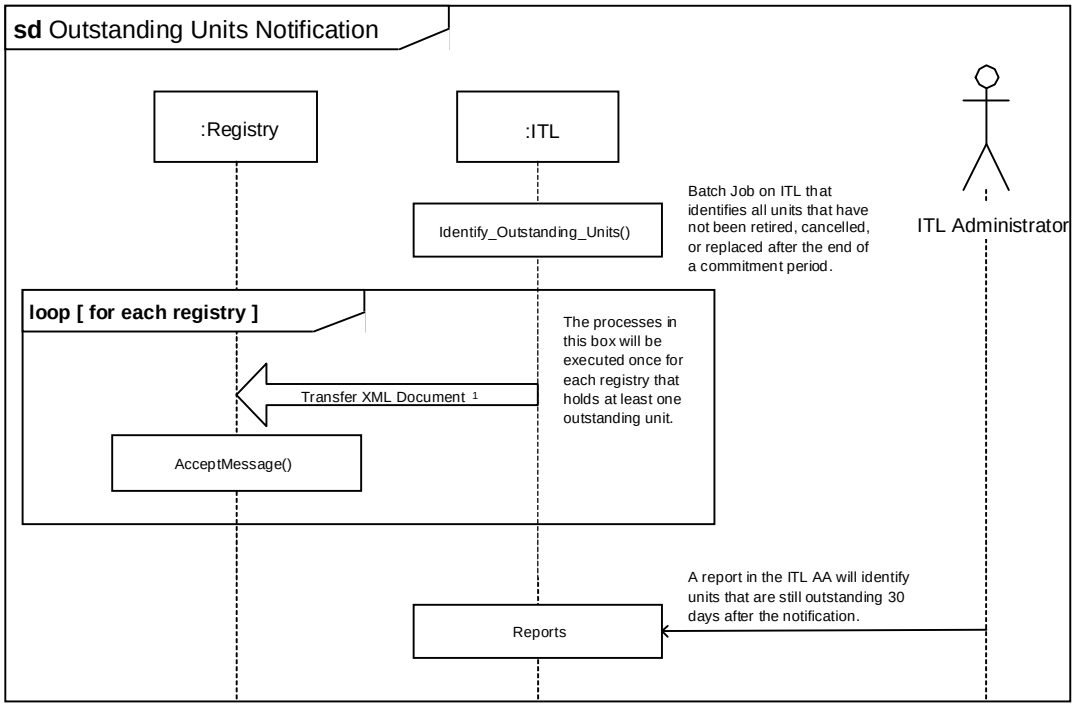


1. The CancelTransaction function on the Transaction Log creates and sends an XML document to the AcceptNotification Web service on the registry to inform the registry that the transaction has been cancelled.

6.1.2 Outstanding Units at the end of Commitment Period

After the end of a commitment period, an ITL process identifies all units for that Commitment Period that have not been retired, cancelled, or carried-over at a time determined by the ITL Administrator. The ITL notifies the registry through the AcceptMessage web service and indicates that action must be taken on these units. Registries must cancel or carry-over these units within 30 days.

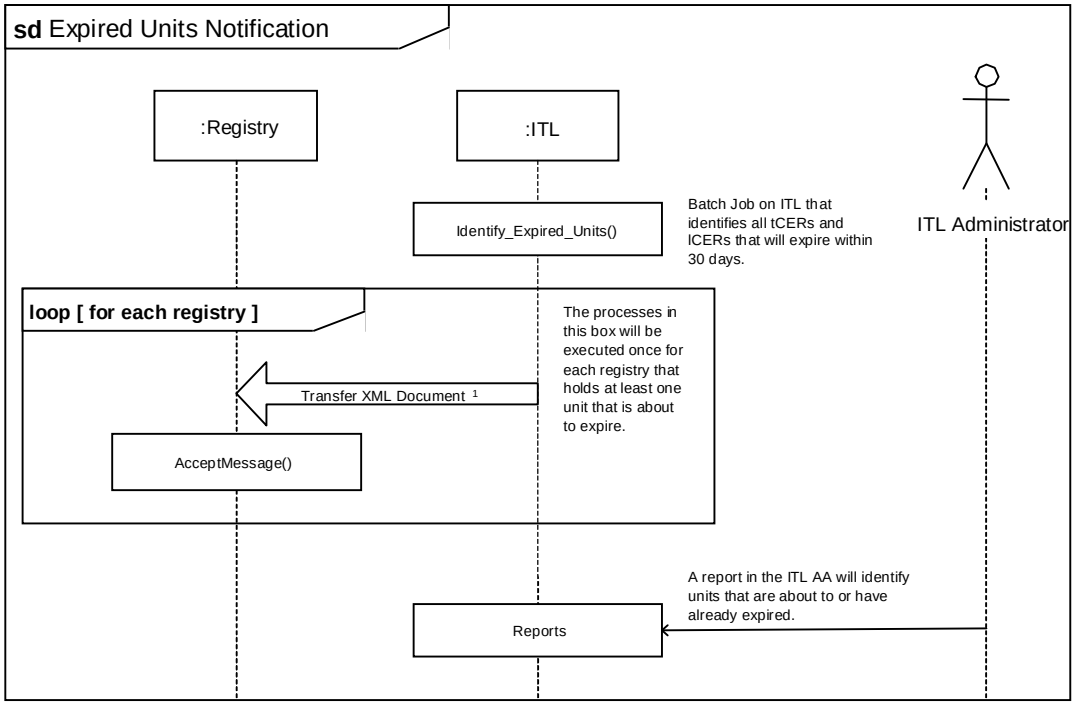
1263 **Figure 6.2: Outstanding Units Notification**
1264



1. The function prepares and sends an XML document to the AcceptMessage Web service on the registry.

1265
1266
1267 ### 6.1.3 Expired Units
1268
1269 The ITL will notify the holder of any tCER or lCER unit when that unit is about to expire. The
1270 ITL will send notification 30 days before the expiration date. The registry must replace or
1271 cancel the units by the expiration date.
1272

Figure 6.3: Expired Units Notification

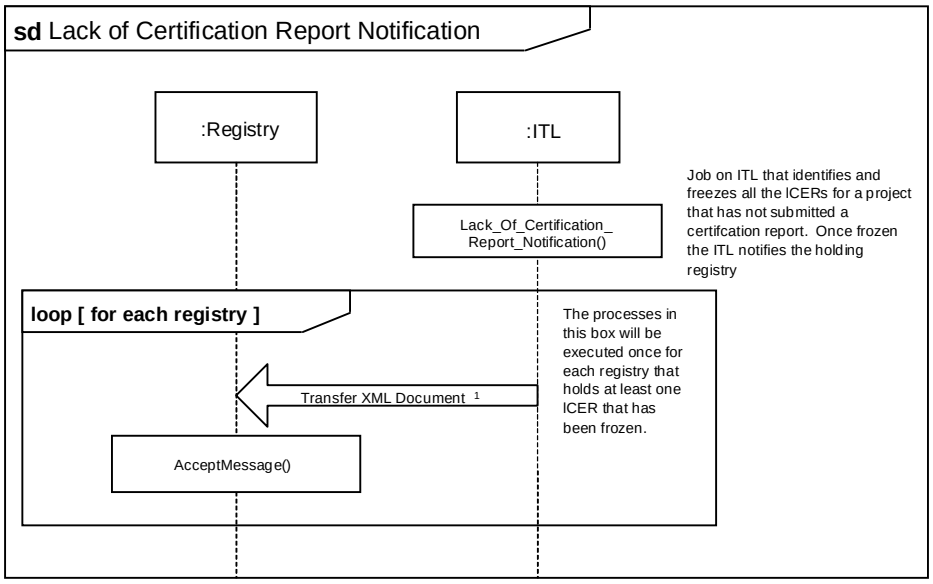


1. The function prepares and sends an XML document to the AcceptMessage Web service on the registry.

6.1.4 Lack of Certification Report

If the persons responsible for a project have not submitted a certification report, the ITL may elect to halt transfers (except cancellation) of all units associated with that project. If this occurs, the ITL will notify all registries holding affected units through the AcceptMessage Web service method at the registry. 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.

Figure 6.4: Lack of Certification Report Notification

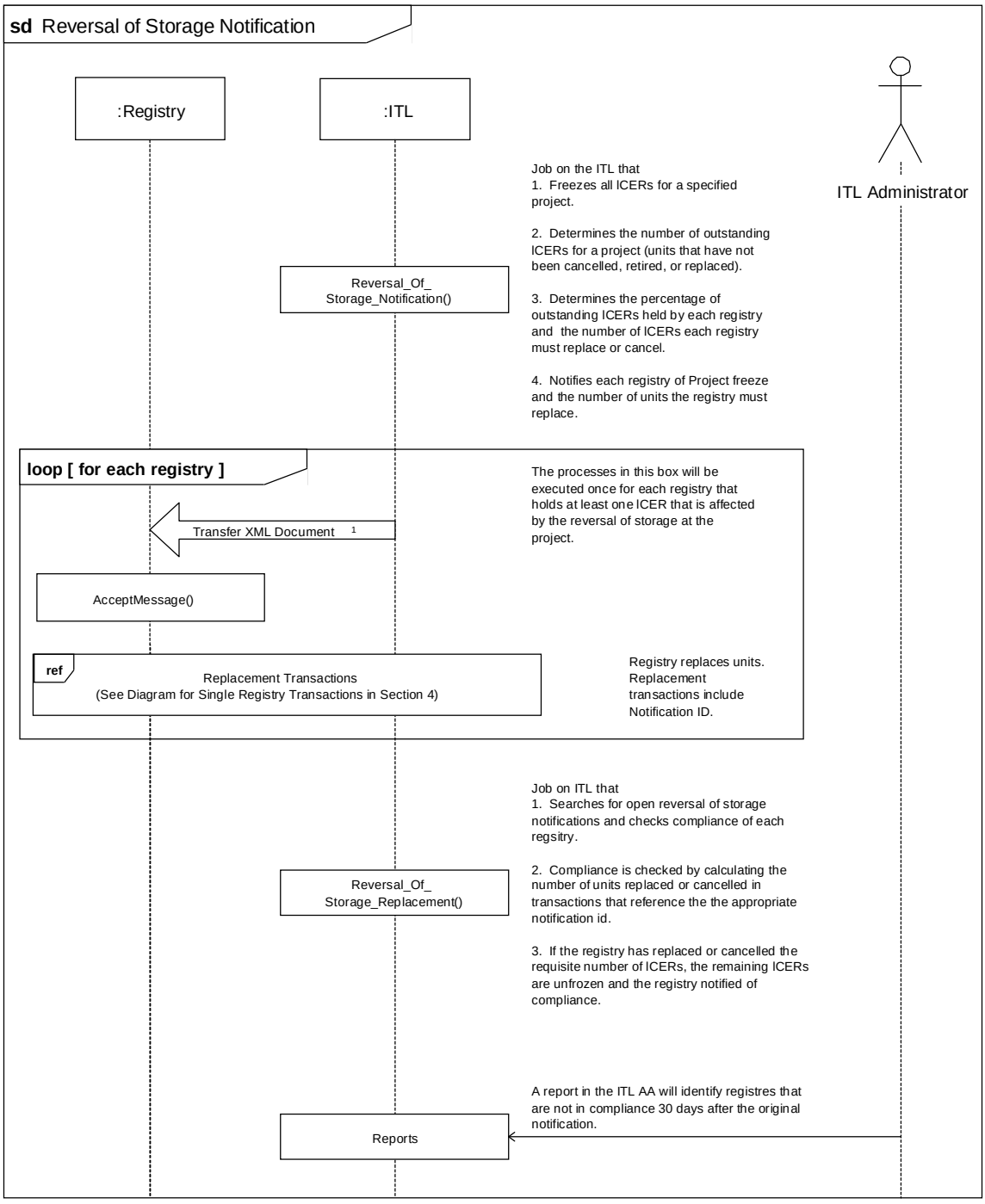


1. The function prepares and sends an XML document to the AcceptMessage Web service on the registry.

6.1.5 Reversal of Storage for Project

If a reversal in the storage of greenhouse gasses occurs at a project, the ITL will temporarily suspend trading of all units associated with the project. The ITL will 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 over total holdings of ICERs from the Project in all registries. The ITL will notify each affected registry through the AcceptMessage web service method. The message will alert each registry to the number of units it must replace. 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.

Figure 6.5: Reversal of Storage Notification



1. The function prepares and sends an XML document to the AcceptMessage Web service on the registry.

6.2 General Messages

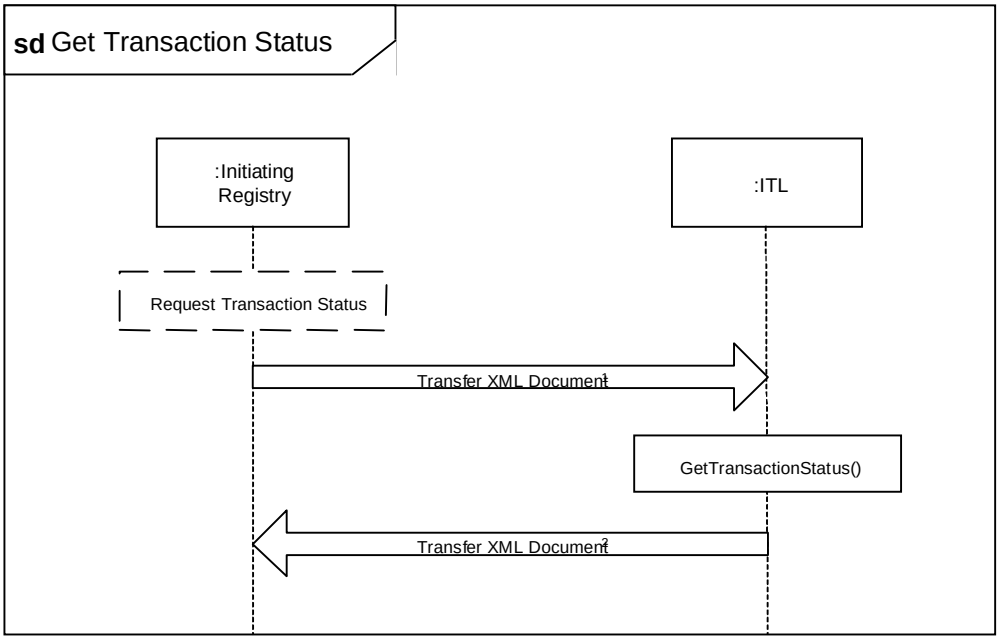
The AcceptMessage Web service at a registry may also be used to deliver general messages to the Registry Administrator. These messages could involve planned ITL maintenance periods, change management or other operational issues and plans.

6.3 Transaction Status Service

The ITL provides a public Web service to return the current status of a transaction at the ITL. This service may be used by registries to query the status of a transaction for which verification has not yet been received.

Registries may call the GetTransactionStatus Web service method on the ITL with a specified transaction identifier, and the most recent transaction status will be returned immediately to the registry.

Figure 6.6: Get Transaction Status Diagram



1. The Initiating Registry prepares and sends a request to the GetTransactionStatus function on the Transaction Log.
2. The GetTransactionStatus prepares and sends a response to the calling function on the Initiating Registry with the status of the specified transaction.

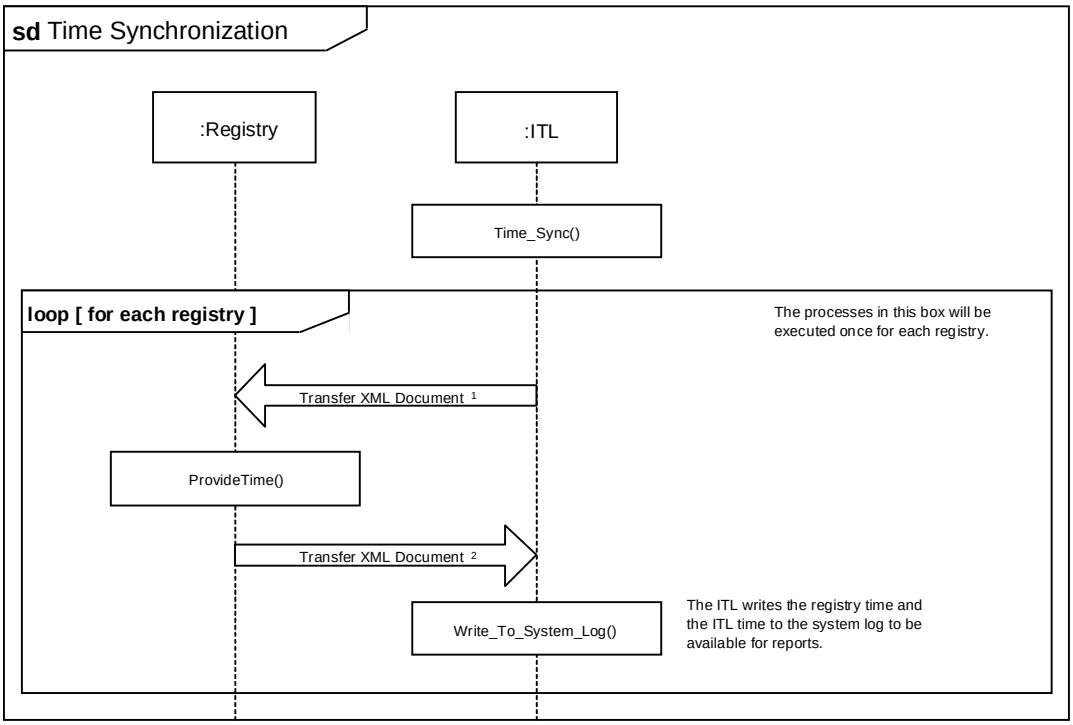
6.4 Time Synchronization

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.

The ITL will log the time synchronization result and contact the registry manager using a manual process or through a general message if a time problem is identified.

1343 **Figure 6.8: Time Synchronization Diagram**
1344



1. The function Time_Sync prepares and sends a request to the ProvideTime Web service on the registry.
2. The ProvideTime function returns a response to the ITL with the current system time on the registry.

1345

7. Technical Specifications for Data Logging

To support the need for the Transaction Log and registries to maintain accurate and consistent information, and to provide tools for use in the reconciliation process to resolve inconsistencies, four types of files shall be maintained by the registries and the Transaction Log:

- A transaction log (including both transaction summary and detailed unit holdings);
- A reconciliation history log;
- A notification log;
- An internal log; and
- A message archive.

These logs are required to support auditing functionality, both internal and external. The reconciliation process constitutes one type of external audit of a registry.

All data in these files shall be maintained for a minimum of fifteen years. Data older than one year may be archived to a secure location outside of the registry or Transaction Log, as long as it is can be retrieved or accessed within a 48 hour period should an inconsistency or question arise.

7.1 Transaction Log

The Transaction Log contains a record of each proposed transaction sent to the ITL. Each record contains a summary of the transaction content and the subsequent outcome of the transaction. Registries will be required to provide Transaction Log data to the ITL involving specific units if an inconsistency is found for specific units during a reconciliation process.

The information in Figure 7-1 shall be maintained in the Transaction Log. A specific data model for these data is not required.

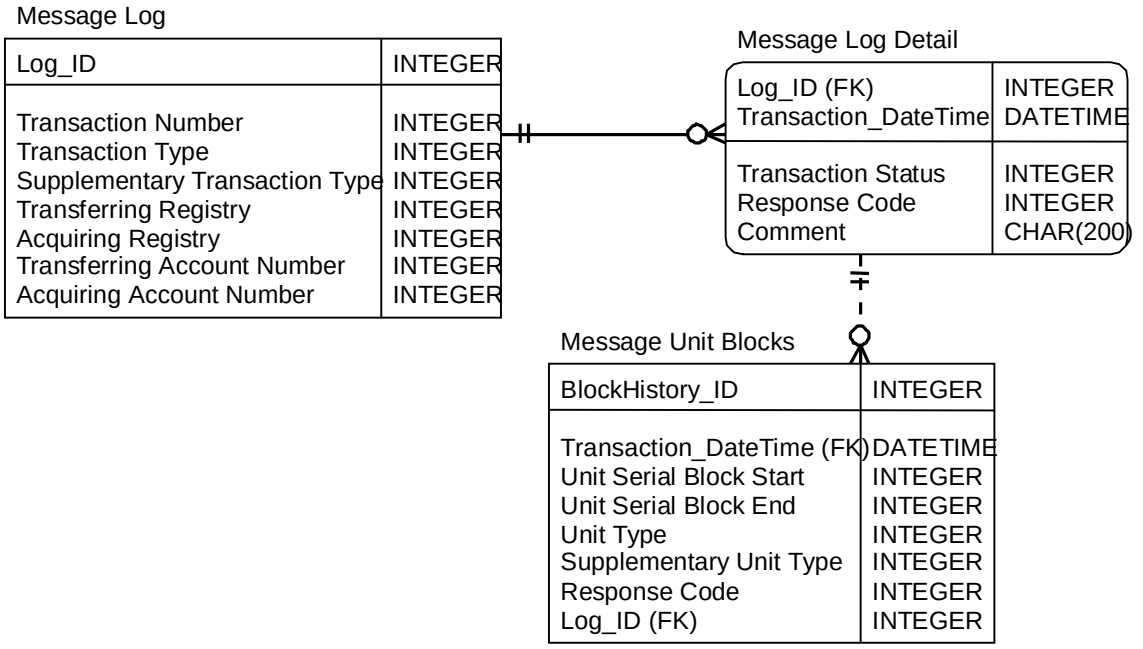
Figure 7.1: Transaction Log Attributes

Attribute	Notes
Transaction Identifier	
Transaction Type	
Supplementary Transaction Type	Required for registries subject to a supplementary program.
Transferring Account Type	
Transferring Account	
Acquiring Registry Identifier	
Acquiring Account	
Acquiring Account Type	
Transaction Status	Contained in child table.
Transaction Status Date-Time	Contained in child table.
Unit Block(s)	Contained in child table.
Response Code(s)	Contained in child table along with unit block data.

To support this information, three tables in parent-child relationships are necessary:

- A parent table containing the transaction identifier, related attributes and status information;
- A child table identifying the various statuses a transaction may be processed through; and
- A second child table identifying serial blocks and response code results. The diagram in Figure 7.2 below contains an example entity-relationship model of these tables.

Figure 7.2: Transaction Log Entity Relationship Diagram



7.2 Reconciliation History Log

The Reconciliation Log contains a record of each reconciliation action conducted by the ITL for a registry. As described in Section 5, each Reconciliation action consists of multiple steps or sub-processes. This Reconciliation Log contains one or more records for each step in a Reconciliation action.

The Reconciliation process is initiated and driven by messages from the ITL to a registry. The registry shall log each request and its response in its Reconciliation Log. The ITL shall maintain a parallel Reconciliation Log containing all requests, responses received, and results sent to a registry. Although information in the Reconciliation Log are not shared directly as part of the Reconciliation itself, access to this information by the registry administrator may be necessary to identify the manual intervention needed in order to resolve inconsistencies.

The information in Figure 7.3 shall be maintained in the Reconciliation Log. A specific data model for these data is not required.

Figure 7.3: Reconciliation History Log Attributes

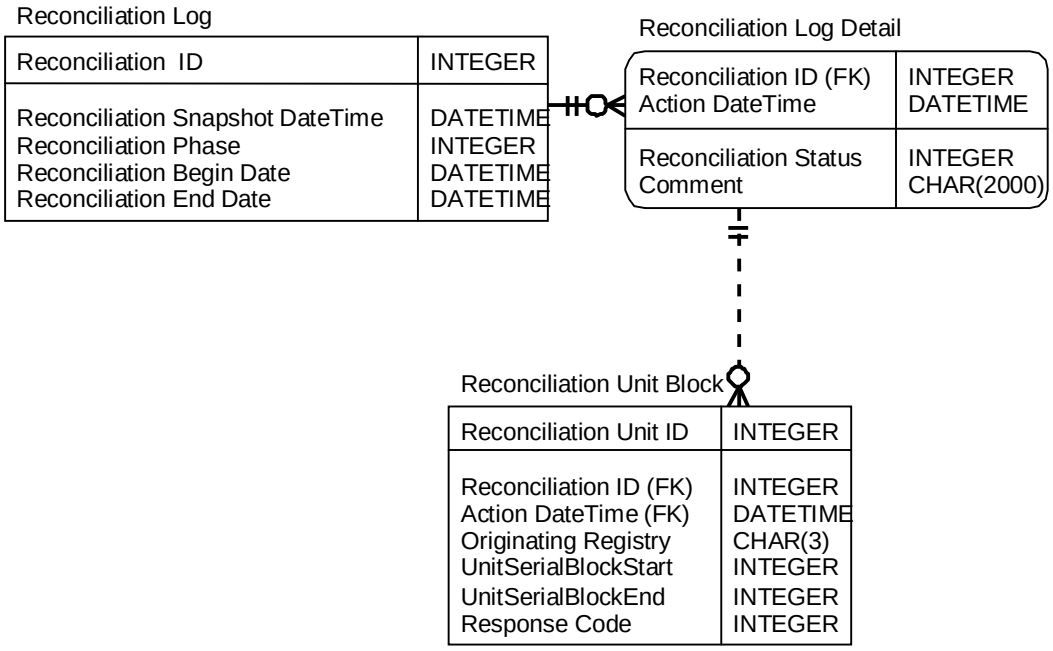
Attribute	Notes
Reconciliation ID	Unique identifier for a reconciliation action as requested by the ITL
Reconciliation Begin Date	
Reconciliation End Date	
Reconciliation Snapshot DateTime	
Reconciliation Phase	
Response Code	Contained in child table.
Unit Blocks	Contained in child table.
Reconciliation Comment	Information recorded by the registry manager regarding corrective actions for manual intervention.
Reconciliation Status	Contained in child table.
Reconciliation Status Log DateTime	Contained in child table.

To support this information, three tables in parent-child relationships are necessary:

- A parent table containing the reconciliation number, date and time reconciliation was initiated, ended, phase requested, and the date and time the snapshot of current holdings was requested;
- A child table identifying the specific activity and response codes necessary to complete the reconciliation; and
- A child table to the specific activity containing the unit blocks that were identified as inconsistent and by the ITL corresponding response code information.

The diagram in Figure 7.4 below contains an example entity-relationship model of this information.

Figure 7.4: Reconciliation History Log Entity Relationship Diagram



7.3 Notification Log

Each registry and the ITL shall also maintain a log of notifications generated by the ITL and sent to a registry. These notifications inform the registry regarding specific actions that should be taken relating to units. See Section 6.

The Notification Log at the registry shall contain the attributes in Figure 7.4. A specific data model for these data is not required.

Figure 7.5: Notification Log Attributes

Attribute	Notes
Notification ID	As generated by the ITL and sent to the registry.
Notification Type	
Notification Received Date	
Total Units	If appropriate. For example, notifications relating to Reversal of Storage.
Project ID	If appropriate.
Notification Text or Message Location	To store a complete copy of the notification content.

7.4 Internal Audit Log

Each registry and the ITL shall also maintain an internal log of changes to data which are critical to the transaction or reconciliation process. The scope and design of this functionality is the responsibility of the registry administrator. The internal log shall capture information on internal and external trades, including in particular the user ID and date/time of all recorded trades. Information contained in this log is not shared directly with the ITL. It is required to provide additional information for use by the registry administrator for manual intervention when an inconsistency is discovered in the reconciliation process.

The internal log shall contain the attributes in Figure 7.6. A specific data model for these data is not required.

Figure 7.6: Internal Log Attributes

Attribute	Notes
Activity type	For example, insert, delete, update, login attempt.
Activity Date-time	
Entity Affected	For example, table name.
Field Modified	Attribute in table that was updated.
Old Value	
New Value	
User ID	Person who executed change if not performed through Web service.
Source of Activity	Identifies what server or workstation activity was submitted on.

7.5 Message Archive

Each registry and the ITL are required to store a copy of messages sent and received, in their entirety, as stand alone files. These files provide additional information for use by the registry or the ITL administrator when an inconsistency is discovered which relates to a messaging problem which cannot be resolved through the use of the transaction history or internal logs.

The location and the medium for this storage are at the discretion of the registry or the ITL administrator. The naming convention of the files must enable an authorized user to retrieve the file for a specific transaction or reconciliation. It is recommended that the files be stored in compressed formats using the following naming convention:

aa_bb-################-cc.zip

where:

aa = Registry country code per ISO3166, and "ITL" for Transaction Log messages, and "CDM" for CDM registry messages

bb-################-## = Transaction identifier
and cc = sequential number generator

For example, a file sent from the ITL about a proposed German transaction would be named:

ITL_DE_152_1.zip

7.6 Support for Testing

The ITL accepts messages from registries for testing and debugging purposes. A separate test environment will be supported by the ITL containing a subset of replicated production data suitable for testing. Web services hosted by the ITL will exist on two separate ports; 80 for production and 443 for testing. Registries must test all web services and transactions through the 443 port before being able to participate in the production environment. See Section 9 on the Initialisation process for further details.

8. Technical Specification for Change Management

8.1 Objectives

The Data Exchange Technical Specification provides a stable, agreed-upon platform for the development and deployment of the communications modules built for registries and the ITL. It is expected, however, that changes to the messages and to the criteria to ensure data quality and accuracy may be necessary over time. It is a requirement of the Data Exchange Technical Specifications that changes provide backward compatibility to the extent possible. It is the goal of these requirements to allow existing functionality to remain valid when new requirements or changes are necessary.

Anticipating these needs, this specification establishes a technical architecture that allows adjustments within the specification without imposing significant additional development costs to registries or the ITL. A change management process, supported by Registry and ITL administrators, to determine when and how this will be managed, shall also be established.

8.2 Procedural Controls

To coordinate the change management process, the following will be defined:

- A process to receive requests for changes in technical specifications, including message content and criteria, etc., and for the assessment of these requests;
- A communication mechanism for informing all participants of upcoming changes, including schedule, specific impacts, instructions, etc.;
- A change management process for developers or technical managers of the registries; and
- The consequences of failing to adopt required changes.

8.3 Technical Specifications

To minimize the impact of changes and to manage the process of ensuring that all participants implement required changes within a necessary timeframe, registries and the ITL must conform to the following technical specification:

8.3.1 Version Definition

The technical specification shall be assigned a version number, managed through the ITL. A version number consists of two elements, a major version number and a minor version number. Major version numbers will change infrequently and reflect a fundamental change in the architecture of a core component that would invalidate any previous versions. It is likely that a major version change would require coding changes to be undertaken. A minor version number indicates small changes to message content or validation rules that would not require coding changes and could be implemented completely within the existing messages structure.

Within each XML message sent or received from the ITL, the major and minor version numbers are checked. A registry that has an incompatible major version number will have all of its requests rejected and receive a response indicating that the registry is out of compliance. A registry that has an incompatible minor version number will be directed to an upgrade site and may or may not have the request processed based on the nature of the change.

8.4 ITL Web Portal

The ITL will provide an intranet web site that will post information on version status as well as allow registered users to review upcoming functional changes, time for implementation, and the technical specification for these changes. Users who have valid user accounts and passwords through the VPN will have access to this site. The site will maintain a history of all

version changes and patches released from the site. It will be managed by the ITL administrator.

8.4.1 Web Service Modifications

Changes to web services or subsequent functions that require new parameters constitute a major version change. All registries will have a specified period of time to comply with the new requirements. Detailed specifications will be provided with sample testing procedures for the registries to test the new components against the ITL.

8.4.2 Support Table Content Modification

Changes to data content in the form of new response codes or support tables are considered minor version changes. These data will be available in XML format for download from the ITL website. Included in these tables are codes identifying which response codes are new, have been modified, or have been retired. Registries must refresh their tables with current support table data as needed.

9. Initialisation of Registries

Initialisation is the process of bringing a registry system on-line, allowing it to fully participate with the ITL in a trading scheme. Prior to a registry participating in message exchange with the ITL, the registry must comply with a series of initialisation requirements and procedures. These tasks ensure that the registry meets both the functional and non-functional requirements of the Data Exchange Standards and will be able to converse consistently with the ITL. The registry will not be able to participate in any transactions until all initialisation tasks are complete.

9.1 Staff Identification and Planning

The first step to initialise a registry is to identify those individuals responsible for the registry and its operation, including the Registry Manager. The Registry Manager, or a person assigned by the party, must submit a schedule and plan for initialisation to the ITL Manager. The schedule and plan, should detail, in writing, the timing projected for each major initialisation task, including projected start and end dates. This is necessary so that the ITL Managers can ensure that the appropriate level of support and assistance is available during this period. Since it is anticipated that multiple registries will be in the process of testing and initialisation simultaneously, the ITL Manager will assign a primary staff person to work with each registry during this period. It will be important for the ITL and registry staff to develop a working relationship and excellent communication.

Although no specific format is required, it is recommended that the schedule and plan address the following areas, which comprise an initialisation checklist:

Registry Checklist

- Assign Registry Manager and support staff
- Define initialisation schedule with projected milestone completion dates. Initialisation should be completed within a 2-month time period.
- Submit Registry Documentation to ITL Manager
- Enable VPN access to and from ITL
- Submit Test Results to ITL Manager
- Obtain and test registry digital signature
- Perform tests and assess results received back from the ITL Manager
- Set up production environment
- Verify time synchronization with ITL

ITL Checklist

- Assign ITL Manager and primary staff to work with Registry Manager
- Review registry schedule and set up ITL schedule
- Review registry documentation
- Enable VPN access to and from registry
- Set up digital signature and participate in security and authentication tests
- Assist with registry tests
- Participate in tests and prepare analysis of test results
- Receive account and contact data from registry
- Set up production data for registry

9.2 Documentation

The first task of a registry is to provide documentation of their registry system. This documentation is needed to show that non-functional requirements of the DES are met and that the registry will be operated in a manner consistent with excellent operating practices. These requirements ensure the national registry has an adequate plan for addressing operational and security requirements of the application.

9.2.1 Database and Application Backup

A database and application backup plan should be submitted outlining a detailed backup plan for the production database and software. It is recommended that database backups be performed at a minimum frequency of daily.

Specific elements of the plan include:

- Identification of personnel responsible for backup (include a primary individual and an alternate, or a staffing plan);
- Identification of specific back up schedule and procedures (i.e., backup at 7pm each evening from terminal X by User ID Y);
- Identification of backup media and its location and media;
- Identification of the number of backup generations planned;
- Definition of strategy to monitor performance of backup tasks, including notification of backup failures, log review, spot checks, audit, management reporting, etc.;
- Identification of scope or content of backup procedures (i.e., database, application software, server logs, etc.); and
- Identification of backup hardware and software.

9.2.2 Disaster Recovery Plan

A disaster recovery plan designed to ensure business continuity in the event of catastrophic failure or disruption of the host environment should be submitted in conjunction with the backup procedures. The primary objective of a disaster recovery plan is to enable an organization to survive a disaster and to reestablish normal business operations as quickly as possible. In order to survive a catastrophic event, the organization must assure that critical operations can resume normal processing within a reasonable time frame. A contingency plan should be laid out in the event that the primary facility cannot perform required daily operations. In order for this plan to be effective, periodic testing and evaluation should be performed to ensure validity and viability.

Specific elements of the plan include:

- Identification of an off-site facility with adequate disk space/storage and availability to serve as an emergency hosting environment;
- Definition of specific minimum hardware and software requirements to host the registry on a temporary basis;
- Definition of roles and responsibilities for primary and alternate personnel at the off-site location;
- Definition of roll-back procedures to step back to the latest backup. This may include obtaining daily transactions from the ITL that were not included in the last backup;
- Notify all appropriate parties that a contingency plan is in effect (i.e., ITL, other registries or users);
- Identification of off-site location of documentation and procedure manuals, as well as any paper-based forms, necessary to deploy under a Disaster Recovery scenario;
- Definition of periodic testing strategy to demonstrate readiness to implement disaster recovery plan; and

- Definition of expectation for time frame in which registry could begin operation following a disaster. The time frame would depend on the volume of transactions, cost and other factors and is not expected to be the same for each registry.

9.2.3 Security Plan

A security plan is defined in order to protect the application and data from unrestricted and unsolicited use. Secure access to the data should be provided at multiple intervals to insure redundancy of protection. For Web security, you should address three primary areas:

1. Server security. The web and/or database server should be secured not only by user id and password but also physically to prevent unauthorized access to the data and application. As with most dynamic connectors to databases, a connection with full access must be granted to the Web server because various queries will need to access different tables or views to construct the HTML from the query. To prevent unauthorized use of these open data connections, the servers should be physically secured. In addition, security can be assigned at the table level on a database.

2. User-authentication security. This level of security insures no unauthorized access to information in the registry. This is accomplished by requiring unique user id's and passwords that are regularly maintained by a Systems Administrator.

3. Session security. This level of security insures that data is not intercepted as it is broadcast over the Internet. This is accomplished by encrypting data passed to and from the registry.

Specific elements of the plan include:

- Definition of rules and responsibilities for security, recognizing that actions by persons are the most significant contributing factor to the success or failure of security planning;

- Determine physical access to the Web and/or Database server;

- Assign a network and database administrator and alternates, user id, passwords, and specific responsibilities;

- Activate audit trails recording activities at the server, database and data levels;

- Employ encryption of data transferred to and from the registry;

- Require frequent changes to password, restricting replication over a period of time;

- Require passwords of specific length with a specific number of alpha and numeric characters. For example, 6 digit passwords with a minimum of 2 numbers; and

- Delete all unused User ids and passwords immediately and remove inactive user ids from the database on a regular basis.

9.2.4 Application Logging Documentation

To demonstrate conformance with Section 7 of the Data Exchange Standards, the registry manager is asked to provide a summary of the registry capability to maintain database logs and activity logs.

- Database Logging. Database administrators are required to implement transaction logging where logs for files can be periodically shipped to a remote server or alternate site. For Oracle databases, this is the equivalent of archive logging; for MS SQL Server this might be implemented with log shipping.

- Activity logging**.** Activity logging should be utilized to track unauthorized attempts to log on to the server as well as general usage.

The documentation should include:

- Definition of regular backup and archival of transaction logs;
- Definition of hardware utilized to store logs; and
- Assignment of personnel to review activity logs on a regular basis.

9.2.5 Time Validation Plan

For the successful data exchange, a registry must define and follow specific procedures to validate server time on a periodic basis.

The plan should include:

- Schedule for periodic time validation;
- Identification of time server to provide validation;
- Assignment of personnel to perform or monitor time validation;
- Maintenance of documentation of time validations and any time adjustments resulting;
- Definition of tolerance for time validation discrepancies; and
- Definition of process for adjusting time.

9.2.6 Version Change Management

A clear migration path should exist to upgrade from version to version of registry software and database schemas. When a new version is released it must go through the testing sequence to insure that it is operable. This invokes preparing a testing environment and a test plan and a migration path to move the code and database schema to production assuming it has passed the testing sequence.

The Change Management Plan should include:

- Deployment Strategy
- Test Plan
- Notification strategy
- Data management/loading plan

9.2.7 Test Plan and Test Report

The test plan ensures that a registry has performed basic testing and is capable of participating in the tests outlined in Annex H which are required of a registry prior to being authorized to submit production transactions to the ITL. The test plan describes the various levels and types of testing that will be done throughout development.

A test plan should be outlined that steps through the basic system tasks to ensure no changes made in the test environment will affect day-to-day processing. This should cover System Administrator functionality, as well as all user-level testing. All test cases should be documented and archived for proof of concept and documentation purposes. A migration plan should be clearly outlined to move the test code, schema, or data to the production environment with minimal impact to the overall system (choose times of least traffic).

The test plan should include:

- Description of overall test strategy, testing procedures and documentation;
- Identification of Test criteria;
- Identification of Testing tools;

- Assignment of personnel to perform testing of the software, both on the initial release and for an upgrade in hardware or software;

- Description of test environment and management of that environment to ensure that results replicate the results expected in a production environment;

- Evidence that the plan provides for systematic testing in logical order of all module; subsystem, and system requirements against a well-defined set of test cases;

- The plan includes a method for documenting the performance of all tests in a test log; a method for identifying and reporting any anomalies or errors; and a procedure for tracking problems from detection to resolution;

- Plans for creating the test environment, including all needed software and hardware purchases, are consistent with the application development schedule; and

- Evidence that regression testing is a fundamental element of the plan.

9.3 Initialisation Tests

Once the ITL administrator is satisfied these requirements have been met, the national registry can begin to establish electronic communication with the ITL. These procedures are performed in stages and are described below. Detailed requirements and processes for these tests are defined in Annex H.

Figure 9.1: Table of Initialisation Tests

Test	Test Type	Who Initiates the Test	Description of Test
Communication Initialisation	Required	Registry and ITL	Installation, initialisation and test of the VPN.
ITL Extranet Login	Required	Registry	Creation and verification of ITL extranet account and password.
Registry Transaction Web Services	Required	Registry	Registry tests ITL Web services.
ITL Transaction Web Services	Required	ITL	ITL tests the registry Web services.
Query Services	Recommended	Registry	Test of querying capabilities.
Registry Reconciliation Web Services	Required	ITL	Web service test from ITL requesting reconciliation data.
ITL Reconciliation Web Services	Required	Registry	Web service test in which registry submits reconciliation data.
Data Request	Required	ITL	Web service initiated by the ITL requesting data from a registry.
Data Identifier Initialisation	Recommended	Registry	Download and import of lookup table data from ITL extranet website.

9.4 Communication Initialisation

A registry must be able to demonstrate that a secure communication channel can be established to and from the ITL communication hub. The Registry Manager may elect for the ITL Manager to remotely administer the VPN. This requires the ITL Manager to assist or direct the installation and configuration of the VPN at the registry network. The test to validate

connectivity can be performed at the time of installation. If the Registry Manager elects to install and configure the VPN, then an appointment shall be negotiated with the ITL Manager to test VPN connectivity. The test must demonstrate that the ITL and registry are able to connect to and send transmissions to and from each other. The IP address for both the registry VPN and ITL VPN shall be recorded and documented as valid and trusted connections to each other. This test shall be conducted and completed within a single business week. The ITL Manager shall notify the Registry Manager whether the test result was accepted or rejected as incomplete. The results of the Communication Initialisation Test are recorded in the ITL database.

The tests conducted are as follows:

- Registry must test for Internet access;
- Registry records IP address of ITL for site-to-site configuration of the VPN. ITL will supply the configuration specifications for the VPN;
- Registry pings ITL IP address to validate VPN hardware can see ITL VPN;
- ITL records IP address of registry;
- Registry acquires digital certificate from 3rd Party Certificate Authority and install appropriate files;
- ITL is sent public key of certificate either from Certificate Authority or from the registry; and
- Authentication test of Digital Certificate is initiated by the sending and receiving of public keys.

9.5 Access to ITL Website

There are two websites that support distribution of data from the ITL database, one which is public and one which requires security to access.

9.5.1 Public ITL Website

Access to the ITL public website for the purposes of querying unit transparent data does not require an account or password. The Registry Manager is responsible for checking the site to review content and alert the ITL Manager of any discrepancies in data. This test is for the benefit of the registry and does not require confirmation from the ITL. This test can be performed at any time.

9.5.2 Access to ITL Extranet

The ITL extranet hosts information regarding change management files or patches as well as XML datasets of all response codes and key identifier tables. Access to this site requires a login and password. The Registry Manager must request an account and password for access to this site. This test is for the benefit of the registry and does not require confirmation from the ITL. This test can be performed at any time.

9.6 Web Services Testing

The ITL Manager will host both a test and production environment for registries to test Web services independent of production data. Testing of Web services includes registries' testing Web services against the ITL and the ITL Manager testing the registries' Web services. All registries must first test all Web services through the ITL test environment. These tests shall confirm that all Web services have met functional specifications. The Registry Manager shall negotiate a timeframe for conducting these tests with the ITL Manager. The ITL Manager shall notify the Registry Manager as to whether the test results were accepted or

rejected as incomplete. The results of the Web Services Test are recorded in the ITL database. These tests shall include:

- Registry Transaction Web Service Tests
- ITL Transaction Web Service Tests
- Query Web Service Tests
- Registry Reconciliation Web Services
- ITL Reconciliation Web Services

9.7 Request for Other Data

The ITL requires that a registry provide information to the ITL for possible distribution to the various ITL Websites. These data include information regarding account and representative information as well as general queries for time synchronization. The ITL Manager shall negotiate a time frame for performing these requests with a Registry Manager. These tests shall be conducted and completed within a single business day. The ITL Manager shall notify the Registry Manager regarding whether the test results were accepted or rejected as incomplete. The results of the Data Request Test are recorded in the ITL database. The following Web services will be tested for the submission of data. For additional information on the content and methodology for Web service testing, see Annex H.

9.8 Data Identifier Initialisation

Registries are expected to load all response code data as well as data for all key identifier lookup tables into their systems. The following table identifies the data that the registry needs to download from the ITL extranet Website for integration into their systems. This initialisation is for the benefit of the registry and does not require confirmation from the ITL. This data initialisation can be performed at any time. The datasets are listed below.

Figure 9.2: Look-up Table Initialisation

Data Set	Description
Account Type Code	Account Descriptions
Registry Code	ISO-1066 Country Codes and Identifiers for Registries
Unit Type Code	Identifies type of unit
Supplementary Unit Type Code	Identifies additional unit type codes for an STL
Transaction Type Code	Identifies the type of transaction
Transaction Status Code	Identifies the status of a transaction
Response Catalog	Lists of all possible response code and descriptors
Supplementary Transaction Code	Identifies additional transaction type codes for an STL

9.9 Full System Test

Once each individual component of the registry-ITL communication system has been tested, the registry should initiate a series of transactions and allow them to continue to completion without manual intervention. These transactions will be documented in the test plan and will be representative of the different types of communications between the registry and the ITL. Once each transaction has ended with a predictable result, the registry is certified to use the production environment.

9.10 Reconciliation Services and Schedule

Initially, requests for reconciliation data shall be requested daily and after a timeframe in which no errors have been reported over a sample period of time involving numerous transactions the ITL Manager and Registry Manager may negotiate a less frequent time frame for reconciliation requests. The Registry Manager may also negotiate the time of day in which it does not present an undo burden to provide reconciliation data requests to the ITL. Failure to provide reconciliation data when requested can cause the suspension of transaction privileges.