

Ministry of Defence



Defence Material Organisation

---

**Matlogco/IV&C Branch, C2 Support Centre**

**TITAAN Logging and Reporting System**

**TITAAN Phase 2/3  
Layer Management**



---

Version : 1.0  
Author(s) : ir. J.C.F. Bijen  
Date : September 28, 2005

© Copyright Ministerie van Defensie Materieellogistiek Commando Systeemgroep IV&C  
All rights reserved. No parts of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of the Ministerie van Defensie Materieellogistiek Commando Systeemgroep IV&C.

**Contents**

<b>1. Introduction.....</b>	<b>3</b>
1.1 Structure of this document .....	3
1.2 Mission .....	3
1.3 Context .....	3
1.4 Scope .....	4
1.5 References .....	4
<b>2. Identification of Stakeholders, Concerns and Viewpoints.....</b>	<b>5</b>
<b>3. Architectural views .....</b>	<b>6</b>
3.1 Functional View .....	7
3.2 Information View .....	9
3.2.1 Logging details .....	10
3.2.2 Reporting details .....	10
3.3 Operational View .....	12
3.3.1 Management .....	12
3.3.2 Procedures .....	13
3.4 Technical View .....	14
3.4.1 Conversion of Logging and Reporting data .....	15
3.4.2 Archive of Logging data .....	15
3.4.3 Building the data warehouse database .....	15
3.4.4 Data transformation into the data warehouse .....	17
3.4.5 Deployment of Reporting engine .....	18
3.4.6 Building reports .....	18
3.5 Global Architecture View.....	19
3.5.1 Distributed architecture .....	19
3.5.2 Zero dependency and graceful degradation .....	19
3.5.3 Usage of COTS products .....	19
3.5.4 Zero maintenance .....	20
3.5.5 Physical constraints .....	20
3.5.6 Configurability and flexibility.....	20
3.5.7 Security .....	21
<b>4. Rationale.....</b>	<b>22</b>
<b>APPENDIX A. TITAAN .....</b>	<b>24</b>
<b>APPENDIX B. Abbreviations .....</b>	<b>26</b>

**List of Tables**

Table 2-1: Stakeholders, their concerns and viewpoints of the logging and reporting system.....	5
Table 3-1: Report types.....	7
Table 3-2: (Converted) Logging and Reporting data .....	9
Table 3-3: (Converted) data sources and DTS data source providers .....	17
Table 3-4: Physical constraints .....	20
Table 3-5: Implemented security.....	21

**Table of Figures**

Figure 1-1 TITAAN Management Framework.....	4
Figure 3-1: Relationship of Global Architecture View to other Views.....	6
Figure 3-2: Reporting Use Case .....	8
Figure 3-3: Information View of TITAAN Logging System.....	10
Figure 3-4: Information View of TITAAN Reporting System.....	11
Figure 3-5: Operational View of TITAAN Logging and Reporting System.....	12
Figure 3-6: Spectrum OneClick interface.....	13
Figure 3-7: Technical View of TITAAN Logging and Reporting System .....	14
Figure 3-8: Data model of the MOM Events Fact and Dimension tables .....	16
Figure 3-9: TITAAN Data Transformation Packages.....	17
Figure 3-10: Relationships in Distributed TITAAN environment.....	19

## 1. Introduction

This document is the architectural description of the Logging and Reporting System within the TITAAN mobile operational environment of the Royal Netherlands Army (RNLA).

The TITAAN Logging and Reporting System is built to archive, collect, consolidate and, if necessary, centralise logging data from various components and to run reports against that data.

This documentation is set-up in conformity with the terminology used in IEEE1471 that describes the relationship of Stakeholders, their concerns, their viewpoints and their corresponding views.

### 1.1 Structure of this document

This document consists of the following sections:

- **Section 1:** describes this introduction, the system's mission, context, scope and used references.
- **Section 2:** describes the identification of stakeholders, their concerns and viewpoints.
- **Section 3:** describes the functional, information, operational and technical architectural views of the corresponding viewpoints.
- **Section 4:** describes the rationale for the system's architecture and design.
- **Section APPENDIX A. :** describes what TITAAN is, its components and the environment in which TITAAN operates.
- **Section APPENDIX B. :** contains a short description for the acronyms used in this document.

### 1.2 Mission

The mission of the Logging and Reporting system in TITAAN as it is laid down by the RNLA is as follows:

*The ultimate objective of logging is to provide an audit trail for detection of security issues, detection of system errors, auditing and the collection of performance data for trend analysis.*

*Within TITAAN, a number of services such as MOM, Radius, Intellitactics, TCTS, network, transmission, TMS etc (may) produce data (events) that have to be stored for future analysis or simply reference (archiving). Sometimes logged data has to be transported to a central location for correlation. The correlation (and transport) may be frequently required (during a mission) or infrequently, e.g. on the end of a mission. The amount of data that has to be handled can be quite considerable and the data can be from a security point of view quite sensitive.*

*Up to now each service handles the logging in its own way. This is incoherent, inconvenient, inefficient and complex for users.*

*For TITAAN Phase 2/3 the aim is to develop an integrated framework and implementation for the logging and archiving of the events produced over the course of a mission. The analysis of the logged data is service specific and may or may not be handled in an integrated manner. The level of integration is dependent upon the availability of current tooling in the TITAAN environment such as Microsoft Operations Manager and SQL Server Reporting.*

### 1.3 Context

The architecture of TITAAN is built in conformity with C3IA (Command Control, Communication and Information Architecture), described in Ref. 1 (C3I Architectuur C3IA Totaaloverzicht).

The C3IA focuses on the core operating tasks of the armed forces.

An important result of the C3IA is a consistent architecture framework that is used as a base for the development of information systems for the operational mobile military domain.

One of the key areas in the C3I Architecture is the Management Framework.

The C3IA Management Framework of TITAAN, that describes the information and communication systems, is depicted in Figure 1-1.

The Logging and Reporting part, of which the architectural description is subject of this document, is part of the Monitoring Fault, Performance, Security component.

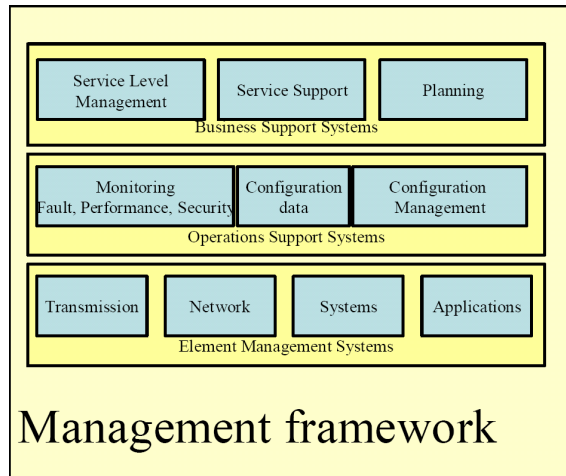


Figure 1-1 TITAAN Management Framework

The specific environment, in which TITAAN has to operate, has considerable influence on the architecture of the system.

The reader is highly encouraged to read APPENDIX A. - TITAAN to understand the usage of TITAAN in the field, its distributed character and the interaction of its components to understand the design consequences in relation to:

- zero dependency (autonomous character of TITAAN building blocks),
- zero maintenance (administrators in the field are soldiers in the first place),
- physical (and consequently disk) space limitations in the field,
- hands-free (unattended) installation,
- (flexibility of) COTS software and
- the challenge of collecting data centrally.

for developing systems for TITAAN in general and for the Reporting and Logging system in specific.

Section 3.5 elaborates more on these topics.

## 1.4 Scope

This architectural description addresses the Logging and Reporting System in the TITAAN mobile environment including the extensions to central locations in the Netherlands.

Although the architecture is suitable for implementation in other highly distributed environments interconnected by slow and unreliable WAN links, its deployment is tailored to fit into the TITAAN priming and staging process (see APPENDIX A. ) of servers in server containers and to archive, collect and consolidate mostly TITAAN specific data.

## 1.5 References

- Ref. 1. C3I Architectuur C3IA Totaaloverzicht -  
[http://www.cibit.nl/site.nsf/0/19BE87A76C7B59B2C1256F5C005723D7/\\$file/C3IA%20totaaloverzicht.pdf](http://www.cibit.nl/site.nsf/0/19BE87A76C7B59B2C1256F5C005723D7/$file/C3IA%20totaaloverzicht.pdf)
- Ref. 2. Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives -  
 Nick Rozanski, Eóin Woods, ISBN: 0321112296
- Ref. 3. Multiple Management Group Rollup Solution Accelerator -  
<http://www.microsoft.com/technet/itsolutions/cits/mo/smc/mmgp05.msp>
- Ref. 4. SQL Server 2000 Administrator's Companion – Marci Frohock Garcia et. Al. ISBN 0-7356-1051-7
- Ref. 5. SQL Server DTS – Carl Rabeler, ISBN 0-7356-1916-6
- Ref. 6. Hitchhiker's Guide to SQL Server 2000 Reporting Services – Peter Blackburn et. Al. ISBN 0-321-26828-8

## 2. Identification of Stakeholders, Concerns and Viewpoints

Within the TITAAN logging and reporting architecture, the following stakeholders are identified:

1. **Owner:** owner of the system RNLA;
2. **Users:** those who need to run reports and handle logging data, including security officer, service level manager, system administrators;
3. **System administrators:** those who must maintain the system;
4. **Security officers:** those who are responsible for security compliancy;
5. **Developers:** developers of the system.

The relationship between stakeholders, their concerns and viewpoints is outlined in the following table:

STAKEHOLDERS	CONCERNS	VIEWPOINTS
Owner	➤ Adherence to global architecture (C3IA)	Global Architecture
	➤ Compliance to statutory regulation	Operational
Users	➤ Functional capabilities	Functional
	➤ External interfaces	
	➤ Data quality	Information
System Administrators	➤ Operational monitoring and control	Operational
	➤ Archiving	
	➤ Deployment	Global Architecture
	➤ Information flow	Information
Security officers	➤ Data integrity	Information
	➤ Audit trail	
	➤ Data protection	Global Architecture
Developers	➤ Identification of real security issues	Functional
	➤ (No) Network capacity required	Information
	➤ Data conversion	
	➤ Interfaces	Technical
	➤ COTS software usage	
	➤ Physical constraints	

Table 2-1: Stakeholders, their concerns and viewpoints of the logging and reporting system

In addition, the following views are identified, corresponding to the viewpoints in Table 2-1:

1. **Functional:** Describes the system's runtime functional elements and their responsibilities, interfaces, and primary interactions.  
A use case data model is included to clarify the usage of one of the primary objectives of logging and reporting.
2. **Information:** Describes the way that the architecture stores, manipulates, manages, and distributes information.  
It contains flows of logging and reporting data to clarify the transformations of data during its operational lifecycle.
3. **Operational:** Describes how the system will be operated, administered, and supported when it is running in its operational environment. Describes the administration model for operational and procedural actions to be performed against the logging and reporting system.
4. **Technical:** Shows the software components, their detailed interfaces and their placement to the physical topology.  
It contains examples of the data models used in the reporting database and data flow within the database during data import (and export).
5. **Global architecture:** Shows how the system complies with fundamentals of the C3I Architecture under which TITAAN is developed such that it is suitable for military operations.  
It includes modelling about the relationship of the logging and reporting system in the distributed TITAAN environment and describes how fundamental C3IA concepts are implemented.

### 3. Architectural views

This section describes the architectural views of the TITAAN Logging and Reporting System that address the specified concerns of the stakeholders of the corresponding Viewpoint.

This section will address the following Views:

1. Functional View;
2. Information View;
3. Operational View;
4. Technical View,
5. Global Architecture View.

The Global Architecture View is the most important view because its compliancy, required in order to properly operate in the mobile distributed TITAAN environment with its physical constraints, greatly affects all other architectural views.

In fact, the Global Architecture view limits the scope of the other views to a single stand-alone system. The relationship of the system, environment and its architectural views is depicted below (the dotted lines represent the possible off-line data exchange).

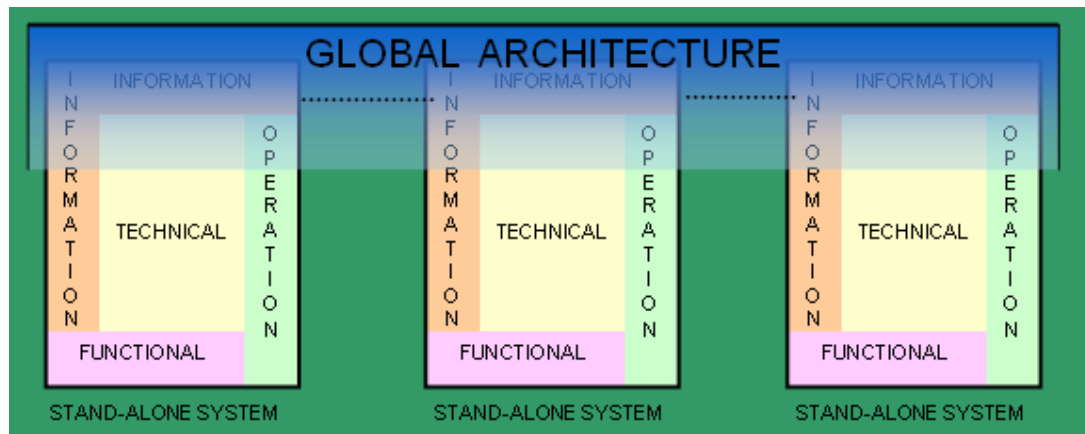


Figure 3-1: Relationship of Global Architecture View to other Views

Because the Global Architecture View contains concepts that are applicable to other views as well, such as security, flexibility and technical measurements, it is inevitable that some inconsistency is introduced because the Global Architecture View is not a completely disjoint partition with respect to the other views.

We could have implemented the Global Architecture concepts within other views, much like what is described in Ref. 2 as 'applying perspectives to views'.

However, because the Global Architecture is so important, it is assumed justified to describe a separate view for it and accept introduced inconsistencies at some level.

### 3.1 Functional View

The functional view defines the architectural elements that deliver the Logging and Reporting system's functionality.

It addresses the requirements and concerns corresponding to the functional viewpoint (see Table 2-1).

The TITAAN Logging and Reporting System provides the following two main functionalities:

1. **Archiving functionality:** Archives of raw logging data for future reference and long term storage.
2. **Reporting functionality:** For easily locating specific events and trend analysis.

For each logical reporting data, typically the following set of basic reports is provided:

REPORT TYPE	DESCRIPTION	EXAMPLE
Table	<p>Provides table views grouped by selected data into which can be drilled down, typically used to search for specific occurrences of events or data or providing top 10 reports of selected variables.</p> <p>The example shows a report of success RADIUS logons for a specific selected device.</p> <p><i>Note: Data has been masked for security reasons.</i></p>	<p>The screenshot shows a table with the following columns: User, Destination Device, Calling Device, Friendly Name, Time, and RADIUS Server. The data rows are partially visible, showing various user names and timestamps.</p>
Bar diagram	<p>A graphical presentation in which the values of the dependent variable are represented by bars, typically used for the number of occurrences or sum of values in specific time intervals, such as hourly, daily, weekly and monthly reports.</p> <p>The example shows a daily report of the number of failure logons</p>	<p>The chart is a 3D bar chart titled "Failure Logons per day". The vertical axis is labeled "Total logons" and ranges from 0 to 400. The horizontal axis is labeled "Day" and shows dates from 15-09-2005 to 22-09-2005. The bars are colored in shades of green and blue, representing different categories of failure logons.</p>
Pie chart	<p>A circular chart divided into segments, illustrating relative values.</p> <p>The example provides a Pie Chart with number of VOIP calls between selected units</p>	<p>The chart is a 3D pie chart titled "Total number of outgoing calls for phone numbers of Calling Unit". It shows two segments: a large blue segment and a smaller green segment. A legend on the right indicates the values for each segment: 751 (blue) and 303 (green).</p>
Line chart	<p>Shows the values of a selected variable typically in relation to time.</p> <p>The example provides a line chart of the Packet_Rate % for serial WAN interfaces.</p>	<p>The chart is a line graph titled "Packet_Rate". The vertical axis is labeled "Packet_Rate" and ranges from 0 to 100. The horizontal axis is labeled "Time" and shows dates from 02-09-2005 12:00 to 09-09-2005 12:00. The line shows fluctuations in the packet rate percentage over time.</p>

Table 3-1: Report types

Reports are available by connecting to the Reports virtual directory of the reporting server and browsing to the desired report.

Typically, reports have parameters so that they can be filtered upon the time interval the data was generated, on the device that logged the data or on user name that attempted to log on for example. Furthermore, generated reports can be stored in other formats such as Excel, PDF and they can be generated at scheduled times.

The following reports are available for users:

- **Security Reports:** based on events such as logon success, failures, policy changes, group membership changes, account lockouts.
- **Network Reports:** based on critical errors and alarms for TITAAN network devices such as routers and switches and specific statistics data of these devices.
- **Telephony Reports:** based on VOIP calls between units.
- **TITAAN Reports:** based upon specified events, alarms and performance data generated by Windows servers and applications.
- **TMS reports:** based upon audits of TMS object changes.

In addition, the following logging data is archived so that data and events can be traced from within their originating data: audit of security events (event logs), RADIUS logs, VOIP data, accounting data from network devices and alarms, events and statistical data from network devices and (windows) servers.

The usage of the Logging and Reporting System can be clarified by the following typical use case.

**Name:** Obtain onus of proof of security breach.

**Description:** Locating specific failed logon attempts in logging data.

**Preconditions:** Logging and reporting data is available for the time interval the security breach is suspected.

**Postconditions:** Logging data will be found if tapes are properly marked and securely stored.

**Basic course of action:**

1. A supposed security breach where an unauthorised user attempted to logon must be investigated and proven.
2. The security officer opens the Security Reports map from the Reports homepage and browses to the daily or weekly logon failure reports, dependent upon the accuracy of the time interval of the security breach.
3. He inserts a time interval in between the attempt is supposed to have taken place as the report parameter and runs the report.
4. He identifies a high peak and notifies the week or day the peak is measured.
5. The security officer attempts to refine the search by opening the hourly reports by users.
6. He inserts a time interval in between the week or day the previous notified peak occurred and runs the report.
7. He notifies the peak(s) of failure logons in the hourly reports and notifies the user account that was used with this logon attempt.
8. The security officer attempts to identify the original timestamps of the failure logon attempts by opening the failure logons per selected user report and inserting the identified user name as well as the appropriate time interval as report parameters.
9. He drills down into the report and finds a huge number of failure logon attempts with specified time stamps on a specific computer.
10. The security officer requests a System Administrator to locate the tape of the target computer where the event log is stored at the given time interval and to restore that log.
11. The System Administrator restores the event log and locates the number of failure logons corresponding to the timestamps found in the table report.
12. The found audit trail can now be used as onus of proof for the security breach.

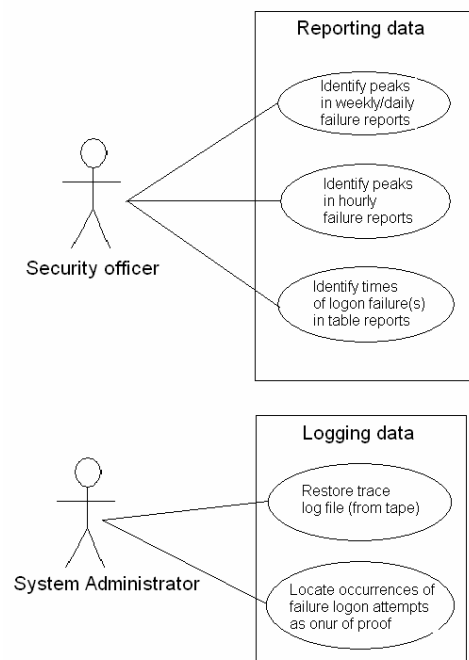


Figure 3-2: Reporting Use Case

### 3.2 Information View

The Information view defines how data and data flow in the Logging and Reporting System is manipulated.

Because some of the raw logging data types are protocols, rather than data, some data transformation must take place to prepare the data into a data type suitable for archiving and for reporting:

1. For logging, the data must be transformed into a data format that can be archived.  
Compatible data formats are files and databases.
2. For reporting, the data must be transformed into a format that can be transported into an SQL Database.  
Compatible data formats are OLEDB (SQL) compatible databases, data accessible via ODBC or plain text files.

The kind of data that must be archived and prepared for reporting (logical logging and reporting data) is defined by the functional requirements of the logging and reporting system, specified in a separate internal requirements document.

The logical logging and reporting data must be manipulated as specified in the table below:

LOGICAL LOGGING & REPORTING DATA	PHYSICAL LOGGING & REPORTING DATA	DATA TYPE	(CONVERTED) LOGGING DATA	(CONVERTED) REPORTING DATA
Security events, Critical Alerts, Errors, TMS object changes, RADIUS Accounting	Windows Event Log	Files	Files	<b>SQL Server database</b>
Performance data, Replistor data	Windows Performance Counters	(WMI) events generated by the Operating System.	<b>SQL Server database</b>	<b>SQL Server database</b>
Call Detail Records	SQL server database	SQL Server database	SQL Server database	SQL Server database
RADIUS Logging, IntelliDEN, Java, TIL	Text files	Files	Files	Files
Event logging	SYSLOG	Protocol	<b>Files</b>	<b>Files</b>
Critical Alerts, Errors, Link utilisation, IntelliTACTICS alerts	SNMP	Protocol	<b>MySQL database</b>	<b>Files</b>

Table 3-2: (Converted) Logging and Reporting data

### 3.2.1 Logging details

The information View of the TITAAN Logging system is depicted in Figure 3-3.

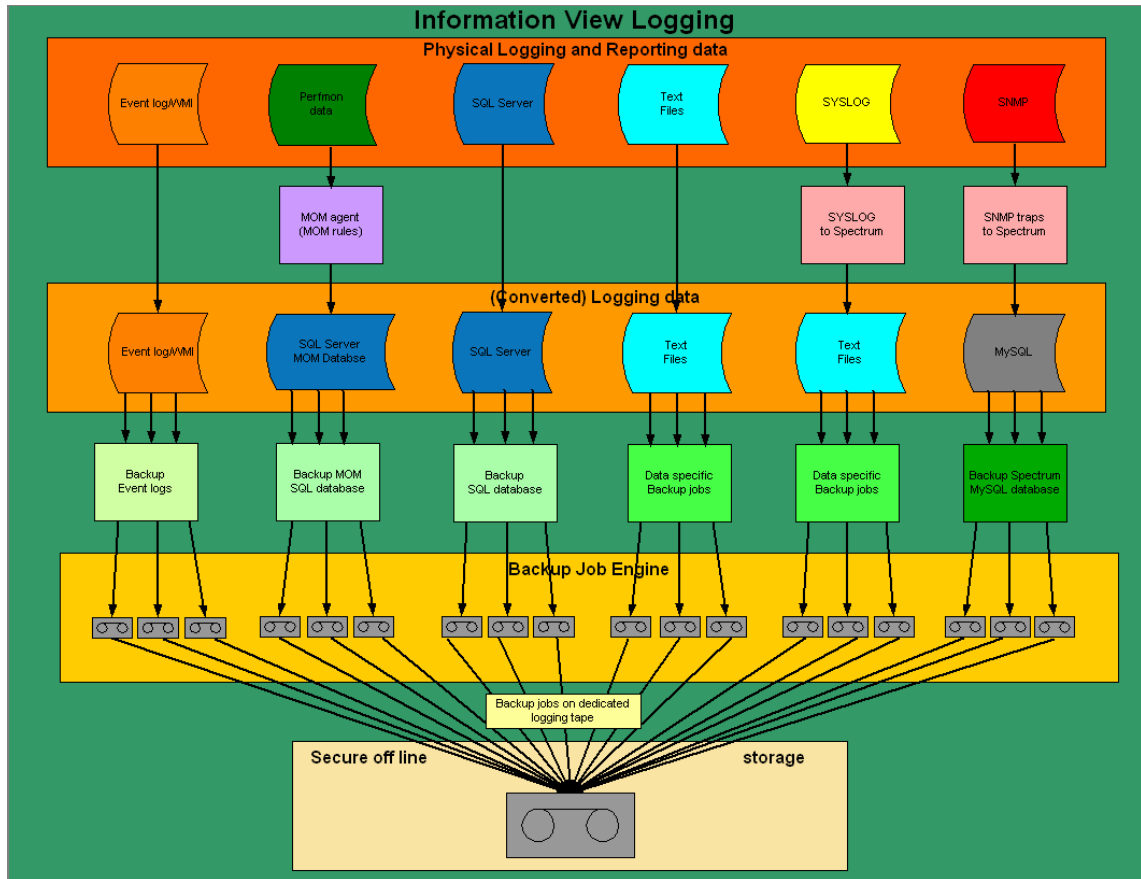


Figure 3-3: Information View of TITAAN Logging System

After the physical logging data has (either or not<sup>1</sup>) been transformed in a format suitable to be backed up by common commercial Backup software ((converted) logging data column in Table 3-2), the remaining task is to schedule the logging backup job with the logging backup selections of the (converted) backup logging data (SQL databases, MySQL database, Event log files and text files) to dedicated logging backup tape.

Because the environment in TITAAN is distributed rather than centralised, each server container has its own logging backup tapes that must be sent to a central data storage centre in the Netherlands (CISCC).

### 3.2.2 Reporting details

The information View of the TITAAN Reporting system is depicted in Figure 3-4.

After the physical reporting data has been transformed into a supported format, ((converted) reporting data column in Table 3-2), it can be transferred by means of a Data Transformation Service into the TITAAN Reporting database.

The standard is to store separate data in separate tables in the reporting database (data warehouse), such that tables exist for:

- Logon events
- Security Events
- RADIUS events

<sup>1</sup> Logging data that is converted typically loses its usage for onus of proof. It can be assumed that data that must be used for onus of proof will never have to be converted and that such data is archived in its originating data format in the TITAAN Logging and Reporting System. The Windows Security Event Log is clearly an example of this.

- Critical alarms
- Statistical data
- Call Detail Records (VOIP)

The rationale for this is:

- (Static) partitioning of data: one big table for fact data has serious performance impact on the server. Generating reports is much faster on smaller tables that only contain the required data.
- Security partitioning: some data is more sensitive than other, role based security can be configured much easier on a table level.  
E.g. Frontoffice Administrators may need to have read permissions on history of account lockouts but they may not have access to other data such as Call Detail Records of VOIP.
- Ease of creating reports: Specific reports can be bound to specific tables (or views) rather than requiring in depth knowledge of the database structure
- Ease of exporting and importing data: export and import data packages are not complex because they can be data specific: they reflect the data in the specific table or view and can be protected accordingly.

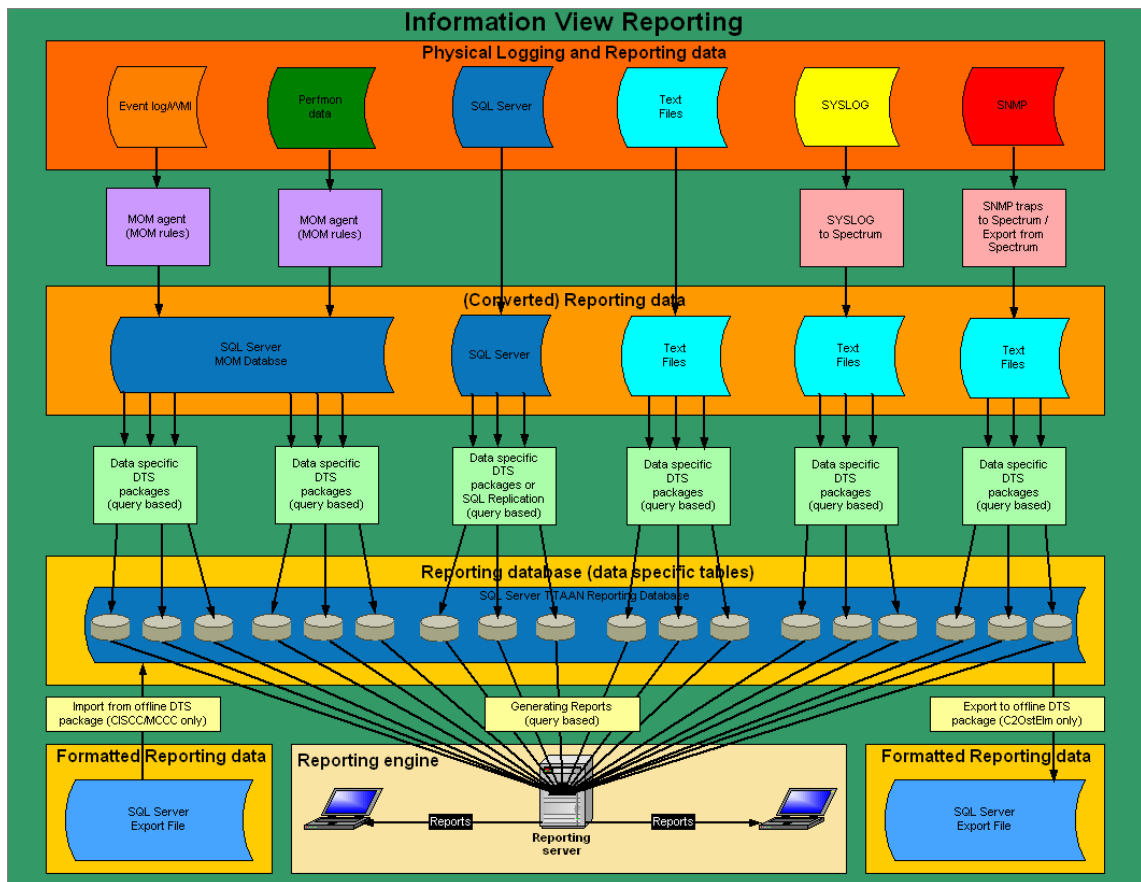


Figure 3-4: Information View of TITAAN Reporting System

Each server container has its own data warehouse (TITAAN reporting database). To centralise the data, data of the data specific tables must be exported to a flat export file (Formatted Reporting Data) that on turn can be imported into any TITAAN Reporting database. (In practice, a central (off-site) data warehouse will be used).

Pre-defined reports are being deployed in the Reporting engine of the TITAAN environment. The reports are run against the data warehouse when requested from a client browser, regardless of whether it is run against a distributed or central data warehouse.

### 3.3 Operational View

The Operational view describes:

- **Management:** How the Logging and Reporting system must be managed within an operation.
- **Procedures:** What procedures must be taken in order to preserve the required logging data?

The operational view is pictured in Figure 3-5.

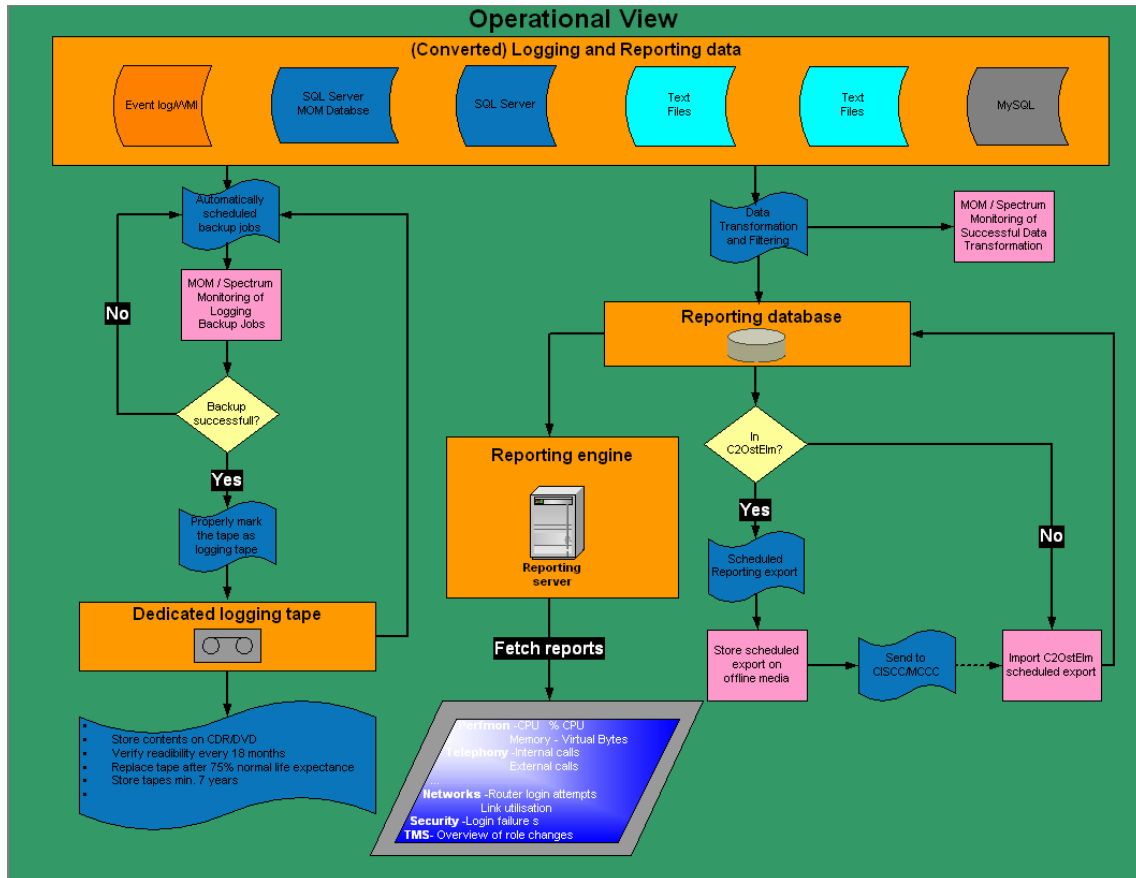


Figure 3-5: Operational View of TITAN Logging and Reporting System

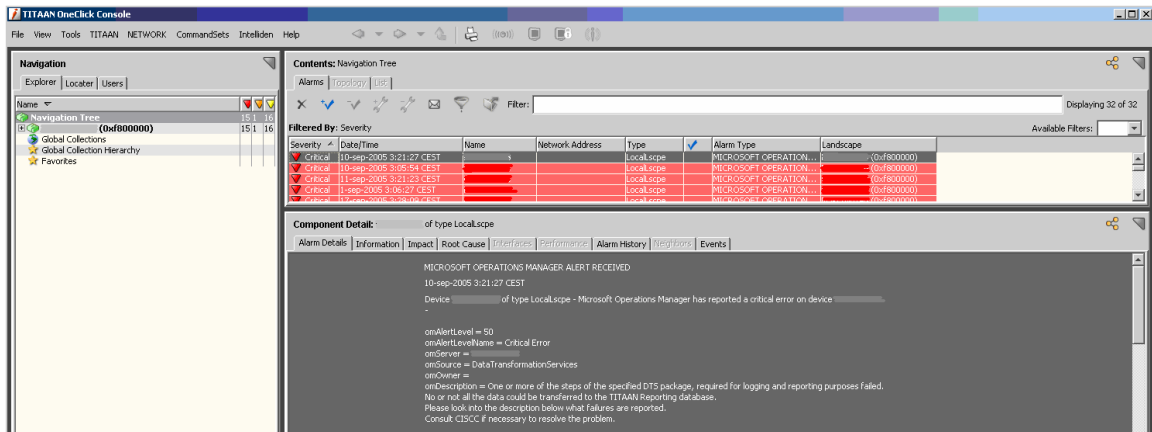
#### 3.3.1 Management

Management of the Logging and Reporting system components such as the monitoring of specific events from the backup and database software being used is provided by the management software used within TITAN (MOM and Spectrum, see section 3.4.1).

Specific failures and warnings will be logged into the Spectrum OneClick interface that is continuously being monitored in the field (see Figure 3-6).

Specific attention has been given to the monitoring of free disk space in the TITAN Reporting database as well as on the server itself, given the disk space limitations in the field.

In addition, the management software will detect missed successful events and failures of the logging backup jobs as well as missed successful events and failure data transformation jobs, so the Spectrum operator will be notified of problems at all times.

Figure 3-6: Spectrum OneClick interface<sup>2</sup>

### 3.3.2 Procedures

Besides technical measurements, some procedural actions must be performed in order to comply with security regulations (NATO and CRAMM) that apply for storage of logging (audit) data and to take care that export files of the TITAAAN Reporting database are being copied to offline media.

The logging tapes as well as the off-line media with exports must appropriately being sent to the central location in the Netherlands where the logging data can be stored appropriately and exports being imported into the central Reporting database.

Details of these procedures will be out of scope for this document but required procedures will be mentioned here:

- Properly mark the logging tape as logging tape.
- Store the contents of the logging data on other off-line media such as CD or DVD.
- Verify readability of the logging tapes every 18 months.
- Replace tapes after 75% of normal life expectance.
- Store the tapes for minimal 7 years.
- Copy / move the export files of the reporting database to off-line media.
- Send logging tapes and exports securely to central location, thereby treating the data as being of the highest security classification for that mission.
- Send tapes to secure off-line storage area (central location only).
- Import reporting exports into central reporting database (central location only).

<sup>2</sup> Note: Some operational data has been masked for security reasons

### 3.4 Technical View

The Technical view describes the implementation and deployment of the software components and their interfaces.

The technical view, integrating both the Logging and Reporting System is depicted below.

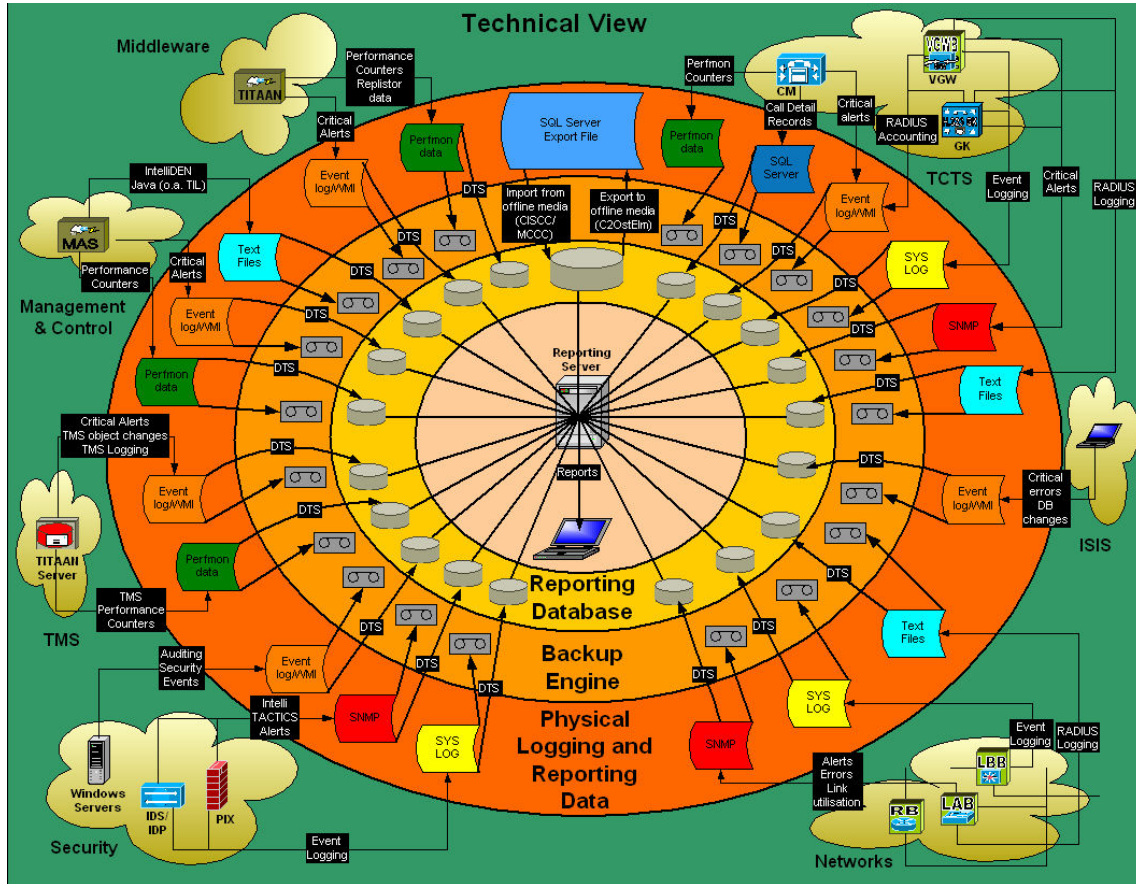


Figure 3-7: Technical View of TITAAN Logging and Reporting System

The TITAAN Logging and Reporting System consist of the following logical components:

- Logical Logging and Reporting Data:** The outer area (including the clouds, see APPENDIX B. for a short description of the used abbreviations) represents the logical logging and reporting data from the various TITAAN components, applications and services, categorised by their respective and responsible TITAAN layers (a TITAAN layer has specific responsibility for a certain part of the components that TITAAN is built of, see also APPENDIX A. ).
- Physical Logging and Reporting Data:** This is the physical logging and reporting data of the affected components / devices. This data is typically heterogeneous; some components log into a database (e.g. SQL Server, MySQL), while others log their data into plain ASCII files, proprietary event logs or are using a specified protocol to log their data such as SYSLOG or SNMP.
- Backup Engine:** The Backup Engine takes care of the archiving of the logging data so that all logging data of an operation can be restored when required. Occasionally, it might be required to migrate the data into a different format (e.g. scheduled database export to a file) if the Backup Engine cannot directly backup the logging data format itself (see also section 3.4.1). The tape icons represent the several required backup selections of the logging backup job's selection list.
- Reporting Database:** The Reporting Database contains all the logging data required to run appropriate reports (data warehouse). Typically, the data must be migrated from heterogeneous data formats into a uniform database format via Data Transformation Services (DTS). The several small database icons represent the specific tables to be created for specific data.

Parts of the data warehouse may be exported to off-line media in a timely fashion and imported into a dedicated central data-warehouse.

Because exporting and importing is a variant of the used Data Transformation (from database to database), the same Data Transformation Service can be used for this import and export of data.

5. **Reporting Engine:** The Reporting Engine, represented by the inner area of Figure 3-7, renders pre-defined reports from data taken from the data warehouse, either scheduled or on demand. Part of the Reporting Engine is the Report Designer with which new reports can be created and deployed to the reporting engine as to adapt to changing operational requirements.

From Figure 3-7, it is clear that the following topics must be addressed:

1. How data is being converted (if required) into a supported format such that it can be backed up securely and can be transferred to the data warehouse.
2. How the (either or not converted) data is being archived.
3. How the data warehouse is efficiently being built.
4. How the data is being transferred to the data warehouse and from the distributed data warehouses to a central data warehouse.
5. How the Reporting Engine is deployed to run reports against the data warehouse.
6. How reports can be created and deployed in a timely and efficient manner.

### 3.4.1 Conversion of Logging and Reporting data

From Table 3-2, the following data transformation must take place so that it can be archived and transferred to the reporting database:

- Windows Event Logs events into SQL Server database
- Windows Performance Counters into SQL Server database
- SYSLOG into files
- SNMP into MySQL database and plain text files

The translation of these data types into appropriate format is performed by the following two management systems already deployed within TITAAN:

- **CA Spectrum** (formerly Concord, formerly Aprisma): implemented as the overall TITAAN (SNMP based) management system.  
Among other functionality, Spectrum collects SNMP traps (events and alarms) and SNMP performance data into its MySQL database which can be backed up and it receives SYSLOG data that it stores in plain ASCII files that can then be backed up from disk.  
In addition, data can be exported to plain text files for import into the reporting database.
- **Microsoft Operations Manager (MOM)**: implemented as the Windows Management System for the Windows based servers. Among other functionality, MOM collects events from the event viewer as well as performance data and stores that data in an SQL database which can be backed up and from which data can be transferred to the reporting database.  
In addition MOM monitors Windows event logs and backs them up and truncates them if they reach a certain capacity (TITAAN Event Log Management Pack).

### 3.4.2 Archive of Logging data

Backup of the (sometimes converted) logging data is provided by Symantec (Veritas) BackupExec that is used as the Backup software within TITAAN. The SQL Server Agent module will be used for online backup of SQL Server databases.

The Logging data will be scheduled as a separate job and will include the following data (see Table 3-2):

- Windows Event Logs from all Windows servers, including backed up event logs by MOM.
- SQL Server databases of Call Detail Records, the MOM databases and the TITAANReporting database.
- Text files including RADIUS logs and SYSLOG generated data from network devices.
- MySQL database of Spectrum which is created as an offline backup file that can be backed up.

In a typical scenario, a full logging back up job will be scheduled monthly accompanied by a differential backup on a daily base.

The BackupExec Overwrite protection will be used so that the logging data will never be overwritten.

### 3.4.3 Building the data warehouse database

To address the physical constraints of the TITAAN environment (limited disk space and limited server resources), a well designed data warehouse is of paramount importance.

The following measurements have been taken to cope with these limitations:

- The data warehouse database is highly normalised and built-up according to the concept of fact and dimension tables according to the star design with foreign-primary key relationships. The fact table contains all the fact data and is huge while the dimension tables are typically small. The rationale for this is to keep the database disk space of a row in a fact table as small as possible. For instance, using a 4 byte integer foreign key in a fact table to a related server name of 32 byte character in a dimension table, will save a lot of disk space on fact tables with thousands of records. The following figure provides the data model for the MOM Events fact and dimension tables.

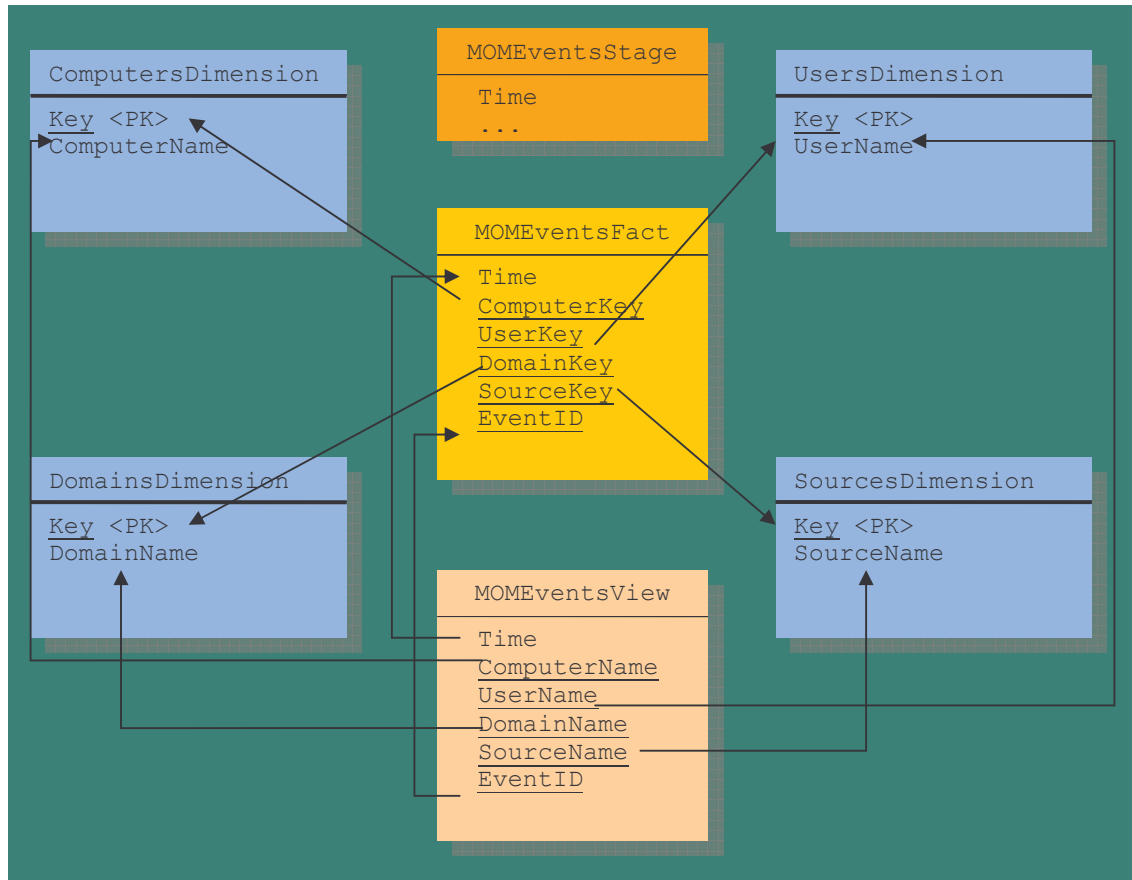


Figure 3-8: Data model of the MOM Events Fact and Dimension tables

- The abovementioned model is not always followed very strictly however; dimension tables are only being used when it saves disk space. Concessions had to be made here; for instance, in a case where devices are only required to log three performance counters simultaneously, it is more cost-effective in terms of disk space to create a column for every performance counter, requiring only one row in a fact table, than to create a dimension table with a primary key for the available performance counter and using a foreign key to that primary key in the fact table, which would require three rows in the fact table and as a consequence three disk space expensive date-time stamps. This makes the system however less flexible and less scalable; when additional performance counters are required, the table must be redesigned, rather than only (automatically) updating the dimension table.
- To speed up data transformations, the usage of staging tables is highly encouraged. It is much more cost effective in terms of system resources to bulk copy data from heterogeneous data sources into a staging table and then transfer, process and filter appropriate data into the destination tables, than to import data rows from the heterogeneous data source one by one.
- The data warehouse database design is optimised for queries performed to generate reports. Each specific kind of data has its own combination of fact and dimension tables using dedicated columns for all data we are interested in, rather than using one big fact table for all data. To ease the creation of new reports, data from the fact table is exposed via views that references typical required data.

The database software used is SQL Server because this software was already available and could satisfy these requirements.

The deployment of the TITAAN reporting database (data warehouse) is unattended using TSQL scripts for the creation of database, stored procedures, views, tables and indexes called by the OSQL command line utility of SQL Server 2000, while SQL Server itself supports silent installation by applying several command line parameters.

### 3.4.4 Data transformation into the data warehouse

Data must be transferred from the disparate reporting data sources into the data warehouse.

SQL Server DTS is being used as the ETL tool for TITAAN for the following reasons:

- Data Transformation Services (DTS) is part of the already purchased Microsoft SQL Server 2000 suite.
- DTS is flexible because data can be manipulated during data transfer with common (VB) scripting tasks.
- DTS can transfer, filter and process data from various types of heterogeneous data sources, including OLEDB compatible database providers (such as the OLE DB Provider for SQL Server), ODBC data sources and text files. It can access all the data sources used within TITAAN.

Figure 3-4 and Table 3-2 show that the following data sources must be used to access the disparate source data, followed by a suitable DTS data source provider:

(CONVERTED) REPORTING DATA	USED DTS DATA SOURCE PROVIDER
SQL Server database	OLE DB Provider for SQL Server
Text files	Text files provider

Table 3-3: (Converted) data sources and DTS data source providers

Data from the disparate data sources will be transferred on a scheduled base, typically once a day, by making use of developed DTS packages.

The lifecycle of package data as it travels in (import of source or offline exported data) through and out (export to offline data) of the database is represented as follows:

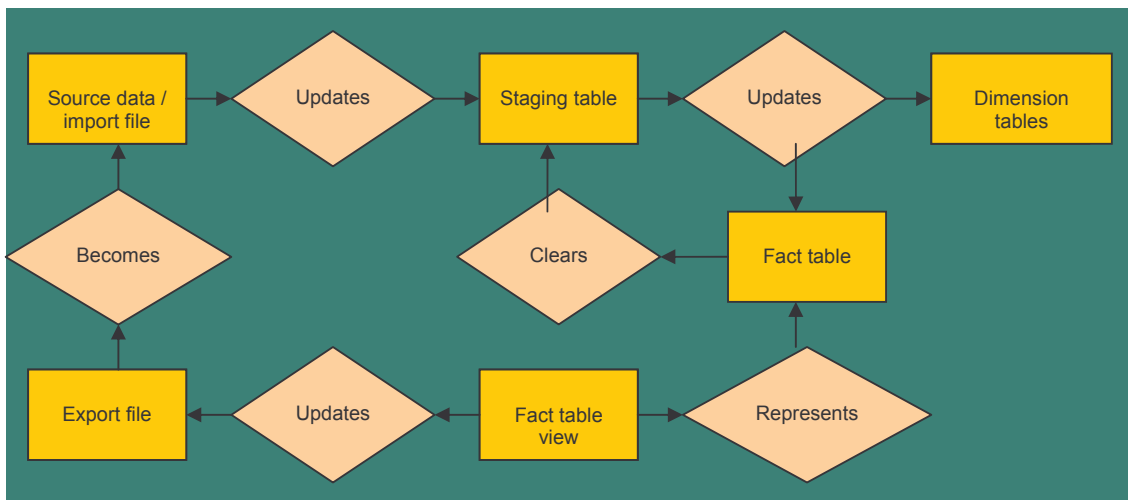


Figure 3-9: TITAAN Data Transformation Packages

Simplified, a typical DTS package for transferring data into the TITAAN data warehouse will include the following components in the order they are mentioned:

- A Transform Data Task to transfer data from the data source into (a) staging table(s), eventually performing some data conversion if necessary with an ActiveX Scripting task, such as converting a UNIX data type into an SQL Server datetime data type.
- If the Transform Data Task was successful, one or more Execute SQL Tasks to update the dimension tables if required and update the fact table using JOIN statements to insert foreign keys to the dimension tables and checking upon inserted duplicates.

- When the preceding steps have executed successfully, one or more Execute SQL tasks to clear the staging table(s) and write the successful execution as well as a date and time stamp into a separate table.
- For export of the data, the package loads data from the corresponding view and loads this into a flat export file that can be imported into a central database.

A DTS package is developed for every data transfer that has its own tables in the data warehouse to eliminate any dependency between disparate data transfers.

To maintain data integrity, data transformation is configured as a transactional process: it either completes successfully, or fails completely. In the latter case, an event is sent to the management system indicating the step where the failure took place and details about the failure.

By scripts, DTS packages are being copied to the server and are automatically added to the Windows Task Scheduler to run silently at predefined times.

DTS is installed as part of the unattended installation of SQL Server.

### **3.4.5 Deployment of Reporting engine**

The Reporting engine makes it possible to run deployed reports against the data warehouse.

For the Reporting Engine, SQL Server 2000 Reporting Services is being used.

The rationale for this choice is that:

- Reporting services is part of the (already purchased) SQL Server 2000 suite.
- Reporting Services is also used by the MOM 2005 Reporting component which makes it easy to integrate MOM reports and TITAAN customised reports.
- All reports can be accessed from a unified interface (same look and feel), rather than using separate interfaces for different kind of reports which are too difficult to use for Administrators in the field (they are soldiers in the first place).

Reports are being deployed in an unattended manner using the rs.exe command line utility, while SQL Reporting Services itself supports silent installation by applying several command line parameters.

### **3.4.6 Building reports**

Before reports can be generated, they must be deployed in a so Called Report Definition Language which is basically an XML schema definition, containing the properties of the report to be generated.

Developing reports for SQL Server 2000 Reporting Services is easy using the Report Designer, accompanied with Reporting Services. This process is also called authoring reports.

Report Designer is integrated into Visual Studio.NET 2003 with which Business Intelligence projects can be created. Authoring reports is basically a two steps process:

- Adding a dataset containing a database query (typically against a view from a specific fact table) that returns the result set with which the report generator will work with.
- Building the report layout; this can be a table, matrix or graphs, including line charts, pie charts and bar diagrams (see also Table 3-1). Items from the result set can be dragged and dropped or a wizard can be used to create the report layout.

Reports can have parameters built-in such as the start and end time of the data to be included in the report, and the report can be previewed from within Report Designer.

Because reports are generated in XML format, they can easily be edited. Templates can be created for frequently used types of reports such as TOP 10, hourly, daily, weekly overviews, and a text parser can be used to change appropriate properties for different data without having to create the reports from scratch.

Reports will be deployed silently using the rs.exe utility.

### 3.5 Global Architecture View

The global architecture (GA) view defines the architectural elements that make the Logging and Reporting system fit into the C3I architecture of RNLA.

The requirements for the GA view have the most impact on the design and operation of the system.

#### 3.5.1 Distributed architecture

Due to its operational deployment in practice, TITAAN is distributed in nature and so are its components.

The components within TITAAN, related to the Logging and Reporting System have the following relationships, depicted below.

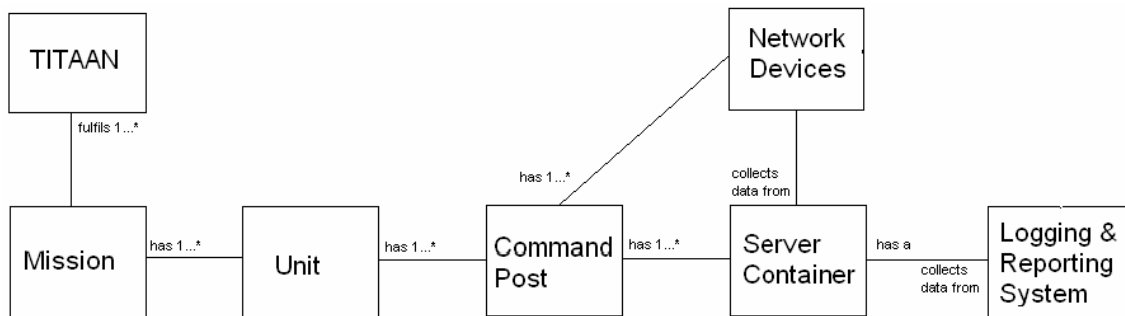


Figure 3-10: Relationships in Distributed TITAAN environment

Command Posts are typically located in different locations interconnected by unreliable and slow WAN links. Typically within command posts and always between components in a server container and nearby network devices, a self-supporting infrastructure is implemented using a LAN.

The Logging and Reporting System only (indirectly) collects data from nearby network devices and from components of the Server Container in which it is implemented.

No logging and reporting data needs to pass through the WAN; there is no logical connection to a central Logging and Reporting System, although data exported from the distributed systems can be imported into any other (including an offline dedicated central located) Logging and Reporting System.

#### 3.5.2 Zero dependency and graceful degradation

Due to its distributed autonomous deployment and configuration, there is no mutual dependency between Logging and Reporting Systems.

In addition, the operation of the Logging and Reporting System is not dependent upon the components from which it collects data. If one or more of these components fail (e.g. failure of Call Manager), new data of that component isn't just collected anymore and the Logging and Reporting System continues happily collecting and presenting data from available components as well as presenting history data of the failed component(s) (graceful degradation).

#### 3.5.3 Usage of COTS products

The Logging and Reporting System is completely built with already existing and purchased software components.

Spectrum and MOM 2005 are being configured to directly collect the appropriate data from TITAAN components, while the Data Transformation Services (DTS), Reporting Service and Report Designer components of SQL Server 2000 suite have been used to extract and load appropriate data (including VOIP data from the Call Manager) into the TITAAN Reporting database and to develop reports.

Already existing Backup Exec software is used to archive the required logging data.

### 3.5.4 Zero maintenance

Adherence to zero maintenance is achieved by the implementation of the following concepts:

1. **Integration into the existing management infrastructure:** Errors that occur anywhere in the Logging and Reporting System (such as a failure in or a missing data transfer) are forwarded to Spectrum and appear in the Spectrum (OneClick) console that is continuously being monitored.
2. **Unattended (silent) deployment:** The deployment of the TITAAN Logging and Reporting system is completely unattended. The database software, including the Data Transformation packages that are scheduled for (daily) execution, the Reporting Engine as well as all developed reports are installed on the server during the staging phase when the server will be deployed and configured before its first usage in the field.  
In addition, a logging backup job is added to the backup procedure including the data to backup.

### 3.5.5 Physical constraints

The physical constraints within TITAAN highly affect the architecture of the Logging and Reporting System.

The following table identifies the physical constraints and how it affects the design of the specified components:

CONSTRAINT	CONSEQUENCE	AFFECTED COMPONENT	SOLUTION / WORKAROUND
Slow and unreliable WAN links	Distributed architecture	(Central) Reporting Database	- Off-line import / export.
Physical space	All components installed on a single (management) server <sup>3</sup> (performance)	Reporting Database	- Highly optimised for queries to generate reports. - Partitioning of database: Separate smaller tables for separate data.
		Data Transformation	- Usage of staging tables to speed up data transfer. - Data transfers scheduled at less busy times.
		Reports	- No use of complex time consuming queries.
	Lack of disk space	Reporting Database	- Highly normalised. - Database designed such that saving disk space has higher priority than flexibility. - Only selective data is transferred. - Source data is removed if not required for logging after successful transfer.
		MOM Reporting database	- Aggressive grooming of old data
		Logging data	- Optional manual removal of logging data that is backed up.

Table 3-4: Physical constraints

### 3.5.6 Configurability and flexibility

Configurability and flexibility is achieved by the implementation of the following concepts:

1. **Selection on model types:**  
New components (servers, routers) are automatically discovered by the underlying monitoring systems (MOM and Spectrum) and, because collected data is based on specific data from model types (e.g. routers, TITAAN servers) rather than from specified (by name) devices or servers, the Logging and Reporting System automatically adapts the changes, collects the required data and can generate reports against that data.

<sup>3</sup> The server is a HP Proliant DL 380 G3 or G4 server with 4 GB of RAM and two hardware mirrored 36GB disks.

## 2. Flexibility of existing software:

When new data types must be collected (e.g. data from VPN service), from a reporting point of view, data extraction from nearly all current and future data types can be supported:

- Applications that generate the new data type typically have ODBC drivers available for data access or have the ability of exporting to a flat file so their data can be extracted and loaded into the reporting database by DTS.
- The new data type maybe compatible with sources that MOM can use (Event log, SYSLOG, text, WMI) and be stored into the MOM reporting database and from there transferred to the TITAAN reporting database with DTS.

New logging data locations can easily be added to the existing archive procedures of logging data.

## 3. Reuse of generic components:

Already existing procedures and management (e.g. extending MOM management packs to collect the new data) can be reused when extending the system and code can be reused to transform the data in a way similar to the current data transformation (see also Figure 3-9).

Creation of additional reports is easy.

### 3.5.7 Security

The following security measures are implemented to protect affected data of the Logging and Reporting System.

Special care is taken for the integrity and archiving of raw logging data because the onus of proof is invalidated when the original logging data is converted for reporting purposes.

COMPONENT	IMPLEMENTED SECURITY
On-line logging data	- Protection by access control lists (ACL) on file and directory level.
Logging data	- While members of the Windows group Backup Administrators can backup and restore even data for which they do not have any permissions, restored data for which they did not have permissions cannot be accessed unless they have been provided permissions by other means. - Organisational procedures to safely protect and archive data. - Data on the tape can be protected with a password.
Spectrum data	- Databases protected by proprietary database users. It requires a user name and password to access data. - None, with the current version of SNMP, SNMP data goes unencrypted over the line.
MOM data	- MOM data is encrypted when it travels from a managed agent to the management server. - Security of the MOM operational database is defined by the role the user has been assigned within the database. MOM supports the following roles of increasing permissions: MOM Users, MOM authors and MOM Administrators. - Access to MOM reporting database is limited to system administrators only by default.
Call Manager database	- Security of the reporting database is defined by the role the user has been assigned within the database and its permissions assigned to tables in the database.
Export data	- Protection by access control lists (ACL) on file and directory level.
Data Transfer	- Data Transformation packages are protected by ACL and configured to run as a scheduled task on the management server under the privileges of a dedicated service account with appropriate permissions to access source and destination data.
TITAAN Reporting database	- Security of the reporting database is defined by the role the user has been assigned within the database and its permissions assigned to tables in the database.
Reporting Engine	- Whenever a Certificate Authority (CA) is deployed within TITAAN, the report generated data is protected with SSL (HTTPS) when it travels from the Reporting Server to the user's browser. - The Reporting Engine uses a configurable role based security model with the following three basic default roles: <ul style="list-style-type: none"> <li>- <b>Browser:</b> limits users to browse through the folder hierarchy and opening reports. Within TITAAN only the browser role is assigned to specific users.</li> <li>- <b>Publisher:</b> allows users to add content to the server.</li> <li>- <b>Content Manager:</b> allows a user to take ownership of the item including the ability to manage security.</li> </ul>

Table 3-5: Implemented security

## 4. Rationale

Some of the rationale is already described in the architectural description in detail.

Most of the architectural decisions are defined by the constraints of the TITAAN environment. As a consequence, building a system for usage in TITAAN typically takes a very different approach than its usage in a business environment.

TITAAN is an evolutionary system and the logging and reporting subsystem is no exception.

A complicated factor within TITAAN is that it is difficult for the user of TITAAN to provide requirements for the system.

As a consequence, architects and developers themselves need to provide the requirements for the system they will have to build and request representatives of the users (soldiers in the field) to review these requirements.

Only after (parts of) the system or prototypes have been built, it can be evaluated by these representatives.

This will be the first time that requirements from the field come available and typically part of the system will be subject to change thereafter.

Therefore, in practice, the most important requirement is flexibility.

This requires a solid stable framework on which further solutions can be built.

This flexibility is achieved by the design of the custom database, the generic way data transformation can take place and the ease of creating new reports.

### Consideration of alternative reporting solution

During the development of the architecture, discussions had been going on whether purchasing dedicated reporting tools, such as a separate reporting tool for VOIP and separate tools for reporting SNMP alerts and statistics. However, besides that they are typically quite expensive, they have a limited number of built-in reports and creating new reports can be difficult if even possible.

These reporting tools typically have their own interfaces which all must be learned and they cannot run on the (central) TITAAN reporting database; they typically can only be run on the stand-alone systems themselves, bound to the system's proprietary database.

Because this database contains a lot of data unimportant to TITAAN, they take a lot of the limited disk space.

In short, these tools are developed for typical business environments which TITAAN is not.

### Consideration of alternative reporting database and data transfer

One of the options for selection of the reporting database was using the MOM 2005 reporting database. MOM 2005 can also collect data from disparate sources, including Event log, WMI, SNMP, SYSLOG and text files. In addition, database triggers could be used to update MOM with database changes, so all required data could be collected with MOM.

Although it is highly encouraged within TITAAN to use such functionality from COTS products rather than developing own solutions, there are several reasons for not choosing this option:

- MOM is event driven; it collects data as it occurs, not by an import procedure.  
It is not possible to directly import history data into the MOM reporting database with MOM.
- MOM collects a lot of other data as well into the reporting database.  
Due to the complexity of this database, cleaning up the database is an all or nothing action and will wipe out required data as well. On the other hand, not cleaning up this database will very quickly fill up all available disk space.
- MOM uses one DTS package to transfer data from the on-line MOM database into the MOM reporting database. Due to the complexity of this package it is not possible or supported to configure what data will be transferred to the MOM reporting database.
- For MOM events (such as events from the Windows Event Viewer), much of the interested data is contained in the description (message) field of the event.  
Extracting this information for usage in reports would require time and resource consuming aggregate queries to be used in reports.
- While MOM even supports uploading data to a central MOM reporting database (see Ref. 3), this is not a feasible option, due to TITAAN's WAN bandwidth limitation.

The only option for TITAAN would be to fill the central MOM reporting database with offline exported packages from the field. Because of the complexity of the MOM reporting database, it is practically impossible to export and import data from all tables of the last time interval from and into the MOM reporting database.

### **Delivery**

The TITAAN logging and reporting system is included for the first time at final delivery of TITAAN Phase 2/3 (December 2005).

### **Future work**

In TITAAN version 4 (release date December 2006), it is foreseen that more data will be added to the logging and reporting system and that data mining technology will be introduced to aid in the answering of complex issues such as what the consequences will be of allowing management traffic on the WAN given historical data of the last mission.

The database software being used also has a data mining component (Analysis Service) that possibly can be used for this purpose. Regarding its successor (SQL Server 2005, expected release November 2005), improvement of the business intelligence functionality is one of the key areas.

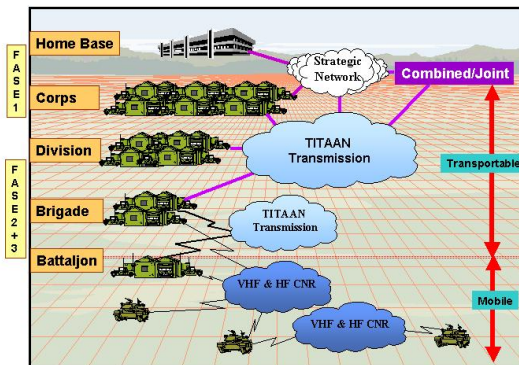
### **Return on Investment**

The Return on Investment of the Logging and Reporting System is clear. Due to the RNLA being subject of NATO requirements and CRAMM analysis, TITAAN must have a system for easy investigation of audit trails.

The alternative of not deploying a logging and reporting system is to manually search into the logging data if even available. This will quickly become unmanageable.

## APPENDIX A. TITAAN

TITAAN (Theatre Independent Tactical Army and Air Force Network) is a Communications and Information Systems (CIS) infrastructure, designed for the mobile military environment. Based on ruggedized vehicles and boxes, TITAAN delivers services for both data and telephony.



TITAAN was developed for the Dutch Army and Air Force according to the following primary considerations:

- The requirements phase is much shorter than a 'normal' military project. It is based on an evolutionary approach.
- It should be possible to incorporate modifications easily.
- In order to make use of state-of-the-art technology, the infrastructure should be based on Commercial-off-the-Shelf (COTS) hardware based on open standards.
- The number of supporting staff required to install and operate this infrastructure in the field should be kept as low as possible.
- All equipment should be easily transportable: i.e. the amount of equipment should be reduced as much as possible.

The smallest entity in the TITAAN model is a Command Post. In a Command Post, a self-supporting infrastructure is implemented using a local area network (LAN) based on 100mbps Ethernet. The servers and clients are connected to the LAN. The network can be fully functional without any external connection. Command Posts are connected with one another using wide area links, e.g. satellite or radio relay.

TITAAN also provides support for TITAAN extensions to central locations in the Netherlands (CISCC). CISCC manages multiple, independent deployed TITAAN infrastructures.

The network (IP-based) has been designed in such a way that configuration of equipment in the field or during the preparation phase of an operation is kept to a bare minimum. All network equipment is configured to be 'plug-and-play', so almost no first-line network knowledge is required in the field. When a network connection becomes unavailable, the network itself reconfigures automatically to make use of alternative paths.

In addition, the installation of servers, its configuration (such as hardening) and the installation and configuration of applications is highly automated (unattended) during a process called priming and staging.

Network devices and servers will be primed and staged at the start of an operation and cleared when the operation is finished. Consequently, network devices and servers will be primed and staged multiple times during their economical life cycle, hence the need for unattended installations.

TITAAN delivers a set of basic data services such as file sharing, printing, E-Mail, Web-based services, data replication and data availability services.

These services are implemented using Intel-based servers running Windows 2003 Enterprise. Servers are located in so-called Server Containers, vehicles that contain all the server equipment. For reasons of reliability, two Server Containers can be configured in an active-standby configuration. If equipment within an active Server Container becomes unavailable, the standby Server Container can take over. In addition to these basic services, applications including video teleconferencing, TMS and the Dutch C2 system, ISIS, have been tested on TITAAN.

TITAAN is an integrated network for all services. Voice communication is implemented with the TITAAN Converged Telephony Service (TCTS), using Voice-over-IP (VoIP), which transports voice telephony over the IP network. Call processing is implemented on a dedicated server running on

Windows 2000.

Management is implemented in a distributed design. Each Server Container has its own management server running state-of-the-art management applications.

The TITAAN 1 Phase 1 infrastructure is being used in the HRF(GE/NL) HQ in the International Security Assistance Force operation ISAF in Afghanistan. The Phase 2/3 implementation is used by Netherlands 43 brigade as a part of NATO Response Force.

### Elements in TITAAN

TITAAN is built-up of few basic elements, so called Realisation Building Blocks or RBBs, such as:

- LAN Backbone Box; part of the LAN backbone within a Command Post. In addition to backbone interconnect ports, this box has access ports for connection to LAN Access Boxes and vehicle switches.
- LAN Access Box; has RJ45 access ports for connecting workstations and IP telephones. The box provides inline power for use with the IP telephones
- Routing Box; provides WAN connectivity to the Command Post's Wide Area backbone. Through this box, serial connections, ISDN- and radio relay links can be used to link Command Posts.
- Server Containers; part of a Command Post's LAN backbone; contains the following servers:
  - TITAAN Server, delivering file management, printing, portal server and e-mail services
  - Management and Administration Server, running the Enterprise management applications incl. Spectrum, MOM, IntelliDEN, MSD (ITIL) and **Logging and Reporting Services**.
  - Call Manager, providing voice call processing functionality within the Command Post
  - Gatekeeper (GK) and Bandwidth Broker (BB), regulating access to bandwidth by the high priority traffic
  - (Secure) Voice Gateway ((S)VGW) for connecting the VoIP network to a traditional POTS or ISDN network.
- Adaptation Boxes, for interconnecting various WAN transmission means to CP's LAN
- Tunnel Box, providing hardened secure IP tunnels

### TITAAN development layers

The following development teams within TITAAN can be distinguished:

- **Security:** developing security services for TITAAN such as RADIUS, intrusion detection (IntelliTACTICS IDS/IDP), encryption and hardening.
- **Network:** responsible for transmission, the IP-network and network devices such as routers and switches (LAB, LBB, RBX), tunnel boxes.
- **TCTS:** responsible for call processing, call routing and video conferencing services and its devices such as CM, GK, BB and VGW.
- **Middleware:** responsible for OS (Windows), AD, e-mail, backup and data replication (Replistor) services and the TITAAN server (Windows 2003 domain controller / Exchange).
- **Management & Control:** responsible for TITAAN management services and the management server (MAS) and applications.
- **TMS:** developers of the RNLA Tactical Messaging System.
- **ISIS:** developers of the RNLA C2 application ISIS.

**APPENDIX B. Abbreviations**

<b>ABBREVIATION</b>	<b>MEANING</b>
ACP	Active Command Post
AD	Active Directory
BB	Bandwidth Broker
C2	Command and control
C2OstElm	C2 Ondersteunend Element
C2SC	C2 Support Centre
C3IA	Command Control, Communication and Information Architecture
CCTA	Central Computer and Telecommunications Agency
CIS	Communication and Information Systems
CISCC	CIS Control Centre
CM	CallManager
CONFPA	Conference and Public Announcement (Component)
COTS	Commercial Off the Shelf
CP	Command Post
CProcs	Call Processor server
CRAMM	CCTA Risk Analysis and Management Method
GK	Gatekeeper
HBL	Home Base Link
HBS	Home Base Service
HQ	Head Quarter
IDS	Intrusion Detection System
IDP	Intrusion Detection Probe
IntelliTACTICS	Intrusion Detection software used within TITAAN
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISIS	Integrated Staff Information System
KL	Koninklijke Landmacht– the Royal Netherlands Army
Klu	Koninklijke Luchtmacht – the Royal Netherlands Air Force
LAB	LAN Access Box (switch)
LAN	Local Area Network
LBB	LAN Backbone Box (router)
MAS	Management Server
MCCC	Mobile Command and Control Centre
MilSatCom	Military Satellite Communication system
MoD	Ministry of Defense
MOM	Microsoft Operations Manager
PIX	Type firewall of Cisco
POTS	Plain Old Telephony ServiceSystem
Replistor	Legato RepliStor, data replication software.
RADIUS	Remote Access Dial In User Service
RBB	Realization Building Block
RBX	Routing Box (router)
RCP	Reserve Command Post
RNLA	Royal Netherlands Army
RNLF	Royal Netherlands Air Force
SC	Server Container
SNMP	Simple Network Management Protocol
STANAG	NATO Standardization Agreement
SVGB	Secure Voice Gateway Box
SVGW	Secure Voice Gateway
TBX	Tunnel Box
TCTS	TITAAN Converged Telephony System
TE	Tunnel Engine
TMS	Tactical Messaging System application (also known as Themis)
TITAAN	Theatre Independent Army & Air Force Network
TITAAN Server	Windows 2003 domain controller / Exchange 2003 server
VGWB	Voice Gateway Box
VoIP	Voice over IP
WAN	Wide Area Network
WMI	Windows Management Instrumentation