InterSystems Corporation
Caché v5.1.0.826.0

Security Target

Evaluation Assurance Level: EAL3
Document Version: 1.1

Prepared for:

InterSystems Corporation
One Memorial Drive
Cambridge, MA 02142
USA
Phone: (617) 621-0600

http://www.intersystems.com/

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
USA
Phone: (703) 267-6050

http://www.corsec.com

Revision History

Version	Modification Date	Modified By	Description of Changes
1.0	2006-12-13	Nathan S. Lee	Initial release.
1.1	2007-01-02	Christie Kummers	Minor updates and changes throughout.

Table of Contents

Table of Figures

Table of Tables

1 Security Target Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The Target of Evaluation is InterSystems Caché v5.1.0.826.0, and will hereafter be referred to as the TOE throughout this document.

1.1 Purpose

This ST contains the following sections to provide mapping of the Security Environment to the Security Requirements that the TOE meets in order to remove, diminish, or mitigate the defined threats:

- Security Target Introduction (Section 1) – Provides a brief summary of the content of the ST and describes the organization of other sections of this document.
- TOE Description (Section 2) – Provides an overview of the TOE security functions and describes the physical and logical boundaries for the TOE.
- Security Environment (Section 3) – Describes the threats and assumptions that pertain to the TOE and its environment.
- Security Objectives (Section 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Security Requirements (Section 5) – Presents the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) met by the TOE and by the TOE's environment.
- TOE and TOE Environment Summary Specification (Section 6) – Describes the security functions provided by the TOE to satisfy the security requirements and objectives.
- Protection Profile Claims (Section 7) – Provides the identification of any ST Protection Profile (PP) claims as well as a justification to support such claims.
- Rationale (Section 8) – Presents the rationale for the security objectives, requirements, and the TOE summary specifications as to their consistency, completeness, and suitability.
- Acronyms (Section 9) – Defines the acronyms used within this ST.

1.2 Security Target, TOE and CC Identification and Conformance

Table 1 - ST, TOE, and CC Identification and Conformance

ST Title	InterSystems Corporation Caché v5.1.0.826.0 Security Target
ST Version	Version 1.1
Author	Corsec Security, Inc. Nathan S. Lee
TOE Identification	InterSystems Caché v5.1.0.826.0
Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 2.2 Revision 326, December 2004 (aligned with ISO/IEC 15408:2004); Interpretation I-0414: Site-Configurable Prevention Of Audit Loss has been applied to this evaluation. This evaluation is Part 2 extended, Part 3 conformant, and EAL3 conformant.
PP Identification	None
Evaluation Assurance Level	EAL3
Keywords	Database, DB, DBMS, SQL, Caché, InterSystems

1.3 Conventions, Acronyms, and Terminology

1.3.1 Conventions

There are several font variations used within this ST. Selected presentation choices are discussed here to aid the Security Target reader.

The CC allows for several operations to be performed on security requirements: assignment, refinement, selection and iteration. All of these operations are used within this ST. These operations are presented in the same manner in which they appear in Parts 2 and 3 of the CC with the following exceptions:

- Completed assignment statements are identified using [*italicized text within brackets*].
- Completed selection statements are identified using [*underlined italicized text within brackets*].
- Refinements are identified using **bold text**. Any text removed is stricken (Example: ~~TSF Data~~) and should be considered as a refinement.
- Iterations are identified by appending a letter in parentheses following the component title. For example, FAU_GEN.1(a) Audit Data Generation would be the first iteration and FAU_GEN.1(b) Audit Data Generation would be the second iteration.

1.3.2 Acronyms

The acronyms used within this ST are described in Section 9 – "Acronyms."

2 TOE Description

This section provides a general overview of the TOE as an aid to understanding the general capabilities and security requirements provided by the TOE. The TOE description provides a context for the TOE evaluation by identifying the product type and describing the evaluated configuration.

2.1 Product Type

The TOE is a post-relational database software program that uniquely offers three integrated data access options which can be used simultaneously on the same data: a robust object database, high performance *Structured Query Language* (SQL), and rich multidimensional access. No mapping is required between object, relational, and multidimensional views of data, resulting in huge savings in both development and processing time. Caché enables rapid Web application development, extraordinary transaction processing speed, massive scalability, and real-time queries against transactional data.

Figure 1 below shows the details of the deployment configuration of the TOE:

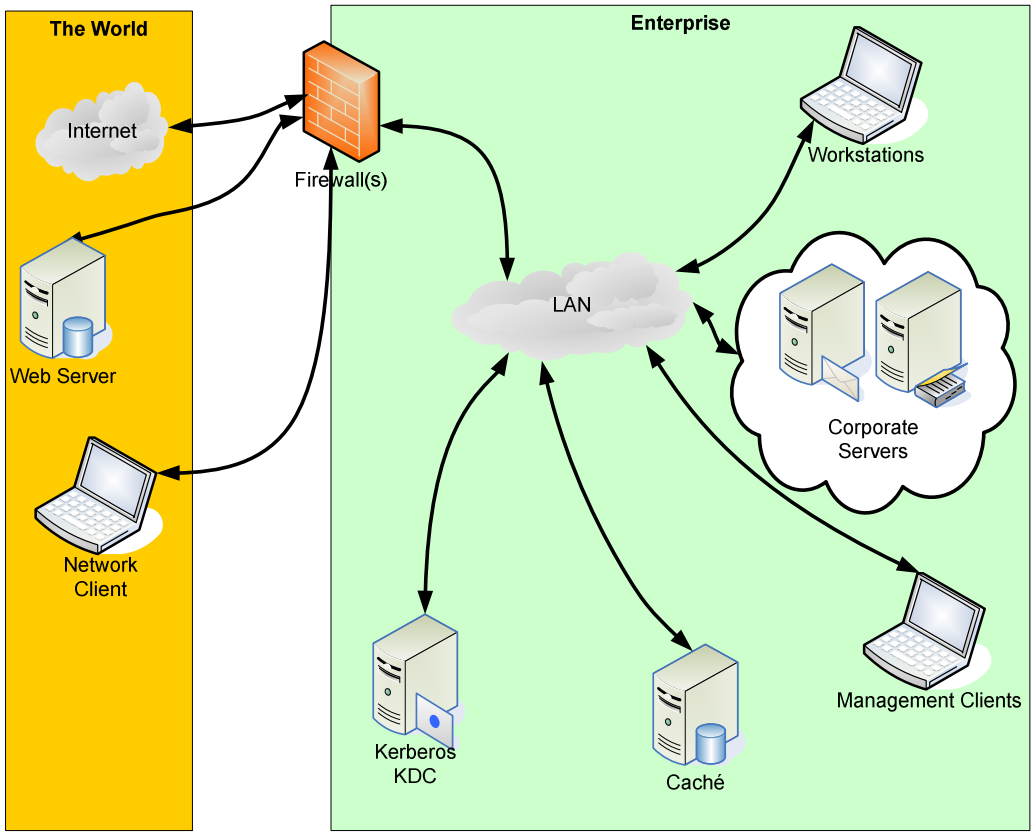

Figure 1 - Deployment Configuration of the TOE

2.2 Product Description

2.2.1 Caché Architecture Overview

The TOE is a high performance, massively scalable post-relational database system designed to enable rapid application development. Caché stores data in multidimensional form, which ensures that Caché delivers high performance even under heavy loads or running on less capable hardware platforms. Data stored within the TOE is

accessible through a wide variety of connection technologies, which promotes both openness and rapid application development because developers can work with familiar, readily-available tools.

2.2.2 The Multidimensional Data Engine

Unlike relational databases, which force data in two-dimensional tables, Caché stores data in multidimensional arrays. In addition to enabling realistic data modeling, multidimensional arrays allow faster access because they eliminate the processing overhead associated with "table-hopping" and "joins" that typify relational technology.

Although data is stored in multidimensional form, Caché gives developers the freedom to model their data any way they choose: as objects, as tables, or as multidimensional arrays. Caché comes with an easy-to-use graphic user interface for creating Caché Objects. It can also accept input from Rational Rose (an object modeling tool) and Data Definition Language (DDL) files (the standard for defining relational tables).

By virtue of the Unified Data Architecture of Caché, all data is automatically accessible as both objects and tables. There is never a need to "map" from one form to the other, and no processing overhead required to convert between forms. The Unified Data Architecture increases both productivity and performance.

Caché also allows choices when it comes to database and business logic scripting. Caché ObjectScript supports all data access methods: objects and multidimensional arrays. Caché Basic is similar to Visual Basic, with a few modifications to take advantage of unique Caché capabilities.

2.3 Product Security Architecture

Caché provides security in three "regions":

- Outside of Caché;
- Inside of Caché; and
- Within a Caché Application.

These "regions" are show in Figure 2 below:

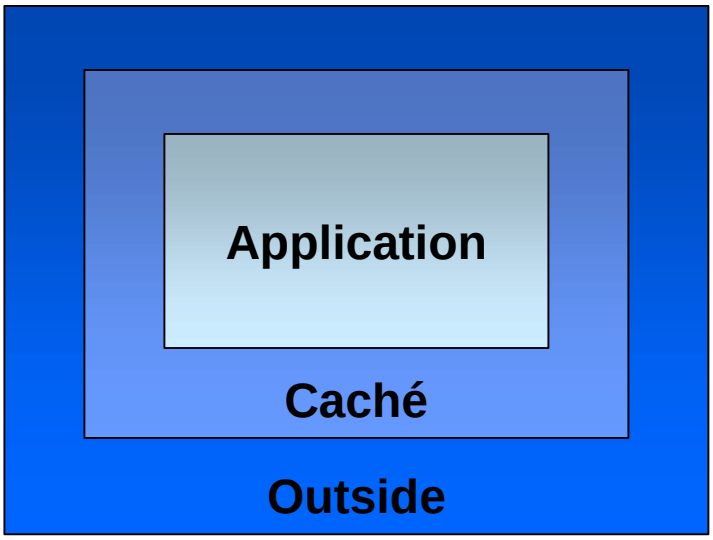

Figure 2 - Security Regions

The first "region" of security is outside of Caché. Security threats that arise outside of Caché include attempts to steal Cache database files or to eavesdrop on network transmissions. Security in this region is enforced by the information technology (IT) environment.

The second "region" of security is inside of Caché. This region of security is concerned with ensuring that only authorised users can use Caché itself, that only authorized services are available to authorised users, and that only authorised users can use a Caché application. Security in this region is enforced by Caché itself via built-in Caché security facilities (i.e. restricting/granting access to Caché utilities).

The third "region" of security is within a Caché Application. A Caché Application is an application written by the user which runs inside of Caché and uses services and data provided by Caché. This region of security is concerned with ensuring that users can only use the portions of the Application for which they have been authorised. Security in this region is enforced by the Application; Caché empowers Application developers to build security into their Applications by providing infrastructure that they can use to control access to Application abilities.

2.4 TOE Boundaries and Scope

This section will primarily address what physical and logical components of the TOE are included in evaluation.

2.4.1 Physical Boundary

Figure 3 below illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE and the constituents of the TOE Environment.

The TOE is software (comprising multiple file libraries and executables). The TOE is installed on a server that resides in the IT environment as depicted in Figure 3 below. The server hosting the TOE must run one of the following three operating systems:

- Windows Server 2003
- OpenVMS for Alpha version 8.2
- Red Hat Enterprise Linux AS (Intel 32-bit) Version 4

Both the server hardware and the server operating system are excluded from the TOE.

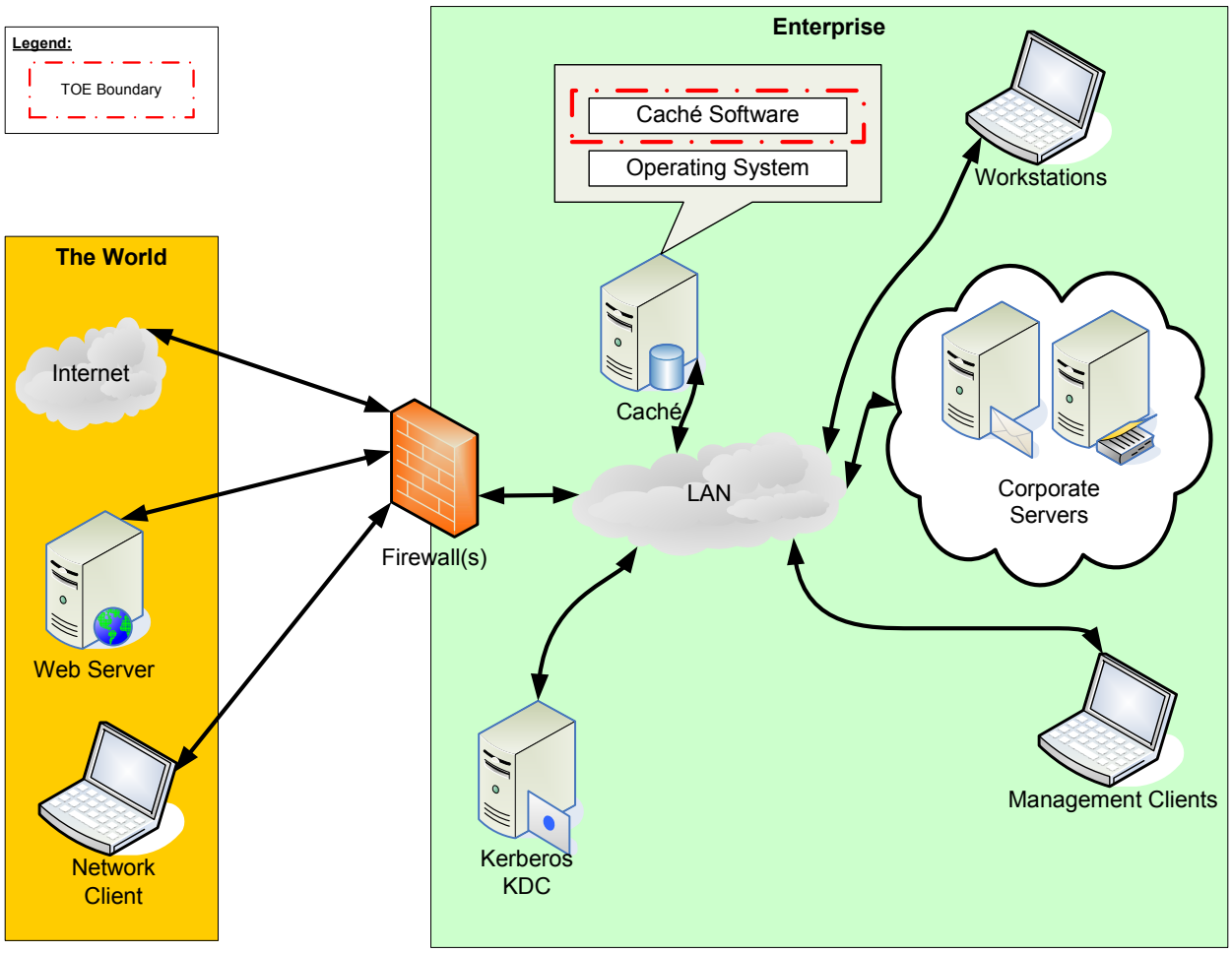

Figure 3 - Physical TOE Boundary

2.4.2 Logical Boundary

The logical boundary of the TOE is shown in Figure 4 below, and includes the Caché software component but not the underlying operating system (OS).

It is worth noting that Figure 4 shows several components labeled "Caché," each within a separate TOE boundary. These components are separate instantiations of the TOE software running on the same physical server and on the same underlying OS, but they have no direct relationship or interaction with each other. The TOE runs a process called the "Super Server" which listens for incoming connections and creates a new instance of Caché for each active connection. After initial start-up, the Super Server has no interaction with any instance of Caché. The Super Server is outside the TOE boundary. There can be numerous instantiations of the TOE running on the same server, and each instantiation is independent of the others.

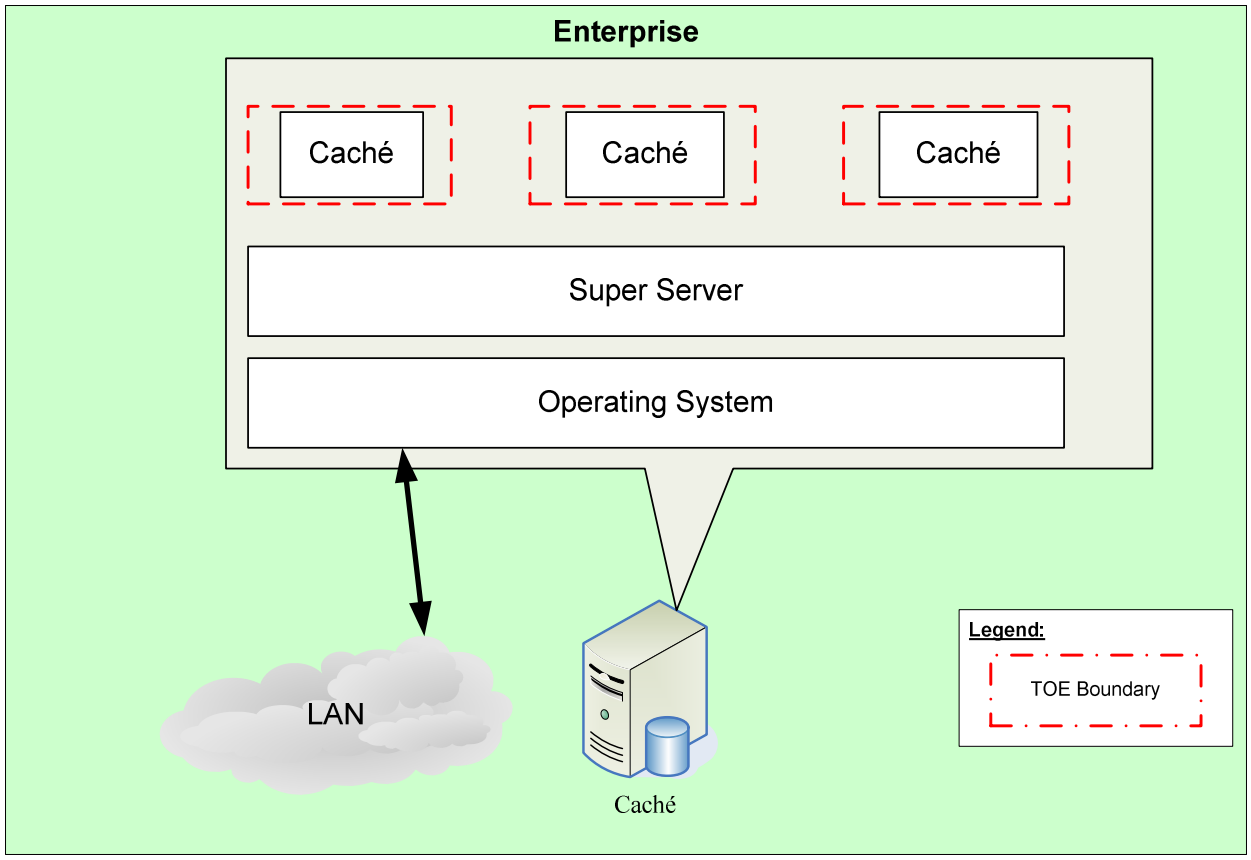

Figure 4 - Logical TOE Boundary

The logical boundary of the TOE embodies security functions that it implements. These TOE security functions are usefully grouped under the following Security Function Classes:

- Class FAU: Security Audit
- Class FDP: User Data Protection
- Class FIA: Identification and Authentication
- Class FMT: Security Management
- Class FPT: Protection of the TSF

Please refer to Section 6.1 for descriptions of these Security Function Classes.

2.4.3 Physical/Logical Features and Functionality Not Included in the Evaluated Configuration of the TOE

The following features and functionality are not part of the evaluated configuration of the TOE:

- Caché Applications written by the end-user

3 Security Environment

This section describes the security aspects of the environment in which the TOE will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all:

- Assumptions about the secure usage of the TOE, including physical, personnel and connectivity aspects
- Known and presumed threats countered by either the TOE or by the security environment
- Organizational Security Policies (OSPs) with which the TOE must comply

3.1 Assumptions

This section describes the security aspects of the intended environment for the evaluated TOE. The operational environment must be managed in accordance with assurance requirement documentation for delivery, operation, and user guidance. The following specific conditions are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 2 - Assumptions

Assumption Name	Assumption Description
A. NO_EVIL	Authorised administrators are non-hostile, appropriately trained and follow all administrator guidance.
A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on Database Management System (DBMS) servers, other than those services necessary for the operation, administration and support of the DBMS.
A.ROBUST_ENVIRONMENT	It is assumed that the IT environment is at least as robust as the TOE.
A.SECURE_COMMS	It is assumed that the IT environment will provide components to support secure data communications.

3.2 Threats to Security

This section identifies the threats to the IT assets against which protection is required by the TOE or by the security environment. The threat agents are divided into three categories:

- Attackers who are not TOE users: they have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings/parameters and no physical access to the TOE.
- TOE users: they have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings/parameters and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)
- TOE developers: they have extensive knowledge of the inner workings of the TOE and how it operates and are assumed to possess a high skill level and resources to modify the TOE during development.

The first two threats are assumed to have a low level of motivation. TOE developers have high motivation to prevent attacks on the TOE. The IT assets requiring protection are the user data saved on or transitioning through the TOE and the hosts on the protected network. Removal and mitigation of the threats are achieved through the objectives identified in Section 4.

Table 3 - Threats

Threat Name	Threat Description
T. ADMIN_ERROR	An authorised administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.AUDIT_COMPROMISE	A process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker's actions.
T.INSECURE_DELIVERY	The authorized administrator may receive the delivered TOE without the appropriate installation guidance, resulting in the improper installation or configuration of the TOE.
T.INSECURE_START	An authorized administrator may configure the TOE in such a way that a reboot will result in insecure state of the TOE.
T.MASQUERADE	An unauthorised user, process, or external IT entity may masquerade as an authorised entity to gain access to data or TOE resources.
T.POOR_DESIGN	The TOE developers may cause unintentional or intentional errors in the requirement specification, design, or development of the TOE.
T.POOR_IMPLEMENTATION	The TOE developers may cause unintentional or intentional errors while implementing the design of the TOE.
T.POOR_TEST	Lack of or insufficient testing by the TOE developers to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.
T.SYSACC	A malicious process or user may gain unauthorised access to the authorized administrator account, or that of other trusted personnel.
T.TSF_COMPROMISE	A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTH_ACCESS	A user may gain unauthorized access (view, modify, delete) to user data.
T.UNDETECTED_ACTIONS	Users of the IT operating system may perform unauthorized actions which are not detected and recorded by the IT operating system.
T.UNIDENTIFIED_ACTIONS	The authorized administrator may fail to identify and act upon unauthorised actions.

3.3 Organizational Security Policies

There are no Organizational Security Policies specified for the TOE.

4 Security Objectives

This section identifies the security objectives for the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.1 Security Objectives for the TOE

The specific security objectives for the TOE are as follows:

Table 4 - TOE Security Objectives

TOE Objective	Description
O.ACCESS	The TOE will ensure that users gain only authorised access to it and to the resources that it controls.
O.ADMIN_GUIDANCE	The TOE will provide authorised administrators with the necessary information for secure management of the TOE.
O.ADMIN_ROLE	The TOE will provide authorised administrator roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information, and alert the authorised administrator of identified potential security violations.
O.DISCRETIONARY_ACCESS	The TOE will control access to resources based upon the identity of users or groups of users.
O.INSTALL	The TOE will be delivered with the appropriate installation guidance to establish and maintain TOE security.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorised administrators in their management of the security of the TOE.
O.INTERNAL_TOE_DOMAINS	The TOE Security Function (TSF) will maintain internal domains for separation of data and queries belonging to concurrent users.
O.PROTECT	The TOE will provide mechanisms to protect user data and resources.
O.SOUND_DESIGN	The TOE will be designed using sound design principles and techniques. The TOE design, design principles and design techniques will be adequately and accurately documented.
O.SOUND_IMPLEMENTATION	The implementation of the TOE will be an accurate instantiation of its design.
O.TESTING	The TOE will undergo developer and independent testing that includes test scenarios and results.
O.TRAINED_USERS	The TOE will provide authorised users with the necessary guidance for secure use of the TOE, to include secure sharing of user data.
O.USER_AUTHENTICATION	The TOE will verify the claimed identity of users.
O.USER_IDENTIFICATION	The TOE will uniquely identify users.

4.2 Security Objectives for the Environment

4.2.1 IT Security Objectives

The following IT security objectives are to be satisfied by the environment:

Table 5- Environmental IT Security Objectives

IT Environmental Objective	Description
OE. NO_GENERAL_ PURPOSE	There will be no general-purpose computing capabilities (*e.g.*, compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
OE.TIME	The IT environment will provide a time source that provides reliable time stamps.
OE.SECURE_COMMS	The IT environment will provide a secure line of communications between the remote user and the TOE.

4.2.2 Non-IT Security Objectives

The following non-IT environment security objectives are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 6 - Environmental Non-IT Security Objectives

Non-IT Environmental Objective	Description
OE.NO_EVIL	Sites using the TOE shall ensure that authorised administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.CONFIG	The TOE will be installed, configured, managed and maintained in accordance with its guidance documentation and applicable security policies and procedures
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
OE.ROBUST_ENVIORNMENT	The IT environment that supports the TOE for enforcement of its security objectives will be of at least the same level of robustness as the TOE.
OE.SELF_PROTECTION	IT environment and its assets will be protected from external interference, tampering or unauthorised disclosure.
OE.TOE_PROTECTION	The IT environment will provide protection to the TOE and its assets from external interference or tampering.
OE.TRUST_IT	Each IT entity the TOE relies on for security functions will be installed, configured, managed and maintained in a manner appropriate to the IT entity, and consistent with the security policy of the TOE and the relationship between them.

5 Security Requirements

This section defines the SFRs and SARs met by the TOE as well as SFRs met by the TOE IT environment. These requirements are presented following the conventions identified in Section 1.3.1.

5.1 TOE Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class. Table 7 identifies all SFRs implemented by the TOE.

Table 7 - TOE Security Functional Requirements

SFR Identifier	SFR Name
FAU_GEN.1	Audit data generation
FAU_GEN.2	User identity association
FAU_SAR.1	Audit review
FAU_SAR.2	Restricted audit review
FAU_SAR.3	Selectable audit review
FAU_SEL.1	Selective audit
FAU_STG.1	Protected audit trail storage
FAU_STG.NIAP-0414	Site-Configurable Prevention of Audit Loss
FDP_ACC.1	Subset access control
FDP_ACF.1	Security attribute based access control
FIA_AFL.1	Authentication failure handling
FIA_ATD.1	User attribute definition
FIA_SOS.1	Verification of secrets
FIA_UAU.2	User authentication before any action
FIA_UID.2	User identification before any action
FMT_MOF.1	Management of security functions behaviour
FMT_MSA.1	Management of security attributes
FMT_MSA.2	Secure security attributes
FMT_MSA.3	Static attribute initialisation
FMT_MTD.1(1)	Management of TSF data (audit events)
FMT_MTD.1(2)	Management of TSF data (audit records)
FMT_MTD.1(3)	Management of TSF data (user authentication data)
FMT_REV.1(1)	Revocation (user attributes)
FMT_REV.1(2)	Revocation (subject, object attributes)
FMT_SMF.1	Specification of management functions
FMT_SMR.1	Security roles
FPT_ITD_EXP.1	SFP domain separation
FPT_RVM.1(1)	Non-bypassability of the TSP

Section 5.1 contains the functional components from the CC Part 2 with the operations completed. For the conventions used in performing CC operations please refer to Section 1.3.1.

5.1.1 Class FAU: Security Audit

FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

FAU_GEN.1.1

Refinement: The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events, **listed in Table 8,** for the *[not specified]* level of audit; and

c) [Start-up and shutdown of the DBMS;

d) Use of special permission (e.g., those often used by authorised administrators to circumvent access control policies);

e) Any standard audit report is run;

f) The list of events being audited is changed;

g) Audit records are erased or deleted;

h) The definition of a user, application, or role is created, changed, or deleted].

FAU_GEN.1.2

The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*the items listed in Table 8.]*

Dependencies: FPT_STM.1 Reliable time stamps

Table 8 - Auditable Events

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_SAR.1 Audit review	None	
FAU_SAR.2 Restricted Audit Review	None	
FAU_SAR.3 Selectable Audit Review	None	

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_SEL.1 Selective audit	All modifications to the audit configuration that occur while the audit collection functions are operating.	
FAU_STG.1 Protected audit trail storage	None	
FAU_STG.NIAP-0414 Site-Configurable Prevention of Audit Loss	Actions taken due to the audit storage failure. Selection of an action to be taken when there is an audit storage failure.	
FDP_ACC.1 Subset access control	None	
FDP_ACF.1 Security attribute based access control	Successful requests to perform an operation on an object covered by the Security Function Policy (SFP).	
FIA_AFL.1 Authentication failure handling	The reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g., disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g., re-enabling of a terminal).	
FIA_ATD.1 User attribute definition	None	
FIA_SOS.1 Verification of secrets	Rejection by the TSF of any tested secret.	
FIA_UAU.2 User authentication before any action	Unsuccessful use of the authentication mechanism.	Identity of the user or authorised administrator that entered the incorrect authentication data, but not the incorrect authentication data itself.
FIA_UID.2 User identification before any action	Unsuccessful use of the user identification mechanism, including the user identity provided.	Identification information entered.
FMT_MOF.1 Management of security functions behavior	None	
FMT_MSA.1 Management of security attributes	None	
FMT_MSA.2 Secure security attributes	All offered and rejected values for a security attribute.	
FMT_MSA.3 Static attribute initialization	None	
FMT_MTD.1 Management of TSF data	None	
FMT_REV.1 Revocation	Unsuccessful revocation of security attributes.	
FMT_SMF.1 Specification of management functions	None	

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FMT_SMR.1 Security roles	Modifications to the users that are part of a role.	
FPT_RVM.1 Non-bypassability of the TSP	None	
FPT_ITD_EXP.1 Internal TOE domains	None	
FPT_SEP.1 TSF domain separation	None	
FPT_STM.1 Reliable time stamps	None	

FAU_GEN.2 User identity association

Hierarchical to: No other components.

FAU_GEN.2.1

The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1

The TSF shall provide [*authorised users*] with the capability to read [*all database audit information*] from the audit records.

FAU_SAR.1.2

Refinement: The TSF shall provide the audit records in a manner suitable for the **authorised** user to interpret the information.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

FAU_SAR.2.1

The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

FAU_SAR.3.1

Refinement: The TSF shall provide the ability to perform [*searching **and/or**, sorting,*] of audit data based on

[a) User identity;

b) Date of event;

c) Time of event;

d) Type of event;

e) Event status (success/failure);

f) Event source;

g) Event data;

h) Process ID which logged the event;

i) User roles;

j) Routine that was being executed by the process when the event was logged;

k) Client's IP address;

l) Client's application identifier;

m) Free-test description of event].

Dependencies: FAU_SAR.1 Audit review

FAU_SEL.1 Selective audit

Hierarchical to: No other components.

FAU_SEL.1.1

The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) [*object identity, user identity, event type*]

b) [success of auditable security events, failure of auditable security events]

Dependencies: FAU_GEN.1 Audit data generation, FMT_MTD.1 Management of TSF data

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

FAU_STG.1.1

Refinement: The TSF shall restrict the deletion of audit records in the audit trail to the authorised administrator.

FAU_STG.1.2

The TSF shall be able to [*prevent*] unauthorised modifications to the audit records in the audit trail.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.NIAP-0414 Site-Configurable Prevention of Audit Loss

Hierarchical to: FAU_STG.4.

FAU_STG.NIAP-0414.1

The TSF shall provide the authorized administrator the capability to select one or more of the following actions to be taken if the audit trail is full: overwrite the oldest stored audit records, alert the authorized administrator.

FAU_STG.NIAP-0414.2

The TSF shall overwrite the oldest stored audit records and alert the authorized administrator if the audit trail is full and no other action has been selected.

Dependencies: FAU_STG.1 Protected Audit Trail Storage, FMT_MTD.1 Management of TSF Data

5.1.2 Class FDP: User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1

The TSF shall enforce the [*Discretionary Access Control*[1] *policy*] on [*all subjects, all DBMS-controlled objects and all operations among them*].

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1

The TSF shall enforce the [*Discretionary Access Control policy*] to objects based on the following: [

a) the authorized user identity associated with a subject, and

b) access operations implemented for DBMS-controlled objects].

FDP_ACF.1.2

Refinement: The TSF shall enforce the following rules to determine if an operation among controlled subjects and **DBMS-**controlled objects is allowed: [

The Discretionary Access Control policy mechanism shall, either by explicit authorized user action or by default, provide that database management system controlled objects are protected from unauthorized access according to the following ordered rules:

1) If the requested mode of access is permitted to that authorized user, permit access.

2) Else deny access].

FDP_ACF.1.3

Refinement: The TSF shall explicitly authorise access of subjects to **DBMS-controlled** objects based on the following additional rules: [*Authorized administrators must follow the above-stated Discretionary Access Control policy, except after starting the TOE in emergency recovery mode]*.

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the [*none*].

[1] Discretionary Access Control is often abbreviated as “DAC”.

Dependencies: **FDP_ACC.1 Subset access control**
FMT_MSA.3 Static attribute initialization

5.1.3 Class FIA: Identification and Authentication

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Interpretation Note: The following element was modified per Common Criteria Interpretations Management Board (CCIMB) Interpretation 111.

FIA_AFL.1.1

The TSF shall detect when [*[a configurable integer within a range chosen by the authorized administrator*] unsuccessful authentication attempts occur related to [all user authentication processes].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [***prevent the*** [entity requesting authorization] ***from performing activities that require authentication until an action is taken by the authorized administrator****]*.

Application Note: The %All role is excluded from the FIA_AFL.1.2 requirement in order to prevent denial of access. FIA_AFL.1 does not apply when Kerberos authentication is being performed.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: [

a) *Database user identifier;*

b) *Security-relevant database roles; and*

c) *Timestamp of most recent successful login (for locally authenticated sessions);*

d) *Device used for most recent successful login (for locally authenticated sessions);*

e) *Service used for most recent successful login (for locally authenticated sessions);*

f) *Number of invalid login attempts (for locally authenticated sessions);*

g) *Timestamp of most recent invalid login attempt (for locally authenticated sessions);*

h) *Service of most recent invalid login attempt (for locally authenticated sessions);*

i) *Device used for most recent invalid login attempt (for locally authenticated sessions);*

j) *Error thrown for most recent invalid login attempt (for locally authenticated sessions)]*.

Dependencies: No dependencies

FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

FIA_SOS.1.1

Refinement: The TSF shall provide a mechanism to verify that secrets meet **the following** [

a) For each attempt to use the authentication mechanism, the probability that a random attempts will succeed is less that one in $5x10^{15}$; and

b)Any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics].

Dependencies: No dependencies

FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1

FIA_UAU.2.1

The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: FIA_UID.1 Timing of identification

Dependencies: No dependencies

FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1

FIA_UID.2.1

The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Dependencies: No dependencies

5.1.4 Class FMT: Security Management

FMT_MOF.1 Management of security functions behaviour

Hierarchical to: No other components.

FMT_MOF.1.1

The TSF shall restrict the ability to [*disable and enable*] the functions [*relating to the specification of events to be audited*] to [*the authorized administrators*].

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1

Refinement: The TSF shall enforce the [*Discretionary Access Control policy*] to restrict the ability to [*manage*] the security attributes **of database users** to [authorized administrators].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.2 Secure security attributes

Hierarchical to: No other components.

FMT_MSA.2.1

The TSF shall ensure that only secure values are accepted for security attributes.

Dependencies: ADV_SPM.1 Informal TOE security policy model
[FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA.3.1

The TSF shall enforce the [*Discretionary Access Control policy*] to provide [*restrictive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the [*authorized administrator*] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MTD.1(1) Management of TSF data (audit events)

Hierarchical to: No other components.

FMT_MTD.1.1

The TSF shall restrict the ability to [***include or exclude***] the [*auditable events*] to [*authorised administrators*].

Dependencies: FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles

FMT_MTD.1(2) Management of TSF data (audit records)

Hierarchical to: No other components.

FMT_MTD.1.1

The TSF shall restrict the ability to [*query and clear*] the [*audit records*] to [the *authorised administrators*].

Dependencies: FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles

FMT_MTD.1(3) Management of TSF data (user authentication data)

Hierarchical to: No other components.

FMT_MTD.1.1

The TSF shall restrict the ability to [***set and reset***] the [*user authentication data*] to [the *authorised administrators*].

Dependencies: FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles

FMT_REV.1(1) Revocation (user attributes)

Hierarchical to: No other components.

FMT_REV.1.1

The TSF shall restrict the ability to revoke security attributes associated with the [*users*] within the TSC to [*the authorised administrators*].

FMT_REV.1.2

The TSF shall enforce the rules [*none*].

Dependencies: FMT_SMR.1 Security roles

FMT_REV.1(2) Revocation (subject, object attributes)

Hierarchical to: No other components.

FMT_REV.1.1

The TSF shall restrict the ability to revoke security attributes associated with the [*subjects and objects*] within the TSC to [*the authorised administrators*].

FMT_REV.1.2

The TSF shall enforce the rules [*none*].

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of management functions

Hierarchical to: No other components.

FMT_SMF.1.1

The TSF shall be capable of performing the following security management functions: [*user management; audit management; database management; and discretionary access control management*].

Dependencies: No Dependencies

FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1

The TSF shall maintain the roles [

a) *authorised administrator; and*

b) *Operator (user);*

c) *SQL (user)]*.

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

5.1.5 Class FPT: Protection of the TSF

FPT_ITD_EXP.1 SFP domain separation

Hierarchical to: No other components.

FPT_ITD_EXP.1.1

The TSF shall enforce separation between the security domains of subjects in the TOE Scope of Control.

Dependencies: No dependencies

FPT_RVM.1(1) Non-bypassability of the TSP

Hierarchical to: No other components.

FPT_RVM.1(1).1

The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

Dependencies: No dependencies

5.2 Security Functional Requirements on the IT Environment

The TOE has the following security requirements for its IT environment. The stated SFRs on the IT Environment of the TOE presented in this section have been drawn from and are conformant to Part 2 of the CC Version 2.2.

Table 9 - IT Environmental Security Functional Requirements

ID	Functional Component
FPT_RVM.1(2)	Non-bypass ability of the TSP
FPT_SEP.1	TSF domain separation
FPT_STM.1	Reliable time stamps

FPT_RVM.1(2) Non-bypassability of the TSP

Hierarchical to: No other components.

FPT_RVM.1(2).1 **Refinement:** The **host OS security functions** shall ensure that **host OS security policy** enforcement functions are invoked and succeed before each function within the **scope of control of the host OS** is allowed to proceed.

Dependencies: No dependencies

FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

FPT_SEP.1.1 **Refinement:** The **security functions of the host OS** shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 **Refinement:** The **security functions of the host OS** shall enforce separation between the security domains of subjects in the **scope of control of the host OS**.

Dependencies: No dependencies

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

FPT_STM.1.1 **Refinement:** The **security functions of the host OS** shall be able to provide reliable time stamps for its own use **and for the TOE**.

Dependencies: No dependencies

5.3 Assurance Requirements

This section defines the EAL3 assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3 and are summarized in Table 10 below.

Table 10 - Assurance Requirements

Assurance Requirements	
Class ACM: Configuration management	ACM_CAP.3 Authorisation controls
	ACM_SCP.1 TOE CM coverage
Class ADO: Delivery and operation	ADO_DEL.1 Delivery procedures
	ADO_IGS.1 Installation, generation, and start-up procedures
Class ADV: Development	ADV_FSP.1 Informal functional specification
	ADV_HLD.2 Security enforcing high-level design
	ADV_RCR.1 Informal correspondence demonstration
Class AGD: Guidance documents	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Class ALC : Life Cycle Support	ALC_DVS.1 Identification of security measures
Class ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: high-level design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Class AVA: Vulnerability assessment	AVA_MSU.1 Examination of guidance
	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.1 Developer vulnerability analysis

6 TOE and TOE Environment Summary Specification

This section presents information to detail how the TOE and the TOE Environment meets the functional and assurance requirements described in previous sections of this ST.

6.1 TOE Security Functions Summary

Each TOE security function is described below and related to the security requirements it satisfies. This serves to both describe the security functions and rationalize that the security functions are suitable to satisfy the necessary requirements.

Table 11 - Mapping of TOE Security Functions to Security Functional Requirements

TOE Security Function	SFR Identifier	Description
Security Audit	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_SAR.1	Audit review (TOE)
	FAU_SAR.2	Restricted Audit Review
	FAU_SAR.3	Selectable audit review (TOE)
	FAU_SEL.1	Selective audit
	FAU_STG.1	Protected audit trail storage (TOE)
	FAU_STG.NIAP-0414	Site-Configurable Prevention of Audit Loss
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
Identification and Authentication	FIA_AFL.1	Authentication failure handling
	FIA_ATD.1	User attribute definition
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	Timing of authentication (TOE)
	FIA_UID.2	Timing of identification (TOE)
Security Management	FMT_MOF.1	Management of security functions behavior
	FMT_MSA.1	Management of DAC security attributes
	FMT_MSA.2	Secure security attributes
	FMT_MSA.3	Static attribute initialization
	FMT_MTD.1(1)	Management of TSF data (audit events)
	FMT_MTD.1(2)	Management of TSF data (audit records)
	FMT_MTD.1(3)	Management of TSF data (user authentication data)
	FMT_REV.1(1)	Revocation (user attributes)
	FMT_REV.1(2)	Revocation (subject, object attributes)
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles

TOE Security Function	SFR Identifier	Description
Protection of the TSF	FPT_ITD_EXP.1	Internal TOE domains
	FPT_RVM.1(1)	Non-bypassability of the TSP

6.1.1 Security Audit

The TOE logs key events in a secure audit log, which is stored in the database CACHEAUDIT located inside the TOE. Only audit records are stored within this database and the TOE protects this database from tampering. A special auditing application programming interface (API) must be used to access the audit database, and existing records cannot be individually modified or deleted (though new entries can be appended to the end). The database is exposed to the SQL environment in "read-only/append-only" mode. Because of this, individual records cannot be deleted or edited from the SQL environment once they have been initially created. The only way the audit log can be cleared is by the administrator via the protected auditing API. However, clearing the audit log will be recorded as an audited event after the log has been cleared. The audit log can be backed up and otherwise managed like any other database by an authorized administrator.

The audit log can contain a finite number of audit records, and when the audit log contains the maximum number of audit records, the audit engine alerts the authorized administrator that the audit log is full and overwrites the oldest audit records with new audit records. This will continue until the authorized administrator creates room in the audit log either by clearing the log or by expanding the maximum size of the audit log. The authorized administrator can review the audit records in the audit log via the special auditing API that can be accessed either via direct SQL "select" commands or via one of the Management graphical user interfaces (GUIs). The TOE allows the authorized administrator to search and sort the audit records.

The TOE audit records contain the following information:

Table 12 - Audit Record Contents

Service	Event	Occurs	Event Data Contains	Mandatory/ Optional
%System	Start	Caché starts	Indication of whether recovery was performed	Mandatory
	ConfigurationChange	Caché successfully starts with a configuration different than the previous start, or new configuration is activated while Caché is running	User name	Mandatory
	Stop	Caché is shut down		Mandatory
%Login	Login	Successful login		Optional
	LoginFailure	Unsuccessful login attempt	User name	Optional
%Security	UserChange	Definition of a user created, changed, or deleted	Action (new, modify, delete), old and new user data	Mandatory
	ApplicationChange	Definition of an application created, changed, or deleted	Action (new, modify, delete), old and new application data	Mandatory
	RoleChange	Definition of a role created, changed, or deleted	Action (new, modify, delete), old and new role data	Mandatory

Service	Event	Occurs	Event Data Contains	Mandatory/ Optional
	AuditChange	Audit is stopped or started, entries are erased or deleted, or the list of events being audited is changed	Action (stop, start, erase, delete, specify), old and new audit settings	Mandatory
	Protect	A security protection error is given to a process	Error	Optional
	AuditReport	Any standard audit report is run	Identification of audit report	Mandatory

TOE Security Functional Requirements Satisfied: [FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_SEL.1, FAU_STG.1, FAU_STG.NIAP-0414].

6.1.2 User Data Protection

There are three different data access permissions that the TOE can grant to users: *read*, *write*, and *use*. Permissions must be explicitly granted, else they will be implicitly denied.

Table 13 - TOE Permissions

Permission	Typical Usage
Read	View (but not change) the contents of a resource
Write	View or change the contents of a resource
Use	Use a resource, such as an application or service

The meaning of each permission is dependent upon the resource with which it is used, as defined below. The TOE defines a set of resources that it protects, and allows TOE applications developed within the TOE's development environment to define application-specific resources. Application resources are not covered by this ST.

6.1.2.1 Database Resources

Database (DB) Resources control access to the contents of TOE databases. Database items (such as routines or globals) are assigned to Resource groups. Permissions are then granted to those general groups instead of the individual items. SQL table-level access control is also provided by the TOE.

In order for a user to have effective access to the database, the user must also have relevant services privileges. At least one service must be granted in order for an effective connection to be established, otherwise the user will not be able to view or modify data in that database. Database privileges are defined Table 14 and Service privileges are defined in Table 15.

Table 14 - Database Privileges

Resource	Permission	Enables
%DB/<Database Resource Name>	Read	Data access and routine execution
	Write	Modification of data

6.1.2.2 Service Privileges

Service privileges control the ability to connect to the TOE using various TOE connection technologies. Some services involve connecting to a local system, while other services are network based. Service permissions only

allow the opportunity to connect to the database, they do not grant a user or a resource any rights on the data stored in the database. In order for a Service privilege to be effective, the user must also hold relevant Database privileges. A list of the service privilege allowed by Caché is defined below in Table 15:

Table 15 - Services Privileges

Resource	Permission	Enables
%Service/CacheDirect	Use	Connection to Caché via Caché Direct
%Service/CallIn	Use	Connection to Caché via call-in
%Service/ComPort	Use	Connection to Caché via Windows COM ports
%Service/Console	Use	Connection to Caché on Windows systems via CSESSION or CSS
%Service/CSP	Use	Connection to Caché via Caché Server Pages
%Service/LAT	Use	Connection to Caché via the Caché LAT service for Windows
%Service/Object	Use	Connection to Caché via Caché object or SQL client and execute object requests
%Service/SQL	Use	Connection to Caché via Caché object or SQL client and execute SQL requests
%Service/Telnet	Use	Connection to Caché via the Caché Telnet service for Windows
%Service/Terminal	Use	Connection to Caché via terminal on non-Windows systems

6.1.2.3 Administrative Privileges

Administrative privileges enable a user to perform designated TOE administration tasks. These privileges are checked regardless of the interface or tool used to carry out the administrative functions. If a user holds administrative privileges the system will not check for additional database privileges. Passwords are never visible even to users with the %Admin/Secure:Use privilege; however, an authorized administrator can change a password or require that a new password be entered the next time a user logs in.

Table 16 - Administrative Privileges

Resource	Permission	Enables
%Admin/Manage	Use	• Create/modify/delete TOE configurations • Create/modify/delete backup definitions • Add/modify characteristics/delete TOE databases • Modify namespace map • Perform database and journal restores
%Admin/Operate	Use	• Start / stop the TOE • Examine / terminate processes • Mount / dismount databases • Perform integrity checks • Start / stop / switch journals • Perform database backups • Examine / delete locks • Examine logs • Start / stop services
%Admin/Secure	Use	• Create / modify / delete users • Create / modify / delete roles • Create / modify / delete application definitions and application resources • Modify audit settings

6.1.2.4 Development Privileges

Development privileges control access to TOE development facilities. The TOE is not only a database but also a database-application development platform and environment, and as such provides a range of TOE-specific application development and debugging facilities. These development facilities and the associated development privilege tree are not included in this evaluation.

TOE Security Functional Requirements Satisfied: [FDP_ACC.1, FDP_ACF.1].

6.1.3 Identification and Authentication

There are three authentication methods in the Caché system which can be used to access the TOE:

- Caché Simple Passwords
- Kerberos
- Local OS

Caché Simple Passwords and Kerberos can provide authentication for local and network connections. The Local OS authentication method can be used to authenticate a user connecting to Caché from the local system. The user's account will be locked after the maximum allowed number of failed authentication attempts has been surpassed; however, if a user is assigned the *%All* role then the account will not be locked. Users must be identified and authenticated before any other TSF mediated action is allowed.

For authentication using Caché Simple Passwords, the TOE maintains a password hash for each user account and compares that hash to a hash of the password provided by the user at each login. If a set of 94 possible characters is used (which would consist of: upper case letters, lower case letters, numbers, and other printable special characters) to meet the requirement as stated in FIA_SOS.1 and the Strength of Function (SOF) claims, the passwords must be a minimum of six characters long; by default Caché requires that passwords be a minimum of eight characters long.

The Local OS authentication method provides authentication for local users. For Local OS authentication, when a user on the local system attempts to log into Caché, Caché retrieves the local user's local OS account name and determines whether or not that user name matches a TOE user name. If the local OS user name is found in the list of authorized TOE users then the user is granted access and no further authentication occurs.[2]

Kerberos authentication is used when "strong authentication" is desired. For Kerberos authentication, the TOE uses the industry-standard Kerberos protocol to enable clients and the TOE to identify each other. The Kerberos protocol provides a centralized key management architecture where users identify themselves to a Kerberos authentication server (AS) which provides the user with an encrypted ticket-granting ticket. Only the authorized user can decrypt and use the ticket from the AS. The user then uses the ticket-granting ticket to obtain a service ticket for the TOE from a Kerberos ticket-granting server. The TOE then examines the service ticket to verify the identity of the user. Once the user is verified via his Kerberos ticket, the TOE then allows or denies access to functionality of the TOE based upon the privileges assigned to that user by the authorized administrator. Detailed information about the industry-standard Kerberos protocol is available in the public domain. The official Kerberos page can be located at http://web.mit.edu/kerberos/www/.

Each TOE user account has the following properties:

[2] This authentication method assumes that the local OS and server upon which the TOE operates are secured and protected in accordance with the assumptions listed previously in this ST.

Table 17 - User Account Properties

Field Name	Content
Name	Can include any characters except the @, which is used to identify a domain. User names are case insensitive.
FullName	Displayable name
Comment	Descriptive text
Enabled	Flag indicating whether or not the account is currently enabled
Expiration Date	Indicates last date that the account can be used
Roles	Comma-separated list of roles assigned to user
Terminal Namespace	Namespace in which to begin execution following a log in from a terminal-type service. This property overrides any namespace value provided via the command invoking Caché. Default: User
Terminal Routine	Routine to execute automatically following a log in from a terminal-type service. This property overrides any routine value provided via the command invoking Caché.
LastLogin Timestamp	Date and time of last successful login (prior to the current session) or 0 if this is the first successful login [Read-Only]
LastLogin Device	Device used for last successful login (prior to the current session) or "" if this is the first successful login [Read-Only]
LastLogin Service	Service used for last successful login (prior to the current session) or "" if this is the first successful login [Read-Only]
InvalidLogin Attempts	Number of invalid login attempts since the last successful login [Read-Only]
InvalidLogin Timestamp	Date and time of most recent invalid login attempt [Read-Only]
InvalidLogin Service	Service used for most recent invalid login attempt [Read-Only]
InvalidLogin Device	Device used for most recent invalid login attempt [Read-Only]
InvalidLogin Status	Error thrown for most recent invalid login attempt [Read-Only]

TOE Security Functional Requirements Satisfied: [FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UID.2]

6.1.4 Security Management

Correct security management of the TOE requires that Caché is correctly installed and all appropriate maintenance or configuration changes are preformed on the TOE. These maintenance activities are performed by identified and authenticated administrators, who access the TOE indirectly via one of the Management GUIs.

TOE Administrators have complete access to the TOE. Some TOE Administrators have additional privileges in the host OS for initial installation of the TOE and exceptional events (e.g. maintenance of the file system or disaster recovery), and all TOE Administrators have indirect access to the TOE via the Management GUIs.

An Administrator can create users at the same or lower levels of access. These accounts will be created with secure default values and privileges, and an authorized administrator can override the default values if necessary.

The product provides functionality that allows administrators with appropriate privileges to control all aspects of the operation of the product. In summary these are: starting and stopping the TOE, managing the TOE, defining and distributing security policy to the database instances, management of audit logs and auditable events, and monitoring the status of the TOE components.

TOE Security Functional Requirements Satisfied: [FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1(1), FMT_MTD.1(2), FMT_MTD.1(3), FMT_REV.1(1), FMT_REV.1(2), FMT_SMF.1, FMT_SMR.1].

6.1.5 Protection of the TSF

Non-bypassability of the TOE is provided by a combination of the basic configuration and enforcement of the security policy rules. The assumed secure basic configuration maintaining physical and logical isolation supports the protection of security functions. The functions that enforce the TSP will always be invoked before any function within the TSF Scope of Control is allowed to proceed. The TOE manages and keeps separate the information space, command space, and memory space of each user session. The design of the TOE architecture and functionality makes it impossible for a user to (purposefully or accidentally) obtain or affect the information space, command space, or memory space of another user. Users with administrative privileges may affect the permissions of other users through use of their administrative privileges to assign or revoke another user's privileges, and a user may interact with databases (for which he has privileges) for which another user also has privileges, but these types of interactions are controlled by the TSF and approved by the TSP.

The architecture of the TOE is such that each incoming command must be processed by the permissions engine before it can be executed. The permissions engine determines whether or not the user requesting execution of the command has the required privileges to execute that command on the specified element – if not, then the execution request is rejected, the execution attempt is logged, and the requested command is discarded; otherwise, if the user has the required privileges, then the command is executed.

No general purpose operating system, programming interfaces or external disk storage is provided by the TOE. Caché maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects. The underlying assumption regarding the operation of the TOE is that it is maintained in a physically secure environment. Furthermore, in order to ensure the correct execution of each process, the OS protects each process' private information (executable code, data, and stack) from uncontrolled interference by other processes. These features ensure that the TSF maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

TOE Security Functional Requirements Satisfied: [FPT_ITD_EXP.1, FPT_RVM.1(1)].

TOE Environment Security Functional Requirements Satisfied: [FPT_RVM.1(2), FPT_SEP.1, FPT_STM.1].

6.2 TOE Security Assurance Measures

EAL3 was chosen to provide a basic level of independently assured security. This section of the Security Target maps the assurance requirements of the TOE for a CC EAL3 level of assurance to the assurance measures used for the development and maintenance of the TOE. The following table provides a mapping of the appropriate documentation to the TOE assurance requirements.

Note to Evaluator: The final versions of these documents have not yet been produced. The version numbers will be completed when the evaluation is close to completion and the documents have been finalized.

Table 18 - Assurance Measures Mapping to TOE SARs

Assurance Component	Assurance Measure
ACM_SCP.1	InterSystems Caché v5.1.0.826.0 – Configuration Management
ACM_CAP.3	
ADO_DEL.1	InterSystems Caché v5.1.0.826.0 – Secure Delivery

Assurance Component	Assurance Measure
ADO_IGS.1	Installation and Setup Procedure
ADV_FSP.1	InterSystems Caché v5.1.0.826.0 – TOE Architecture: High Level Design, Functional Specification, and Representation Correspondence
ADV_HLD.2	InterSystems Caché v5.1.0.826.0 – TOE Architecture: High Level Design, Functional Specification, and Representation Correspondence
ADV_RCR.1	InterSystems Caché v5.1.0.826.0 – TOE Architecture: High Level Design, Functional Specification, and Representation Correspondence
AGD_ADM.1	Administrator Guides
AGD_USR.1	User Guides
ALC_DVS.1	InterSystems Caché v5.1.0.826.0 – Development Security
ATE_COV.2	InterSystems Caché v5.1.0.826.0 – Functional Tests and Coverage
ATE_DPT.1	
ATE_IND.2	Performed by Laboratory
ATE_FUN.1	InterSystems Caché v5.1.0.826.0 – Functional Tests and Coverage
AVA_SOF.1	InterSystems Caché v5.1.0.826.0 – Vulnerability Assessment
AVA_VLA.1	
AVA_MSU.1	

6.2.1 ACM_CAP.3, ACM_SCP.1: Configuration Management Document

The Configuration Management document provides a description of the various tools used to control the configuration items and how they are used internally at InterSystems. This document provides a complete configuration item list and a unique referencing scheme for each configuration item. Additionally, the configuration management system is described including procedures that are used by developers to control and track changes that are made to the TOE. The documentation further details the TOE configuration items that are controlled by the configuration management system.

6.2.2 ADO_DEL.1: Delivery and Operation Document

The Delivery and Operation document provides a description of the secure delivery procedures implemented by InterSystems to protect against TOE modification during product delivery. The Installation Documentation provided by InterSystems details the procedures for installing the TOE and placing the TOE in a secure state offering the same protection properties as the master copy of the TOE. The Installation Documentation provides guidance to the TOE Users(s) on configuring the TOE and how they affect the TSF.

6.2.3 ADO_IGS.1: Installation Guidance, AGD_ADM.1: Administrator Guidance, AGD_USR.1: User Guidance

The installation guidance document provides the procedures necessary for the secure installation, generation, and start-up of the TOE for administrators and users of the TOE.

The administrator guidance documentation provides detailed procedures for the administration of the TOE and description of the security functions provided by the TOE.

The User Guidance documentation provided directs users on how to operate the TOE in a secure manner. Additionally, User Guidance explains the user-visible security functions and how they need to be exercised.

6.2.4 ADV_FSP.1: Informal Functional Specification, ADV_HLD.2: High Level Design, ADV_RCR.1: Representation Correspondence.

The InterSystems design documentation consists of several related design documents that address the components of the TOE at different levels of abstraction. The following design documents address the Development Assurance Requirements:

- The Functional Specification provides a description of the security functions provided by the TOE and a description of the external interfaces to the TSF. The Functional Specification covers the purpose and method of use and a list of effects, exceptions, and errors message for each external TSF interface.
- The High-Level Design provides a top level design specification that refines the TSF functional specification into the major constituent parts (subsystems) of the TSF. The high-level design identifies the basic structure of the TSF, the major elements, a listing of all interfaces, and the purpose and method of use for each interface.
- The Representation Correspondence demonstrates the correspondence between each of the TSF representations provided. This mapping is performed to show the functions traced from the ST description to the High-Level Design.

6.2.5 ATE_COV.2: Analysis of Coverage, ATE_DPT.1: Testing: High-Level Design, ATE_FUN.1: Functional Testing

There are a number of components that make up the Test documentation. The Coverage Analysis demonstrates that testing is performed against the functional specification. The Coverage Analysis demonstrates the extent to which the TOE security functions were tested as well as the level of detail to which the TOE was tested. The testing Depth analysis provides assurance that the TSF subsystems have been correctly realized and that no subsystem flaws are present. Test Plans and Test Procedures, which detail the overall efforts of the testing effort and break down the specific steps taken by a tester, are also provided in order to meet the assurance requirement Functional Testing.

6.2.6 AVA_MSU.1: Misuse, AVA_SOF.1: Strength of Function Analysis, AVA_VLA.1: Vulnerability Analysis

The Misuse documentation investigates whether the TOE can be configured or used in a manner that is insecure but that an administrator or user of the TOE would reasonably believe to be secure.

The Strength of TOE Security Function Analysis demonstrates the strength of the probabilistic or permutational mechanisms employed to provide security functions within the TOE and how they exceed the minimum SOF requirements.

A Vulnerability Assessment is provided to demonstrate ways in which an entity could violate the TSP and provide a list of identified vulnerabilities. Additionally, this document provides evidence of how the TOE is resistant to obvious attacks.

7 Protection Profile Claims

This section provides the identification and justification for any Protection Profile conformance claims.

7.1 Protection Profile Reference

There are no protection profile claims for this Security Target.

8 Rationale

This section provides the rationale for the selection of the security requirements, objectives, assumptions, and threats. In particular, it shows that the security requirements are suitable to meet the security objectives, which in turn are shown to be suitable to cover all aspects of the TOE security environment.

8.1 Security Objectives Rationale

This section provides a rationale for the existence of each assumption and threat that compose the Security Target. Table 19 demonstrates the mapping between the assumptions and threats to the security objectives is complete. The following discussion provides detailed evidence of coverage for each assumption and threat.

Table 19 - Relationship of Security Threats to Objectives

Threats, Assumptions \ Objectives		O.ACCESS	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_PROTECTION	O.AUDIT_REVIEW	O.DISCRETIONARY_ACCESS	O.INSTALL	O.MANAGE	O.INTERNAL_TOE_DOMAINS	O.PROTECT	O.SOUND_DESIGN	O.SOUND_IMPLEMENTATION	O.TESTING	O.TRAINED_USERS	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	OE.CONFIG	OE.NO_EVIL	OE.NO_GENERAL_PURPOSE	OE.PHYSICAL	OE.ROBUST_ENVIRONMENT	OE.SECURE_COMMS	OE.SELF_PROTECTION	OE.TOE_PROTECTION	OE.TIME	OE.TRUST_IT
Threats	T.ADMIN_ERROR		✓						✓	✓										✓								
	T.AUDIT_COMPROMISE				✓	✓																✓	✓		✓			
	T.INSECURE_DELIVERY		✓						✓										✓	✓								
	T.INSECURE_START		✓							✓																		
	T.MASQUERADE																✓	✓		✓				✓				
	T.POOR_DESIGN												✓		✓													
	T.POOR_IMPLEMENTATION													✓	✓													
	T.POOR_TEST														✓													
	T.SYSACC	✓	✓							✓							✓	✓		✓		✓						
	T.TSF_COMPROMISE																					✓	✓	✓		✓		✓
	T.UNATTENDED_SESSION	✓										✓				✓	✓											
	T.UNAUTH_ACCESS	✓		✓				✓			✓	✓										✓	✓		✓			
	T.UNDETECTED_ACTIONS				✓	✓																✓	✓				✓	
	T.UNIDENTIFIED_ACTIONS		✓				✓			✓																		
Assumptions	A.NO_EVIL																		✓	✓								
	A.NO_GENERAL_PURPOSE																				✓							
	A.PHYSICAL																					✓						
	A.ROBUST_ENVIRONMENT																						✓					✓
	A.SECURE_COMMS																							✓				

T. ADMIN_ERROR

An authorized administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms

Improper administration could result if the authorized administrator is unknowledgeable or if the TOE does not provide the proper administration tools. There is always the possibility that the administrator will make an honest mistake.

This threat will be mitigated as long as the TOE provides the necessary administrator support (O.MANAGE) and the authorized administrator is provided with knowledge necessary to carry out administrative duties (O.ADMIN_GUIDANCE). The authorized administrator is provided with necessary installation instructions from the developer that details how to securely install the TOE (O.INSTALL). The authorized administrator must not act in a malicious manner against the system (OE.NO_EVIL).

T.AUDIT_COMPROMISE

A process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker's actions.

Not recording auditable events could also make it harder to determine if a security breach has occurred or if there is a weakness in the system. Without the log of auditable events changes to the system configuration would be harder to detect.

The TOE will generate an audit log (O.AUDIT_GENERATION). The environment must address the possible compromise of audit data due to physical means (OE.PHYSICAL). The IT environment must also protect itself and its assets (OE.SELF_PROTECTION). The TOE shall only be installed in an IT environment that is at least as robust as the TOE (OE.ROBUST_ENVIRONMENT). The TOE must also provide protection for its audit data (O.AUDIT_PROTECTION).

T.INSECURE_DELIVERY

The authorized administrator may receive the delivered TOE without the appropriate installation guidance, resulting in the improper installation or configuration of the TOE.

This threat is addressed by ensuring the appropriate installation guidance necessary to properly and securely install the TOE is provided (O.INSTALL), and that authorized administrators performing the installation have adequate knowledge on how to install the TOE properly and securely (O.ADMIN_GUIDANCE). Care must be taken when installing the TOE to ensure the configuration settings are as specified in the installation guidance for proper, secure installation (OE.CONFIG). The authorized administrator must not act in a malicious manner against the system (OE.NO_EVIL).

T.INSECURE_START

An authorized administrator may configure the TOE in such a way that a reboot will result in insecure state of the TOE.

This threat is addressed by ensuring that the authorized administrators have the knowledge necessary to start the system in a secure state (O.ADMIN_GUIDANCE and O.MANAGE).

T.MASQUERADE

An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.

Addressing the threat of a process or user masquerading as a different process or user produces an objective of uniquely identifying each user (O.USER_IDENTIFICATION). Unique user identification must be supported by the objective of requiring all users of the TOE to prove their claimed identity (O.USER_AUTHENTICATION). The environment must provide a secure line of communication for transfer of the authentication information (OE.SECURE_COMMS) and the authorized administrator must not act in a malicious manner against the system (OE.NO_EVIL).

T.POOR_DESIGN

The TOE developers may cause unintentional or intentional errors in the requirement specification, design, or development of the TOE may occur.

Bugs may appear at some point in the design or development of the system. These bugs could be due to a mistake in coding or production.

Faults in the TOE's design can be reduced by eliminating errors in the design through the use of sound design principles and documentation of the TOE design (O.SOUND_DESIGN). Design flaws can be mitigated through discovery resulting from testing the implementation (O.TESTING).

T.POOR_IMPLEMENTATION

The TOE developers may cause unintentional or intentional errors while implementing the design of the TOE.

Testing the security functions of the TOE (O.TESTING) can discover implementation errors and show whether the implementation is a faithful instantiation of its design (O.SOUND_IMPLEMENTATION).

T.POOR_TEST

Lack of or insufficient testing by the TOE developers to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.

This threat deals with the sufficiency of security tests to show that the TOE security functions behave correctly. Addressing this threat requires the developer to demonstrate that adequate testing methods are used that exercise security features. (O.TESTING).

T.SYSACC

A malicious process or user may gain unauthorized access to the authorized administrator account, or that of other trusted personnel.

If a malicious process or user was able to gain unauthorized access to the system, they may be able to modify the system, the system's auditing functions, or perform some other action that would cause the TOE to enter an insecure state.

The threat of the wrong individual gaining unauthorized access to the authorized administrator's account (O.ACCESS) may be addressed by physical means (OE.PHYSICAL), such as in cases where the authorized administrator console is behind a locked door. For other cases, the threat may be mitigated by requiring the authorized administrator to be uniquely identified (O.USER_IDENTIFICATION) and authenticated (O.USER_AUTHENTICATION). Authorized administrators will have to know (O.ADMIN_GUIDANCE) to check this information at each login. The authorized administrator must also be aware that he/she must protect the authentication information that allows access to the authorized administrator account (O.ADMIN_GUIDANCE). The TOE will provide mechanisms for the authorized administrator to set the security attributes

for users so they are not allowed admin access (O.MANAGE). The authorized administrator must not act in a malicious manner against the system (OE.NO_EVIL).

T.TSF_COMPROMISE

A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified, or deleted).

The IT environment will protect the TSF data and executable code from a compromise through physical protection on the site where the computer is located. (OE.PHYSICAL). The TOE shall only be installed in an IT environment that is at least as robust as the TOE. (OE.ROBUST_ENVIRONMENT). The TSF data and executable code is protected under the environmental objective for TOE protection (OE.TOE_PROTECTION) and the environment must provide a secure line of communication for transfer of the authentication information (OE.SECURE_COMMS). The IT entities in the environment are correctly installed, configured, managed and maintained (OE.TRUST_IT).

T.UNATTENDED_SESSION

A user may gain unauthorized access to an unattended session.

If a user was to leave their computer unattended an unauthorized user could go to their location and masquerade as the authorized user.

Unattended sessions must be protected (O.PROTECT) from unauthorized access (O.ACCESS). The TOE must meet objectives for detecting when sessions are unattended and preventing access to those sessions, unless the user re-authenticates. This might be accomplished by simply alerting users that they must not leave sessions unattended (O.TRAINED_USERS) or by requiring users to re-authenticate themselves (O.USER_AUTHENTICATION) after returning to the unattended session.

T.UNAUTH_ACCESS

A user may gain unauthorized access (view, modify, delete) to user data.

The threat of unauthorized physical access is addressed by the environment (OE.PHYSICAL). Logical unauthorized access is mitigated by protecting user data and access to the TOE. (O.PROTECT). The TOE must satisfy the objective of ensuring that only authorized users may gain access to the TOE and the resources it protects, and that users are not allowed to access protected data for which they are not authorized (O.ACCESS). Access to TSF data is controlled by a discretionary policy (O.DISCRETIONARY_ACCESS). The discretionary policy will be maintained by an authorized administrator. (O.ADMIN_ROLE) The TOE maintains internal domains to keep data and processes of concurrent users separate, so users cannot observe or interfere with other users' data or queries (O.INTERNAL_TOE_DOMAINS). The IT environment must also protect itself and its assets (OE.SELF_PROTECTION). The TOE shall only be installed in an IT environment that is at least as robust as the TOE (OE.ROBUST_ENVIRONMENT).

T.UNDETECTED_ACTIONS

Failure of the IT operating system to detect and record unauthorized actions may occur.

If unauthorized access occurs, and is not detected, a user could modify TOE security functions and bring the TOE into an insecure state. If unauthorized access occurs, and it is not recorded, it will be considerably harder to determine the time and method of the breech.

The threat of undetected physical manipulation of the TOE is addressed by the physical protection in the environment (OE.PHYSICAL). The protection applied to the IT environment must be at

least as strong as the level of security maintained inside the TOE (OE.ROBUST_ENVIRONMENT). Other actions are detected and a record is made (O.AUDIT_GENERATION) including timestamps (OE.TIME). However, it is important to understand that since this evaluation is at the Basic Robustness level, only the minimum level of audit generation is required, which is commensurate with Basic Robustness. To prevent removing evidence of unauthorized actions, the audit records need to be protected from unauthorized modification (O.AUDIT_PROTECTION).

T.UNIDENTIFIED_ACTIONS

The authorized administrator may fail to identify and act upon unauthorised actions.

The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the facilities (O.MANAGE) to review audit records (O.AUDIT_REVIEW) and knowing how to do so (O.ADMIN_GUIDANCE).

A.NO_EVIL

Authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance.

All authorized administrators are trustworthy individuals, having background investigations commensurate with the level of data being protected, have undergone appropriate admin training, and follow all admin guidance (OE.NO_EVIL). Authorized administrators are trusted to properly configure the TOE so it enforces its security policies (OE.CONFIG).

A.NO_GENERAL_ PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on Database Management System (DBMS) servers, other than those services necessary for the operation, administration, and support of the DBMS.

The DBMS server must not include any general-purpose commuting or storage capabilities (OE.NO_GENERAL_PURPOSE). This will protect the TSF data from malicious processes.

A.PHYSICAL

It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment (OE.PHYSICAL).

A.ROBUST_ENVIORNMENT

It is assumed that the IT environment is at least as robust as the TOE.

The TOE shall only be installed in an IT environment that is at least as robust as the TOE. The TOE is basic robustness, therefore, all elements in the environment the TOE depends on for enforcement of its security objectives are also assumed to be basic robustness. These elements could include the operating system, encryption devices, and/or boundary protection devices (OE.ROBUST_ENVIORNMENT).

The IT entities in the environment are correctly installed, configured, managed and maintained (OE.TRUST_IT).

A.SECURE_COMMS

It is assumed that the IT environment will provide components to support secure data communications.

The environment must provide a secure line of communication for transfer of TSF data (OE.SECURE_COMMS). This is necessary because the TOE may be distributed geographically with users and authorized administrators in different locations. It may also be the case that the TOE is a distributed architecture, with database servers in different geographic locations.

The objective OE.SECURE_COMMS does not necessarily mandate that the communications between the remote administrator and the TOE be encrypted. Remote administration implies administration from any location other than the TOE console. In many implementations, remote administration will be done from another workstation on the same LAN as the TOE, but within a protected enclave. In this case, there is no need for cryptographic protection of the communications between the authorized administrator and the TOE.

8.2 Security Functional Requirements Rationale

The following discussion provides detailed evidence of coverage by SFRs for each security objective. The security objectives cover a wide range of concerns. Likewise, the security requirements must cover a wide range of functionality. User identification and authentication requirements allow the enforcement of user data protection. Additional security is added by including audit requirements. The security management requirements specify how these other functions will be managed by the TOE administrators. Lastly, the protection requirements ensure the TOE and the rest of the functions are protected. The selection of these requirements was fairly complex involving analyzing the threats to the TOE and considering how all the objectives would be met. The set of requirements has been analyzed and it has been determined that together the requirements forms a mutually supportive whole.

Table 20 - Relationship of Security Requirements to Objectives

	SFRs \ Objectives	O.ACCESS	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_PROTECTION	O.AUDIT_REVIEW	O.DISCRETIONARY_ACCESS	O.INSTALL	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.PROTECT	O.SOUND_DESIGN	O.SOUND_IMPLEMENTATION	O.TESTING	O.TRAINED_USERS	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	OE.SELF_PROTECTION	OE.TOE_PROTECTION	OE.TIME
TOE	FAU_GEN.1				✓																
	FAU_GEN.2				✓																
	FAU_SAR.1						✓														
	FAU_SAR.2					✓															
	FAU_SAR.3						✓														
	FAU_SEL.1				✓																
	FAU_STG.1					✓															

SFRs \ Objectives	O.ACCESS	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_PROTECTION	O.AUDIT_REVIEW	O.DISCRETIONARY_ACCESS	O.INSTALL	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.PROTECT	O.SOUND_DESIGN	O.SOUND_IMPLEMENTATION	O.TESTING	O.TRAINED_USERS	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	OE.SELF_PROTECTION	OE.TOE_PROTECTION	OE.TIME
	FAU_STG.NIAP-0414					✓														
	FDP_ACC.1	✓						✓				✓								
	FDP_ACF.1	✓						✓				✓								
	FIA_AFL.1																✓			
	FIA_ATD.1																	✓		
	FIA_SOS.1																✓			
	FIA_UAU.2																✓			
	FIA_UID.2																	✓		
	FMT_MOF.1				✓	✓					✓						✓			
	FMT_MSA.1							✓			✓									
	FMT_MSA.2										✓						✓			
	FMT_MSA.3							✓			✓									
	FMT_MTD.1(1)				✓	✓					✓									
	FMT_MTD.1(2)					✓					✓									
	FMT_MTD.1(3)										✓						✓			
	FMT_REV.1(1)	✓																		
	FMT_REV.1(2)											✓								
	FMT_SMF.1										✓									
	FMT_SMR.1			✓																
	FPT_ITD_EXP.1									✓		✓								
	FPT_RVM.1(1)				✓			✓		✓		✓								✓	
Env.	FPT_RVM.1(2)																			✓	
Env.	FPT_SEP.1																		✓	✓	
Env.	FPT_STM.1				✓		✓														✓
Assurance	ADO_DEL.1		✓						✓												
Assurance	ADO_IGS.1		✓						✓												
Assurance	ADV_FSP.1												✓	✓							
Assurance	ADV_HLD.2												✓	✓							
Assurance	ADV_RCR.1												✓	✓							

SFRs \ Objectives	O.ACCESS	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_PROTECTION	O.AUDIT_REVIEW	O.DISCRETIONARY_ACCESS	O.INSTALL	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.PROTECT	O.SOUND_DESIGN	O.SOUND_IMPLEMENTATION	O.TESTING	O.TRAINED_USERS	O.USER_AUTHENTICATION	O.USER_IDENTIFICATION	OE.SELF_PROTECTION	OE.TOE_PROTECTION	OE.TIME
AGD_ADM.1		✓																		
AGD_USR.1															✓					
ATE_COV.2													✓	✓						
ATE_FUN.1													✓	✓						
ATE_IND.2													✓	✓						
AVA_MSU.1		✓										✓	✓							
AVA_SOF.1												✓	✓							
AVA_VLA.1												✓	✓							

O.ACCESS

The TOE will ensure that users gain only authorized access to it and to its resources that it controls. The subjects and objects within the TOE are under the enforcement of a discretionary access control policy. (FDP_ACC.1)

The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules will be based on certain attributes of those subjects and objects. (FDP_ACF.1)

Security attributes associated with subjects and objects are the basis for access control. Revocation of these security attributes would modify the access control policy. The authorized administrator should have control over security attributes associated with users (such as user authentication data), being the only role that can revoke them. (FMT_REV.1(1))

O.ADMIN_ROLE

The TOE will provide authorized administrator roles to isolate administrative actions.

The TOE will establish, at least, an authorized administrator role. An authorized administrator may choose to specify more roles. The authorized administrator will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit information and security functions. (FMT_SMR.1)

O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users.

This objective is satisfied in part by the requirement that the TOE generate audit records according to the not specified level of auditing, as defined by the Common Criteria. (FAU_GEN.1) Each audit record written must be descriptive of the event that caused a record to be generated, and must

be associated with the unique identity of the user that caused the event. (FAU_GEN.2) The TOE enables the authorized administrator to pre-select events to include in the audit log. (FAU_SEL.1)

The TOE ensures that the authorized administrator role is the only role authorized to manipulate the behavior of the audit generation mechanism. (FMT_MOF.1) The TOE allows only authorized administrators to perform pre-selection of auditable events. (FMT_MTD.1(1)) The mechanisms providing self-protection are always invoked and not able to be bypassed. (FPT_RVM.1(1))

Reliable time stamps are assumed to be provided by the IT environment. (FPT_STM.1)

O. AUDIT_PROTECTION

The TOE will provide the capability to protect audit information.

Users must not be able to read the audit records, unless they have been granted explicit read-access to the audit log. (FAU_SAR.2) The TOE prevents unauthorized deletion or modification of audit records. (FAU_STG.1)

The TOE provides site-configurable options to prevent loss of audit data in the event the audit storage space is exhausted. (FAU_STG.NIAP-0414)

The TOE ensures that the authorized administrator role is the only role authorized to manipulate the behavior of the audit generation mechanism. (FMT_MOF.1)

Only the authorized administrator has the ability to query or clear audit records. (FMT_MTD.1(1), FMT_MTD.1(2))

O.AUDIT_REVIEW

The TOE will provide the capability to selectively view audit information, and alert the authorized administrator of identified potential security violations.

The authorized administrator will be the only user allowed access to the database audit information. This will prevent unauthorized users from modifying the audit information. In order for the authorized administrator to review the audit logs they must be in a suitable form for the authorized administrator to read, which means the authorized administrator should have the appropriate software needed to interpret the data. (FAU_SAR.1)

The authorized administrator can perform queries on the audit data based on date, time, type of event, event status (success or failure), or any other criteria chosen for FAU_SAR.3 in section 5 of the ST. This will allow the authorized administrator to search for specific events more efficiently. (FAU_SAR.3)

Reliable time stamps are assumed to be provided by the IT environment. The host operating system must provide accurate time stamps for its own use as well as for the TOE. These time stamps will be used for documenting auditing events. (FPT_STM.1)

O.DISCRETIONARY_ACCESS

The TOE will control access to resources based upon the identity of users or groups of users.

The subjects and objects within the TOE are under the enforcement of a discretionary access control policy. This policy can be configured by an authorized administrator to apply to a subset of the objects under control of the TOE. The administrator may configure some objects to be publicly accessible, and not under the control of the Discretionary policy. For example, the database system could have an interface to the Internet that lets users view certain public

information using their browser, while protected portions of the database are available only to certain users of the database. Consider a database for a financial institution. Public information might include current rates for savings accounts and for various types of loans. Private information might include information associated with each user's account, such as account balances and status of loan applications. (FDP_ACC.1)

The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules will be based on certain attributes of those subjects and objects. First the discretionary access control (DAC) mechanism will check to see if the user is authorized fro the requested action. If they are not authorized, the action will be denied. These rules are further defined in the FDP_ACF.1 section of heading 5. (FDP_ACF.1)

Only authorized administrators may manipulate the security attributes of database users. (FMT_MSA.1, FMT_MSA.3)

The Discretionary Access Control policy is not to be bypassed or optional. The discretionary aspect of the policy is that users who control access to objects can set that access to be restrictive or permissive to other users at their discretion. The policy is to be always enforced, never optional. (FPT_RVM.1(1))

O.INTERNAL_TOE_DOMAINS

The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.

The mechanisms providing self-protection are always invoked and not able to be bypassed. (FPT_RVM.1(1))

The TSF enforces separation between the security domains within its scope of control. (FPT_ITD_EXP.1)

O.MANAGE

The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE. (FMT_SMF.1)

Only the authorized administrator will be able to enable or disable functions of the audit log.

This will prevent a malicious user from turning off the audit log while he/she performs a malicious act, then turning it back on when he/she is done. (FMT_MOF.1)

Only authorized administrators may manipulate the security attributes of database users. (FMT_MSA.1, FMT_MSA.2, FMT_MSA.3)

Only authorized administrators are able to manage the inclusion/exclusion of specific events to be audited. (FMT_MTD.1(1))

Only authorized administrators are authorized to query or clear the audit log. (FMT_MTD.1(2))

Only authorized administrators are authorized to set or reset user authentication data. (FMT_MTD.1(3))

O.PROTECT

The TOE will provide mechanisms to protect user data and resources.

The Discretionary Access Control policy applies to all operations between subjects and objects controlled by the TOE. (FDP_ACC.1)

The subjects and objects under the discretionary access control policy will have certain rules that apply to all accesses between them. The rules will be based on certain attributes of those subjects and objects. First the DAC will check to see if the user is authorized fro the requested action. If they are not authorized, the action will be denied. These rules are further defined in the FDP_ACF.1 section of heading 5. (FDP_ACF.1)

Authorized administrators are allowed to modify the security attributes of subjects and objects as permitted by the Discretionary Access Control policy. (FMT_REV.1(2))

Users will not be able to bypass the security policy in order to enter the TOE. This means they must identify and authenticate themselves before accessing the TOE, and that the Discretionary Access Control policy is always enforced (FPT_RVM.1(1))

The TOE enforces security domains within its scope of control. (FPT_ITD_EXP.1)

O.USER_AUTHENTICATION

The TOE will verify the claimed identity of users.

To prevent brute force attacks on authentication data, the administrator must specify an upper bound on the number of unsuccessful authentications that will be allowed. Surpassing that threshold could indicate a brute force user authentication attack, and the TOE needs to take appropriate action. (FIA_AFL.1)

User authentication is meaningful only if there is an extremely low probability of success for random attempts to authenticate as an authorized user. The requirement that the secret authentication data be computationally difficult to guess randomly (FIA_SOS.1) Also, users authorized to access the TOE must identify themselves to the TOE. (FIA_UAU.2)

Only authorized administrators may access administrative resources. Specifically, only authorized administrators may manipulate the audit policy by enabling or disabling audit events. (FMT_MOF.1) The user authentication data is to be set only by an authenticated individual in an authorized role. (FMT_MTD.1(3))

The security attributes cannot be set to insecure values. Specifically, the security attributes for user authentication is the user authentication data. .(FMT_MSA.2)

O.USER_IDENTIFICATION

The TOE will uniquely identify users. Each database user will have a list of security attributes associated with them. (FIA_ATD.1) Users authorized to access the TOE must identify themselves to the TOE. (FIA_UID.2)

8.3 Security Functional Requirements Rationale for the IT Environment.

OE.SELF_PROTECTION

The IT environment and its assets will be protected from external interference, tampering or unauthorised disclosure.

Both the TOE and the IT environment will be protected from interference and will maintain its own secure domain. (FPT_SEP.1)

OE.TOE_PROTECTION

The IT environment will provide protection to the TOE and its assets from external interference or tampering.

Both the functions preformed in the TOE and the IT environment will execute correctly and will be protected from interference. (FPT_RVM.1(1), FPT_RVM.1(2), FPT_SEP.1).

OE.TIME

The TOE operating environment shall be able to generate reliable timestamps for the TOE's use. The TOE environment provides reliable time stamps for use by the TOE. (FPT_STM.1)

8.4 Security Assurance Requirements Rationale

O.ADMIN_GUIDANCE

The TOE will provide authorized administrators with the necessary information for secure management of the TOE.

When the TOE is delivered for installation, the authorized administrator must have confidence that it is the genuine, unaltered TOE procured from the TOE vendor. Procedures for delivery of the TOE will give the authorized administrator confidence in the TOE, its security mechanisms, and authorized administrator documentation that describes how to perform administrative duties securely. (ADO_DEL.1)

Installation and start-up procedures give the authorized administrator information necessary for initial generation of the TOE as intended by the developer. (ADO_IGS.1)

Since this is a software-only TOE, there are some requirements that may be allocated to the IT environment. The host operating system will be depended upon for security support and some security mechanisms. The administrator guidance must exist for the IT environment components that the TOE depends on. (AGD_ADM.1)

The Misuse assurance requirement forces the developer to provide guidance documentation, and for the evaluator to analyze the guidance for misleading, unreasonable or conflicting guidance that could hamper secure management of the TOE. (AVA_MSU.1)

O.INSTALL

The TOE will be delivered with the appropriate installation guidance to establish and maintain TOE security.

The developer must provide and adhere to procedures for secure transfer of the TOE from the development site to the customer’s site. This will ensure the TOE is delivered with all necessary security components and is not maliciously modified before it has been installed in the environment. (ADO_DEL.1)

The developer must provide the customer with all steps necessary for the secure installation and startup of the TOE. This must include the configuration of the TOE and its initial startup in a secure state. (ADO_IGS.1)

O.SOUND_DESIGN

The TOE will be designed using sound design principles and techniques. The TOE design, design principles and design techniques will be adequately and accurately documented.

The evaluators examine the developer's guidance on configuring the TOE securely. The purpose for the examination of the guidance is to ensure that it is not self-contradictory, confusing or unreasonable. (AVA_MSU.1)

The developer's analysis of the strength of the functions of the TSF shows that the functions meet or exceed SOF-basic. (AVA_SOF.1)

The developer conducts a vulnerability analysis that shows whether any identified vulnerabilities in the TOE provide an obvious way to circumvent the TSF. (AVA_VLA.1)

The developer provides an informal functional specification of the TOE that describes the user-visible interface and behavior of the TSF. (ADV_FSP.1)

The developer must document the informal high- level design of the TOE, describing the TOE in terms of major structural units and the security functions each unit provides. (ADV_HLD.2)

The correspondence between the various levels of abstraction of the TOE representation shows that there is correspondence between the high- level design and the functional specification. (ADV_RCR.1)

O.SOUND_IMPLEMENTATION

The implementation of the TOE will be an accurate instantiation of its design.

The developer provides an informal functional specification of the TOE that describes the user-visible interface and behavior of the TSF. (ADV_FSP.1)

The developer must document the informal high- level design of the TOE, describing the TOE in terms of major subsystems and the security functions each subsystem provides. This will assist the developer in finding any flaws in the design before it is implemented. (ADV_HLD.1)

The correspondences between the various levels of abstraction of the TOE representation show that there is correspondence between the high- level design and the functional specification. (ADV_RCR.1)

The coverage of testing is sufficient to show that the TSF is tested and shown to operate as specified in the functional specification. This will help to reduce implementation flaws. (ATE_COV.2)

The functional components of the TSF are tested, and shown to operate as specified. (ATE_FUN.1)

An independent party other than the developer conducts testing. This overcomes the risk of incorrect assessment of the test outcomes on the part of the developer. This will help to reduce implementation flaws. (ATE_IND.2)

The evaluators examine the developer's guidance on configuring the TOE securely. The purpose for the examination of the guidance is to ensure that it is not self-contradictory, confusing or unreasonable. (AVA_MSU.1)

The developer must perform a strength of TOE security function analysis on all mechanisms that hold a strength of function claim. The developer must show it meets or exceeds its strength of function level, which in this case is SOF-basic. (AVA_SOF.1)

The developer conducts a vulnerability analysis that shows whether any identified vulnerabilities in the TOE provide an obvious way to circumvent the TSF. This analysis will show the developer if there are any vulnerabilities that he/she will have to fix. (AVA_VLA.1)

O.TESTING

The TOE will undergo developer and independent testing that include test scenarios and results.

The developer will show evidence of the test coverage. This must correspond with the tests identified in the test documentation. (ATE_COV.2)

The developer must test the TSF and document the results. The documentation must include test plans, procedures, expected results, and actual results. The plans must identify the security functions tested. (ATE_FUN.1)

The developer must have the TOE tested by an independent party. The evaluator will test a subset of the TSF and confirm it operates as specified by the developer. The evaluator will then provide the appropriate evidence that it was tested. (ATE_IND.2)

O.TRAINED_USERS

The TOE will provide authorized users with the necessary guidance for secure use of the TOE, to include secure sharing of user data.

The developer of the TOE must provide appropriate user training in order to avoid misuse of the TOE resulting in a leak of protected data. The training will be consistent with all other documentation for the TOE. The training does not need to include instruction on administrative functions. (AGD_USR.1)

8.5 Dependency Rationale

This ST does satisfy all the requirement dependencies of the Common Criteria. Table 21 lists each requirement to which the TOE claims conformance with a dependency and indicates whether the dependent requirement was included. As the table indicates, all dependencies have been met.

Table 21 - Functional Requirements Dependencies

SFR Identifier	Dependencies	Dependency Met
FAU_GEN.1	FPT_STM.1	✓
FAU_GEN.2	FAU_GEN.1, FIA_UID.1	✓
FAU_SAR.1	FAU_GEN.1	✓
FAU_SAR.2	FAU_SAR.1	✓
FAU_SAR.3	FAU_SAR.1	✓
FAU_SEL.1	FAU_GEN.1, FMT.MTD.1(1)	✓
FAU_STG.1	FAU_GEN.1	✓
FAU_STG.NIAP-0414	FAU_STG.1, FMT_MTD.1	✓
FDP_ACC.1	FDP_ACF.1	✓
FDP_ACF.1	FDP_ACC.1, FMT_MSA.3	✓
FIA_AFL.1	FIA_UAU.1	✓
FIA_ATD.1	[none]	✓
FIA_SOS.1	[none]	✓

SFR Identifier	Dependencies	Dependency Met
FIA_UAU.2	FIA_UID.1	✓
FIA_UID.2	[none]	✓
FMT_MOF.1	FMT_SMF.1, FMT_SMR.1	✓
FMT_MSA.1	FDP_ACC.1 OR FDP_IFC.1, FMT_SMF.1, FMT_SMR.1	✓
FMT_MSA.2	ADV_SPM.1, FDP_ACC.1 or FDP_IFC.1, FMT_MSA.1, FMT_SMR.1	See below.
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1	✓
FMT_MTD.1(1)	FMT_SMF.1, FMT_SMR.1	✓
FMT_MTD.1(2)	FMT_SMF.1, FMT_SMR.1	✓
FMT_MTD.1(3)	FMT_SMF.1, FMT_SMR.1	✓
FMT_REV.1(1)	FMT_SMR.1	✓
FMT_REV.1(2)	FMT_SMR.1	✓
FMT_SMF.1	[none]	✓
FMT.SMR.1	FIA_UID.1	✓
FPT_ITD_EXP.1	N/A	✓
FPT_RVM.1(1)	[none]	✓
FPT_RVM.1(2)	[none]	✓
FPT_SEP.1	[none]	✓
FPT_STM.1	[none]	✓

ADV_SPM.1 is a dependency of FMT_MSA.2. ADV_SPM.1 has not been included in this evaluation because the TOE security policies are clearly defined in the ST. The one explicit security policy is the Discretionary Access Control policy. This policy is clearly defined by the requirements FDP_ACC.1 Subset access control, FDP_ACF.1 Security attribute based access control, FMT_MSA.1 Management of security attributes, and FMT_MSA.3 Static attribute initialization. These requirements clearly define subjects, objects, attributes and actions for the Discretionary Access Control policy. There are four additional implicit security policies defined in the ST. These are the audit policy, identification and authentication policy, security management policy, and TSF protection policy. These policies are also clearly defined by the requirements listed in this ST. Since all of the security policies explicitly or implicitly listed in the ST are defined by the ST, it was determined that the Security Target document is sufficient to meet this dependency.

8.6 TOE Summary Specification Rationale

8.6.1 TOE Summary Specification Rationale for the Security Functional Requirements

Each subsection in the TOE Security Functions Summary (Section TOE Security Functions Summary) describes a security function of the TOE. Each description is organized by set of requirements with rationale that indicates how these requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality. This section, in conjunction with the TOE Summary Specification section, provides evidence that the security functions are suitable to fulfill the TOE security requirements.

Table 22 identifies the relationship between security requirements and security functions, showing that all security requirements are addressed and all security functions are necessary (i.e., they correspond to at least one security requirement).

The only security mechanism that is realized by a probabilistic or permutational implementation is the password mechanism. For an analysis of the Strength of Function, refer to Strength of Function (SOF) Rationale section.

Table 22 - Mapping of Security Functional Requirements to TOE Security Functions

TOE Security Function	SFR	Rationale
Security Audit	FAU_GEN.1	Audit data generation is required for having a secure audit, because it creates the initial audit information. Without generating audit records no other audit functions can occur.
	FAU_GEN.2	User identity association is required for having a secure audit because it allows someone reviewing the audit log to know who triggered a particular event. This information would be invaluable in determining an attempt to gain unauthorized access.
	FAU_SAR.1	It will not matter if audit records are created if there is not a method to review the records.
	FAU_SAR.2	Some of the audit records may contain sensitive information and this information should only be viewed by people with a need to know.
	FAU_SAR.3	Some audit records may not need to be reviewed at all times. Also all administrators may not need to have the right to view all audit records.
	FAU_SEL.1	An authorized administrator may not need to audit everything the system can perform. Since audit space is finite this will lessen the chance of the audit logs becoming full, or at least delay the problem.
	FAU_STG.1	Since the audit logs maintain the history of all security related actions, it is important to make sure these records can only be erased by authorized administrators. This prevents someone from breaking into the system, and then hiding their attack by clearing the audit log.
	FAU_STG.NIAP-0414	The audit logs have a finite space for records. It is important to establish a policy of what actions to take if the space becomes full. In the case of Caché the oldest audit records are overwritten and the administrator is contacted.
User Data Protection	FDP_ACC.1	A DAC policy is important because it provides a mechanism to enforce access control on subject, objects and operations. Caché uses an Access Control List to enforce its DAC policy.
	FDP_ACF.1	It is important to enforce the DAC policy in a specific order so objects can only use privileges that are explicitly granted.

TOE Security Function	SFR	Rationale
Identification and Authentication	FIA_AFL.1	It is important to have an upper bound of unsuccessful attempts so that a user cannot attempt to gain unauthorized access into the TOE. Specifically this precaution helps to mitigate brute force and dictionary attacks.
	FIA_ATD.1	By assigning each user security attributes it will be possible to see if they are subject to an attack. It will also be possible to see what threat to the overall system if the attack is successful.
	FIA_SOS.1	It is important to have strong passwords so it will not be easy to perform a brute force attack against the system. Caché surpasses the minimum requirement for this by requiring a password of 8 characters.
	FIA_UAU.2	It is important that a user cannot perform any functions inside the TOE until they are authenticated. If the user cannot be properly identified or authenticated, then they cannot be trusted to not perform malicious acts.
	FIA_UID.2	It is important that a user cannot perform any functions inside the TOE until they are authenticated. If the user cannot be properly identified or authenticated, then they cannot be trusted to not perform malicious acts.
Security Management	FMT_MOF.1	It important to limit the ability to change the audit functions to only authorized administrators. Otherwise a user could turn off audit functionality before an attack occurred.
	FMT_MSA.1	It is important to limit the ability to change security attributes to authorized administrators only. If all users could change these attributes, then a user could grant themselves more privileges than they needed. This would compromise the security of the TOE.
	FMT_MSA.2	There will be no insecure security attributes. This will force all the security functions of the product to perform in an expected manner.
	FMT_MSA.3	It is important to initialize accounts with the minimum privileges possible so that users only have rights that are explicitly assigned to them. This will prevent users having rights they do not need.
	FMT_MTD.1(1)	It important to limit the ability to change the audit functions to only authorized administrators. Otherwise a user could turn off audit functionality before an attack occurred.
	FMT_MTD.1(2)	It important to limit the ability to change the audit data to only authorized administrators. Otherwise a user could turn erase audit data after an attack occurred.
	FMT_MTD.1(3)	It is important to restrict the ability to set and reset user authentication values to authorized administrators so that a user could not reset the account of someone with greater privileges. Being able to reset anyone's password would allow a user to masquerade as another user and view data they were not authorized to view.
	FMT_REV.1(1)	It is important to restrict the ability to revoke user attributes because if anyone could do this, an attacker could launch an attack merely by deleting everyone's accounts, which would prevent the authorized TOE users from gaining access.
	FMT_REV.1(2)	It is important to restrict the ability to revoke subject and object attributes because if anyone could do this an attacker could launch an attack merely by deleting the privileges associated with subjects and objects. Even if a users was able to authenticate, it would be very hard to have a productive use of Caché because they would not be able to harness the system's functionality

TOE Security Function	SFR	Rationale
	FMT_SMF.1	An authorized administrator must be able to effectively manage its users, data and access control policy.
	FMT_SMR.1	Roles provide a method for an authorized administrator to define a user's attributes. This cuts down on an administrator's time in creating new accounts and ensures that the correct set of rights are assigned to a user.
Protection of the TSF	FPT_ITD_EXP.1	It is important for each subject in the TOE scope of control to remain separate so there is no bleeding over of privileges or information from one subject to another.
	FPT_RVM.1(1)	It is important that it is not possible to bypass the security implemented by the TOE. If the security functionality protecting the TOE can be bypassed then the methods used to protect the TOE are irrelevant.

8.6.2 TOE Environment Summary Specification Rationale for the Security Functional Requirements

This section provides evidence that the TOE Environment security functions are suitable to fulfill the TOE Environment security requirements.

Table 23 identifies the relationship between security requirements and security functions, showing that all security requirements are addressed and all security functions are necessary (i.e., they correspond to at least one security requirement).

Table 23 - Mapping of Security Functional Requirements to TOE Security Functions

TOE Security Function	SFR	Rationale
Protection of the TSF	FPT_RVM.1(2)	It is important that it is not possible to bypass the security implemented by the TOE Environment. If the security functionality protecting the TOE Environment can be bypassed then the methods used to protect the TOE are irrelevant.
	FPT_SEP.1	It is important that environmental components maintain their own security domains in order to protect themselves from interference and tampering.
	FPT_STM.1	It is important that the environment provide reliable time stamps so that the administrator can be assured that audited events truly happened at the specified time.

8.6.3 TOE Summary Specification Rationale for the Security Assurance Requirements

EAL3 was chosen to provide a basic level of independently assured security in the absence of ready availability of the complete development record from the vendor. The chosen assurance level is consistent with the postulated threat environment.

While the TOE may monitor a hostile environment, it is expected to be in a non-hostile position and embedded in or protected by other products designed to address threats that correspond with the intended environment. The chosen assurance level was also selected for conformance with the client's needs.

8.6.3.1 Configuration Management

The Configuration Management documentation provides a description of tools used to control the configuration items and how they are used at the InterSystems. The documentation provides a complete configuration item list and a unique reference for each item. Additionally, the configuration management system is described including procedures that are used by developers to control and track changes that are made to the TOE. The documentation further details the TOE configuration items that are controlled by the configuration management system.

Corresponding CC Assurance Components:

- Configuration Items

8.6.3.2 Delivery and Operation

The Delivery and Operation documentation provides a description of the secure delivery procedures implemented by InterSystems to protect against TOE modification during product delivery. The Installation Documentation provided by InterSystems details the procedures for installing the TOE and placing the TOE in a secure state offering the same protection properties as the master copy of the TOE. The Installation Documentation provides guidance to the administrator on the TOE configuration parameters and how they affect the TSF.

Corresponding CC Assurance Components:

- Delivery Procedures
- Installation, Generation and Start-Up Procedures

8.6.3.3 Development

The InterSystems design documentation consists of several related design documents that address the components of the TOE at different levels of abstraction. The following design documents address the Development Assurance Requirements:

- The Functional Specification provides a description of the security functions provided by the TOE and a description of the external interfaces to the TSF. The Functional Specification covers the purpose and method of use and a list of effects, exceptions, and errors message for each external TSF interface.
- The High-Level Design provides a top level design specification that refines the TSF functional specification into the major constituent parts (subsystems) of the TSF. The high-level design identifies the basic structure of the TSF, the major elements, a listing of all interfaces, and the purpose and method of use for each interface.
- The Correspondence Analysis demonstrates the correspondence between each of the TSF representations provided. This mapping is performed to show the functions traced from the ST description to the High-Level Design.

Corresponding CC Assurance Components:

- Informal Functional Specification
- Descriptive High-Level Design
- Informal Representation Correspondence

8.6.3.4 Guidance Documentation

The InterSystems Guidance documentation provides administrator and user guidance on how to securely operate the TOE. The administrator Guidance provides descriptions of the security functions provided by the TOE. Additionally, it provides detailed accurate information on how to administer the TOE in a secure manner and how to effectively use the TSF privileges and protective functions. The User Guidance provided directs users on how to operate the TOE in a secure manner. Additionally, User Guidance explains the user-visible security functions and how they are to be used and explains the user's role in maintaining the TOE's Security. InterSystems provides

single versions of documents which address the administrator Guidance and User Guidance; there are not separate guidance documents specifically for non-administrator users of the TOE.

Corresponding CC Assurance Components:

- Administrator Guidance
- User Guidance

8.6.3.5 Tests

There are a number of components that make up the Test documentation. The Coverage Analysis demonstrates the testing performed against the functional specification. The Coverage Analysis demonstrates the extent to which the TOE security functions were tested as well as the level of detail to which the TOE was tested. InterSystems Test Plans and Test Procedures, which detail the overall efforts of the testing effort and break down the specific steps taken by a tester, are also provided.

Corresponding CC Assurance Components:

- Evidence of Coverage
- Functional Testing

8.6.3.6 Vulnerability and TOE Strength of Function Analyses

A Vulnerability Assessment is provided to demonstrate ways in which an entity could violate the TSP and provide a list of identified vulnerabilities. Additionally, the document provides evidence of how the TOE is resistant to obvious attacks. The Strength of TOE Security Function Analysis demonstrates the strength of the probabilistic or permutational mechanisms employed to provide security functions within the TOE and how they exceed the minimum SOF requirements.

Corresponding CC Assurance Components:

- Strength of TOE Security Function analysis
- Vulnerability Analysis

8.7 Explicitly Stated Requirements Rationale

The following explicitly stated SFRs were added to this Security Target to more accurately reflect the way that the TOE operates with regard to the Security Functions:

FAU_STG_EXP.NIAP-0414: The FAU_STG family does not support a way for the administrator to specify that the actions taken by the TSF to prevent audit data loss when the audit trail is full can be site selectable. In fact, the FAU STG.4 component explicitly states the actions to be taken by the TSF when the audit log is full. This wording implicitly prevents a site from selecting the action taken to prevent loss of audit data. This TOE provides the administrator with a selectable list of actions to be taken in the event that the audit log is full. At least one of the actions must be selected and there is a default action. For this reason, the explicitly stated security functional requirementFAU_STG_EXP.NIAP-0414 which is modeled on FAU_STG.4 has been included in the ST.

FPT_ITD_EXP.1: The FPT_SEP family does not provide a way for the TSF to enforce separation between the security domains of the subjects in the TSC, but not maintain a security domain for its own execution. The requirement FPT_SEP.1, from which this requirement is based, provides both these requirements. This TOE does provide enforcement of separation between the security domains of the subjects in the TSC. However, the environment provides a security domain for its execution. For this reason, the explicitly stated security functional requirement FPT_ITD_EXP.1 which is modeled on FPT_SEP.1 has been included in the ST.

8.8 Strength of Function

Strength of function rating of SOF-basic was claimed for this TOE to meet the EAL3 assurance requirements, this SOF is sufficient to resist the threats identified in Section 3.2. Section 4 provides evidence that demonstrates that TOE threats are countered by the TOE security objectives. Section 8 demonstrates that the security objectives for the TOE and the TOE environment are satisfied by the security requirements. The evaluated TOE is intended to operate in commercial and Department of Defense low robustness environments processing unclassified information.

The overall TOE SOF claim is *SOF-basic* because this SOF is sufficient to resist the threats identified in Section 3.2. Section 8.1 provides evidence that demonstrates that TOE threats are countered by the TOE security objectives. Section 8.2 demonstrates that the security objectives for the TOE and the TOE environment are satisfied by the security requirements.

The relevant security function(s) and security functional requirement(s) which have probabilistic or permutational functions are:

- FIA_UAU.2

The SOF for this requirement is explicitly stated in FIA_SOS.1.1. The requirement states that for each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in $5x10^{15}$. This exceeds the overall SOF claim. Thus, the overall SOF claim is stronger than the requirements of SOF-basic.

9 Acronyms

Table 24 - Acronyms

Acronym	Meaning
API	Application Programming Interface
CC	Common Criteria
CCIMB	Common Criteria Interpretations Management Board
DAC	Discretionary Access Control
DB	Database
DBMS	Database Management System
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
IP	Internet Protocol
IT	Information Technology
OS	Operating System
PP	Protection Profile
SFR	Security Functional Requirement
SFP	Security Function Policy
SOF	Strength of Function
SQL	Structured Query Language
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy