FUJ00152145

Document Title: Fraud Case Management System: Architecture Specification

Document Type: Specification

Abstract: This document specifies the software architecture proposed to satisfy the requirements for the Pathway Fraud Case Management System. It covers the software components and logical structures for the Pathway Fraud Case Management System.

Status: Draft – Version Complete

Distribution:

Mark Wilcox	(ICL Pathway)
James Henry	(ICL CFM)
Declan Sheehan	(ICL CFM)
Declan Brady	(ICL CFM)
Library	

Authors: Declan Sheehan
David Stapleton

Approval Authority: Mark Wilcox, Declan Sheehan

Quality Authority: David Groom

Comments to: Declan Sheehan

Comments by: 31/03/98

FUJ00152145

0 DOCUMENT CONTROL

0.1 DOCUMENT HISTORY

Version	Date	Reason
0.1	5/11/96	Initial Draft
0.2	10/7/97	Added sections on backup, recovery, functionality
0.3	15/7/97	Added sizing information.
0.4	17/7/97	Final draft
0.5	21/08/97	Included comments from workshop
1.0	10/03/98	Internal CFM Review
1.1	01/03/99	Document baselined for Release 2. CP1157 applied

0.2 ASSOCIATED DOCUMENTS

	Reference	Vers	Date	Title	Source
[1]	DW/REQ/0003	0.3	27/10/96	Pathway Corporate Services: MIS Non-functional Requirements	CFM
[2]	RS/SPE/0004	0.3	23/9/96	Fraud Risk Management Service Design	CFM
[3]	DW/REQ/0005	0.2	22/9/96	Data Warehouse: Fraud Risk Management Requirements	CFM
[4]	DW/REQ/0006	0.1	27/9/96	Data Warehouse: BPS, SLA and MIS Requirements	CFM
[5]	DW/REQ/0017	0.1	27/9/96	FRMS System Requirements Specification	CFM

FUJ00152145

0.3 ABBREVIATIONS

Please refer to [6].

0.4 CHANGES IN THIS VERSION

None.

FUJ00152145

0.5 TABLE OF CONTENT

1 INTRODUCTION..... 6

1.1 SCOPE..... 6
1.2 ASSUMPTIONS..... 6
1.2.1 ACCURACY..... 6
1.3 DOCUMENT STRUCTURE..... 6
1.4 NON-FUNCTIONAL REQUIREMENTS..... 7
1.5 FRAUD RISK MANAGEMENT REQUIREMENTS..... 7

2 ARCHITECTURAL OVERVIEW..... 8

2.1 FRAUD CASE MANAGEMENT SYSTEM PROCESS..... 9
2.2 SOURCE SYSTEMS..... 10
2.2.1 REFERENCE DATA..... 10
2.2.2 INTERFACES TO SOURCE SYSTEMS..... 10
2.3 USER SYSTEMS..... 11
2.3.1 INTERFACES TO USER SYSTEMS..... 11

3 NON-FUNCTIONAL REQUIREMENTS..... 12

3.1 SECURITY..... 12
3.2 AUDIT..... 12
3.3 ACCURACY..... 12
3.4 DATA STORAGE..... 13
3.5 WORKLOAD..... 13
3.6 ARCHIVAL..... 13
3.7 PERFORMANCE..... 13
3.8 TIMING..... 13
3.9 EXTERNAL INTERFACE..... 13
3.10 LAYERED SOFTWARE..... 13
3.11 CONTINGENCY..... 14
3.11.1 BACKUP AND RECOVERY..... 14
3.11.2 DISASTER RECOVERY..... 14
3.12 INSTALLATION..... 14

4 ENTERPRISE BUSINESS MODEL..... 15

4.1 ENTITY RELATIONSHIP DIAGRAM..... 15
4.2 ENTITY TYPES..... 16
4.3 CORE ENTITY DESCRIPTIONS..... 16
4.3.1 CASE..... 16
4.3.2 TRANSACTION..... 16
4.3.3 ENCASHMENT..... 17
4.3.4 AUDIT..... 17
4.3.5 CARDHOLDER..... 18
4.3.6 CARD..... 18
4.3.7 CARD EVENT..... 18
4.4 LIIMTED SCOPE ENTITIES..... 19
4.4.1 STORED QUERY..... 19
4.4.2 ANOMALIES..... 19
4.5 REFERENCE ENTITIES..... 20
4.5.1 POST OFFICE..... 20
4.5.2 REPORT TYPES..... 20
4.5.3 CODE TABLES..... 20
4.6 ADDING AND UPDATING DATA IN THE FRAUD CASE DATABASE..... 21
4.6.1 USER-UPDATABLE TABLES..... 21
4.6.2 SYSTEM-UPDATABLE TABLES..... 21

5 PROCESS ARCHITECTURE..... 22

FUJ00152145

5.1 SOFTWARE FUNCTIONAL OVERVIEW..... 22
5.2 SOFTWARE PROCESSING..... 22
5.3 PROCESSING SCHEDULE..... 24
5.3.1 BATCHED QUERIES..... *24*
5.3.2 BACKUP..... *24*
5.3.3 BUSINESS OBJECTS DOCUMENT AGENT..... *24*

6 *AD HOC* QUERIES AND STANDARD REPORTS..... 25

6.1 *AD HOC* QUERIES..... 25
6.2 STANDARD REPORTS..... 25

7 TABLE DESIGN..... 26

7.1 NAMING CONVENTIONS..... 26
7.2 INDEXING STRATEGY..... 26
7.2.1 CORE TABLES..... *26*
7.2.2 REFERENCE TABLES..... *26*
7.3 TABLESPACES..... 26
7.3.1 SIZING STRATEGY..... *27*
7.3.2 MAIN DATA..... *28*
7.4 DISK SIZING..... 29
7.4.1 SUMMARY..... *29*

8 ARCHIVAL..... 30

9 AUDIT..... 31

10 RECOVERY STRATEGY..... 32

10.1 PRIORITY SCHEDULE FOR DATA RECOVERY..... 32
10.2 PRIORITY SCHEDULE FOR PROCESS RECOVERY..... 33

11 TEST STRATEGY..... 34

11.1 TESTING PHILOSOPHY..... 35
11.1.1 TEST DATA..... *35*
11.1.2 TEST STAGES..... *36*

1 INTRODUCTION

This document specifies the software architecture proposed to satisfy the requirements for the Fraud Case Management System stated in [1], [2], [5].

1.1 SCOPE

This document covers the software components and logical structures for the Pathway Fraud Case Management System. It excludes hardware, contingency (disaster recovery) issues, networking – these issues are to be covered in the Infrastructure Specification and Design documents (TBA).

1.2 ASSUMPTIONS

This document makes the same base assumptions as given in [5], with

FUJ00152145

the following *additions*:

1.2.1 ACCURACY

As stated in [1], it is a requirement that the accuracy and integrity of data from source systems is entirely the responsibility of the said source system. Thus it is assumed that data will be accepted from source systems at "face value" i.e. the Fraud Case management System will not attempt to validate source data, or verify cross references, in any way. The Fraud Case Management System's processing software will thus make the following assumptions:

1. Data that is accepted form the data warehouse will not need to be validated.
1. Data that is accepted from CMS/PAS will not need to be validated.

1.3 DOCUMENT STRUCTURE

The remainder of this document is structured as follows:

- **Architectural Overview**
  This section gives a high level overview of how the data warehouse is constructed and operated.
- **Non-functional Requirements**
  This section addresses how each of the non-functional requirements identified in [1] which are pertinent to the Fraud Risk Management System will be addressed.
- **Enterprise Business Model**
  This section presents the data warehouse understanding of the Pathway Business Model.
- **Process Architecture**
  This section gives a high level breakdown of the processes required to operated the data warehouse.
- ***Ad Hoc* Queries and Standard Reports**
  This section lists the standard reports that will be implemented for the data warehouse and describes how *ad hoc* query capability will be supplied.
- **Table Design**
  This section details the base version of the data warehouse physical data model.
- **Archival**
  This section describes how data will be archived from the system, and how the archive will be restored to allow end-user queries.
- **Audit**
  This section details how the auditability requirement for the data

FUJ00152145

warehouse will be addressed.

- **Recovery Strategy**
  This section describes the backup and recovery strategy for the data warehouse.
- **Test Strategy**
  This section describes the strategy for testing the various components of the data warehouse, and the data warehouse as a unit.
- **Appendices**

1.4 NON-FUNCTIONAL REQUIREMENTS

Those non-functional requirements which affect the specification of the Data Warehouse are addressed in Section 3 “Non-Functional Requirements”.

1.5 FRAUD RISK MANAGEMENT REQUIREMENTS

The Fraud Risk Management requirements for ad-hoc query and standard reports [3] are addressed in Section 6 “*Ad hoc* Queries and Standard Reports”.

2 ARCHITECTURAL OVERVIEW

Figure 1 depicts the Fraud Case Management System in the context of the Data Warehouse External Architecture. The inputs or sources of data for the warehouse are shown on the left of the diagram, while the outputs or users of the data are shown on the right. The scope of this document is *to the right of the* left-most interface (depicted by the vertical dotted lines).

FUJ00152145

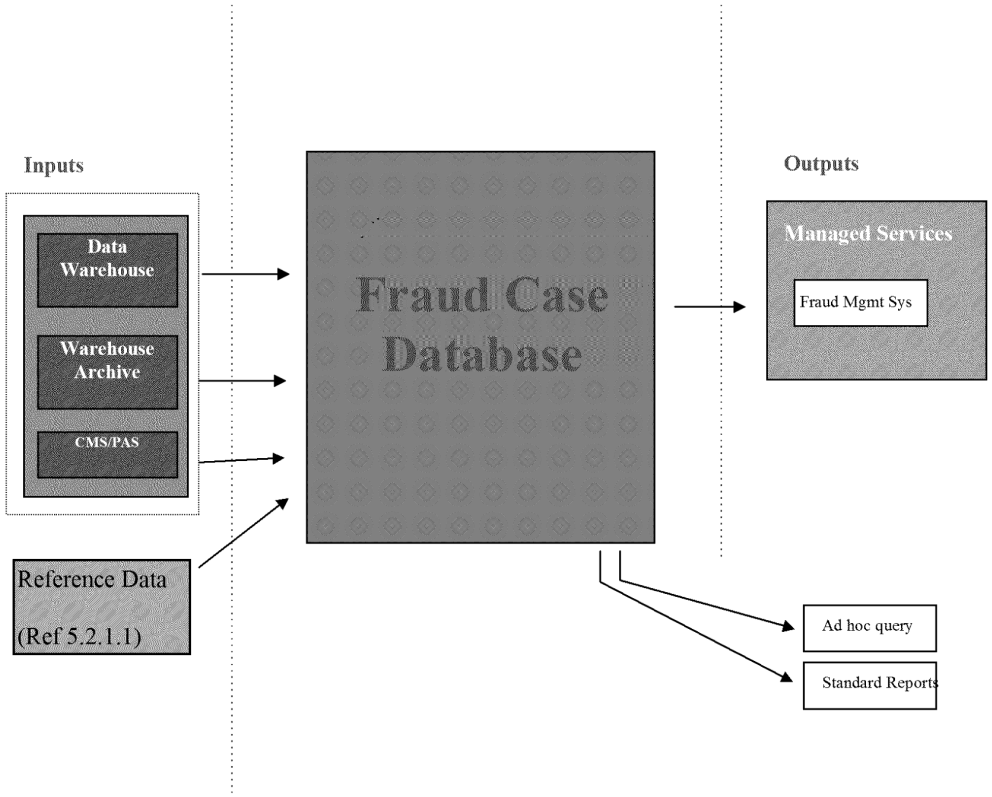

Figure 1 – Fraud Case Management System External Architecture

2.1 FRAUD CASE MANAGEMENT SYSTEM PROCESS

The Fraud Case Database is a case management system, and as such the principal function is the management of cases. A case may consist of multiple encashments. The encashments in the case are investigated and flagged as fraudulent.

FUJ00152145

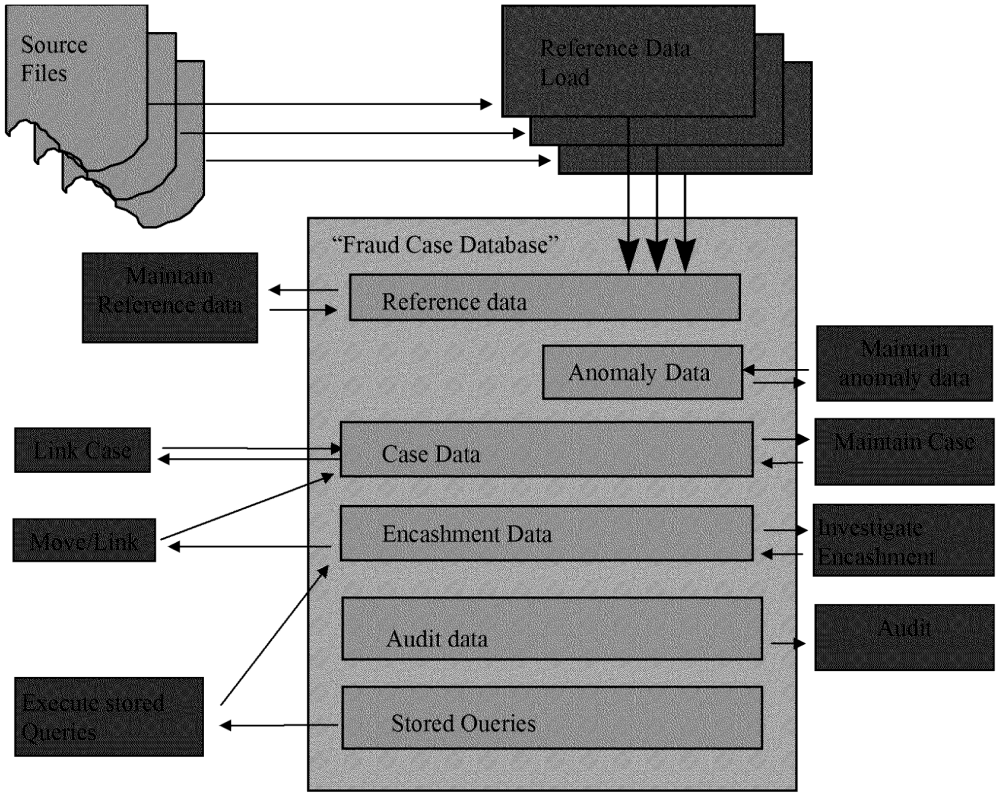

Figure 3 – Fraud Case Database Conceptual Process Architecture

Two types of reference data is required:- POCL and local. The POCL reference data is loaded at the start of each day from flat files. The local reference data (e.g. local codes/decodes) is maintained through a screens based interface. Anomaly information is maintained using a similar screens-based interface whereby certain objects (post offices and card-holders) can be defined as anomalous on particular reports.

Most encashment data is sourced from the data warehouse through a database link. However, for data that is more than 20 weeks old, it is necessary to source data from an archive. This means that the archive needs to be restored and when restored, the query should be run against the archive. Queries can be defined and stored, and then run against the restored archive at a later date.

Audit data relating to encashment and case enquiry and update is stored and can be reported against.

FUJ00152145

2.2 SOURCE SYSTEMS

Source systems are depicted in **Figure 1** (as "inputs"). These are as follows:

- Data Warehouse
- Data Warehouse archive
- CMS/PAS (operational system and help desk)
- Reference Data

2.2.1 REFERENCE DATA

"Reference data" can be regarded as any data used to encode something which is used by more than one system within Pathway. Such data needs to be consistent throughout all systems, with a synchronised release schedule. The reference data system will manage the release of new reference data to all systems within Pathway.

However in the first release of the Fraud Case Database, the reference data will be sourced from the data warehouse (this is primarily due to the uncertainty over whether a reference data system actually will actually exist in Pathway by the time release 1c goes live).

2.2.2 INTERFACES TO SOURCE SYSTEMS

Interfaces to all source systems (including "Reference Data") will be defined in separate "interface definition" documents (one [at least] for each service operator, where each supplier may own more than one source system). These documents will describe the mechanism by which data is extracted from the source systems, the availability of data in the source systems, the format of the data and any special configuration that is required to extract the data from the source system.

2.3 USER SYSTEMS

User systems are depicted in **Figure 1** (as "outputs"). User systems fall into two categories:

1) Managed services being provided by CFM:

- Standard screen access to the Fraud Case Database

FUJ00152145

2) Ad-hoc query and reporting access (via the "Business Objects" end-user query tool).

2.3.1 INTERFACES TO USER SYSTEMS

Interfaces to all user systems (except "Ad-hoc query" and "Standard reports") will be defined in the same manner as those for source systems (see Section 2.2.2 "Interfaces to Source Systems").

3 NON-FUNCTIONAL REQUIREMENTS

This section addresses how non-functional requirements affecting the design and development of the Data Warehouse will be satisfied (cf [1]).

3.1 SECURITY

There are two aspects to security: Security of the Fraud Case Database server itself, and security of the data residing on the server. The former will be addressed in the Infrastructure Specification (TBA). The latter will be addressed using standard Oracle data access security mechanisms.

Users of the data will have access to one or more Oracle "roles", as determined and controlled by the DBA. These Oracle roles in turn confer access rights to the underlying data. The initial set of Oracle end user roles will include:

1. Fraud Risk Management Reporting role
   This will confer read access to all data required by the Business Objects user

1. Fraud Risk Management Case Management roles
   This will confer read/write access to all data owned by the user. This role denies read/write access to the audit table.

1. Fraud Risk Management Supervisor role
   This role will confer read/write access to all case data in the system, and read access to the audit data.

1. Fraud Risk Management Super role
   This role will confer all access to the user, allowing archive, backup, recovery procedures to take place.

FUJ00152145

3.2 AUDIT

This is addressed in Section 9 “Audit”.

3.3 ACCURACY

This is addressed in Section 1.2.1 “Accuracy”.

3.4 DATA STORAGE

Data will be retained in the Fraud Case Database indefinitely until an archive policy is defined. When such an archive policy has been developed, data which has aged beyond its retention period, after being archived (see Section 8 “Archival”), will be deleted from the database.

3.5 WORKLOAD

The fraud case database will be designed (where possible, subject to hardware constraints – see [x]) to support the following:

- Support *ad hoc* query access by the users identified in [x]

3.6 ARCHIVAL

This is addressed in Section 8 “Archival”.

3.7 PERFORMANCE

There are no stated performance requirements for the Fraud Case Database. However, the Fraud Case database will be designed in such a way as to facilitate the production of “standard reports” (see [2][3]) and, where possible, ad hoc queries against data stored in the fraud case database within a reasonable time frame.

3.8 TIMING

Batched queries against the archive will be driven by a simple scheduler (e.g. cron).

3.9 EXTERNAL INTERFACE

All external interfaces (except for ad hoc queries and standard reports) will be addressed in the appropriate Interface Specification (TBA).

FUJ00152145

3.10 LAYERED SOFTWARE

The Fraud Case database will be constructed using Oracle RDBMS version 7.3.2 (or greater) and the associated tools and utilities, SQL/NET version 2 (or greater).

3.11 CONTINGENCY

3.11.1 BACKUP AND RECOVERY

This is addressed in Section 10 “Recovery Strategy”.

3.11.2 DISASTER RECOVERY

This falls outside the scope of this document. It is to be addressed by Pathway.

3.12 INSTALLATION

Supplied software will include installation scripts and supporting documentation.

4 ENTERPRISE BUSINESS MODEL

This section presents the *Enterprise Business model* of Fraud Case database which describes the business area of interest along with attribute lists for each entity. The model is a third normal form entity relationship diagram which outlines the scope and content of the data warehouse (see Figure 4).

4.1 ENTITY RELATIONSHIP DIAGRAM

FUJ00152145

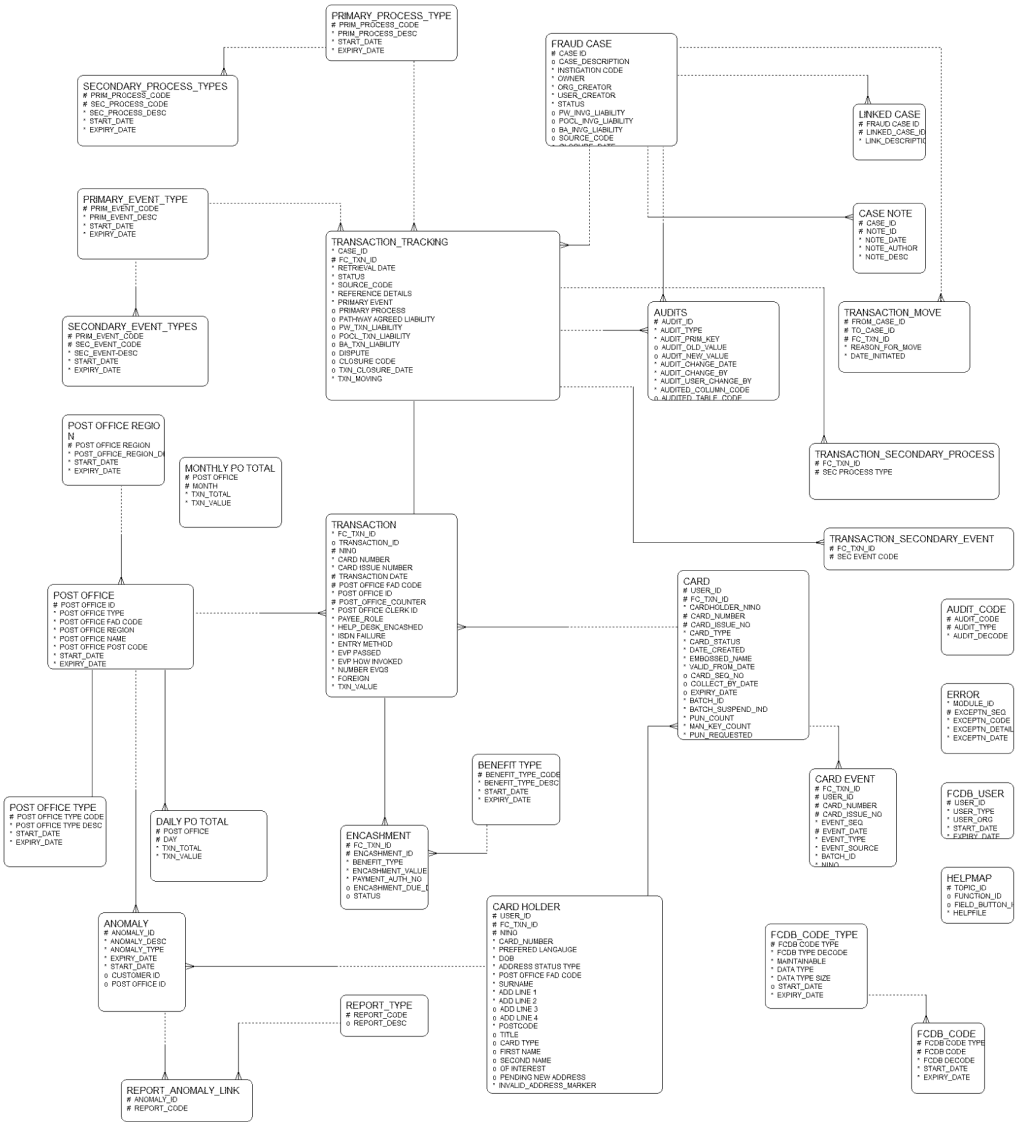

Figure 4 – Enterprise Business Model

4.2 ENTITY TYPES

Figure 4 shows four types of entity; core entities (i.e. those entities considered to be part of the fraud case database), reference entities, and finally limited scope entities (where the fraud case database is not interested in the entities themselves, but in statistics about those entities).

The core entities can be considered as those which shall hold the data which is critical to the fraud case database. The reference entities are essentially lookup tables that are required to describe the core data.

FUJ00152145

4.3 CORE ENTITY DESCRIPTIONS

4.3.1 CASE

A case is a collection of transactions[1] that are being investigated. The transactions within the case will have some common feature such as the same card holder, post office. A case is primarily used as a container of transactions that are being investigated for the same reason.

This data is entered by the operator during case creation and maintenance. A case has the following attributes:

- Case owner
- Instigation code
- Unique Id
- Case creator
- Case status
- Investigation Liability

4.3.2 TRANSACTION

The transaction entity describes both the transaction that took place at the counter of the post office, and the current state of the investigation of that transaction by the Fraud Investigator. In the data model, this conceptual entity has been broken into two logical entities:

- Specific Transaction details. These details include the transaction date, the value, the payee role, the card number etc. Full details are available in the logical data model.
- Transaction tracking details. These details include the status of the investigation of the encashment, the closure codes that can be assigned and any other codes that are used to track the investigation of the encashment.

The first represents the physical transaction that took place at the counter; the second (a more logical entity) represents how this transaction is being tracked during the course of the investigation.

Relationship to case

A case may contain one or many transactions, and a transaction must belong to one and only on case.

[1] with associated encashments

FUJ00152145

4.3.3 ENCASHMENT

The encashment entity represents the details of the benefit that was encashed at a particular post office on behalf of a particular beneficiary. The details regarding an encashment are sourced from the data warehouse. These encashment details are supplemented by a set of codes that facilitate the tracking and investigation of the encashment. An encashment has the following attributes:

- Specific encashment details. These details include the encashment date, the value, the benefit type, the benefit due date etc. Full details are available in the logical data model.
- Encashment tracking status.

Relationship to transaction

A transaction must contain one or many encashments, and an encashment must be part of one and only one transaction.

4.3.4 AUDIT

The audit entity is designed to contain details of all updates and enquiry accesses on the case and transtaion entities. The audit entity contains the following attributes:

- Author of change
- Date of change
- Change of ownership
- Status
- old and new values before and after change

4.3.5 CARDHOLDER

A “cardholder” is somebody that holds a benefit card. The card may or may not entitle the holder of the card to draw benefit. The cardholder has the following principle attributes:

- NINO
- Card Number
- Name
- Address details

FUJ00152145

- Card type

4.3.6 CARD

A card is required by a beneficiary in order to draw benefit. The card detail information is sourced from CMS. The principal attributes are as follows:

- Date created
- Embossed name
- Card Number
- Card issue Number
- Valid from date
- Expiry date
- Valid from date
- Collect by date

4.3.7 CARD EVENT

A card event details any event that has occurred to a card. The principal attributes associated with a card event are:

- Event date
- Card number
- Card issue number
- Event type
- Event Source
- Status

4.4 LIIMTED SCOPE ENTITIES

There will be a number of limited scope entities that will be stand alone.

4.4.1 STORED QUERY

The stored query has the following attributes:

- **Query Number ***
- **Case identifier**.

FUJ00152145

- **Query SQL**
- **Month and Year**
- **Query Status**
- **Date defined**

4.4.2 ANOMALIES

The anomalies entity defines the nature of the anomaly The anomaly has the following attributes:

- **Anomaly Id ***
- **Anomaly Description**.

This entity is linked to the report entity to define which anomalies appear on which report:

- **Anomaly Id ***
- **Report Code ***.

4.5 REFERENCE ENTITIES

4.5.1 POST OFFICE

A post office has the following attributes:

- **Post Office Number ***
- **Post Office Region**.
- **Post Office Type**
- **Post office Fad code**
- **Post Office post code**

It is assumed that the Post Office Number will be unique within the Pathway System.

4.5.2 REPORT TYPES

A report has the following attributes:

- **Report Id ***
- **Report Description**

FUJ00152145

4.5.3 CODE TABLES

4.5.3.1 FCDB_CODES

All codes will be driven by a generic code/decodes table. The table is driven by a code and a code type. This entity has the following attributes:

- **Code Type ***
- **Code ***
- **Decode**

4.5.3.2 FCDB_CODE_TYPES

This entity describes the type of the code that is stored in the generic codes table. It defines the data type, its size and whether it is maintainable. This entity has the following attributes:

- **Code Type ***
- **Code Type Description**
- **Is Maintainable**
- **Data type**
- **Data type size**

4.6 ADDING AND UPDATING DATA IN THE FRAUD CASE DATABASE

This section discusses at high level how new source data will be added to the warehouse and how changes to data will be handled.

4.6.1 USER-UPDATABLE TABLES

The following tables can be updated by the users of the system:

- **Cases**
- **Transactions**
- **Linked Cases**
- **Report Anomalies**
- **Case Notes**

FUJ00152145

4.6.2 SYSTEM-UPDATABLE TABLES

The following tables are updated by system:

- audit

5 PROCESS ARCHITECTURE

5.1 SOFTWARE FUNCTIONAL OVERVIEW

BH to fill in

5.2 SOFTWARE PROCESSING

5.2.1.1 REFERENCE DATA LOAD

A certain amount of POCL reference data is required in the Fraud Case database. Normally, reference data is available from a centralised reference data system. This centralised reference data system is being developed by Pathway. In the absence of this centralised reference data system, the POCL reference data will be sourced from the Data warehouse.

Reference data is loaded into the Fraud Case database at the start of each day. A process on the Fraud Case Database will be invoked to select the required reference data from the Data Warehouse into a flat file. The flat file will then by transferred electronically to the Fraud Case database platform, and the Reference data will then be uploaded into the set of Reference data tables on the Fraud Case database.

5.2.1.2 CASE/TRANSACTION MAINTENANCE

Cases are maintained through a screens-based interface. The functions that can be performed on a case by a normal user are:

- Create/Open Case
- Populate case
- Link case

FUJ00152145

- **Add/Show Case Notes**
- **Open/Update Transaction**
- **Define stored query**

The additional functions that can be performed by a supervisor are:

- **Re-define Anomaly**
- **Execute stored query**
- **Schedule report generation**
- **Define/maintain codes**

5.2.1.3 CODES MAINTENANCE

Codes are maintained through a screens-based interface. The following functions can be performed by a supervisor:

- **Update decode**

5.2.1.4 BATCHED QUERIES TO RESTORED ARCHIVE

There may be a number of queries that are unable to execute against the data warehouse because the data is older than 20 weeks. In this case, the operator may store the query so that it runs against the database when the archive has been restored.

The query details can be stored in a query table. The query details must include the case id to which the result set will be attached, and the month against which the query should be run. During the definition of the stored query, the query will be constrained to a single month (since only a single monthly archive can be restored at any one time).

At the end of a logical working day, a process will be invoked to run any queries that can be run against the restored archive. The process will check the date of the restored archive (if an archive has been restored) and will select the set of queries that can be run against this date. The queries will be run sequentially, and the result set for each query will be attached to the case that has been referenced in the query table. When a query has been run, the status of the query is updated in the query table.

FUJ00152145

5.2.1.5 ARCHIVING STABLE DATA

Refer also to Section 8 "**Archival**".

5.2.1.6 ARCHIVING OTHER DATA

Data that can be archived includes the stored queries that have already been run successfully against the restored archive. The retention period for queries that have been run will be 3 months.

5.3 PROCESSING SCHEDULE

5.3.1 BATCHED QUERIES

Stored queries will be run against the database at 20.00 hours each night.

5.3.2 BACKUP

Backup is addressed in Section 10 "**Recovery Strategy**" (but will be described in more detail in the Infrastructure Specification (TBA)).

5.3.3 BUSINESS OBJECTS DOCUMENT AGENT

Schedule is TBD

6 *AD HOC* QUERIES AND STANDARD REPORTS

6.1 AD HOC QUERIES

Ad hoc query functionality against data in the Fraud Case Database will be provided using Business Objects version 4.1.1. An appropriate Business Objects "universe" will be designed in conjunction with end-users and documented separately.

6.2 STANDARD REPORTS

Standard reports (as identified in [x]) will be implemented with

FUJ00152145

Business Objects version 4.1.1, using the Business Objects "universe" identified in Section 6.1 "*Ad hoc* Queries". Thus, although data for standard reports may be pre-prepared (in the form of pre-computed aggregates etc., see section 5 "Process Architecture"), the reports themselves will not be printed in batch mode but will be printed directly at the end-user's request.

Standard Reports will be (see [x]):

- Per month: xxxxxx
- Per quarter: vvvvvv
- Per quarter: ffffff

7 TABLE DESIGN

7.1 NAMING CONVENTIONS

The following conventions have been adopted:

- Table Names. Tables have been named after the corresponding entities.
- Entities. Entities have been named to reflect the purpose of the entity.
- Indexes. Indexes have been named after the table name with a suffix of _IXn where n is a numeric ranging from 1 to N, where N is the number of indexes.
- Tablespaces. Data Tablespaces have been named to reflect the purpose of the tablespace. Index tablespaces are name after the data tablespace with a suffix of _IDX.

7.2 INDEXING STRATEGY

7.2.1 CORE TABLES

Indexes have been defined on all primary keys and on most foreign keys in the core tables.

7.2.2 REFERENCE TABLES

Indexes have been defined on the primary keys of all reference data tables.

FUJ00152145

7.3 TABLESPACES

Tablespaces are required for the following object types:

- Reference data tables
- Reference data indexes
- Audit tables
- Audit indexes
- Fact tables
- Fact table indexes

7.3.1 SIZING STRATEGY

The large fact tablespaces will be designed to be able to hold the maximum expected data sizes, in order to avoid the problems associated with tablespaces filling up at run-time. The maximum data sizes are based on a retention period of 7 years. The sizes have been calculated as follows:

- the CASE and ENCASHMENT tablespaces are sized based on the assumption that 150,000 cases and 300,000 encashments are expected per year. The retention period is assumed to be a maximum of 10 years;
- the REFERENCE tablespace is sized on the assumption that the only significant contribution to the tablespace is the POST OFFICE reference data (20,000 post offices);
- the METADATA tablespace is based on sizing required for the Business Objects data dictionary;
- the AUDIT tablespace is based on the fact that each every encashment will be updated 6 or 7 times, every encashment will be enquired on 4 or 5 times, and a case will be enquired on approximately 15 - 20 times per encashment in the case.
- Data is assumed to be retained for 7 years.
- It is assumed that the average number of enquiries per case is 20 and that a case will be updated 10 times during its lifetime.
- It is assumed that an encashment will be enquired upon approximately 20 times on average and will be updated 10 times (on average).
- Average number of reports is 20.
- Number of primary event codes is assumed to be 20.
- Number of secondary event codes per primary code is assumed to be 10.
- It is assumed that there are an average of 2 card per cardholder.

FUJ00152145

- It is assumed that approximately 25,000 cardholders will be investigated over 7 years.
- It is assumed that the average number of card events per car is 5.
- It is assumed that the number of anomalies reported will be approximately 1000 per year.

7.3.2 MAIN DATA

The following tablespaces are proposed for the main data, i.e. data not restored from archive:

Tablespace	Description	Size (bytes)
REFERENCE	All reference data tables	4,358,144
REFERENCE_IDX	All reference data indexes	48,611,328
METADATA	All metadata tables including Bus	25,000,000
AUDIT	All tables containing audit data	2,804,875,264
AUDIT_IDX	Indexes on audit data	677,298,176
CASE	All case data	1,351,680,000
CASE_IDX	Indexes on case data	28,213,248
CASE_NOTES	Case notes	24,576,000,000
CASE_NOTES_IDX	Index on case notes	1,118,339,072
ENCASHMENT	All encashment data	412,917,760
ENCASHMENT_IDX	Indexes on encashment data	327,409,664
TRANSACTION_TRACKING	All transaction tracking data	621,969,408
TRANSACTION_TRACKING_IDX	Index on tracking data	707,796,992
SYSTEM	Oracle data dictionary	100,000,000
CARD_DATA	Table of Card data	255,492,096
CARD_DATA_IDX	Index on card data	130,170,880
TEMP		50,000,000
ROLLBACK		100,000,000
Total		33,340,132,032

Table 1 – Proposed Main Tablespaces

FUJ00152145

The details for the above spreadsheet are shown in appendix A.

7.4 DISK SIZING

7.4.1 SUMMARY

Assuming that 9 GB disks are used, then 5 to 6 disks are required to support the requirements (34 GB) of the Fraud Case database over the next seven years. The growth in the database is approximately 5 GB. This growth could be accommodated by adding a disk every year to the platform. ***The performance implications of this policy need to be considered and are beyond the scope of this document***.

The table below summarises the overall disk requirement.

Storage	Storage 34 GB	Raid level 5 41 GB
Number of 9GB disks	4	5
Number of 4.5 GB disks	9	10

The table below summarises the growth in the number of disks per year (if disks are added on an as-needed-basis). Note that if disks are added when required, there will be an I/O bottleneck in the early years when all data and indexes are stored on the same disk. Note also that when disks are added, a re-mapping will be required to optimise the I/O.

Storage	Storage 5 GB	Raid level 5 6 GB
Number of 9GB disks	1	1
Number of 4.5 GB disks	2	2

The recommended solution is to design a production sized database from the outset for all entities apart from the case notes entity. The case notes tablespace should be enlarged/grown over time.

FUJ00152145

8 ARCHIVAL

An archive policy has not been defined for the first release of the Fraud Case Database.

9 AUDIT

The requirements for Audit within the Fraud Case Database (cf. [1]) may be summarised as:

a) All updates to cases are to be audited

a) All updates to encashments are to be audited

a) All attempted connections to the fraud case database are to be audited.

a) Any change of relationship between a case and an encashment will be audited.

Requirements c) and d) will be satisfied as follows:

1.1) all updates to the fraud case and encashment table will be audited by means of a trigger on update/insert to the relevant table. This trigger will invoke a stored procedure to capture key audit data about the update/insert.

1.1) An audit report will be available showing the audit details for any particular case.

Requirement e) will be satisfied as follows:

2.1) Any actual/attempted connections to the database will be audited by using the Oracle audit facility to audit sessions and connections.

2.1) An audit report will be available showing the audit details.

Requirement f) will be satisfied as follows:

3.1) All changes of relationship between a case and an encashment will be audited by defining triggers on the relationship table, and capturing and logging all details of the change using a stored procedure.

10 RECOVERY STRATEGY

The requirements for backup and recovery are given in [??]. The principal constraints may be summarised as:

FUJ00152145

a) there shall be no data loss

a) service shall be resumed within ???? hours of (notification of) service loss

Requirement a) will be satisfied as follows:

1.1) The Fraud case Database will be accommodated on mirrored disk. This minimises the possibility of loss due to media failure, without undue impact on update performance. If update performance is not so critical, then the database could be built on RAID storage (level 5).

1.1) The Fraud case Database will be subject to a normal operational backup and recovery regime, with full and incremental backups taken.

Requirement b) will be satisfied as follows[2]:

2.1) If the database is corrupted, it will be recovered according to the priority schedule given in Section 10.1 “Priority Schedule for Data Recovery” (below).

2.1) Processes within the data warehouse will be recovered according to the priority schedule given in Section 10.2 “Priority Schedule for Process Recovery” (below).

The mechanisms for achieving the above will be detailed in the Infrastructure Specification (TBA).

10.1 PRIORITY SCHEDULE FOR DATA RECOVERY

If the database becomes corrupt, it shall be recovered according to the priority given below (in decreasing order of precedence):

1. Database Schema
2. Fact table tablespaces
3. Audit table tablespaces
4. Reference data tablespaces
5. Index tablespaces

[2] assuming software failure or loss of database without loss of server or media (disaster recovery is beyond the scope of this document)

FUJ00152145

10.2 PRIORITY SCHEDULE FOR PROCESS RECOVERY

If software running on the data warehouse server crashes, it will be restarted according to the priority given below (in decreasing order of precedence):

1. Operating system services
2. Database Server(s)
3. Scheduling services
4. Re-establish database links

11 TEST STRATEGY

The test strategy needs to cover all aspects of the FRMS System.

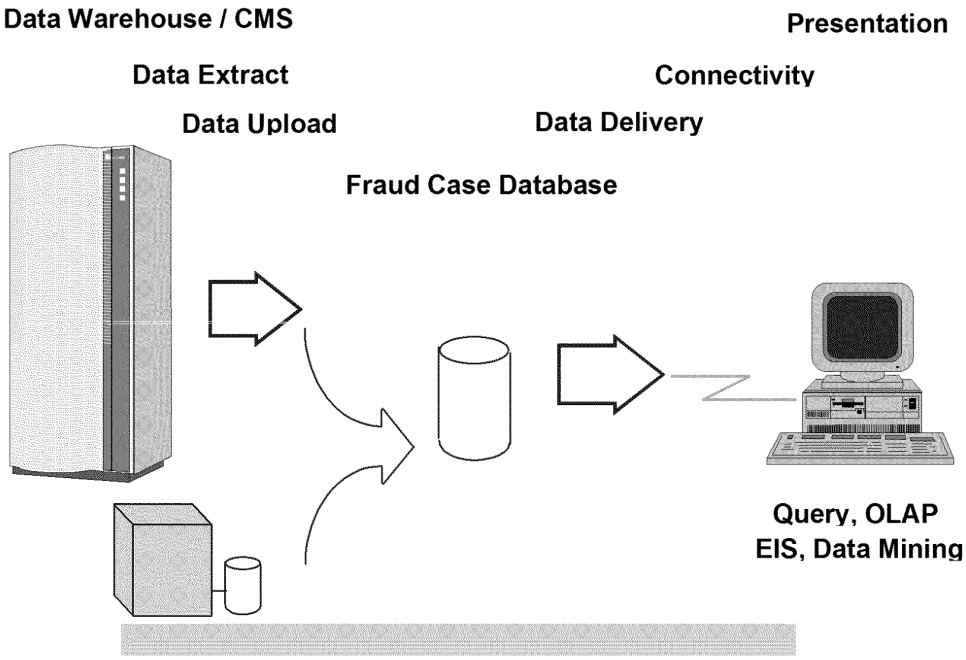

Figure 6 – Data Warehouse Process

As can be seen from the diagram there will need to be a test programme devised for:

- Overnight Harvest Processing

  ⇒ The harvesting programs that extract the required CMS card holder and card data from CMS.

  ⇒ The harvesting programs that extract the archived data from

FUJ00152145

the restored data warehouse archive.

- Overnight Load Processing
  - ⇒ The loading programs that take the reference data and upload it into the Fraud Case database
  - ⇒ The loading process that upload the CMS data into the fraud case database
  - ⇒ The loading process that upload the encashment data from the restored data warehouse archive.
- Backup and Recovery Mechanism
- The Archive and Restore Programs
- Alert mechanism
  - ⇒ Notification to an administrator that an archive needs to be restored.
  - ⇒ Notification that reports were not generated.
- The Client Interface
  - ⇒ The maintenance of the RCDB reference data
  - ⇒ The business objects reporting interface
  - ⇒ The client access to the Fraud Case database

There will also need to be series of tests to cover the operational maintenance of the fraud case database:

- Database maintenance Scripts
- Unix Maintenance Scripts
- Emergency Procedures such as power failure
- Media failure

11.1 TESTING PHILOSOPHY

In an operational system testing will concentrate on making sure that the data entered into the database is correct and has integrity across the whole database. This is particularly important in testing the batch interfaces.

It is important to carry out end to end system tests early as possible in the development cycle because any performance issues on load and transfer or building of the aggregates need to be identified and fixed before acceptance testing can take place. The development team should not under estimate the amount of time that will be required for system testing and devise a comprehensive test plan at the start of the development phase.

FUJ00152145

11.1.1 TEST DATA

Test data will need to be provided early on in the development process and this will need to be in the form of source files and database tables.

Also the test data files will need to be valid and have integrity so that the other areas can also be tested such as database consistency and client interfaces. Often, for load performance reasons, data warehouses do not have data integrity rules built into the table definitions, therefore, it is necessary to have a set of queries that check for integrity after data load e.g. making sure all the dimension keys added to the fact tables are valid.

11.1.2 TEST STAGES

As with any system under development the fraud case database programs need to go through a series of testing stages which are typically:

- Code Reviews
- Unit testing
- Integration testing
- End to End System Testing
- Acceptance testing

It is currently assumed that the first three will be conducted by CFM, and the last two by CFM in conjunction with Pathway.

It is assumed that each stage will follow existing CFM standards and be under configuration management.

Code reviews and unit testing are usually carried out by the development team and when they are satisfied that the code modules have passed unit testing they are handed on to the next stage where they are integrated together and full tests run. An example of an integration test would be to load one day's set of source files then run a set of queries from Business Objects against that data and then check the results against the operational source systems. Acceptance tests would be similar to the integration tests but with the users carrying out the queries and could be carried out as part of the uses training program.

FUJ00152145

Appendix A – Fraud Case Database Sizing


"FCDB Sizing.xls"

Appendix B – Logical Data Model

The logical data model is defined in the following OLE object.


ckentdef.lis