TECHNICAL ARCHITECTURE

SPECIFICATION

<Name of Application>

V <X>
Category of Application: <A, B, or C>

<Hard code date>

Administrative Office of the Courts
Information Services Division

TABLE OF CONTENTS

1 ABOUT THIS DOCUMENT

This Technical Architecture Specification documents <Name of Application> V <X> architecture.

1.1 DOCUMENT OWNER

The Technical Architect assigned to the application project owns this document. He or she is responsible for completing this document in collaboration with the:

- Project manager
- Vendor
- California Courts Technology Center (CCTC)

At initiation of project, the architect should incorporate known details about the architecture in the appropriate topic.

1.1.1 *Scope of Documentation*

The scope of documentation for which the technical architect is responsible depends on the category of this application, which is referenced on the title page. There are three categories:

A	Provides end-user functionality to an existing environment
B	Provides end-user functionality and proposes a new (or green field) environment
C	Provides infrastructure or services to (multiple) applications of type A and B.

1.2 REVISION HISTORY

This section will be used to track changes to the document.

Change #	Date	Who	Description of changes	Sections

2 INTRODUCTION

2.1 PURPOSE

<Describe the business purpose of this application>

2.2 PROJECT OVERVIEW

<If Business Spec is available, cut appropriate information from the document and paste it here. Otherwise, provide a brief overview of the project.>

2.3 AOC TECHNICAL ARCHITECTURE MODEL

<Provide a high-level overview of existing environment and reference the ETAG document as necessary

2.4 PROPOSED ENVIRONMENT

2.4.1 *Exceptions to AOC Technical Architecture Model*

2.4.2 *Rational for Modifications to Model*

3 REQUIREMENTS

<If functional and/or business requirements are available for this application project, reference them here; otherwise gather the requirements and document them here.>

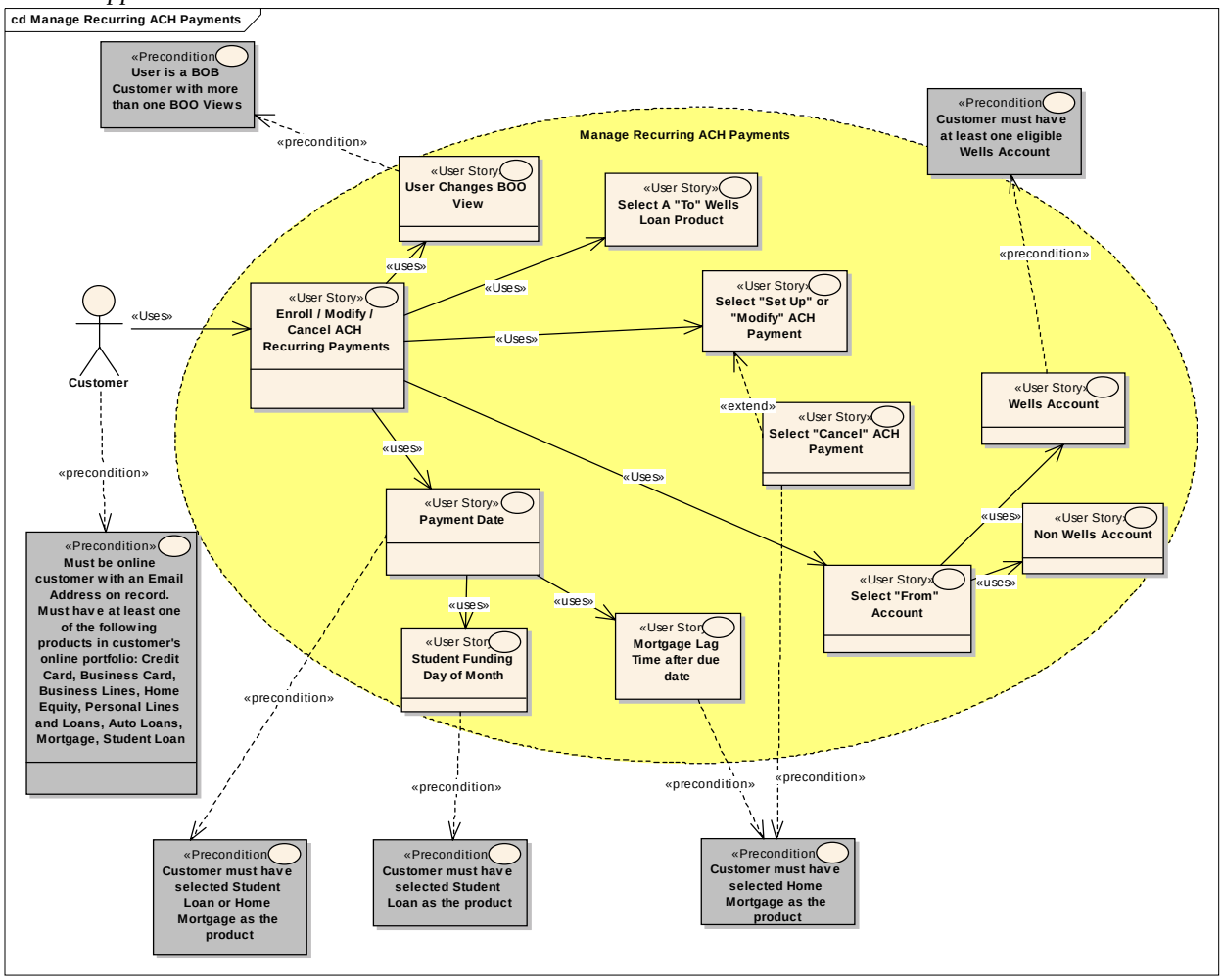
4 <APPLICATION NAME> ARCHITECTURE

4.1 USE CASES

The use cases documented in this section show all the behavior of the proposed system, including the "happy path" functionality as well as representative errors and variants. Human actors show how the user interacts with the system; system actors who how the components interact with each other. <Also included are examples of how multiple sub-use-cases realize an area of functionality.>

4.1.1 *Use Case Graphic Overview*

<The following is an example of a use case model. Replace it with a model that represents this application>



4.1.2 *Use Case Text Overview*

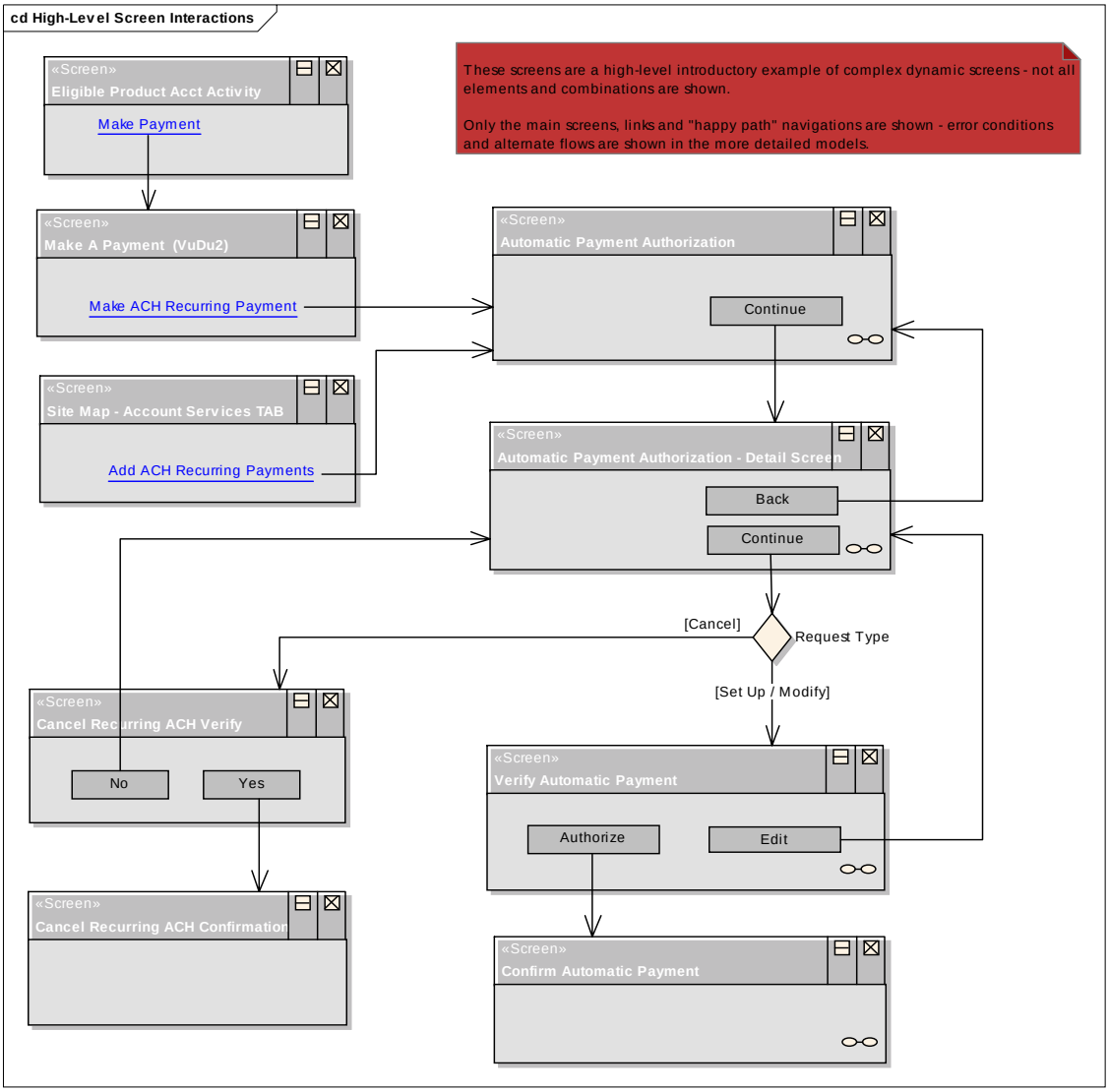
<The text in the right column provides you with an example of use case text. Replace the text with applicable information.>

Description	To change his password, the user must navigate to Siteminder's "Change Password" screen and enter his current and new password.
Preconditions	The user must have a valid current password The user must not have been locked-out by a marker in AD The new password must be valid (user enters *something*, it is different from current password, has a valid format).
Main Flow	1. To change his password, the user must navigate to Siteminder's "Change Password" screen. This can be done by a) Selecting an appropriate link from within the application b) By selecting the "Change Password" button when presented with the option from Siteminder (e.g. when the user's password is about to expire) c) By being forced onto this screen by Siteminder (e.g. on first use of a "temporary password") 2. The user must enter his current password correctly and then a new password twice. 3. User presses the "Change Password" button
Variant 1	If the user enters the new password in an invalid format, the "change password" page is repainted with the fields cleared and an error message is displayed, e.g. "new password format error: password must be X characters."
Error 1	User enters incorrect current password …
Post-condition	If the password is changed successfully, a confirmation screen displays to tell the user.

4.2 SCREEN MOCKUPS/WIREFRAMES

4.3 CHANGED SCREENS

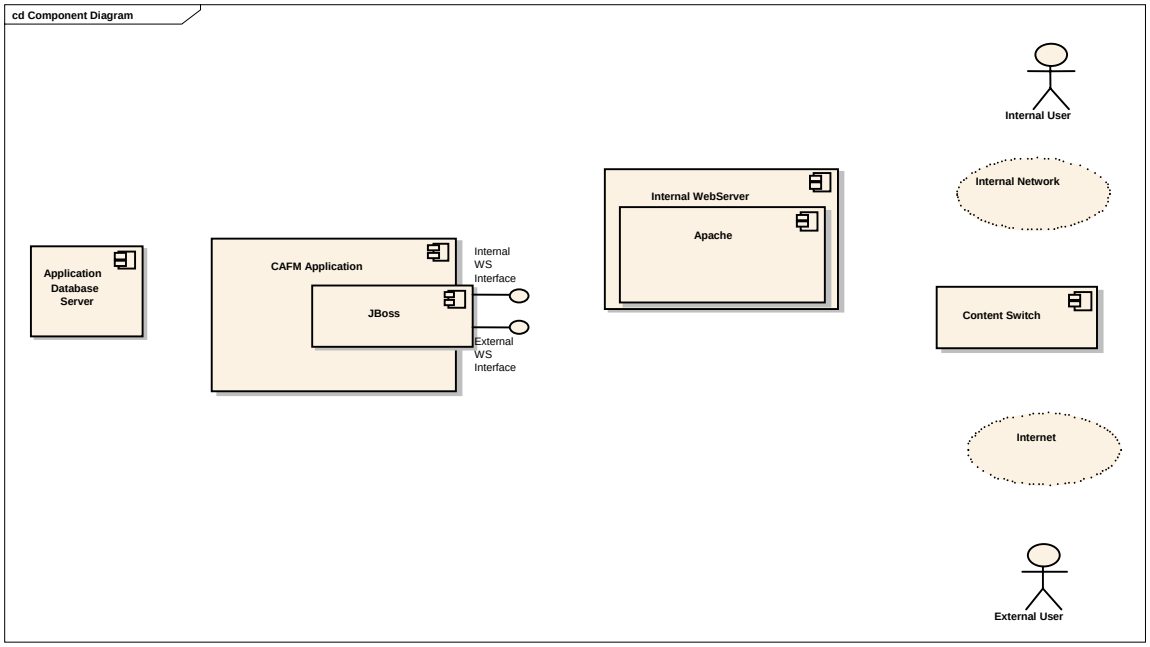
4.4 SCREEN MODELS



4.5 LOGICAL COMPONENT ARCHITECTURE

4.5.1 *Existing Logical Component Architecture*

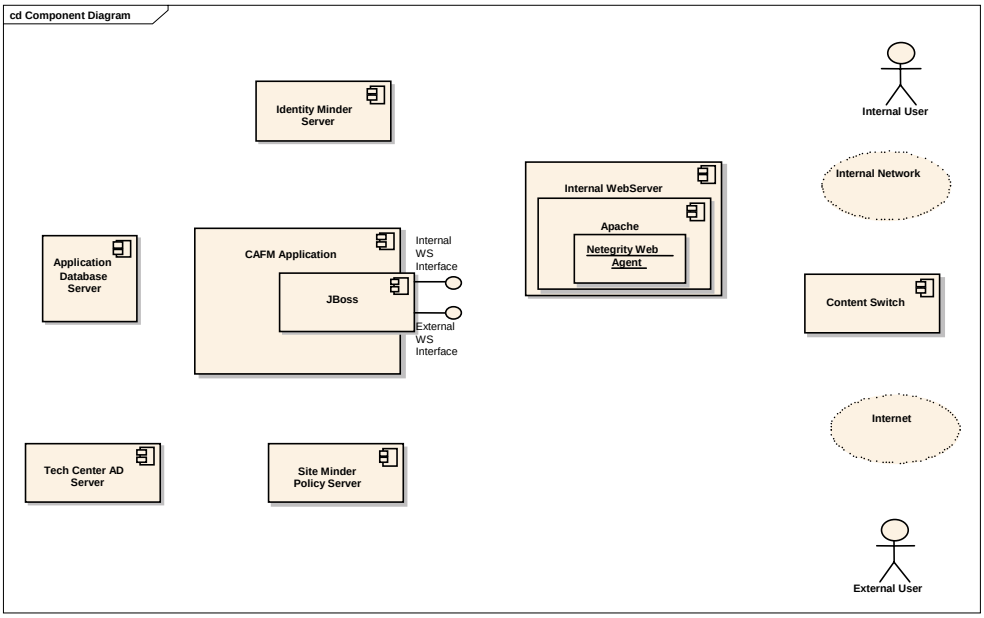
These diagrams show all the components of the existing logical component architecture into which the newly proposed architecture must fit. It will reference the ETAG document as necessary.



4.5.2 *Proposed Logical Component*

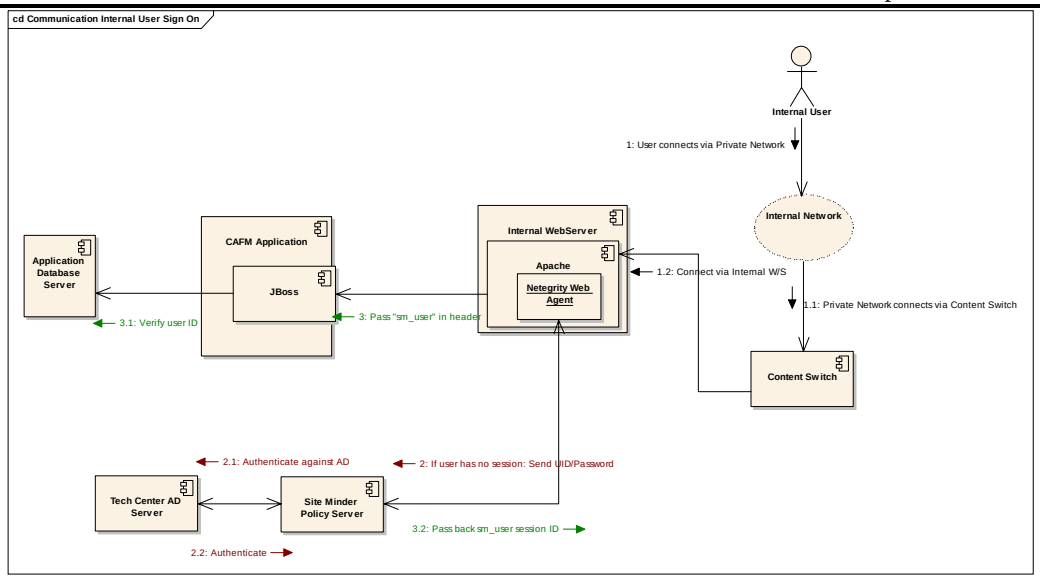
These diagrams show how the logical component architecture will look once the proposed changes are made.

Example:



4.6 COMPONENT COLLABORATIONS

These diagrams show how the logical components collaborate to realize the use cases. A few well-chosen examples are often enough to show how all the functionality will be achieved.


cd Communication Internal User Sign On
Internal User
1: User connects via Private Network
Internal Network
Internal WebServer
Apache
Netegrity Web Agent
1.2: Connect via Internal W/S
1.1: Private Network connects via Content Switch
Content Switch
CAFM Application
JBoss
Application Database Server
3.1: Verify user ID
3: Pass "sm_user" in header
2.1: Authenticate against AD
2: If user has no session: Send UID/Password
Tech Center AD Server
Site Minder Policy Server
3.2: Pass back sm_user session ID
2.2: Authenticate

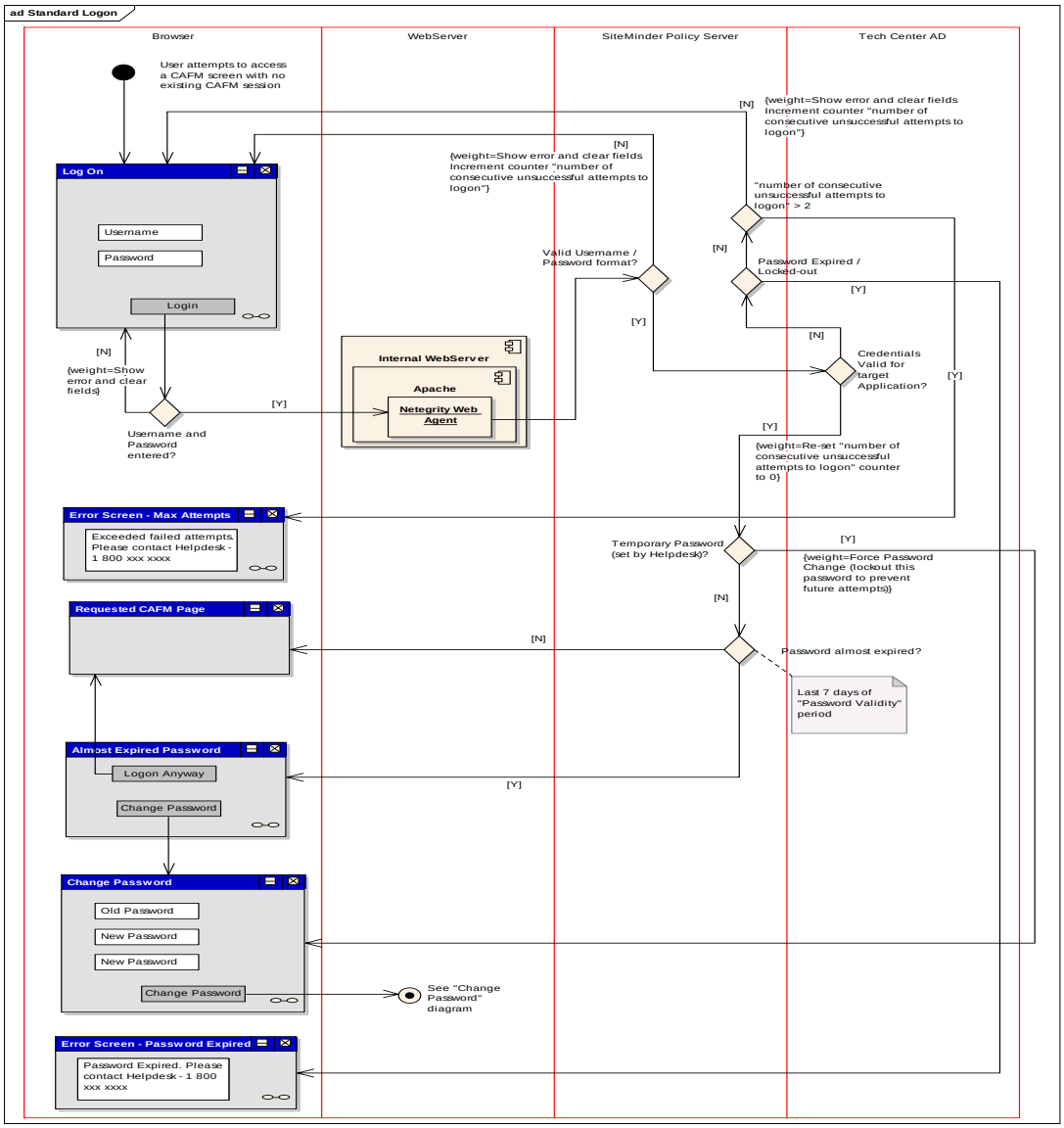
4.7 (SCREEN) ACTIVITY DIAGRAMS

Based on the use cases and the logical component model: These diagrams show how the logic, performed by individual components, collaborates to realize the use case. The swim-lanes represent logical architectural components and the logical steps required by each are shown within the appropriate swim-lane. Where details of the components are unknown, high-level versions of these diagrams can show just 2 lanes: Actor and System.

Since the discrete pieces of logic are linked by control-flow[1] arrows, these diagrams show an exact order in which this logic is executed.

For projects that don't primarily focus end-user functionality (category C), the left hand column could just as easily represent systemic actors that are triggered by system events.

Example:



[1] UML 2.0

4.8 SERVICE PROVISION

This section describes how other applications use the service.

NOTE: Must be completed if design provides enterprise services (Category C application).

4.8.1 *Overview of Services*

4.8.2 *Prequisites for Usinge Service*

4.8.3 *Interface(s) to the Service*

4.8.4 *Error Handling*

4.8.5 *Availability*

4.8.6 *Performance*

4.8.7 *License Requirements*

4.8.8 *Obligations to Cooperate with othe rUsers of the Service*

4.8.9 *Security*

4.8.10 *Known Problems*

4.8.11 *Enterprise Information Services*

This section describes how application uses functionality not housed within the application.

5 DATA ARCHITECTURE

5.1 DATA REQUIREMENTS

This section should be deduced from the business requirements and the earlier parts of this design. It should address what's being stored, why and how it will be used.

5.2 DATA MODEL (ERWIN MODELS)

A standard representation of the data required by the application. This design allows us to add in whatever data types and relations are required.

5.2.1 *Normalized Data Model*

This model is taken from the previous model but has had some rigorous structure applied through Normalization.

5.2.2 *De-normalized Data Model*

This is taken from the previous model but has been tuned to handle expected data usage efficiently. Justify each step away from Normalization.

5.3 VOLUMES AND SIZING

Talks to the expected usage of the data. This section is developed in parallel with 5.2.2

5.4 BACK-UPS / RELIABILITY / FAILOVER

Addresses the data administration, the expected performance, recovery and availability of the data.

5.5 DATA TRANSACTIONS AND ROLL-BACK

Highlights any specific data operations that must be coordinated as a group.

5.6 DATABASE PROCEDURES

Operations that should be performed within the database, e.g. report generation.

5.7 S/W AND H/W TO HOUSE DATA

The type of database to meet these requirements.

5.8 DATA ACCESS

How the application will interact with the database.

5.9 INTER COMPONENT DATA TRANSPORT

How the components will pass data between themselves. This might cover: component APIs, XML standards, Business Object data types, etc.

6 PHYSICAL INFRASTRUCTURE

6.1 PHYSICAL MODEL

6.1.1 *Graphic Overview*

6.1.2 *Enterprise Integration Overview*

6.1.2.1 IP Scheme

6.1.2.2 VP LANs

6.1.2.3 <Other>

6.1.3 *Workstations*

6.1.4 *Workstations*

6.1.4.1 User Workstation

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.4.2 <Workstations unique to environment>

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5 *Servers*

<Delete server(s) that are not applicable to this application>

6.1.5.1 WebServer

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5.2 Application Server

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5.3 Database Server

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5.4 Reports Server

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5.5 Print Server

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5.6 Mail Server

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5.7 Workflow Server

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5.8 Active Directory Server

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

6.1.5.9 Netegrity Server

Product Name	<Replace with product name>	
Hardware requirements	<Replace with minimum requirements>	
Software requirements	<Replace with minimum requirements>	
Exception to ETA model?	____ No	____ Yes DRP reference: <Paste text from DRP here>

A Appendix