

“Infrastructure Planning for VistA- EHR Implementation”

A dissertation submitted in partial fulfillment of the requirements
for the award of

**Post-Graduate Diploma in Health and Hospital Management with
Specialization in Healthcare Information Technology**

by

DIVYA CHATRATH



International Institute of Health Management Research

New Delhi -110075

January, 2011

“Infrastructure Planning for VistA- EHR Implementation”

A dissertation submitted in partial fulfillment of the requirements

for the award of

**Post-Graduate Diploma in Health and Hospital Management with
Specialization in Healthcare Information Technology**

by

DIVYA CHATRATH

Under the guidance of

Mr. Krishan Kumar Bhardwaj

Designation: Software Dev. Mgr.

Organization: DELL SERVICES

Dr. Anandhi Ramachandran

Designation: Asst.Professor

Organization: IIHMR, New Delhi



International Institute of Health Management Research

New Delhi -110075

January, 2011

Certificate of Internship Completion

Date:

TO WHOM IT MAY CONCERN

This is to certify that Miss Divya Chatrath has successfully completed her 3 months internship in our organization from August 9, 2010 to November 9, 2010. During this internship she has worked on “**Infrastructure Planning for VistA- EHR Implementation**” under the guidance of me and my team at DELL Services.

We wish him/her good luck for his/her future assignments.

(Signature)

_____ (Name)

_____ Designation

Certificate of Approval

The following dissertation titled "**Infrastructure Planning for VistA- EHR Implementation**" is hereby approved as a certified study in management carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite for the award of **Post-Graduate Diploma in Health and Hospital Management** for which it has been submitted.

It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the dissertation only for the purpose it is submitted.

Dissertation Examination Committee for evaluation of dissertation

Name	Signature
_____	_____
_____	_____
_____	_____

Certificate from Dissertation Advisory Committee

This is to certify that **Miss Divya Chatrath**, a participant of the **Post- Graduate Diploma in Health and Hospital Management**, has worked under our guidance and supervision. She is submitting this dissertation titled "**Infrastructure Planning for VistA- EHR Implementation**" in partial fulfillment of the requirements for the award of the **Post- Graduate Diploma in Health and Hospital Management**.

This dissertation has the requisite standard and to the best of our knowledge no part of it has been reproduced from any other dissertation, monograph, report or book.

Faculty Advisor
Designation
IIHMR
New Delhi
Date

Organizational Advisor
Designation
Organization
Address
Date

Acknowledgement

I owe my deep sense of gratitude to **Mr. Krishan Kumar Bhardwaj** for giving me an opportunity to learn various aspects of Healthcare Information Technology with special emphasis on Technical Aspects including the Infrastructure Planning for Implementation of EHR.

My special thanks to **Dr. Rajesh Gupta**, Principal consultant and Manager, for his guidance, support, interest, involvement and encouragement. He left no stone unturned in updating us about the subject.

I also thank **Prof. Anandhi Ramachandran** and **Prof. Indrajit Bhattacharya** for their continuous guidance throughout the dissertation period.

My sincere gratitude to **Mrs. Maitreyi R. Kollegal**, Director, International Institute of Health Management Research, New Delhi, who has always been a source of motivation and inspiration.

ABSTRACT

Infrastructure Planning for VistA- EHR Implementation

By

Divya Chatrath

Technical architecture, also known as Tarchitecture, is one of several architecture domains that form the pillars of an enterprise architecture or solution architecture. It describes the structure and behaviour of the technology infrastructure of an enterprise, solution or system. It covers the client and server nodes of the hardware configuration, the infrastructure applications that run on them, the infrastructure services they offer to applications, the protocols and networks that connect applications and nodes. It addresses issues such as performance and resilience, storage and backup. The technical Architecture for any EHR Implementation in a Hospital must be reliable, secure, must have a good Business Continuity Plan and must be fault tolerable.

This project on Infrastructure Planning for EHR will cover the Technology Stack, Network Architecture, Application Architecture, Disaster Recovery Plan and Hardware requirements for Implementation of EHR in a Hospital.

The major findings are:

1. The technical architecture of VistA is built using a client server architecture. This helps to provide optimal system performance in the hospital software, as well as a richer user experience. The operating system is Linux on the server and Delphi based GUI applications and SSH applications (roll and scroll) for the client computer. **Secure Shell** or **SSH** is a network protocol that allows data to be exchanged using a secure channel between two networked devices. The encryption used by SSH is intended to provide confidentiality and integrity of data over an unsecured network, such as the Internet. GT.M or Cache hierarchical DB are used for the database.
2. The Application Architecture of VistA- EHR is the Layered Architecture. It increases application performance, scalability, flexibility, and have a myriad of other benefits.

3. The initial estimate of Hardware Requirements based on Site Assessment Data.

The Hardware Requirements studied were: i) number of workstations, ii) number of laptops, iii) number of printers, iv) number of barcode scanners, readers, v) number of COWs and other Hardware with their specifications to be procured by the Hospital Sites.

The methodology adopted was that used for a sequential development of a software in a Phased Manner that is the Waterfall Model. Also, the study of secondary data sources.

The major Data Sources were the Technical Architecture Document of VistA and the Site Assessment Data. In addition , detailed study of various company documents like VistA Manual, Technical Manual, VistA Fileman Document etc. was also undertaken.

Table of Contents

ACKNOWLEDGEMENT	6
ABSTRACT	7
LIST OF FIGURES	11
ABBREVIATIONS/KEY-WORDS	12
Part I: Internship Report	13
1.0 Organization Profile	14
2.0 Managerial Tasks Assigned.....	17
3.0 Reflective Learning	18
PART II. Dissertation on- “Infrastructure Planning for VistA- EHR Implementation”	19
PART A. DISSERTATION OVERVIEW	20
1.0 Problem Statement	21
2.0 Objectives.....	21
2.1 General Objectives:	21
2.2 Specific Objectives:.....	21
3.0 Need of the study:	22
4.0 Scope of the Study:.....	22
5.0 Benefits Of The Study:.....	22
6.0 Assumptions:	23
7.0 Data Source:	23
8.0 Work Plan:	24
PART B. PROJECT OVERVIEW.....	25
9.0 Introduction.....	26
9.1 Guiding Principles For Architecture.....	27
10.0 Literature Survey	28
11.0 Methodology:	31
12.0 VistA- Technical Architecture	32
12.1 Database of VistA-EHR	36
12.2 Application Architecture	37
12.3 VistA Technology Stack.....	41
12.4 Data Center Deployment.....	42
12.5 System Architecture	43

12.6 Disaster Recovery Plan	48
12.7 Failover In Disaster Recovery	54
12.8 Key Design Assumptions.....	58
12.9 Design and Implementation constraints	59
12.10 High Availability	61
12.11 Recoverability	63
12.12 Performance	63
12.13 Security.....	64
12.14 Stability.....	69
12.15 Maintainability	70
12.16 Scalability.....	71
12.17 Deployment and environment	72
12.18 Patient ID configuration in VistA	73
12.19 Data standardization	74
12.20 VistA Hardware Requirements.....	74
13.0 RESULTS AND FINDINGS:	78
14.0 CONCLUSION	79
C. REFERENCES	80
D. ANEXXURES	81
1. CHANGE REQUEST PLAN.....	81
2. DISASTER RECOVERY PLAN	83
3. QUESTIONNAIRE.....	84

List Of Figures

1.ARCHITECTURE AND DESIGN.....	26
2.WATERFALL MODEL.....	31
3.VISTA EHR SYSTEM.....	36
4.APPLICATION ARCHITECTURE.....	39
5.VISTA APPLICATION ARCHITECTURE.....	40
6.VISTA TECHNOLOGY STACK	41
7.DATA CENTER DEPLOYMENT.....	42
8.NETWORK DIAGRAM FOR VISTA.....	47
9. DISASTER RECOVERY STEPS.....	49
10.DISASTER RECOVERY LAYERS.....	50
11.SAN STORAGE REPLICATION OVER WAN.....	54
12.FAILOVER CLUSTERING ARRANGEMENT.....	57
13.VISTA IMAGING SYSTEM DIAGRAM.....	60
14.VISTA IMAGING NETWORK TOPOLOGY.....	61

ABBREVIATIONS/KEY-WORDS

API	Application Program Interface
BCMA	Bar Code Medication Administration
DBA	Database Administrator
DMZ	De-Militarized Zone – Term for the portion of the network between the external Internet and the internal private network. The DMZ is protected from the outside by a Firewall.
EAR	Enterprise Archive
IDS	Intrusion Detection System
ICD	International Classification of Diseases
GT.M	Greystone Technology M
GUI	Graphical User Interface
HL7	Health Level 7
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JDBC	Java Database Connectivity
LAN	Local Area Network – Communications network confined to the same physical building.
MPLS	Multiprotocol Label Switching
MRD	Machine Readable Dictionary
MUMPS	Massachusetts General Hospital Utility Multi-Programming System
NTFS	New Technology File System
ODBC	Open Database Connectivity
OPAS	Orchestra Planning and Administration System
PING	Ping is a computer network administration utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer.
RAID	Redundant Array of Independent Disks
RPC	Remote Procedure Call
SCCM	System Center Configuration Manager
SMTP	Simple Mail Transport Protocol – Standard method of delivering internet email messages
SNMP	Simple Network Management Protocol
SSH	Secure Shell is a network protocol that allows data to be exchanged using a secure channel between two networked devices
SQL	Structured Query Language
SAN	Storage Area Network
UML	Unified Modeling Language
VPN	Virtual Private Network
XML	Extensible Markup Language
XVGA	Extended Video Graphics Array, A display standard with a resolution of 1024 by 768 pixels of 256 colours

Part I: Internship Report

1.0 Organization Profile

1.1 DELL-services Profile

Dell Services (formerly Perot Systems) is an information technology services provider based in Plano, Texas, USA. Peter Altabef has served as president and chief executive officer since 2004. On September 21, 2009, Perot Systems agreed to be acquired by Dell for \$3.9 billion.

1.2 History:

H. Ross Perot and eight associates founded Perot Systems in June 1988 after having sold EDS to General Motors. Before its acquisition by Dell Inc., Perot Systems was a Fortune 1000 corporation with more than 23,000 associates and 2008 revenues of \$2.8 billion. The company maintains offices in more than 25 countries around the world, including the United States, Europe, India, China and Mexico.

1.3 Acquisition:

The acquisition resulted in a compelling combination of two iconic information-technology brands. As a top-five finisher for the third consecutive year, Perot Systems was named to the Fortune magazine “Most Admired Companies in America” list for IT Services in 2008. Company ratings are based on eight criteria, including everything from investment value and quality of products/services to innovation and quality of management.

The expanded Dell is better positioned for immediate and long-term growth and efficiency driven by:

- Providing a broader range of IT services and solutions and optimizing how they’re delivered.
- Extending the reach of DELL Services’ capabilities, including in the most dynamic customer segments, around the world.

1.4 Location:

Express Way, Noida

Perot Systems TSI (India) Ltd.

Corporate Office Plot No. 3, Sector-125, Noida- 201301, U.P

1.5 Vision:

Dell services will be the most trusted industry leader in global information technology and business process services.

1.6 Mission:

- Dell services will be a vital contributor to the overall success of dell.
- Through our expertise execution and professional integrity we will develop and maintain lasting relationships with our customer.
- We will develop and deploy advanced and differentiated
- Support and next generation services, deepen our industry domain expertise, and expand our geographic depth and presence.
- We will always deliver real and measurable results for our customers.
- We will invest in training and development for our team, value and respect one another, focus maniacally on serving our customers and have fun doing it.
- The CIO organization will be recognized for technical excellence and industry, leading efficiency, planning and execution

1.7 Industries:

- Consumer Products and Services
- Federal Government
- Financial Services
- Logistics & Distribution
- Healthcare
- Insurance
- Manufacturing
- Telecommunications
- Travel and Transportation

1.8 Services:

Dell Services is a worldwide provider of information technology

- Application services like Application Modernization.
- Business process services like Finance and Accounting Solutions.
- Consulting services like Finance and Accounting Solutions.
- Infrastructure services like End-User Computing.
- Virtual services like Cloud Integration Services.

1.9 Healthcare IT Vertical of Dell Services

In Healthcare IT³, DELL Services provides various IT solutions to the healthcare provider's. Dell Services provides the right combination of clinical and IT tools to help healthcare providers in this rapidly advancing region of the world achieve an environment that is interconnected, streamlined, efficient, and patient-focused. Their comprehensive experience with healthcare systems architecture will help any organization to build the best possible foundation now and help it prepare for future growth and expansion. Their clinical implementation solutions are a gold-standard, holistic approach that includes change management as well as clinical and business process optimisation that leverages the value of applications and ensures end user buy-in. One of the healthcare IT solution provided by DELL is EHR (Electronic Health Record).

1.9.1 Electronic Health Record (EHR)

An Electronic Health Record is an evolving concept defined as a systematic collection of electronic health information about individual patients. It is a record in digital format that is capable of being shared across different healthcare settings by being embedded in network-connected enterprise wide information systems.

1.9.2 Advantages of an Electronic Health Record:

- Easy access to information
- Comprehensive and standardized documentation

- Improved quality of patient care
- Increased efficiency of healthcare professionals
- Improved process communication
- Reduced medication errors
- Meet various accreditation requirements
- Reduced TPA denials
- Better control of Management
- Reduced pilferages

1.9.3 Number of Departments allotted for EHR implementation³:

EHR Implementation:

- Clinical Transformation
- EHR
- Training
- Infrastructure and Application
- Integrating HIS System

2.0 Managerial Tasks Assigned

The department allotted to me was EHR Technical Department where I worked under Software Dev.Manager who was working on Technical Architecture of VistA- EHR. Following tasks were allotted to me during this training:

1. Prepared the presentation on technical architecture of VistA- EHR to be presented to the client hospital by DELL'S Technical Team.
2. Collected data on Imaging Modalities through site assessments.
3. Made a presentation on technical and clinical recommendations.
4. Installed VistA- CPRS(Computerized Patient Record System) on various systems in the organization.
5. Calculated Hardware to be procured by the client Hospital on the basis of VistA hardware master sheet.

3.0 Reflective Learning

The key learnings from the complete internship tenure are as follows:

1. Learnt various technical aspects of a software to be implemented, learnt structure and design of web-bases applications.
2. Learnt the step by step process to install VistA (CPRS)
3. Types of hardware required for Infrastructure Planning and specifications of these hardware.
4. Application Architecture- 2-tier, 3-tier and N-tier structure. (Layered Architecture)
5. Learnt about VistA- Fileman (Database of VistA).
6. Overview of GT.M. (Database) and MUMPS (Language).
7. Learnt the Methodolgy adopted for VistA- EHR Implementation Project.
8. Key features and limitations of Hospital Information System.
9. Learnt to make Templates- SOAP Format of templates.

PART II. Dissertation on- “Infrastructure Planning for VistA- EHR Implementation”

PART A. DISSERTATION OVERVIEW

1.0 Problem Statement

The problem identified is to outline the technical design of VistA- Electronic Health Record to be implemented in the Client Hospital and provide an overview of an Architectural Deployment for EHR.

Its main purpose is to -

1. Provide the Infrastructure Planning for EHR Implementation Project.
2. Detail the functionality which will be provided by each component or group of components and show how the various components interact in the design.
3. Provide a basis for the detailed design and implementation of VistA- EHR.

This document is not intended to address installation and configuration details of the actual implementation.

2.0 Objectives

2.1 General Objectives:

- Create an integrated platform to drive the capture of complete patient diagnosis and to help improve quality of Healthcare by reducing wrong medication.
- Implement VistA EHR for the clinical requirements of Hospital.
- Use Clinical Transformation methodology ADOPTS to drive user adoption of VistA and the hospital derive the expected return on investment on VistA.

2.2 Specific Objectives:

- To study the architectural deployment for EHR Implementation.
- Technical Design Assumptions and Constraints.
- Design a Disaster Recovery Plan for VistA.
- Analyzing the need for Hardware Requirements for VistA-EHR Implementation.

3.0 Need of the study:

The need of this project is to design the architecture for an EHR implementation project including the various hardware and software requirements for the development of the same, and interaction between these components. This study also aims to develop a Disaster Recovery Plan for VistA- EHR and to site various hardware requirements for implementation of this project in the client hospital.

Thus, the main rationale behind this study is to know the technical aspects of any EHR software being implemented in a Hospital.

4.0 Scope of the Study:

The study mainly analyzes the technical requirements of the EHR Project for its successful Implementation. The results of this study will give us the Architectural Deployment for VistA- EHR including the Network Architecture, Application Architecture and System Architecture. This study also helps in gathering the hardware requirements for the project like number of workstations, number of laptops, number of printers, barcode scanners/readers to be procured by the client hospital along with their detailed specifications.

5.0 Benefits Of The Study:

1. The Infrastructure Design talked about in this study can be used for any EHR Project being implemented in a Hospital whose Architectural Deployment is similar to that of VistA.
2. The Disaster Recovery Plan and Failover Mechanisms in the study will give a complete picture of steps to be adopted for Business Continuity Planning to prepare before a Disaster strikes that is necessary for an organization before any project is implemented.
3. The Design for VistA- EHR has been developed after a complete Requirement Analysis.
4. The Hardware and Software Requirements for VistA elicited in this project will give complete details to the client about what all Hardware is to be procured.

5. This study covers the entire Architecture Deployment for software before its Implementation including its Network Topology, Hardware, and Software and how these components relate to each other.

6.0 Assumptions:

1. User Acceptance Test/System Acceptance Test Environment will be available for performing Usability testing, Installation and Configuration testing and Performance Testing.
2. General Architecture principles based on past experiences and Dell's Best Practices and Methodologies will be used in designing the solution.
3. The basic TCP/IP Protocol will be the only one used to access the application.
4. The web browser will be the primary client used by employees and public users.

7.0 Data Source:

The major Data Sources were:

1. Technical Architecture Document of VistA
2. Site Assessment Data- Hardware and Software data.
3. Various company documents like VistA Manual, Technical Manual, VistA Fileman Document, HIS Manual etc.

8.0 Work Plan:

ID	Task Name	Start	Finish	Duration	Aug 2010			Sep 2010			Oct 2010							
					8/8	15/8	22/8	29/8	5/9	12/9	19/9	26/9	3/10	10/10	17/10	24/10	31/10	
1	Defining the Problem	8/9/2010	8/13/2010	1w														
2	Literature Survey	8/13/2010	8/26/2010	2w														
3	Methodology Adopted	8/26/2010	9/1/2010	1w														
4	Data Collection	9/1/2010	9/28/2010	4w														
5	Compilation and Analysis	9/28/2010	10/18/2010	3w														
6	Documentation	10/18/2010	11/5/2010	3w														

PART B. PROJECT OVERVIEW

9.0 Introduction

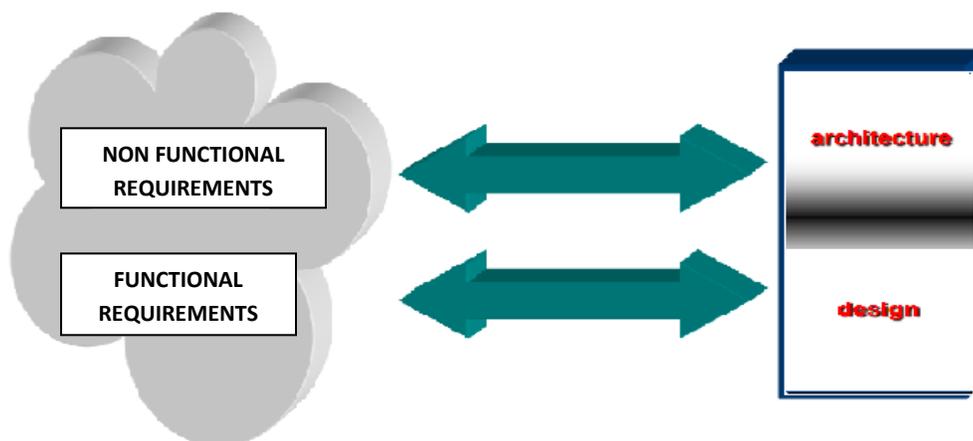
Architecture² is the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution. Architecture encompasses the set of significant decisions about the organization of a software system:

- Selection of the structural elements and their interfaces by which a system is composed
- Behavior as specified in collaborations among those elements
- Composition of these structural and behavioral elements into larger subsystems
- Architectural style that guides this organization

Every system has an architecture (even a system composed of one component). Architecture defines the rationale behind the components and the structure. Architecture serves as the blueprint for the system but also the project:

- Team structure
- Documentation organization
- Work breakdown structure
- Scheduling, planning, budgeting
- Unit testing, integration

Fig.1 Architecture And Design



Architecture: where non-functional decisions are cast, and functional requirements are partitioned

Design: where functional requirements are accomplished

9.1 Guiding Principles For Architecture

Guiding principles¹³ provide a foundation upon which to develop the target architecture for the EHR, in part by setting the standards and measures that the Software must satisfy. These in turn drive design principles that can be used to validate the design and ensure that it is aligned the EHR overall Architecture, Design Principles and Standards.

Some of the guiding principles that will be followed during the VistA EHR Architectural design and development are outlined below:

1. Scalable

Scalability is the ability of the platform to scale both up and down to support varying numbers of users or transaction volumes. The application should be able to scale horizontally (by adding more servers) or vertically (by increasing hardware capacity or software efficiency).

2. Flexible

Flexibility is the ability of the application to adapt and evolve to accommodate new requirements without affecting the existing operations. This relies on a modular architecture, which isolates the complexity of integration, presentation, and business logic from each other in order to allow for the easy integration of new technologies and processes within the application.

3. Standards-Based

Compliance with established industry standards. The standards-compliance will not only apply to application development but also to design, platform/infrastructure and other parts of the Application.

10.0 Literature Survey

10.1 An Infrastructure for Integrated Electronic Health Record Services⁸

Dimitrios G Katehakis; Stelios Sfakianakis; Manolis Tsiknakis; Stelios C Orphanoudakis

ABSTRACT

Background: The sharing of information resources is generally accepted as the key to substantial improvements in productivity and better quality of care. In addition, due to the greater mobility of the population, national and international healthcare networks are increasingly used to facilitate the sharing of healthcare-related information among the various actors of the field. In the context of HYGEIAnet, the regional health telematics network of Crete, an Integrated Electronic Health Record environment has been developed to provide integrated access to online clinical information, accessible throughout the island.

Objectives: To make available comprehensive medical information about a patient by means of incorporating all the distributed and heterogeneous health record segments into an Integrated Electronic Health Record that can be viewed on-line through a unified user interface and visualization environment.

Methods: The technological approach for implementing this Integrated Electronic Health Record environment is based on the HYGEIAnet Reference Architecture, which provides the necessary framework for the reuse of services, components, and interfaces. Seamless presentation of information is achieved by means of the Extensible Markup Language (XML), while its underlying capabilities allow for dynamic navigation according to personalized end-user preferences and authorities.

Results: The Integrated Electronic Health Record environment developed in HYGEIAnet provides the basis for consistent and authenticated access to primary information over the Internet in order to support decision-making. Primary information is always kept at the place where it has been produced, and is maintained by the most appropriate clinical information system, contrasting traditional store and forward techniques, or centralized clinical data repositories.

Conclusions: Since documents are much more easily accessible rather than data inside a database, Extensible Markup Language has the potential of becoming a very cheap technology provided, of course, that the underlying Healthcare Information Infrastructure exists. XML can be introduced incrementally and its implementation is completely transparent to the end user.

10.2 The challenge of electronic health records (EHRs) design and implementation⁹

[Jenkins KN, Wilson RG.](#)

Abstract

BACKGROUND AND AIM: To investigate the use of animation tools to aid visualisation of problems for discussion within focus groups, in the context of healthcare workers discussing electronic health records (EHRs).

METHOD: Ten healthcare staff focus groups, held in a range of organisational contexts. Each focus group was in four stages: baseline discussion, animator presentation, post-animator discussion and questionnaire. Audio recordings of the focus groups were transcribed and coded and the emergent analytic themes analysed for issues relating to EHR design and implementation. The data allowed a comparison of baseline and post-animator discussion.

RESULTS: The animator facilitated discussion about EHR issues and these were thematically coded as: Workload; Sharing Information; Access to Information; Record Content; Confidentiality; Patient Consent; and Implementation.

CONCLUSION: We illustrate that use of the animator in focus groups is one means to raise understanding about a proposed EHR development. The animator provided a visual 'probe' to support a more proactive and discursive localised approach to end-user concerns, which could be part of an effective stakeholder engagement and communication strategy crucial in any EHR or health informatics implementation programme. The results of the focus groups were to raise salient issues and concerns, many of which anticipated those that have emerged

in the current NHS Connecting for Health Care Records programme in England. Potentially, animator-type technologies may facilitate the user ownership which other forms of dissemination appear to be failing to achieve.

10.3 Health Information Exchange: Architecture Implementation Guide¹⁰

This document specifies a technical architecture designed by **Connecting for Health** for communication of protected health information between sub-network organizations (SNOs) on a Nationwide Health Information Network (NHIN). The architecture and messaging standards discussed here make up part of **Connecting for Health's** Common Framework, a proposed collection of technical standards and policies designed to make it possible to build a NHIN that is effective and achievable in incremental steps, and supports the required discovery and transport of patient records between authorized parties while protecting those records from unauthorized access or use.

Technical interactions between entities in the NHIN involve transactions between software clients (typically an EHR, secure browser, or proxy server, and called NHIN clients throughout) and either the RLS or the ISB.

The Record Locator Service (RLS) is essentially a master patient index within a SNO that refers to record locations at more than one institution within that SNO. It has two required interactions with the participating entities in a SNO: it accepts updates to patient demographics and record locations; and it accepts queries for the location of patient records and returns record locations when it finds matches.

11.0 Methodology:

The Methodology adopted for the project is The **waterfall model** that is a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through various phases. The phases are as follows:

1. Requirements specification
2. Design
3. Construction (implementation or coding)
4. Integration
5. Testing and debugging(Validation)
6. Installation
7. Maintenance

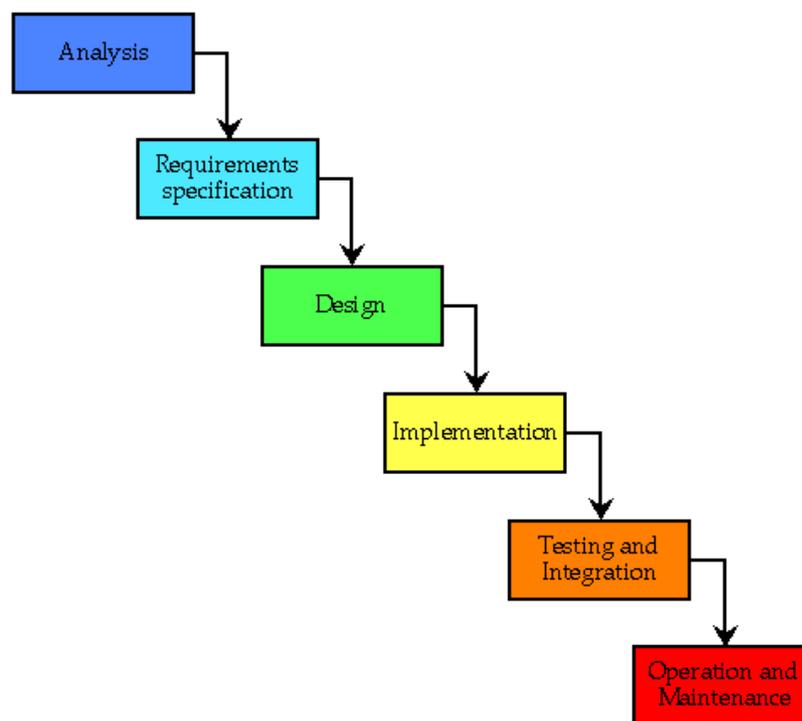


Fig.2 Waterfall Model

12.0 VistA- Technical Architecture

- Complete EMR Solution - Veterans Health Information Systems Technology & Architecture
 - Electronic Medical Record
- Over 130 clinical modules to select from (VistA Monograph)
- Thousands of man years of code development along with an evolving architecture
- Thousands of application programs (business logic) wrote in Mumps.
- Infrastructure provided by many platforms and architectures

WorldVistA¹¹ is an open source implementation of the Veteran Administration's Electronic Health Record system intended for use in health care facilities outside the VA.

Background

The US Veterans Administration developed the most widely distributed Electronic Health Record used in the US, the Veterans Health Information Systems and Technology Architecture (VistA). In an effort to make the system widely available to institutions outside the Veterans Administration health system, the software code was placed in the Public Domain under the Freedom of Information Act.

The foundation for the WorldVistA EHR was formed to extend and collaboratively improve the VistA electronic health record and health information system for use outside of its original setting. It was originally developed as part of the VistA-Office project, a collaborative effort funded by the United States Centers for Medicare and Medicaid Services (CMS), an agency of the US Department of Health and Human Services (DHHS).

WorldVistA EHR VOE/ 1.0 is based on and compatible with the U.S. Department of Veterans Affairs (VA) world renowned EHR, Veterans Health Information Systems and Technology Architecture (VistA). A fully open-source (GPL v2 licensed) project, WorldVistA has also developed software modules (such as pediatrics, obstetrics, and other functions) not used in the veterans' healthcare setting.

In 2006, WorldVistA EHR VOE/ 1.0 was the only open source EHR that met Certification Commission for Healthcare Information Technology (CCHITSM) ambulatory electronic health record (EHR) criteria, and in January 2008, it was released with full CCHITSM EHR.

As a free product developed in co-operation with the US government, WorldVistA is not marketed in a similar fashion to commercial EHRs.

Core VistA functions

- patient registration
- clinical reminders for chronic disease management
- clinical order entry
- progress note templates
- results reporting

Customizable functions

The structure of WorldVistA is modular, and a wide variety of customization is possible. Because it is fully open source, this can be done without restriction (although CCHIT certification is granted only to the officially maintained package).

- ability to interface to existing practice management / billing systems, lab services and other applications
- scanning and inclusion of scanned documents into the medical record
- prescription finishing and faxing
- clinical quality measure reporting capabilities
- support for disease management, using clinical reminders
- templates for obstetrics/gynecology (OB/GYN) and pediatrics care

Server Platforms⁶

- For Linux-based servers³, WorldVistA server uses the (free open source) Fidelity GT.M MUMPS database, available as an integrated package along with WorldVistA Server. This software is part of the VistA Public Domain software, and does not require licensing.
- For Windows-based servers, WorldVistA can be implemented used the commercial Caché MUMPS database, which requires a database license and software from Intersystems Corporation.
- For Mac OS-X-based servers, a development effort to port GT.M (and the Server software) to that platform has begun.

Client platforms⁶

- The Client software is an implementation of CPRS, which is Windows-based. This allows Windows terminals to access the central server database. This software is part of the VistA Public Domain software, and does not require licensing.
- For Linux terminals, CPRS²⁴ can be run as a Wine package or from within a virtual machine.
- A separate (Windows-based) module is available to capture and view vital signs as well as graphing of other clinical data. This is meant to be used on client terminals.
- A separate (Windows-based) module allows the scanning, capture and integration of paper documents as part of an individuals medical record. It can also be used to add a variety of non-diagnostic quality images to the medical record. This is meant to be used on client terminals.

Development History

WorldVistA is developed by a series of physicians (and other medical professionals) and software professionals that donate their efforts as volunteers. This group loosely referred to themselves as Hardhats (and continues to do so) before the name of the project was officially changed to WorldVistA.

WorldVistA¹¹ has developed and distributes a "toaster" version of VistA, which is a self-contained software package that integrates both the MUMPS database (GT.M version) and the VistA software.

In 2009, the self-installing Linux toaster version was enhanced with a GUI-based patient registration module, web interface, and other enhancements, and incorporated into a self-installing package for both Debian/Ubuntu and Red Hat Linux. This freely available version of WorldVistA is known as Astronaut VistA. This version is packaged with both an enhanced GUI as well as a web interface (which allows connection through an intranet or through the Internet).

A similar package for Windows-based servers is in alpha (early development) stage.

VistA functionality for the EHR solution for end users can be divided into the following modules. These modules are:

- CPRS – Computerized Patient Record System
- Radiology – Roll and Scroll
- VistA Lab – Roll and Scroll
- VistA Imaging – This is a GUI and linked to CPRS²⁴
- Pharmacy – Roll and Scroll
- Surgery
- Dietitics
- PIMS- Patient Information Management System

12.2 Application Architecture

The term "application architecture"¹⁴ is commonly used for the *internal* structure of an application, for its software modularisation. Application architecture is the critical software that bridges the architectural gap between the application server and the application's business logic, thereby eliminating the complexities and excessive costs of constructing, deploying and managing distributed enterprise applications. Applications Architecture is the science and art of ensuring the suite of applications being used by an organization to create the composite application is scalable, reliable, available and manageable.

The Application Architecture of VistA- EHR is the Layered Architecture i.e. the design is based on a tiered approach. "A tier is a logical partition of the separation of concerns of the system. Each tier is assigned its unique responsibility in the system. We view each tier as logically separated from one another. Each tier is loosely coupled with the adjacent tier." A decomposition of services such that most interactions occur only between neighboring layers. Layered application designs increase application performance, scalability, flexibility, and have a myriad of other benefits. When used appropriately, a layered design can lessen the

overall impact of changes to the application. Layered application architecture provides some of the key features below –

- **STRUCTURE:** Organizing applications along business-level boundaries and not technical boundaries.
- **SPEED & FLEXIBILITY:** Making application changes through configuration and not programming.
- **CONTROL:** Modifying, extending or overwriting any architectural element.
- **REUSE:** Achieving greater reusability and integration by loosely coupling application logic to infrastructure.

The layers include:

- **The data layer:** manages the physical storage and retrieval of data
- **The business layer:** maintains business rules and logic
- **The presentation layer:** houses the user interface and related presentation code.

Inside each of these layers, there may also exist a series of sub-layers that provide an even more granular break up the functional areas of the application.

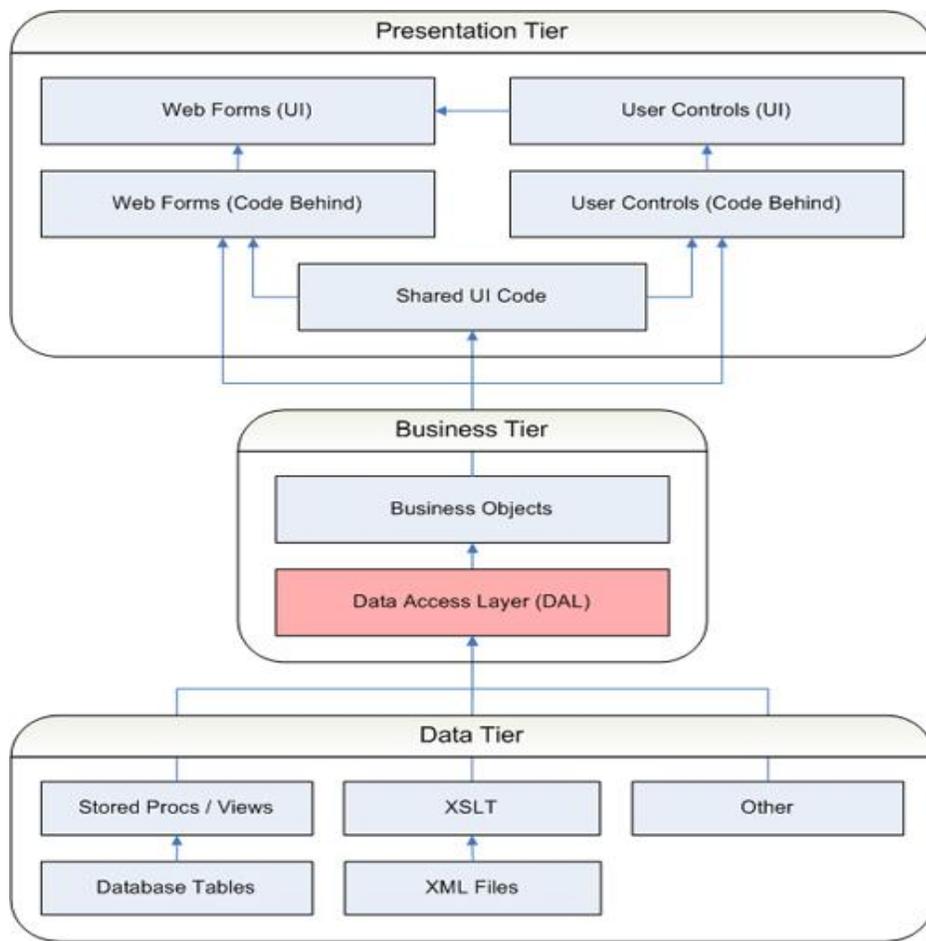
The presentation tier: The markup file defines the look and layout of the web form and the code behind file contains the presentation logic. It's a clean separation because both the markup and the code-behind layers house specific sets of functionality that benefit from being apart.

The data tier: Tables define the physical storage of data in a database, but stored procedures and views allow you to manipulate data as it goes into and out of those tables. If you access tables directly in the business layer, then you are forced to update your business tier to account for the changes to the table. If you use a layer of stored procedures and views to access the data, then you can expose the same logical structure by updating a view or stored

procedure to account for the physical change without having to touch any code in your business layer.

The business tier: Two sub-layers within the business tier- business objects and the Data Access Layer (also known as the DAL). A business object is a component that encapsulates the data and business processing logic for a particular business entity. It is not, however, a persistent storage mechanism. Since business objects cannot store data indefinitely, the business tier relies on the data tier for long term data storage and retrieval. Thus, your business tier contains logic for retrieving persistent data from the data-tier and placing it into business objects and, conversely, logic that persists data from business objects into the data tier. This is called data access logic.

Fig.4. Application Architecture



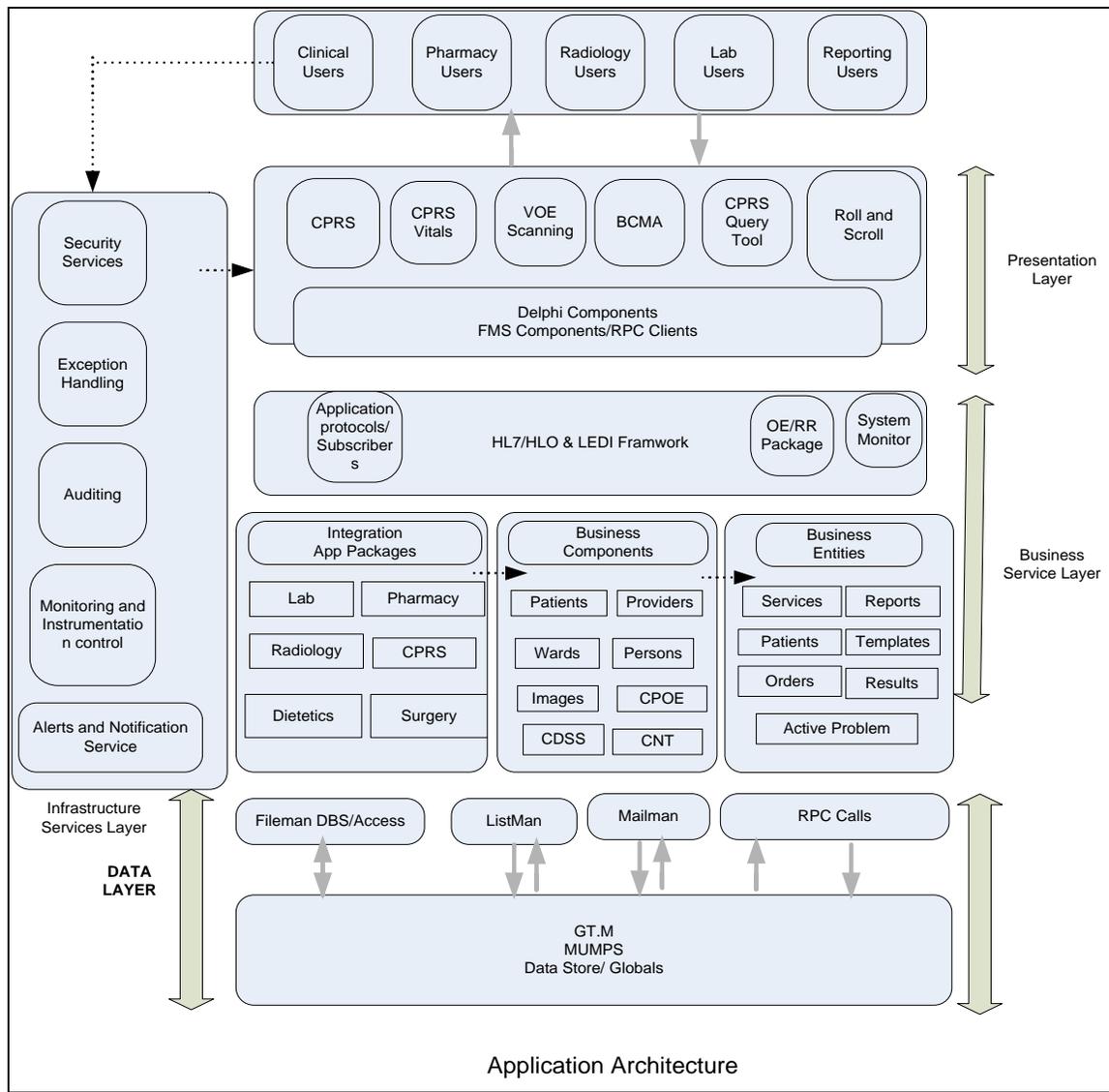


Fig.5. VistA Application Architecture

GT.M¹⁵, an abbreviation for *Greystone Technology M*, was developed by the Greystone Technology Corp in the 1980s. **GT.M** is a high-throughput key-value database engine optimized for transaction processing. (It is a type also referred to as "schema-less", "schema-free," or "NoSQL.") GT.M is also an application development platform and a compiler for the ISO standard M language, also known as MUMPS.

The database engine, made open source in 2000, is maintained by Fidelity Information Services. It is used as an open source backend for the US Department of Veterans Affairs

Electronic Health Record system WorldVista and other open source EHRs such as Medsphere's OpenVista¹². It is listed as an open source healthcare solution partner of Red Hat. Today it consists of approximately 2 million lines of code.

MUMPS¹⁶ (Massachusetts General Hospital Utility Multi-Programming System), or alternatively **M**, is a programming language created in the late 1960s, originally for use in the healthcare industry. MUMPS is a language intended for and designed to build database applications. Secondary language features were included to help programmers make applications using minimal computing resources.

12.3 Vista Technology Stack

A **technology stack** comprises the layers of components or services that are used to provide a software solution or application. Traditional examples include the OSI seven layer model, and the TCP/IP model.

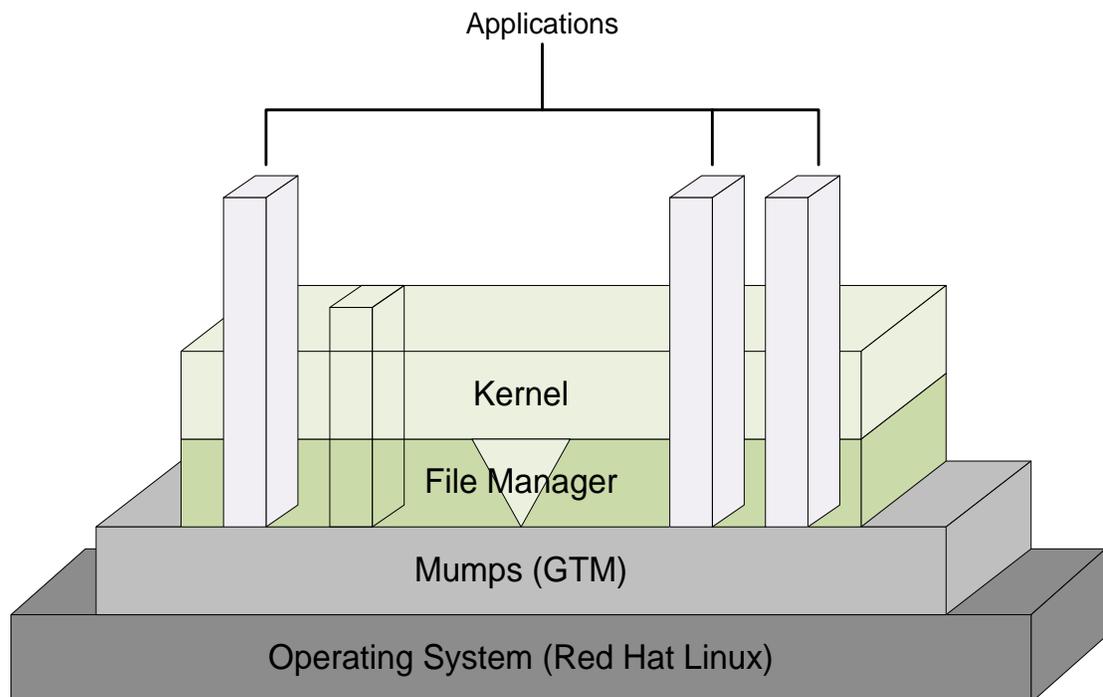


Fig.6. Vista Technology Stack

12.4 Data Center Deployment

Multiprotocol Label Switching (MPLS)¹⁷ is a mechanism in high-performance telecommunications networks which directs and carries data from one network node to the next. It is deployed to connect as few as two facilities to very large deployments. MPLS is a highly scalable, protocol agnostic, data-carrying mechanism. In an MPLS network, data packets are assigned labels. Packet-forwarding decisions are made solely on the contents of this label, without the need to examine the packet itself. This allows one to create end-to-end circuits across any type of transport medium, using any protocol. The primary benefit is to eliminate dependence on a particular Data Link Layer technology, and eliminate the need for multiple Layer 2 networks to satisfy different types of traffic.

- All hospital sites are connected with Dell Noida Data center through MPLS cloud.
- VistA applications will be accessed over MPLS cloud in all Hospitals.

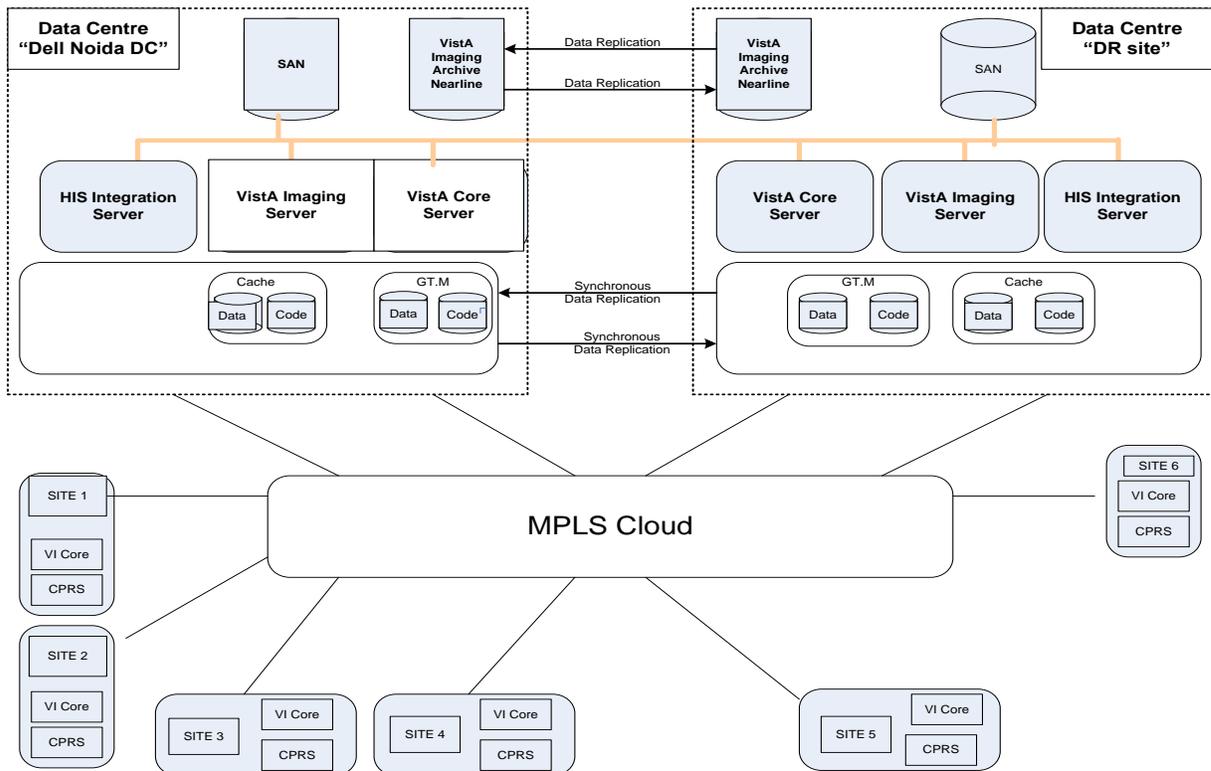


Fig.7. Data Center Deployment

12.5 System Architecture

The defining characteristic of VistA is flexibility. It has innovated at every level of the technology stack to transition VistA from a clinically effective but organizationally beholden tool to one limited neither by environment nor the inclusion of different components.

Client

The VA's VistA system employs a graphical user interface (GUI) created in Delphi that uses a Remote Procedure Call (RPC) broker to communicate with the MUMPS server. Significantly, the OpenVista client employs either the Microsoft .NET framework or Mono, an open source implementation of Microsoft .NET, to provide modern functionality in a cross-platform graphical user interface (GUI). (Interoperable, language-independent and simple to deploy, the .NET framework is key to OpenVista's flexible nature)

The OpenVista architecture enables the client to run natively on Windows or Linux systems. This flexibility allows facilities to leverage current hardware investment, expand resources, and reach a greater clinical user base.

Network

At the network level, VistA's approach to connecting client and server is an RPC broker that causes actions executed on one computer to also execute on another. Updation to this approach is through the use of the OpenVista Bridge, a middleware component that manages interaction between server and client, handles raw communication with the VistA Broker (server), and provides a Binary Remoting interface to the client (the client uses one or the other for a given connection).

The implementation of the OpenVista Bridge gives the ability to use on-the-wire encryption for greater levels of security and enables the system to interact with Web-based applications.

Server

The VistA server layer uses Java¹, a platform-independent programming language intended to run anywhere, to give the overall technology improved market viability and extensive commercial market benefits. Through the use of Java, the legacy VistA MUMPS code at both the server and application server (see below) levels is preserved while enabling the advancement of OpenVista and preserving the longevity of the product. The incorporation of recognized interfacing technology standards such as Health Level 7 (HL7) enables OpenVista to communicate effectively with required third-party applications (e.g., administrative, financial, PACS) in creating a complete, efficient solution.

Database interoperability

Leveraging open-source technologies, developing a database projection technology that will allow customers to view VistA's hierarchical FileMan (MUMPS-based) database management system as a modern, relational database via a MySQL storage engine. Use of a standard Java Database Connectivity (JDBC) interface enables commercial off-the-shelf (COTS) reporting tools and data warehousing through the use of Structured Query Language (SQL). This greatly expands OpenVista's ability to provide an organization with data analysis tools, performance metrics, and operational reporting.

Application server

At the Application Server level, use of Java technologies to provide OpenVista with more flexibility via an object domain layer. Development of this Java layer as part of the OpenVista technology stack to realize wider interoperability, develop greater application functionality, and increase development velocity. Through the use of this domain layer, OpenVista applications can seamlessly connect to legacy modules and other Java-based applications, leveraging modern development methodologies and tools. As an example, this tool is used to support interoperability with pharmacy billing systems. Medsphere provides additional flexibility at the Application Server level by giving clients the choice of either InterSystems Caché or Fidelity GT.M, a commercial Open Source solution.

Operating system

Historically, use of the OpenVMS operating system has made VistA dependent on Alpha (VAX) hardware. Medsphere also offers freedom of choice at the OS level by certifying the use of OpenVista on multiple platforms, allowing organizations to choose Linux or Windows operating systems, and hence a wider variety of hardware.

Hardware

VistA is limited to Alpha or Intel Itanium technology.

Server Hardware Configuration

Vista's core clinical systems are incredibly efficient and thus require only moderate commodity compute power even for large facilities. These requirements are purposefully generalized as every site has local requirements and architecture to consider. These quotes are meant to provide an idea of compute, memory and storage requirements. It is also important to note that these quotes are for the server that powers the core clinical systems -- depending on a specific site's needs, additional ancillary servers may be desired for utility systems such as an Interface Engine, Fax Server, Instrument Manager, etc.

Small configuration (<200 bed facility)

- Compute: Dual Intel Xeon (suggest purchase of Quad processor-capable server for upgradeability)
- Memory: 4 GB
- Storage: 1.25 TB of available disk storage (Please see notes [1] and [2] below on Capacity and Hardware requirements)
- Other: Network adapter, back-up device/system, UPS, etc.

Medium Configuration (200-500 bed facility)

- Compute: Dual Intel Xeon (suggest purchase of Quad processor-capable server for upgradeability)
- Memory: 8 GB
- Storage: 2 TB of available disk storage (Please see notes [1] and [2] below on Capacity and Hardware requirements)
- Other: Network adapter, back-up device/system, UPS, etc.

Large Configuration (>500 bed facility)

- Compute: Quad Intel Xeon
- Memory: 8 GB
- Storage: 3 TB of available disk storage (Please see notes [1] and [2] below on Capacity and Hardware requirements)
- Other: Network adapter, back-up device/system, UPS, etc.

For large medical centers or multi-site/regional installations a detailed discussion of architecture, database design, datacenter locations, business continuity, disaster recovery, etc. is required.

[1] Storage capacity requirements: The amount of storage required is directly affected by utilization at each site. Specifically, the number of scanned images/documents will be the main driver behind this number and this is determined by workflow and system integration.

[2] Storage hardware requirements: The storage solution you select should be compatible with the intended cluster configuration. Depending on your cluster technology this may be direct attached storage, a SAN, a NAS or a mixture of technologies.

Client Hardware Configuration

- Compute: 2Ghz Intel
- Memory: 1.5 GB (45 MB Available per application. - CIS, BCMA, Rad Worklist, OV Meds)
- Storage: Applications range in size, but 500MB of storage for OpenVista applications will provide more than necessary
- Other: Network adapter, Display: XVGA or better

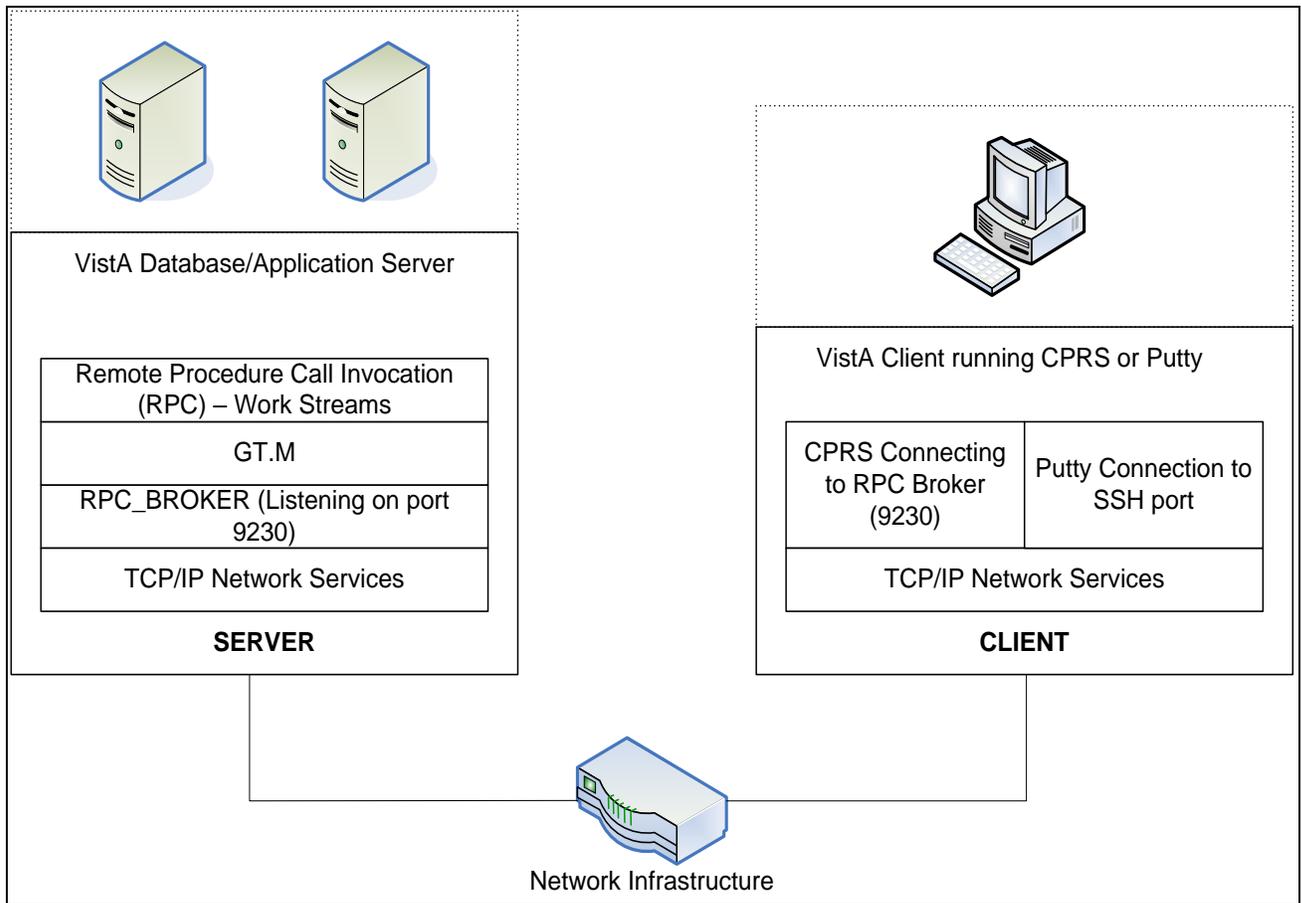
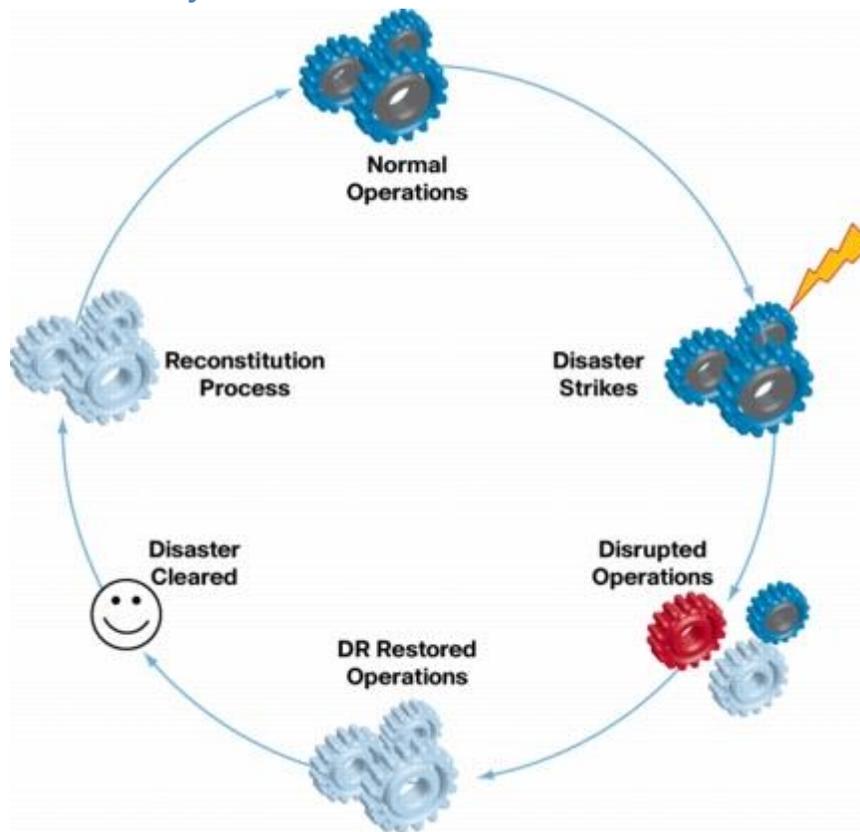


Fig.8. Network diagram for VistA application access from Hospitals

12.6 Disaster Recovery Plan



Disaster recovery¹⁸ is the process, policies and procedures related to preparing for recovery or continuation of technology infrastructure critical to an organization after a natural or human-induced disaster. Disaster recovery is a subset of business continuity. While business continuity involves planning for keeping all aspects of a business functioning in the midst of disruptive events, disaster recovery focuses on the IT or technology systems that support business functions.

As IT systems have become increasingly critical to the smooth operation of a company, and arguably the economy as a whole, the importance of ensuring the continued operation of those systems, or the rapid recovery of the systems, has increased.

It is estimated that most large companies spend between 2% and 4% of their IT budget on disaster recovery planning, with the aim of avoiding larger losses in the event that the business cannot continue to function due to loss of IT infrastructure and data. Of companies

that had a major loss of business data, 43% never reopen, 51% close within two years, and only 6% will survive long-term.

As a result, preparation for continuation or recovery of systems needs to be taken very seriously. This involves a significant investment of time and money with the aim of ensuring minimal losses in the event of a disruptive event.

Developing a technical disaster recovery strategy is just one step in the overall IT Disaster Recovery Planning process. This process is common to all IT systems and utilizes the following seven steps:



Fig.9. Disaster Recovery Steps

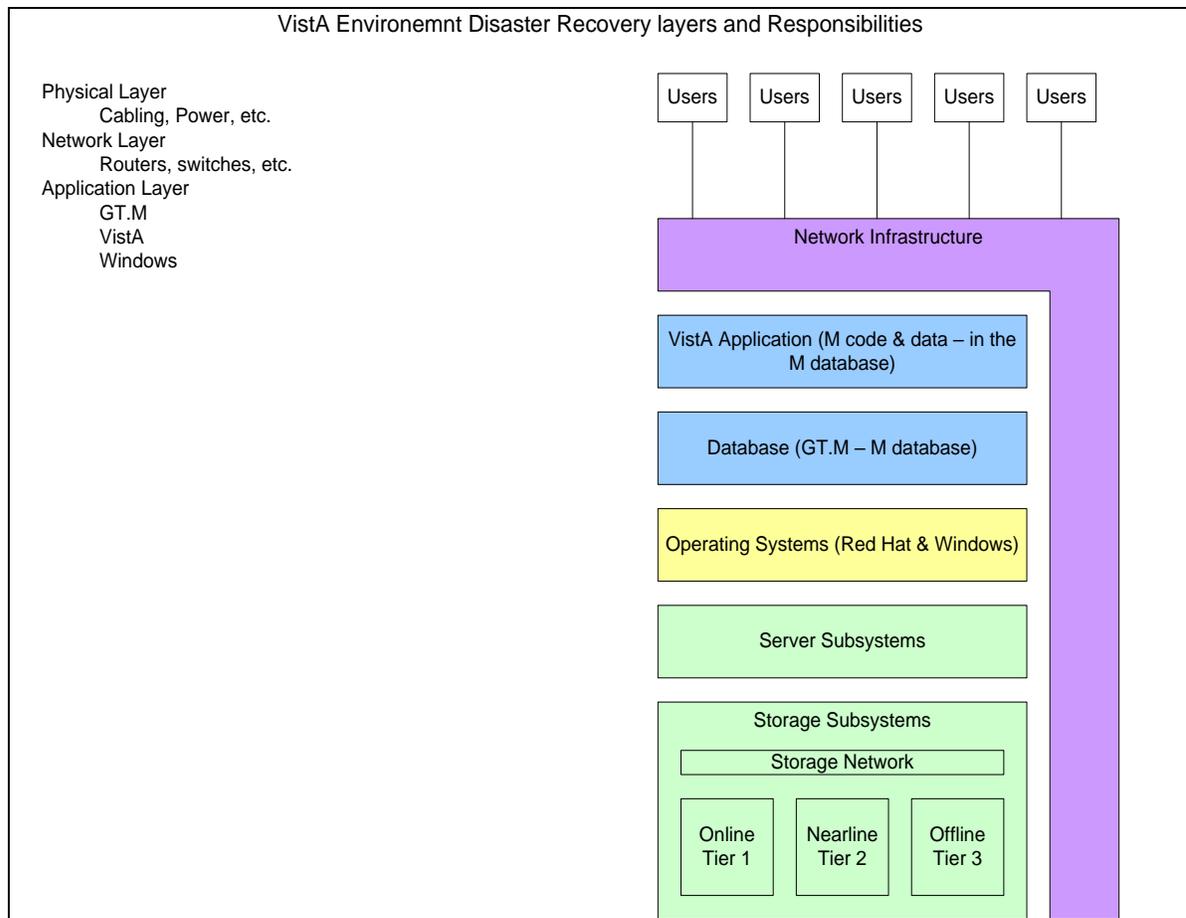


Fig.10

The Disaster Recovery plan of VistA environments should have following components

Replication of database systems (log shipping) between primary and DR data centers.

- GT.M Multisite configuration to enable online database replication (Data) at multiple sites
- Shadowing of Cache database servers (Code and Data) to enable online replication of VistA Imaging DB
- SAN level replication of Mirth DB (SQL server DB) from production site to DR site
- SAN level replication of VistA routine folders

Failover clustering of application/DB servers

- VistA core servers on Linux cluster to provide active passive failover.
- VistA Imaging servers on Windows cluster to provide active passive failover and load balancing.

- HIS Integration servers on Windows cluster to provide active failover . Load balancing implemented by windows network load balancing service (NLBS). Hardware load balancer could be implemented to distribute load balance across multiple integration server cluster nodes.

Redundant network LANs and critical network devices (routers, switches etc)

- Redundant core and distribution switches and routers.
- Redundant network VLAN and automatic failover from failed network to active network
- Redundant SAN switches

Redundant MPLS links

Redundant MPLS link availability from two independent ISP vendors at each Hospital Site.

Redundant servers

Multiple APP/DB servers for VistA Core, Imaging and HIS Integration servers

Storage replication

- Online storage replication - Backup on Tape and optical media.
- Nearline storage replication - storage on archival medium
- Offsite (Bunker storage) – storage in offsite bunker facility

Business Continuity of VistA EHR applications

The availability of VistA production environment can be ensured by the following ways:

- a. GT.M level Dual site replication
- b. SAN level replication

GT.M level Dual site replication¹⁵

A configuration that uses GT.M replication between one primary and one secondary is termed a dual-site configuration. Likewise, replication between one primary and one secondary is termed dual-site replication. A configuration having a primary and secondary in proximity for operational efficiency, however, would not provide protection against a disruption that affects both systems. A separate and distant "disaster recovery" (DR) third system can provide the operational convenience of proximal systems for routine operations, and a distant system for continuity of business in the face of catastrophic events.

The following steps characterize database updates. The first two steps occur with or without replication:

1. The journal file is written.
2. The database is updated.
3. The logical (M-level) journal file entry is delivered to a replication Source Server which in turn delivers it to the secondary system.

Once the first step completes, the transaction is recoverable even if the primary system crashes.

The database is replicated via M-level journal records. The journal records are replicated as units related to a database transaction. Replication to the secondary system is asynchronous with the transaction on the primary system.

SAN level replication¹⁹

A **storage area network (SAN)** is an architecture to attach remote computer data storage devices (such as disk arrays, tape libraries, and optical jukeboxes) to servers so the devices appear as locally attached to the operating system. Like many storage management applications, data replication functionality is finding its way into the network. SAN-based replication solutions bridge the gap between low-end server-based, and high-end array-based approaches, and support a variety of storage services, including local and remote synchronous mirroring, asynchronous mirroring over long distances, point-in-time snapshot replication and rollback capabilities, providing a variety of local and remote data replication alternatives.

SAN-based replication solutions support heterogeneous storage devices and heterogeneous server platforms, giving the widest possible range of configurations. By enabling any-to-any connectivity, SAN-based solutions deliver the flexibility storage managers need to implement a tiered-storage model. Support for heterogeneous connectivity effectively decouples the performance and availability features of high-end storage arrays, and enables business applications to be mapped to the most cost-effective storage solution, regardless of vendor.

SAN storage replication over WAN (Dual Node cluster at DR site)

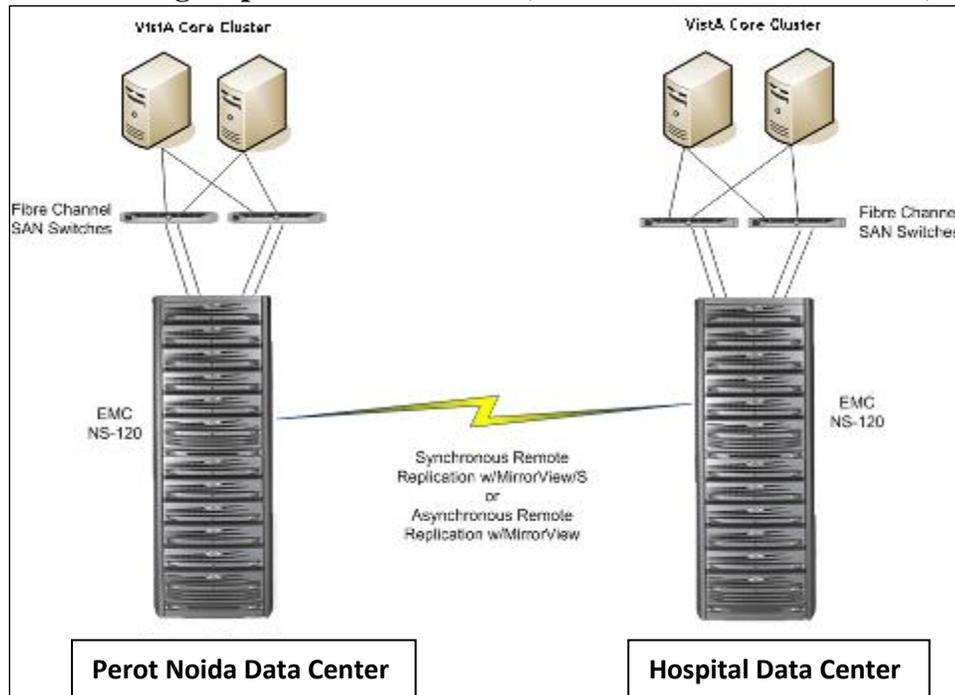
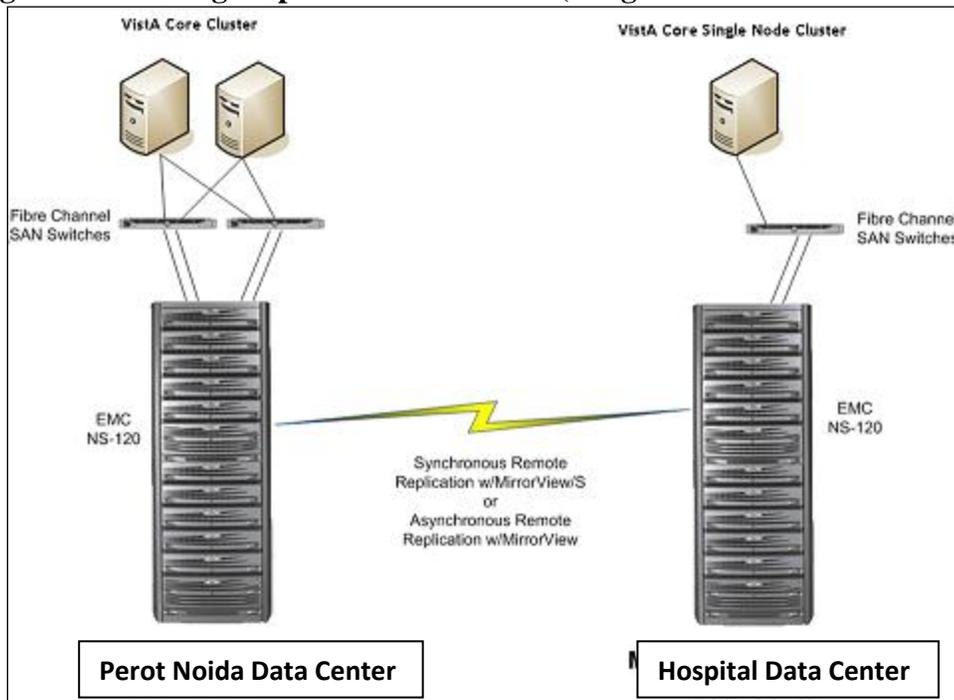


Fig.11. SAN storage replication over WAN (Single Node cluster at DR site)



12.7 Failover In Disaster Recovery

In computing, **failover**²⁰ is the capability to switch over automatically to a redundant or standby computer server, system, or network upon the failure or abnormal termination of the previously active application, server, system, or network. Failover happens without human intervention and generally without warning, unlike switchover. Systems designers usually provide failover capability in servers, systems or networks requiring continuous availability and a high degree of reliability.

At server-level, failover automation takes place using a "heartbeat" cable that connects two servers. As long as a regular "pulse" or "heartbeat" continues between the main server and the second server, the second server will not initiate its systems. There may also be a third "spare parts" server that has running spare components for "hot" switching to prevent down time. The second server will immediately take over the work of the first as soon as it detects an alteration in the "heartbeat" of the first machine. Some systems have the ability to page or send a message to a pre-assigned technician or center.

Some systems, intentionally, do not failover entirely automatically, but require human intervention. This "automated with manual approval" configuration runs automatically once a human has approved the failover.

Failover Clusters

High-availability clusters²⁰ (also known as **HA Clusters** or **Failover Clusters**) are computer clusters that are implemented primarily for the purpose of providing high availability of services which the cluster provides. They operate by having redundant computers or **nodes** which are then used to provide service when system components fail. Normally, if a server with a particular application crashes, the application will be unavailable until someone fixes the crashed server. HA clustering remedies this situation by detecting hardware/software faults, and immediately restarting the application on another system without requiring administrative intervention, a process known as Failover. As part of this process, clustering software may configure the node before starting the application on it. For example, appropriate filesystems may need to be imported and mounted, network hardware may have to be configured, and some supporting applications may need to be running as well.

HA clusters are often used for critical databases, file sharing on a network, business applications, and customer services such as electronic commerce websites. HA cluster implementations attempt to build redundancy into a cluster to eliminate single points of failure, including multiple network connections and data storage which is multiply connected via Storage area networks.

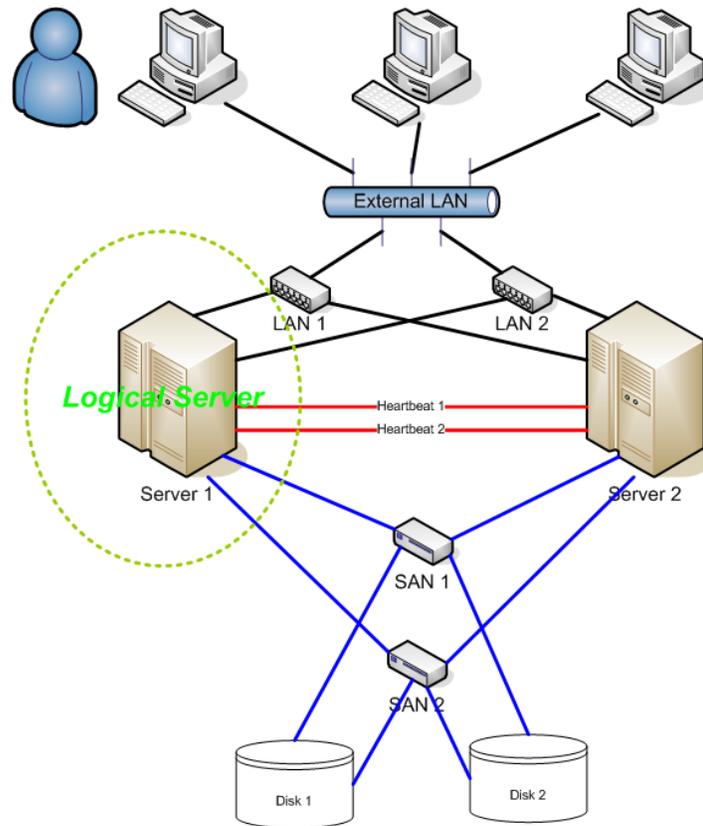
HA clusters usually use a **heartbeat** private network connection which is used to monitor the health and status of each node in the cluster. One subtle, but serious condition all clustering software must be able to handle is split-brain. Split-brain occurs when all of the private links go down simultaneously, but the cluster nodes are still running. If that happens, each node in the cluster may mistakenly decide that every other node has gone down and attempt to start services that other nodes are still running. Having duplicate instances of services may cause data corruption on the shared storage.

Server Cluster

- Active node. The cluster server that currently owns cluster group resources and responds to network requests made to those services.
- Alternative passive node. The cluster server that does not currently own cluster group resources but is available if the active fails over and the primary passive node is unavailable.
- Primary passive node. The cluster server that does not currently own cluster group resources but is available if the active node fails over.
- Virtual server. A collection of services that appear to clients as a physical Windows-based server but are not associated with a specific server. All virtual servers must include a Network Name resource and an IP Address resource.

Server clusters can take two forms: **active/passive clusters** and **active/active clusters**. In active/passive clustering, the cluster includes active nodes and passive nodes. The passive nodes are only used if an active node fails. In active/active clusters, all nodes are active. In the event of a failover, the remaining active node takes on the additional processing operations, which causes a reduction in the overall performance of the cluster. Active/passive cluster configurations are generally recommended over active/active configurations because they often increase performance, availability, and scalability.

Fig.12. Failover Clustering Arrangement for High Availability of Vista Core servers



- Cluster Node 1 will be ACTIVE and Node 2 will be PASSIVE. GT.M and its database files will be shared on SAN storage location so that when primary server node fails, application could switch over to secondary server node and minimal latency would be involved in application switch over as GT.M application would be running on shared node and only latency would be the time taken by clustering node to move from active to passive node.
- There need to be 2 NICs Bonded together with Primary and Secondary IPs setup on each NIC.
- There will be a Cluster IP setup. The application will connect to Clustered IP which will direct the requests to ACTIVE NODE. Heartbeat monitoring will be setup between the 2 nodes and failover and Floating IP failover will be configured. In case Heartbeat monitors a node failure the application automatically switches to passive node becoming active.

Unplanned Switchover from Active to Passive Node

- Both Active and Passive nodes sharing the same database files. GT.M will be installed at SAN with all database folders appear in active and passive nodes as shared drive.
- Heartbeat monitoring between active and passive nodes
- In case of failover, Linux clustering service will move active IP from failed active node to passive node and Passive node will become active now.

Planned Switchover from Active to Passive Node

- VistA database server will be shutdown from the active server node.
- Switch the active IP from active to passive node
- Start the database server from passive node (which will become active now)

12.8 Key Design Assumptions

1. VistA¹ will send/receive any HL7 message to/from IE only and handle acknowledgements generated between IE and VistA. It will be the responsibility of IE to route the message to the recipient application and handle acknowledgments generated between IE and receiving application.
2. VistA does not support compensating transactions and any transaction committed in VistA cannot be rolled back if any supporting business transaction in integrated applications fails without any programming changes.
3. All message transformations will be handled at IE level only
4. No design decision in HIS will affect VistA deployment architecture.
5. GT.M based MUMPS does not support active- active node clustering and Active Passive node clustering will be implemented to provide server redundancy for ensuring Availability of application.
6. GT.M support vertical scalability only and any growth in future business load will be met by increasing more hardware capacity in existing servers.

7. GT.M is a high performance DB which can handle high concurrent load and is used in banking applications supporting ten thousands concurrent users with high performance however there is no VistA site running under GT.M at as high number of users as planned for the Hospital. The performance behaviour of GT.M will be monitored in production environment for some time after go-live and any performance constraints if observed shall be finetuned with active support from Fidelity Inc (GT.M vendor).
8. Current server sizing is done with an upward scalability limit of up to 2500 concurrent users. However it is assumed that these 2500 users will not be logging into system at same point of time and there shall not be more than 1000 concurrent users accessing the application at the same time at any given point of time.
9. Hospital will provide network bandwidth upgradation in existing infrastructure to support high concurrent user access of VistA, HIS and Imaging applications.
10. DR site will have exactly the same environment as Production environment in terms of hardware, software, monitoring, security etc

12.9 Design and Implementation constraints

1. The current solution is not portable on windows and can run only under Linux environment as GT.M can run under Linux environments only.
2. VistA on GT.M support only asynchronous mode of message communication with any application integrated with VistA. However a near synchronous message communication can be achieved by reducing the queue length of messages . This can be done by setting up muti listener jobs to receive messages in VistA, setting up different ports for each message type, setting up different message channel bodies for individual sites, using hardware load balancer to distribute message load over IE, setting up taskman jobs to purge successful message enteries from the message queues and using application level acknowledgement rather than commit level acknowledgement.
3. GT.M support only active passive failover.
4. A very strong network bandwidth is required to transfer images online from PACS server to central Imaging SAN¹⁹.

5. GT.M does not support ODBC , web services and XML.
6. GT.M based server solution support only vertical scalability and does not support horizontally scalable distributed deployment of servers.
7. If VistA Imaging solution is used, Image will be viewed in CPRS and Diagnostic/Display workstations only when image is stored in VI short term or long term storage.

If VistA Imaging is used, proprietary hardware /software components will be required for VI installation.

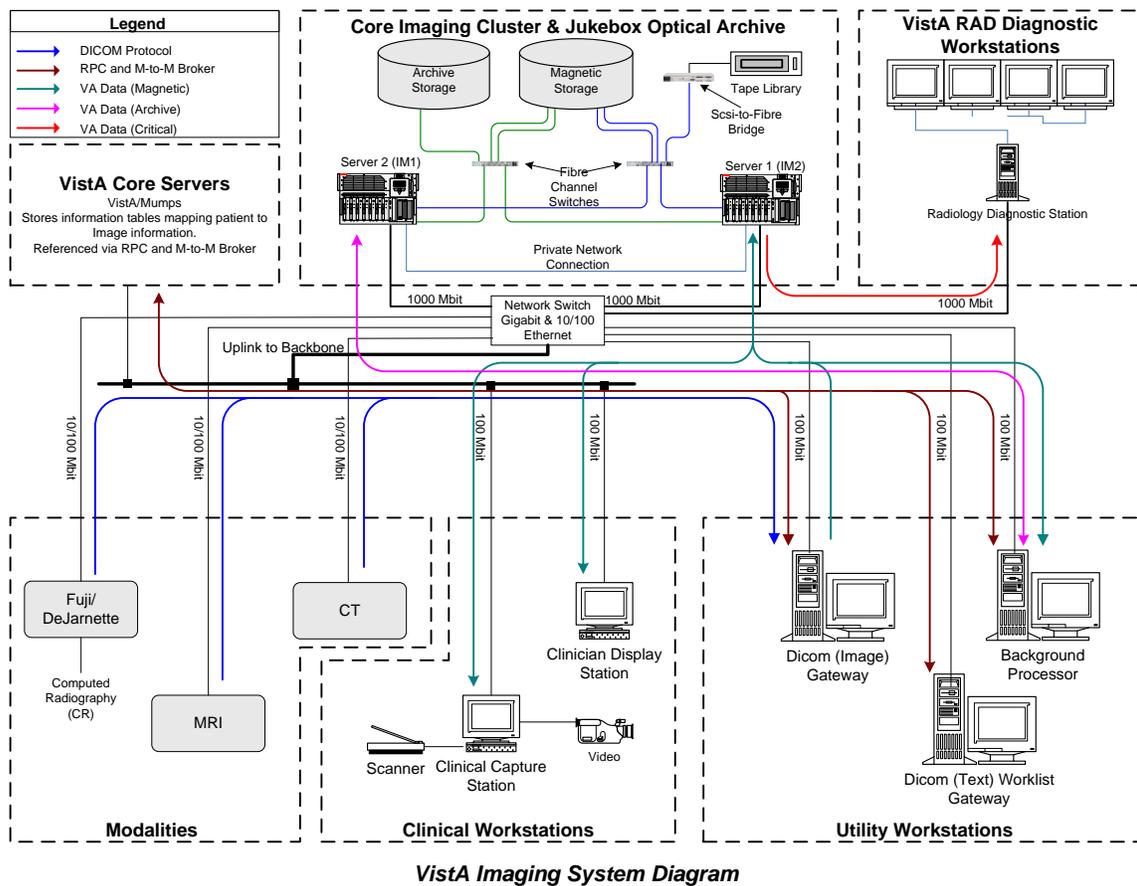


Fig.13

VistA Imaging Network Topology

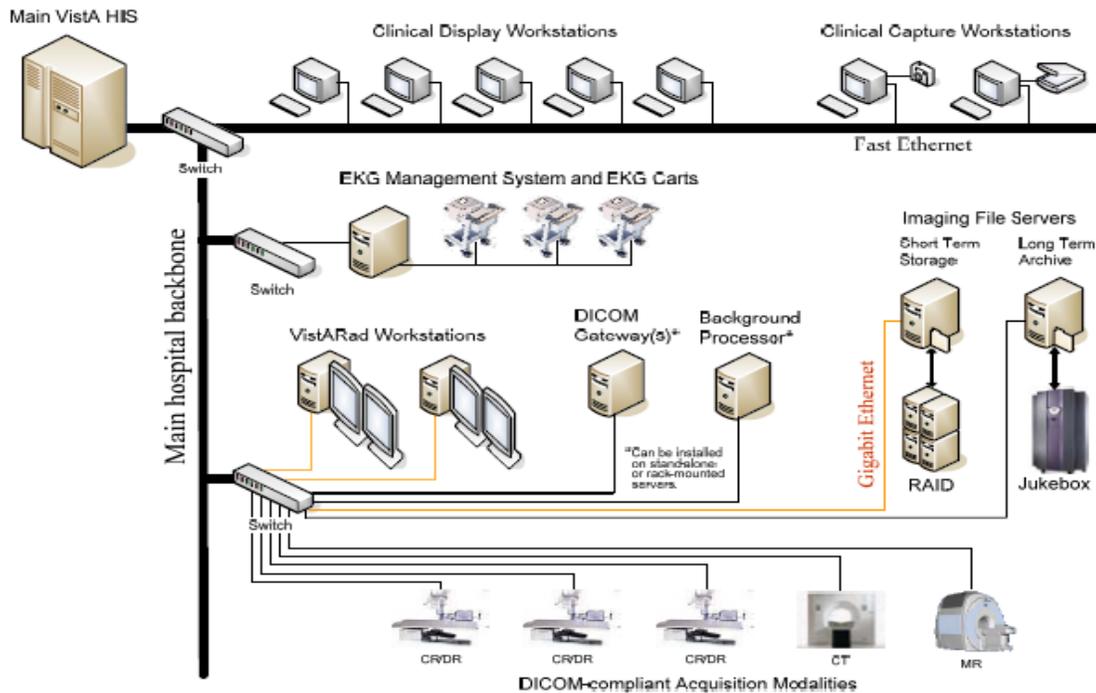


Fig.14

12.10 High Availability

High availability and Resilience for VistA applications will be achieved by the following:

12.10.1 Infrastructure Availability

- Redundant servers for VistA Core, Imaging and HIS Integration servers
- Redundant SAN switches
- Redundant power supplies
- Multiple Processors (SNMP)
- Segmented memory
- Redundant disks
- Redundant Network critical devices (switches and routers)
- Redundant NIC teaming cards on all critical servers.
- Server availability monitoring (Advanced and medium level monitoring of all VistA servers, Heartbeats maintained between active and passive nodes for automatic failover, PING for Imaging and HIS servers)
- Critical Device failure monitoring

12.10.2 Network Availability

- Redundant MPLS link for Data center , DR site and all Hospitals Sites
- Redundant Core Routers and Switches
- Redundant links from Core to Distribution switches
- Redundant multilayered switches
- Link aggregation (trunking) by distributing network traffic distribution over multiple physical links.
- Network availability monitoring

12.10.3 Application Availability

- Dual site GT.M Replication for VistA servers between Primary and DR sites
- Shadowing of VistA Imaging database servers (cache based) to provide online mirroring of Imaging data globals.
- SAN replication of HIS Integration database (message repository) between primary datacenter and DR site.
- Clustered deployment of servers to provide failover.

12.10.4 Storage Availability

- RAID (1+0 or 10) configuration for VistA EHR application data storage provide high performance and disk mirroring storage of data.
- RAID 5 for Imaging and HIS Integration storage provide Low cost redundancy solution which uses block-level striping with parity data distributed across all member disks.

12.10.5 Messaging availability

- Clustered deployment of Integration servers
- Messages never lost as will always persisted in Mirth SQL server based message DB and ability to resend failed messages. Notifications generated in case of message errors and error logged with messaging team.

- Message storage on SAN (RAID 5) to provide storage redundancy.
- SAN level replication to provide message data replication across primary and DR data centers.

12.11 Recoverability

- Linux scripts for failover of VistA database from failed node to passive node
- Redundant SAN disks for data storage. VistA data stored in RAID 10 configuration will stripe data into multiple disks simultaneously ensuring data availability in case of disk failure.
- Automatic OPAS ticket generation for critical device and network failures with sev 1 tickets generated for any major failure.
- Regular Backup (online, nearline and offline) to protect data from failure and data restoration practices in place to recover data in case of any database integrity failure.
- GT.M Dual site replication enable online replication of database sites at secondary sites from primary site.

12.12 Performance

GT.M is a high performance database which is used in banking and healthcare applications supporting high number of concurrent users.

Linux is high performance operating system than windows

- Linux is modular and thus less resource hungry OS and bare minimum kernel installation is required. Other Linux modules can be installed on need basis. Windows installation requires all services to be deployed.
- Most hard drive installations of Linux utilize a "swap partition", a partition dedicated exclusively for paging operations. This reduces slowdown due to disk fragmentation from general use. Linux also allows adjusting aggressiveness of the kernel when deciding whether to swap out an application or leave it on RAM. Windows does not support such features.

- Windows' file system NTFS works causes files to become fragmented, degrading the performance of the system significantly over time, and it requires regular defragmenting to combat this whereas Linux filesystems don't require de fragmentation.

VistA core servers on Linux operating system will give comparatively high performance than windows OS.

For high performance, it is recommended to have Integration servers deployed using Linux operating system instead of windows. Mirth runs well on Linux OS.

12.13 Security

Security framework for VistA EHR applications⁶ has following components:

12.13.1 Application level security

- Authentication**– Process to prove the identity of user accessing the application.
- Each user accessing VistA will have an entry in NEW PERSON (200) file in database. The VistA application will authenticate the user entry in NEW PERSON file when he attempts login in system.
- Authorization** - process to permit an authenticated user to perform an action
- Users will be authorized by using access and verify codes(access and verify codes stored in encrypted form in database), unique electronic signatures to authorize a transaction e.g. signing orders , controlled access to menus and options based on security keys, CPRS tab access based on user profile access, setting up access rules on clinical documents based on user class.
- Sensitive Record Access** – Critical Patient records can be made sensitive and any user attempting to access these records will get recorded as audit log.
- Confidential data access** – Confidential documents should be titled as a particular document class and business rules can be setup on these document titles to allow

access to particular user class only and not to all users. Clinical documents (e.g. clinical templates) can be retracted and be available only to MRD user class.

- vii. **Application audit logs** - Audit logs can be enabled and generated for any activity at user level or option level System generated audit reports for failed attempts at logon or data dictionary modifications since a time etc
- viii. **Application access security** - An application is defined in a Kernel Option file and a permission to use the application will be given to each user individually. Restricted use of application by users will be allowed by security keys.
- ix. **File fields and template security** – Read, write, delete and laygo access permission can be granted to users on file, fields and template level.

12.13.2. OS level security – users and groups will be setup at Linux OS level to provide controlled access to Linux filesystems. Read, write and execute permissions will be set at user and group levels to control access.

There are four type of users that will be created for VistA application access

- i. Instance Manager User and Group (Super DBA). An Instance user is the Manager of the Instance and has access to all databases that reside in that instance.
- ii. Database Manager User and Group (DBA). A Database user is the owner/manager of just that database and has read access to the “common” area under the Instance.
- iii. Database SSH User. This is the user account used by anyone accessing the database via the terminal (SSH), like Putty. For example, in setting up of private/public key pairs for accessing VistA (going directly to Access/Verify code) this user would be used for those keys.
- iv. Database RPC User. This is a user account used by anyone accessing the database via an RPC Broker connection created via Xinetd (CPRS for example).

Linux root users and system users shall not be given read, write and execute access on database files.

12.13.3. Network security

- i. Network IDS is currently being implemented for network security.
- ii. End to end VPN tunneling can be set for message transfer between VistA and HIS integration servers or images transferring from local storage to central storage.
- iii. Firewall controlled access for all server traffic, all internet traffic content filtering done by Cisco Proxy firewalls.
- iv. Persistent IPtable rules should be defined for Linux users.
- v. Network security monitoring by Security operations team.

12.13.4. Infrastructure security

All critical servers will be placed in server DMZ zone in Dell Noida data center. The traffic to servers will be controlled by using firewalls. All critical network devices will be managed devices and security operations team will continuously monitor the availability and security of all critical devices.

12.13.5. Data security

- i. Passwords will be stored in encrypted format in database.
- ii. Message transmission between VistA and HIS integration servers over a secure transport layer. Network transport security protocols e.g. IPSec, end to end VPN tunneling for intra server data transmission.
- iii. Internet port on servers to be scanned continuously for any malicious data.
- iv. Encryption of GT.M database and journal files using public and private keys. This will prevent database access by any unauthorized users. However this is a tradeoff between integration performance and data security.
- v. User activity to be logged on servers using SUDO logs in Linux systems.

- vi. No database SSH user and RPC user have access to folders containing database and journal files.
- vii. All inbound HL7 messages into VistA will contain the access code of integration user in hospital segment which will be authenticated in VistA before message is accepted.
- viii. Secure WAN links to transfer image data from PACS storage to central backup storage.
- ix. All servers will be placed in secure DMZ inside Dell Noida Data center.
- x. All outward traffic to servers will be scanned through firewalls . Persistent IPtable rules in firewalls to prevent DoS attacks.
- xi. All servers having both inward and outward connectivity (e.g. mail servers) will be placed in another DMZ and the outward traffic will be scanned through Cisco internet firewalls for content filtering and then filtered through server firewalls before directed to servers.

12.13.6. Communication security

- i. All RPC broker calls will be registered in broker file and VistA kernel will authenticate all RPC broker calls initiated from client connections.
- ii. The port range for HL7 message transmission between VistA servers and HIS IE engine will be secured and scanned continuously for any malicious attack.
- iii. All switches and routers will be managed devices and monitored continuously by network monitoring team for any network flooding.
- iv. Port to access RPC broker applications will be secured.

12.13.7. Auditing

- i. Auditing shall be enabled for all user login activities, failed login attempts , device failures.
- ii. Auditing can be enabled/disabled at field level.
- iii. All sensitive patient record access will be audited.
- iv. Auditing can be enabled for access to confidential data. All Integration error logs will be maintained in error queues with proper error code and error reason.
- v. All database SSH user activities will be logged at Linux level using SUDO logs.
- vi. System security audit logs will be generated regularly by application monitoring team.
- vii. Linux scripts will be developed to provide GT.M database monitoring and system status reports.

12.13.8. Messaging Security

- i. Host level security will be implemented by securing port range and defining persistent IPTable rules.
- ii. Message transport layer on a separate VLAN than user VLAN inside data center.
- iii. Integration user credential passed in hospital segment and any inbound message received in VistA will be authenticated (this integration user should be a valid VistA user defined in new person file.)

12.13.9. Data center security

- i. All VistA servers and HIS integration servers will be placed in DMZ zone along with existing HIS servers in data center.

- ii. All server traffic filtered through proxy firewall and server firewalls before directed to server zone.
- iii. Managed physical access to data center.
- iv. Servers on separate VLAN than user VLAN.
- v. No internet port access on servers.
- vi. DR sites should have same security control as for primary data center.
- vii. All core routers and switches placed in data center continuously monitored for network flooding or Dos Attacks.

12.13.10. Anti virus security

- i. AV patches will be installed on all user workstations
- ii. Linux is virus free OS.
- iii. Security vulnerability patches will be regularly updated on Windows and Linux servers
- iv. AV patches regularly updated on windows OS.

12.14 Stability

1. Regular full and incremental backups (cron tab jobs will be scheduled running linux scripts to take incremental and full backups of VistA database at specified time intervals). Imaging data regularly backed up from centralized imaging RAID to backup media (jukebox) by using online backup agents e.g. Netbackup. Messages persisted in Integration server DB will be online replicated to DR site and there is no need to maintain online backups for messages persisted in Integration database.
2. DR site should have exact replica for primary data center in terms of server hardware capacity , network bandwidth availability , security controls etc

3. GT.M dual site configurations to enable near line replication of data from primary site to DR site. This will ensure business continuity in case of primary site failure.
4. Redundancy planned for all critical hardware components (servers ,network devices , storage , NIC cards etc)
5. Server deployment in Cluster configuration to ensure fault tolerance.
6. High class server hardware to ensure continuous computing by automatic fault detection, reporting and fault recovery.
7. Monitoring and logging system to take proactive actions to detect and recover from any fault.
8. Run time exception handling in VistA code.
9. Error handling for failed HL7 messages and ensuring failed messages resend after taking corrective actions.
10. Defined process for patch management in production environment.

12.15 Maintainability

1. Loose coupling between integrating partner applications – all integration through IE (as enterprise service bus)
2. Well defined patch management process of localized code changes in VistA applications.(creating local kids patch files of code changes)
3. Defined configuration management process for all changes
4. Error log and system trace logs
5. Code changes following VistA SAC recommendations
6. Single DB architecture makes maintainability a simpler task.
7. Well defined upgrade policy for application and OS patches.

8. Simplifying GT.M DBA operations by developing automated Linux scripts.
9. Contant upgradation of security and antivirus patches on all servers and workstations
10. OS level patch upgradation (n-1 approach for OS patches)
11. Security vulnerability patch upgradation on Linux and windows servers by Patching team.
12. Automatic application upgrade patches by using SCCM tools.

12.16 Scalability

1. Scalability of present architecture can be increased.
2. GT.M support vertical scalability (by adding more processors, increasing RAM, network bandwidth etc.)
3. By using Multisite GT.M replication (the replicated Databases could be deployed locally at each site and could be used for reports or remote patient access using PDX module in VistA)
4. Servers should have extra slots for additional hardware capacity to be upgraded in existing servers.
5. SAN storage should have extra slots to add more storage in future.
6. All core network devices (routers and switches) should support 10 GigE network bandwidth.
7. Network distribution switches should be upgraded from 100baseT to 1000BaseT to handle extra load due to VistA Imaging.
8. LAN cards at user workstations should be upgraded to support 100 mbps network load
9. Existing network topology should be upgraded to cat 6 cabling.
10. Addition of bandwidth capacity in existing MPLS cloud to support more bandwidth requirements.

11. Integration of all external applications through IE only.
12. GT.M 64 bit version has no limit on database file (only constrained by available disk space)
13. GT.M database use optimistic locking (M locks act as traffic lights instead of road blocks) that increase application concurrency.
14. Since GT.M uses the normal system semaphores for database startup or shutdown (but not for normal database operation), this puts an unintended load on the system semaphores. In the real world, hundreds of users will not be moving in lock-step – even when you have shift changes, the users may logout and login over a span of ten seconds to minutes rather than in lock-step.

12.17 Deployment and environment

Environments

1. **Development environment-** Development environment shall be used by Technical team to develop, modify and Unit test the VistA components. Technical team shall own the environment.
2. **Configuration environment-** Configuration environment shall be primarily used by Configuration team to build configuration content for Hospital. Configuration team shall own the environment.
3. **Integration environment-** Integration environment shall be primarily used by Integration team. Mirth will be installed in this environment. Integration team shall own the environment.
4. **Testing environments-** There shall be three test environments. Testing team shall own all the testing environments. The first environment shall be maintained for doing VistA application testing in isolated mode. The second environment shall be used for doing end to end System Testing. The purpose of last environment is to maintain defect tracking tool.

5. **Staging environment-** Staging environment shall be near replica of Production environment. This environment shall be used for UAT Testing and testing of patches in pre production environment before deploying these in production. Snapshots from production shall be regularly applied on production. Technical team shall own the environment.
6. **Training environment-** Training environment will be used by training team to impart VistA training to the VistA users. Training team shall own the environment.
7. **Performance testing environment–** This environment should be exact replica of production environment in terms of server capacity and application setup. Snapshots from production shall be applied on this environment before conducting stress and performance load testing.

12.18 Patient ID configuration in VistA

WorldVistA VOE has Social Security Number as Mandatory field for Patient Identifier which needs to be changed for Hospital Requirements.

Code change will be implemented in VistA code to implement following changes:

- Making SSN non mandatory and populating Pseudo SSN at time of registration.
- Using MRN(a.k.a HRN) field in VistA to store Patient Identifier.
- Populating cross references (VistA indexes) to use MRN as primary identifier for all Patient lookups. (SSN based cross reference will not be deleted)
- MRN to be configured to store long patient ID (up to 13 length)and will be alphanumeric.
- CPRS reports to display MRN field (Patient Identifier) instead of SSN.
- Patient will be searched based on MRN field in CPRS and SSH (Roll and Scroll) applications.
- Changes in GUI clients to display Patient Identifier.

12.19 Data standardization

VistA has ICD9, CPT codes but HIS does not. To standardize data across VistA, HIS and other applications, following approach could be considered:

1. Adding a new column in existing HIS master tables and loading the VistA coding for the master items.
2. Data mapping between VistA and HIS through HL7 messages.
3. Data transformation rules will be defined in Mirth transformer components . The data mapping information will be maintained in external mapping tables maintained in SQL database in Integration server.
4. Data mapping tables maintained in VistA and HIS – Mapping done for all inbound and outbound messages passing between VistA and HIS.

12.20 VistA Hardware Requirements

The following Hardware Requirements are based on the Site Assessment Data(Number of Nursing Stations, ICUs, Patient Bays, Doctor’s Chambers etc. and the Hardware required at each place) from all six Hospital Sites where VistA- EHR is to be implemented.

12.20.1 Workstations (Quantity)

Hospital Sites	No. of Workstations required for VistA
SITE 1	396
SITE 2	54
SITE 3	49
SITE 4	162
SITE 5	77
SITE 6	60
Total	798

12.20.2 Laptops (Quantity)

Hospital Sites	No of Laptops to be procured
SITE 3	5
SITE 6	10
SITE 5	12
SITE 2	10
SITE 1	30
SITE 4	15
Total	82

12.20.3 Printers

Hospital Sites	Vista (Laser) Printers
SITE 1	173
SITE 2	35
SITE 3	30
SITE 4	68
SITE 5	25
SITE 6	31
Total	362

12.20.4 Barcode scanners (Normal)

Hospital Sites	Barcode Scanners
SITE 1	94
SITE 2	5
SITE 3	11
SITE 4	33
SITE 5	17
SITE 6	15
Total	175

12.20.5 Barcode Readers

Hospital Sites	Barcode Readers
SITE 1	102
SITE 2	32
SITE 3	30
SITE 4	47
SITE 5	29
SITE 6	20
Total	260

12.20.6 COW (Computer On Wheels)

Hospital Sites	COW
SITE 1	83
SITE 2	4
SITE 3	13
SITE 4	34
SITE 5	19
SITE 6	15
Total	168

12.20.7 Tablet PC

Hospital Sites	Tablet PC
SITE 1	177
SITE 2	0
SITE 3	9
SITE 4	86
SITE 5	29
SITE 6	28
Total	329

13.0 RESULTS AND FINDINGS:

1. The technical architecture of VistA is built using a client server architecture.
2. The operating system is Linux on the server and Delphi based GUI applications and SSH applications (roll and scroll) for the client computer.
3. GT.M or Cache hierarchical DB are used for the database.
4. The Application Architecture of VistA- EHR is the Layered Architecture comprising of Data Source Layer, Business Layer and Presentation Layer.
5. All hospital sites are connected with Dell Noida Data center through MPLS cloud and VistA applications will be accessed over MPLS cloud in all Hospitals.
6. The Disaster Recovery Plan of VistA includes replication of Database Systems between Primary and DR Sites that is ensured in following ways: GT.M level Dual site replication and SAN level replication.
7. Security framework for VistA consists of Application Level Security, OS Level Security, Network Security, Infrastructure Security, Data Security etc.
8. Scalability of present architecture can be increased.
9. VistA has ICD9, CPT codes but HIS does not. To standardize data across VistA, HIS and other applications, data standardization approach is being used.

14.0 CONCLUSION

Developing an Architectural Design for EHR Software and implementing the same requires Scale, Process, Cost, Schedule, Skill and Development Teams, Materials and Technologies, Stakeholders and Risks. The software architecture is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. The Technical Architecture of VistA is built on the basis of Architectural Requirements that are a subset of the system requirements, determined by architectural relevance. The business objectives for the system, and the architecture in particular ensure that the architecture of VistA is aligned with the business agenda. Thus, Architecture represents the set of earliest design decisions; **Hardest to change and Most critical to get right.**

C. REFERENCES

1. James A. Senn (2000), Analysis and Design of Information Systems (Second Edition), McGraw-Hill International Editions.
2. Joseph Tan with Fay Cobb Payton (2002), Adaptive Health Management Information Systems, Concepts, Cases, and Practical Applications (Third Edition), Jones and Barlett Publishers.
3. The Veterans Health Administration: Quality, Value, Accountability, and Information as Transforming Strategies for Patient-Centered Care , The American Journal of Managed Care, November, 2004.
4. Goldstein, Ponshe, Maduro (2005), “Profile of Increasing Use of OSS in the Federal Government and Healthcare”
http://www.medicalalliances.com/downloads/files/Open_Source_Software-Government_and_Healthcare_White_Paper-Medical_Alliances_2.doc
5. http://www.perotsystems.com/MediaRoom/Library/IndustryOverviews/IndustryOverview_HealthcareFaceSheet_EU.pdf
6. <http://www.va.gov/vdl>
7. <http://www.hardhats.org>
8. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1761885/>
9. <http://www.ncbi.nlm.nih.gov/pubmed/17877871>
10. www.connectingforhealth.org/.../docs/T2_HealthInfo.pdf
11. <http://www.worldvista.org>
12. <http://www.medsphere.org>
13. <http://www.in.gov/fssa/files/QualCheck.pdf>
14. http://en.wikipedia.org/wiki/Application_Architecture
15. <http://en.wikipedia.org/wiki/GT.M>
16. en.wikipedia.org/wiki/MUMPS
17. en.wikipedia.org/wiki/Multiprotocol_Label_Switching
18. en.wikipedia.org/wiki/Disaster_recovery
19. http://findarticles.com/p/articles/mi_m0BRZ/is_6_24/ai_n6145654/
20. [http://technet.microsoft.com/es-es/library/cc732488\(WS.10\).aspx](http://technet.microsoft.com/es-es/library/cc732488(WS.10).aspx)
21. <http://www1.va.gov/CPRSdemo/>

D. ANEXXURES

1. CHANGE REQUEST PLAN

ORIGINATOR OF REQUEST: _____ DATE: _____

FILE/ REFERENCE: _____

SOURCE OF THE PROBLEM REQUIRING THE CHANGE:

- | | |
|--|---|
| <input type="checkbox"/> SRS Baseline document | <input type="checkbox"/> Web Design / Content |
| <input type="checkbox"/> Design Specification as Implemented | <input type="checkbox"/> Project Plan |
| <input type="checkbox"/> Implementation | <input type="checkbox"/> Test Plans |
| <input type="checkbox"/> Database | <input type="checkbox"/> Manuals |
| <input type="checkbox"/> Other _____ | |

DESCRIPTION OF REQUESTED CHANGE:

CHANGE

REQUIRED:

SPECIFY ITEM:

- | | |
|--|-------|
| <input type="checkbox"/> GUI / Web Pages | _____ |
| <input type="checkbox"/> Software Design | _____ |
| <input type="checkbox"/> Implementation | _____ |
| <input type="checkbox"/> Database | _____ |
| <input type="checkbox"/> Document | _____ |

DESCRIPTION OF CHANGE:

Include identification of the affected software component, document, or database change – or a combination of these. If the software change and/or a database change approved, attach the specification for the change. Most changes will also require a baseline document update. If this is the case, attach the change, identifying the document, section number, and the wording to be changed or added.

IMPACT ANALYSIS:

Describe the work required to implement the change and estimate the effect of the work on the project schedule.

DISPOSITION OF CHANGE REQUEST:

- Approved.
- Not approved. *(Explain)* _____

AUTHORIZATIONS:	SIGNATURES:	DATE:
Sponsor	_____	_____
Project Faculty Adviser	_____	_____
Team – Project Manager	_____	_____

2. DISASTER RECOVERY PLAN:

Type of Data	Minimal Backup Policy	Backup Retention Policy
System software	Latest Version plus patches At Least Weekly	Annual (verified) Backup Monthly Generations Weekly Generations
Application software	Latest Version plus patches At Least Weekly	Annual (verified) Backup Monthly Generations Weekly Generations
System data	Daily	Annual (verified) Backup Monthly Generations Weekly Generations Daily Generations
Application Data	Daily with real time transaction files	Annual (verified) Backup Monthly Generations Weekly Generations Daily Generations
Software licenses, encryption keys, & Protocol Data	Weekly	Annual (verified) Backup Monthly Generations Weekly Generations

S No.	Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Business Goals						
1	There has been “defining” event in the organization that has demanded an EMR.					
2	The EMR initiative is mentioned in the organization’s strategic plan and is linked to achieving specific future organizational goals.					
3	Senior management views EMR as key to meeting future organizational goals.					
4	The CEO and executive body understand EMR and the business benefits it can bring.					
Communication/Perception						
5	All stakeholders potentially affected by an EMR initiative have been identified.					
6	Clinicians have had an opportunity to ask questions regarding the EMR initiative.					
7	Clinicians understand the benefits of an EMR and are enthusiastic about using the new system.					
8	Stakeholders have been/will be included as part of the project team from the start of the project.					
9	All the stakeholders understand their role in making the EMR initiative a success.					
Patient Orientation						
10	A strong patient focus permeates every department in the organization.					
11	Business decisions are driven by patient needs.					
12	Methods for capturing and enhancing patient care have been identified and documented.					
13	All current patient touch-points in the organization have been identified and mapped.					
14	EMR Design will be driven by					

	what is important to patient care and patient satisfaction.					
Workflow and Processes						
15	Current workflow and processes have been identified and documented.					
16	The organization has identified and prioritized areas where EMR could be best applied.					
17	Ways in which EMR will improve current workflow and processes have been identified.					
Technology Evaluation						
18	A list of evaluation criteria was used in the EMR vendor selection process.					
19	The EMR vendor is very familiar with the organization's environment and customer interaction methods.					
20	A clinician-defined user interface was a primary consideration in EMR software selection.					
21	The need of remote access to EMR system has been assessed.					
22	Hardware choices are/will be based on the applications and environment in which the hardware will be used.					
23	An IT infrastructure is either in place or under development that will support the processes of the EMR with minimal downtime during its implementation.					
24	The organization has established service levels that must be met by the EMR system used to deliver patient care.					
Data Management						
25	Top-level executives recognize the importance of integrating					

	databases containing patient information.					
26	It has been decided and documented which data from which systems will need to integrate with the EMR solution.					
27	Data accuracy and integrity procedures have been addressed and rectified.					
Training/Support						
28	A budget is in place to provide end-user training.					
29	Training for all user groups has been scheduled well in advance of the final rollout.					
30	Training includes reference materials that can be used before, during, and training.					
31	A budget is in place to provide reasonable coverage for EMR support services.					
32	Staff is in place to implement, provide support for, and maintain the EMR system.					
Total						

Outcome Interpretation

Enter the totals for each column below:

_____ Strongly Agree

_____ Agree

_____ Neutral

_____ Disagree

_____ Strongly Disagree

A high number of Strongly Agree and Agree selections (30+) mean that you are well positioned to implement an EMR initiative.

If your responses fall mostly into the Agree-Disagree range (25-30), then your organization needs to further develop its current processes, attitude, and strategic plans before pursuing an EMR initiative.

If the majority of your responses include Disagree and Strongly Disagree (25+), implementing an EMR initiative at this time would likely result in failure.

Regardless of your results, take a good look at those statements with which you did not Strongly Agree. These areas are candidates for improvement, and by pursuing this path you will further the chances of success for your EMR solution.