

“© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Smart Hospital Management System: An Integration of Enterprise Level Solutions Utilising Open Group Architecture Framework (TOGAF)

Zenon Chaczko, Christopher chiu and Avtar Singh Kohli
Faculty of Engineering and IT,
University of Technology Sydney
NSW 2007 Australia
zenon.chaczko@uts.edu.au, christopher.chiu@uts.edu.au
and avtar.s.kohli@uts.edu.au

Venkatesh Mahadevan
Faculty of Higher Education,
Swinburne University of Technology
Vic 3140 Australia
vmahadevan@swinburne.edu.au

Abstract—A significant portion of the Hospital Information Systems currently consists of various individual legacy applications that have to be integrated, to deliver a more unified solution. The performance, reliability and other factors of these applications can alter the performance, reliability and other characteristics of integrated Solution, the Smart Hospital Management System (SHS). The actual evaluation of these parameters of these applications is outside the scope of this document. The SHS being an infrastructure component relies heavily on the actual resources made available to it for its proper functioning, operation and maintenance. This article aims to deliver an approach in architecting solutions which can be utilised as framework to address common issues in integration of enterprise level solutions. The methodologies discussed in TOGAF version 9 are utilised to demonstrate the feasibility of proposed solution. This paper introduces the problem space/scenarios, constraints, requirements, enablers, risks, sample legacy application architectures and proposed integration solution presented with TOGAF components. The growing number of waiting lists, rising pressure on medical professionals and accountability for medical negligence are only part of the motivation to take initiative towards holds a core model integration strategy in various legacy infrastructure systems.

Keywords—Smart Hospital System, the Open Group Architecture Framework (TOGAF), Integration Frameworks, Service Oriented Architecture (SOA)

I. INTRODUCTION

The Smart Hospital System is a solution aimed to present architecture Integration Framework using TOGAF's Architecture Development Method. The key to TOGAF is the TOGAF Architecture Development Method (ADM) - a reliable, proven approach for developing enterprise architecture descriptions that meets the needs of the specific business. There are 8 phases to the TOGAF core model which include:

- Architecture Vision
- Business Architecture
- Information Systems Architecture
- Technology Architecture
- Opportunities and Solutions

- Migration Planning
- Implementation Governance
- Architecture change Management

The Architect iterates through these phases and analyse contextual information which aids requirements management, as shown in figure attached.

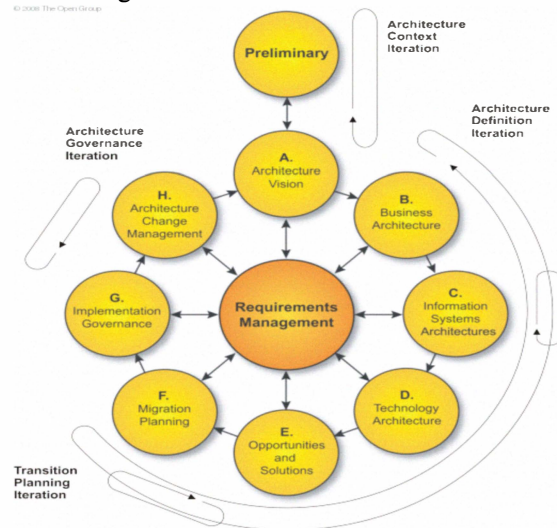


Figure e: Iteration Cycles

Figure 1. Iteration Cycles

A. Phase 1: Architecture Vision:

As detailed in the SHS Project Proposal document, the main business mandates for the SHS are to build a new IT Integration Platform/framework that shall be able to:

- Respond to new business demands of the organisation (Hospitals/Medical facilities) for the future (Scalability/new applications/new devices/ more users).
- Deployed quickly at any new location within restricted time frame, and with minimal configuration and no new development / customisation required.

- Be able to reliably service the current workload for urban hospital serving a population of 1 million i.e. up to 10,000 user accounts, up to 200 concurrent users. Along with the new workloads projected of up to 100,000 user accounts and up to 5,000 concurrent users.
- Provide 99.999% availability (which equates to 5 minutes of downtime in a year).
- Fast and efficient enough to be able to simultaneously service several hospitals and mobile units in geographically diverse areas of the country.
- Have no additional operational costs compared to the current infrastructure.
- Provide a single UI from all the applications to the user.
- Integrate the current Sample application Patient Management System (PMS) and Accounting and Payroll Package (APP) applications with minimal effort.
- Provide flexibility in choice of application providers, to avoid vendor lock-in.

B. Phase II: Business Architecture:

The following Figure 2 is just a scaled down version of the problem space.

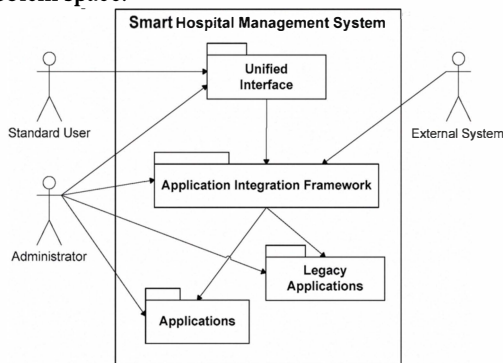


Figure 2. The Boundary Context Model of the SMHS

Administrator: A user who has administrators privileges to the HSIF. Access is provided to all applications and components within the SHS. The administrator has access to administration and maintenance functionalities within SHS.

Standard User: A user with no administrator privileges to the SHS. Restricted access is provided through the Unified Interface only. No HSIF functionalities are surfaced to this user. Users have access based on their user profile to specific functionalities within the existing PIMS and ERP systems (legacy systems); however this access will be mediated via the HSIF.

External System: An external system that can access specified services through Application Integration Framework. The following are the Architectural representations of PIMS and AAP for visual aid:

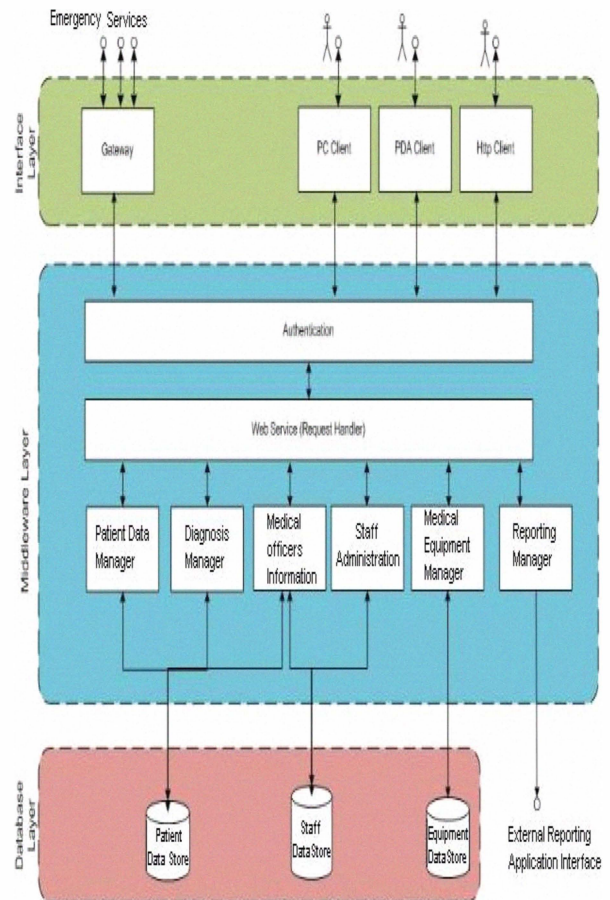


Figure 3. Components of the legacy application APP

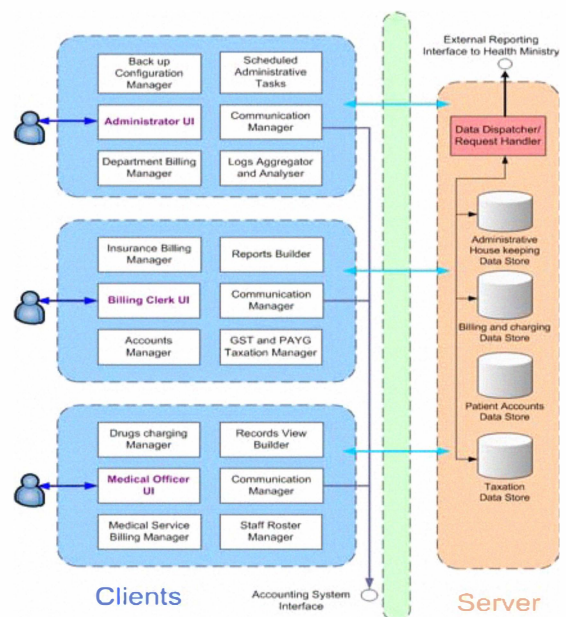


Figure 4. Components of the legacy application PMS

C. Phase III: Information System Architecture:

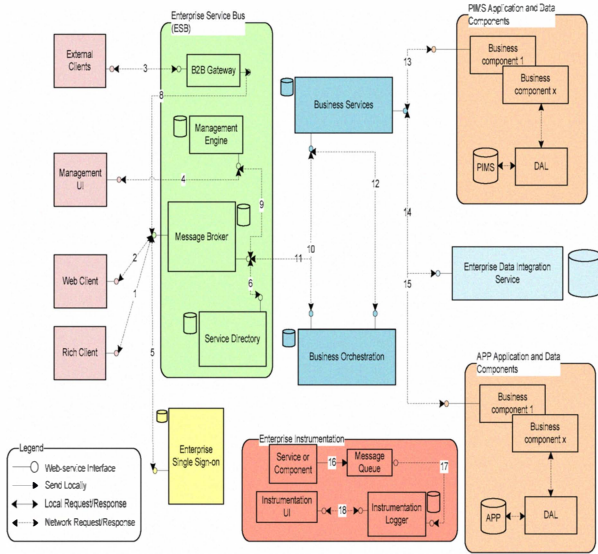


Figure 5. SHS Integration Architecture (Conceptual View)

The *SHS Database* is an aggregation of the *Database Components* of all the *SHS Applications* deployed in the system. This is hosted on the *SHS Database Server*. The *Application Components* of all the *SHS Applications* deployed are hosted on *SHS Application Servers* which have an *SHS Agent* each. Each agent knows the *SHS Application Components* that are available on the server. The *SHS Interface Server* hosts *SHS Website* and the *SHS Web services*. The website is an aggregation of the entire website based UI's of all the deployed *SHS Applications*. The web services of all the deployed *SHS Applications* are similarly aggregated, and both these aggregations are hosted on the *SHS Interface Server*. The *SHS Server* is a centralized server application that interacts with all the *SHS Agents* in the system, and orchestrates the application integration process.

Currently, the two applications are operating separately without sharing any data. As such it is possible that some data is duplicated in the two applications' data stores. This brings up issues of merging of such data. Although this could be a temporarily problem for the life of the legacy applications in their existing structure, this problem needs to be addressed.

The Architecture Team considered two solutions to this problem:

Data Rewrite: This required manually reconciling conflicting data in the two data stores, and modifying the associated business logic to conform to this. This approach is not suggested as it is too intrusive of the legacy applications, and can pose a major risk to the stabilities of these applications.

Data integration using a separate database: This approach envisages a separate data store called Enterprise Data Integration Service which will contain information for data mapping amongst the existing legacy data, as well as the

location of where the data exists (which legacy data store). The Business Service component, will query the Enterprise Data Integration Service upon receiving a client request, in order to find the location of the required information to service the client request. This information is then used to invoke the appropriate legacy application logic to carry out the requested task.

D. Phase IV: Technology Architecture:

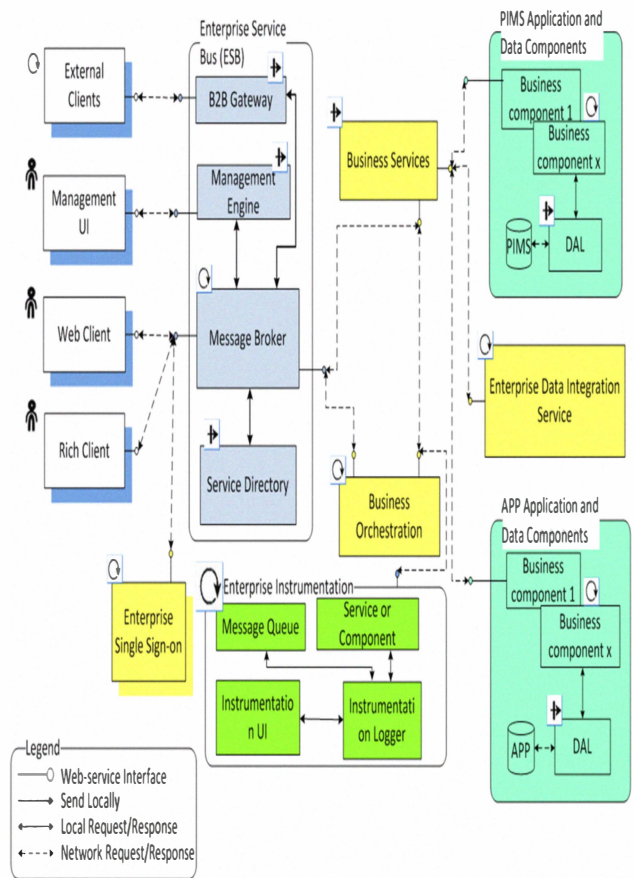


Figure 6. Execution view of SHS

Evaluation of technology Architecture:

Evaluation Criteria: The Evaluation was based on Research on available runtime environments which are of enterprise scale and crossing checking the reviews/pros/cons/stability from various accompanying documentation and blogs by originators.

Evaluation Methods: The idea of choosing IIS 7 and JBoss 5.2 are more related to what suits best in a Enterprise level Application runtime for applications(legacy/proprietary) written in Java/C#/C++. The IIS was obvious choice as the Application "PIMS", is written in .Net which requires a Microsoft Windows runtime environment. For the "APP", JBoss 5.2 was chosen as a deployment server of choice which is a significant upgrade from jetty/apache/tomcat.

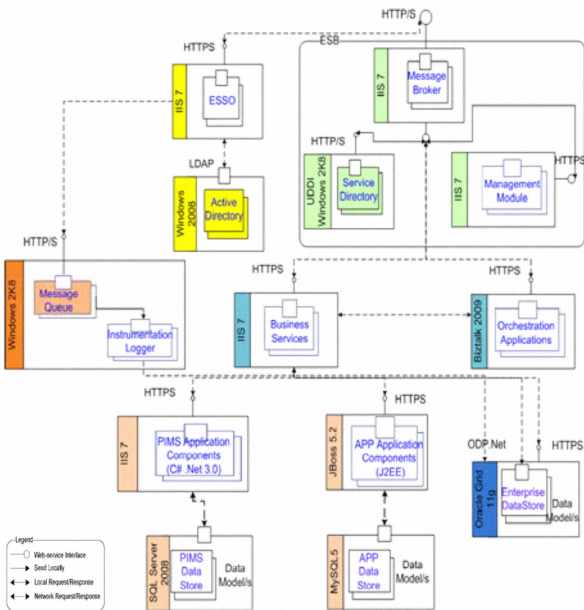


Figure 7. Deployment view of SHS

Benefits: The core benefit of a hybrid approach of running 2 separate runtime Environments and an ESB to allow seamless request/response style integration, include Less Development/modification on API on existing legacy systems and Clear separation of middleware, improves performance (logical Queues) /security (Separate single sign on(SSO) module + HTTPS connections).

Pitfalls: The scalability of the system (adding more independent web applications) would pose major configuration surgery, if implementation does not address Data synchronization issues appropriately. Due to extensive use of web services the Performance/Quality of service may be hampered if the transfer of large files is between systems (format conversion may be an issue within 2 applications .e.g. .odt to .docx formats or large image file formats).

Technological constraints: The execution of APP on JBoss i.e. running on a UNIX/Linux server and IIS running on a Windows server and the integration (ESB) supporting 2 applications deployed on these 2 very different runtime environments, may cause Developers a some grief: like windows slashes(“/”) and Linux slashes(“\”), maintaining the consistency between Date/Data formats. The processing times may hugely vary and depend on legacy applications internal performance/dependencies.

Enablers: The availability of source code for the legacy systems and other available resources to build a suitable ESB that meets all stakeholder needs. The Clear separation of application components, middleware and (System/user) Data shall allow developers to make a head start on integration process through ESB and web services.

E. Phase V:: Opportunities and Solutions:

The opportunities presented by the integration solution can be viewed in form of core quality attributes, which can be considered:

Performance: The clear separation of middleware that uses high-performance enterprise-class message queuing solution results in this component being a performance enabler. The use of web-services for other messaging (especially internal) results in this factor becomes a performance constraint. However, this can be managed by using web-servers having sufficient capacity for the projected loads. Caching of service endpoints, scale-out of web components based on performance requirements, use of enterprise class technologies.

Usability: A separate module dedicated to providing management services of the system is an enabler for the usability attribute. A separate module for the Instrumentation Logger is yet another enabler for usability. This will allow system administrators to easily track transactions through the system for the following purposes Troubleshooting, Debugging and Auditing (for compliance to organisational processes, and for statutory compliance). Other enablers listed below have to be developed as part of the HLD and LLD. These can be either scripts or software modules with user interface.

Reliability: Runtime reliability shall be ensured in the system by having redundant failover modules identified and implemented during deployment. Stateless services allow load-balancing of critical components using specialised hardware devices. Non-runtime reliability shall be assured by having all newly developed modules specifically designed to cater to the boundary conditions of Initialisation, Failure, Recovery and Termination.

Security: The entire SHMS System shall be deemed to be running within a corporate firewalled environment. The Security aspect is covered from 5 angles: single sign on which is on authentication gateway, encrypted data access within Business services, Random queue number allocation by message broker, Double firewall and Instrumentation

Logger for auditing attacks. It shall employ a layered architecture with critical assets in the inner area:

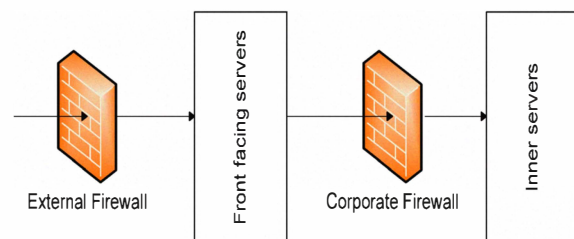


Figure 8. Security context of the SHMS System

F. Phase VI:: Migration Planning:

The SHS aims to smooth out any incompatibilities by following a migration plan which aims to address common issues/challenges faced by enterprise level amalgamation of applications include:

Adaptability: The use of generic components infrastructural components.

Simplicity: The proposed architecture uses the minimum no. of components in the simplest possible way.

Flexibility: The modular architecture proposed can be re-combined, enhanced, and scaled-out in a wide variety of ways, giving flexibility at deployment and maintenance time.

Modularity: The use of layers, components, and other techniques aid in a highly modular internal structure, enhancing overall system reusability.

Consistency: The consistent use of the underlying philosophy behind designing the component responsibilities and interfaces, and special attention being given towards achieving a final list of nearly equal-sized components enhances reusability, and the overall aesthetics of the proposed architecture.

Orthogonality: The clear-cut responsibilities of the various components, without any overlap, aids in the orthogonality of the components within the overall system boundary.

For Details on component level responsibilities see appendix.

G. Phase VII:: Implementation Governance

Pending

H. Phase VIII:: Architecture Change Management

Pending

II. CONCLUSION

The objective of this paper has been achieved by investigation into various available architecture models/frameworks and patterns that fit the category of integration facilitators, with a vision of the future demands for scalability and extensibility. While SHS is aimed to adequately meet all the stated and implied requirements, TOGAF supported in less rework on the existing applications to fit into the new framework. It also provided a smoother transition of the system from the immediate role of an application integration framework for legacy applications, to its eventual role as a pure application integration framework. Thus, HSIF shall be deemed the proposed architecture of the Architecture Team.

The proposed architecture – the HSIF – meets the business case requirements and allows existing systems (APP and PMS) to be open for feature rich front-end which provides secure interfaces. Therefore, both recent developments and our research outcomes in this field project are found to be very encouraging. However, more investigation is required before full confidence and wider acceptance is to take place in the ICT industry.

REFERENCES

[1] Reekie, John and McAdam Rohan, A Software Architecture Primer, 2005, Angophora Press.
 [2] Applied SOA, Service-Oriented Architecture and Design Patterns, Michael Rosen, BorisLubinky, Kevin T. Smith, Marc J. Bacler, Wiley Publishing Inc. 2008

[3] Java EE and .NET Interoperability, Integration strategies, patterns and Best Practices, Marina Fisher, Ray Lai, Sonu Sharma, Laurence Moroney, Prentice Hall, Sun Microsystems Press, 2006
 [4] IT Architecture and Middleware, Strategies for Building Large Integrated Systems, Second Edition, Chris Britton, Peter Bye, Addison-Wesley 2004
 [5] <http://msdn.microsoft.com/en-us/library/ms954595.aspx> Application Architecture for .NET: Designing Applications and Services, Last visit 4/11/2009
 [6] <https://www.ibm.com/developerworks/library/ws-esbscen/> Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture, Part 1, The Role of Enterprise Service Bus, last viewed 4/11/2009 Gunasekaran A., B. Koku (2002), Citation from [8], pp25-24.

Appendix: Details of components:

Smart Hospital System (SHS)	An integration platform that aligns hospital business strategies with IT investments through unification of hospital's existing core applications (Patient Management System and Accounting and Payroll System), providing a single point of access via implementation of a single UI (can be web based or rich-client) and enabling future integration of other systems (existing systems, or future deployments) into the Hospital's IT ecosystem.
SHS Application Development Standard	A published set of specifications for software applications that have to be conformed to, in order for them to integrate with the HSIF. Also provide interface description for HSIF.
User Interface	Combination of the existing user interfaces enabled to communicate via web-services using web service proxies that interface with the Message Router within Enterprise Service Bus (ESB). This is the interface for users to access existing legacy functionality as well as any future additional application via the HSIF.
Management UI	Management user interface to enable maintenance and administration tasks required to be carries out by the administrators of HSIF. This user interface is developed as part of this project for management of all the in-house developed components of the ESB. Vendor supplied technologies used within the ESB will have their own management interface.
External Service Requesters	Any business service or third-party application that will be authorised to access specified hospital services using HSIF.

Message Router	First point of access for the UI and External service requesters which will dynamically locate the appropriate service requested and route the request accordingly. Message Router, mediates communication and service calls to provide a separate contract with the service requesters and the service providers. This separation of service contracts enables changes in the service providers with minimal effect on the consumers.
----------------	--

Service Directory	Repository of services available in the hospital services ecosystem. Service directory will enable service location transparency in conjunction with the Message Router.
B2B Gateway	Make a selected subset of hospital's services available to external organisations in a controlled and secure manner.
Management Module	The management engine that the Management UI interfaces with in order to carry out administration and maintenance operations on the custom components of the ESB. This Module may not be required depending on the choice of technology used to build the ESB.
Enterprise Service Bus	Collection of Message Router, Service Directory, Instrumentation Logger and the Management Module which carries out the core integration responsibilities within HSIF and enables future progression of hospitals IT ecosystem towards a more standard Service Oriented Architecture.
Single Sign-On	Provides authentication (does user have access?) and authorisation (what user has access to?) services for the hospital as a whole, enabling a users to have a single credentials for accessing multiple legacy applications and new systems deployed in future.

Service Choreographer	Combines the various existing services across multiple legacy applications (PIMS and APP) into a higher level business service. Service Choreographer hides the granularity of the existing legacy functionality into more atomic SOA like services in line with future direction of the hospital.
Business Services	Constitutes business services designed to align to the hospital business requirements based on to encapsulate the legacy application API calls into an aggregated, higher level, and atomic service call. Each business service is the invocation point for the legacy functionality.
Business Workflow Orchestration	Describes and coordinated the workflow of how services interact (legacy and new), including the logic and the order of interactions.
Enterprise Data Integration Service	Provides for data integration by mapping the relevant information from the legacy system
PIMS Business	Business logic layer of existing Patient Information Management System in the hospital. The business functionalities contained in this layer are to be web service enabled for integration with the other systems in the hospital via HSIF.

APP Business	Business logic layer of existing Accounting and Payroll System in the hospital. The business functionalities contained in this layer are to be web service enabled for integration with the other systems in the hospital via HSIF.
Business Component	Business components that provide the existing functionality within their application. The purpose of this project is to integrate the functionality of these components transparently and provide access via a single user interface. The internal functionality of these components is not required at this level.
APP and PIMS Databases	Existing databases of each legacy application in the hospital.
Data Access Logic Components	DAL Layer of existing Accounting and Payroll System in the hospital.
Data Access Logic Component	Data Access components that provide data access to the application data store.
APP Data Store	Legacy Accounting and Payroll Database
PIMS Data Store	Patient Information Management System Database
Service or Component	Any service or application component already listed in this table that is required to be monitored and or audited for performance, errors and or exceptions.
Message Queue	Persistence message queue which service or component that are being monitored will submit events to.
Instrumentation Logger	A service that will read the submitted messages submitted from the message queue and writes them to the instrumentation database.
Instrumentation UI	Web user interface to display the instrumentation information in the database

Details of SHS component Interfaces

Interface no.	Description
1,2	Web-service interfaces between UI, external service requesters and the Message Router. This interface is responsible for mediating requests from service consumers to the service providers by being the first and only point of contact between service consumers and the Enterprise Service Bus (ESB). The advantage of the separation of the service provider interfaces to the UI is that the consumer service contracts and policies will not have to change by changing the service providers and/or their service contracts.
3	External organisations that consume some services provided by the hospital
4	Web-service interface between the management User Interface and the Management Module of the ESD. The management module will only control the custom developed component of the ESB.
5	Web-service interface of SSO is invoked by Message Router in order to authenticate the request as well as retrieve the callers roles based access profile.
6	Once a service request is received by the Message Router and authenticated and etherized by SSO, message router will request the most suitable service provider for processing the request by engaging the Service Directory and receiving the service end-point where it will forward the request to.
7	B2B gateway will relay the external client's request to the message router for appropriate action.
8	Management Module will send a command to the modules it can manage and will receive the outcome of the request (failed, succeeded) and execution details.
9	ESB will invoke the appropriate business service based on the request message.
10	In case the service request if for a long running process and required a business workflow, ESB will forward the request to the Business Workflow Orchestration which will take the request through the appropriate business workflow process and return the results once completed.

11	Orchestration invokes legacy functionality via the business service.
12,14	Business service invokes the legacy functionality through their web-service interfaces
13	Business service queries the enterprise database (before legacy systems) to locate where the required information lives (which legacy system) and get the appropriate data record keys
15	Service or application components that are setup for monitoring will send a status message to a message queue
16	Instrumentation service that reads the messages in the queue and writes them into the instrumentation database
17	Instrumentation interface will read the information from the database for display