

# Success Story

“Migration to an integrated .Net-based Web solution results in interoperability, cost savings & improved customer service”



## Background

As the healthcare industry shifts from a traditional paper-based system to on-demand live data access, the client had several IT applications for bringing a highly fragmented physician network into a 21<sup>st</sup> century IT framework in compliance with the regulatory and technological requirements of the AMA, HIPAA, CMS, NCQA and other federally mandated regulatory organizations. Client wanted to take their core healthcare applications to next generation architecture to achieve technology leadership in the healthcare market. The product was also desired to be highly interoperable, location-transparent and capable of integrating with other healthcare industry applications. The existing thick-client system provided good functionality, but it had limited scalability. To improve profitability and customer service, the client needed a Web-based solution.



### The Client

A Pennsylvania, USA based technology solution provider operational in several key cities enabling information access, data sharing, enhanced clinical care and improved productivity in the healthcare sector by developing practical applications and infrastructure deployment to aid the healthcare industry shift from a traditional paper-based system to on-demand live data access.

### SUMMARY

The imperative to improve profitability and customer service made the client re-examine its IT needs. It needed to integrate the features of all its applications into a single integrated Web application. The existing IT system was a thick-client PowerBuilder solution designed for the physician community. A complex client/server solution with a rich user interface, the solution and its associated database schema offered extensive functionalities to meet all the requirements of mid-size physician offices. However, the client/server architecture provided limited scalability, catering only to a limited number of clients at a few locations.

Enosis partnered with client to migrate the existing applications running on PowerBuilder based client server architecture, to a new web architecture based on the Microsoft.Net framework. Enosis was instrumental in implementing all GUIs to newly designed web based applications. The new browser based applications helped client to improve usability and scale up to service very large number of customers. Enosis collaborated with client on the design of the new architecture to enable rich functionality to integrate with leading applications in the Healthcare continuum.

### Major Challenges

- Mapping the functional and non-functional requirements onto a Microsoft ASP.NET based application.
- Meet the stringent performance and scalability requirements under low bandwidth conditions.
- Mitigate the technology risks involved in a complex migration project
- Provide an efficient, effective, and low-cost IT solution
- Develop a technical architecture at a low total cost of ownership (TCO) for a Web-based application that would eventually replace the thick-client applications.
- Demonstrate the benefits and business value of migrating to a “future state” IT configuration based on Microsoft .NET technologies, early in the development life-cycle
- Improve Usability, reduce desktop footprint and maintenance

### ENOSIS' APPROACH

As the client realized the benefits of developing a Proof-of-Concept (POC) instead of waiting one year to see the full application in production, the project started with a small POC phase to validate the Microsoft .NET architecture, demonstrate the benefits of the solution and mitigate the technology risks involved in a complex migration project. It involved building a set of infrastructure and common application blocks based on the design and development of best practices as recommended by Microsoft.

After the successful execution of the POC, Enosis used the learning to re-engineer the large PowerBuilder client/server application and execute a full-application migration (full scale application development) from PowerBuilder to the .NET-based Web application which includes:

**System Architecture**

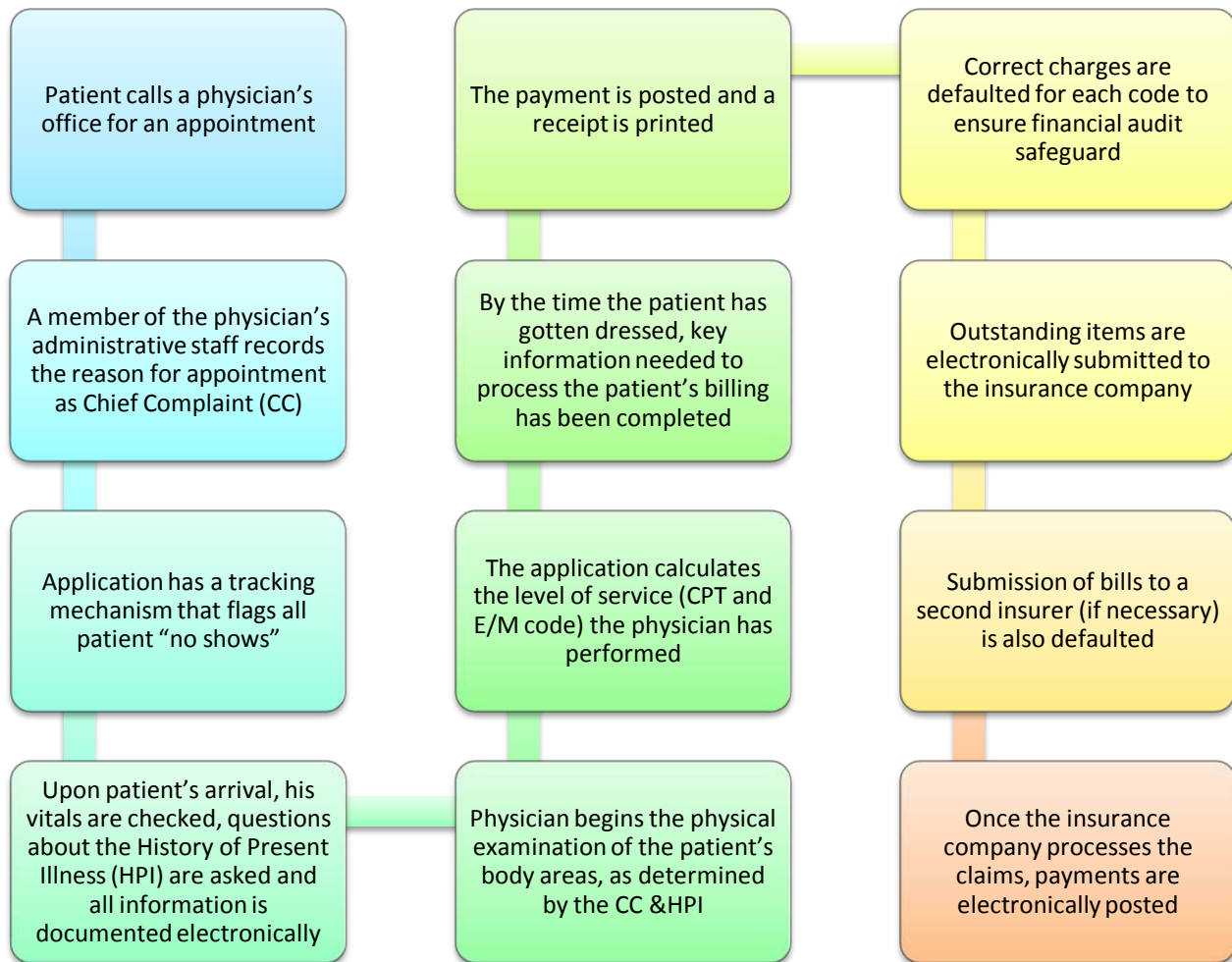
This involved defining the external interfaces the application would expose to integrate with partners like insurance, diagnosis companies and other customer solutions. The POC demonstrated integration with third-party solutions like insurance claim processing, electronic payment gateways etc.

**Business Re-Engineering**

Since the existing application was a tightly coupled two-tier application, one of the biggest challenges was to insulate business rules from the UI and extract business rules from the existing PowerBuilder application quickly. There was no readily available methodology or tools to extract them, so this had to be done manually.

**Reporting on Browser**

Crystal Reports was used for report generation. Reports in the existing client/server application were very complex, running into tens of thousands



Application Work-flow

of rows and 40–70 columns. This functionality had to be mapped to relatively simpler reports on the browser without losing the functionality provided by the existing application.

### Effective Feature Mapping

Another challenge on this project was to deal with the “resistance to change” of rich UI features which were client/server specific in legacy applications. Most screens had tightly-bound data controls, tabs, multiple grids, and pop-up windows. The team had to re-engineer a lot of screens in the application to fit the browser-based UI. The team built multiple prototypes and carried out performance and stress tests to show the impact of each choice. This enabled the client to make decisions based on the implications and limitations of the Web application.

### User Interface Architecture

Enosis carried out a review of the desired user experience to understand the usage patterns. This led to the development of the user interface architecture. Enosis and the client jointly engineered a user interface architecture that adopted a template-driven development approach, and also standardized the look and feel of the product. The general architecture of the whole system was also validated apart from creating the user interface architecture.

### Best Practices

Enosis designed, developed, tested and delivered all modules of the product suite on the Microsoft .NET platform. Software engineering best practices were applied to develop a loosely coupled, standards based application and enhance the interoperability. To reduce the time-to-market, Enosis adopted a pyramid approach to development. Enosis identified a minimal set of modules that were mutually exclusive and collectively exhaustive and encapsulated all the complexities in the system. Using this minimal set of modules, Enosis built a proof-of-concept to demonstrate the feasibility of the architecture and then prioritized the

development of the remaining modules to meet the client’s stringent timelines.

Enosis built standard components for navigation, security access, error handling, data-grids etc. to reduce the time-to-market and lay the platform for shared services. Enosis also helped the client with the migration of data from legacy applications. The application was rolled out in phases to end-client locations.

### BENEFITS

- Deployment of the new application did not entail a huge one-time infrastructure upgrade across the sites because the application has been tuned for optimal performance in low-bandwidth networks.
- Since the solution is Web-based, it reduced the costs involved in installation and maintenance of the application considerably. This was a key benefit for the client since it is expected to rapidly grow over the upcoming years.
- The execution of the POC helped to ensure key functional requirements such as UI, customization, and personalization, and it helped the company meet non-functional requirements such as performance, scalability, security, and reliability.
- Simplified maintenance process and significantly reduced Total Cost of Ownership (TCO)
- Enosis forged a strategic partnership with client to provide management services, including implementation and database administration support, for product rollout at end-customer sites

### TOOLS AND TECHNOLOGIES

**Programming Language:** ASP.NET

**Database:** SQLServer2000