SIRUSTI

case snapshot

# Intranet web application with web services

SIRUSTI APPLICATION SERVICES

## Client

Our client is an IT service company in Singapore providing IT services to government organizations. Sirusti worked as technical partner (Vendor) for this project, involved in

- Application Design
- Development
- Testing
- UAT & Live deployment at client's environment
- UAT Support

## Business Challenge

- This is an Intranet Web Application has features include the Registering of the Consultant, Exam Schedule, Tracking of the Exam and Results for the Consultant and Issue Certificate and Renewal of Certification. Various Reports are generated by using Reporting Services for the Application users to track easily.

- This application facilitate capturing the information of the current and new consultants, awarding certification to new participants and tracking the number of clinical case conferences and continuing education workshops/talks the consultant have attended so as to fulfill the re-certification requirement.

## Solution

**Design**

This system is three tier system, consists of

- **Web** - The presentation layer of the application
- **BAL (Business Logic Layer)**- Web services serves the required data requested from presentation
- **Database Access Layer (DAL)** - Database, where data has been stored

WEB

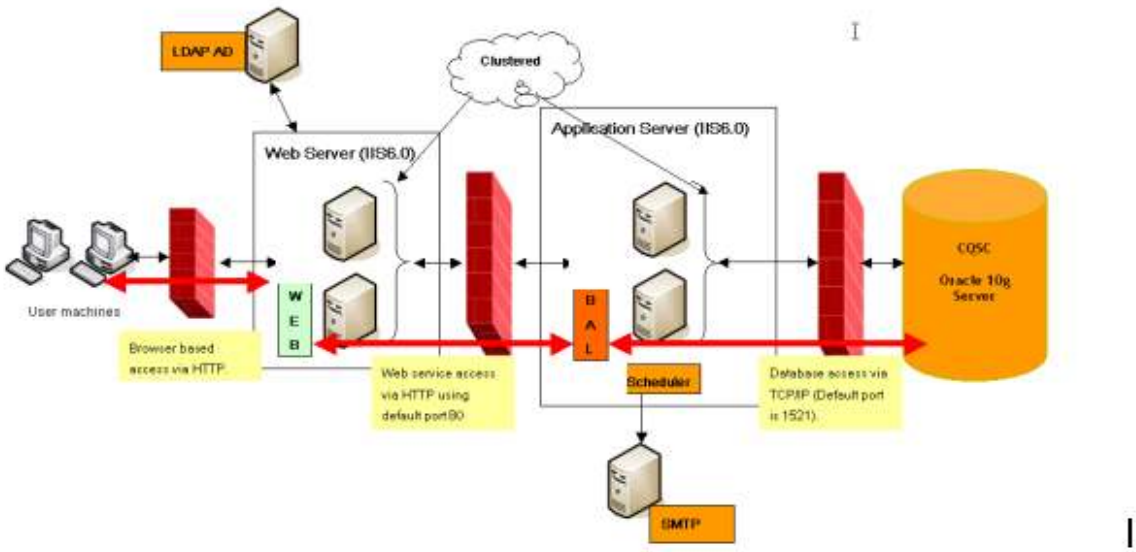
This is the web presentation layer, which will handle user requests and serve pages to the user via HTTP.

BAL

This BAL application service is used to connect to the database and retrieve data requested by the web application. All the business validations will be performed at this service. Web services will be used in this layer.

SCHEDULER

This will be separate application run as a scheduler service in the application server. This mailer will send notification e-mail notification to the users 3 months before any consultant is due for recertification using SMTP service based on application.




**Web Services**

Web services that sends & receives messages from & to **Web** application. The **BAL (provider)** and **Web (requester)** entities become known to each other. The requester and provider entities agree on the service description and semantics that will govern the interaction between the requester and provider agents.

The discovery files in Web are the act of locating a machine-processable description of a **BAL** Web services. Whenever, the webservice location changes, the discovery file must be updated with proper signatures.

**Web** Service engagement using a discovery service proceeds in the following steps.

1. The requester and provider entities "become known to each other":
    a. The discovery service obtains both the Web service description ("WSD") and an associated functional description of the service.

       The functional is a machine-processable description of the functionality (or partial semantics) of the service that the provider entity is offering.
    b. The requester entity supplies criteria to the discovery service to select a Web service description based on its associated functional description, and potentially other characteristics.
    c. The discovery service returns one or more Web service descriptions (or references to them) that meet the specified criteria. If multiple service descriptions are returned, the requester entity selects one, perhaps using additional criteria.

    The **Web** (requester) and **BAL** (provider) entities agree on the semantics of the desired interaction.
2. The service description and semantics are input to, or embodied in, both the requester agent and the provider agent, as appropriate.
3. The requester agent and provider agent exchange **SOAP** messages on behalf of their owners.

**Development Environment**

- **ASP.NET 2.0,**
- **C#,**
- **Web Services,**
- **SQL 2005,**
- **Reporting Services**

**Server Environment**

- Web Server OS - Windows 2003
- Web Server - IIS 6.0
- App Server OS - Windows 2003
- App Server - IIS 6.0
- Database - Oracle 10g

**Client**

- Browser - IE5.5 & Above

## Service Excellence

SIRUSTI successfully developed in very short period (18 days) and deployed in end client's (government organization) environment.