Integrated use of Web Technologies to deliver a secure collaborative web portal

Victor Chang
School of Electronics and Computer Science
University of Southampton, United Kingdom
icvc@ecs.soton.ac.uk

Abstract

This paper describes the design and operation of the OMII web site, which has to date achieved three major targets. The first target is the successful integrations in different technologies (Server 2003 + IIS server + SQL server + .NET and SUSE Linux + Apache Server) and standards (W3C / Macromedia and .NET). The second target is the enhancement in authenticating users, website security and back-ups and the final target is the development of a working procedure for static website development, dynamic .NET development and data migration on the two test servers and another two live web servers. The web site supports the easy registration and login, so that registered users can download and install our software.

Index Terms – Web Application Architecture, Software Engineering for WWW, .NET Framework, W3C / Macromedia standards, web-based technology integrations, middleware, and static/dynamic website development.

1. Introduction

1.1 Technologies for building the Web Application Architecture

Definitions for the web application architecture vary from organisation to organisation. Many organisations have regarded XML as a web service language and also as an important component for web application architecture, and they have provided both theoretical and practical demonstrations [1, 2, 3, 4, 9]. Some organisations have regarded Java, J2EE and JSP as powerful languages and platforms for web application architecture, and they have provided their software to download [6, 7, 8, 11]. There are also organisations adopting Java / JSP and XML as the web service platform and also making their own versions of software and publications [5, 8, 10]. This is true and valid due to the increasing levels of development and investment in the open-source Java-XML applications [6, 10] and the evidence of successful integration of both Java and XML technologies such as Apache Tomcat. However such definitions focus on the software applications but not on hardware solutions, and that is why these applications are known as web services, which are application interfaces, not object interfaces, for the platforms such as .NET and J2EE [15].

Therefore the purpose for this paper is to define the term web application architecture - the successful integration of both software applications and system requirements. This takes considerations of computer / laptop hardware specifications, operating systems, internet connection middleware, network protocol and physical databases. System requirements specify that a minimum of CPU 2.60 GHz, Pentium Four, 20 GB hard-disk and Memory 256 MB required for the computers or laptops, running on the operating systems of Windows 98 / ME / 2000 / XP / 2003, SUSE Linux 9.0 and Macintosh 10.x . The internet connection middleware includes modem, cable modem, local area network, broadband, wireless network, mobile network, high speed internet or middleware, running operationally at 56 KBps and running at the optimum with at least 500 Kbps.

Currently, HTTP is the default network protocol for our web application architecture with the secure firewall, updated anti-virus software and authentication mechanism turned on. SMTP is the protocol to send the first-time registered users their login and password. In order to transfer files between the test and live servers, authenticated network mapping, secure shell (SSH) and windows secure copy (WIN-SCP) are used for enhancing the network robustness, data migration and website security. Physical databases include the SQL server (which can also be considered as a huge software application / middleware) and back-ups, and the relationship between the SQL server and web servers will be discussed later. Whereas back-ups rely more on hardware mechanism, which include backing up the website and SQL data on the developer's laptop and two machines and CDs, so that different versions of development and progress can be tracked down and can run on different machines in case of system failure or application faults.

1.2 Middleware and its relation to the Web Application Architecture

The definitions of middleware vary across different research institutes, and the followings provide the closer definitions to the focus of this paper:

"Software that mediates between an applications program and a network. It manages the interaction between disparate applications across the heterogeneous computing platforms." [19]

"Software that facilitates interoperability by mediating between an application program and a network, thus masking differences or incompatibilities in network transport protocols, hardware architecture, operating systems, database systems, remote procedure calls and so on" [20]

"Application Server: A server that centralizes and consolidates application services. Web servers create tiered environment by being between a browser client and database server. Software that facilitates the communication between these two applications is called middleware. It provides an API through which applications invoke services and it controls the transmission of the data exchange over the network. There are three basic types: communications middleware, database middleware and systems middleware." [21]

Based on the above definitions, the term middleware can be defined as "the software and hardware infrastructure to support model of computation and information utilities on demand" [22]. Middleware is very closely related to grid.

However, the focus for this web application architecture is still on integrations of different technologies and standards on the software applications, which follow up two main streams of web standards. The first one is .NET Framework [16, 17, 18, 19] which includes .NET dynamic page development on the test servers and the dynamic web applications on the live website. We follow standards from W3C [2, 3, 4, 5, 7] / Macromedia [20, 21, 22] and standards for general / static website development and an interoperable web application. The details are discussed in Section 2.

2. Web application architecture: Development

2.1 Overview

The common practice for web development is to first create a working environment on a test machine, which includes the software programs for web programming, database management and system administrations. The next step is to create a mirror to the live website, and this means that the working environment and application software on both the test server and live server must be identical. Figure 1 shows the structure of the live machine.

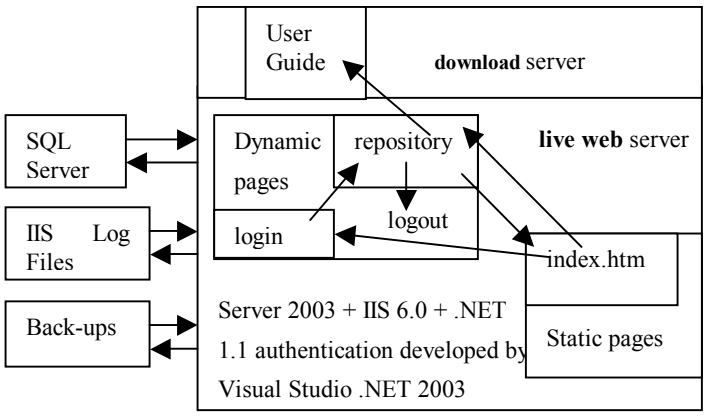

Figure 1: The structure of live machine

The structure of test machine 1: Identical to the live web server since they are mirror to each other. The only exception is that live SQL server is the primary database and test SQL server is the secondary database. Primary database is kept up-to-date and all the updates are then securely transferred from live database to test database.

The next step is to integrate both dynamic and static page development into a single web application in order to maximise the functionality and information management of the website. This concept is similar to a computer that has dual-boot systems such as Windows XP and Linux running on the same machine. Figure 2 below is the explanation.

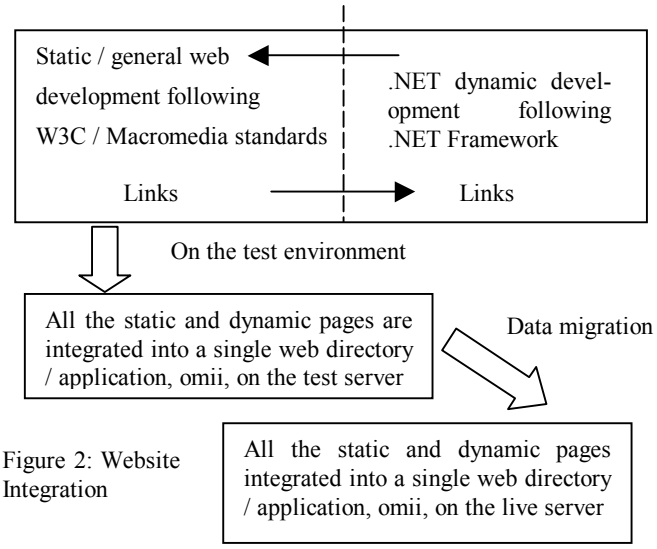

Figure 2: Website Integration

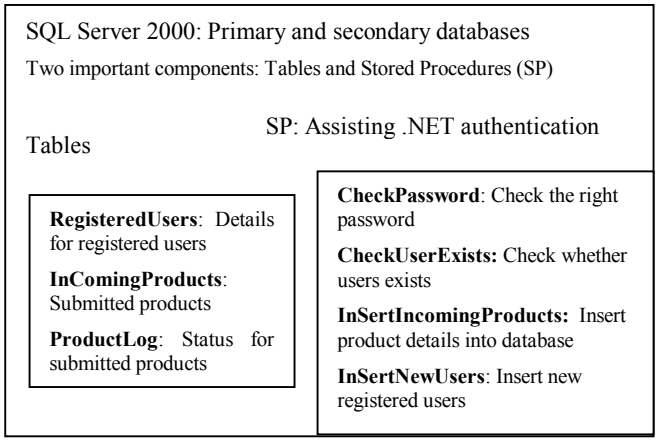

Figure 3: Database components and functions

2.2 Development stage: static and dynamic web development

The new OMII website development has undergone two major stages between August 2004 and November 2004. Throughout these two stages, all the major development, modifications and testing have taken place in two testing servers, which include "test1" server for static website development and test machine/server for dynamic website development and website integrations. The static and dynamic development starts on the test environment. The static web development consists of a working combination of HTML 4.01, JavaScript 1, JavaScript 1.2 and CSS

scripts on each page. Each static webpage has followed the W3C standard. At the end of each webpage development, it is necessary to run the codes on the W3C syntax validator on the W3C site [8] and debug those codes with errors.

The next step is to work on detailed testing and open up six types of browsers on the three different operating systems, to see the effects on the text, drop-down menu and browsers. One test result shows that the drop-down menu works differently for Firefox and Konqueror browsers, and all the codes with errors are then debugged. Another observation was that static pages look differently on the Macintosh and SUSE Linux, and the errors are then debugged. Thorough testing on all web pages on the different resolutions, internet connecting middleware, internet speed and computer requirements, it is clear that hardware-oriented factors affect the optimisation of using the website. Getting through this step is important because generic static codes are embedded for all the dynamic pages in order to have the same appearance on the website, and therefore the success of this step means the dynamic pages work as good as the static pages on the different settings of operating systems, browsers and other hardware-oriented factors.

The OMII Repository is based on the combined technologies of Windows Server 2003, Internet Information Service (IIS) 6.0, SQL Server 2000 with Service Pack 3, .NET 1.1 and Visual Studio .NET 2003. The coding style is that one static ASP.NET file acts as a front-end page and one dynamic VB.NET file acts as the back-end page. For instance, login.aspx is the static page and login.aspx.vb is the dynamic file executing login function. There is an XML file, login.aspx.resx, generated by login.aspx and login.aspx.vb upon successful rebuilds.

Referring to Figure 2, the integrated web directory must be tested thoroughly on the test server. The first step is to browse all the static pages and spot any technical or grammar bugs. The next step is to test on the dynamic pages by registering on the test website, receiving a password, logging into the website, downloading software, submitting a product and going back to repository page. The critical step is to check whether the configuration of the SQL server on the test server is the same as the primary SQL server. Referring to Figure 3, Stored Procedures are used to encapsulate a set of operations or queries to execute on a database server. For example, "CheckPassword" is used in one part of login.aspx.vb (code behind login.aspx) and "CheckPassword" is performing the SQL query in SQL Server 2000. Table 1(a) and 1(b) below show the SQL syntax for "CheckPassword". .NET security depends on the .NET configuration and SQL Server database checks.

```
dbconn = OpenOMIIDatabase()  'Now Open the database and check the password
sqlcmd = New SqlCommand("CheckPassword", dbconn)
sqlcmd.CommandType = CommandType.StoredProcedure
sqlcmd.Parameters.Add("@password", SqlDbType.NVarChar).Value=password
sqlcmd.Parameters.Add("@email", SqlDbType.NVarChar).Value = user
```

Table 1(a): One part of login.aspx.vb that has a Stored Procedure

```
CREATE PROCEDURE [CheckPassword]
        (@Email    [nvarchar](50),
        @Password    [nvarchar](8))
AS SELECT * FROM [OMII].[dbo].[RegisteredUsers] WHERE Email
LIKE @Email AND [Password]=@Password
GO
```

Table 1(b): SQL script for "CheckPassWord"

3. Web Application Architecture on the live website

3.1 Data Migration

As a rule of thumb, when the web application on the test server (test) and the database for the test server (test_SQL) is fully functional, then migrate the whole web application from the test server to the live server (live) and the web application on the live and its database (live_SQL) is ideally functional. Using Windows Explorer and mapping the authenticated network drive/directory, are sufficient to move the web application from test to live. There is another web server running on Apache 2.0 SUSE Linux for User Guide. WIN-SCP is the software for data migration because it is very secure and suitable software to transfer files on the Windows to SUSE Linux. Before performing data migration, backing-up files on the test, test_SQL, live, live_SQL and Apache website is necessary.

3.2 On the live website

Following the data migration, the IIS server is restarted and the web application tested by closely following these steps: registering the website, getting the password generated by the SQL server (live_SQL), logging into the website, navigating the dynamic pages, downloading the software, checking the updated information on live_SQL and finally logging out the website. If all the steps work, that means the web application is functional on live and live_SQL. Figure 5 expands on this workflow. Furthermore, the interactions between static pages and dynamic pages must be checked, particularly the drop-down menu of the website, which provides links between static and dynamic pages.

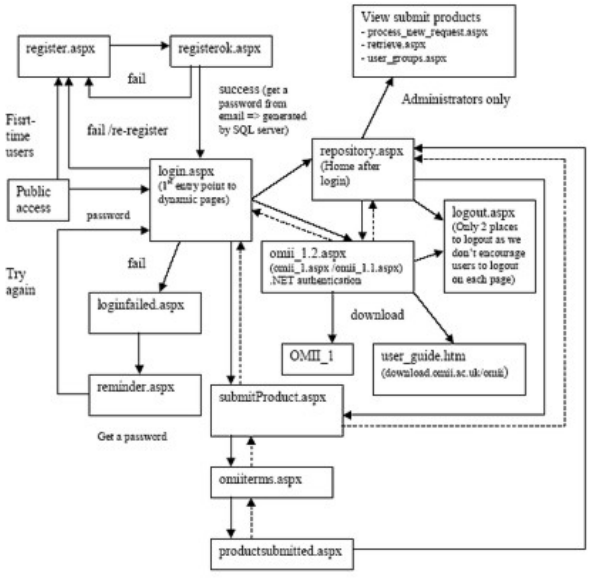

Figure 4: Navigations for OMII dynamic .NET pages

3.3 Log files

Log files for the website serve important functions as they provide information for the organisation on all the activities when the users visit the website. This information includes:

(a) IP addresses of the users (essential);
(b) Date, time and duration of the visits on the website;
(c) Pages that the users have visited;
(d) Software or files the users have downloaded (essential).

The daily log files also assist the webmaster to understand whether the daily activities on the website are "healthy" by making remarks when there are abnormal activities such as forced entry without logging into the secure website. The daily log files also assist the webmaster to understand the user behaviour, which include the number of downloads, the number of hits / visits a day, the particular users' habits when visiting the website. Therefore the web server, live, is having daily log files analysis and also protected by an anti-virus software, firewall and .NET authentication to enhance the security. The SQL server and the web application are backed up on the regular basis.

4. The entire Web Application Architecture

Section One to Section Three of this paper have described each stage of development, and each of development components altogether can make up the entire web application architecture. Figure Three on the next page is the diagram of the complete OMII Web Application Architecture. On the Apache server, an interesting observation is that .NET files cannot run smoothly on Firefox and Mozilla browsers as these non-Microsoft browsers cannot interpret what "codebehind" of .NET means. Therefore, all the dynamic pages on the Apache server are linked back to the IIS server of live server. Another observation is the .NET authentication, which can be easily administered by configuring Web.conf file, which is similar to web.xml in the Apache Tomcat Server. The Stored Procedure, "CheckUserExists", can also check whether those who login are the real users.

5. Conclusion

This paper presents the three major targets achieved by the OMII web application architecture. The first target is the successful integrations in different technologies (Windows Server 2003 – IIS server and SUSE Linux – Apache Server) and standards (W3C / Macromedia and .NET) on the test website and live website. The second target is the enhancement in authenticating users (.NET authentication), website security (interoperable servers, log files, firewall, anti-virus software) and back-ups (laptop, machines and CDs) and the final target is the development of a working procedure for static web development, dynamic .NET development and data migration on the two test servers and another two live web servers. This paper also re-consolidates the definitions of middleware: "The software and hardware infrastructure to support model of computation and information utilities on demand" [22].

The web site supports the organisational goal by providing easy registration and login, so that registered users can download and install our software 'product' – OMII_1, an emerging platform and standard for the web service applications. There is also a growing number of user communities in the UK, US, EU, China, Australia and East Asia, and the increasing uses of OMII_1 may drive the evolution of the OMII web application architecture and the repository.

6 References

[1] Henderson P, Hey T, Dunlop A, Newhouse S, De Reoure D et al, "Web Service Grids: An Evolutionary Approach", the UK e-Science Core Programme strategy paper, July 2004.

[2] W3C, "Extensible Markup Language (XML)." http://www.w3.org/XML/.

[3] W3C, "XML Schema." http://www.w3.org/XML/Schema.

[4] Jacobs I, Walsh N, "Architecture for the World Wide Web", W3C Working Paper 1st Edition, November 2004 http://www.w3.org/TR/2004/PR-webarch-20041105/

[5] W3C Syntax Validator http://validator.w3.org

[6] "Apache Jakarta Tomcat 5 Servlet / JSP Documentation" http://jakarta.apache.org/tomcat/tomcat-5.0-doc/

[7] "Java 2 Platform, Standard Edition (J2SE)" http://java.sun.com/j2se/

[8] "Gridsphere Portal Framework Documentation", http://www.gridsphere.org/gridsphere/gridsphere

[9] Dagiene V, Laucius R, Montvilas M, "A Metaprogramming-Based Model for Generation of the e-Learning-Oriented WEB pages", 2004 IEEE, London Metropolitan University, June 2004.

[10] Dagiene V, Laucius R, "Internationalisation of Open Source Software: Framework and Some Issues", 2004 IEEE, London Metropolitan University, June 2004.

[11] Costagoliola G et al "A Web-based Tool for Assessment and Self-Assessment", 2004 IEEE, London Metropolitan University, June 2004.

[12] "Microsoft .NET Framework Developer Centre" http://msdn.microsoft.com/netframework/

[13] "The .NET Six-Week Series Guide" http://msdn.microsoft.com/net/guide/

[14] Homer et al "Professional ASP.NET 1.1" Wrox Publication, ISBN 0-7645-5890-0

[15] Wall B, Lader A, "Building Web Services and .NET Applications", Osborne Publication, ISBN 0-07-213047-4.

[16] "Macromedia – Accessibility" http://www.macromedia.com/macromedia/accessibility/

[17] "Dreamweaver API Reference", Macromedia Edition, January 2004.

[18] "Dreamweaver MX – Getting started with Dreamweaver", Macromedia Edition, January 2004.

[19] ULI UIUC Glossary, http://dli.grainger.uiuc.edu/glossary.htm

[20] University Illinois AITS Glossary http://www.aits.uillinois.edu/live/Site.xml?document=Glossary.xml&focus=N16#m

[21] Thomas M, "XML Glossary", Minnesota State Colleges and Universities, http://krypton.mnsu.edu/~spiral/eta/glossary/indxGlossOOxml.html

[22] Calder M "The National e-Science Centre and UK e-Science Programme presentation", http://www.nesc.ac.uk/presentations/

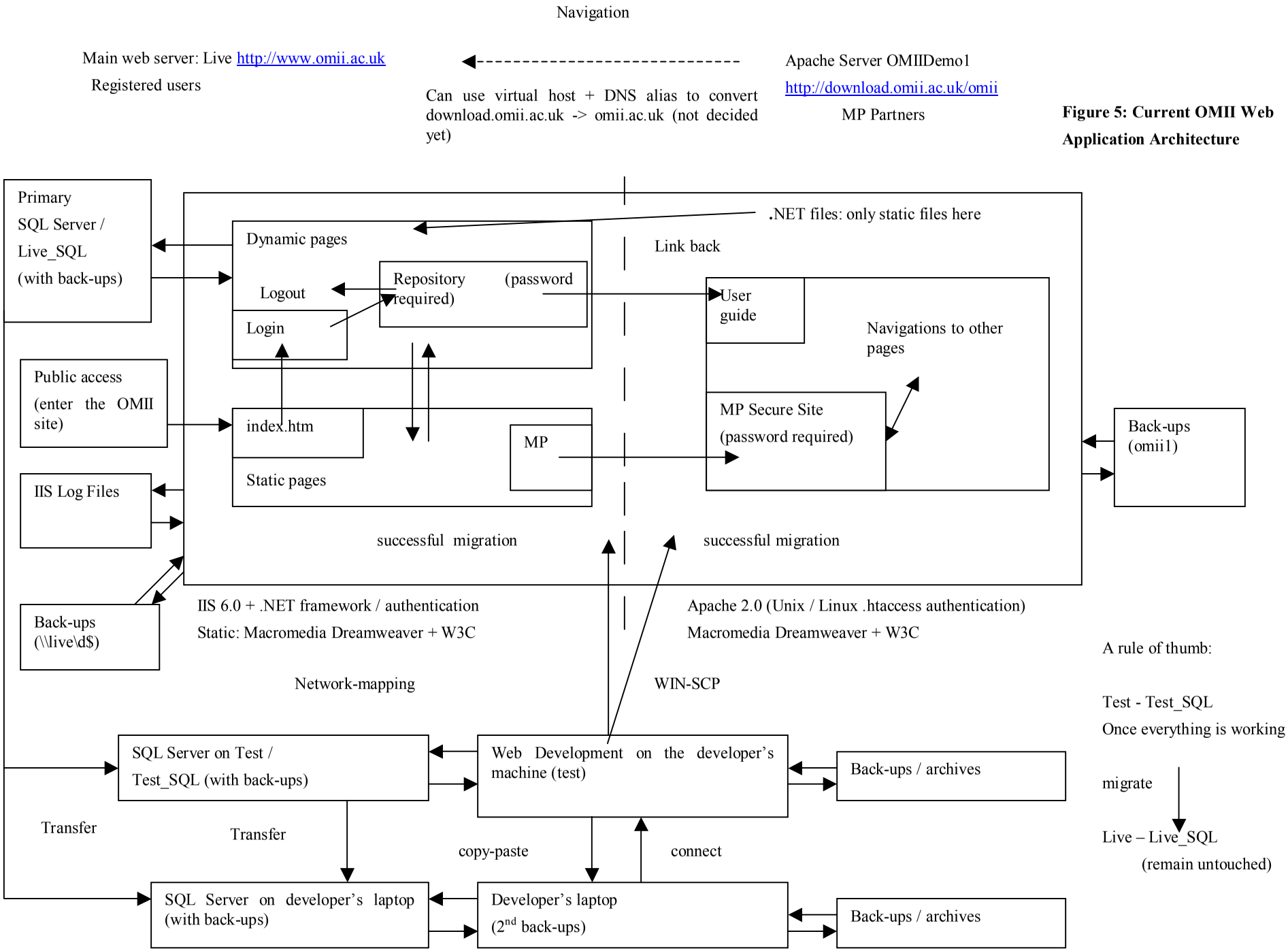

Figure 5: Current OMII Web Application Architecture