

OPEN SPECTRUM SOLUTION

Date : Dec 01, 2006
Subject : Paten Filing Number OSS_100_011206
Patent Name : OSS © Application Building Block (v2005.NET)
From : Open Spectrum Solution LLP
Michael Ng / mike@open-spec.com

This patent filing contains the following sub-sections:

Section I	Introduction and Objective of the Patent -----	Page 2
Section II	Concept & Architecture Specifications -----	Page 3
	1.1 Basic Concept & Diagram [Diagram II_1.1].	
	1.2 Logical and System Architecture [Diagram II_1.2]:	
	1.3 Application Logic & Architecture	
	1.3.1 N-tier	
	A. N-tier Hardware Architecture	
	B. N-tier Application Architecture	
	C. N-Tier User Interfaces	
	D. N-tier Data Access Layer (DAL)	
	1.3.2 SOA (Service Orient Architecture) and	
	1.3.3 Middleware – DAL and SOAP	
Section III	Practical & Competitive Advantages -----	Page 5
Section IV	Screen Shots -----	Page 6
Section V	Claims -----	Page 6

Section I: Introduction and Objective of the Patent

Today most business and government enterprises are not utilizing their overall IT resources efficiently due to they lack of a systemic architecture platform to achieve that purpose. For instance, it is not uncommon to find a large public organization has 6 IT hardware silos and maintains 6 different Facility Management if they deploy 6 different application software.

This happens due to corporations deployed their various applications on piece meals over time. It was not designed for accommodating large set up (or accommodating multiple applications with large concurrent users).

This method however is considered very costly and unproductive for enterprises for the following reasons:

1. IT Resources Utilization – under this scenarios, from our study, most silo of server's utilization rate is not more than 12% of its total capacity (CPU, RAM, OS, Database and other available resources);
2. Multiple Facility Management Team – in this case, total 6.
3. Commonly required utility software are seen at all application software – It is not uncommon to find the commonly required utility software are repeatedly re-coded and appeared at each application software. These include centrally control functions – LDAP, Encryption, Business Intelligence, Audit Trace & Track, Report Portal, Identity Authenticity, System Security Setting and others; and
4. Heterogeneous Platforms– It is uncommon to find that among the six (6) different application software, they are split evenly into two camps - Microsoft ASP.NET/VB.NET and Java J2EE, some in PHP and others. It makes IT streamlining difficult due to different programming codes required different skill set.

Objective of the Patent

The purpose of this patent filing is to explain how all the above mentioned issues can be resolved, streamlined, simplified and ultimately reduce the IT management cost considerably thru our **OSS © Application Infrastructure Building Block** (ASP.NET) or (J2EE). The concept can be depicted at Diagram [Diagram SV_1.1]

Its concept is based on:

1. Centralized approach – by consolidating all commonly and frequently used utility functions built into one single block. In this manner, IT resources utilization will be boosted up dramatically and it eliminates future application software from having to re-write the similar utility software;
2. Enforcing all application software, which sit on top of the utility block, to access these common utility software at continuous way thru http/XML (or SOAP) at distributed network (TCP/IP);
3. Flexible Access to heterogeneous database thru DAL (Data Access Layer) – build in a DAL where it permits any of the application to access any type of RDMS database system thru distributed network (TCP/IP);
4. Expanding hardware (servers) when the required elapse time (web transaction) is more than 3 seconds only to ensure consistent performance;
5. SOA (Service Orient Architecture) – structure all the commonly used utility software as services, instead of isolated and off-limit capsule;
6. N-tier – structure the source codes as N-tier platform where it can be duplicated and expanded thru multiple hardware as they grow;

To be more specific, the centralized utility software is listed below:

- 1 OSS © System Administration management module,
- 2 OSS © RSA Encryption module,
- 3 OSS © Authentication and Authorization Management module,
- 4 OSS © Audit Trace and Track management module,
- 5 OSS © Data Access Layer (DAL) management module, which allow any application to connect to any RDMS (Oracle, My SQL, MS SQL, IBM-DB2 and others)
- 6 OSS © Alert Management (Email, SMS and others)
- 7 OSS © File Access Right & Right Allocation Management

- 8 OSS © Middleware SOAP_DAL API: it permits heterogeneous applications (VS or J2EE) to access with no limitation;
- 9 OSS © LDAP (Lightweight Directory Access Protocol)
- 10 OSS © BI (Business Intelligence) Management Module,
- 11 OSS © Work Flow Management Module, and
- 12 OSS © Report Portal
- 13 OSS © AJAX Framework

The above utility software can be expanded to accommodate new or third party's.

Section II: Concept, Architecture Specifications of OSS © Application Infrastructure Building Block (v2005.NET)

- 1.1 **Basic Concept & Diagram** – The basic concept, architecture platform, flow and physical hardware attributes are depicted in [Diagram II_1.1].
- 1.2 **Logical and System Architecture**– is depicted at [Diagram II_1.2]:
- 1.3 **Application Logic & Architecture**– is built around SOA, N-tier and Middleware as explained below:
 - 1.3.1 N-tier: the core purpose of N-tier is to separating data from hardware server's processing power which in turn support optimal hardware's configuration.
 - A N-tier Hardware Architecture: At current prevailing technology, the N-tier allows hardware scalability up to 32 web servers, 12 application servers and 2 database servers. It can be further expanded the scale as the prevailing hardware and Operating System improve. [Diagram II_A to II-E];
 - B. N-tier Application Architecture: The N-tier architecture allows developers to construct the main application software with many de-attachable modular components where they can physically accommodate the hardware structure as explained in (A) above – that split the application modules into web, application and database servers. [Diagram II_1.2];
 - C. N-Tier User Interfaces: The N-tier is capable to surface the application's main source codes in different User Interfaces, for instance it can be deployed in Windows and Web, without the need to altering the underneath scripting codes. This means the N-tier will not require developers to rewrite the underlying application codes for different user interfaces. [Diagram II_1.2];
 - D. N-tier Data Access Layer (DAL): Since N-tier allows layer and layer of coding, it permits developers to a separate DAL, or the layer between Application Software and Database. DAL is not only designed as separate layer but is also as services where it permits communication between application with any other prevailing RDMS, such as Microsoft SQL, IBM DB2, Oracle 10i etc. [Diagram II_1.2].
 - 1.3.2 SOA (Service Orient Architecture): its features are built as services of interfaces, where irregardless of the application's platforms (whether .NET or J2EE) they all can exchange data or communicate with it thru SOAP (Small Object Access Protocol, via Http and XML). As it is in SOA, it can be re-used and redeployed again for many years irregardless of the changes of hardware, OS version and others; and
 - 1.3.4 Middleware API for inter-operatability between application software and SOA block- It permits heterogeneous application software platforms (ASP.NET, J2EE, PHP and others) to access the centrally encapsulated utility software thru our OSS © Middleware_SOAP_DAL, which are based on http, XML and Data Access Layer.

Section III: Practical & Competitive Advantages

The said architecture, thru **OSS © Application Infrastructure Building Block**, can yield the following cost saving via:

1. Consolidating heterogeneous hardware platform: Instead of maintaining multiple silos of servers for multiple applications, the corporations can now consolidate all different application software in a single, clustered server platform. In this manner, the corporations derive the cost saving via a single IT Facility Management Team (instead of multiple Facility Management maintenance teams) and it allows multiple application software to share commonly utility functions at similar physical unit;
2. Support Multiple Software Platform – the utility block is capable to support the the world major prevailing application software platforms, such as Microsoft VB.NET/ASP.NET/C++ , Sun Java J2EE, PHP and others due to the underlying coding architecture is parlayed into service orient (or SOA) parameter.
3. Resource Utilization: By consolidating multiple software under one single (scalable) hardware, it makes the IT resources – database, server Operating System (OS), rigid hard-disk, Memory DRR/SDRAM, main processor, and others –'s utilization scaled upward which derive major saving from subsequent eliminating multiple hardware and system software maintenance (man-power salary, reduced man-power's training need & upgrading and property occupation costs);
4. Make legacy system to be service orient (SOA) block or re-usable thru OSS © Middleware DAL – One of the core features of OSS © Application Infrastructure Building Block is the built-in Data Access Layer (or DAL). Its particular function is its capability to derive selective field of data from different databases, in a distributed network (TCP/IP), where the captured data be deposited into a centralized content depository, and then thru other customized middleware the corporation can generate much enhanced and different types of report and functionalities. In this method, corporations do not need to re-program existing legacy system to derive the desired enhanced report and functionalities, when business environment changes;
5. High Availability (HA) thru N-tier architecture – As the utility block is architected with N-tier. Then business and government enterprises can bring High Availability (HA) to its system. In other word, it supports high volume of concurrent users due to HA permits enterprises to add server to its existing hardware platform as clustered, up to 32 web servers+ 12 application servers + 2 database servers. The overall effect is acting as single computer unit but parcel out the total workload evenly among them. In HA sense, if any one of the said clustered servers fails to functions, the balance good ones will take up the required load and continue to provide the computing services continuously;
6. Database Independence – The built-in OSS © Data Access Layer (DAL) allows client to deploy any database of his/ her preferences, from Oracle to IBM DB2, to Microsoft SQL 2005 to My SQL and others;
7. Consistent Performance (3 seconds of elapse time on at least 90% of web transactions) thru N-tier Scalability– it is a term that describes when the servers are clogged (when elapse response time is more than 3 seconds for more than 10 % of the on-line transactions) due to sustained period of time of increasing number of users who access the system simultaneously. This can be overcome thru the built in N-tier platform, where it permits adding more servers to the existing hardware platform to permit larger number of concurrent log-in access. This scalability, at this current technology, permits up to 32 web servers and 12 application servers to serve concurrent users of 32,000. Scalability can be deployed thru 2 tier, 3-tier, 4 tier and many other tiers, depending on the client's requirement;

Section IV: Screen shots

Some of the screen shots of the patent is depicted and briefly explained below.

1. System Security – This feature allows System Administrator to remotely configure various log-in parameters for each user. [Diagram IV_A].
2. System Security @ Audit Trace & Track Module – It allows System Administrator to keep tracks each user's log-in activities where the system tabulates the user activities in Date, Time, Action, Field he/she accessed and others. [Diagram IV_B].
3. System Security – LDAP (Lightweight Directory Application Protocol), it permits System Administrator to coordinate and comply with central/ corporate security policies. Alternatively, client can de-activate this feature if LDAP is not required. [Diagram IV_C].
4. System Security @ Name/Role-Based Access Right Allocation– It permits System Administrator to assign specific access right to each user to the files, in accordance to the user's role and responsibility. [Diagram IV_D] and [Diagram IV_E].

Section V Claims

The patent intends to claim for the right and consolidated methods of how to group a set of commonly used utility software in a single block; making them as service orient architecture (SOA); building access right to these SOA thru SOAP and DAL; allowing integration with N-tier based technology as described below:

7. Centralized approach – by consolidating all commonly and frequently used utility functions built into one single block. In this manner, IT resources utilization will be boosted up dramatically and it eliminates future application software from having to re-write the similar utility software;
8. Enforcing all application software, which sit on top of the utility block, to access these common utility software at continuous way thru http/XML (or SOAP) at distributed network (TCP/IP);
9. Flexible Access to heterogeneous database thru DAL (Data Access Layer) – build in a DAL where it permits any of the application to access any type of RDMS database system thru distributed network (TCP/IP);
10. Expanding hardware (servers) when the required elapse time (web transaction) is more than 3 seconds only to ensure consistent performance;
11. SOA (Service Orient Architecture) – structure all the commonly used utility software as services, instead of isolated and off-limit capsule; and
12. N-tier – structure the source codes as N-tier platform where it can be duplicated and expanded thru multiple hardware as they grow;

----- End of Page -----