

OSS ©
RSA ENRYPTION SERVICES
Middleware

FOR Application Developers

Version: VB.NET (v2.0)
Updated: 4Q/200, g1d0r1
Reference: Middleware 0462/86

CONTENT DESCRIPTION

1. INTRODUCTION

- 1.1 Introduction
- 1.2 Business Applications

2. TECHNICAL SPECIFICATIONS

- 2.1 Programming Codes
- 2.2 Database
- 2.3 Application Architecture
- 2.4 Hardware Architecture

3. APPLICATION'S ARCHITECTURE, FEATURES, DESCRIPTIONS AND SCREEN SHOTS

- 3.1 Application Architecture
- 3.2 Application Feature Descriptions
 - 3.2.1 Fundamental
 - 3.2.2 Screen shots of Deploying OSS © RSA Encryption Services (VB.NET)

4. HARDWARE CONFIGURATIONS

1. INTRODUCTION

1.1 INTRODUCTION

The application, **OSS © RSA Encryption Services for Application Developers (VB.NET) v2.0**, was architecturally designed:

- 1.1.1 As a **SOA (Service Oriented Architecture)** based platform where its Encryption services are to be exposed clearly to the developers who can embed them in their applications where it helps eliminate the duplicate effort of writing N number of such encryption-and-decryption services if their applications have N number of modules or fields that call for such requirements; and
- 1.1.2 As **N-Tier** code-structure, where it supports the embedded codes in tier-based hardware, such as cluster-server environment. This design is basically to enable the developers who design their applications that support High Availability (or Failover), Scalability and provide Consistent Performance (exp: 3 seconds for all web transactions' elapse time).

1.2 BUSINESS APPLICATIONS

Some of the widely used real life applications for RSA Encryption services are:

- 1.2.1 **Encrypted Highly Sensitive Fields:** For enterprise-grade corporations which deploy their centralized and complete-web-enabled IT applications across public network (internet) and wide span of geographic locations for their multiple business divisions, they require certain fields be highly hardened and encrypted, some examples are:
 - A. Human Resources Solution- the services enable the enterprises to encrypt highly sensitive fields, from resumes to employment offered letters, from salary records to discretionary spending data for certain management members;
 - B. Enterprise ERP System- the services enable the non-related employees from getting to access the data such as, products' selling prices, accounting records, customer contact information as well as certain sensitive legal documents; or
 - C. System Security: For major enterprises which enforce LDAP (Lightweight Directory Access Protocol) based access to certain privileged information, the services of encryption is the added enabler for their requirement and enforcement policies.
- 1.2.2 **Professional Services Providers:** For major professional service providers, such as international legal, accounting, audit, company secretarial firms or financial institutions (banks, trust etc), identity management and privacy protections are the pre-requisite for their professional, the services of encryption is an enabler to help them to achieve such goal;
- 1.2.3 **Online Public Tender Portal:** For major public sectors, the government's tender process is a regular and on-going activity, where certain information is required to be encrypted and protected. In this application, only the members of Open Committee Panel are allowed to access the bid information after the closing tender's date.

For those pre-closing date, all the submitted bids' are required to be encrypted; and

- 1.2.4 **Online e-Commerce Portals & Credit Card Payment:** For the new age of e-Commerce, the services of encryption from credit card payment's records as well as online users' identities, are the pre-requisite for building the websites' long-enduring trust and integrity.

2. TECHNICAL SPECIFICATIONS

- 2.1 **Programming Tools:** The application's source codes were written in Microsoft VB.NET (.NET Framework 2.0) and AJAX,
- 2.2 **Database:** The application has a built-in connectivity middleware, OSS © Data Access Layer, which allows client to deploy any Relationship Database Management System (Oracle 11g, MS SQL 2005, IBM DB2 or My SQL);
- 2.3 **Application Software Architecture:** The application's architecture was designed in:
- A. **SOA** (Services Orient Architecture, details are at Section 2.2 below) which its functions are exposed to the developers as a service; and
 - B. **N-Tier:** It supports the tier-based hardware (cluster-server) environment for deployment.
- 2.4 **Hardware Architecture:** As the core application's base architecture is based on SOA and N-tier, it supports either the deployment of Client-Server (single server's hosting) or layer-based hardware (or cluster-server) in its physical deployment.

In a cluster-server environment, it can be hosted at the Application Server tier. The N-tier is inherently providing clients the great flexibility of having High Availability (HA) and Scalability features. For more details, please refer to Section 3 below.

3. APPLICATION'S ARCHITECTURE, FUNDAMENTAL & SCREEN SHOTS EXAMPLES

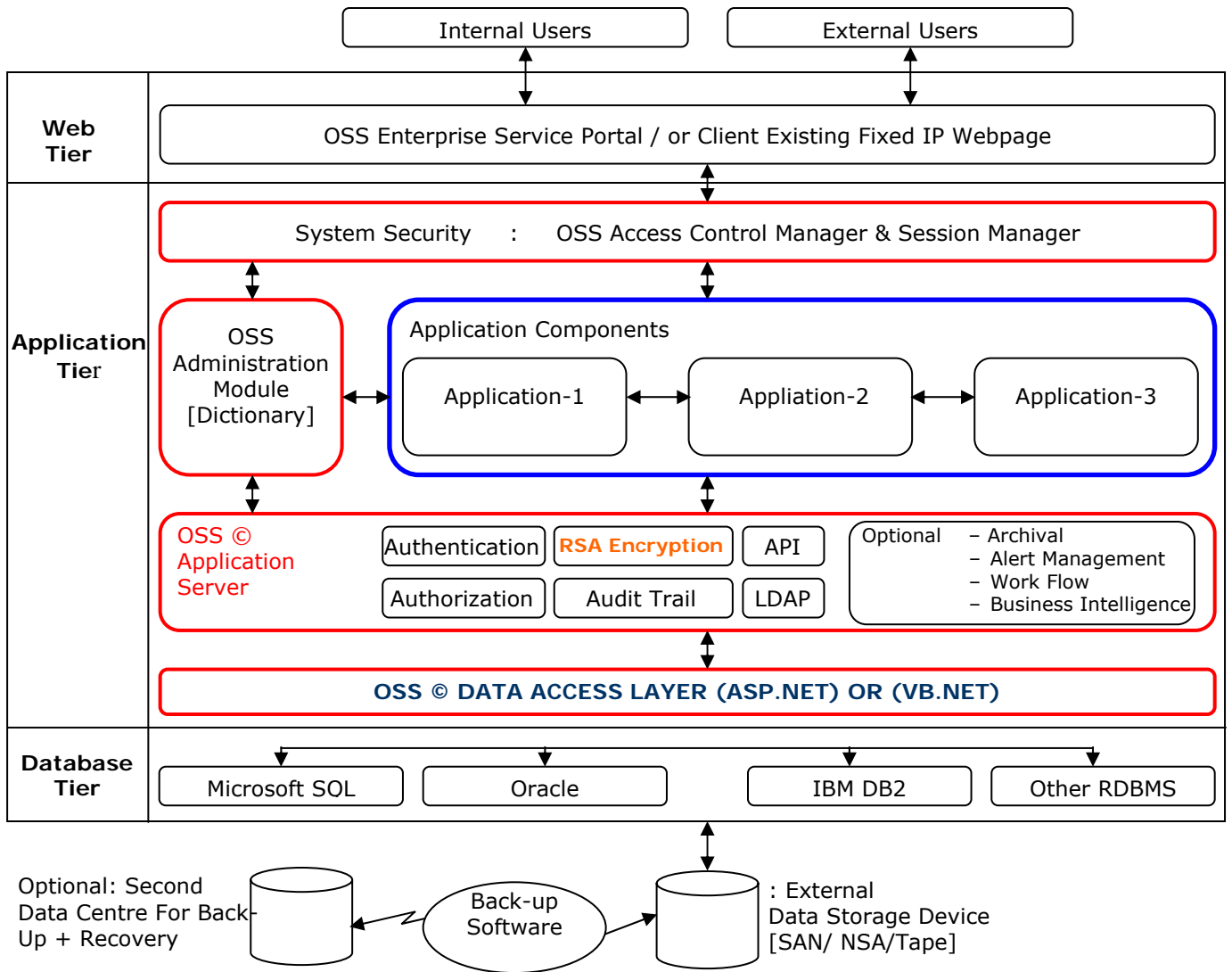
3.1 APPLICATION ARCHITECTURE

The Application Software's architecture is based on **SOA (Service Orient Architecture)**, which means the users (Application Developers) can embed it at any of their fields of application to invoke such invocation and action.

The diagram below (**Figure-1**) illustrates the application's architectural topology where it is part of the middleware that is stacked up at the Application Servers (in Cluster-server environment) that offers its services to the above core Business Applications (such as ERP, Accounting, and Human Resources etc).

For application (or code) development environment, the developers can insert it as one of the syntax in their application and it can be invoked by a simple end-user's mouth click.

Figure-1: Architectural Topology



3.2 APPLICATION FEATUTRE DESCRIPTION

3.2.1 FUNDAMENTAL

Operation

RSA involves two keys: a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct large random [prime numbers](#) p and q

2. Compute $n = pq$

Note: n = is used as the [modulus](#) for both the public and private keys

3. Compute the [totient](#): $\phi(n) = (p - 1)(q - 1)$

4. Choose an integer e such that $1 < e < \phi(n)$, and e and $\phi(n)$ share no factors other than 1 (coprime)

Note: e is released as the public key exponent.

5. Compute d to satisfy the [congruence relation](#) $de = 1 + k\phi(n)$ for some integer k .

Note: d is kept as the private key exponent

Notes on the above steps:

- Step 1: Numbers can be [probabilistically tested](#) for primality.
- Step 3: changed in PKCS#1 v2.0 $\lambda(n) = \text{lcm}(p - 1, q - 1)$ instead of $\phi(n) = (p - 1)(q - 1)$.
- Step 4: A popular choice for the public exponents is $e = 2^{16} + 1 = 65537$. Some applications choose smaller values such as $e = 3, 5, \text{ or } 35$ instead. This is done to make encryption and signature verification faster on small devices like smart cards but small public exponents may lead to greater security risks.
- Steps 4 and 5 can be performed with the [extended Euclidean algorithm](#); see [modular arithmetic](#).

The **public key** consists of the modulus n and the public (or encryption) exponent e . The **private key** consists of the modulus n and the private (or decryption) exponent d which must be kept secret.

- For efficiency a different form of the **private key** can be stored:

p and q : the primes from the key generation,

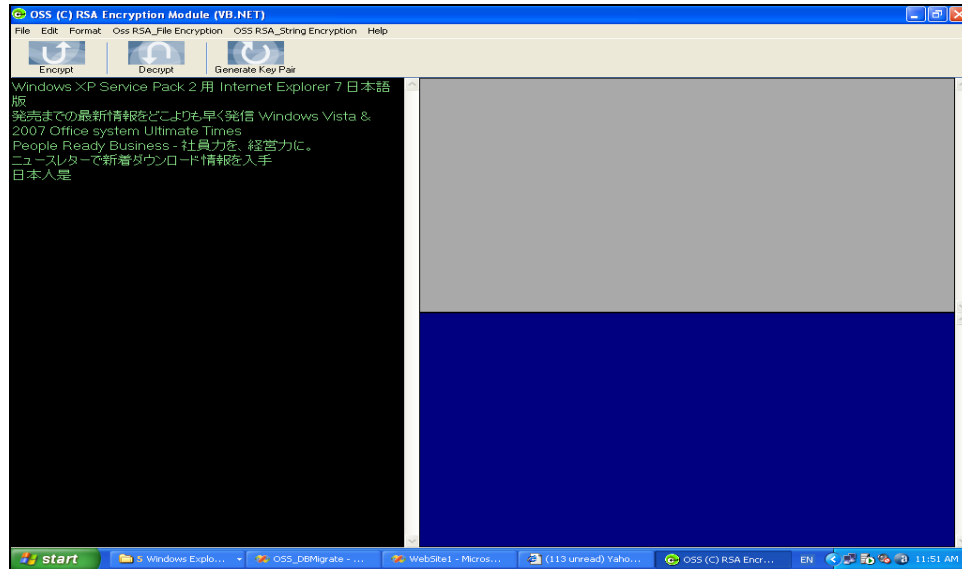
$d \bmod (p - 1)$ and $d \bmod (q - 1)$: often called d_{mp1} and d_{mq1} .

$q^{-1} \bmod (p)$: often called *iqmp*

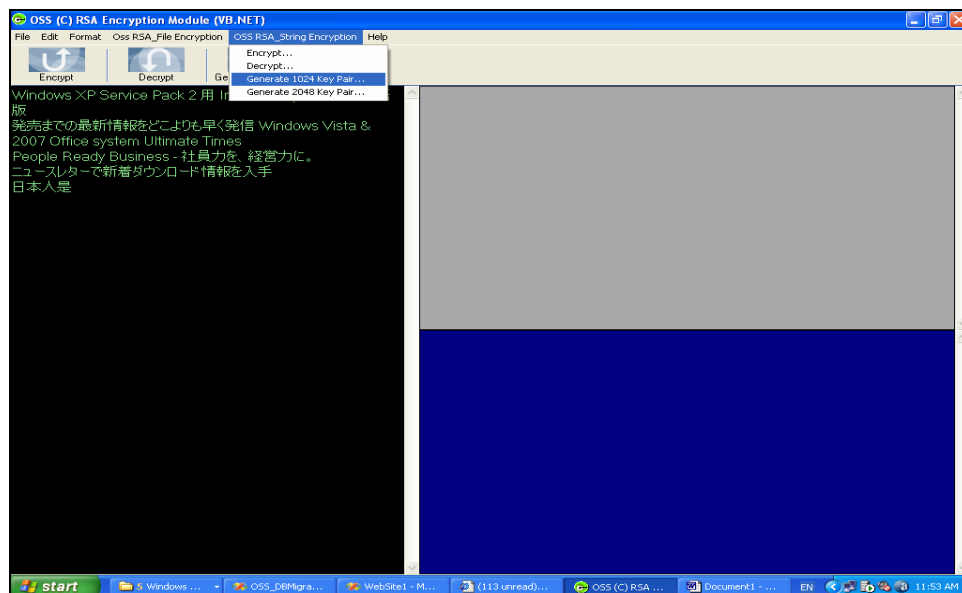
- All parts of the private key must be kept secret in this form. p and q are sensitive since they are the factors of n , and allow computation of d given e . If p and a q are not stored in this form of the private key then they are securely deleted along with other intermediate values from key generation.
- Although this form allows faster decryption and signing by using the [Chinese Remainder Theorem](#), it is considerably less secure since it enables [side channel attacks](#). This is a particular problem if implemented on [smart cards](#), which benefit most from the improved efficiency. (Start with $y = x^e \bmod n$ and let the card decrypt that. So it computes $y^d \bmod p$ or $y^d \bmod q$ whose results give some value z . Now, induce an error in one of the computations. Then $\gcd(z - x, n)$ will reveal p or q).

3.2.2 SCREEN SHOT EXAMPLES of APPLYING OSS © RSA Encryption Module (VB.NET)

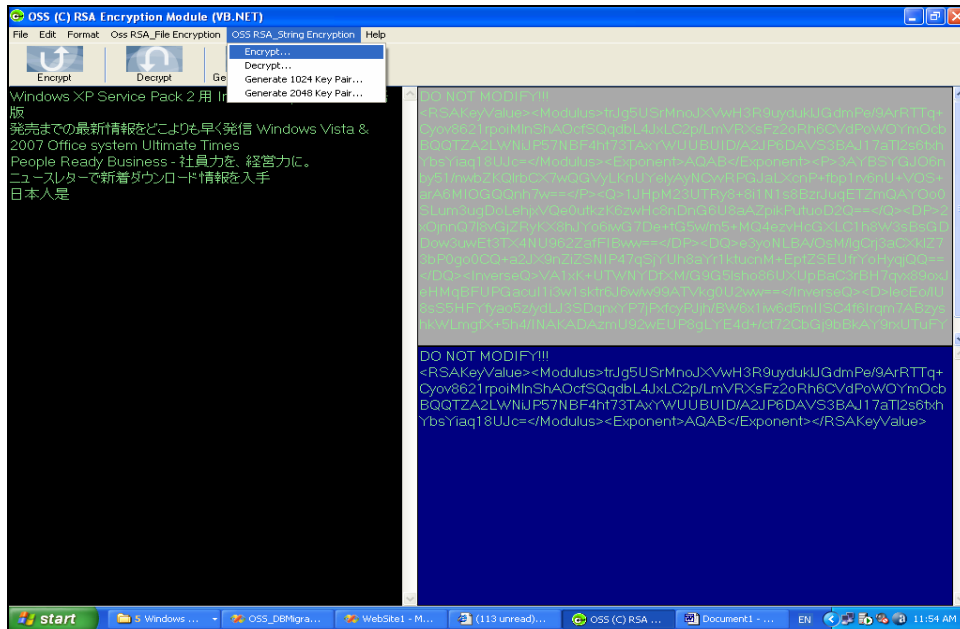
1. **Original Test Message** (contained English, Japanese and Chinese characters) prior to being encrypted- is being showed at the left-hand panel when it is before encrypted with OSS © RSA Encryption Module (VB.NET).



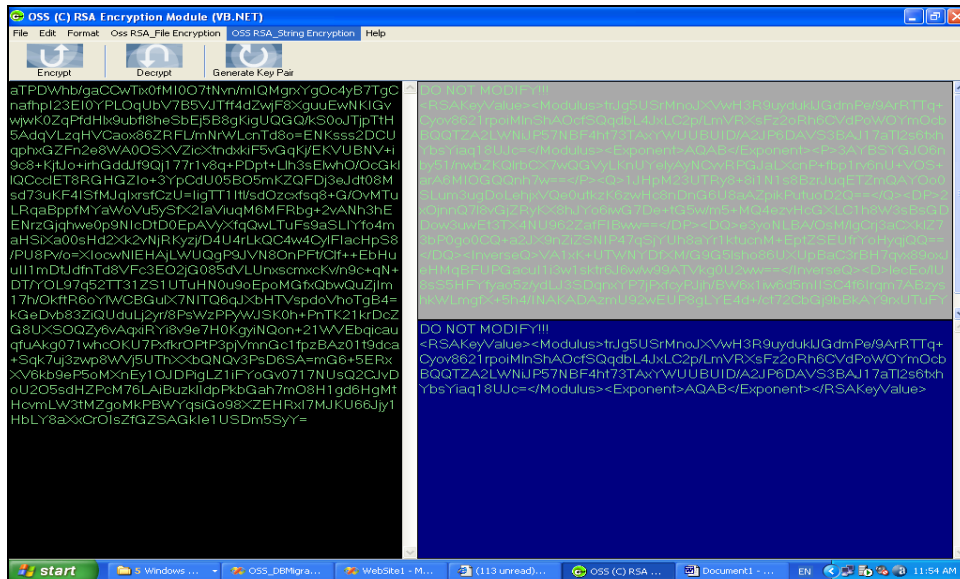
2. **Encryption Algorithm:** User is provided with two (2) types of encryption density: 128 bit or 2046 bits type



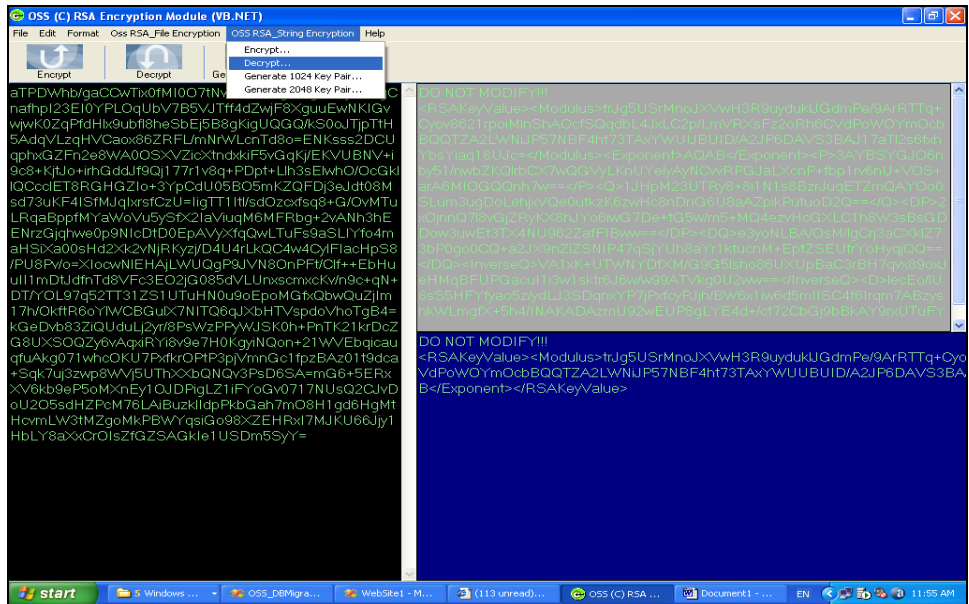
3. **Choose 128-bit Encryption Algorithm:** The encryption script, before being applied to the original text, is showed at the right-hand panel at the screen below.



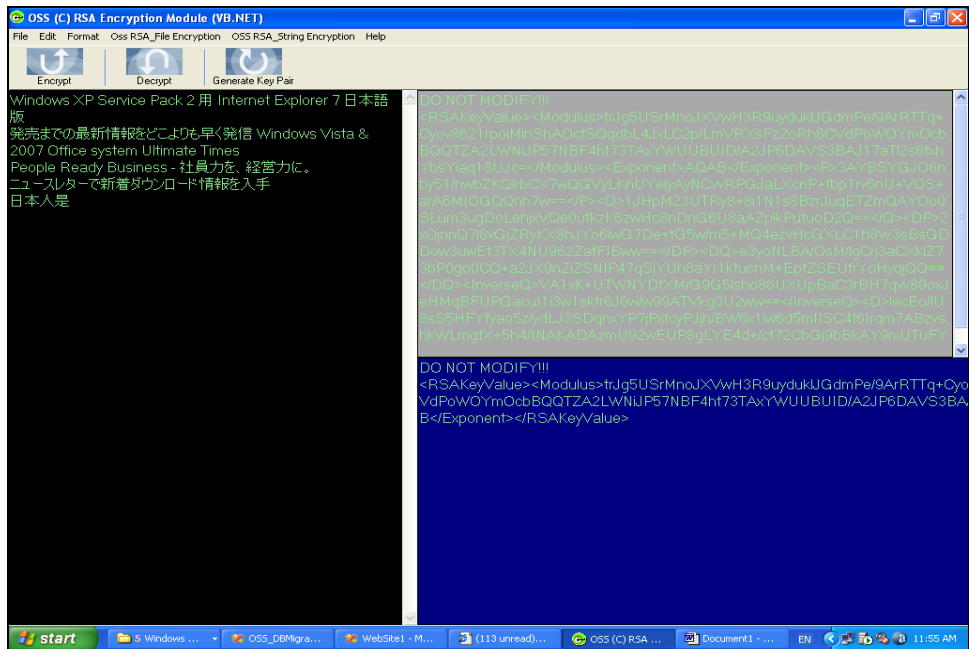
4. **Applied the 128 bit encryption to the original text:** The result text – mixture between original text and encryption scripts – is showed at left hand panel.



- Decryption Process:** This screen shot shows the user is intending to use decryption to return the original text when he/she receives the encrypted text.



- Revert back to the Original Text** (English, Japanese and Chinese characters) after decrypted- is being restored after decrypted, as showed at the left-hand panel.



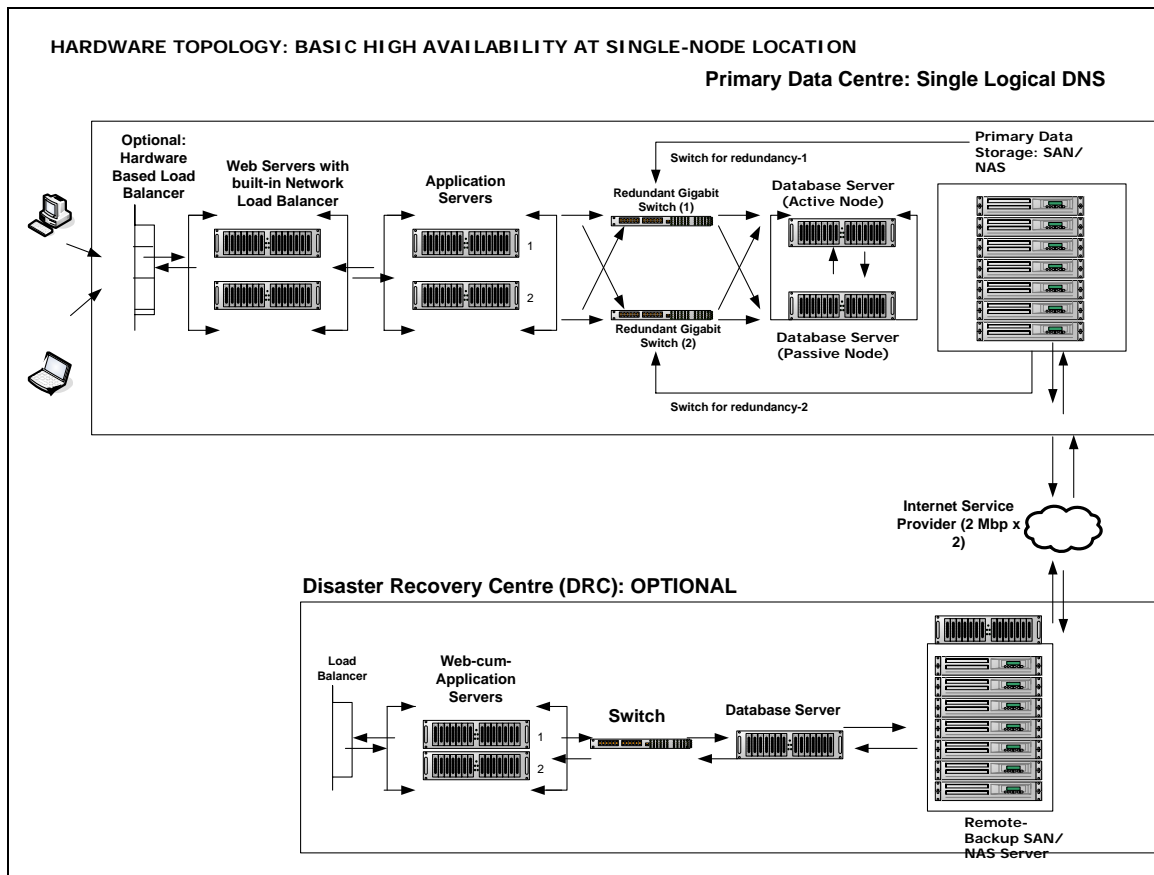
4. HARDWARE CONFIGURATIONS

HARDWARE DEPLOYMENT BASED ON HA (High Availability)

As the application is architected and designed in N-tier, it supports tier based hardware deployment (or Failover and Scalability).

In this diagram below (**Figure 4.1**) of the hardware configuration, we show how the Application Developers can deploy their N-tier based applications couple with our RSA Encryption Services together, located at Application Servers tier, in a cluster-server environment. The cluster-server can be structured with web, application and database servers.

Diagram 4.1



[Open Spectrum Solution]

1. The above information is correct at the time of this article went to print and release on the OSS website. OSS reserves the absolute right to alter and change any of them at any given time without notifying the installed clienteles;
2. For most updated information on the said application, please contract your nearest OSS authorized resellers or logon to www.open-spec.com for contact; and
3. For reporting of error and mistakes at the above article's, please send your message to Documentation@open-spec.com.