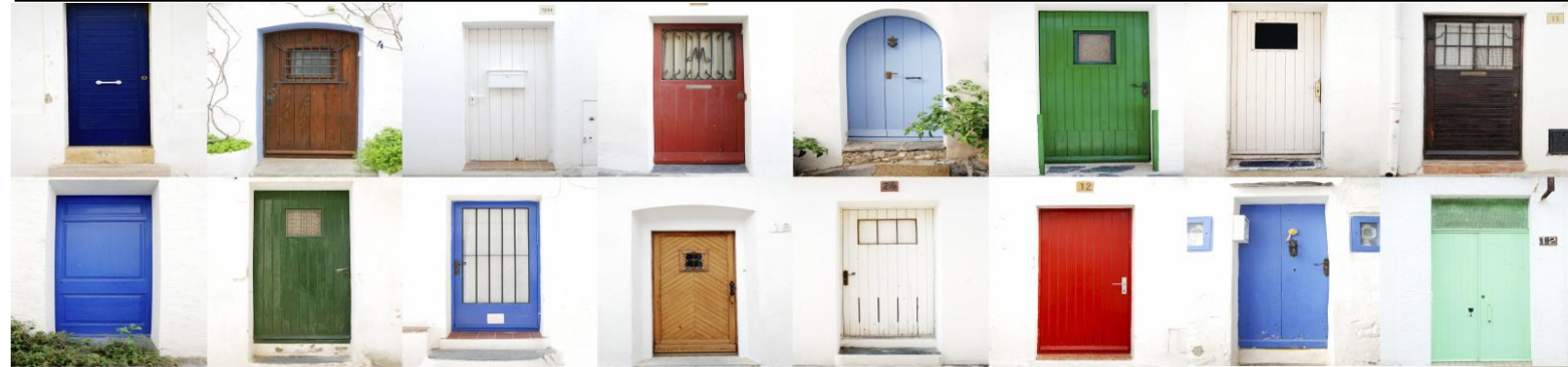


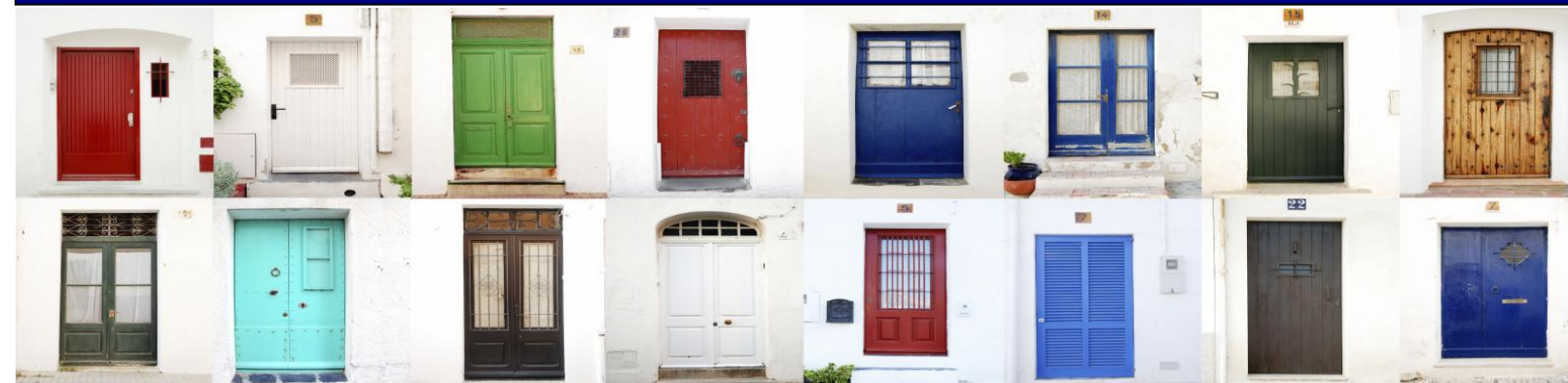
*Opening Doors Around the World...*



Thought Leadership Paper from  
Jeitosa Group International

Software-as-a-Service:  
The HCM Perspective

By John Macy



*...Making Global Business Possible!*

PROPRIETARY

Copyright © 2009 IHRIM and Jeitosa Group International, LLC

All rights reserved. This white paper and the information contained herein are protected by the federal copyright and other applicable laws and regulations. No portion of this paper may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from Jeitosa Group International, LLC.



## Table of Contents

<b>INTRODUCTION.....</b>	<b>4</b>
<b>BACKGROUND TO THE SAAS MODEL .....</b>	<b>4</b>
<b>THE SAAS MODEL .....</b>	<b>5</b>
<b>SAAS ENABLING TECHNOLOGY: SOA .....</b>	<b>6</b>
<b>THE EVOLUTION OF THE SAAS MODEL .....</b>	<b>6</b>
<b>HOSTED SAAS MODELS .....</b>	<b>8</b>
<b>MULTI-TENANT SAAS.....</b>	<b>11</b>
<b>THE HCM BUSINESS CASE FOR SAAS .....</b>	<b>12</b>
<b>QUESTIONS TO ASK A POTENTIAL SAAS VENDOR.....</b>	<b>12</b>
<b>CURRENT BARRIERS TO SAAS ADOPTION .....</b>	<b>13</b>
<b>CONCLUSION .....</b>	<b>17</b>
<b>ABOUT THE AUTHOR.....</b>	<b>18</b>
<b>ABOUT JEITOSA .....</b>	<b>19</b>

# **SOFTWARE-AS-A-SERVICE: THE HCM PERSPECTIVE**

***By John Macy, Jeitosa Group International***

## **INTRODUCTION**

Not since the appearance of Client/Server technology in the early 1990s has the Human Resource (HR) business community experienced such marketing hype surrounding a new technology as we have seen recently with Software-as-a-Service (SaaS). From an IT perspective SaaS is a distributed computing solution and from a business perspective it is a hosted application delivery solution. It is a revolutionary concept that has changed the enterprise perception about computing economics and created a new business model for the purchase and utilization of business applications. As the world sinks into a global recession all forms of business expenditure comes under close scrutiny, and the SaaS business model is a perfect way to reduce Information Technology (IT) costs and help steer companies through the recession period. Although recession-friendly SaaS still has two major barriers to overcome before there is universal adoption of the SaaS concept. They are security and integration. This article will describe the SaaS model and identify why security is a customer concern and outline the integration problems associated Human Capital Management (HCM) software applications.

## **BACKGROUND TO THE SAAS MODEL**

HCM applications have been traditionally hosted on a company's own infrastructure on the company's premises. Companies purchased a license to use the software and paid an annual maintenance fee of around 20 percent of the license price to receive upgrades and support. Companies paid to implement the HCM and for hardware maintenance, operating systems and networking were all maintained by the company's IT resources or contractors.

The arrival of SaaS changed the traditional HCM delivery landscape. Customer Relations Management was the first business domain to sound the SaaS trumpet, heralding the arrival of Salesforce.com, but Human Capital

Management (HCM) solutions, led by the company Workday, were not far behind. Salesforce.com is credited with packaging and branding Software-as-a-Service as a commercial option for hosting software. The enabling features of Web 2.0, including Web Services, SOA (Services Oriented Architecture) and the Internet connected infrastructure service platforms for storage and development (now metaphorically referred to as "clouds"), suited the SaaS model perfectly and HCM solutions were an ideal fit. Attractive as it may be, and seemingly a vast improvement on the on-premise model, SaaS is just another step towards a new computing ecosystem and needs to be kept in perspective.

## **THE SAAS MODEL**

Software-as-a-Service is a software distribution model that has evolved from previous hosted software models such as ASP (Application Service Provider) and plays a prominent role in the new on-demand environment for business applications. It is part of new family of "services" making their appearance in the marketplace such as Hardware-as-a-Service, Integration-as-a-Service, and Platform-as-a-Service. SaaS has become a very popular option with business because it leverages an economy of scale by hosting multiple clients and can provide better, cheaper and more reliable applications than individual company owned data centers are able to match.

The remote distributed application platform credited with enabling SaaS applications is referred to as "cloud" computing. A "cloud" is a remote data center connected to the Internet with all the computing capacity that is expected from a 2009 operation that enables a hosting service for data or software or all of infrastructure. The physical location of the data center is not important unless it affects performance. From an IT perspective it is the logical evolution of network computing and resource provisioning. It represents the best use of resources and the most economical way of making computing power available to the enterprise.

In the future companies will have the option of maintaining IT resources or outsourcing the operation to a hosting company. Companies will have the option to rent a private "cloud" or use one of the many public "clouds". As we look at the needs of HCM we can identify several "clouds" that will be in our HCM stratosphere. There will be a "cloud" for data hosting, application services, possible business intelligence and business rules.

## SAAS ENABLING TECHNOLOGY: SOA

Software-as-a-Service has been enabled through the maturity of several technologies collectively referred to as Web 2.0. They include Web Services and Services Oriented Architecture (SOA). The diagram in Figure 1 illustrates the relationship of the components that make up the architecture.

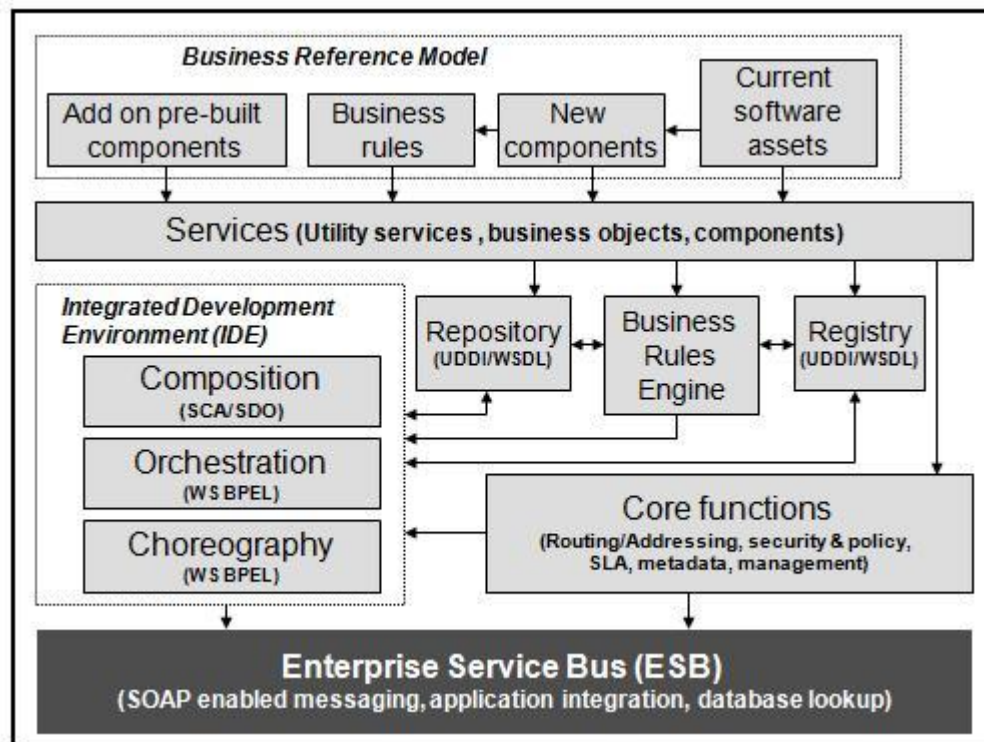


Figure 1. *Services Oriented Architecture*

The IDE (Integrated Development Environment) facilitates the orchestration of workflow processes, the Enterprise Services Bus (ESB) facilitates the integration of third-party applications. SOAP (Simple Object Access Protocol) is the protocol that allows communication between all of the components of SaaS. The business rules engine, component repository and registry are the under-utilized features of SOA that will play a big part in the next phase of SaaS evolution.

## THE EVOLUTION OF THE SAAS MODEL

SaaS emerged from, and improved on, previous hosting models such as Application Service Provider (ASP). The ASP model lost popularity because

the ASP organizations often used an HCM application they purchased on a perpetual license basis from a vendor and did not develop the application themselves. They took an “off-the-shelf” HCM application and used that for customers. The deployment was usually a single instance for each customer on a server in the same way a customer would deploy the application on their own premise. The problem ASPs encountered was that the software they were hosting was not written for the web platform and some legacy systems used were antiquated and not net-native designed for access by web browsers. The response times were often slow and upgrades incurred the same difficulties as internally hosted systems. Customization was out of the question and each client used the same off-the-shelf “vanilla” version of the application.

With the SaaS model the software is owned by the provider and is made available to clients via the Internet for a subscription fee based on number of users, employees or usage rates. The model uses a multi-tenant architecture to achieve maximum resource usage, computing efficiency and economy of scale. The vendor takes the responsibility for delivering regular upgrades and continuous enhancements to all clients by using a single instance of the software application. The SaaS model also offers cost of ownership benefits over the traditional on-premise or perpetual license model by spreading costs over the life of the contract and avoiding the early upfront cost for a license payment. It offers faster deployment by having the necessary infrastructure, already in place, and provides more flexible usage of internal IT resources, avoids costs for hardware and infrastructure, reduces the cost of internal maintenance staff, reduces risk, improves productivity and business processes and establishes a close vendor relationship for delivery of services and accountability. In practice, the cost of applications, hardware and people services can be operated more cheaply by a SaaS vendor.

SaaS applies a multi-tenant architecture to enable clients to share the same hardware and computing resources. Clients all use the same instance of the software application but, depending on the SaaS vendor, can have different configurations according to the need for application extension and dedicated hardware for their database.

In the SaaS context an “instance” of software is the implementation of a default system containing the application code and database. Each client uses the same application code to generate screens and perform transaction processing. Clients may configure their own business rules to govern data

validation and drive workflow functions. The default HCM system is usually proprietary and has been developed for the SaaS vendor's usage and contains functionality the vendor wishes to deliver.

## HOSTED SAAS MODELS

The following are all variations of a hosted service model. There are several options for configuring a SaaS hosted service for a client. On one end of the extreme range is an isolated service and on the other is the shared resources option. The details of the default system are reflected in a diagram known as the schema. The Wikipedia description of a schema is a "diagram that represents the elements of a system" a schema is also used to represent relationships in a data model and when used for XML documents it defines structure and content. The most popular options are:

### **Option ONE: Separate Databases, Separate Schemas.**

The physical positioning of each client's database on a separate server and providing each client with their own application code, based on the same version of the default system, is the easiest and safest way of ensuring data isolation for security and providing flexibility.

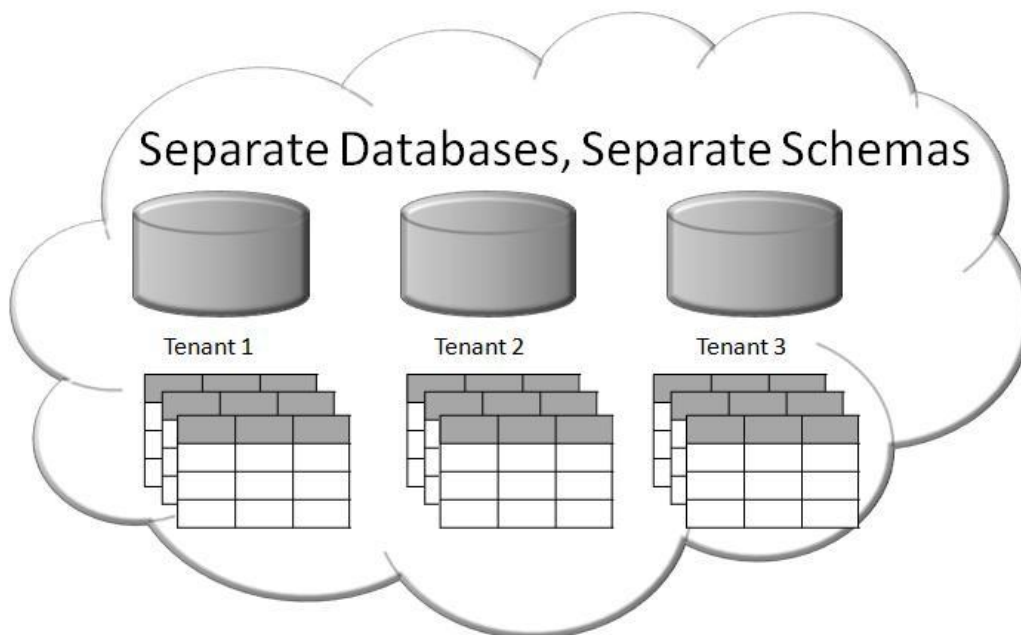


Figure 2. *Separate Database, Separate Schemas* diagram based on Microsoft's *Multi-Tenant Data Architecture* description

Each tenant has their own database and data model (Schema). The support of data is easy and backups and recovery is simple compared to other options. Hardware costs are higher and a client is expected to pay more. Customization is easy because each client is able to extend the data model to suit their own purposes without disturbing other clients' system operation and available functionality. This model suits some organizations requiring high data security, such as banking or storing medical records or companies wishing to extend the functionality of their default system.

### Option TWO: Shared Database, Separate Schemas.

This option provides each client with their own set of tables within the database based on a separate schema specifically designed for them.

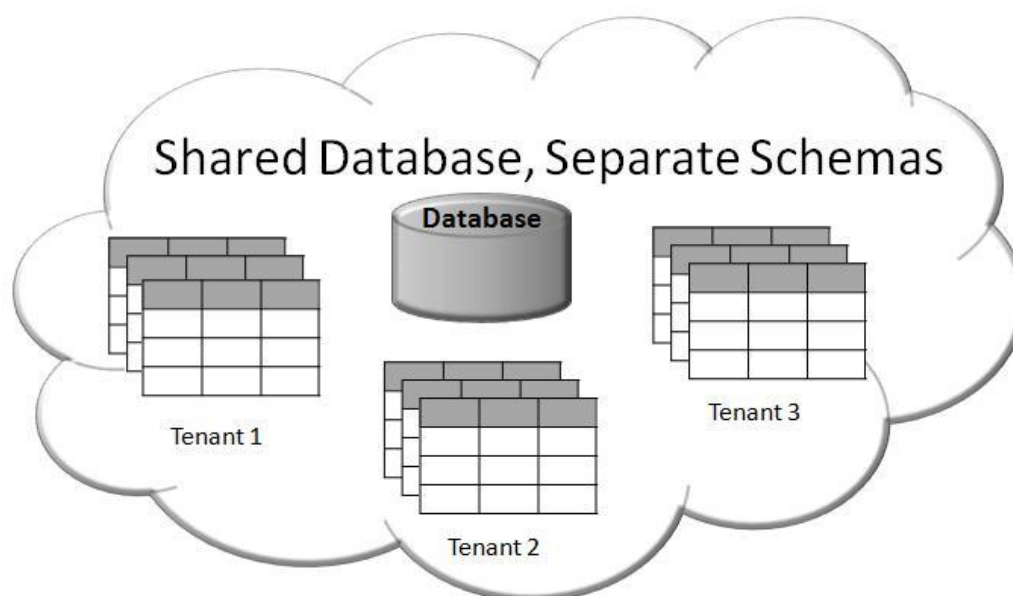


Figure 3. *Shared Database, Separate Schemas diagram based on Microsoft's Multi-Tenant Data Architecture description*

That means their data physically resides on a server with other clients using the same database but they have their own set of tables. Tables are rows and columns. The columns represent fields and the rows represent records. Obviously this extends the size of the database because each client has their own set of columns grouped for their exclusive use and separated for security access. They do not have the same level of security as the isolated database but it is acceptable to most clients. The client has some customization flexibility because they can extend the data model to add columns (fields) to store data specific to their needs and are not bound by the default table structure. In the event of system failure restoration is

harder because it means overwriting data with backup data for every client on the same database irrespective of whether the system failure affected every client or just one. Costs for maintaining the shared database is lower and that can be passed onto the clients.

### Option THREE: Shared Database, Shared Schema.

This model provides the same database and the same schema for all clients. Client data is separated using a client ID column (field).

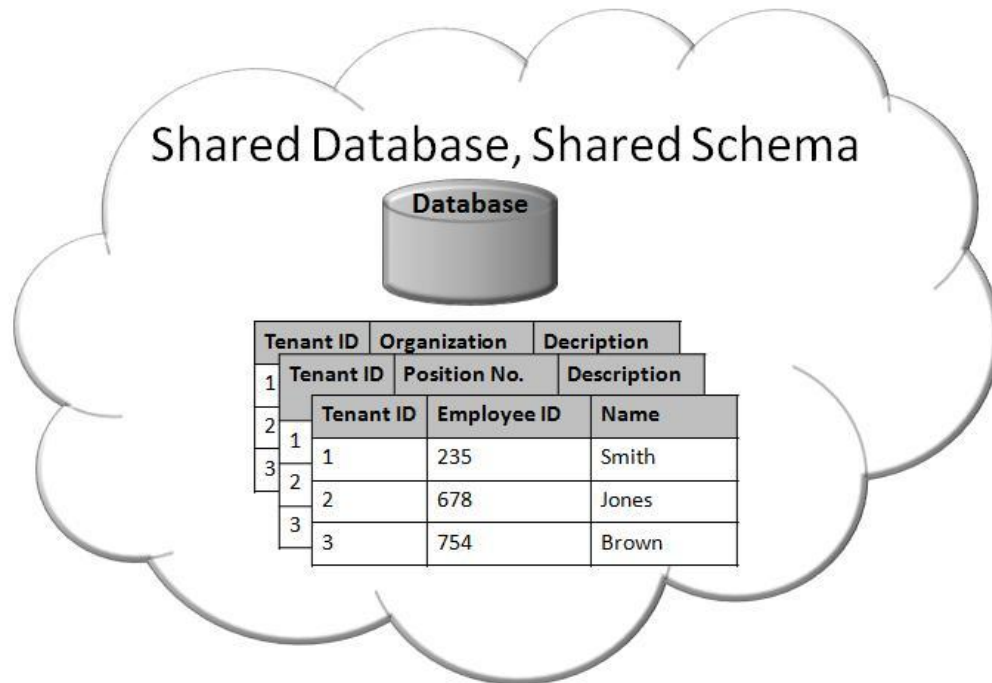


Figure 4. *Shared Database, Shared Schema diagram based on Microsoft's Multi-Tenant Data Architecture description*

Records can be stored in any order, so multiple client data is stored in sequential rows but isolated by virtue of their client ID and can only be accessed by users permitted to use that client ID. It is a logical separation rather than a physical separation of data. Isolation of client records can be reinforced through mechanisms such as permissions, SQL views and encryption. This model has the lowest cost to maintain because it is able to backup and restore data and house multiple clients on the same physical server, thereby minimizing hardware and operating costs. Customization of the schema is not possible because any changes made to the schema affect all clients. The application can only be extended if the vendor provides spare fields for individual use or publishes an API (Application Program Interface)

that allows clients to plug in third-party applications for their own exclusive use, but external to the resources shared by other clients.

### **Choice of SaaS Model**

From a cost viewpoint Option 3, the Shared Database, Shared Schema model takes longer to configure and set up but is it cheaper to operate and ongoing charges will be less. The isolated model may be faster to configure and implement but the additional expenditure required by the vendor to operate an individual environment will be passed back to the client and the special needs associated with an individual operation will mean higher ongoing costs.

From a security viewpoint a properly designed shared database can offer just as high a level of security as an isolated model. The Service Level Agreement (SLA) and the vendor's own reputation provide a guarantee that should satisfy most companies.

The choice of model is usually determined by the number of records, size of the storage required and usage rates. However, regulatory requirements for data protection and skill level of the selected vendor will help determine the most appropriate model. Because multi-tenant single instance architecture is a relatively new area of technology, the necessary skills to maintain a shared model may come into play. Perhaps the isolated model, which is easier to configure and maintain, may be the safest model for some. In many cases the vendor may not offer a choice, and client must accept the option available or seek another vendor.

## **MULTI-TENANT SAAS**

True SaaS must be based on a multi-tenant model. However, many SaaS vendors claim their product to be multi-tenant. The situation is similar to when client / server technology emerged, and every vendor's product suddenly became "client / server". It was not until Gartner published an industry accepted client / server diagram that non-technical people were able to identify what flavor of client / server they were evaluating. It also helped when we progressed to terms like thin client and fat client: we could go back to the diagram to see where thin was. Now with multi-tenant databases we are not sure. Even Oracle claim multi-tenancy for their E-Business Suite product, however, some analysts would query that classification as their community of ISV (Independent Software Vendors) may apply different database structures. For the HR business community we do not want to get into debates about what is true multi-tenant and how

that relates to separate pods in a database or to delve into SAP's Business By Design and see if they have gone down the single instance road. I prefer to use Microsoft's classifications based on database and schemas.

Some SaaS applications in use by commercial vendors are not purpose-built for the multi-tenant environment. It requires investment to take software from a single instance application where each client has their own software to an architecture where many clients run off the same instance. The efficiencies of running multi-clients off the same instance can only be achieved by re-engineering the code of the current system or building a new application from the ground up, as some companies have done.

## **THE HCM BUSINESS CASE FOR SAAS**

The annual cost to maintain software can be up to four times the cost of the initial purchase. It is not surprising that a large proportion of IT budgets go on maintenance rather than proactively addressing changing business requirements with supporting technology.

According to the Software-as-a-Service Executive Council of the SIIA (Software & Information Industry Association) key arguments in favor of SaaS applications include making the IT budget go further by managing the cost allocation for software, hardware and people (or professional) services, leveraging SaaS economies of scale, and avoiding the underestimation of people services costs.

Specific benefits that flow from SaaS are lower costs, faster deployment, less financial risk, higher reliability and more frequent upgrades. When taking into account all factors associated with an on-premise model such as license costs, support, upgrades, database servers, application servers, implementation, training, etc., and comparing costs to a SaaS subscription model for a mid-size business, the savings can be more than half million dollars over a five year period.

## **QUESTIONS TO ASK A POTENTIAL SAAS VENDOR**

Before entering into an arrangement with a SaaS vendor there should be many questions asked about the type of service they provide and the underlying architecture. Questions include:

- **Billing Arrangements:** Are they based on usage rates or active employee records. If payroll is part of the service offering the vendor will have access to active record information.
- **Vendor Suitability:** How long has the vendor been in business, how much experience have they had in the SaaS space, and who are their major clients. During the recession period ahead there will be casualties and there will be some important indicators in the marketplace.
- **Growth Capability:** If your business grows is the service scalable to accommodate future expansion?
- **Type of Architecture:** Is the application fully Web-enabled? If any of the software needs to be installed on your premise, it is probably not the kind of SaaS solution you are looking for.
- **Data Center Credentials:** Check the data center to establish who is hosting the service and check their reliability and continuity record. Check how often they perform upgrades to the default system and what their future strategy is.
- **Security:** What type of security do they use? It is critical to establish that there is 24 by 7 physical security of the data center's premises and the type of security, such as onsite guards. What the type of data security provided, such as encryption standards. Whether the vendor does background checks on the data center operators. What type of application and database configuration is used and what would be the physical and logical separation of your data.
- **Integration Capabilities:** What form of integration does the SaaS vendor offer? What form of cross firewall access is possible for third-party real time access or regularity of application programming interface updates? Does the vendor provide an open software application platform for independent software vendors to build add-on solutions to the vendor's default system?

## CURRENT BARRIERS TO SAAS ADOPTION

The two areas that customers feel the most concern for are security and integration. Almost all surveys identify these two items as the most important considerations when examining the SaaS option.

### **Security Concerns**

The concerns about data security are usually unfounded: SaaS providers are far more conscious of client data security than most internal IT departments. That is because data security is one of their core products and any breach of security could mean commercial disaster. Furthermore data security is the subject of Service Level Agreement focus.

In a database environment that has multiple client data on the same physical server it is hardly surprising that customers may wish to receive added assurance. Even then some companies and organizations such as

investment brokers, defense departments, banks and companies with large client databases may still opt for their own private server and a private “cloud”.

Confidence in data security means improved chances of customer retention: The ongoing financial viability of SaaS vendors is dependent on customer retention. Data security is the life-blood of SaaS vendors. Valid questions that should be addressed to any potential SaaS supplier deal with how safe your data will be. Also, in the event of a disaster, how long does it take to recover and restore operations. The continuity of data storage services is another critical item, and the SaaS vendor’s ability to survive a downturn in the market should be considered. If a company goes under what happens to your data?

### ***Integration Issues***

No single vendor can ever provide all of the functionality needed by HCM systems, especially if a global operation exists. Integration of additional products will always remain an important part of an enterprise’s system architecture. There are several approaches SaaS vendors can take to extend the functional capability of the product, such as:

- Salesforce.com offers a development platform for their CRM application and encourages a community of developers to build composite applications to extend the default system hosting clients in a fully integrated environment.
- Some vendors use the capabilities of Web Services to integrate software services (or components) from ISVs (Independent Software Vendors).
- Some vendors publish an API (Application Program Interface) for third-party products to map to.

In the case of HCM SaaS systems today, the options mentioned are either not available or less than a satisfactory solution.

Salesforce.com provide a market for HCM third-party products in their AppExchange but the products listed do not share the same default system design or instance of the software hosted. Similarly the SaaS Showplace online marketplace lists numerous HCM products that offer a SaaS delivery model but are still based on individual and proprietary schemas and do not readily integrate with other SaaS products in an open environment. Companies listed include Authoria, ADP, iCIMS, Interview Exchange, Journyx, Kronos, SuccessFactors, Taleo, UnicornHRO, Workday and most rely on an API integration and translation of data elements to align with partner products.

From a cost and system management viewpoint when introducing third-party SaaS applications into the architecture it will mean multiple SLAs (Service Level Agreements), multiple databases and multiple pricing commitments. Technically, applications will have a mismatch of data models, reside on different “clouds”, possibly duplicate data and probably have incompatible data structures requiring application integration for consolidated reporting, as in illustrated in Figure 5.

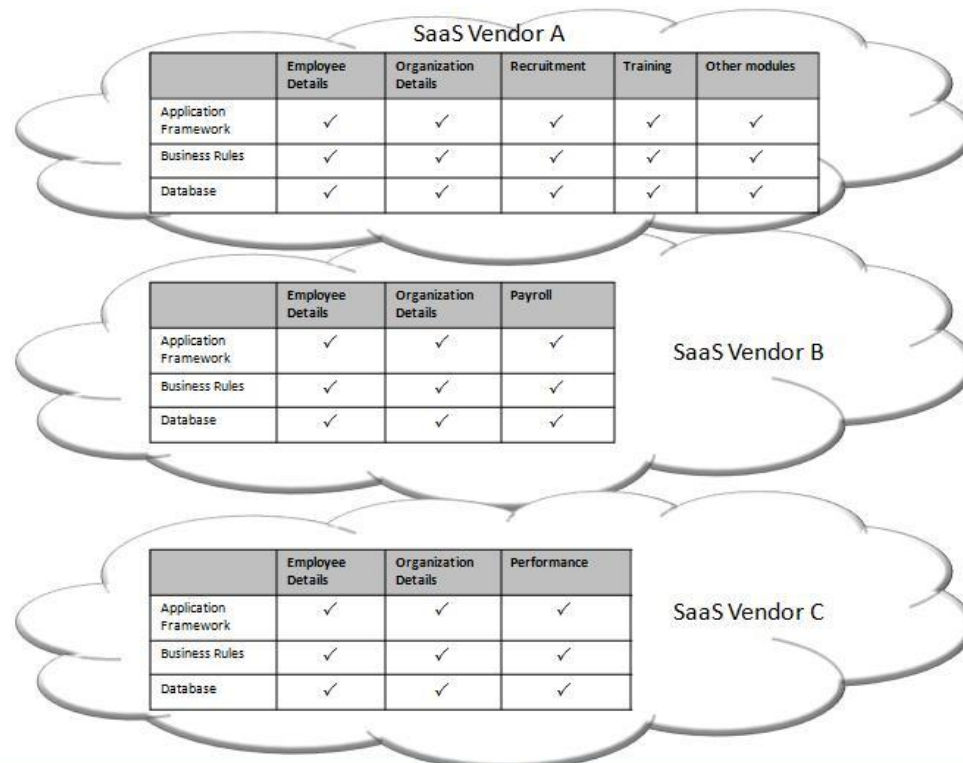


Figure 5. *Third-Party Integration: Third-party products to be integrated reside on multiple “clouds”.*

Each vendor hosts the same components such as the application framework, business rules and database. Each product will have its own Employee and Organization data defined to its own database and no simple integration is possible. Data must be translated from one source to another before reporting can occur.

Apart from the technical issues associated with integration, companies should ensure before committing to multiple vendors that the total cost of ownership has been carefully considered. By the time all of the fees are added up for all of the vendors it may be cheaper to stay with the on-

premise solution on a single instance of the software, using an isolated database and maybe migrating the solution to a private “cloud” to reduce the infrastructure costs. The integration is an important issue for prospective clients of SaaS.

### **Integration Methods**

Over the last five years the method of integration has changed from the on-premise model to the new SaaS or on-demand operation, as indicated in Figure 6. The main difference has been the introduction of Web Services and advances made in middleware technology.

#### **Multiple Layers to the Integration Challenge**

On-Premise		On-Demand
Presentation	↔	User Interface
Middleware	↔	APIs
Data Sources	↔	Data Sources
Operating Systems	↔	Web Services

*And, across multiple SaaS platforms and services.*

Figure 6. *The integration challenge based on THINKstrategies’s model*

To add further complexity to the integration process, integration in the SaaS environment may not always be from “cloud” to “cloud”. It could involve integration with legacy systems residing on the on-premise infrastructure, some of which may be proprietary platforms.

Most techniques for integrating SaaS applications involve buying complex software or writing custom code. Integration can blow out expenses considerably and some companies now prefer to buy integration appliances that are plug-and-play and tend to reduce the complexity and accelerate the implementation process.

Integration methods using APIs (Application Programming Interface) and WSDL (Web Service Definition Language), a feature of SOA, contribute to exchanging data but both have their critics. Unfortunately much published HCM integration success relates more to the flashy mashup type features, such as Google maps and other nice-to-have items, but do not contribute greatly to the broader application integration challenges.

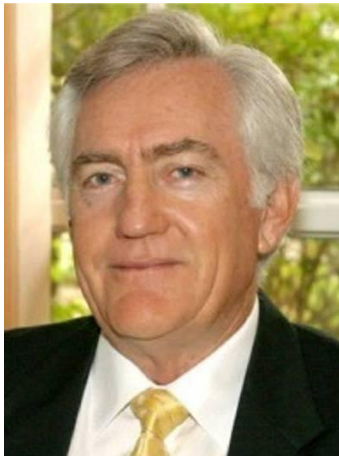
Perhaps one of the most serious companies addressing the integration issue is Workday. Workday acquired Cape Clear to use their ESB (Enterprise Services Bus) as the backbone of their integration strategy and have established four major components to facilitate integration to their SaaS product. They have productized integration services and set up their WIN (Workday Integration Network) and created integration tools for developers. Additionally they have developed the Workday integration appliance for lightweight client-side capability and created the WDN (Workday Developer Network).

## **CONCLUSION**

Many companies have invested heavily in ERP (Enterprise Resource Planning) systems over the last ten years and will be reluctant to change to SaaS. Similarly many large software vendors have spent millions of dollars on their current ERP suite of products and are reluctant to re-invest in the new SaaS model whilst buyers still want their on-premise products. Vendors are also still committed to current clients for support and upgrades. Many current HCM vendors will have to adopt a hybrid model as they migrate to a true SaaS offering and their own multi-tenant platforms.

The Service Ecosystem (SE) that SaaS is part of is gaining strength daily boosted by the poor economic outlook for 2009 and the need for alternate on-premise delivery models. Many companies wish to defer the capital investment needed for the on-premise model and minimize operating costs. Minimizing IT costs is no longer a purely competitive factor but a survival necessity. However, before making any major decision in the current environment to convert from perpetual licensed software to Software-as-a-Service, HR leaders should develop a rigorous business case and total of cost ownership analysis to ensure that the decision is the right move for the company and will not adversely disrupt operations or introduce unexpected integration complexities.

## ABOUT THE AUTHOR



**John Macy** is Vice President of Asia Pacific and senior global advisor with Jeitosa Group International ([www.jeitosa.com](http://www.jeitosa.com)), a worldwide strategic business consultancy focused on making global HR possible. He is also founder and president of Competitive Edge Technology ([www.cet-hr.com](http://www.cet-hr.com)), a consulting company formed in 1994 specializing in Human Capital Management software.

Macy has over 35 years experience as a senior HR manager with an international airline and consultant in effective HR technology usage. His particular areas of expertise include business requirements definition, business process design and optimization, business modeling and scenario planning, software selection, and systems implementation and management. His strength is working closely with clients to help improve HR business processes through better use of technology. He has worked throughout United States, Australia, and Asia, with a strong focus on Malaysia, Singapore, Thailand, and China. Some of his clients include Hyatt Corporation, Qantas Airways, Seagate Technologies, United Nations, Lend Lease, Citibank, State Street, Bankers Trust, Curtin University, Thies Contractors.

Macy is an advocate of Services Oriented Architecture (SOA), Web services, component-based applications and business driven standards to achieve greater flexibility and integration. In 2002 he published the HR component software application standard and set up the first component software brokerage. In 2006 he introduced the first commercial component registry for HR products. Macy has written several books and published numerous articles on HR technology. He can be reached at [john.macy@jeitosa.com](mailto:john.macy@jeitosa.com).

## ABOUT JEITOSA

### ***Our Company***

Jeitosa Group International was founded in 2004 as a global advisory and systems deployment firm servicing companies across the world in the areas of human resources, finance, and information technology. Jeitosa is a global organization with 50+ global business advisers covering 40+ countries focusing on the mission critical challenges and issues companies face as they grow — challenges and issues that are local, global, regional, and transnational.

### ***Our Name***

The name Jeitosa (pronounced zhay-taw-za) comes from the Brazilian Portuguese word *jeitosa* [from *jeito* + *-osa*], an adjective meaning able to find astute or ingenious solutions; skillful, adroit, clever, innovative, solution-oriented, problem-solving; making the impossible happen (Dicionário Aurélio). Working in the global, digital, virtual world demands a creative approach in finding elegant solutions to complex business problems, which is why this name aptly describes the kinds of solutions we provide for our clients.

### ***Our Mission***

Our mission is quite simple: *"we bridge people, process, and technology through agile, modern solutions to make global business possible."* Our goal is to be a global partner to our clients, enabling "Better HR" through more adroit and knowledgeable people, more effective and efficient business processes, and best-fit modern technologies.

### ***Our Values***

Jeitosa abides by six core values that govern interactions with all of our associates:

- ***Integrity*** – fostering openness and fairness in all our business relationships.
- ***Motivation*** – going the extra mile to ensure our clients' and partners' success.
- ***Partnership*** – engaging with all our associates for long-term relationships.
- ***Knowledge*** – sharing best-in-class global business and technology solutions.
- ***Excellence*** – delivering superior value to our clients, partners, & shareholders.
- ***Balance*** – building symbiosis in work, life, diversity, profitability, and fun.

***Jeitosa opens doors around the world...  
making global business possible!***

**SAN FRANCISCO**

**NEW YORK**

**TORONTO**

**PARIS**

**FRANKFURT**

**SÃO PAULO**

**SYDNEY**

**CHENNAI**

**DALIAN**

**SHANGHAI**

**TOKYO**

**[WWW.JEITOSA.COM](http://WWW.JEITOSA.COM)**