Convert your web application to a Multi tenant SaaS solution using "SaaS-Tenant™" Framework

An account of directions and steps to speedily convert your web application into a cloud application

Summary

After creating a single-tenant web-enabled application, are you looking forward to make it attuned and efficient in a cloud environment? Are you searching for the steps needed to transform your application to an absolute and perfected multi-tenant, cloud-ready SaaS application? Here we offer simple answers to your questions as we discuss the essential steps to transform your web application to a cloud-winner. We will also display the software that our company has designed to offer a “plug-in” approach to multi-tenancy.

In case you have a web application that you are planning to sell in the market, you will understand that Software as a Service (SaaS) on a cloud infrastructure is essential. But, the challenge lies in converting to SaaS promptly and ably so that it retains your chances of profitability. There exists a wide level of differentiation between a SaaS application and a regular web application that must be taken care of. A few are technical where as others are related to alteration in a business model that a company needs to become accustomed when opting for SaaS.

Typical web application vs. SaaS

The characteristic that defines SaaS is the ability of the consumers to make use of an application that is on a pay-as-you-go subscription basis. Here a license in not required for the installation, hosting and management of the software. The company providing the SaaS application becomes the liability for these.

Multi-tenancy is the key to successful SaaS

A SaaS application is essential to be a multi-tenant application as well. The reasons for this are purely economic. As agreed by the CEOs of leading SaaS companies that a SaaS business cannot grow without multi-tenancy.
Steps to convert web application to SaaS

Seven steps are required to convert web application to a SaaS application:

Step 1. This application must be supported by multi-tenancy.

Step 2. This application must consist some level self-service sign-up.

Step 3. The subscription billing needs to be in order.

Step 4. The application needs to scale effectively.

Step 5. The functions need to be in order monitor, configure, and manage the application and tenants.

Step 6. There needs to be a mechanism in order to support exclusive user identification and authentication.

Step 7. Certain mechanisms needs to be in place to maintain some level of customization for each tenant.

Step 1: Support multi-tenancy

Multi-tenancy is decisive for SaaS effectiveness. An application would support multiple users if the users belong to the same organization. The model given below is a fine example of the pre-SaaS world where an organization purchases an application for the use by its members. But in this age of SaaS a company must allow each member to access the data regarding their company. This potential to have multiple organizations, also referred as tenants exists on the same application without having to compromise on the security level.

There are numerous levels of multi-tenancy (as seen in the figure that follows the list):

1. Simple virtualization in the cloud where only the hardware is shared.
2. Single application with separate databases per tenant.
Multi-tenancy models

Model 1 is not multi-tenancy but promoted as a multi-tenancy and is used in a cloud with visualized servers. It has minor advantages over ASP model. A competent level is one where the application completely shares a database and application business logic. This can be a difficult process as it requires changes to the database and missing a single code can lead to compliance of the data security of the application. Also, in order to keep the tenants available to particular tenants, applications like report writers or utility applications should be customized. This type of access also gives rise to problems that must be kept under a check. An authorized user should be prevented from accessing data that is able to write a report of the tenant of which they don’t belong.

Step 2 : Self-service sign-up

An application needs to have some of the levels of self-service sign up, even if it’s just a request mechanism needed in a business process for the addition of a tenant.

Step 3 : Subscription and billing

Given that SaaS applications require payments due to the application options, usage duration and number of users per tenant, there needs to be a certain kind of mechanism to keep a track and manage the application and produce billing information that is handy for the tenant administrators.
Step 4: Scaling and managing the application

It is important to scale up with the rise in subscriptions. A reasonable way to achieve this is cloud infrastructure as it represents the need for successful scaling. The administration and application management functionality to check, arrange, and direct the application and all the tenants should be provided.

Step 5: User ID and authentication

Since multi-tenancy requires the users to be identified to find out which tenant they belong, a mechanism to sustain user identification and authentication is important. The chief information that is used to restrict the data accessed by the user is the user-to-tenant relationship. The classic way for assurance and identification is Email addresses. Even if the external identification of the user is necessary, it’s the liability of the SaaS application to find out that the recognized user is a member of the tenant they belong.

Step 6: Per-tenant customization

A basic level of customization needs to be provided for every tenant that has a sole URL, landing page, logos, color scheme, fonts, and perhaps even language. Being able to provide customization is a trademark of a good SaaS structural design.

Step 7: Security issues to consider

Majority of the SaaS application structures take data security measures that avoid a tenant from seeing data of another tenant. But:

- A chief capacity that SaaS applications must have is to amalgamate and act together with other applications.
- A few of other applications may be outer applications (not controlled by the SaaS provider).
- Not all SaaS structures are designed with convenience for outside applications.

A good SaaS structure must not only have the data under control but also safe for every tenant.
Flexibility is the key to technological evolution

With the rapid levels of technology change, the foremost cloud providers will bring in new middleware applications and capabilities that will be beneficial for SaaS application. Service oriented manner provides the dexterity and elasticity crucial for SaaS applications. Flexibility is provided due to lose coupling which enables calculated and planned growth.

Automatically SaaS-ify applications with Leo TechnoSoft’s SaaS-Tenant Multi Tenant Server

Leo TechnoSoft’s "SaaS-Tenant™" Multi Tenant Server enables ISVs to go for a different approach through a middleware layer that gives crucial multi-tenancy which preserves the asset in the existing code. The four steps required to convert billing into a multi-tenant SaaS version are:

Step 1. Transformation of a database schema to an abstract model

The typical stack for a web application is shown below. For the transformation of an application to a multi-tenant application, the redesigning of the database needs to be managed which would permit the data be filtered by the tenant. To read the presented database the SaaS application is used. A model for that database is created that has the following additional tables.

A typical application stack for the cloud

- Application
- Application server
- Database
- Infrastructure / Cloud
Step 2. Extention of the user authentication process

A multi-tenant SaaS application is required to have the knack to manage the required session information for the authenticated users since the tenant to which they fit in can be recognized. Many methods for application identification is involved but the conversion should be analyzed and the authentication method should be implied.

Leo TechnoSoft’s SaaS-Tenant Multi-Tenant Server

Leo TechnoSoft’s "SaaS-Tenant™" Multi-Tenant Server makes it convenient for our customers to swiftly transform the prevalent single-tenant web applications to fully multi-tenant SaaS solutions that become well-matched without having to rewrite the application. Leo TechnoSoft offers the highly competent and
affordable lookout to multi-tenancy. Due to the plug-in middleware solution that is offered to multi-tenancy, the method becomes a safe and reasonable solution that helps to lower the costs significantly. Multi-tenant server provides a flexible structure that the customer can choose as to how to position their SaaS application, in-house or on any type of cloud. Doing this enables services to a restricted audience like a big venture with divisions that develop the tenants. Due to this the provision, management and monitoring of the SaaS customers is possible. Leo TechnoSoft’s solutions will definitely help to transform SaaS application faster.

The goal discussed here is to apply a method that will expand the application’s user identification process so that if once a user has logged on, he is then matched with his tenant and that tenant eventually becomes a part of the session information. Various ways are available depending on the implementation of the application. Having a genuine user and knowing its tenant is adequate information to filter the data so that only a particular data belonging to that tenant can be accessed.

Step 3.Configuration of the database connection

Leo TechnoSoft’s tenant server has a service oriented structure that uses a MetaModel Database. It uses an abstraction layer that models the original database of the application and then the application’s database communications are directed to the MetaModel abstraction instead of the actual database implementation. This takes care of the security of the data which is always stringently maintained. Even for an external application the data is restricted to the data for the tenant.

Figure 3.The per-tenant customizations structure
Step 4. Deploy the new multi-tenant SaaS application to the cloud

The "SaaS-Tenant™" can be positioned with a SaaS application to a desired server which also includes servers in the cloud. After selection of the server required for the application and database the target database is generated. Also, the Multi-Tenant Server application is arranged as packaged file to the application server. The application is then able to handle multiple tenants. Yet, the original application doesn’t have an administration and management interface that manages tenants or monitors multi-tenant application.

A companion application, "SaaS-Tenant™" Admin gets installed in the "SaaS-Tenant™", which gives these basic multi-tenant services. This has all the rights to the MetaModel Database as the main application and also has administration screens for provisioning tenants, assigning Tenant Administrator accounts and configuring the basic parameters of the various per-tenant application configurations that are available. Administration facilities for keeping a check on tenants and reporting are also available. SaaS applications have a trait to track tenant subscriptions and billing, services for these are available as well.
Basic levels of deployment for SaaS application are used. The characteristics remain the same even after converting the deployment including the cloud management tools required to create template and transmit instances of the application which is used to arrange an operational structural design that meets the requirement for scalability, elasticity, resiliency, and redundancy. A archetypal SaaS application has a set of application servers that is accessed through a load balancer and connected to a database server. The database server may then be deployed as a cluster with the multiple database servers providing redundancy and scalability.

The inclination for the database availability and fail over is fitting more sophisticated for SaaS applications because the number of customers/tenants affected by any outages is larger than that in any traditional customer premise software (Figure 5). Figure 5.SaaS deployment architecture for scalable resilient cloud operations
Some tenants can be configured to use a more expensive encrypted version of the database if they want more security. They still share the benefit of the elastic cloud scalability for cost effective performance.
Conclusion

The advancement in the IT industry to SaaS is ongoing and has already started to cause some foremost changes in the setting. Cloud computing is mounting at rapid rates far greater than any IT wave and SaaS has been the driver for that growth. This conversion has forced many business organizations to change and build novel ways to deliver services in a cloud-centric IT world. It is imperative especially for the software vendors to comprehend, plan for, and make the changeover or else be left behind. The SaaS-on-a-cloud model is different from the classic software vendor model in both the technical and business fields. These differences make the switch to SaaS a high risk venture for the software vendors. However, with reduced amount of risk involved, the ISVs can now take advantage of the established technologies, products, and partners to lend a hand in this evolution.

Why Choose SaaS-Tenant?

For our partner, SaaS-Tenant is the framework of choice for the considerations of multi-tenancy for a few reasons:

- Multiple web tenant enables a multiple instances of your application to serve thousands of your vendors.
- Single web tenant to serve thousands of your customers.
- Multi-tenant data model designs added which become smart configurations with SaaS Tenant.
- Provide option form a single coupled database, specific database schema per web-tenant or an isolated database per web-tenant.
- Provide Multi-tenant modules for customer on demand.

To learn more about SaaS-Tenant, email us on enquiry@leosys.net
About Us

Leo Technosoft - Cloud Computing R&D Center for product development, IT services and infrastructure management

Leo Technosoft is a hybrid Software Product Development Company based in India, US and UK. We partner with organizations, empowering them to attain cost effective product development in SaaS environment.

Our prime focus is on reducing your company's expenses while delivering you quality services. We have a global presence with offices in India, USA and UK. Our international footprint allows us to meet our client's needs regardless of their location. This has made us a partner-of-choice for small and medium sized companies looking for on-time delivery and high quality product development solutions.

We specialize in outsourced product development and IT services providing our global clients value for their money to meet specific business needs.

Visit our website - http://www.leotechnosoft.net