Migrating Within the Cloud, SaaS to SaaS

A Real World Experience
How do you know when a technology has ‘arrived’? For cloud technology, perhaps it’s when the questions change from “Should I migrate to the cloud?” to “How do I migrate within the cloud?” or “How can I leverage my current cloud, extend it, and expand my business value?” If that is the case, what new challenges are presented to us at that level of maturation?

Collaborative recently led an enterprise systems migration effort for a medical publishing company. Their core publishing system was already in place, implemented on hybrid cloud architecture. They wanted to enhance this existing hybrid cloud architecture to better service the customer base. These enhancements included:

• Migrating the existing core publishing system from one Software as a Service (SaaS) vendor to a new SaaS vendor

• Enhancing existing core systems and ancillary SaaS services in support of the migration

• Extending and integrating the services (both internal and external) in order provide their customers more value and an enriched user experience

This paper will discuss the experiences and challenges that were encountered along the way and the lessons learned when addressing those challenges that should be considered when migrating within the world of the cloud. This paper will focus on one of the key challenges: data migration within the cloud, from one SaaS vendor to a new SaaS vendor.

I. Existing system, a SaaS hybrid model

The publishing system was at the core of the existing system architecture. It was implemented using a SaaS publishing system vendor. Additional components included a number of other key SaaS vendors as well as a variety of internal applications and packages to complete the hybrid implementation.

II. Data migration between SaaS vendors

A colleague of mine is very fond of the statement, “It’s all about the data”. As you may guess, he comes from a data background. His viewpoint is very applicable here – the existing SaaS implementation was the system of record and enforcement
for authentication, authorization and web identity management. The client's user data included:

- 800,000 paid subscribers
- 1.8 million registered users
- 25,000 institution subscriptions

All of which had a wealth of information and attributes that needed to be migrated to the new SaaS implementation. In addition to the user/subscriber data, there was a need to migrate:

- Audit compliant traffic reports
  - Subject to specific compliance audit requirements, more complex than simple log files
- Exam subsystem information which included data such as:
  - Exam answers
  - Exams taken
  - Exam credits, etc.

If you were migrating this data from one database platform to another, the process would be straightforward. You have direct access to the database design and content. The process is well understood.

When you are in the cloud and using a SaaS vendor as part of your architecture, data migration becomes more of a challenge. Your data is in the hands of the SaaS vendor; you do not have direct access to the database or file system. You
must use the tools and support services provided by the vendor as your only way accessing your data. The quality of those tools and that support can make or break your migration efforts.

When migrating from one SaaS vendor to another, the challenge becomes compounded. You no longer have direct access to the data on the source or the destination. You must utilize the tools and support provided by both SaaS vendors to perform the data migration. The task was not insurmountable, and we were able to perform the migration, but it lead to an important lesson learned for the client.

III. When picking a SaaS vendor, think about your backout plan

I know this may sound like planning for failure, but that's not what I am saying. When moving to a SaaS platform, especially one that will contain key business data, you must protect yourself and your data. To do this you should:

• Understand the tools and support the SaaS vendor provides
  - Both for the importing of your data as well as the exporting of the same
• Develop a canonical model of your data within the SaaS system
  - You know your data best, then map it to the vendor import/export tools capabilities
• Test the tools against your model to confirm you can move data in and out of the SaaS system
  - If you have large volumes, performance test as well, understand your load/unload impacts

This not only protects you from vendor lock-in but from unforeseeable events that may occur. One need only see the news recently about Nirvanix, a cloud storage company, abruptly closing its doors to understand. The old adage ‘failing to plan is planning for failure’ is one of the keys to succeeding in the cloud. When talking about your key business data, this is critical to your success.

IV. Selecting solution model

When looking at solution alternatives, there is never a perfect solution; everything is a tradeoff, and you decide which tradeoffs you are willing to accept. Way back in the old days, (okay, maybe not that long ago) you had full control over your environments: what hardware you had, how that hardware would be used, and the deployment for your different environments. All this was completely under your control. Once you enter the world of the cloud and SaaS, one of the tradeoffs involved is the level of environment and deployment control.

The new SaaS vendor the client had chosen provided two different models for utilizing its product. While both models were hosted solution models, there were significant differences in the requirements for deployment topology and implementation approaches.

Option 1: Multi-tenant COTS Model

The first option was a standard Commercial Off The Shelf (COTS) implementation of the SaaS product using a multi-tenant model with shared physical environments and separate application run times for web and application server components for each client. The one exception in this model was the workflow component of the system, which processed batch customer updates and content submissions (an extremely key piece of a publication system architecture). In the multi-tenant model, the workflow server was a shared resource.

In addition to the deployment topology, the environment topology for implementation testing consisted of only a beta (development) and production environment. This would severely constrain the implementation team’s ability to perform key system test, configuration and deployment events. There was additional concern about potentially competing for application availability as well as creating data integrity problems within the environment.

Option 2: Full Customization Model

This model allowed for large scale customization of the SaaS product itself. This included direct customization of the user interface and integration code in the application.
This model also allowed for a private topology deployment of the web, application and workflow components of the system. There were still some limitations around the environment topology, but were workable (more on that below). Given the enhanced customer experience the client wanted to provide their customers, and the environment concerns, they made the choice to go with this option.

V. We have a model, now we needed buy in on an approach

The next key issue was to gain a shared understanding between the client and the SaaS vendor on the development approach for the project. Once that understanding was reached, we would be able to determine what infrastructure was needed to execute the approach. To accomplish this, we need to get all parties in agreement on what the SDLC process would be for this migration and implementation. We needed to define the flow of activities and to what degree these activities needed to be isolated in discretely separate infrastructure.

After several conversations, all the groups involved, including the SaaS publishing system vendor and other associated third party SaaS vendors, agreed to adopt a coordinated and iterative release based model. See Figure 2.

VI. What about the testing?

Now that we had the approach defined, it was time to identify the actual tactics of executing this model. One of the challenges the migration presented was that it involved substantial migration of content and data to a cloud SaaS vendor environment. This environment was highly integrated in both real-time and batch, with multiple new system interfaces and new data feeds. Testing in the environments was found to require multiple resets. The nature of the configuration created test data challenges,

Figure 2: Multiple release iteration approach

![Multiple Release Iteration Approach](image-url)

- Multiple functional releases, allowing dedicated QA team to test discreet sets of functionality at various scheduled intervals through the project lifecycle. These test intervals take place in a separate and more stable environment than those where the developers unit and integration test components.
- Releases to the test class environment will be scheduled in advance and occur on a 2-4 week basis.
- At the end of iteration based testing, a code complete period of End 2 End UAT will occur. This should occur in a production-like environment and will include performance testing.
where key system test scenarios were data destructive and had complex dependencies. The existing user data was large and complex, and constrained by the data extract/load capabilities of the SaaS vendors.

All of these challenges highlighted that a system test environment that included all systems would be required for success. The issue there was a lack of system test environments for most of the key systems involved, including but not limited to:

- The publishing system SaaS vendor
- Many of the other third party SaaS vendors
- Several of the internal built applications including eCommerce and some rich internet applications

All that existed were a development tier and a production tier. This was simply not sufficient for an implementation of this scale. In order to mitigate the risks associated with these challenges, a new system test environment was agreed on for all systems as picture in Figure 3. By selecting the customization model option from the key SaaS vendor (the publishing system), we were able to work with the vendor on creating this configuration. If the COTS model had been chosen, this option would not have been available and the entire project would have been at risk.

VII. Test data management a key challenge

Test data management for large, complex systems is always a challenge; this is not new or specific to the cloud environment. SaaS environments can add to that challenge as you do not have direct access to the database layer. You are constraining to the mechanisms provided by the SaaS vendors for the extraction and the import of your user data, content, and configuration information.

What was developed was a cloning process. The clone process was at the heart of the project strategy. It allowed system test environments to retain valid test data for content, customer and system configurations. The highly orchestrated process was executed multiple times throughout the development and test cycle in the project. There was an added advantage of developing this process and executing it multiple times. It provided the ideal testing and rehearsal of the mechanisms that would be required when going live with the migration.
VIII. Technology does not replace planning

You may be saying to yourself at this point, ‘A lot of this is not very different from migration challenges that have nothing to do with the cloud’. You would be correct. Migrating to or within the cloud may heighten some of the challenges you would normally encounter, but these are not insurmountable problems. Many times, when new technologies arrive, there is a perception that the technology will solve everything. Cloud technology provides many advantages to businesses, but as with all innovation, it will never negate the need for proper design and planning.

IX. This is a publishing system, preparing the content

Since a hybrid system by definition includes both internal and external systems, one of the basic challenges is preparation of the internal content for use by the external portions of the hybrid system. In particular for this publishing system, with a need to add significant metadata to the content to be published, a net new internal publishing system was developed to integrate with the external SaaS publishing system as diagrammed in Figure 4. Throughout the initial data load phase of the project, the content team leveraged this future production repository as the “gold” or master copy of content ready for launch. This allowed the repository to grow incrementally over an extended period of time, without incurring the substantial costs of content submissions and processing to the entire test environment infrastructure. By integrating this repository into our cloning process for testing, it allowed us the benefit of testing with as close to live data as possible, and enabled our performance testing process a good sense of ‘reality’ of how the actual production system would behave at go live.

X. Content without customers is not much use

My content process is in place and tested. Without my customers, it’s not going to see a lot of usage. One of the
key challenges on this migration effort is the customer data. The complexity of the data load needed from the variety of internal and external systems is diagrammed in Figure 5.

The key to the solution was the introduction of an operational data layer (ODL). The ODL acted as the interim staging area. ETL processes were built that facilitated the migration of the various data elements that would be pushed to the new SaaS publishing system. The ODL was also the center of the production synchronization processes. This facilitated a combination of real-time and batch-based integration between the SaaS publishing and client internal systems. As with the ‘gold copy’ content repository discussed earlier, this ODL was an integral part of the testing cloning process. This allowed us to test not only with actual live data at all points of the QA and performance testing, it allowed us to test the actual migration process before going live. The structure of the ODL is diagramed in Figure 6.

XI. How do we manage all of this?

For the content, that was a simple challenge. The ‘gold copy’ process we had already built and tested was the ongoing management process for publishing content to the new system, no further effort was required.

The customer data, as always, presented additional challenges. Changes to the customer data could occur through multiple channels. These channels included:

- eCommerce management
- Subscription management
- Web site identity portals (for end user self-service transactions)

To address this challenge, we designed a set of bi-directional synchronization processes. These processes facilitated the multiple channels that could originate changes to customer data. The overall synchronization process is shown in Figure 7.
XII. Summary

Migrating to the cloud, or within the cloud, should not be viewed as a scary undertaking that is going to introduce new and potentially insurmountable challenges. There are challenges with any migration from one technology to another, and the lessons from this migration are that the cloud is no different. Technology does not negate the need for proper design and planning. By ensuring your cloud migration efforts challenges are understood, with well thought out design and planning for those challenges, you will help ensure a successful migration that provides benefit back to the business.
Collaborative Consulting is a leading information technology services firm dedicated to helping our clients achieve business advantage through the use of strategy and technology. We deliver a comprehensive set of solutions across multiple industries, with a focus on business process and program management, information management, software solutions, and software performance and quality. We also have a set of offerings specific to the life sciences and financial services industries. Our unique model offers both onsite management and IT consulting as well as U.S.-based remote solution delivery.

To learn more about Collaborative, please visit our website at www.collaborative.com, email us at sales@collaborative.com, or contact us at 877-376-9900.