


Making the Move to Cloud Computing

The important work of moving the world forward does not wait to be done by perfect men.

—George Eliot (1819–1880)



Okay, now we have a data-, services-, and process-level understanding of our problem domain. We know how to test it, and we know how we are going to govern it. In Chapter 10, “Defining Candidate Data, Services, and Processes for the Clouds,” we figured out which processes, services, and data should reside on-premise and which should be cloud-based. Now we need to implement our final physical architecture, meaning we pick the proper platforms, test those platforms so that we know they meet our requirements, and move and/or create the processes, services, and data on the clouds.

There are a few things to remember here. First, this is just a physical instance of our architecture. The technology will change, but our architecture should remain fairly stable. This is more so the case with cloud computing, since changing cloud computing providers is much easier and less costly than changing on-premise systems.

Second, we select the technology or cloud computing provider during this final step. We reserved this decision until now because we wanted to remain objective up to this point to consider the valuable information that came to light during the processes we followed in the last several

chapters. If we get into this with the technology in mind, we are likely to skew the architecture toward that technology, which could be the wrong choice.

Finally, the number of hardware, software, and cloud computing providers leveraged will be many or few, depending on the needs of the architecture. No matter the numbers, our solution simply needs to be the appropriate one. Some target architectural instances will be complex, some simplistic, depending on the needs of the business and what we determined in the last several steps outlined in this book.

In this chapter, we focus on the cloud computing part of the architecture, including all on-premise and cloud computing–based systems. We are, as you may recall, simply extending our SOA to the platform of the clouds. We must deal with all on-premise hardware and software issues as well, including leveraging existing systems, creating new systems and services, adding new technology and governance, incorporating security, and so on.

Also, this chapter introduces the concept of the private cloud, which we covered briefly in Chapter 1, “Where We Are, How We Got Here, and How to Fix It.” Private clouds are virtualized hardware and software resources that exist within the firewall, within the data center, providing cloud computing–like characteristics around the ability to better utilize hardware and software resources within the enterprise. This is also an architectural option.

Selecting Platforms

As you can see in Figure 11.1, there are many patterns, or categories, in the world of cloud computing that you can leverage to meet the needs of your architecture. Some, such as security-as-a-service and testing-as-a-service, solve specific problems, and others, such as platform-as-a-service and infrastructure-as-a-service, provide complete platforms. They all have trade-offs and different problems that each solves. However, you must consider them all in light of your architecture.

While we covered the characteristics of these cloud computing providers in Chapter 3, “Defining the Clouds for the Enterprise,” it is a good idea to look at how they can fit into our architecture here, starting first with granularity of the providers. The categories are

- Storage-as-a-service
- Database-as-a-service
- Information-as-a-service

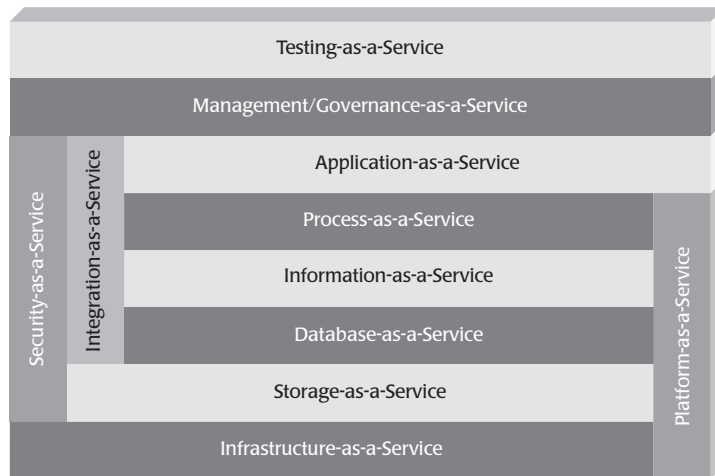


Figure 11.1 The patterns or categories of cloud computing providers allow you to leverage a discrete set of services within your architecture.

- Process-as-a-service
- Application-as-a-service
- Platform-as-a-service
- Integration-as-a-service
- Security-as-a-service
- Management/governance-as-a-service
- Testing-as-a-service
- Infrastructure-as-a-service

We can further break them down into fine-grained solutions, or those providers who solve very specific problems that alone cannot be considered a platform, and coarse-grained providers, or those who unto themselves are a complete platform.

Fine-Grained

1. Storage-as-a-service
2. Database-as-a-service
3. Information-as-a-service
4. Process-as-a-service
5. Integration-as-a-service
6. Security-as-a-service

7. Management/governance-as-a-service
8. Testing-as-a-service

Coarse-Grained

9. Application-as-a-service
10. Platform-as-a-service
11. Infrastructure-as-a-service

It is helpful to do this breakdown because one coarse-grained cloud computing provider can actually be made up of many fine-grained resources. For example, a single platform-as-a-service provider could offer storage-as-a-service, database-as-a-service, process-as-a-service, security-as-a-service, and testing-as-a-service.

However, while it may seem easier to leverage a coarse-grained cloud computing solution because it provides many fine-grained resources, the requirements of your architecture may dictate a finer-grained solution. You may find that selecting many fine-grained cloud computing solutions is a much better fit for your architecture when considering your requirements and/or the ability to mesh effectively with the on-premise portion of the architecture.

Also, we need to look at the capabilities of each platform that hosts the services, processes, and information we defined and refined in the previous chapters. The candidate cloud computing provider categories are, by architectural component,

Processes

- Application-as-a-service
- Platform-as-a-service
- Infrastructure-as-a-service
- Process-as-a-service
- Integration-as-a-service

Data

- Application-as-a-service
- Platform-as-a-service
- Infrastructure-as-a-service
- Storage-as-a-service
- Database-as-a-service
- Information-as-a-service

Services

- Application-as-a-service
- Platform-as-a-service
- Infrastructure-as-a-service
- Information-as-a-service

To make this point clearer, here are a few examples of physical instances of architecture. We first selected categories of cloud computing providers, and then we selected the providers (Example 11.1).

Example 11.1

Processes:

Process-as-a-service
Appian Anywhere

Data:

Infrastructure-as-a-service
Amazon EC2
Database-as-a-service
Amazon Simple DB

Services:

Infrastructure-as-a-service
Amazon EC2

For instance, we may store our data within Amazon Simple DB as well as on the Amazon EC2 platforms. Then, we might build and/or host the services on the Amazon EC2 platform, say, using an application server they provide on-demand within that platform. Finally, we could leverage Appian Anywhere as the platform where those processes live. Keep in mind that the processes are connected to the services, and the services are connected to the data, as we described in earlier chapters. We are just selecting the target platforms here.

This solution could become more complex by leveraging more cloud computing providers (Example 11.2).

Example 11.2

Processes:

Process-as-a-service
Appian Anywhere
Application-as-a-service
Salesforce.com

Data:

- Infrastructure-as-a-service
 - 3Tera Cloudware
 - Amazon EC2
- Database-as-a-service
 - Amazon Simple DB

Services:

- Infrastructure-as-a-service
 - Amazon EC2
 - 3Tera Cloudware
- Application-as-a-service
 - Salesforce.com
- Platform-as-a-service
 - Force.com

Or, as in Example 11.3, it could become a bit less complex by leveraging a single infrastructure-as-a-service cloud computing provider.

Example 11.3*Processes:*

- Process-as-a-service
 - Amazon EC2

Data:

- Infrastructure-as-a-service
 - Amazon EC2

Services:

- Infrastructure-as-a-service
 - Amazon EC2

We must also consider the other core components of the architecture, including security and governance, which can be deployed as on-premise or cloud-based, depending on our needs. Testing also can be delivered as a service or be on-premise as well.

The purpose of this exercise is to illustrate the number of architectural options we have, and how we can mix and match them, to form our final architecture using as many or as few as needed to address the requirements of the architecture and the business.

The Process of Moving to the Clouds

Figure 11.2 depicts the high-level process we can leverage to find the right cloud computing category or categories and the right cloud computing provider or providers to move the processes, services, and data we selected as good cloud computing candidates.

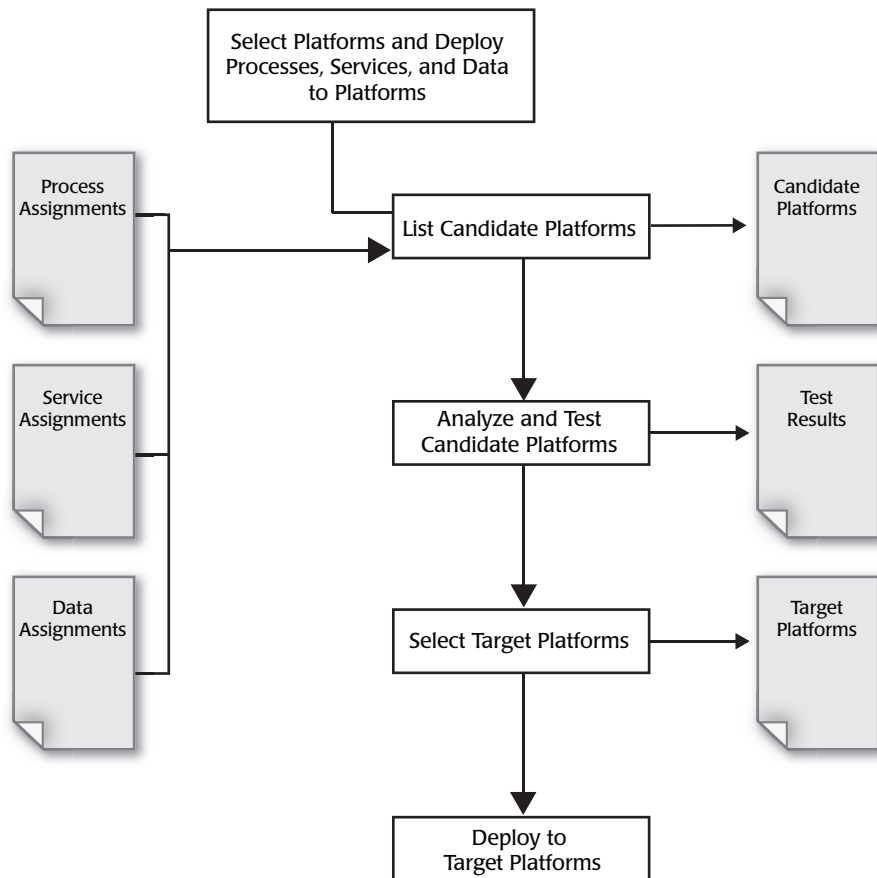


Figure 11.2 This, the final step in our process, is all about taking the process, service, and data requirements and mapping them to the right technology.

The core steps are as follows:

- List candidate platforms.
- Analyze and test candidate platforms.
- Select target platforms.
- Deploy to target platforms.

Let's look at each step in more detail.

List Candidate Platforms

Listing candidate platforms is pretty simple considering the information presented earlier. You need to list any and all cloud computing platforms that may be a fit for your to-be architecture. This requires that you understand what solutions are available, their categories, and what they do.

There are no hard and fast rules for defining a cloud computing solution. Thus, many software providers, whether they have a true cloud computing solution or not, have a tendency to say they do. For example, a few software vendors claim that since their software can be downloaded over the Web to an on-premise computing system, they are an on-demand or cloud computing platform. They are not. Therefore, this step can be a bit about separating the wheat from the chaff, more so than just tossing together a list.

We do not list cloud computing providers in this book, since that world changes monthly, with providers constantly being added, deleted, or combined. Mastery of SOA using cloud computing is as much about keeping up with the market space as it is about understanding what the vendors provide. In support of this process, you can visit the book's Web site to see an updated list of vendors and their categories.

In choosing your candidates, you must answer two key questions:

1. What categories do you need?
2. Which cloud computing providers in these categories should we consider?

The categories you leverage depend on the final logical architecture and the requirements you identified through this process. We can make some generalizations, though, including the fundamental layers you require and what to look for within each layer (see Figure 11.3):

- Storage
- Database

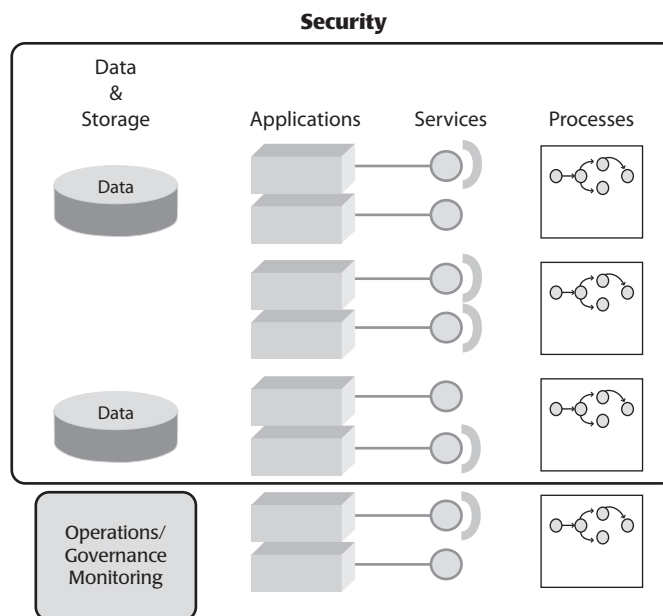


Figure 11.3 The core architecture requires you to find places for storage, data, operations, governance, security, services, and processes.

- Processes
- Services
- Security
- Governance
- Management

Storage is the category that supports part or all of the architecture in storing, sharing, and managing file systems. You typically leverage storage-as-a-service for this, either from a cloud computing provider that only provides storage-as-a-service, or as part of an infrastructure-as-a-service cloud computing provider.

Things you need to look out for here are capacity and performance. Capacity is your ability to scale your storage needs to support your architecture. Performance is your ability to move files to and from the cloud computing service at a speed that supports the business. Performance problems are the most likely issue here, so make sure you do your testing.

Database and database-as-a-service is the storage and retrieval of data using a platform-as-a-service, database-as-a-service, or infrastructure-as-a-service. Things you need to consider here include the ability for the cloud-delivered database to support the features and functions you may require for your architecture, including the use of stored procedures and triggers, the function of the API, adherence to standards, and performance.

Within the case of infrastructure-as-a-service, cloud computing providers typically allow you to leverage name brand databases, such as Oracle or MySQL. However, the database-as-a-service providers typically use a home-grown rather than brand-name database, and they tend to be more proprietary in nature.

Performance comes into play here as well. Most on-premise and traditional applications are data I/O bound, so you will find that similar performance problems may exist here. Consider the overhead of I/O on a multitenant platform and the latency that can occur when you send large amounts of data to and from your enterprise to the cloud computing provider and back again over the Internet. This could also make a case for placing the database closer to the processes and services that leverage that database, which is a core tenet of architecture when you consider performance and the reliability of databases.

Always Check on the Frequency of Outages

For all of this to work, reliability is a core requirement of your cloud computing provider. As you select cloud computing providers for your to-be architecture, make sure to look at the reliability of each provider. Typically, this means looking at the number of outages they experienced over a 2-year period. Also, look at how they support failover and other recovery operations when events such as network, hardware, and software failures occur.

While many of these outages make the IT press, the lesser known cloud computing providers often go unnoticed when outages occur. Make sure to call a few references—both those the vendor provides and perhaps a few they do not—and ask about the frequency of outages. Also, if the cloud computing provider is good, it should have a record of outages, why they occurred, and what they are doing to prevent such outages in the future.

Processes can exist on process-as-a-service, platform-as-a-service, application-as-a-service, and infrastructure-as-a-service providers, for the most part. You need to consider a few issues here.

When using process-as-a-service providers, keep in mind that processes are all they do. You must therefore bind the other architectural components (typically services and data) to those processes. The data and service assets exist either within on-premise systems or with other cloud computing providers, so you must make sure that integration occurs and is reliable.

Application-as-a-service providers typically do not provide a platform for you to create your own processes but allow you to leverage prebuilt processes on their platform. This is handy because, for instance, you need not create a custom fulfillment process for your business—you can just leverage theirs. However, as with the process-as-a-service, the processes are isolated and thus must be linked back with other on-premise and cloud computing–delivered systems that are part of the architecture.

When considering infrastructure-as-a-service providers and platform-as-a-service providers, you are typically dealing with platforms that provide the “complete stack,” including storage, database, processes, applications, services, development, testing, and more. These processes are just a component of those platforms. It may seem tempting to leverage “complete stack” providers, since they do indeed provide one-stop shopping for cloud computing. However, you will have to make trade-offs: You might love the application development features of one platform-as-a-service provider but hate the way its product manages processes or find that its process engine is sluggish. In many cases, it may be better to leverage other cloud computing providers or even on-premise software to address processes, trading simplicity for complexity but leveraging a process engine that is the right fit for the architecture.

Services (e.g., Web Services), generally speaking, can live on most cloud computing platforms. However, only a few (including platform-as-a-service, process-as-a-service, and infrastructure-as-a-service) provide the capabilities to create and host services through which application-as-a-service and information-as-a-service provide access to their hosted prebuilt services, which you can use but cannot change.

The most common issue here is performance. Services such as Web Services (whether using REST or SOAP) tend to cause performance problems if the platform hosting the service cannot provide enough computing resources, or if there are too many services and they saturate the platform and

the network. Again, you need to test for performance by actually using the services, and adjust your platform, the number of services you leverage, and the way those services are designed to optimize the performance of your architecture.

Security is not a platform or a piece of software that exists on-premise or on cloud computing platforms. If done right, it should be systemic to the holistic architecture, no matter how much of it is on-premise or cloud computing–delivered. You address security by creating a strategy and a model to secure your architecture based on the requirements you identified. Then you select the proper approach and supporting enabling technology. Security typically centers on identity management and the standards that support identity management.

With the increasing interest in identity management, in support of more complex and distributed architectures such as SOA and SOA using cloud computing, the need for standards to better define this space has arisen. These standards all aim to bind together identity management systems within all organizations into a unified whole, allowing for everyone to be known to everyone else, securely.

Why do we need identity management? It is a fact that services are not for internal use anymore, as is the case when leveraging cloud computing. Those who leverage services (consumers) and those who produce services (providers) must be known to each other; otherwise, we risk invoking malicious or incorrect behavior, which could cost us dearly. This is clearly the case with cloud computing that leverages services.

Governance brings its own set of issues when considering architecture and cloud computing. While there are governance systems that are cloud delivered, and they work well for some types of architecture, governance systems that implement, manage, and enforce policies are runtime in nature and are typically on-premise.

Issues to look out for here again include performance, since, in some instances, executing policies could cause latency issues. Also important is the governance solution's ability to govern resources, which are typically cloud-delivered services. This means having the ability to track remote services within the governance technology's repository as well as to monitor those services during runtime.

Management of a widely distributed and complex architecture, such as SOA using cloud computing, requires a management technology that can see

both on-premise systems, which most do, and cloud computing–based systems, which only a few do well. Moreover, you should check whether the cloud provider has an interface on their software that allows management technology to talk to it.

The core idea is to provide a management platform that sees all on-premise and cloud computing–based systems at the “working or not working” level, at the very least, meaning we can see whether a system is down and how that status will affect other systems in the architecture. However, it is preferable to have a management system that can see systems such as services, processes, data, storage at more granular levels, which makes it much easier to diagnose issues and spot troubles before they happen.

Management and governance are clearly linked and have very similar patterns.

Analyze and Test Candidate Platforms

Once you select the candidate cloud computing platforms, you need to make sure they live up to the requirements we established. You do this through some deep dives into each candidate platform you selected and then through testing.

We covered testing extensively in Chapter 9, “Testing from SOA to the Clouds,” so we do not go too deep into it here. However, this testing is a bit different in that you are actually testing the generic capabilities of the cloud computing platform. Specifically, you look at how that cloud computing platform will support the requirements of the architectural components, including services, data, and processes, but you are not yet deploying on those platforms. They could be the wrong choices, which is why we do the testing.

The only thing to add from Chapter 9 is the use of performance modeling and performance testing. Modeling creates a simulation of how the system should perform under different types of loads, typically light, medium, and heavy. Performance testing determines how the architecture performs under stress. It involves modeling the architecture, including how the information will flow and the services will be invoked, and how flow and invocation affect the different computing resources, both on-premise and cloud-based. You should have a general idea as to what performance you can expect from the cloud computing platforms and how things such as decreasing processing power or expanding bandwidth should affect overall performance.

While proving the performance models, you should leverage performance testing, determining how well and how fast the holistic architecture, both on-premise and cloud-based, will support the business. Moreover, measure how the system performs during an ever-increasing storage, database, process, and service-processing load. If they do not perform well, find out where the bottlenecks are: in the network? the database? services? If necessary, work with the cloud provider to correct them.

Select Target Platforms

Once we go through all of the analysis, including a service-, process-, and data-level understanding of our problem domain, and have considered both security and governance, compiled a list of candidate systems, and completed the validation testing, it is time to pick the cloud computing platforms.

This step is pretty easy considering that any issue around the platform's ability to meet the requirements of the architecture, and thus the business, should be well known and understood by now. Also, keep in mind that it is more likely that the final selection of the suite of target cloud computing platforms is very different from what you first envisioned, but if you did your homework and followed each step in this book, they should be the proper platforms for your architecture.

Also worth mentioning is the ease of switching, or should we say, the relative ease of switching, from one cloud computing platform to another if for some reason you make the wrong call, or more likely, if some business event occurs with the cloud computing platform, such as the cloud computing provider going out of business or a merger or acquisition changes or removes that platform. Of course, this depends on the cloud computing provider you selected, its use of standards, and your ability to find another provider that offers similar characteristics and features.

The business issues are more important if you are looking to create an SOA using cloud computing, since that scenario depends entirely on the cloud provider to stay in business and keep up and running. You need to carefully consider

- The viability of the provider and the likelihood that it will provide ongoing support for your cloud computing platforms.
- The provider's ability to recover from hardware, software, and network failures, dynamically and with minimum downtime.

- The service-level agreements, or SLAs, and a meeting of the minds between you and the cloud computing provider as to what service levels need to be supported for your architecture.
- A complete understanding of the policies of the cloud computing provider and what denotes a violation. In some instances, cloud computing providers have, without notice, canceled accounts due to policy violations.

Deploy to Target Platforms

This is the “just do it” step, meaning that we actually port code; migrate data; create new services, processes, and databases; and test and validate that all services, databases, and processes are working correctly and as defined, using the steps in this book.

The approach you should leverage here should be focused on migration and development over time, not a “big bang approach.” You should select which components of the architecture should move to or be created on the cloud computing platforms, going from the most important to the least.

As you move these architectural components to the cloud computing platforms, make sure they are functioning correctly and have been properly tested before moving on to the next architectural component. While the pressure may be on to make “the big switch,” the reality is that this evolutionary approach prevents problems and does not overwhelm those who deploy services, data, and processes to the cloud computing platforms. Also, this approach provides the value of learn-as-you-go, meaning that your knowledge of how to make cloud computing platforms work for your architecture will increase significantly as we move through this process.

Where Social Networking Fits with Cloud Computing

Opinions on social networking vary widely, from “No way, it’s too risky” to “It’s a way of life; you might as well learn to leverage it for productivity.” Social networking has already been lumped in with cloud computing, so it is a good idea to consider its value and risks. How will you integrate social networking within your SOA using cloud computing architecture? Now is a good time to form a set of policies.

continued

It does not matter whether you understand the differences between MySpace and Facebook. Most of the people who work in your enterprises, IT or not, leverage some sort of social networking system, and most look at it at least once a day during work hours. Assuming you could put your foot down and declare this stuff against policy, most employees would find that a bit too Big Brother-ish and would find a way to do it anyway, perhaps on their cell phones or PDAs. Social networking in the workplace is a fact of life you must deal with, and perhaps it could be another point of value that comes down from the clouds.

To figure out the enterprise opportunities or risks involved with social networking, you first must define the reasons that people leverage social networking:

- To communicate, both passively and actively, in an ongoing manner and through various mediums, with people in whom they are interested—usually with friends and family, but in some cases, the activity is all work related. Typically, it's a mixture of both.
- To learn more about areas of interest. For example, LinkedIn groups, such as SOA, Web 2.0, and enterprise architecture.
- To leverage social networking within the context of the SOA using cloud computing architecture, such as allowing core enterprise systems, on-premise or cloud-based, to exchange information. For instance, social networking can be used to view a customer's Facebook friends list to find new leads, and thus new business opportunities, by integrating Facebook with your sales force management system.

There are risks involved in online social networking, however. People can (and do) lose their jobs because of a posting on a social networking site that put their company at risk. People can be (and have been) publically embarrassed by posting pictures, videos, or other information they thought would be, uhm, private. Also, there are many cases of criminal activity using social networking as a mechanism to commit a crime.

Here is the gist of it. Social networking, in one form or another, is always going to be around. So if you are doing enterprise IT, including cloud computing, you might as well accept it but learn how to govern through education, policies, and perhaps some technology. While there are risks, there are also opportunities, such as the ability to leverage information gathered by social

networking sites to enhance marketing and sales, and by integrating those systems with your core business systems, both on-premise and cloud-based.

Make sure to define to all employees when and where it is appropriate to leverage social networking within the workplace. Try not to be too restrictive, but instead inform them of what is a good social networking practice and what is unacceptable. You will find that 99% of those who already leverage social networking are already using their heads.

Keep in mind that leads are being developed, sales made, and customers supported using social networking systems. Not surprisingly, the correct use of social networking can have a very positive effect on the bottom line, especially considering the access to valuable information that these social networking sites provide and the ability to leverage that information to provide better business intelligence to support the core business systems. You may also find that employee-to-employee communication improves using social networking systems, internal or public.

Make sure to work with your legal department to define written policies for social networking, and make sure all employees are aware of and committed to adhering to these policies. The idea is to cover the company in case someone does something stupid. Again, you are mitigating the risk, not eliminating it.

Finally, monitor the use of social networking sites with standard Web governance technology, including logging and trending. This is not to catch a particular person who is leveraging social networking but to determine the patterns of use over time. Also, if particular sites do become a problem, you can shut them off.

What about Private Clouds?

Until now, we have yet to hit on the notion of private clouds, beyond our introduction in Chapter 1. It is important that we dive a bit deeper into the concept here as an architectural option for our SOA using cloud computing while we select platforms for deployment.

Private clouds are cloud computing–like infrastructures that leverage virtualization and exist within private data centers. The core notion is that cloud computing is a great approach to optimize the use of hardware and software, and we can obtain the same value by doing the same trick within our data center using virtualized resources, or private clouds. Most computing resources,

such as database servers, application servers, and governance servers, are underutilized, typically running at only 5% of their capacity at any given time (based on my experience and observations).

A private cloud, or more exactly, the use of virtualization software such as VMWare, gives you the ability to address many physical servers as one virtual server and thus to leverage the processing power of all computing resources as if they were a single resource. You can optimize the use of all hardware and software resources more so than if they were bound to a particular hardware and software platform. The virtualization software can allocate the process load between all available servers, which improves the utilization of each computing resource.

The end result is that you can support more data, services, and processes on a fewer number of servers, and this virtualization mechanism reduces costs. This approach is called *private clouds* because it features many of the same benefits of cloud computing, including the ability to reduce costs.

Enterprises are interested in private clouds because, in many instances, they cannot host their data outside of their firewalls due to privacy and legal issues, but they want to take advantage of the cloud computing architecture. Many of them want to remain in control of their systems and information and have already invested in hardware and software, the cost of which cannot be recovered.

Again, private clouds are basically virtualized platforms, and all of the issues that virtualization has attempted to resolve in past years is applicable here. The patterns of virtualized platforms and the patterns of some cloud computing platforms, such as infrastructure-as-a-service, are almost identical other than the location of processing and provisioning. Where public clouds are for anyone who can sign up, and in many instances the cloud users are not verified, private clouds allow only authorized persons, or internal users, to provision themselves on the private clouds.

Many view this utilization of virtualization systems as simply jumping on the bandwagon to ride the cloud computing wave. However, it is the reality of the forthcoming modernization of the enterprise: the need to do much more with much less and to get smarter with sharing resources, both on-premise and remote. While cloud computing will clearly drive some aspects of modern enterprise architecture, the ability to create similar value within existing and paid-for data centers is a viable architectural option and should be considered in the mix.

New “Cloudy” Platforms

The activities outlined in this chapter represent some of the most fun you will have around cloud computing: actually moving systems to the clouds and making those systems work for the business. It is “doing” rather than planning or analyzing, but it is also the trickiest of all the activities we have outlined, and it carries the most risk.

In addition, unless you are reading this book well into the future, you know that the cloud computing platforms are a bit of a moving target, meaning that as the hype and the market heat up, new providers will appear weekly, and existing providers will pack in as much functionality as they can to capture the market.

Cloud computing platforms are easily changed, since they do not require the distribution of software to enterprises, and change will be an ongoing activity: constant upgrades, bug fixes, and other changes to the platform. Hopefully, these changes will move the overall system in better directions and not break your architectural components that exist on these platforms—they will be backward compatible.

What seems like an unnatural act today, as you relocate and create architectural components on cloud computing platforms, will seem second nature as time progresses. Clearly, as we move many of our services, processes, and data out to the clouds, clouds will become a major component of enterprise architecture and SOA.

SOA using cloud computing is the best architectural approach, as you have seen throughout the book. SOA using cloud computing provides the ability to address computing resources using the best possible configuration, and it matters not where those computing resources reside. We continue to extend them to the clouds, and more clouds will surely appear.