**19SFC162 CLOUD SECURITY-MODULE 2 NOTES**

**1.Explain the various Cloud customer responsibilities with its key points**

For any business thinking of embracing public cloud services, or any variation on X-as-a-Service paid features, it is vital that they understand how security works on public cloud servers. The importance of this is hard to overstate. According to Gartner research, [95% of all security incidents involving cloud servers are the fault of the cloud customer](https://www.gartner.com/smarterwithgartner/is-the-cloud-secure/), rather than the cloud provider. Not only that, they predict this statistic to hold true until at least 2022!

This clearly illustrates a need for better understanding of cloud security models among business users, particularly given [how destructive and costly](https://www.auditwolf.com/blog/your-data-exposed-how-cloud-misconfiguration-weakens-us-all) cloud security incidents can be.

So in this article, we want to talk in some depth about the shared security model – the standard model for cloud security on Azure, AWS, and all other major public cloud systems.

**What Is the Shared Security Model Of Cloud Security?**

**“Who is responsible for cloud security, the cloud provider or the customer?”**

According to the shared security model, the answer is both. Both provider and customer have areas of responsibility for maintaining security, and there are relatively few places where that responsibility overlaps. In our experience, a large number of security problems among cloud customers involve misunderstanding those realms of responsibility.

Further, it’s safe to say that there are more areas of cloud security for which the customer is responsible, rather than the provider. Companies who understand this basic delineation of responsibility will be a much better position to maintain a high level of cloud security.

**What Aspects Of Security Does The Cloud Provider Oversee?**

Broadly speaking, in most public cloud services, the cloud provider is only responsible for the physical security of their systems, as well as protecting the underlying infrastructure powering the cloud system.

This breaks down to several clear areas of responsibility.

**Physical premises security.** The cloud provider is wholly responsible for protecting their physical facilities from any intrusion and ensuring that their hardware is not directly compromised.

**Environmental security.** Along with security against in-person tampering, cloud providers will also be expected to provide a reasonable level of protection against environmental problems, such as earthquakes or weather-related disasters. Of course, there will be a practical upper limit to what they can guarantee here – nature can always potentially conjure a disaster which will overwhelm even the best human-built protection.

**Underlying server-level security.** If we’re talking about attacks which would affect an entire cloud server, and all its customers, that’s the responsibility of the service provider. They are the ones who should be providing protection against DDOS attacks, Man-In-The-Middle attacks within their network, and similar broad-spectrum attacks.

**Cloud systems updates and patching.** The provider is responsible for the underlying software powering their cloud systems. If, for example, a zero-day exploit is discovered in Azure that could affect any customer, it’s Microsoft’s responsibility to fix it. However, this is only true for software being directly provided by the vendor. The customer is still responsible for any software they deploy.

**Business continuity services and contingencies.** Most, if not all, reputable cloud providers will provide some contingencies in the case of accident or system failure, such as backup servers. As with the primary cloud infrastructure and software, they are responsible for securing any such backup measures.

And that sums up what cloud providers are reasonably expected to provide, in terms of security. That means everything else is the customer’s responsibility.

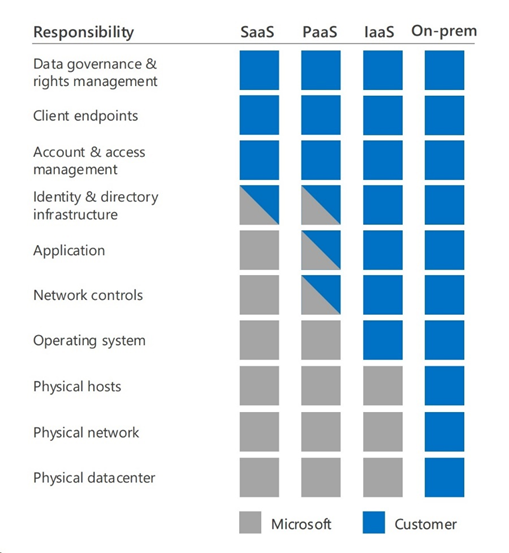
**What Aspects Of Cloud Security Does The Customer Control?**

Since cloud services can be used for a wide variety of services, it’s impossible to present a 100% complete list, but this hits the most common highlights:

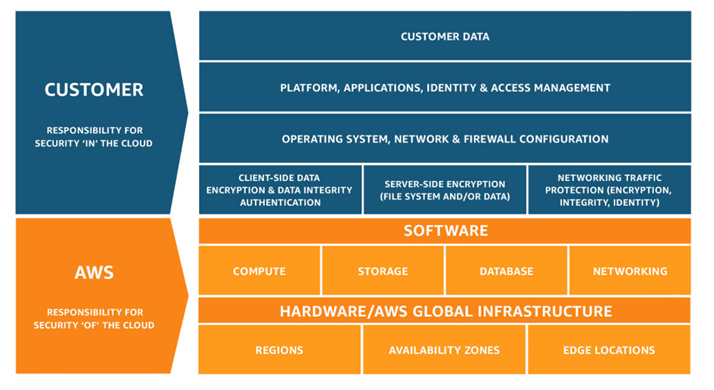
* Properly configuring systems for detecting when a cloud account has been breached.
* Managing and handling all matters relating to logins, authentication, and access permissions.
* Controlling what data is uploaded to the cloud, and ensuring that proper encryption is utilized.
* Any and all monitoring of access to the cloud services, including virus scanners and port scanning.
* Updating and maintaining user-side software, such as guest operating systems, VMs, third-party apps, etc.
* Understanding platform-specific usage and documentation to ensure provided tools are being used in the recommended manner.
* All device-side permissions and security, such as which devices are allowed onto the cloud server.

Are there any areas of overlap? A few. For example, say a cloud provider has created tools that allow reporting on usage and access to cloud access. It would be the provider’s responsibility to ensure the tools are functioning properly without any low-level security holes, but it would be the customer’s responsibility to read the documentation and configure those tools appropriately for their needs and with the correct access privileges attached.

Both Microsoft and Amazon have some great imagery to help you understand the shared responsibility model. Here is Microsoft's:



And here is Amazon's:



**2.Summarize the key features of Compliance and Audit Security Recommendations with illustrations**

**How to manage a successful audit**

* Establish a security baseline through annual audits.
* Spell out your objectives.
* Choose auditors with "real" security experience.
* Involve business unit managers early.
* Make sure auditors rely on experience, not just checklists.
* Insist that the auditor's report reflects your organization's risks.

**Don't let this happen to you. And it won't, if you know how to:**

[Choose a good auditor.](https://searchsecurity.techtarget.com/tip/Best-practices-for-choosing-an-outside-IT-auditor)

Spell out your requirements.

[Make sure the audit is conducted properly.](https://searchsecurity.techtarget.com/feature/Preparing-for-auditors-Checklists-for-before-during-and-after-an-IT-audit)

Intelligently evaluate the ultimate deliverable--the auditor's report. An audit can be anything from a full-scale analysis of business practices to a sysadmin monitoring [log files](https://searchsecurity.techtarget.com/answer/What-security-log-management-best-practices-should-my-team-follow). The scope of an audit depends on the goals. The basic approach to performing a security assessment is to gather information about the targeted organization, research security recommendations and alerts for the platform, test to confirm exposures and write a risk analysis report. Sounds pretty simple, but it can become quite complex.

**Establish a Security Baseline**

Your security policies are your foundation. Without established policies and standards, there's no guideline to determine the level of risk. But technology changes much more rapidly than business policies and must be reviewed more often. Software vulnerabilities are discovered daily. A yearly security assessment by an objective third party is necessary to ensure that security guidelines are followed.

Security audits aren't a one-shot deal. Don't wait until a successful attack forces your company to hire an auditor. Annual audits establish a security baseline against which you can measure progress and evaluate the auditor's professional advice. An established security posture will also help measure the effectiveness of the audit team. Even if you use different auditors every year, the level of risk discovered should be consistent or even decline over time. Unless there's been a dramatic overhaul of your infrastructure, the sudden appearance of critical security exposures after years of good reports casts a deep shadow of doubt over previous audits.

If you don't have years of internal and external security reviews to serve as a baseline, consider using two or more auditors working separately to confirm findings. It's expensive, but not nearly as expensive as following bad advice. If it isn't practical to engage parallel audit teams, at least seek a second opinion on audit findings that require extensive work.

**Objectives: Know What You Want**

[Spell out what you're looking for](https://searchsecurity.techtarget.com/feature/Surviving-an-audit) before you start interviewing audit firms. If there's a security breach in a system that was outside the scope of the audit, it could mean you did a poor or incomplete job defining your objectives.

Let's take a very limited audit as an example of how detailed your objectives should be. Let's say you want an auditor to review a new Check Point firewall deployment on a Red Hat Linux platform. You would want to make sure the auditor plans to:

* Review and document the security mechanisms configured on the Check Point firewall and the Check Point Management Station.
* Review the Check Point firewall configuration to evaluate possible exposures to unauthorized network connections.
* Review the Red Hat Linux OS configuration to harden it against security exposures.
* Review router configuration and logging procedures.

From a security perspective, certify the firewall and OS for production.

Document disaster recovery procedures for the firewall and OS and "good housekeeping" procedures for Check Point's Object Management.

Perform a penetration test once the firewall and OS are in production.

**Hiring an Auditor**

You may be tempted to rely on an audit by internal staff. Don't be. Keeping up with patches, making sure OSes and applications are securely configured, and monitoring your defense systems is already more than a full-time job. And no matter how diligent you are, outsiders may well spot problems you've missed.

Technical audits identify risks to the technology platform by reviewing not only the policies and procedures, but also network and system configurations. This is a job for computer security professionals. Consider these points in the hiring process:

Look at the auditing team's real credentials. Don't be influenced by an [alphabet soup of certification letters](https://searchsecurity.techtarget.com/tip/Security-certifications-Are-they-worth-the-trouble). Certifications don't guarantee technical competence. Make sure the auditor has actual work experience in the security field acquired by years of implementing and supporting technology.

Résumés of the auditors should detail security projects--not just audits--they have worked on, including references. Real-world experience implementing and supporting security technology gives an auditor insight into subtle issues that could reveal serious security exposures. Any published works should be included to demonstrate the auditor's expertise.

**Incomplete audits**

In general, when we talk about audits--especially by outside auditors--we're talking about security assessment reviews. A complete security assessment includes penetration testing of internal and external systems, as well as a review of security policies and procedures. It's a cooperative, rather than adversarial, exercise to learn about the security risks to your systems and how to mitigate those risks.

There are other kinds of audits that have a much narrower focus and are of far less value. In the worst-case scenarios, they can do more harm than good:

**Black Box Audits:**   
Some IT managers are enamored with "black box" auditing--attacking the network from the outside with no knowledge of the internal design. After all, if a hacker can perform digital reconnaissance to launch an attack, why can't the auditor?

A black box audit can be a very effective mechanism for demonstrating to upper management the need for increased budget for security. However, there are some drawbacks in emulating the actions of malicious hackers. Malicious hackers don't care about "rules of engagement"--they only care about breaking in. They have plenty of time to gather information and have no concern about what they break in the process. Who owns the first router into the network, the client or a service provider? A malicious hacker wouldn't care. Try hacking an ISP and altering a site's DNS records to break into a network--and maybe get a visit from the FBI.  
A black box audit is a view from a single perspective--it can be effective when used in conjunction with an internal audit, but is limited on its own.

**Surprise Inspections:**   
Audit departments sometimes like to conduct "surprise inspections," hitting an organization without warning. The rationale behind this approach is to test an organization's response procedures. In reality, it's usually an attempt to catch someone with their pants down rather than a proactive effort to improve an organization's security posture.

Surprise inspections can backfire badly if critical work is interrupted by such a "fire drill." Think of a trading floor getting flooded with port scans during prime business hours. Some auditors seem to believe an organization will take extra security measures if they know an audit is pending. In reality, even if the organization performs a quick cleanup, it won't disguise embedded security problems. Surprise inspections run the risk of causing as much service interruption as an actual hacker attack.

And don't be impressed by people who call themselves "[ethical hackers](https://searchsecurity.techtarget.com/magazineContent/Weighing-the-risk-of-hiring-hackers)." Many so-called ethical hackers are just script-kiddies with a wardrobe upgrade. Do your homework. Network with people you know and trust in the industry. Find out what they know about prospective auditing firms. See if you can track down clients who have used the firms but are not on their reference list.

**Find the right fit.** Meet with a range of auditing firms. Consider the small firms specializing in security, along with the Big 4 accounting firms to see which best meets your needs. An auditing firm needs to know if this is a full-scale review of all policies, procedures, internal and external systems, networks and applications, or a limited scope review of a specific system.

Smaller firms may choose not to bid on a large-scale project, and larger companies may not want to bother with a review of one system, because they're reluctant to certify a system without looking at the entire infrastructure.

**Insist on the details.** Some firms may be reluctant to go into great detail about their methods without a contract. They may simply slide a sales brochure across the table and say, "Our record speaks for itself." Don't be hoodwinked by this; while it's nice to know they have a combined 200 years of security expertise, that doesn't tell you a lot about how they plan to proceed with the audit.

If they're serious about bidding for your business, the auditors will put together a [statement of work](https://searchitchannel.techtarget.com/definition/statement-of-work-SOW) (SOW), which details how they plan to meet your objectives--the methodologies and deliverables for the engagement. The devil is in the details, and a good SOW will tell you a lot about what you should expect. The SOW will be the basis for a project plan.

The SOW should include the auditor's methods for reviewing the network. If they balk, saying the information is proprietary, they may simply be trying to hide poor auditing methods, such as simply running a third-party scanner with no analysis. While auditors may protect the source of any proprietary tools they use, they should be able to discuss the impact a tool will have and how they plan to use it. Most good auditors will freely discuss their methods and accept input from your organization's staff. Basic methodology for reviewing systems includes research, testing and analysis.

**Agree on the appropriate payment plan.** The bottom line for the bid is how much it will cost and what you're getting for your money. Some auditing firms quote a flat rate in return for a report detailing their findings and recommendations. Others may estimate the number of days an audit will take, with both sides agreeing to a flexible cost, within limits.

For a complex audit of an entire company, many unanticipated issues could arise requiring extensive time from the auditors, making a flat rate more attractive for the contracting organization. If the organization has good documentation or if the scope is limited, a flexible rate may be more economical.

**Prepare to Be Audited**

Auditors must make certain assumptions when bidding on a project, such as having access to certain data or staff. But once the auditor is on board, don't assume anything--everything should be spelled out in writing, such as receiving copies of policies or system configuration data. These assumptions should be agreed to by both sides and include input from the units whose systems will be audited.

Nobody likes surprises. Involve the business and IT unit managers of the audited systems early on. This will smooth the process and perhaps flag some potential "Gotchas!", such as a dispute over the auditor's access.

Consider the case of one respected auditing firm that requested that copies of the system password and firewall configuration files be e-mailed to them. One of the targeted organizations flatly refused. In fact, they thought the request was a social engineering test. Their security policy prohibited external release of any files requiring privileged access to read. If the audited organizations had been involved in the process from the start, problems like this might have been avoided.

So, [set the ground rules](https://searchsecurity.techtarget.com/tutorial/Network-security-audit-guidelines-Inside-the-importance-of-audit-planning) in advance:

1.) Your managers should specify restrictions, such as time of day and testing methods to limit impact on production systems. Most organizations concede that denial-of-service or social engineering attacks are difficult to counter, so they may restrict these from the scope of the audit.

2.) Make sure the auditors conform to your policy on handling proprietary information. If the organization forbids employees from communicating sensitive information through nonencrypted public e-mail, the auditors must respect and [follow the policy](https://searchenterprisedesktop.techtarget.com/tip/Checklist-How-to-configure-the-audit-policy). The audit report itself contains proprietary data and should be handled appropriately--hand delivered and marked proprietary and/or encrypted if sent through e-mail.

3.) Give the auditors an indemnification statement authorizing them to probe the network. This "get out of jail free card" can be faxed to your ISP, which may become alarmed at a large volume of port scans on their address space.

As part of this "prep work," auditors can reasonably expect you to provide the basic data and documentation they need to navigate and analyze your systems. This will obviously vary with the scope and nature of the audit, but will typically include:

**Copies of all relevant policies and procedures.** Policies may include end-user policies (password expiration, virus scanning, acceptable use); privacy (for internal users and client data); privileged access (sysadmins) and incident handling. Some of the procedures to review are data backup, disaster recovery, incident response and system administration.

A list of OSes.

Network topology, specifying target IP ranges.

External security devices (firewall software, IDS).

List of application software.

Digging In

The entire process of analyzing and then testing your systems' security should be part of an overall plan. Make sure the auditor details this plan up front and then follows through. For instance, using the Check Point/Red Hat example cited above, a general outline would include analyzing and then testing vulnerabilities:

**For the OS:** Directory structure, application packages installed, logging capabilities and services available for the Linux OS.

**For the firewall and management console:** system configuration and authentication mechanisms, in addition to logging capabilities and available services.

The auditor should begin by reviewing all relevant policies to determine the acceptable risks. They should check for unauthorized implementations such as rogue wireless networks or unsanctioned use of remote access technology. The auditor should next confirm that the environment matches management's inventory. For example, the auditor may have been told all servers are on Linux or Solaris platforms, but a review shows some Microsoft servers. If the auditing team was selected for Unix expertise, they may not be familiar with Microsoft security issues. If this happens, you'll want the auditor to get some Microsoft expertise on its team. That expertise is critical if auditors are expected to go beyond the obvious. Auditors often use security checklists to review known security issues and guidelines for particular platforms. Those are fine, but they're just guides. They're no substitute for platform expertise and the intuition born of experience.

The auditor will use a reputable vulnerability scanner to check OS and application patch levels against a database (see cover story, "How Vulnerable?") of reported vulnerabilities. Require that the scanner's database is current and that it checks for vulnerabilities in each target system. While most vulnerability scanners do a decent job, results may vary with different products and in different environments. The auditor should use several tools (see "The Auditor's Toolbox") and methods to confirm his findings--most importantly, his own experience. For example, a sharp auditor with real-world experience knows that many sysadmins "temporarily" open system privileges to transfer files or access a system. Sometimes those openings don't get closed. A scanner might miss this, but a cagey auditor would look for it.

Discovering security vulnerabilities on a live production system is one thing; testing them is another. Some organizations require proof of security exposures and want auditors to exploit the vulnerabilities. This can be dangerous. A successful system compromise may be a graphic way to convince management of the dangers of the exposure, but are you prepared to risk compromising or even bringing down a live system?

The SOW should specify parameters of testing techniques. And the auditor should coordinate the rules of engagement with both your IT people and the business managers for the target systems. If actual testing isn't feasible, the auditor should be able to document all the steps that an attacker could take to exploit the vulnerablility. For example, if the system password file can be overwritten by anyone with specific group privileges, the auditor can detail how he would gain access to those privileges, but not actually overwrite the file. Another method to prove the exposure would be to leave a harmless text file in a protected area of the system. It can be inferred that the auditor could have overwritten critical files.

**The Audit Report**

The audit's done, and you [look at the report](https://searchnetworking.techtarget.com/answer/Where-can-I-find-a-sample-security-audit-report-How-can-I-run-my-own). Did you get your money's worth? If the findings follow some standard checklist that could apply to any organization, the answer is "no." If you see pages of reports generated by a vulnerability scanner, but no independent analysis, the answer is, again, "no."

**What is the most underrated best practice or tip to ensure a successful audit?**

While some commercial vulnerability scanners have excellent reporting mechanisms, the auditor should prove his value-added skills by interpreting the results based on your environment and a review of your organization's policies.

That analysis should reflect your organization's risks. Tools lack analytical insight and often yield false positives. You hired expert people, not tools, to audit your systems. So, how do you know if the auditor's risk assessment is accurate? For starters, have your IT staff review the findings and testing methods and provide a written response.

The auditor's analysis should follow established criteria, applied to your specific environment. This is the nitty-gritty and will help determine the remedies you implement. Specifically, the report should outline:

The source of the threat--from internal users or the public Internet.

The probability of exploitation. Have other sites suffered intrusions because of this exposure?

The impact of the exposure. Bottom line, how much money--or loss of reputation, etc.--will it cost the organization if this exposure is exploited?

Recommended actions to fix problems. Is it an amendment to the policy, stating something like, "all software must be licensed appropriately," applying patches or a redesign of the system architecture? If the risk is greater than the cost of repair. A low-risk problem, like not displaying warning banners on servers, is easily fixed at virtually no cost. Using an application with a history of repeated security problems may be a higher risk, but it may be more costly to integrate a more secure application. The most secure application may not be the best business application. Security is a balance of cost vs. risk.

Potential legal liability. Could your systems become a repository for contraband (e.g., child porn, pirated software)? For example, a Web server may have an exposure that would permit an outsider to post files to it, though not overwrite content. This may not seem like a big issue, but people who trade in contraband look for untraceable storage locations for their data.

The risk of service interruption, such as a DoS attack.

The auditor's report should include a brief executive summary stating the security posture of the organization. An executive summary shouldn't require a degree in computer science to be understood.

A statement such as "fingerd was found on 10 systems" doesn't convey anything meaningful to most executives. Information like this should be in the details of the report for review by technical staff and should specify the level of risk.

Finally, there are occasions when auditors will fail to find any significant vulnerabilities. Like tabloid reporters on a slow news day, some auditors inflate the significance of trivial security issues.

What do you say if there's nothing to say? Rather than inflate trivial concerns, the auditors should detail their testing methods and acknowledge a good security posture. To add value, they could point out areas for future concern or suggest security enhancements to consider.

However, it should be clear that the audited system's security health is good and not dependent on the recommendations. Remember, the purpose of the audit is to get an accurate snapshot of your organization's security posture and provide a road map for improving it. Do it right, and do it regularly, and your systems will be more secure with each passing year.

**3.List the key factors that leads to Changing providers reasons**

**6 things you can do to avoid cloud vendor lock-in**

Vendor lock-in is a concern for those moving to the cloud. Here’s what you can do to mitigate the risks.

[Migrating to the cloud can bring a multitude of benefits](https://www.thorntech.com/2016/06/cloud-computing-the-15-ways-your-business-can-benefit-from-the-cloud/) to your company, such as increased agility, flexibility, and cost savings.

Despite all of these positives, many companies who are considering a move to the cloud have concerns. And one of the primary issues is vendor lock-in.

When the foundation of your company’s IT is in the hands of an outside vendor, these concerns are valid.

**You might ask yourself these questions:**

* What if the cloud service provider’s (CSP) offerings don’t meet my needs?
* What if the CSP makes a major product change that doesn’t work for my business?
* What happens if the CSP goes out of business?

Most cloud migrations, if planned and executed correctly, go pretty smoothly. But if something goes wrong with your CSP after your migration, moving to another cloud vendor can incur substantial costs, technical problems, and more.

So what can you do to avoid cloud vendor lock-in, or at least minimize its negative effects?

Causes of cloud vendor lock-in fears

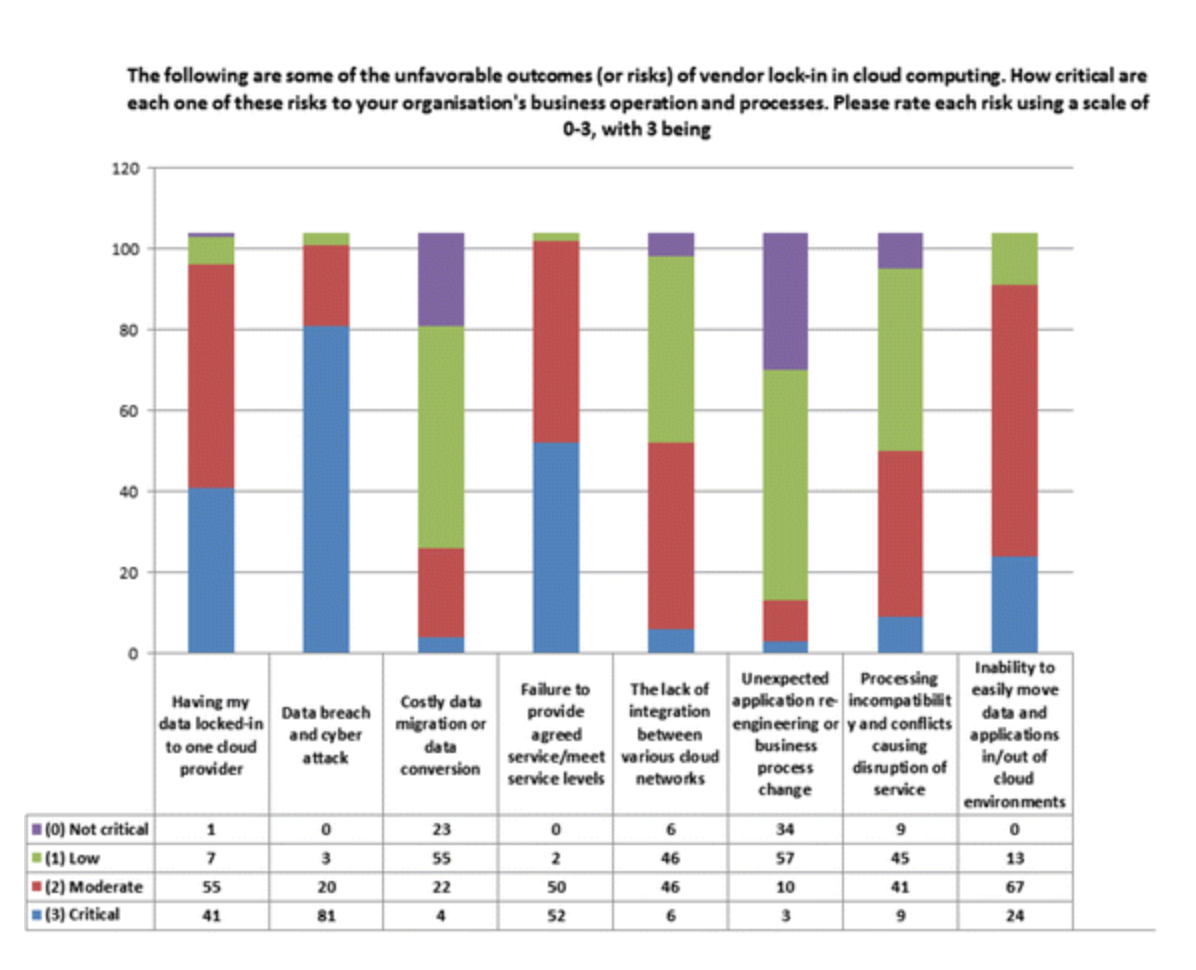
Fears of cloud vendor lock-in stem from a number of places.

First, it’s the loss of control over the data and infrastructure that power business’ applications. Not having complete control over aspects like security, uptime, and overall infrastructure management can be a scary thing.

Next, it’s the dependence on a single vendor for so many critical needs. Your servers, data, networking, user management, and much more are in the hands of one company, so the dependence on your provider is huge. And if something goes wrong, it can be very detrimental to your business.

Also, there may be fear that one cloud provider might not meet your current or future needs. Your CSP might fail to meet service level agreements or incur a data breach, and you’ll be forced to rethink your relationship.

Even worse, the risk of that vendor going out of business is something you’ll have to account for.

The difficulty and cost of switching to a new vendor looms large in every IT manager’s mind when deciding to move to the cloud and selecting a cloud service provider.

Vendor lock-in fears. Graph courtesy of Journal of Cloud Computing,  
<https://doi.org/10.1186/s13677-016-0054-z>

**Types of vendor lock-in risks**

The issue with vendor lock-in is the difficulty in moving to another cloud service provider if something goes awry. You hope that this never has to happen, but it’s a possibility.

There are four primary lock-in risks that you’ll take working with a single cloud provider. These include:

* Data transfer risk
* Application transfer risk
* Infrastructure transfer risk
* Human resource knowledge risk
* Data transfer risk

It is not easy to move your data from one CSP to another.

A myriad of questions will arise during a data migration process, such as:

Who is responsible for extracting the data from the cloud databases and data warehouses?

In what format will the data be? Will that format work with the new cloud provider, or will significant changes need to be made to the data?

How can the data be transferred without loss of application functionality?

How long will it take and how much will it cost to move all of this data?

While some industry groups have tried to create standards for data interchange, sometimes it’s difficult for companies to implement them due to their unique business requirements.

**Application transfer risk**

If you build an application on one CSP that leverages many of its offerings, the reconfiguration of this application to run natively on another provider can be an extremely expensive and difficult process.

For instance, let’s say you’ve developed a business intelligence platform on Microsoft Azure. You leverage basic cloud services like compute, storage, databases, and networking. But the app also includes Azure’s machine learning, data lake analytics, and bot services.

**Can you imagine all the changes you’ll have to make to your application if you had to move this to another CSP?**

One reason for this difficulty is a lack of standard interfaces and open APIs. Every CSP has their own proprietary specifications and standards, which make it very tough to move from one to another.

**Another reason is that technology and customer needs change so rapidly.**

You know first hand that your customers and partners continuously demand changes and improvements to your product. The faster that you add and edit features of your cloud-native application, the deeper entrenched you get with your CSP, and the tougher it will be to move to another cloud vendor.

**Infrastructure transfer risk**

Every major CSP does things a little bit differently.

Virtual machine formats and their associated pricing vary from vendor to vendor, making it difficult to ensure that you have the appropriate resource usage and cost savings if you switch providers.

Database offerings and formats may differ as well.

And one cloud provider may have more attractive offerings in certain infrastructure components, while lacking in other services that you may need.

These differences in the underlying infrastructure result in difficulties moving from one cloud service provider to another.

**Human resource knowledge risk**

If you’ve been working with a single CSP, your IT team has likely gained a lot of institutional knowledge about that provider’s tools and configurations.

If you have to move your applications to another CSP, it will take time for your engineers to ramp up their knowledge of the new cloud platform. They’ll have to learn about new infrastructure formats, implementation processes, and more.

Additionally, any newly required certifications will take a long time to earn.

The knowledge risk is a factor that isn’t often thought about, but is just as important as the risks highlighted above.

**Steps to take to avoid vendor lock-in**

The risks that you take with having all your data, applications, and infrastructure with one cloud provider seem ominous. But there are a few things that you can do to ensure that your vendor lock-in risk is minimized.

1) Do your due diligence

Before you select your CSP, you should thoroughly vet that they will give you everything that you need to run your applications reliably.

Your CSP selection process should look something like this:

Determine your goals of migrating to the cloud

Assess your current IT situation, including a thorough audit of your current infrastructure and cost and resources levels

Select the type of cloud environment needed – public, private, or hybrid?

Determine the specific cloud components necessary

Choose the right cloud provider for your situation

You should consider all of the CSPs’ offerings to see if they match your needs, look at the different pricing models to determine the cost savings you can realize, understand their service level agreements, consider their data transfer processes and costs, and get to know other companies similar to yours with whom they’ve worked.

A deep understanding of your potential CSP is critical in mitigating the risk of vendor lock-in.

Thinking about migrating to the cloud? [Click here to grab our eBook, ﻿﻿﻿10 steps you have to take to ensure a successful migration to the cl﻿oud, and kickstart your migration!](https://www.thorntech.com/cloudmigrationebook/)

**2) Plan early for an exit**

It’s kind of like a cloud pre-nuptial agreement.

It might be weird to plan for an exit before you even “get married” to your cloud provider, but it’s an important step to protect your company in case things go south.

While you plan your implementation strategy, include an exit plan and potential costs. And don’t plan out further than a couple of years; doing so may hamper your flexibility in migrating to another CSP if things go wrong.

**3) Design your application to be loosely coupled**

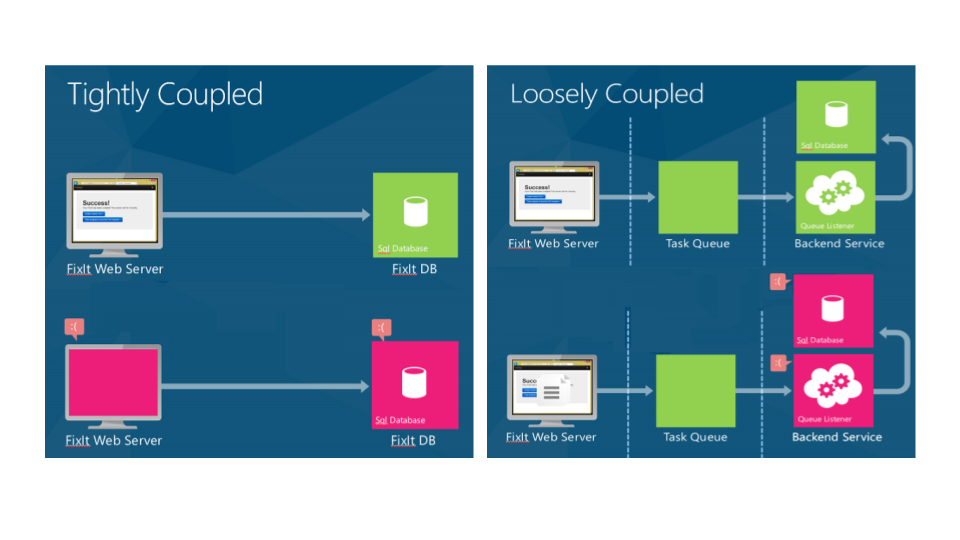
To minimize the risk of vendor lock-in, your applications should be built or migrated to be as flexible and loosely coupled as possible.

Cloud application components should be loosely linked with the application components that interact with them.

You can do this by incorporating REST APIs with popular industry standards like HTTP, JSON, and OAuth to abstract your applications from the underlying proprietary cloud infrastructure.

Also, any business logic should not only be separated from the application logic, but should be clearly defined and documented. This will avoid the need to decipher business rules in case a migration to a new CSP occurs.

Not only does this reduce the level of lock-in to a single vendor, but it also gives your application interoperability that’s required for fast migration of workloads and multi-cloud environments (more on this later).



Tightly and loosely coupled apps. Photos courtesy of Microsoft

**4) Maximize portability of your data**

Data is one of the biggest sticking points in cloud migrations, as different formats and models can cause portability issues.

[The Open Data Element Framework](https://publications.opengroup.org/c163) was created to help standardize the documentation, categorization, and indexing of data, and the [Cloud Data Management Interface](https://www.snia.org/cdmi) helps define how to create, retrieve, update and delete data elements from the cloud.

Unfortunately these standards aren’t always well understood, accepted, nor applied.

To maximize the portability of your data, avoid proprietary formatting. Describe data models as clearly as possible, using applicable schema standards to create detailed computer- and human-readable documentation.

Additionally, you should ensure that your cloud provider provides a way for you to extract data easily and economically.

Data lock-in is probably the most difficult risk to mitigate, so taking these steps will go a long way in easing your data’s transition from one CSP to another.

**5) Consider a multi-cloud strategy**

More businesses are moving to a multi-cloud environment, where you can leverage multiple CSPs to power your applications.

For example, you might use Amazon EC2 for your compute power and Redshift for your data warehouse while using IBM Bluemix’s Watson as your artificial intelligence platform.

By going multi-cloud, you become less dependent on one CSP for all of your needs. Another benefit is that you can cherry-pick offerings from each cloud provider so you can implement best-of-breed services into your applications.

There are some cons to a multi-cloud approach, such as an increased burden on development teams, more security risk, and others. (See here for an [in-depth list of the pros and cons of multi-cloud environments](https://www.thorntech.com/2017/05/pros-cons-going-multi-cloud/).)

But you may find that it’s a viable option to mitigate vendor lock-in.

**6) Implement DevOps tools and processes**

DevOps tools are increasingly being implemented to maximize code portability.

Container technology provided by companies like Docker and CoreOS help isolate software from its environment and abstract dependencies away from the cloud provider. And since most CSPs support standard container formats, it should be easy to transfer your application to a new cloud vendor if necessary.

Additionally, configuration management tools like Chef and Puppet help you automate the configuration of the infrastructure on which your apps run. This allows you to deploy your application to diverse IT environments, which can reduce the difficulty of moving to a new CSP.

These technologies reduce the lock-in risks that stem from proprietary configurations and can ease the transition from one CSP to another.

**4.Identify various reasons for Changing providers expectations in portability and interoperability**

Interoperability and Portability Scenarios

This section leverages a set of scenarios to describe interoperability and portability considerations and requirements including recommendations on how to address them:

* Customer switches between providers for a cloud service
* Customer uses cloud services from multiple providers
* Customer links one cloud service to another cloud service
* Customer links in-house capabilities with cloud services
* Migration of customer capabilities into cloud services

These scenarios are described in the sections that follow. Note that the scenarios focus primarily on public cloud since this deployment model presents the greatest interoperability and portability challenges to customers.

**Scenario 1: Customer Switches Providers for a Cloud Service**

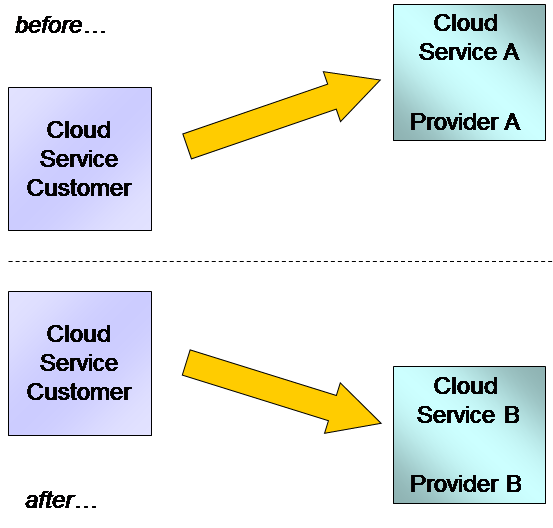
This scenario addresses the straightforward case of a cloud service customer currently using a cloud service of cloud service provider A, who wishes to switch to using an equivalent cloud service of cloud service provider B. This use case is key from the perspective of the customer since it is essential to enabling customers to take advantage of the marketplace in cloud services and avoid the issue of lock-in to a single cloud service provider.

Figure 3: Customer Switches Providers for a Cloud Service

While this scenario is outwardly simple, the reality is that this scenario touches on many of the issues associated with both interoperability and portability. The exact set of issues will vary depending on the nature of the cloud service.

Interoperability Considerations

For SaaS cloud services, the application belongs to the cloud service provider.[2](#_1fob9te) In this case, moving from provider A to provider B does not involve porting the application as the application may be completely different between the two cloud service providers. What is important in the SaaS case is the compatibility of the functional interface for the application – in particular, any user interfaces presented to end users and also any APIs made available for use by customer applications. It is probably unrealistic to expect that user interfaces will be identical for cloud service A and cloud service B, however, it is reasonable to expect that similar functionality will be presented in a broadly similar way to reduce the cost and effort of retraining end users.

2 Some SaaS providers make available a PaaS platform alongside their SaaS services specifically to allow customers to write and run code which customizes the use of the SaaS applications, and also to develop and run new applications. In such cases, transitioning to a new SaaS service of a different provider can also include code portability concerns. Refer to the PaaS description in this section for interoperability and portability implications for this type of code.

However, any functional APIs made available by a SaaS service are likely to be used by customer applications. These applications will need to deal with the switch from cloud service A to cloud service B. If the APIs are not interoperable, as is quite likely to be the case, then the implication is that any customer applications using the APIs would need to be changed as part of the process of switching from cloud service A to cloud service B (see the Interoperability and Portability Challenges section).

For IaaS and PaaS services, interoperability is not an issue for any functional interfaces offered by the application when moving from provider A to provider B since the customer owns the application and its functional interfaces. Similarly, the user interfaces presented to end users are likely to be the responsibility of the application and will not be directly affected by the cloud service itself. As long as the application can be ported, then the user interface will also port and be available when using cloud service B. However, the cloud service APIs used to upload, deploy, and control the application in the cloud service are an interoperability concern, since tooling used by the customer operations staff uses these APIs and this tooling needs to connect to cloud service A and then to cloud service B as part of the migration (for example, the automation tooling discussed earlier).

Portability Considerations

For SaaS services, it is typical that the format and the content of the cloud service customer data is in the hands of the cloud service provider while the data itself is an asset of the SaaS customer. Thus data portability is a major consideration in moving from cloud service A to cloud service B if the cloud service is a SaaS service. Ideally the data syntax should be the same for both service A and service B. In addition to the format, the data content (extent and semantics) should be the same for service A and service B. Data portability can still be achieved if the syntax is different between service A and service B, since there are straightforward standard tools that can be used to perform some data transformations. Where there are not standard tools, it may be possible to build a custom tool. Differences in the extent or in the semantics of the data are much more serious and could be a major barrier to achieving data portability.

For IaaS services, the data syntax and data semantics for cloud service customer data is usually in the hands of the customer, since the facilities provided by the cloud service are typically relatively low level, such as providing volumes for binary file or object storage (i.e., the cloud service does not know or care about the detailed syntax and semantics of the customer data). As a result, data portability is not likely to be a major concern for IaaS services.

Similar considerations can apply to data portability for PaaS services, but the situation is often more complex. For customer data, the PaaS service may provide instances of databases ready-to-use, in which case the actual database(s) provided may be sensitive to the data syntax of the customer data, although there are generalized formats (CSV, XML, etc.) which are supported by many types of database. How data is loaded into a PaaS cloud service and how it is retrieved needs to be examined by the customer and migration from cloud service A to cloud service B involves data portability questions that need to be answered.

For IaaS and PaaS cloud services, the application belongs to the customer and the question of application portability is of utmost importance. What does it take to move the application from cloud service A to cloud service B? The first question relates to application syntactic facet: what format each cloud service accepts for the application artifacts– for example, do they accept the same format of VM image or container image? Application instruction facet asks whether the machine architectures for the two services are the same (i.e., can the target deal with the application instruction sets). Finally, there is the question of whether the target environment can support all the application dependencies, such as runtimes, operating systems, and so on.

In the ideal case, application portability implies that the application artifacts that run on cloud service A will run on cloud service B without any changes. There may be cases where application portability cannot be achieved without some changes. In this case, it is the amount of change and the nature of the change that must be considered. Rebuilding the application code, possibly against a different version of the operating system or a different version of the libraries used by the code, may be simple and low cost. Redesigning the application code to adapt to changed interfaces is likely to be more costly and is less desirable.

For PaaS services, the considerations for application artifacts portability can be much more complex. The application dependencies can consist of a substantial stack of software with many APIs which are used by the application artifacts. In addition, many capabilities can be presented to the application as services of various kinds (databases, messaging, rules engines, etc.). The application may have dependencies on a particular set of services, via their APIs, and it is vital to know that the set of services available within cloud service A is matched by the set of services available in cloud service B.

Admin Interface Considerations

Cloud service admin interfaces, which are used to monitor and manage applications, are significant for all forms of cloud services. These admin interfaces may involve web applications or other visual interfaces and may also involve APIs.

Moving from cloud service A to cloud service B requires that the admin interfaces are compatible (particularly in the case of visual interfaces) and also interoperable (particularly in the case of APIs).

It may be the case that the admin interface is divided into separate sections dealing with particular capabilities – for example, monitoring and reporting capabilities may be delivered by one interface, while management and administration capabilities may be delivered by a different interface.

One approach is for the cloud service customer to use a management solution that has adapters or connectors for the different cloud service offerings, thus dealing with any differences between cloud service A and cloud service B.

Business Interface Considerations

The business interfaces apply to all forms of cloud services and include the capabilities relating to subscription management, billing and invoicing.

Moving from cloud service A to cloud service B requires compatible and/or interoperable business interfaces to ensure that the tools or program components used by the cloud service customer for business capabilities can be used successfully following the move.

Security Considerations

Security aspects of the cloud service include authentication and authorization of users and administrators of the cloud service, the configuration and operation of encryption for data stored within the cloud service and transmitted to and from the cloud service, firewalls and the configuration of other security capabilities.

The security aspects have a number of parts. Some parts apply to the running of the cloud service itself, other parts consist of the administration of the security components, including setting up or modifying user identities and the capabilities they are authorized to use. It is important to check that equivalent security capabilities available for cloud service A are available for cloud service B.

The support of some technologies by the cloud services makes the transition from one cloud service to another one potentially simpler. An example is the support of third party Id and Access Management function where it is possible that the IdAM system could be one installed and operated by the cloud service customer and where that system is used by both cloud service A and cloud service B with few changes - not requiring the porting of a large set of data about users and not requiring the need to change the interface used to administer the user data. Other standard technologies that can help include the increasingly common support of OAuth 2.0 and OpenID that permit use of commonly available ID and access management services across a range of cloud services.

**Recommendations:**

For SaaS, ensure user interfaces, APIs, protocols and data formats are well-defined for cloud services. Whenever possible, insist on standard APIs, protocols, and data formats.

For PaaS, ensure that the application dependencies are based on open technologies to increase the number of viable alternative cloud service providers which can facilitate migration if a change in provider is warranted.

For IaaS, ensure that the cloud service accepts standard or widely accepted application packaging formats such as OVF and Docker and that any interfaces and APIs are open and/or standard.

Insist that your cloud service provider supports key open technologies (open standards and/or open source) for admin and business interfaces.

Leverage the support of third party ID and Access Management functionality to authenticate and authorize access to cloud services.

Scenario 2: Customer Uses Cloud Services from Multiple Providers

This scenario concerns the case where a cloud service customer uses one (or more) cloud services from cloud service provider A and one (or more) cloud services from cloud service provider B. The cloud service(s) from provider A might be equivalent in functionality to the service(s) from provider B, or they may have different functionality, depending on the business needs that the customer is aiming to satisfy.

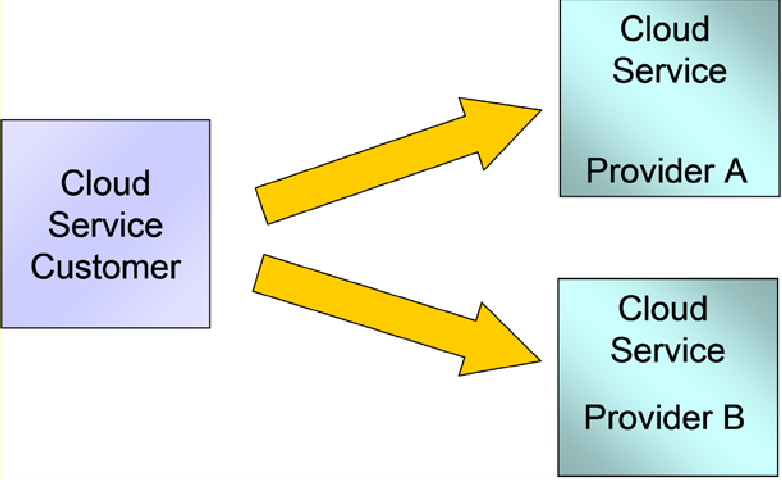


Figure 4: Customer Uses Cloud Services from Multiple Providers

An example where the cloud services would have equivalent functionality is the case where the customer uses two providers to get resilience (i.e., continue to access necessary computing resources in the case where one of the providers has a service outage). A different example is where the best SaaS service for one capability is offered by provider A while the best SaaS service for a second capability is offered by provider B and the customer wants to use both capabilities to satisfy business needs.

This scenario is mainstream. Enterprises have multiple choices when it comes to fundamental cloud services as well as a choice of unique or niche capabilities. Some cloud vendors may provide unique capabilities, some generally useful and some tailored for specific industry types A best of breed cloud implementation enables organizations to choose the best cloud service for each particular capability. From an interoperability and portability perspective, this scenario touches on many of the issues discussed in Scenario 1 above. The use of multiple providers has grown along with the adoption of open standards; however there is still no assurance that standards implementation will be consistent.

Interoperability Considerations

Whether the functional interface of cloud service 1 from Provider A needs to be interoperable with the functional interface of cloud service 2 from Provider B depends on whether the two cloud services are dealing with equivalent functionality or different functionality.

In the case where the cloud services deal with equivalent functionality then it is likely that the same customer components will interact with both services. As a result, it is best if the two services use the same or interoperable interface(s). If services with equivalent functionality do not have interoperable interfaces then customer components will need to be updated to support the two interfaces – not an ideal situation.

When the two cloud services deal with different functionality, the need to use the same or interoperable interfaces is lower. Indeed, it is probably not reasonable to expect interoperable interfaces. However, there may be aspects of the two functional interfaces that should be based on the same technologies. One example is the technology used for Identity and Access Management, since this is a common feature of most cloud services.

Just as there is no assurance of standard interfaces between cloud service providers, neither is there assurance of consistency of cloud service agreement language, types, or terms and conditions. As complex, hybrid cloud architectures become the norm for business critical systems, identifying gaps and assuring equivalence of service availability, incident response, and compliance between cloud service providers is important for smooth operations. At minimum, a RACI chart (responsible, accountable, consulted, informed) for each integration point should be maintained. See the CSCC Practical Guide to Cloud Service Level Agreements [3] and Practical Guide to Hybrid Cloud Computing [5] for guidance. Also refer to ISO/IEC 19086-1:2016 which “seeks to establish a set of common cloud SLA building blocks (concepts, terms, definitions, contexts) that can be used to create cloud Service Level Agreements (SLAs)”. Table 8 in ISO/IEC 19941 can help you identify critical focal points for review of agreements.

Portability Considerations

Data portability can be a requirement between cloud service 1 and cloud service 2. It is usually not a significant issue for IaaS services, since the cloud service customer is typically in control of the data syntax and semantics used for these services. For PaaS services, the cloud service customer usually has a lot of control of the data formats, but this may be limited where the PaaS services make use of particular database technologies – for example, in the case where cloud service 1 uses one database technology while cloud service 2 uses a different database technology.

For SaaS services, data portability can be a significant issue where cloud service 1 is equivalent to cloud service 2 (as in the case of scenario 1). Even in cases where the two SaaS services are not equivalent in functionality, there may be a need to use some data extracted from cloud service 1 in the operation of cloud service 2. In this case, it is best if the data extract has the same format, extent and semantics for cloud service 1 and cloud service 2. If not, then some form of data transformation may be required in order for the customer to use both cloud services successfully.

For IaaS and PaaS services which involve the deployment of application code into the cloud service, application portability between the cloud services is important. This is necessary where the same application gets deployed to cloud service 1 and cloud service 2. However, it can also be important in cases where different application code is deployed to the different cloud services, since the developers may well want to use the same knowledge, technologies and tooling to build both applications, which may be difficult if the capabilities offered by the cloud services differ substantially. For IaaS services, VM image formats and container formats are an important component of application portability. A portable VM image format improves portability across different service providers. An example of a standard VM image format is Open Virtualization Format (OVF).

Admin Interface Considerations

Administering cloud services from multiple cloud service providers used to mean interacting with two or more sets of admin interfaces to monitor and manage each of the cloud services. If the interfaces were not interoperable, the result was duplication of effort. There might be multiple user accounts to manage, multiple sets of access controls to maintain, and multiple administration portals to learn and use. Even a relatively simple task, like checking to see which machine instances are running, can become a multi-step process. Cloud Management Platforms (CMP) able to support hybrid cloud architectures are becoming more common, as are tools that provide rules-based brokerage. A detailed explanation of CMPs is given in the CSCC paper, Practical Guide to Cloud Management Platforms [7].

CMPs support the ideal case, in which the customer staff use one set of tools and applications to monitor and manage all cloud services, irrespective of which cloud service provider is being used. In the ideal case, any admin APIs are interoperable and any admin visual interfaces are composable - for example, based on standard web technologies that can be integrated on a single browser. CMPs typically have a series of adapters which enable them to interact with a variety of different administration interfaces offered by different cloud service providers.

Business Interface Considerations

Using cloud services from provider A alongside cloud services from provider B implies a requirement on the cloud service customer to integrate the capabilities relating to subscription management, billing and invoicing. This is necessary in order to keep a good grip on expenditures and is also a requirement for allocating what are likely to be dynamic costs to the right internal budgets and projects. The CMP and cloud brokerage mentioned above frequently support the aggregation of business interfaces of multiple cloud services.

In the ideal situation, the business interfaces offered by Provider A are compatible and interoperable with those offered by Provider B. This enables the customer to use a single set of business tools to manage the usage of all the cloud services. Where the business interfaces are not all interoperable, then the customer should look for business tools such as CMPs that can perform mapping or transformation of each of the different business interfaces offered by the different providers.

Security Considerations

Using multiple cloud services from different providers requires various aspects of security to be carefully assessed. One aspect pertains to the running of the cloud service itself, other aspects pertain to the administration of the security components, including setting up or modifying user identities and the capabilities they are authorized to use or implementing directory federation. Additional complexity may be added if all or some portion of the organization’s Identity and Access Management is running on or is itself a cloud service. The ideal situation is that these aspects are interoperable between the different providers, enabling a single set of tooling and procedures to be used by customer staff. In some cases this can be achieved by delegation of capabilities from the cloud service to the customer, such as the support of third party Id and Access Management or directory federation, which may be a system owned and run by the customer. In other cases, the provision of capabilities using a standard interface and standard technology is a useful approach – for example, with respect to data encryption.

**Recommendations:**

Refer to recommendations for Scenario 1.

For SaaS, ensure user interfaces, APIs, protocols, and data formats are identical (or have a clear mapping) for equivalent functionality running on different cloud service providers.

Consider implementing an Enterprise Service Bus (ESB) to perform interface, protocol, and data transformations to address differences between cloud services from different providers.

For PaaS, ensure that the application dependencies (web server, database server, etc.) supported by different cloud service providers are compatible.

Consider the use of an intermediary, an “inter-cloud provider”, to help address and solve the issues of integration, interoperability and portability of multiple cloud services. Alternatively, make use of tools such as CMPs that can integrate with multiple cloud services.

Scenario 3: Customer Links One Cloud Service to Another Cloud Service

In this scenario, the customer uses two cloud services, but one of the cloud services is used directly by the other one. The ability to link cloud services together in support of a single application or an integrated set of applications is a useful approach where different cloud services can each provide specific capabilities which can be even more effective when linked together. Business technology leaders now understand that a single cloud service provider may be challenged to meet the needs of their entire organization.

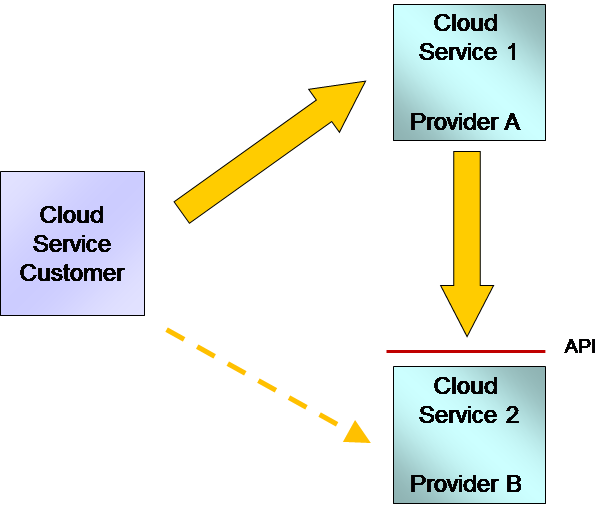


Figure 5: Customer Links One Cloud Service to Another Cloud Service

A cloud service linking to another cloud service as shown in Figure 5 can be illustrated using a SaaS-to- PaaS example. In this example, the SaaS application delivers the functionality/feature sets that an organization needs as a business solution; however, the SaaS application may not have the advanced functionality to deliver analytics and business intelligence. The organization can leverage additional capabilities by using a PaaS service from another cloud service provider and developing a custom analytics application that consumes data from the SaaS solution via an API, combining it with other data sets to drive additional revenue or market differentiation.

Interoperability Considerations

Linking one cloud service with another cloud service requires that the second cloud service has a well- defined API that the first cloud service can utilize remotely. It is assumed that the cloud services leverage SOA techniques (such as REST interfaces, stateless interactions) to facilitate invocation. The impact of performance variations on the use of the API due to network constraints should be considered.

The categories of the two cloud services influence the nature of the connection between the two services. When the first cloud service is either an IaaS or a PaaS service, the application code running in that service belongs to the cloud service customer and the main concern is if the code can successfully utilize the API of the second cloud service. It is likely that the IaaS or PaaS platform will be capable of supporting invocations of remote service APIs, although there may be issues to consider relating to security capabilities, such as authentication credentials and encryption technologies.

If the first cloud service is a SaaS offering, the situation is more complex as the application code belongs to the cloud service provider. In this case, the application code of the first cloud service must be structured to enable the use of the API of the second cloud service. For this to be possible, it is a likely requirement that the API be standardized in some way (possibly a de-facto standard rather than a formally standardized one). In addition, there may be a need for the customer to configure the first cloud service to use the second cloud service.

Regarding the second cloud service, if it is an IaaS or PaaS service, then the code belongs to the customer and the customer is in control of the API which it offers to the first service. When the second cloud service is a SaaS service and the API is dictated by the cloud service provider, it is important that the API is offered using standard technologies. Ideally, the whole interface should be defined by a standard, but in the common case where it is not, the basic protocols should be standard (e.g., use of REST/JSON or REST/XML protocols and data formats). When the API is not standardized, which is common, there is the danger that the customer will get locked-in to the particular cloud service and find it difficult and expensive to move to a different provider. As a result, the customer should consider mitigation techniques (see the Interoperability and Portability Challenges section).

Portability Considerations

This specific scenario does not involve moving or transferring either applications or data from one system to another, therefore, portability is not an issue. Refer to scenarios 1 and 2 for portability considerations relating specifically to cloud services.

Admin Interface Considerations

For the customer to administer cloud services from different providers means that multiple admin interfaces need to be taken into account. Administering multiple providers can mean duplication of effort and the need to adapt to different interfaces from each provider. Refer to scenario 2 above for a thorough discussion on admin interface considerations.

Business Interface Considerations

For the customer to utilize cloud services from different providers requires that the customer deal with the business interfaces offered by each cloud service provider. Refer to scenario 2 for a thorough discussion on business interface considerations.

Security Considerations

From a security perspective, linking cloud services from different providers requires not only appropriate security measures for each cloud service individually, but also requires security measures to be applied to the connection from the first cloud service to the second cloud service. The connection between the two services will require appropriately strong ID and Access Management capabilities to be applied. This will require that the technology for this is supported at both ends of the connection (i.e., cloud service 1 must be able to send appropriate credentials when it invokes cloud service 2). It may also be required to encrypt the data sent between the two cloud services. This requires that they mutually support encryption technology of the right strength.

The support of third party ID and Access Management functionality, where it is possible that the IdAM system could be one installed and operated by the cloud service customer and where that system is used by the different cloud service providers, would significantly reduce the need to port/duplicate user security information.

**Recommendations:**

Refer to recommendations for Scenario 2.

Ensure that services provided by the cloud service providers leverage SOA design principles and can utilize and expose APIs to enable interoperability.

Consider the use of an intermediary (an “inter-cloud provider”) to help address and solve the issues of integration, interoperability and portability of multiple cloud services.

Ensure that the security technologies supported by the second cloud service are usable by the first cloud service when it uses the capabilities of the second cloud service.

Scenario 4: Customer Links In-house Capabilities with Cloud Services

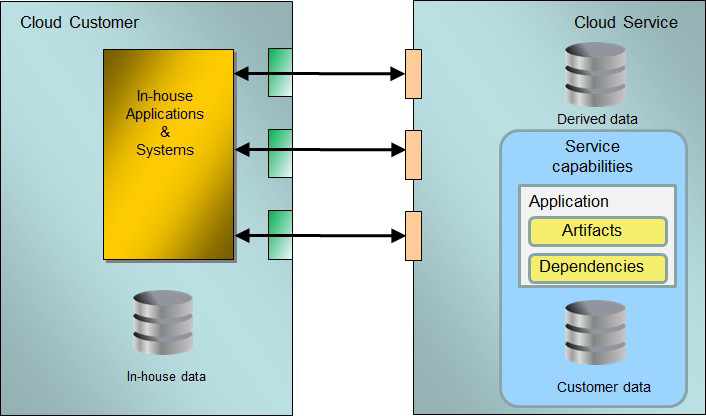
As more enterprises consider their cloud computing strategy, they will inevitably face the challenge of addressing how they will leverage their existing in-house IT investment with their newly adopted cloud services. In addition, enterprises will also have to assess how other in-house capabilities will be leveraged in their cloud strategy. These capabilities include people, processes, and of course technology. Developing an ‘in-house’ view of cloud adoption based on these critical criteria is the challenge that enterprises face during the early cloud computing adoption phase and continues as more workloads and projects are committed to cloud services.

Figure 6: Linking In-house Capabilities to Cloud Services

As new cloud services are deployed, the need to connect them with various on-premises applications and systems becomes important, as illustrated in Figure 6. Cloud service owners need to understand the impact of these connections and address it. Integration between applications is typically classified into three types:

Process (or control) integration, where an application invokes another one in order to execute a certain workflow

Data integration, where applications share common data, or one application’s output becomes another application’s input

Presentation integration, where multiple applications present their results simultaneously to a user through a dashboard or mashup.

The purpose of these integrations may be to perform an end-to-end workflow that crosses the boundaries between multiple business capabilities or systems (for example, entering a transaction in an accounts receivable system when a customer places an order in an e-commerce application). Another form of integration is when the cloud service must continue to be monitored and managed by an existing suite of on-premises IT tools.

Refer to the CSCC whitepaper, Migrating Applications to Public Cloud Services: Roadmap for Success [8] for a detailed discussion on integration considerations for connecting cloud services with on-premises services.

Interoperability Considerations

Linking in-house capabilities with cloud services requires that on premises functionality and data needed by the cloud service are clearly identified and vice versa. For each of these functions and data sources, there must be a well-defined API in place that can be utilized remotely. If the on premises applications and cloud services leverage SOA techniques (REST interfaces, stateless interactions, etc.) then the integration effort should be reduced. If not, the impact of redesigning the applications and services to provide suitable interfaces could be significant. In addition, the impact of performance changes due to network constraints need to be considered as part of the integration.

For SaaS services, the cloud service provider must make available the necessary information (such as API descriptions and security requirements) to address functional integration requirements since the provider controls the cloud service. For PaaS services, the customer should be able to address most of the functional integration requirements, since the app code running in the cloud service is controlled by the customer, although provider assistance may be needed to address middleware specific integration requirements. For IaaS cloud services, the cloud customer should be able to fully address all functional integration requirements since it is expected that the interfaces offered and used by the app code running in the cloud service are controlled by the cloud service customer.

Portability Considerations

This specific scenario does not involve moving or transferring an entity from one system to another, therefore, portability is not an issue. However, it is recommended that any design required considers interoperability and portability to minimize impacts should migrations take place in the future. Refer to scenarios 1 and 2 above for portability considerations relating specifically to cloud services.

Admin Interface Considerations

Linking cloud services with on premise capabilities will require the integration of administration platforms and processes. As part of the design of this hybrid platform solution, careful consideration needs to be given to the design of the integration and deployment process within the Software Development Lifecycle (SDLC). This is necessary to enable appropriate management of change across the service providers (internal and external).

In addition to changes that are likely to be required to the SDLC process, system monitoring, management and support models (for example, code configuration management) should link in-house technology with cloud-based technology to formulate a unified solution. Techniques, such as Continuous Integration and Continuous Deployment (CI/CD) are commonly used in these circumstances.

Existing support models and processes must be revisited to determine if there is a need to modify the current processes to address potential disparities between the on premise components and the cloud services. Critical functions such as backup/recovery, disaster recovery, fail-over and high availability requirements must be re-examined with cloud services as a part of the scope and planning process.

Maintenance plans and support processes must also be re-visited to determine how the integrated solution will be monitored, managed and maintained on an ongoing basis.

Ideally, a consolidated monitoring and management platform is possible as production environments require constant maintenance and support to ensure performance, security, and availability to meet the expectations of the business units and their user base. Can the management tools used in-house still be used, or is it necessary to adapt to new monitoring and management facilities supplied by the cloud service? The answer to these questions will depend on the admin interface offered by the cloud service(s) and in particular whether the interface conforms to existing interoperable standards, which can be used by the in-house tools and systems. It may be necessary to consider adapter code or the use of mapping capabilities such as an ESB to facilitate the integration in cases where the cloud service admin interface is not directly interoperable with the in-house systems.

Business Interface Considerations

Business interfaces involve capabilities relating to the business aspects of the cloud service including subscription information, billing and invoicing. Many organizations do not adopt internal service billing (for many different reasons), but public cloud services may require the implementation or amendment of such systems.

As a result, the introduction of cloud services may require customers to adopt new business-related user interfaces and/or APIs defined by the cloud service provider and possibly involving the purchase or development of in-house systems for the management of business capabilities. In cases where on premise business management systems do exist, cloud service customers may need to adopt new tools and systems, or acquire adapters to match existing systems to the interfaces offered by the cloud service provider. Wherever possible, standard methods or data formats should be used to share data between providers and customers.

Security Considerations

From a security perspective, linking in-house capabilities with cloud services may require strong authentication and authorization services to ensure proper access is being granted to potentially sensitive services and data. The support of third party Identity and Access Management (IdAM), where it is possible that the IdAM system used to control access to the application running in the cloud service could be an existing one installed and operated by the cloud service customer, significantly reduces the need to port or duplicate user security information. The analysis of the IdAM system should include not just the application layer, but also access to the underlying infrastructure services (network, IaaS, PaaS or SaaS) to ensure no security holes are opened.

Given that the connection to the cloud service is likely to traverse the public internet, consideration should be given to what encryption should be applied to that connection and whether any customer data stored in the cloud service should also be encrypted. It is worth considering carefully whether there are any regulatory or compliance rules that specifically define the need or level of encryption to be used.

Not only must the cloud service support the necessary encryption capabilities, but items such as certificates and encryption keys must be managed in a way compatible with the customer’s security policies. Technologies such as Virtual Private Network can also be considered for securing access to the cloud service.

In the case where the cloud service accesses APIs or data that are supplied by in-house systems, there is a further, very significant set of security considerations that must be addressed. The APIs and/or data access implies the creation of new, publicly exposed interfaces, which need careful control since they offer a potential new attack surface. Access control, firewall configuration, denial-of-service countermeasures, and encryption techniques must all be considered in relation to these new APIs. The deployment of suitable API Management capabilities may be part of the response to these considerations.

**Recommendations:**

Refer to recommendations for Scenario 1.

Ensure that on-premises applications are leveraging SOA design principles and can utilize and expose APIs to enable interoperability with remote cloud services.

Examine whether existing in-house systems are available to deal with the business aspects of using cloud services. If they are not available, consider installing new systems to cover these aspects; if they are available, consider how those systems can connect to the business capabilities of the cloud service(s).

Consider implementing an Enterprise Service Bus (ESB) to perform interface, protocol, and data transformations to address differences between on-premises systems and cloud services.

If cloud service(s) need access to on-premises APIs or data, address the security issues raised by enabling access to these capabilities from the cloud environment - for example, put in place suitable API Management capabilities to prevent unauthorized access.

Scenario 5: Migration of Customer Capabilities into Cloud Services

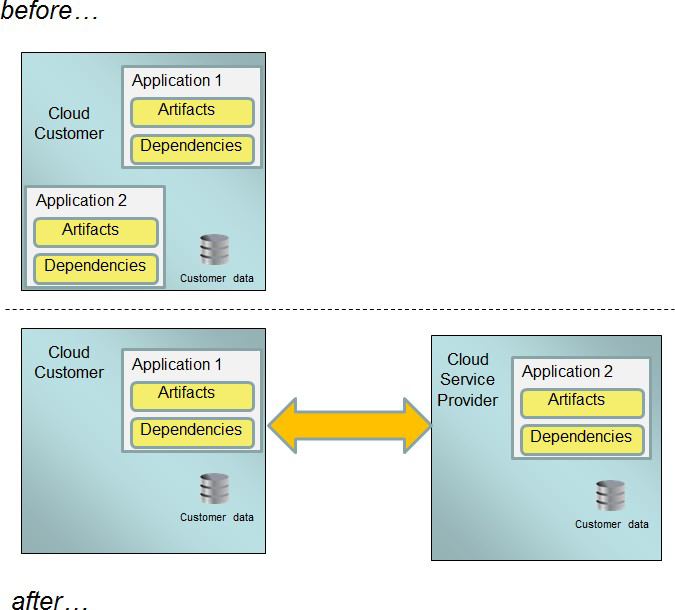
This scenario addresses the case of a customer currently running an application or service on premises who moves that capability to a public cloud environment, as shown in Figure 7. This use case is a key one from the perspective of the customer since cloud computing offers several benefits including elasticity, potential cost savings and business transformation improvements.

Figure 7: Migration of Customer Capabilities into Cloud Services

Assessing applications and workloads for readiness for migration to a cloud service is required for organizations to determine which applications and data can – and which cannot – be readily moved to a public cloud environment and what service models (IaaS, PaaS, or SaaS) can be supported. It often makes sense to start with the lowest-risk applications—those with minimal customer data and other sensitive information—or applications that can take advantage of the elasticity of cloud computing.

The table below highlights suitable and less suitable types of applications for migration to cloud computing. Refer to the CSCC whitepaper, Migrating Applications to Public Cloud Services: Roadmap for Success [8] for a detailed discussion on the steps customers should take to ensure successful migration of existing applications to cloud computing.

| Suitable Candidates for Cloud | Less Suitable Candidates for Cloud |
| --- | --- |
| Applications that do not contain extremely sensitive data.  Applications that run across distributed locations to service a distributed user base.  Applications that are required to serve large volumes of data (eg., streaming services).  Applications that are used by a group of mobile workers to manage their time and activity.  Applications that are run infrequently but require significant computing resources when they run.  Development, testing and prototyping of application changes, even if the final applications will be run on your own infrastructure.  Service Oriented Architecture (SOA) applications. | Applications that involve extremely sensitive data, particularly where there is a regulatory or legal risk involved in any disclosure.  Applications that require frequent and/or voluminous transactions against an on- premises database that cannot be migrated to a cloud service.  Applications that run on legacy platforms that are typically not supported by cloud providers. |

Interoperability Considerations

For SaaS cloud services, the application code belongs to the cloud service provider. In this case, migrating an on-premises application or service to a public cloud service provider does not involve porting the application code (i.e., the on-premises application is being replaced by the application offered as a cloud service). What is important in the SaaS case is the compatibility of the functional interface for the application - and, in particular, any user interfaces presented to end users and also any APIs made available to other customer applications. It is unrealistic to expect that user interfaces provided by the SaaS application will be identical to the on-premises application; however, it is desirable for the functionality to be presented in a broadly similar way to reduce the cost and effort of retraining end users.

Moving from using an on-premises application to a SaaS cloud service will likely require end user training and/or modifications to existing business processes, in that the cloud service is unlikely to be an exact match for the original on-premises application. This must be factored in to the planning for the migration.

Any functional APIs made available by a SaaS service should in the ideal case be the same as the interface provided by the on-premises application or service that is being replaced (i.e., the APIs are interoperable). Where the APIs are not interoperable, then the implication is that the customer applications using the APIs will need to be changed as part of the process of migration to the SaaS equivalent. Depending on the extent of customer applications affected and to minimize change, it may be advisable to create a common mapping layer that converts old API calls to the format required by the new SaaS offering or deploy an Enterprise Service Bus (ESB).

In the case of PaaS and IaaS cloud services, the user interfaces presented to end users and the APIs presented to external applications are likely to be the responsibility of the application code - and this code is the responsibility of the cloud service customer. However, these interfaces may be impacted when the on-premises application is migrated to a public cloud service due to the nature of the network connection from the on-premises location to the cloud service. If the on-premises application already leverages SOA techniques (REST interfaces, stateless interactions, etc.) then the impact of migration is minimized. If not, the impact to other applications invoking the migrated application or service could be significant. In addition, the impact of performance changes due to network constraints need to considered as part of the migration.

Portability Considerations

For SaaS services, it is typical that the format and the content of the cloud service customer data is in the hands of the cloud service provider. Thus, data portability is a major consideration when migrating an on-premises application to a public cloud SaaS service. Ideally, the data format and data content (syntax and semantics) should remain the same after SaaS migration. Data portability can still be achieved even if the data formats are different between the on-premises application and the SaaS application since there are straightforward standard tools that can be used to perform data transformations – and where there are not standard tools, it may be straightforward to build a custom tool. Differences in the extent or in the semantics of the data are much more serious and could be a major barrier to achieving data portability, particularly if the SaaS application requires more data than is available from the on-premises application, or if the SaaS application has a different meaning for some of the data.

The impact of migrating an on-premises application to a PaaS cloud service should have minimal impact on data portability unless data format and data content need to be altered as part of the migration.

There may be exceptional cases where a database provided as part of the PaaS environment may be sensitive to the data format of the customer data, although there are generalized formats (CSV, XML, etc.) which are supported by many types of database. How data is loaded into a PaaS cloud service and how it is retrieved needs to be examined by the customer. Migration from an on-premises data store to a PaaS data store may involve data portability questions that need to be answered.

For PaaS migration, since the application code belongs to the customer the question of application portability becomes important – what does it take to move the on-premises application to the PaaS cloud service? One of the most important factors for application portability is represented by the App environment shown in Figure 2 above. In effect this represents the "API" that the cloud service presents to the application code – and the application code must be able to use this API in order for the application to run. In the ideal case, application portability implies that the application code which runs on-premises will run on the PaaS service without any changes (i.e., the App environment is compatible).

If the App environments differ, the application code will need to be changed to account for the differences.

The requirements for migrating an on-premises application to an IaaS service tend to be lower since the entire software stack is migrated: the application code itself, plus any supporting code it requires, potentially including the underlying operating system. To achieve this, it must be possible to package the complete software stack as one or more virtual machine (VM) images, which can then be imported into the cloud service and executed there.

Whether the software stack involved will work in a virtual machine environment may depend on whether there is use of specialized device drivers or hardware devices that are unlikely to be supported by an IaaS cloud service provider; an application depending on these capabilities is not a good candidate for migration. This tends to be an issue for legacy or specialist platforms only.

Assuming the software stack will work in a virtual machine environment, the question of application portability becomes less of an issue for migration to an IaaS service. That is, the on-premises application should run on the IaaS service with few if any changes. For IaaS services, the data format for cloud service customer data is usually in the hands of the customer, since the facilities provided by the cloud service are typically relatively low level, such as providing volumes for binary file or object storage (i.e., the cloud service does not know or care about the detailed format of the customer data). As a result, the impact of migrating an on-premises application to an IaaS cloud service should have minimal impact on data portability. It is important to note that to ensure application performance the database and application code generally need to be migrated together.

One consideration which applies to the migration of customer application code to either a PaaS or IaaS cloud service is whether the code is capable of taking advantage of the capabilities of the cloud service. It may be possible to port the code to the cloud service and get it to run successfully, but to enable capabilities such as elasticity and scaling may require recoding or redesign. However, this may be handled as part of a phased migration process, where full exploitation of the capabilities of the cloud service is part of the later phases, following migration of the basic functionality of the customer application.

Admin Interface Considerations

The management and monitoring of the migrated application running in the cloud service must be considered including the capabilities to deploy and configure the application, modify the resources assigned to the application (CPU, storage, etc.) and monitor the application’s usage and status. Can the tools used in-house still be used, or is it necessary to adapt to new monitoring and management facilities supplied by the cloud service? These considerations could be some of the most significant when migrating an application to a cloud service, since it is likely that the in-house facilities for monitoring and control will not match the equivalent facilities available with the cloud service.

These admin interfaces may involve web applications or other visual interfaces and they may involve APIs. Migrating an on-premises application to a cloud service requires that the admin interfaces are compatible (particularly in the case of visual interfaces) and also interoperable (particularly in the case of APIs). If not, significant and potentially costly adaptation will be required. In the most extreme cases, it may be necessary to change the customer systems used for the various administration tasks, introducing new systems capable of interacting with the facilities made available with the cloud service.

Business Interface Considerations

Business interfaces involve capabilities relating to the business aspects of the cloud service including subscription information, billing and invoicing. It may well be the case that such capabilities do not exist at all for an on-premises application – and even where they do exist, it is likely that the interface(s) provided to support them will not match the equivalent interface(s) made available by the cloud service provider.

Moving an on-premises application to a cloud service provider requires compatible and/or interoperable business interfaces to ensure that the tools or program components used by the cloud service customer for business related capabilities can be used successfully following the move. In the most extreme case, this may involve the customer adopting new tools and systems, or acquiring adapters to match existing tools to the interfaces offered by the cloud service provider.

Security Considerations

Refer to scenario 4 above for a thorough discussion on security considerations.

**Recommendations:**

Refer to recommendations for Scenario 4.

For SaaS, consider compatibility with on-premises applications and the migrated cloud service. Ensure user interfaces, APIs, protocols and data formats are well defined for migrated cloud services. Whenever possible, insist on standard APIs, protocols and data formats.

For PaaS, ensure that the application environment (web server, database server, etc.) supported by the cloud service provider is compatible with your on-premises application environment.

Examine the cloud service provider interfaces for administration and business capabilities and ensure that they can be used directly or integrated with existing or new in-house systems.

**5.Discuss your view on Recommendations all cloud solutions in Compliance and audit**

**Cloud Compliance: Tackling Compliance in the Cloud**

Moving to a cloud environment brings compliance challenges, but they’re not insurmountable.

Most organizations already have started to use virtualization technology or cloud computing. Yet some still may be reluctant to move their mission-critical—tier-1 – applications to these relatively new environments. While the flexibility and cost benefits of virtualization are widely accepted, questions linger on how to adapt to new and different risks. Security and compliance top the list of organizations’ reasons to delay adoption.

Concerns about security in a virtual environment almost always begin with a study of the relationship between guest and host. That is just the tip of the iceberg. In the end a far more comprehensive view of risk management is necessary, which includes virtual machines (VMs), hypervisors, networking, storage and management. From configuration of software-based networking devices to software-based data centers, the process and procedures for managing resources are an important part of an assessment of cloud risk and compliance. An assessor not only will review configuration of the VM and hypervisor technology, but also look at how logical concepts such as port groups, resource pools and clusters are being managed in relation to data flows and business logic.

Let’s take a look at some of the ways virtualization and cloud computing impact compliance and how organizations can tackle [cloud compliance](https://searchcloudsecurity.techtarget.com/news/2240031767/Cloud-compliance-cloud-encryption-top-enterprise-security-concerns) issues.

**Start with a standard baseline**

A good strategy to [manage cloud compliance](https://searchcompliance.techtarget.com/news/2240179327/Product-Spotlight-Compliance-monitoring-tools-for-finance-firms) is to establish a clear and transparent relationship with a cloud service provider. This can be facilitated by standards such as the [SSAE 16](https://searchcloudsecurity.techtarget.com/definition/SSAE-16) SOC 2 or [ISO 27001](http://searchsecurity.techtarget.co.uk/definition/ISO-27001). A framework that both parties can agree on makes it easier to get through the sections to focus on finding resolution in areas of concern. A provider that refuses to provide on-site physical assessments, for example, may not be acceptable to an assessor or a cloud customer. They might be concerned that despite what cloud providers say about identical controls in their many physical locations, which can be verified on paper, the human element of managing controls can still cause controls to drift out of place and warrant on-site audits.

Perhaps the easiest way to work through [cloud compliance challenges](https://searchcloudcomputing.techtarget.com/feature/Provider-user-teamwork-key-in-cloud-compliance) with cloud providers is to approach them first at a technical level and in terms of how compliance has been handled in the past. An operating system has typically been brought into compliance by hardening it to a set of published guidelines. Systems within government must adhere to a set of documented security standards, such as the U.S. Defense Information Systems Agency (DISA) Security Technical Implementation Guide (STIG), or publications from the [National Institute of Standards and Technology (NIST](https://searchsoftwarequality.techtarget.com/definition/NIST)). Systems within a commercial environment may need to be measured against completely different guidelines from the [Center for Internet Security (CIS)](http://www.cisecurity.org/) or by an industry group such as the Payment Card Industry (PCI) Security Standards Council (SSC). Like the ISO and SSAE 16 standards, although with a regulatory authority overseeing their adoption, they can help clarify what exactly has to be done by a provider to achieve compliance.

**Take control of continuous change**

Let’s say that a Windows 7 system on hardware could be configured to meet the CIS Benchmark version 1.2.0 released on March 30. Move that same Windows 7 system from hardware to a VM on a hypervisor managed by a provider and an assessment of compliance for that system can be seriously different. Move it into a cloud environment and it changes again. The operating system itself remains almost identical, but an updated benchmark is required to account for the relationship with the hypervisor and then the systems used to manage hypervisor resources. Consequently, hardening takes on new and different meanings based on virtualization and how it is managed. Why? The flexibility and efficiencies of cloud mean new and different configuration options, which have different risks compared to hardware-based infrastructure.

For example, a hardware-based operating system will have configuration files that define storage. Migration to a virtual machine means the configuration files that describe the hardware move outside the system and onto the hypervisor. The boundaries for a VM are defined by those configuration files. In other words, a Red Hat Enterprise Linux system would normally use a configuration file in the OS (e.g. /etc/fstab) to determine which hardware file systems to mount when it boots. The OS file has to be very particular to equipment it was installed with (e.g. bus type, file system type, partition number). Virtualization, however, will make the same file in the OS generic to reflect the typical—or at least reduced—set of options available from the hypervisor. It then moves the hardware details to a file read by the hypervisor but invisible to the VM’s OS.

In terms of compliance, this means there has to be a shift in how to assess technical controls when looking at a virtual environment. A hypervisor should put a VM in a sandbox, isolated from other VMs. The sandbox is defined in part by how the hypervisor controls access to its hardware. A VM therefore should have no expectation that it can achieve direct hardware access by changing its configuration file; it should only see what it is provided. At a cloud provider level, this means a provider always should be validating configuration information that is uploaded with a VM before allowing that VM to run. A simple failure to validate a VM setting, such as allowing a VM to directly mount hypervisor storage, could potentially compromise other VM data on that hypervisor. Optical drives have little or no need to be connected to a VM in a data center environment, so they usually can be disabled. Likewise, attacks on serial and parallel ports do not work if those ports are disabled.

The key to this example is that a customer will need to know whether a provider validates VMs as well as disables features unused or unnecessary. It is the same concept as traditional compliance requirements—validate input and reduce the attack surface—but applied to the new processes and control points of cloud.

While the requirements in regulations do not yet spell out this level of technical detail for provisioning and de-provisioning systems, they do have language that is relevant and useful to assessors. The [PCI Data Security Standard (DSS)](https://searchfinancialsecurity.techtarget.com/definition/PCI-DSS-Payment-Card-Industry-Data-Security-Standard) version 2.0 states in Requirement 2.2 that a regulated entity must “develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.”

Cloud providers and vendors already are stepping forward to address the language of this [regulatory requirement for standards](https://searchservervirtualization.techtarget.com/tip/Meet-data-center-compliance-standards-in-hybrid-deployments). New security and compliance products, as well as detailed hardening guidelines, address the need for industry-accepted control requirements or recommendations. VMware’s vCenter Configuration Manager (VCM) is the type of tool that customers can request from their cloud providers to get a centralized and continual collection of configuration changes to infrastructure. A unified report will show systems that are out-of-sync with vendor hardening guides, or in violation of policy or regulations such as SOX, PCI DSS, HIPAA and FISMA. An emerging standard called the [NIST Security Content Automation Protocol](https://searchsecurity.techtarget.com/tip/Understanding-SCAP-NIST-guidance-and-using-SCAP-tools-to-automate-security) (SCAP), also supported by VCM, can even provide a detailed guide on current security configuration of operating systems and applications.

**Establish trusted zones**

Software-based networks also can be a sticking point for compliance. Segmentation between VMs, explained above in terms of the hypervisor, also is relevant to the configuration and maintenance of virtual switches. The migration of a VM from one hypervisor to another is often done in the clear for reasons of performance and availability. In other words, the VMs are sent by providers without encryption, so anyone with access to the network would potentially intercept and view or modify data. The memory contents of a VM could be viewed or altered. Confidentiality and integrity both are at risk when this is the configuration.

To reduce the risk of these attacks, the management-related traffic of the hypervisor should be set to isolated and dedicated networks that are non-routable (i.e. no layer-3 route to other networks). The port group should be on a dedicated VLAN. The virtual switch can be shared but the port group VLAN should never have any other VM connected. This also allows for monitoring for that VLAN ID on other port groups. Another option is to further separate the port group with a management-dedicated virtual switch and to monitor the switch for non-management traffic.

Taking this one step further, a management network should be set up at a cloud provider to restrict access only to known endpoints. Although requirements such as PCI DSS do not explicitly state this, the PCI Security Standards Council (SSC) in 2011 made it clear with the publication of its [virtualization guidelines](https://searchcloudsecurity.techtarget.com/news/2240036974/PCI-virtualization-report-cites-challenges-with-PCI-compliance-in-the-cloud) that reducing the management interface attack surface is a best practice. An attacker is likely to target the network to gain privileged access to a cloud provider’s management interface.

That is why the management layer should be protected by giving it a dedicated VLAN for the management port group on a shared virtual switch. Other VM traffic may be allowable on a switch if the port group for the management VLAN is restricted only to management traffic. An additional level of security, such as stateful packet inspection and intrusion detection monitoring, will help further segment the traffic and tends to be required under some regulations such as PCI DSS. An even better step to segment management communication is to move the management VLAN to a dedicated virtual switch that does not allow for any non-management port groups. The network segment also should not be routed except to other isolated and protected management networks.

Another important step in overcoming cloud compliance challenges is related to the human element; The cloud provider’s administrators and users must be trained on policy and procedures. SSL certificates not only have to be carefully managed and secured, but the administrators themselves also have to be vigilant about verifying SSL certificates before entering their passwords. Impersonation of a VMware vCenter Server or vCloud Director with an incorrect SSL certificate would force the client software to display a security warning. An administrator might override the warning if he or she isn’t properly trained to report it and/or investigate the error as a security incident.

**Compliance as a cost-saver**

One of the more interesting effects of cloud environments is that, when engineered properly, they actually can reduce compliance costs while improving security coverage. Anti-malware controls are an excellent example of how automation and consolidation reduce overhead. There is no doubt that antivirus is required under practically every regulation; from SOX to PCI DSS, there is a need to prevent unauthorized code. Requirement 5 of PCI DSS v2 states simply, “Use and regularly update antivirus software or programs.” Finding viruses with an ever-increasing blacklist is a resource-intensive process. Software to catch viruses tends to disappear into the underutilized capacity common on dedicated hardware. A virtual environment, by comparison, makes far more efficient use of shared hardware; however, VMs can end up performing scans in competition with each other out of a limited pool of resources.

Hypervisor companies and their antivirus vendor partners are working to address this problem. For example, VMware’s vShield Endpoint offloads work from VMs to a shared and dedicated security VM on the same host. Centralized control and elimination of redundant load means a dedicated agent per VM is no longer necessary for virtual environments to achieve compliance requirements. The increased efficiency, while performing the same or better level of protection and compliance, might seem familiar to those wanting to move to cloud.

Consider how taking this newly centralized model of compliance in the cloud can affect the storage footprint for each VM versus a traditional anti-malware agent. The traditional agent, plus several signature files for rollback capability, often is several GB in size. For the sake of argument, run a quick calculation for 1,000 VMs on 10 hosts with an anti-malware footprint of roughly 5 GB per host and SAN storage for the VM at $5K per TB:

(1,000 VM) x (5 GB per VM) = 5 TB  
5 TB x ($5K per TB on SAN) = $25,000 in host-based antivirus storage space

Next, for comparison, run a calculation for a host running anti-malware on behalf of the VMs. The host-based anti-malware is likely to be larger than a VM anti-malware agent, so 7 GB instead of 5 GB gives the following result:

(10 Hosts) x (7 GB per host) = 70 GB  
.07 TB x ($5K per TB on SAN) = $350

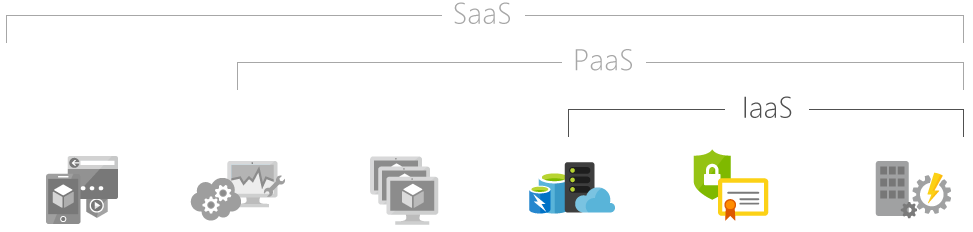
The cost savings for cloud compliance using a host-based anti-malware model shows storage is reduced more than $24K (or $24 per VM) and saves 4 TB. Network resource benefits also are possible. The hypervisor-based solution downloads malware signatures once for all the guests on a host; 10 systems have to communicate updates and events instead of 1,000. Factoring in keep-alive packets, scan start/stop status and signatures for 1,000 systems is roughly 2 MB of overhead that could be eliminated from the network. A carefully planned and controlled cloud provider environment may therefore find significant financial benefits when properly addressing [the challenges of cloud compliance](https://searchcloudcomputing.techtarget.com/feature/Cloud-compliance-strategies-for-a-multi-cloud-world).

Today, organizations are eager to take advantage of the cost efficiencies of cloud computing, but they need to ensure the move won’t jeopardize their compliance efforts. Emerging standards and improved solutions from vendors are helping to guide customers and their providers to comply with many governmental and industry regulations. In some cases, it is proving to easier to be compliant in the cloud than ever before.

**6.Summarize the core concepts of IaaS Cloud Solutions**

Infrastructure as a service (IaaS) is an instant computing infrastructure, provisioned and managed over the internet. It’s one of the four types of cloud services, along with software as a service ([SaaS](https://azure.microsoft.com/en-in/overview/what-is-saas/)), platform as a service ([PaaS](https://azure.microsoft.com/en-in/overview/what-is-paas/)), and [serverless](https://azure.microsoft.com/en-in/overview/serverless-computing/).

IaaS quickly scales up and down with demand, letting you pay only for what you use. It helps you avoid the expense and complexity of buying and managing your own physical servers and other datacenter infrastructure. Each resource is offered as a separate service component, and you only need to rent a particular one for as long as you need it. A [cloud computing service provider](https://azure.microsoft.com/en-in/overview/choosing-a-cloud-service-provider/), such as [Azure](https://azure.microsoft.com/en-in/overview/what-is-azure/iaas/), manages the infrastructure, while you purchase, install, configure, and manage your own software—operating systems, middleware, and applications.



| Hosted applications/apps | Developmenttools, database management, business analytics | Operating systems | Servers and storage | Networking firewalls/security | Data center physical plant/building |
| --- | --- | --- | --- | --- | --- |

**Common IaaS business scenarios**

Typical things businesses do with IaaS include:

**Test and development.** Teams can quickly set up and dismantle test and development environments, bringing new applications to market faster. IaaS makes it quick and economical to scale up dev-test environments up and down.

**Website hosting.** Running websites using IaaS can be less expensive than traditional web hosting.

**Storage, backup and recovery.** Organisations avoid the capital outlay for storage and complexity of storage management, which typically requires a skilled staff to manage data and meet legal and compliance requirements. IaaS is useful for handling unpredictable demand and steadily growing storage needs. It can also simplify planning and management of backup and recovery systems.

**Web apps.** IaaS provides all the infrastructure to support web apps, including storage, web and application servers and networking resources. Organisations can quickly deploy web apps on IaaS and easily scale infrastructure up and down when demand for the apps is unpredictable.

**High-performance computing.** High-performance computing (HPC) on supercomputers, computer grids or computer clusters helps solve complex problems involving millions of variables or calculations. Examples include earthquake and protein folding simulations, climate and weather predictions, financial modeling and evaluating product designs.

**Big data analysis**. Big data is a popular term for massive data sets that contain potentially valuable patterns, trends and associations. Mining data sets to locate or tease out these hidden patterns requires a huge amount of processing power, which IaaS economically provides.

**Advantages of IaaS**

**Eliminates capital expense and reduces ongoing cost.** IaaS sidesteps the upfront expense of setting up and managing an on-site datacenter, making it an economical option for start-ups and businesses testing new ideas.

**Improves business continuity and disaster recovery.** Achieving high availability, business continuity and disaster recovery is expensive, since it requires a significant amount of technology and staff. But with the right service level agreement (SLA) in place, IaaS can reduce this cost and access applications and data as usual during a disaster or outage.

**Innovate rapidly.** As soon as you have decided to launch a new product or initiative, the necessary computing infrastructure can be ready in minutes or hours, rather than the days or weeks—and sometimes months—it could take to set up internally.

**Respond quicker to shifting business conditions.** IaaS enables you to quickly scale up resources to accommodate spikes in demand for your application— during the holidays, for example—then scale resources back down again when activity decreases to save money.

**Focus on your core business.** IaaS frees up your team to focus on your organisation’s core business rather than on IT infrastructure.

**Increase stability, reliability and supportability.** With IaaS there is no need to maintain and upgrade software and hardware or troubleshoot equipment problems. With the appropriate agreement in place, the service provider assures that your infrastructure is reliable and meets SLAs.

**Better security.** With the appropriate service agreement, a cloud service provider can provide security for your applications and data that may be better than what you can attain in-house.

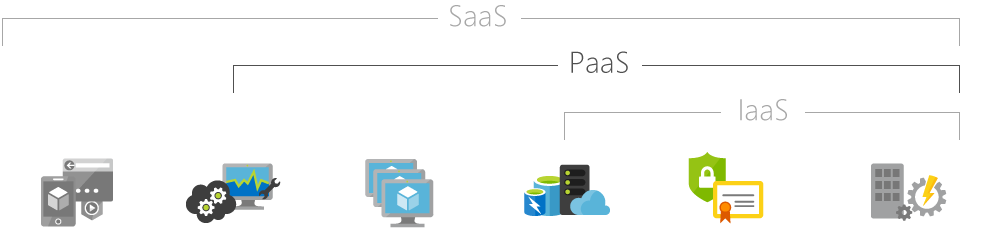
**Gets new apps to users faster.** Because you don’t need to first set up the infrastructure before you can develop and deliver apps, you can get them to users faster with IaaS.

**7.Draw and explain the essential features of PaaS Cloud Solutions**

Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications. You purchase the resources you need from a cloud service provider on a pay-as-you-go basis and access them over a secure Internet connection.

Like IaaS, PaaS includes infrastructure—servers, storage and networking—but also middleware, development tools, business intelligence (BI) services, database management systems and more. PaaS is designed to support the complete web application lifecycle: building, testing, deploying, managing and updating.

PaaS allows you to avoid the expense and complexity of buying and managing software licenses, the underlying application infrastructure and middleware, container orchestrators such as Kubernetes or the development tools and other resources. You manage the applications and services you develop and the cloud service provider typically manages everything else.



| Hosted applications/apps | Development tools, database management, business analytics | Operating systems | Servers and storage | Networking firewalls/security | Data center physical plant/building |
| --- | --- | --- | --- | --- | --- |

Common PaaS scenarios

**Organisations typically use PaaS for these scenarios:**

**Development framework.** PaaS provides a framework that developers can build upon to develop or customise cloud-based applications. Similar to the way you create an Excel macro, PaaS lets developers create applications using built-in software components. Cloud features such as scalability, high-availability and multi-tenant capability are included, reducing the amount of coding that developers must do.

**Analytics or business intelligence.** Tools provided as a service with PaaS allow organisations to analyse and mine their data, finding insights and patterns and predicting outcomes to improve forecasting, product design decisions, investment returns and other business decisions.

**Additional services.** PaaS providers may offer other services that enhance applications, such as workflow, directory, security and scheduling.

**Advantages of PaaS**

By delivering infrastructure as a service, PaaS offers the same advantages as IaaS. But its additional features—middleware, development tools and other business tools—give you more advantages:

**Cut coding time**. PaaS development tools can cut the time it takes to code new apps with pre-coded application components built into the platform, such as workflow, directory services, security features, search and so on.

**Add development capabilities without adding staff.** Platform as a Service components can give your development team new capabilities without your needing to add staff having the required skills.

**Develop for multiple platforms**—including mobile—more easily. Some service providers give you development options for multiple platforms, such as computers, mobile devices and browsers making cross-platform apps quicker and easier to develop.

**Use sophisticated tools affordably**. A pay-as-you-go model makes it possible for individuals or organisations to use sophisticated development software and business intelligence and analytics tools that they could not afford to purchase outright.

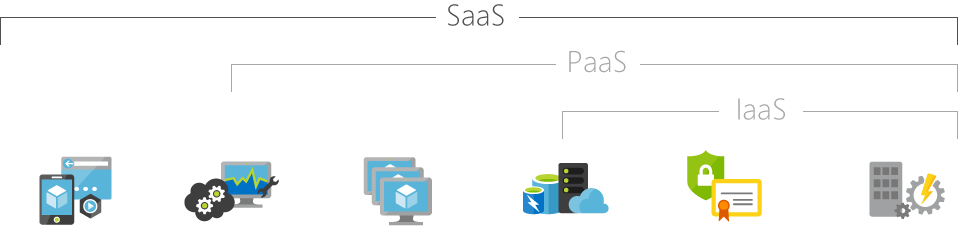
**Support geographically distributed development teams.** Because the development environment is accessed over the Internet, development teams can work together on projects even when team members are in remote locations.

Efficiently manage the application lifecycle. PaaS provides all of the capabilities that you need to support the complete web application lifecycle: building, testing, deploying, managing and updating within the same integrated environment.

**8.Describe the pros and cons of SaaS Cloud Solutions with its architecture**

Software as a service (SaaS) allows users to connect to and use cloud-based apps over the Internet. Common examples are email, calendaring and office tools (such as Microsoft Office 365).

SaaS provides a complete software solution which you purchase on a pay-as-you-go basis from a cloud service provider. You rent the use of an app for your organisation and your users connect to it over the Internet, usually with a web browser. All of the underlying infrastructure, middleware, app software and app data are located in the service provider’s data center. The service provider manages the hardware and software and with the appropriate service agreement, will ensure the availability and the security of the app and your data as well. SaaS allows your organisation to get quickly up and running with an app at minimal upfront cost.

****

| Hosted applications/apps | Development tools, database management, business analytics | Operating systems | Servers and storage | Networking firewalls/security | Data center physical plant/building |
| --- | --- | --- | --- | --- | --- |

**Common SaaS scenarios**

If you have used a web-based email service such as Outlook, Hotmail or Yahoo! Mail, then you have already used a form of SaaS. With these services, you log into your account over the Internet, often from a web browser. The email software is located on the service provider’s network and your messages are stored there as well. You can access your email and stored messages from a web browser on any computer or Internet-connected device.

The previous examples are free services for personal use. For organisational use, you can rent productivity apps, such as email, collaboration and calendaring; and sophisticated business applications such as customer relationship management (CRM), enterprise resource planning (ERP) and document management. You pay for the use of these apps by subscription or according to the level of use.

**Advantages of SaaS**

Gain access to sophisticated applications. To provide SaaS apps to users, you don’t need to purchase, install, update or maintain any hardware, middleware or software. SaaS makes even sophisticated enterprise applications, such as ERP and CRM, affordable for organisations that lack the resources to buy, deploy and manage the required infrastructure and software themselves.

**Pay only for what you use.** You also save money because the SaaS service automatically scales up and down according to the level of usage.

**Use free client software.** Users can run most SaaS apps directly from their web browser without needing to download and install any software, although some apps require plugins. This means that you don’t need to purchase and install special software for your users.

**Mobilise your workforce easily.** SaaS makes it easy to “mobilise” your workforce because users can access SaaS apps and data from any Internet-connected computer or mobile device. You don’t need to worry about developing apps to run on different types of computers and devices because the service provider has already done so. In addition, you don’t need to bring special expertise onboard to manage the security issues inherent in mobile computing. A carefully chosen service provider will ensure the security of your data, regardless of the type of device consuming it.

**Access app data from anywhere.** With data stored in the cloud, users can access their information from any Internet-connected computer or mobile device. And when app data is stored in the cloud, no data is lost if a user’s computer or device fails.

**PREPARED BY**

**SRINIVASAN.L**

**SAP/ISE/NHCE**