Cloud
Standards
Customer
Council

Migrating Applications to Public Cloud Services: Roadmap for Success Version 2.0

February, 2018

Contents

Acknowledgements

The major contributors to this white paper are: Claude Baudoin (cébé IT & Knowledge Management), Chris Carlson (Retriever Consulting), Giuseppina Cretella (Second University of Naples), Beniamino Di Martino (Second University of Naples), Mike Edwards (IBM), Shamun Mahmud (Cloud Security Alliance), John McDonald (ClearObject), John Meegan (IBM), Sujatha Perepa (IBM), Keith Prabhu (Confidis), Ram Ravishankar (IBM), Michael Salsburg (Unisys), Muralidhar Seelam (IBM) and Joe Talik (NEORIS).

Additional contributors to Version 2.0 are Christian Boudal (IBM), Steve Cirian (IBM), Ash Heda (Wipro), George Malekkos (PowerSoft), Nya Murray (Trac-Car), Bill Parker (Aspen Solutions), Srinivasa Reddy Karri (Schlumberger), Karolyn Schalk (IBM), Lisa Schenkewitz (IBM), Sri Shetty (IBM), and Prasad Siddabathuni (Edifecs).

Executive Overview

Across all industries, discussion of migration to cloud services have become commonplace. In fact, it is one of the first considerations when discussing IT cost reduction. While cost savings, speed of deployment and scalability top the list of business motivations, an increasing number of enterprises also view cloud computing as *a key enabler of business transformation* – one that can help improve customer engagement, forge new partnerships and drive competitive advantage while ensuring compliance with standards and future-proofing solutions.

However, the migration of applications to cloud computing must be done in a strategic and methodical manner. Existing enterprise applications must be thoroughly assessed to determine which workloads can benefit most from early migration to the cloud. Cloud customers must take into account the costs of migration, the potential need for application redesign, longevity, performance and availability, security and privacy requirements, the selection of locations, and other potential regulatory requirements. Moreover, the relative importance of these considerations may vary over time, and users should think of them over the long term.

The aim of this guide is to provide a practical reference to help enterprise information technology (IT) and business decision makers analyze and consider application migration to the cloud. The paper focuses primarily on the migration of applications to public cloud services, although the increasing adoption of hybrid cloud architectures (discussed in detail in another CSCC white paper) makes this guide relevant to such situations. The guide includes a sequence of steps, along with guidance and strategies, that take into consideration both business and technical requirements.

An enterprise strategy for cloud computing must identify individual business problems with existing applications that cloud computing can potentially address and provide specific business justification that the cloud is the right strategic alternative. The section titled “Motivation and Considerations” provides an overview of the potential impact that the migration of enterprise applications to cloud computing will have on new and existing business processes, and guidance on the types of applications that are best suited for migration. A business case for migrating applications to the cloud must consider the current state of both the applications and the infrastructure, and evaluate the ability of cloud computing to support the enterprise strategy and deliver meaningful business value.

In most cases, applications are moved in “move groups,” which are based on qualifying criteria that assess them as having minimal, low, medium, or high impact to the enterprise. Starting small and expanding after initial success is usually the most prudent approach to application cloud migration.

The section titled “Migration Roadmap” is the heart of the guide and includes the basic steps of a formalized migration process. It details both strategic and tactical activities for decision makers to develop a business plan and detailed migration plan.

Motivation and Considerations

The business community might have the following motivations for the migration of an application to the cloud:

- Greater scalability
- Making access easier by a mobile workforce
- Business agility, collaboration and flexibility
- Improved security
- Better analytics on application usage
- Reduced software maintenance effort
- Improved availability and responsiveness
- Improved compliance with laws and regulations (e.g., on privacy, data residency, or trade compliance)
- Reduced and/or re-allocated infrastructure costs

Most of these objectives are well matched by key cloud computing characteristics:

- *Rapid elasticity*. The ability to rapidly scale the IT infrastructure (up or down) to match changing requirements, on a pay-per-use basis, is extremely attractive to large and small organizations alike. Applications designed to benefit from automated scaling of resources to match demand are increasingly common. This behavior, combined with the pay-by-usage characteristic of a cloud, can lead to significant financial savings, especially for businesses with varying or cyclical IT usage needs.
- *Pay-as-you-go vs. install-and-own*. The shift in up-front capital requirements from the customer to the service provider is equally attractive. In particular, small organizations and start-ups face much lower infrastructure costs than were necessary pre-cloud.
- *Automatic, timely vendor updates to software.* Cloud deployments typically entail suppliers rolling out software and security updates. That is one less responsibility for the IT department to handle.
- *Technology streamlining and standardization*. Migrating to the cloud often provides an organization with a more updated/standardized suite of technologies and reduces the proliferation of tools and solutions requiring support. This also helps obtain and maintain compliance (regulatory or other) and stay current with industry/technology trends.
- *Organizational streamlining*. Buying capabilities such as “security as a service,” “collaboration as a service,” “communication as a service,” etc., decreases the need for specialized in-house IT skills and can free up internal resources for other priorities.
- *Geographically distributed computing capacity*. Many cloud providers offer the ability to add infrastructure capacity rapidly in various worldwide locations, enabling a business to quickly expand to new territories.

There are of course concerns to be addressed. While most of them are increasingly tractable, they need to be considered early on in the migration decision process:

- *Security, Privacy and Data Residency*. Moving data and code to a third-party provider creates risks related to control over sensitive data – as well as concerns about the handling of intellectual property embedded in software. Although the technology to make cloud computing safe is available, securing cloud workloads often requires new concepts and skills that may take time to acquire.

- *Loss of control*. For software-as-a-service (SaaS) and some platform-as-a-service (PaaS) solutions, the entire control of hardware, software, security policies, etc., is placed in the hands of a third-party provider.
- *Integration*. Most customers need to integrate internal systems with cloud systems. Having these two types of systems communicate with each other is always a challenge.
- *Availability and reliability of cloud applications*. Issues may arise from a combination of server performance, configuration errors, network design, and application architecture (possibly in combination, which can initially make them difficult to resolve).
- *Cloud service provider lock-in*. The concern is that once the cloud service of a specific provider is adopted, it will not be easy to switch to an equivalent cloud service of a different provider. Emerging standards will increase the portability and interoperability of systems across cloud service providers. This will in turn reduce or eliminate this current barrier to cloud adoption, although this will not be complete until providers offer standard service patterns.

To prioritize applications for migration to cloud computing, it is necessary to first identify and understand the business and technical factors for the migration.

Migration Roadmap

As customers transition their applications and data to the cloud, they need the level of service provided in the cloud environment to be at least comparable to the service provided by their traditional IT environment. In fact, a move to the cloud may alter internal or external (customer-facing) key process indicators (KPIs) that support or relate to changes in the business. Failure to properly migrate applications to cloud computing could ultimately result in higher costs and potential loss of business, thus canceling the potential benefits of cloud computing.

This section provides a prescriptive series of steps that end users should take to ensure successful migration of existing applications to cloud computing:

1. **Assess Your Applications and Workloads**
2. **Build the Business Case**
3. **Develop the Technical Approach**
4. **Adopt a Flexible Integration Model**
5. **Address Security, Privacy, and Data Residency Requirements**
6. **Manage the Migration**

Requirements and best practices are highlighted for each step in the sections that follow.

Step 1: Assess your Applications and Workloads

Assessing applications and workloads for cloud readiness allows organizations to determine what applications and data can – and cannot – be readily moved to a cloud environment and what delivery models (public, private, or hybrid[1]) can be supported. Alternatively, you might start by determining which applications you *do not* want to move to the cloud initially – for example, because there are serious doubts that compliance with certain laws and regulations could be fully ensured.

What is a "workload," and how does it differ from an application? Typically, a workload is defined as a certain amount of processing one wishes to perform, and consists of one or more application and system images, with a certain number of users connected to and interacting with the applications. The reason this concept is important in a cloud context, and specifically when considering cloud migration, is that migrating only one part of the workload to the cloud will usually result in performance issues.

This assessment will not only result in "go/no-go" decisions, but also in prioritization of the applications – what to move first, or what to include in a pilot project or Proof-of-Concept project, and what to defer.

The decision criteria will often be refined as the assessment progresses. A large enterprise with a sizable portfolio of applications may want to split the assessment into the following steps:

- Assign all applications to a "business component" – something that may leverage an enterprise capability map if one exists.
- Determine which business components are more cloud-ready than others. This generates a high-level view of what is possible, where to make the effort, and how to set priorities for the migration roadmap.
- Then, dive deeper on the assessment of the various workloads within the prioritized business component(s).

When we talk about applications being more or less cloud-ready, the following criteria may be considered:

- Complexity of the application (and the level of understanding of its architecture)
- Impact to the business (positive impact of a successful migration, as well as potential negative impact of migration problems)
- Transactional dependencies
- Benefits of ending support for legacy applications
- Presence or absence of sensitive data content, especially customer information or personally identifiable information (PII)
- Likelihood of taking advantage of the cloud's elasticity

Table 1 highlights examples of suitable and less suitable types of applications for migration to cloud computing. Note that the suitability of an application or workload can be a complex decision, rarely a black-or-white one. First, there are various degrees of suitability on a continuous scale, based on many criteria. Second, it is not just a question of "should we migrate it or not?" but also a question of "how

[1] The hybrid model of cloud computing is discussed extensively in a separate CSCC White Paper [1].

can or should we migrate it?” Some applications are cloud-ready, others are appropriate for a “lift and shift” strategy, and yet others should be redesigned to fit in a cloud-native architecture such as microservices. For a list and definition of these strategies, see the “Six Migration R’s” at the top of Appendix B.

Table 1 - Application Candidates for Migration to Cloud Computing

Most Suitable Candidates for Cloud Migration	Least Suitable Candidates for Cloud Migration
• Applications that are run infrequently but require significant computing resources when they run. • Business-to-Consumer (B2C) applications used by a broad base of consumers, where you do not know how many users will connect and when (i.e., they require rapid scaling of resources). This also applies to Peer-to-Peer (P2P) and Peer-to-Business (P2B). • Applications used by mobile workers to manage their time and activity, and which contribute only limited information to the company's broad management information databases. • Applications that are run in a time zone different from that where your company's IT personnel are located. • Development, testing and prototyping of application changes, even if the final applications will be run on your own infrastructure. • Service Oriented Architecture (SOA) applications. • Loosely coupled applications. • Applications built to provide and use APIs. • Applications that require rapid provisioning of infrastructure. • Applications that would require a capital expenditure and for which a recurring operating expense is preferred. • Evaluation projects for which it is not certain that the application will be retained and licensed.	• Applications that involve extremely sensitive data, particularly where there is a regulatory or legal risk involved in any disclosure. These at minimum require special treatment to be run in a cloud service. • Applications now being run on the company's private network and that are very performance- or latency-sensitive. • Applications that require frequent and/or voluminous transactions against an on-premises database that cannot be migrated to the cloud. • Applications that run on legacy platforms that are typically not supported (or may not be supported in the long run) by cloud providers. • Applications not yet virtualized. • Highly customized applications (for migration to SaaS). • Applications that require specific types of servers or are otherwise inflexible about the type of computing resources needed.

Having identified which applications or workloads are candidates to be migrated to the cloud, the next question is how *ready* these applications are to be migrated – that is, whether the target operating model (e.g., single public cloud, hybrid cloud, multi-cloud, microservices, etc.) and state are feasible in the short term. This readiness assessment spans the following areas:

- *Business Considerations*. Business considerations include the overall organizational readiness for using cloud computing. Is the application owner willing and comfortable with a cloud platform? How important is the application to the business or the mission? What is the risk tolerance level of the business, and is the culture favorable or resistant to change? Is the business evolving in such a way that it requires higher KPIs, which the cloud may be better able to fulfill?
- *Application Lifecycle Considerations*. Is the application still being defined? Is it up for a refresh? Is the application approaching retirement? Can the application be redesigned or undergo a technology refresh for cloud computing? Will there be an efficiency gain in using cloud computing? Instead of migrating the existing application to cloud computing, using an IaaS or PaaS approach, would it be better to replace it with a new SaaS solution?

  Applications must be able to be redesigned in view of serverless "functions" (see Step 3 for a short discussion of the Function-as-a-Service model) or risk having to be re-migrated within a couple of years. Virtual machines and containers are currently the main form of migration, and they create integration challenges as network virtualization and discrete serverless functions evolve rapidly in the public cloud.
- *Application Architecture Considerations*. Is the application web-based, or built with a service-oriented architecture (SOA)? Does the application provide and consume APIs? If not, can the application be split into modular services (microservices)? Is it monolithic, two-tier, three-tier, or n-tier? What is the level of effort required to modularize it or separate the tiers? Does the application scale out? Does it scale up? What are the demand fluctuations in the application? What impact will the move to cloud computing have on demand?
- *Data Considerations*. Data governance, confidentiality, integrity and quality need to be preserved by the migration. Is the data bound by statutory compliance? Are there data sensitivity and privacy or confidentiality concerns? What data integrity concerns are there? How does the application manage data requests from a safety and security perspective? How much data exchange will occur between the components of the application and between the application and the user? Are there legal or regulatory requirements about the physical location of the data? Frequent data transfers may impose a higher cost as well as a performance lag.
- *Infrastructure Considerations*. These include the performance and resiliency of the network. The migration design must account for multiple components communicating across network boundaries. Techniques such as network isolation, virtual private networks, elastic addressing and network segmentation can provide for a very robust and secure cloud environment. The application must be designed (or must be modifiable) for resiliency – immunity to the interruption of transactions in midstream, as well as local fault tolerance. Is the application designed for high availability and disaster recovery? Standard and open protocols are more readily supported across firewalls and on a public infrastructure than proprietary ones. Is the

application supported by a standard infrastructure environment and a supported operating system version? Being several releases behind the current ones will increase migration cost and effort.

- *Security Considerations*. The different parties – application owner, cloud service provider(s) and the customer's IT department – must understand that security is their joint responsibility. Authentication and authorization remain the responsibility of the customer at the application level. The cloud service provider is responsible for security controls, identification and correction of system vulnerabilities, and defense against specific cloud-oriented attacks (e.g., at the virtual machine level), consistent with the level of service selected. Continuous monitoring is now common among cloud service providers and should be expected. Customers also have to develop the capability to analyze and respond to the provider's security and access monitoring services, and often overlook this significant change in the role they play in incident response.
- *Integration Considerations.* What are the dependencies between the application being migrated and other systems? Applications may depend on each other through control integration (they invoke each other), data integration (they read or write the same databases or files), or presentation integration (they are mashed up on the same window or Web page). The migrated application may even be the "system of record" for some key data in a Master Data Management (MDM) scheme. Finally, the migrated application may rely on common facilities such as a user directory for single sign-on and access control. The assessment must discover how extensive these integrations are, what protocols they use, what additional utilities or runtime libraries they rely on, and what their performance requirements are, including the frequency of connections and the amount of data involved.
- *Operations considerations*. How will the solution be run and managed? Will the cloud service provider manage only the infrastructure, middleware, and basic security, while the customer manages the application, data, integration, and compliance – or will the cloud provider manage the entire solution? How will service level responsibilities be split between the two parties? A robust RACI matrix is essential to defining which party is responsible for what.

Appendix A provides examples of some of the most cloud-ready workloads and their benefits.

Step 2: Build a Business Case

Developing a business case for migrating applications to cloud computing requires an overall cloud computing strategy, including specific information that describes the current state and demonstrates the advantages of cloud computing to not only reduce costs but to deliver meaningful business value. High level value propositions for cloud computing, including the shift of capital expenditures (CAPEX) to operational expenses (OPEX), cost savings, faster speed of deployment, elasticity, etc., are necessary but insufficient unless quantified. Within the context of an enterprise strategy for cloud computing, individual business problems with existing applications that cloud computing can potentially address need to be identified, and specific business justification must prove that cloud computing is the right strategic alternative. Refer to the CSCC *Practical Guide to Cloud Computing* for specific considerations that need to be taken into account when developing an enterprise strategy for cloud computing [2].

Cost and Savings Analysis

Once an application is identified as a potential candidate for migration, a thorough analysis of the migration costs, as well as any expected savings and revenue enhancement, must be performed. In order for meaningful comparisons to be made, one must have specific baseline costs for the current environment.

The overall cost of application migration to cloud computing must include the following elements:

- *On-going cloud service costs*. The cloud service provider fees must be taken into account, including the effects of variable demand, such as extra fees to handle peak loads. Careful attention should be given to network cost as it may vary between cloud service providers.
- *Service management*. Few customers are experts in managing services and service providers, yet this is of critical importance to the successful use of the cloud. Even if the client uses external managed services, the internal cost of managing the client‘s activities has to be included.
- *Security management*. Cloud providers may provide basic security features such as firewalls and anti-virus, but the solution may require additional security features for compliance in highly regulated domains. Examples of such regulations in the U.S. are the Health Insurance Portability and Accountability Act of 1996 (HIPAA) and the Health Information Technology for Economic and Clinical Health (HITECH) Act; the Payment Card Industry Data Security Standard (PCI-DSS); and the Federal Risk and Authorization Management Program (FedRAMP). The cost for additional services, software, and potentially hardware must be taken into account based on the needs of the business solution.
- *License management*. It is important to understand third-party software dependencies and impact to licensing contracts (and ongoing management of these licenses) when migrating an application to the cloud. For example, how does autoscaling of infrastructure impact license requirements, and how is the consumption of licenses measured?
- *Application re-designs*. The application may require design changes in order to be compatible with or to take full advantage of cloud deployment.
- *Data and application integration*. Many cloud-based solutions are no longer stand-alone but are hybrid solutions with, for example, a System of Record (SoR – the enterprise system) remaining on client premises while the System of Engagement (SoE – front-end) runs in the cloud. Integration between the two is fairly common but may need new APIs or an Internet-facing API gateway, with attendant security enhancements. This can be a significant cost, depending on what the client already has in place on premises.
- *Application deployment and testing*. The application must be configured, deployed, and tested in the cloud environment. This is especially obvious if a redesign was required to adapt it to the new environment, but some testing is necessary in all cases, and adds a cost.
- *Application maintenance and administration*. In an IaaS or PaaS scenario, ongoing maintenance and administration of the cloud-based application will remain the client’s responsibility.

- *Human resources, training and talent management*. The existing organization may lack skills and abilities such as preparation and deployment of virtual machine images. Internal personnel may need to be retrained to support the migration to cloud computing. Some certifications may be advisable to better support applications moved to the public cloud and to work in an Agile or DevOps framework. A cloud strategy may require changes to supervisory, control and compensation systems. Job descriptions, bonus plans, etc., may change.

On the other hand, there are expected savings from such a migration, otherwise it would not make financial sense.

- *Move from CAPEX to OPEX.* A main source of savings is the shift from capital expenditures, which can be depreciated over time but require cash up front, to operating expenses.
- *Savings on the handling of peak loads*. The ability to scale server capacity up and down to match spikes in demand should result in lower costs for a cloud-based solution than for an on-premises one, since the customer no longer needs to pay all the time for the infrastructure required to meet peak capacity needs.
- *Short cloud contract periods*. On-premises solutions usually require a nonrefundable investment in hardware and in software licenses. Cloud contracts allow cancellation or reduction of services with certain advance notification requirements.
- *Staff reduction of reassignment.* Moving applications to the cloud will reduce the needs for system administration and/or application maintenance. This should result in reduced headcount – perhaps making resources available for reassignment to other tasks, or to cope with the growth of other operations that remain in-house.

Service Levels

The level of service provided in the cloud should be comparable to (or better than) prior service levels. In fact, the service levels provided by an internal IT department to its business customers are often not well specified, or not specified at all, and migrating an application to the cloud will shine a useful spotlight on those essential commitments. The required service levels should be agreed with the cloud service provider and explicitly documented in the Cloud Service Agreement. Refer to the CSCC *Practical Guide to Cloud Service Agreements* and the CSCC *Public Cloud Service Agreements: What to Expect and What to Negotiate* for specific aspects to consider [3], [4].

For each application being migrated to cloud computing, consider the following application characteristics:

- *Application availability*. The criticality of the application to business operations will determine the availability requirements that must be clearly specified in the cloud SLA.
- *Application performance*. Depending on the performance requirements of the application, specific performance targets may need to be achievable with the cloud service.
- *Application security*. Moving an application to the cloud will require due diligence on the part of the cloud service customer to ensure proper security controls are in place and operating effectively, especially given the severe consequences of security breaches under regulations

such as the European Union's General Data Protection Regulation (GDPR), which comes into effect on 25 May 2018.

- *Privacy*. Personally Identifiable Information (PII) handled by a cloud-based application must be properly stored and maintained. Access to PII stored in a cloud service must be restricted as required, including from cloud service provider personnel.
- *Regulatory compliance*. Government and industry regulations may require additional measures, such as restricting the migrated applications and data to reside in a specific geographic region.

Business Impact

Assuming the cost and service analyses described above are favorable, then additional business factors must be weighed in order to develop a complete business analysis, and should be *monitored* on an ongoing basis:

- *Revenue impact*. If the application is used to generate revenue, is the move to cloud computing expected to increase that revenue? A move to the cloud may allow the business to scale up or offer new services. Improved responsiveness to customer needs (from response time improvements to the more rapid deployment of application upgrades) may also result in additional revenue through greater customer loyalty and retention rates.
- *Customer acquisition or engagement impact*. For a customer-facing application, is the move to cloud computing expected to increase the number of customers accessing it?
- *Customer satisfaction*. Does one expect an improvement in availability or response times that will result in increased user satisfaction?
- *Time to market improvements.* Will the move to cloud computing shorten the time it takes to deliver functional enhancements to end users?

Obtaining executive support for the initiative is critical. Executives from IT, Lines of Business (LOBs), procurement and executive management must review and approve the business plan before proceeding. Getting key executives on-board early in the process will help alleviate potential issues down the line.

Understanding Organizational and Role Changes

Some larger organizational changes need to be planned for. IT management, its executive sponsors, and the IT department staff must all understand that in order to support the use of the cloud, the IT department organization must be different from what was in place to support a traditional IT infrastructure. New roles are needed, while some old jobs will disappear.

Step 3: Develop a Technical Approach

Broadly speaking, the potential target service models for the migration of an existing application may be a combination of Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Container as a Service (CaaS), and Function as a Service (FaaS). If the existing application is a packaged one, there is potential to move it to Software as a Service (SaaS), either with the vendor of the software package or with a cloud provider specializing in hosting of the packaged application.

In addition to choosing the service model, embracing DevOps capabilities and automation are increasingly seen as critical to realize the technical and business benefits of cloud adoption.

A key impact of the choice of an “XaaS” service model is the positioning of the “service responsibility line” (SRL), which delineates the respective responsibilities of the cloud provider and the cloud customer. Figure 4 of the CSCC’s *Practical Guide to Hybrid Cloud Computing* [1] describes the SRL for the three classical service models (IaaS, PaaS and SaaS) and will need to be extended to cover CaaS and FaaS.

Cloud customers should note that the rapid evolution of computing architectures (network virtualization, detachable network interfaces, orchestration, etc.) requires a significant effort in technology watch to take advantage of these game-changing capabilities.

PaaS Migration

To migrate to PaaS, the application itself must be designed or redesigned for one or more runtime environments available on the target platform. An example of such an application is one where the business logic is implemented as a set of components that run on an application server (such as IBM's WebSphere, Oracle's WebLogic, or the JBoss server) in combination with a database (such as the Oracle database or IBM's DB2) containing the application's data and also possibly associated code in the form of stored procedures.

In general, a PaaS solution must provide the elements of the particular software stack required by applications such as the operating system, an application server and a database, so that the customer only has to be concerned with the specific application components and data. One must also ensure that the PaaS environment offers the configuration(s) required by the application. This may include software levels, the ability to run scripts, and the presence of certain tools for setup, reporting, logging, monitoring, auto-scaling, etc., identical or similar to those present before migration. Software versions sometimes differ between what is available on a PaaS compared to what is available on dedicated hardware in the customer’s data center.

If there is no PaaS solution that contains the exact same software stack used by a legacy application, that application may need to be rearchitected to run on the target environment. This can be an opportunity to move from a monolithic architecture to a microservices architectural style.

IaaS Migration

To migrate an application to IaaS (Infrastructure as a Service), the requirements on the cloud service itself tend to be lower. The entire software stack is migrated: the application code itself, plus any supporting code it requires – including the underlying operating system. Two approaches tend to dominate traditional IaaS:

- *Virtual machines (VM) servers*. In this approach, the customer receives virtual machine(s) running on shared hardware servers. To use this, one must be able to package the complete software stack as one or more VM images, which can then be copied into the cloud service and executed there.

- *Bare Metal (BM) servers*. In this approach, the customer receives a dedicated server, in much the same way they would on premises or in a traditional hosting environment. This offers much greater flexibility for applications requiring high performance, or those with compliance requirements specifying dedicated hardware, or applications where software licensing terms force hardware decisions.

There are variants of those approaches depending whether the target infrastructure is single-tenant, multi-tenant, or require the replication of the same images over many instances ("hyperscale"), the last one being particularly adapted to content distribution networks (CDN).

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 provider; an application depending on these capabilities is not a good candidate for migration. This can be tested by preparing the virtual machine image of the application and of the supporting code and attempting to execute it on a trial VM environment (either in-house or a test system offered by the cloud service provider). Hidden dependencies can thus be found, corrected, and the process repeated until successful or until it is determined that there is no affordable solution.

In a bare metal cloud environment, some components – such as the underlying network – will still be shared even if the server is dedicated to one customer. Even so, some virtual network designs will help secure customers in a multitenant environment. For example, VMs and instances can be organized with subnet isolation. Implementation details are beyond the scope of this white paper.

A "Bring your own device" (BYOD) capability is sometimes available, and may be needed if a specific encryption/decryption or tokenization/detokenization appliance is required. This varies greatly between cloud providers; customers having this need should check whether the provider allows it, or research whether a suitable virtual device (running as software on a VM) exists.

CaaS Migration

CaaS (Containers as a Service) is the next generation of PaaS, potentially allowing multiple PaaS environments and also FaaS (Function as a Service) on a CaaS infrastructure. CaaS features include all PaaS features, plus an orchestrator engine allowing reliable software rollout, service healing, auto-scaling, and load balancing. A major benefit of containerization is the portability of the containers, which once built can be run anywhere.

To use CaaS as the target for migration without architecture changes, the software stack required by an application must be made available as a container, and the application code itself must also be packaged as a container. When a legacy application and its supporting software stack are not available as containers, one has to rearchitect the applications into microservices, package them as a container, and run the container on an open-source or third-party runtime environment. In addition, the architecture of the container service should be considered ahead of time, as there are multiple options such as Docker or Warden, with a component such as Kubernetes for orchestration.

There is a fork in the road when it comes to orchestration options. The orchestration of containers, with their inherent security limitations, ignores some of the moves toward "network as a service" and IaaS orchestration made by providers such as Microsoft or Amazon. It is clear that orchestration of containers is always going to be constrained by VM design. Containerization is a means to quick migration of monolithic applications, and may not be a long-term solution compared to a microservice architecture.

FaaS Migration

To use a FaaS (Function as a Service) as the target for migration, application logic is invoked in response to events or direct invocation from the Web or a mobile environment over HTTP. The benefits of this new paradigm are that one does not need to explicitly provision servers, or worry about auto-scaling, high availability, updates, maintenance, or pay for hours of processor time when the server is running but is not serving requests. The customer gets billed by millisecond of execution time, or on some platforms per request, not per hour of VM (regardless whether that VM is doing useful work or not).

This programming model is a perfect match for microservices, mobile, IoT and many other applications – auto-scaling and load balancing are inherent benefits without having to manually configure clusters, load balancers, HTTP plugins, etc. The customer also gets the benefit of "zero administration" – meaning that all of the hardware, networking and software is maintained by cloud service provider.

DevOps

Implementing a DevOps approach to maintenance and support (with the associated role changes for development teams) is practically a necessity in the cloud.

One of the key differentiators between an on-premises legacy application and a cloud-deployed one is that legacy support – with its associated ticketing system and support personnel – are no longer available. The application team now needs to support their applications themselves, including receiving and handling the first call for help, PD (Problem Determination) and PSI (Problem Support Identification). A plan should be in place to understand each component that needs to be supported, who will do the support, and desk procedures for invoking support.

DevOps helps streamline these new roles, allowing for automated code builds, testing, and deployment. This reduces both the administrator time required by the development team and the risk of errors going into production. On the other hand, DevOps comes with certain risks, in particular lack of compliance to an enterprise architecture. This needs to be addressed through governance mechanisms. Just like Agile, DevOps cannot be an excuse for jettisoning the disciplines of planning, architecture, or documentation.

Common Technical Considerations

In the PaaS, IaaS, CaaS and FaaS cases, the following technical considerations must be taken into account:

- *Skills*. The organization needs to possess (or acquire) the skills needed to prepare and migrate the application components. The preparation of virtual machine images and their deployment to a cloud service may involve skills new to the organization.
- *Security.* The cloud service's security features may be very different from those of the in-house environment, and the security risks and the measures applied to counter them must be assessed

carefully. For example, if the application is only used by the customer's employees, it may be wise to place the application within a Virtual Private Network (VPN) in the cloud service, providing secured access for staff while preventing access from outsiders over the Internet[2]. Data that is migrated to a cloud service needs similar technical decisions – can the data be stored in cleartext, or does it need to be encrypted, even at rest, to reduce the risk of exposure or theft?

Other technical choices depend on the security measures applied by the cloud provider to the cloud service. Does the provider implement strong user authentication techniques? Does it offer other security tools, for example to implement encryption of data in transit or at rest?

- *Integration.* Integration with other applications and services within the customer organization, which may be bidirectional, may involve configuration changes (e.g., to reflect new addresses), new authentication methods, and other technical changes (e.g., data replication and synchronization) to avoid network latency and throughput. This is addressed in greater depth in Step 4. The current absence of integration or interoperability needs between two applications does not mean that such a need will not arise later as the result of a change in business requirements.

  In view of security and privacy requirements such as GDPR, all integration should rely on a complete data model and classification. Without this, the integration architecture will have to be completely reworked to meet security constraints.

- *Monitoring and Management*. Can in-house tools still be used, or is it necessary to adapt to new monitoring and management facilities supplied by the cloud provider? Monitoring resource usage by the application is important, since undetected high usage is likely to inflate the cloud fees. Do both parties, cloud provider and customer, need access to the management data? The detailed data gathered by the provider's tools to monitor the service levels vs. the SLA (e.g., uptime) may be of benefit to the customer.

  Some cloud providers offer to share these measurements, while others present the cloud service as more of a black box, where the only data provided to the customer is the rolled-up SLA measurement. There are advantages to using dashboards provided out of the box by cloud providers as part of the cloud service subscriptions. Major cloud suppliers also offer endpoint detection and protection toolkits that can be applied to the monitoring data they offer.

- *Scalability*. While scalability is a common advantage of cloud services, applications have to be structured appropriately to take advantage of scalable cloud resources, and this may require changes to the application code. In particular, the challenge of reprogramming an application to use multiple processors or multiple machines in parallel can be significant.

- *Availability and Backup*. In-house designs to support the availability of the application may need significant modification to deal with the cloud service environment, especially for PaaS services.

[2] There is a distinction between SSL (Secure Socket Layer) VPN and IPsec (IP Secure) VPN, which is considered more secure. Other technologies may apply, such as private virtual gateways.

Backup processes for the application may need to be adapted to the environment of the cloud service.

These technical considerations must feed back into the business case for migration and possibly may call into question its very feasibility. The technical issues also feed into the migration plan (refer to Step 6 for details).

Patterns

One approach that may help with the migration of applications to cloud computing is the use of *patterns*. Patterns describe common aspects of cloud computing environments and of application design for cloud computing. Some patterns can be useful in understanding the appropriate organization of the software stacks on which applications depend. Patterns can also be useful in understanding what changes may be necessary to the application code for successful migration.

Some of the general patterns which apply to cloud computing are described in the paper "A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-based Application Architectures" [5]. Patterns can also be specific to a cloud computing platform like those provided by Amazon, Microsoft, IBM, and others [6], [7], [8].

Step 4: Adopt a Flexible Integration Model

It is common for applications to have several points of integration with both internal and external systems, providers, etc. Such integrations respond to various needs, including but not limited to:

- Performing 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).
- Sharing of common master data (e.g., a product catalog or a customer database) between applications.
- Single sign-on across multiple applications and systems.
- Monitoring an application in the cloud using a suite of on-premises IT tools.
- API management – to support flexibility, interoperability and portability.

Migration to the cloud increases the likelihood of such connections. These connections can be classified into:

- Cloud to Cloud: newly migrated applications integrating with other applications, services or providers that are also in the cloud.
- Cloud to on-premises: The application migrated to the cloud integrating with an application on premises or vice versa.

Integration between applications is also typically classified into three types:

- Process (or control) integration, where an application invokes another 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, a mashup, frames, etc.

In many cases, the challenge with migration is not so much “integration” as it is “reintegration” or “maintaining the integration” between pieces of the entire system that are coupled in a certain way.

The first task in addressing integration is to inventory the connections or “integration points” in question. If the application makes use of SOA or APIs, that architecture is a good place to start. Once that is done, identify an integration approach (noting that there is usually not a one-size-fits-all solution). Pragmatic integration recommendations include:

- Be flexible – potentially use several different techniques according to specific situations.
- Use standards – in order to be more maintainable and less fragile with respect to changes the cloud provider might make later.
- Consider the possibility that more migrations may occur in the future – therefore cloud migration is an opportunity to modernize the architecture and render it more resilient to such changes.
- Rate the complexity of each integration point (e.g., establish some criteria to determine what will require a small, medium or high level of effort) in order to estimate the duration and resources required by the migration.

Once that classification is complete, the effort and resource levels are identified, and priorities are established, the integration patterns might entail one or more of the following:

- One is to “bite the bullet” and modify each integration point individually. This will probably be costly, and may not yield a reusable solution next time there is another application migration to undertake.

- One is to actually migrate *more* systems than initially planned – that is, move the entire “spaghetti” of integrated applications to the cloud. The connections between the applications, which used to be local to the customer’s data center, are now local to the cloud service, and as a result they are less likely to be impeded by firewall rules or performance issues.

- Devise data stores and processes to account for performance and security/sensitivity concerns -- an example might be to place a partial cache of the master data on the same system (or the same local network) where the migrated application will reside, and synchronize the cache with the master as the network bandwidth permits, or on a more periodic or on-demand basis.

- A more substantial architectural change may be required. This may in turn take several forms:
    1. Using microservices to replace a legacy architecture.
    2. Using an established message-based integration approach, such as an Enterprise Service Bus (ESB), that is compatible with Internet-based communications – in fact, extending the concept of service bus so that it can be used between on-premises and cloud-based systems.
    3. Using special cloud integration solutions, typically based on customizable templates, specifically designed to connect cloud solutions to internal enterprise applications. Refer to

step 7 of the CSCC whitepaper *Convergence of Social, Mobile and Cloud: 7 Steps to Ensure Success* for more details [9].

Finally, in order to support the applications in the cloud, one must document their architectural *operational* model on the cloud infrastructure. This documents such aspects as the communication between components; interactions and interfaces that cross security zones; and how component redundancy and failover are set up. This information will be needed for problem source identification in the event of outages or other failures.

Step 5: Address Compliance, Security, Privacy and Data Residency Requirements

Compliance, security, and privacy are key issues that concern cloud service customers the most, and data residency has been added to the mix in the last few years. Depending on the economic sector, these may be just above or below concerns about availability and performance as highest priority. At the same time, cloud service customers should remember that many of the compliance, security, and privacy concerns raised by cloud computing have existed since the first forms of IT outsourcing were introduced.

Compliance concerns include those imposed by external entities such as governments or other regulatory bodies, but can also be imposed by the customer's own internal IT or legal departments. Common concerns include:

- Location limitations on where data can be stored or processed, such as that it must remain within a country's borders.
- Specific types of security features, equipment, or software required – e.g., Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) – or two-factor authentication for privileged user access such as the root/superuser accounts.
- Specific controls on processes and procedures.
- Specific services mandated such as logging or data retention.

Security involves multiple concerns. Without restating the details contained in the CSCC *Security for Cloud Computing: 10 Steps to Ensure Success* [10], they include such aspects as:

- How hard is it for an intruder to steal confidential data from the cloud provider's systems (external threat)?
- If this happens, will you even know it? How do you prevent it from happening again? To whom and how do you communicate about this incident?
- Can you trust the provider's personnel, especially system administrators who have many privileges over the systems you use (internal threat)?
- What does the SLA promise you in terms of security measures?
- What is the impact on your business if a denial-of-service attack occurs, which may not endanger your data but prevents users from accessing the application?

- How do you authorize an employee to access a system or application in the cloud? What levels of trust do you grant different users, and how do you identify and authenticate trusted users?
- How do you monitor administrator or user activities and detect suspicious activity?
- How do you prove to a client or an auditor that adequate security measures are in place, now that this is not only your problem, but a shared responsibility between you and a cloud provider?
- How can you verify that the virtualization platform or cloud management software running on the systems you use, which you did not install and do not control, does not contain malware?
- How can you protect your systems from malware that could be introduced by another customer in a multi-tenant environment?
- What is the risk that your data will be delivered to a domestic or foreign law enforcement agency by the cloud service provider in response to a legally binding request?

Privacy is closely related to security, but it carries with it the additional burden that a violation of privacy, for example the disclosure of PII about your own users or customers to people who do not have a right to access it, could cause major damage to your company, including but not limited to:

- Loss of business
- Legal action by the people whose information has been disclosed
- Impact of non-compliance with government regulations, including fines and loss of permits

In addition, data subjects may have rights to inspect and correct PII that relates to them, which will need to be supported by the application even when it runs in a cloud service.

No discussion of cloud privacy would be complete without mention of the European Union's General Data Protection Regulation (GDPR), which was published in March 2016 and goes into effect on 25 May 2018. The regulation will affect any organization that processes and/or stores personal information of European residents. It is important to emphasize that *even non-EU-based organizations* are subject to GDPR enforcement [11].

Data residency has been defined by the Object Management Group and the CSCC as "*the set of issues and practices related to the location of data and metadata, the movement of (meta)data across geographies and jurisdictions, and the protection of that (meta)data against unintended access and other location-related risks."* While related to privacy, it is distinct because there is a complex and contradictory web of laws and regulations that limit where certain types of data may be located, even when such data does not contain PII. Data in transit may also fall within the scope of the laws of countries and jurisdictions it passes through. These risks, and potential actions to mitigate them, are discussed extensively in a 2017 CSCC paper, *Data Residency Challenges* [12].

Now that we have examined all the risks and threats that arise when migrating an application to the cloud, it turns out that doing so may, in fact, *improve* its security. This statement is based on two facts:

- Cloud service providers have, in all likelihood, more expert resources at their disposal than many of their customers. They need to make that investment because a successful attack could damage their entire business. Therefore, customer data may be safer in the cloud provider's custody than in the customer's in-house systems. This is the same principle that leads people to rent a safe deposit box at their bank rather than keeping their valuables at home.

- Once your data is held in a cloud service, an attacker who specifically wants to gain access to your information no longer knows exactly where to attack. Even if they successfully penetrate the network of the cloud provider, there may be thousands of virtual servers whose names do not reveal data ownership details.

Knowing all the above, here are some logical steps to follow (again, see the CSCC *Security for Cloud Computing: 10 Steps to Ensure Success* [10] white paper for more information). Note that as a result of performing these tasks, you will never be 100% protected, and after a risk analysis you may even end up deciding that you cannot in fact migrate certain data or applications. But they will certainly increase the chances of success.

1. Understand exactly what data (including what code, since code may be the confidential asset to protect) will be migrated to the cloud service.
2. Map this data to your security classification. If a security classification does not exist, or if it does not specify where and in which format (cleartext vs. encrypted) data may be held on the basis of its classification, this is an issue that must be resolved.
3. Identify which information raises privacy concerns (for example, account numbers, dates of birth, addresses, etc.) or data residency concerns (for example, natural reserves data or government records).
4. Examine applicable regulations (especially in the finance and health domains) and determine what needs to be done to meet these regulations, and whether it is possible to meet these demands while migrating to cloud computing.
5. Perform the normal risk management tasks of assessing the risk of security or privacy violations, and the impact on the business.
6. Review the cloud providers' security/privacy measures (including physical security, personnel screening, incident notifications, etc., not just the technical security protection measures), and make sure that they are documented in the cloud SLA.
7. Determine whether the results of these steps actually allow the project to continue.
8. Consider and implement ways in which the information can be protected in four different situations:
    a. During the bulk migration of data from the on-premises system to the cloud service, when the cloud service is provisioned. This can be a weak point of the whole process, as an entire database backup may be carried physically, or shipped via courier, to the cloud service provider's site.
    b. "Data at rest," while stored in the cloud. An obvious solution for sensitive data is to encrypt it, and the practical question is whether the provider can perform this service, or whether the client needs to research and implement a solution.
    c. "Data in motion," during the routine exchange of data that occurs while using the cloud-based application. Encrypting data in transit is advisable, but runs into some issues: the

cloud provider must support the encryption chain, cryptographic keys may need to be installed at both ends (requiring a key management solution), and on-the-fly encryption may affect transfer speeds.

   d. "Data in use," that is when the data is actually read and processed by an application. For sensitive data, it may be advisable for the application to encrypt the data. This may not be possible if the migrated application is a commercial one that can only read the data in cleartext from a database. A customer-written application, on the other hand, can be modified to read/write encrypted data, so that only some temporary memory buffers will contain cleartext data. The handling of encryption keys is a concern.

9. Design how to authenticate and authorize users. For systems that have their own sign-on facility, there may be no impact (as long as passwords are not sent in clear text from the user's workstation to the cloud-based system, which should not be the case even for an on-premises system). But if there is any form of Enterprise or Single Sign-On (SSO) facility, making this work from an application running in a cloud service may require integration work. An enterprise identity and access management system (IdAM) needs to be accessible from the application migrated to the cloud service. You will need to understand which protocols are supported by the IdAM and by the cloud service – additional integration components may be required to enable them to interoperate.[3] The silver lining is that once that effort has been made for the first migration, it should make future migrations easier.
10. Regardless of the solution chosen for authentication and authorization, you need to make sure that your user *de-provisioning* process can be executed quickly. Disabling a user's credentials for access to cloud systems may be even more critical than disabling their access to an on-premises system. The reason is that access to an internal system may be made immediately impossible or more difficult if someone has been escorted out the door; but might still be able to access the login page of a cloud application from the browser on their smartphone. If a single federated IdAM is used, this risk is reduced since the user is removed once centrally for all applications.

Step 6: Manage the Migration

Finally, having thoroughly defined on paper the "why, what, and how" of the application migration project, the IT department can plan, execute and manage the actual application migration. Executing a migration is a complex and delicate project, and as such it should have a formal project plan and a skilled project manager. The migration plan, like all project plans, should track tasks, durations, resources, costs, and risks.

In general, it is recommended to conduct a pilot for one or two of the suitable applications, test thoroughly, document the lessons learned, and gather customer feedback so that improvements can be made before going live. Once the application migration process has proved successful and the required

[3] Example IdAM protocols include LDAP, OpenID, SAML, WS-Federation and Active Directory.

technical cloud computing and cloud migration skills have been developed, migration of more business-critical applications can be considered.

Table 2 highlights the key components and considerations of the application migration procedure.

Table 2 - Application Migration Procedure

Migration Procedure	Migration Details
1. *Deploy the Cloud Environment.* Provision, install and test the necessary storage, compute, network and security resources that constitute the cloud environment in which the migrated application will run.	● The first part of the cloud environment to be laid down is the structure of the virtual network. In a private cloud architecture, this would typically be done according to the organization's pre-established standards for network addressing. For public cloud services, however, the network structure is often prescribed in advance by the cloud service provider. For virtual private cloud implementations, connecting the VPN to existing internal networks may require significant work to match network addressing spaces, namespaces and other network aspects. ● Create individual virtual machines and attach them to their respective storage units. Reconfigure the domain name service (DNS) by updating the name servers to resolve the newly created VMs through the network gateways. ● Provision security devices including firewalls and VPN routers. Configure directory services access by implementing and testing the connections between the cloud service and the organization's directory server (LDAP, Active Directory, etc.) or, if specified by the architecture, the federation between the cloud service provider's authentication system and the customer's.
2. Implement Monitoring and Management Services and Components. Set up the organization, processes, procedures, and tools that will be used to manage and monitor the environment.	● Set up the organization that will be responsible for operating the environment. This may be a hybrid team composed of people from both the cloud provider and customer, working closely together to ensure service levels are met. ● Ensure the RACI matrix is well defined across the lifecycle, from design to build to run and manage. ● Define the processes and procedures required to manage the environment. The ITIL framework is one that can be used as a starting point. ● Ensure the management and monitoring toolchain is in place to execute the processes and to support functional and non-functional requirements (NFRs), including security. ● Setup the toolchain required for the continuous delivery of software and application changes.

Migration Procedure	Migration Details
3. *Install and Configure the Applications and supporting middleware.* Cloud service providers frequently do this through automated deployment of templates.	● Implement all integrations between cloud applications and other applications or resources, including directory services. ● All monitoring solutions should be implemented and tested, including any add-on monitoring tools. ● If the cloud application servers are to manage and monitor licenses, apply the activation kits and keys. If the existing monitoring and key services are to be reused, make and test the connections from the application servers to these resources.
4. *Harden the Production Environment*. Install additional utilities for business continuity and security. Some of these services may be provided by the cloud service provider, in which case they do not need to be installed, but they should still be tested.	● Put in place and test automated backup capabilities. ● Install and configure anti-virus software or malware protection. ● Issue to all project team members their initial credentials for cloud service access, according to their role in the project or ongoing operation.
5. *Execute a Mock Migration*. Undergo a trial run of the migration project plan to uncover unintended results or unnoticed issues during the planning phase. The mock migration date should be sufficiently distant from the desired final cutover date to have time to rectify problems. Involve the cloud service provider in the migration date selection.	● Ensure that all contractual aspects are in place with the cloud service provider, since the subsequent tasks will start consuming cloud services. ● Since it is important to simulate all aspects of the final migration, schedule downtime for the existing systems during the time required to make the migration, and notify users in advance. ● Import application data and configuration settings into the cloud environment. ● Run test scripts to validate application and data migration, connectivity from all endpoints, and proper access and authority. ● Start the cloud environment and the applications. ● Ask a preselected group of test users to validate that their work environments and systems are functional on the cloud-based system. These test users should follow formal test plans, designed in advance to exercise as many possible features of the applications within the allotted time. ● Restart the on-premises production environment. ● Document migration duration and metrics.

Migration Procedure	Migration Details
6. *Execute operational readiness testing.* This will help ensure that the operations team, processes, and tools are ready.	● Test processes such as incident and problem management and hand-off between the cloud provider's operations team and the customer's IT staff. ● Walk through the RACI matrix and validate that the team has each responsibility covered. ● Test execution of the Disaster Recovery plan. ● Simulate key failure modes such as loss of connectivity between the cloud environment (e.g., the system of engagement) and the enterprise on-premises systems (e.g., the system of record).
7. *Cutover to Production Cloud*. Assuming a successful mock migration, or one that only encountered minor issues with a clear fix, establish a formal cutover schedule. If the mock migration ran into serious issues, then it needs to be repeated after correcting the causes.	● Update the migration plan – pre-migration, migration and post-migration steps (usually called run-book) – taking into account the lessons learned during the mock migration about the tasks to be added or removed, the actual durations measured, the change in resources required vs. initially expected, etc. ● Make sure the run-book contains key resource names and contact details, as well as escalation instructions in case of unforeseen issues. ● Line up the necessary resources from the cloud service provider and other resources required during migration, which may be different from those needed for the mock migration. ● Schedule a meeting with all the key resources required to complete the migration. Share the run-book, address questions and concerns. These resources may need to be available for an extended period of time if unforeseen delays occur. All resources need to know the contingency methods for communication during migration. ● Communicate the migration plan and impact to all users (and a summary to their management), including application downtime and instructions for "day one" steps that individual users must perform to access cloud services. ● Re-execute the "Mock Migration" procedures but at the end, instead of restarting the old production environment, engage key application users to run required functional tests against the migrated application. Upon successful completion of the tests, consider releasing the application to the rest of the user base. ● Inform all users to apply the instructions they have received to restart their work using the migrated application. Ask users to report issues through normal channels (e.g., helpdesk). ● Begin license, application and database monitoring for the production environment. This monitoring continues indefinitely. ● For some time after the cutover, a special "hotline" should be established for triaging and solving issues during initial usage. ● Hold one or more formal checkpoint meetings after migration to track any issues, until resolution, that need additional project tasks and resources.

An even more detailed description of migration tasks is provided in Appendix B.0

Appendix A: Examples of Cloud-Ready Workloads

This appendix highlights some of the most cloud-ready workloads and the benefits of migrating them to the cloud.

Workload	Description	Benefits
IoT, Big Data, and Analytics	The analysis of massive data sets in near real-time or batch mode allows the generation of new information and intelligence about the business. The iterative exploration and investigation of past business performance provides insight and drives business planning. This is usually done on copies of operational data, not on the original, and therefore can run elsewhere.	• Reduce the capital and operating expenses needed to introduce or support enterprise wide BI services. • Scale up rapidly when a massive Business Intelligence project kicks off (and scale down rapidly when it ends).
Collaboration and Unified Communications	Federated collaboration tools provided to participants include voice, e-mail, presence and instant messaging, web conferences, file sharing, and enterprise social networking.	• Work beyond the boundaries of a single company and outside firewalls - share information more easily with customers, suppliers and business partners. • Affordable and accessible – lower upfront investment and extremely easy to acquire. • Eliminate individual departments' motivation to set up their own ad-hoc collaboration environments, which creates a confusing mix of often insecure solutions.
Development and Testing	Project environments that are used for all phases of the Software Development Life Cycle (SDLC) —and even production rollout, if the organization follows the DevOps approach. Development environments are used to design and build applications, test environments are used for various testing levels, including system integration, security, high availability, and user acceptance.	• Reduce the costs (capital, licenses) and labor to procure, configure, operate, manage, and monitor the environments, which may be different from project to project. • Reduce provisioning cycle times from weeks to minutes. • Advanced automation through DevOps tooling enabling implementation of Continuous Innovation and Continuous Deployment (CI/CD).

Workload	Description	Benefits
Sporadic and Seasonal Compute-Intensive Applications	Simulation and other CPU-intensive tasks are often needed for only a limited period of time, for example during the validation phase of a new electronic circuit design. During the next burst of CPU demand, weeks or months later, a different application may be required. Moving these applications to a rapidly scalable computing environment, such as a large cluster provided as a cloud service, is much preferable to buying those expensive servers.	• Resources provisioned in minutes rather than weeks. • Dynamic response to resource demand with elastic scalability. • Consumption-based usage charges. • No infrastructure to manage between bursts in demand.
Storage	This category includes on-demand access to sporadic, overflow, or specialized storage resources, including high-performance and highly available consolidated storage for demanding or critical applications. In addition, the demand for cloud-based backup is growing, and it allows mobile devices to be backed up from anywhere there is an Internet connection. Many storage management solutions now support the use of cloud services as part of the storage pool. Longer-term archival solutions are also now supported as cloud services.	• High degrees of scalability: petabytes of data and billions of files. • Professionally managed security that integrates into existing authentication systems. • Built-in data placement and Information Lifecycle Management (ILM), including backup, archiving and retention management, via a global policy engine. • Support for multiple tiers of storage including low-cost tape technology. • High performance and availability. • Access to online backup from anywhere without requiring a VPN connection into the enterprise network.

Appendix B: Sample List of Application Migration Tasks

Phase	Description
Determine Migration Strategy	The complexity of migrating existing applications varies, depending on the architecture and existing licensing arrangements. If you think about the universe of applications to migrate on a spectrum of complexity, you might put an existing virtualized, service-oriented architecture on the low-complexity end of the spectrum, and a monolithic mainframe at the high-complexity end of the spectrum. The following are the 6 migration R's [13]. ● **Rehost (Lift and Shift)** – a migration strategy that usually uses migration tools to replicate applications in a Cloud environment, without redesigning the application [14]. This is indicated if the answers to the following questions are affirmative: a. Is this a legacy application with limited access to the underlying source code? b. Does this application rely on underlying IT infrastructure for availability, security and data protection services? c. Is this application tightly coupled together with no ability to separate the application components? ● **Replatform (Migrate and Redesign to Cloud-Ready Application and Infrastructure)** [15] – Cloud-aware applications follow four different levels of maturity: (i) Virtualized, (ii) Loosely Coupled, (iii) Abstracted and (iv) Dynamic. Cloud-ready applications are typically virtualized but still rely heavily on the core infrastructure for security, availability and data protection. Related questions: a. Is this application deployed in a virtual environment today? b. Does the application support its own security considerations independent of underlying hardware or systems? c. Does the application support failover of core services? d. Is the application loosely coupled with separate individual sub-components that can be managed and controlled individually? e. Is the application dependent on underlying IT infrastructure services for security, availability, and data protection services? f. Does the application provide control or interaction via APIs? ● **Refactor to Native Cloud Application (NCA)** [16] – NCAs are designed to take advantage of cloud computing frameworks, which are loosely coupled, abstracted and dynamic with minimal to zero dependency on the underlying infrastructure for availability and data protection. Relevant questions: a. Is the application to be written using advanced cloud architecture practices? b. Do you have access to the application's source code and APIs? c. Is the application to be available for deployment in multiple locations across multiple clouds?

Phase	Description
	d. Is the application or software built using microservices? e. Can the application be deployed in a container-based architecture? f. Does the application support dynamic failover and recovery with limited-no dependencies on the underlying infrastructure? ● **Repurchase** – Moving to a different product, most commonly a functionally equivalent SaaS solution. For example, moving from an on-premises Customer Relationship Management (CRM) package to Salesforce.com, from an HR package to Workday, from a Content Management System (CMS) to Drupal, etc. ● **Retain** – A solution to be considered by customers who are still riding out some depreciation, are not ready to prioritize an application that was recently upgraded, or are otherwise not inclined to migrate some applications. Only migrate what makes sense for the business; later, as the portfolio progressively shifts from on-premises to the cloud, reasons to retain applications in-house will decrease. ● **Retire** – After discovering everything in your environment, ask each functional area who owns and uses each application. Sometimes, 10 to 20% of these applications can simply be turned off (e.g., old Exchange servers, Web servers, and data stores). These savings can boost the business case, focus IT's scarce resources on the things that people do use, and shrink the attack surface you have to secure. **Budgeting for Multiple Migration Initiatives –** In order to get a budgetary estimate, first determine what type of migration is required for each individual application or workload. Second, estimate the "T-shirt size" of each migration (e.g., S/M/L/XL) and the related average cost. Next, a simple math formula can yield a high-level budgetary estimate (for example, for migrate-to-cloud-ready applications, multiply the cost of a medium effort by the number of applications in that category).
Initiation	1. **Kick-off Meeting** – This event sets the stage for the migration, and should include all members of the project team. Business stakeholders must be included so they to understand the scope, duration, resources, and to discuss the success factors and the success criteria that will be used at the end of the entire process. **2. Gather Technical Discovery Data** – The objective of this step, which can be started at the kick-off meeting but will usually be completed through interviews and e-mail, is to understand every possible technical objective and constraints. For instance, the amounts of data to migrate, the required availability of the systems during migration, and network performance targets might all be discussed. 3. **Identify User Load Parameters** – Discover and document groups of users, and the pattern of their use. For example, are there teams of contractors in foreign countries? When do they need access to the system? Where are the major groups of internal users? Are there any transient users, such as management, that should

Phase	Description
	be counted? Is there a particular time of the day or of the month when usage is especially heavy? 4. **Open Trouble Ticket System** – Put in place a trouble ticketing and issue resolution process. While this might be considered early in the process based on traditional projects, this establishes a clear discipline and method of communication and tracking, and logging issues through a traceable system is important from the start.
Design the Production Architecture	1. **Determine integrations** – Decide how each interdependency between the migrated application and the on-premises systems that are not moving to the cloud will be supported. This frequently involves the establishment of sub-teams of subject matter experts to discuss these integrations. Refer to Step 4 of this guide's body. 2. **Specify Database Product and Version** – A critical part of nearly any IT system stack is the database engine, and this step is to specify the version level and operational information for the database core. This is a critical part of the cloud architecture, as data currency, replication and security control are paramount. 3. **Specify the Network Topology** – Look carefully at the network aspects of the architecture. Consider wide-area network and local-area network performance characteristics relative to the current and future locations of server and storage assets. Hop counts, latency and reliability of network links should be measured, and targets for post-migration performance established. 4. **Specify a Directory Architecture** – An aspect nearly as difficult as database integration is integration with corporate/organizational directory resources. Many organizations lack a single directory to serve as the "authoritative source" for access control and role-based permissions, and adding cloud computing to the architecture can often complicate the issue. Examining the directory status and desired target architecture is critical. 5. **Document Architecture Details** – Produce a complete architecture document that will serve in later stages as a guidebook for the implementation.
Deploy the Cloud or Hybrid Environment	1. **Provision Virtual Infrastructure** – The first part of the cloud environment to be laid down is the structure of the virtual network. For public clouds, the network structure is often prescribed in advance by the cloud service provider, for the purposes of maintainability and automation. For virtual private cloud implementations, connecting the VPN Virtual LAN to existing internal networks may require significant work to match network addressing spaces, namespaces and other network aspects.

Phase	Description
	2. **Provision the Edge Firewall** – This is the gateway to the cloud service, and usually involves creating some sort of virtual private network Uniform Resource Identifier (URI) on the edge of the internal network and tying it to on-premises network resources. 3. **Provision Storage** – Create the logical units on the cloud's Storage Area Network (SAN), according to the architecture documents. 4. **Deploy Virtual Machines** – Create individual virtual machines and attach them to their respective storage units. 5. **Reconfigure the Domain Name Service (DNS)** – Update the name servers to resolve the newly created VMs through the network gateways. 6. **Test Network and Server Connectivity** – Fully test the network connectivity, noting performance characteristics and measuring them against the desired targets from the architecture. 7. **Update Documentation** – Update the documentation with the test results and any modifications made from the initial architecture. 8. **Configure Site-to-Site VPN** – If the architecture requires a static site-to-site VPN connection, implement and test it. 9. **Configure Directory Service Connectivity** – Implement and test the connections between the cloud service and the organization's directory service (LDAP, Active Directory, etc.) or, if specified in the architecture, the federation between the cloud service provider's authentication system and the customer's.
Install and Configure Applications	1. **Install Server Software** – Install and configure the application server software on the cloud servers. Cloud providers frequently do this through automated deployment of templates. 2. **Implement the Database** – Implement the database per the architecture. Again, this is often done through automated, template deployment. 3. **Configure the Application** – Configure the application servers and tools as specified, including applying any customizations or templates. 4. **Enable License Tracking** – If the cloud service is to manage and monitor licenses, apply the activation kits and keys. If the existing monitoring and key services are to be reused, make and test the connections between the application servers to these resources. 5. **Implement Integrations** – Implement all integrations between the migrated application and on-premises resources.

Phase	Description
	6. **Configure Monitoring** – Implement and test all monitoring solutions, including SNMP (Simple Network Management Protocol) services and other add-on monitoring tools.
Harden the Production Environment	1. **Configure Anti-Virus** – Ensure that the cloud service provider installs and configures sufficient anti-virus software or malware protection. 2. **Configure Database Backups** – Implement any specific procedures or servers used to back up the application data (a database backup often requires specific techniques other than ordinary disk backup). 3. **Establish Password Change Mechanism** – Depending on the cloud service provider, there may or may not be automated mechanisms for password and ID changes. Implement and test the process for these frequently used and security-sensitive operational systems, including notifications of changes to appropriate managers. 4. **Obtain and Install SSL Certificates** – For any access secured through SSL (secure browsing or SSL VPN), install the signed certificates. 5. **Establish Management IDs** – Issue to all project team members their initial credentials for cloud service access, per their role in the project or ongoing operation. 6. **Establish User IDs** – Load user IDs and initial passwords into the directory, unless existing directory resources are serving as the credentials source.
Execute Mock Migration	1. **Set Migration Date and Schedule** – The purpose of this set of steps is to undergo a "mock migration," which is a trial run of the project plan to uncover unintended results or issues overlooked during the planning phase. The date should be sufficiently distant from the desired final implementation/cutover date to have time to rectify problems. Involve the business users and the cloud service provider in the migration date selection. 2. **Activate Cloud Services Agreement.** Ensure that all contractual aspects are in place with the cloud service provider, since the subsequent tasks will start consuming cloud services. 3. **Notify Users of Outage** – Since it is important to simulate all aspects of the final migration, schedule downtime for the existing systems during the time required to make the move to the cloud service, and notify users in advance. 4. **Stop Applications** – At the appointed day and time, stop the current application servers and shut down on the current production environment.

Phase	Description
	5. **Capture Database Backups** – Make complete backups of the on-premises databases that will be migrated. Execute validation scripts to ensure the integrity of the backed-up databases. 6. **Export Application Configurations** – Export configurations and customizations from the servers that will be migrated. 7. **Import Application Configurations** – Apply the exported configurations and customizations to the target application servers in the cloud. 8. **Configure Manual Settings** – Apply any additional settings that did not migrate in the earlier configuration export/import. 9. **Restore Database** – Create the cloud-based databases by restoring the validated backups of production data created earlier. 10. **Start Cloud Application Servers** – Restart the cloud application servers and test for integrity and access to data. 11. **Run Database Validation Jobs** – Run a validation of the database again to ensure integrity. 12. **Compare Source and Destination Data** – An extra desirable validation step is to compare sample data from the source and target systems for extra integrity validation. 13. **Customer Validation and User Acceptance Test (UAT)** – Grant a pre-selected group of test users access to the cloud-based system to validate that their work environments and systems are functional. Make sure users understand that they will enter mock transactions, and that if they take them from their real workload, they will have to re-apply them to the current system after the test. Test all user access methods (Web, mobile, etc.) and locations for connectivity and performance. 14. **Test Authentications** – Test samples of all roles and authentication mechanisms for accessibility. 15. **Document Migration Duration and Metrics** – Document all migration steps and performance characteristics in the project plan. 16. **Restart the on-premises application.** Perform any steps necessary to terminate the mock migration, restoring access to the current environment and application.

Phase	Description
Production Cloud Migration	1. **Formalize Cutover Schedule** – If the mock migration was unsuccessful or exhibited major issues, go back to the appropriate phase in order to correct the problems and run a new test. Assuming a successful mock migration, establish and communicate a formal cutover schedule that takes into account the lessons learned from the mock migration on how much time each task really took. Include the scheduling of all necessary resources from the cloud service vendor. 2. **Communicate Changes to Users** – Communicate the migration steps, timeline and impact to users, including instructions for day-one steps that individual users must perform to access cloud services. Inform users of the procedure to report issues, and train the Help Desk on the new trouble ticket types and escalation rules for migration-related issues. At this point, all the steps of the mock migration should be repeated, followed by these additional steps: **3. Open Cloud Migration Hotline** – For some time after the cutover, a special “hotline” should be operated to triage and resolve issues. 4. **Flip DNS** – The relevant domain records should be changed to point to the cloud services. 5. **Migration Go/No Go Meeting** – Hold a formal rollback/proceed decision meeting between all stakeholders, including the users affected by the migration and the cloud service provider. 6. **Enable License Monitoring** – Begin the license monitoring for the production cloud service. This process will continue for the life of the application. 7. **Configure Application Monitoring** – Begin application and database monitoring for the production cloud service. This process will also continue indefinitely. **8. Post Migration Checkpoint** – Hold a first formal checkpoint meeting shortly after migration to assess any large-scale issues that need additional project plans and resources. This meeting ends with a decision: has the system reached sufficient stability and productivity that this is now “business as usual?” If so, decide to close the project. If not, assign corrective actions, communicate the actual plan to management and to users, and schedule the next checkpoint meeting. **9. Project Closure.** Archive all relevant documents, release any temporary resources assigned to the migration, document lessons learned, celebrate success and reward key contributors.

Phase	Description
A Factory Approach to Large Scale Migrations	When conducting the migration of multiple workloads or applications, a scalable approach is often preferable. Sometimes referred to as an “Application Migration Factory”, this programmatic approach enables multiple project teams to leverage shared, or pooled, resources to accelerate migrations. An Application or Cloud Migration Factory fulfills the governance requirement for executing a selected cloud strategy, moving both applications and workloads to pre-determined cloud service providers. This factory approach provides the policies, processes, and tools to consistently, efficiently, and securely migrate applications and workloads while protecting existing operational capability during the transition. There are a number of roles and responsibilities that can contribute to the overall success of a factory migration approach. These may include, but are not limited to: 1. **Migration Program Management Team** – This team is instrumental in promoting project success and consists of: • Program Manager • PMO Reporting and Analytics Lead • Cloud Strategy Lead • Key stakeholders (Application Owner, Executive Sponsor, and others) 2. **Migration Project Teams** – Depending on the scope of the overall program, multiple project teams may be required. In this case they are often aligned with the type and scope of individual migrations. Although this is not an exhaustive list, the project team may include several of the following roles: • Project Manager • Application Lead • Operating System Lead • Language, Interface, and Middleware Lead • Database Lead • Quality Assurance (QA) or Testing Lead • Documentation Lead • Service Delivery Manager (Compute, Storage, Network) 3. **Shared Resources** – Database Administrators, Network Engineers, Storage Engineers, Cloud architects, Security Engineers, etc. 4. **Third-Party Providers** – Consulting Partners, Cloud Service Providers (CSPs), Software as a Service (SaaS) Providers, etc. 5. **Escalation Paths** – For each migration, it is critical to adequately define escalation paths for rapid resolution of any outstanding issues.

Appendix C: Bibliography

Works Cited

[1] Cloud Standards Customer Council: *Practical Guide to Hybrid Cloud Computing.* 2016. http://www.cloud-council.org/deliverables/practical-guide-to-hybrid-cloud-computing.htm

[2] Cloud Standards Customer Council: *Practical Guide to Cloud Computing*. 2017. http://www.cloud-council.org/deliverables/practical-guide-to-cloud-computing.htm

[3] Cloud Standards Customer Council: *Practical Guide to Cloud Service Agreements*. 2015. http://www.cloud-council.org/deliverables/practical-guide-to-cloud-service-agreements.htm

[4] Cloud Standards Customer Council: *Public Cloud Service Agreements: What to Expect and What to Negotiate*. 2016. http://www.cloud-council.org/deliverables/CSCC-Public-Cloud-Service-Agreements-What-to-Expect-and-What-to-Negotiate.pdf

[5] Fehling, Christoph, Frank Leymann, Ralph Mietzner, Walter Schupeck: *A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-based Application Architectures*. University of Stuttgart Institute of Architecture of Application Systems, Report 2011/05, May 2011. http://www.iaas.uni-stuttgart.de/institut/mitarbeiter/fehling/TR-2011-05%20Patterns_for_Cloud_Computing.pdf

[6] CloudDesignPattern.org: *AWS Cloud Design Patterns.* November 2013. http://en.clouddesignpattern.org

[7] Microsoft Corp.: *Cloud Design Patterns.* 2017. https://docs.microsoft.com/en-us/azure/architecture/patterns/

[8] Zhi Xian Chen et al.: *IBM Workload Deployer – Pattern-based Application and Middleware Deployments in a Private Cloud.* IBM Corporation, March 2012. http://www.redbooks.ibm.com/redbooks/pdfs/sg248011.pdf

[9] Cloud Standards Customer Council: *Convergence of Social, Mobile and Cloud: 7 Steps to Ensure Success*. 2013. http://www.cloud-council.org/deliverables/convergence-of-social-mobile-and-cloud-7-steps-to-ensure-success.htm

[10] Cloud Standards Customer Council: *Security for Cloud Computing: 10 Steps to Ensure Success, Version 3.* 2017. http://www.cloud-council.org/deliverables/security-for-cloud-computing-10-steps-to-ensure-success.htm

[11] Gilbert, Françoise: *EU General Data Protection Regulation: What Impact for Businesses Established Outside the EU and EEA?* Cloud Security Alliance, November 2017. https://gdpr.cloudsecurityalliance.org/wp-content/uploads/sites/2/2017/11/EU_GDPR_Impact_for_Businesses-Established_Outside_the_EU_and_EEA.pdf

[12] Cloud Standards Customer Council: *Data Residency Challenges – A Joint Paper with the Object Management Group.* May 2017. http://www.cloud-council.org/deliverables/data-residency-challenges.htm

[13] Orban, Stephen: *Six Strategies for Migrating Applications to the Cloud*. Medium.com Amazon Web Services Collection, November 2016. https://medium.com/aws-enterprise-collection/6-strategies-for-migrating-applications-to-the-cloud-eb4e85c412b4

[14] Knapp, Kristin: *When to Adopt the Lift-and-Shift Cloud Migration Model*. TechTarget, July 2015. http://searchcloudcomputing.techtarget.com/feature/When-to-adopt-the-lift-and-shift-cloud-migration-model

[15] Open Data Center Alliance: *Architecting Cloud Aware Applications.* https://opendatacenteralliance.org/article/open-data-center-alliance-best-practices-architecting-cloud-aware-applications-rev-1-0-2/

[16] Stine, Matt: *Migrating to Cloud-Native Application Architectures*. O'Reilly report. http://www.oreilly.com/programming/free/migrating-cloud-native-application-architectures.csp

[17] CIO Magazine: *The Cloud Journey, Part 3: Why A Cloud Migration Factory Is the Catalyst for Cloud Enablement*. November 2017. https://www.cio.com/article/3237585/cloud-computing/the-cloud-journey-part-3-why-a-cloud-migration-factory-is-the-catalyst-for-cloud-enablement.html

Additional References

Bridgewater, Adrian: *Cloud Migration: The Problem with Legacy Software.* CloudPro, August 2012. www.cloudpro.co.uk/saas/4249/cloud-migration-problems-legacy-software

Ul Haq, Salman: *Issues in Migrating Legacy Systems to the Cloud.* Cloud Tweaks, July 2013. www.cloudtweaks.com/2013/07/issues-in-migrating-legacy-systems-to-the-cloud/

Vellante, David: *IT's Online Enterprise Integration Crisis.* Internet Evolution, March 2010. www.internetevolution.com/author.asp?section_id=654&doc_id=188706

Varia, Jinesh: *Migrating Existing Applications to the AWS Cloud.* Amazon Web Services, October 2010.
http://media.amazonwebservices.com/CloudMigration-main.pdf

Strauch, Steve, et al.: *Using Patterns to Move the Application Data Layer to the Cloud*. Proceedings, PATTERNS 2013 conference. http://www.iaas.uni-stuttgart.de/RUS-data/INPROC-2013-17%20-%20Using%20Patterns%20to%20Move%20the%20Application%20Data%20Layer.pdf