Cloud Computing Standards, Compatibility and Interoperability:
Ensuring a thriving competitive market

An Executive Summary

Businesses, public administrations and individuals are eagerly embracing cloud computing as the next key technology driver. And for good reason: in the ongoing advancement of computing architectures, cloud offers unprecedented flexibility, agility and productivity. As technology has evolved from mainframe computers with customized software to networked desktop systems to the Internet, an ever higher premium is being placed on the ability of disparate systems to seamlessly interact. Cloud computing, if effectively implemented, will put more freedom and power in the hands of the technology user than ever before.

While organizations are rushing to cloud computing, however, it is essential that enterprises do not overlook the implications of switching vendors in the future. Care needs to be taken up front in order to assure maximum freedom and flexibility.

Open interfaces and data formats—based on open standards—are key. An interface is the software that serves as the conduit by which systems, programs and applications interact. Closed or proprietary interfaces cede many key decisions and options to the discretion of the cloud provider. Closed interfaces and data formats may also limit the ability of the user to efficiently transfer their data in the future. This is the very definition of lock-in.

It is therefore important to have a meaningful discussion in relation to the nuances of cloud adoption and to the implications of switching vendors. Enterprises need to look well ahead and anticipate that they may wish to move part or all of their data to another cloud environment or they may want to move certain implementations into or out of the cloud that they're seeking to use or build.

This paper describes the evolution of cloud architecture, discusses the key differences of the main cloud implementations and highlights the key role that interfaces and data formats play, as well as some of the exit and migration issues that should be considered. This is intended to initiate a broad discussion on the importance that standards, compatibility and interoperability play in assuring that the promise of enterprise cloud computing can be fully realized in Europe.

ECIS is an international non-profit association founded in 1989, which endeavours to promote a favourable environment for interoperable ICT solutions. It has actively represented its members regarding issues related to interoperability and competition before European, international and national fora, including the EU institutions and WIPO. ECIS' members include large and smaller information and communications technology ("ICT") hardware and software providers. More information on ECIS is available at www.ecis.eu.

1 INTRODUCTION

The world is abuzz with the latest technology driver - Cloud Computing. It is not just about how you use your smartphone or tablet device, or their apps. Cloud computing is pervasive within the broader IT market and will continue to have an increasing economic and innovation impact on a wide spectrum of enterprises, consumers and competition. It is a worldwide phenomenon, and at its cornerstone it is the move away from traditional ownership of IT infrastructure and software to a dynamic set of resources shared over networks.

The buzz about cloud computing obscures a central fact - there are actually a multitude of ways to build your own cloud or use third party cloud computing solutions. There is no one-size-fits-all for the cloud. Depending on any given customer's[1] needs, existing technological environment, business model and the resources they have available, enterprises may choose from several different types of third party cloud services within the market. Additional choices are available for organizations that decide to build their own cloud services. Many technology vendors offer a variety of competing platforms that can be used to build and utilize cloud services, and some vendors have already established an early leadership presence in the market for cloud computing.

Education is crucial because *the choices a business makes when building or using a cloud service will have a direct, immediate and long term effect on the ability - or lack thereof - to leverage the value of competitive alternatives*.

One lesson that was learned with the rise of enterprise computing, and the Internet in general, is that proprietary components of any solution can limit the availability of competitive alternatives *by design*, which can have an enormous impact on the overall market. It is important to have a basic understanding of the variety of cloud computing types and how they can potentially affect competition.

At this critical moment, enterprises should have a meaningful discussion to explore the nuances of cloud adoption **and** should not overlook the possibility of switching vendors in the future. This paper highlights some of the *exit strategy* and *migration issues* that should be considered.

[1] The term Customer is used in this paper to convey a party that engages in a business relationship with one or more cloud providers. A customer may be an enterprise or a public body which may build its own cloud or use third party cloud computing solutions.

2 Cloud architecture overview

2.1 Evolution of the Cloud Ecosystem

People use technology to *do something*. Whether the objective is to communicate with others, to calculate a company's payroll or to manage complex manufacturing, there are a multitude of products available to address customer needs. Traditionally, products have been offered via software packages or application programs which a customer would license, install and maintain in order to solve their need to carry out a task. In the very first computer systems, these applications were programmed directly onto the hardware – nothing was compatible or portable. As the industry has evolved from centralized computers to networked desktop (or client) systems and the World Wide Web, market demands have also driven the need for increasing flexibility and adaptability for the components used to create software. We now have a commonly accepted process within which technology vendors (a) establish a new technology, (b) abstract to identify common functions and then (c) standardize the interface to those sets of functions. This "Establish Abstract Standardize" process has been continually repeated: first with hardware and operating system interfaces and then with software packages, and from centralized systems through to the distributed, networked world in which we live in today. By standardizing interfaces and data formats that pass across these interfaces, we enhance interoperability and allow vendors to compete on quality of implementation. This improves their ability to innovate by providing extended functionality. It also has resulted in significantly more flexibility in accessing software functionality beyond what was originally envisioned. For example, the value of the World Wide Web has exploded, as additional data sources have been created together with the huge variety of clients that can access that content. Mobile and smartphone computing was largely streamlined by the prior investments made in desktop and client/server platforms around open standards for data formats and protocols.

Moving from the mainframe environment, which was centrally administered and had single deployments, into the desktop world required each user to acquire an appropriate license and install the software on their computer before they could use that specific software. With one single mainframe, or a small number of users, this model is manageable, but as the number of users and desktops grow, the overhead cost for managing a separate copy of every software package on every computer and the associated licensing becomes overwhelming. The client/server model evolved to address this overhead cost and allowed servers to host functionality that could be consumed by many clients. Unfortunately, most software designed for the client/server model still had very specific requirements as to which client could access which server, since these two entities were typically designed and developed together as a package. The client/server model really only fixed part of the problem. With the advent of the World Wide Web and the pervasiveness of the web browser and related protocols, the client/server model has evolved and broken the dependency between client and server software. However as we move into an era of web based service models the dynamic nature of server interfaces brings its own set of issues which we now explore.

As the pace of technology utilization within business has increased, some important market requirements can no longer be met by traditional software deployment models. The businesses that can react the fastest to the market - the most *agile, cost effective and/or the high performers* – are likely to be the most successful. Cloud computing, at its core, is reshaping information technology by changing the way technology is acquired, provisioned, deployed and utilized. Cloud enables quantum leaps in agility, while significantly reducing costs. It challenges traditional ownership of IT resources and software through the concepts of elasticity of provisioning, "pay for what you use" and changes to capital and operational expenditures. This has been possible because Service Oriented Architecture (SOA) concepts have been embraced, breaking up software into separate functional components which are accessed through interfaces. *Agility* is increased by reusing and composing different components and by adding new components with additional or different features. This in turn allows features to be added faster, which increases agility further. Thus, cloud computing enables much more flexibility for the enterprise in that it allows new offerings to be composed by combining services.

The promise of cloud computing is that it creates a huge ecosystem of flexible, related options for businesses. This in turn leads to more flexible implementation methods, automated deployment and management of new functions and the software and hardware resources they require. However, as is the case with any machine or system: if two parts don't fit together properly, they won't work. Thus, with increasing importance of decomposed parts within an architecture, there must also be a concurrent focus on well-defined, up-to-date, accurate and open *interfaces* for the services where the various parts can fit together.

2.2 Why are there different types of cloud implementations?

There are many reasons why there is no-one-size-fits all approach to implementing cloud – either through building your own cloud or though using a third party cloud solution. The variety of cloud solutions are dependent on and are driven by different business models. These different solutions and business models provide options in how flexible the cloud solution is, how agile it is and to what degree it can scale. It is important to bear in mind that modern systems must be able to scale outwards (in other words to increase their performance and capacity) if they are able to support rapid increases in demand.

Although there are many emerging types of services and architectures in cloud computing, the most prevalent are grouped into three types: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). These can be provided as private, public and hybrid implementations, as well as aimed at specialist market sectors and functions such as finance, healthcare and HR.

Enterprise customers want to invest where they believe they can get the biggest return. Prior to cloud, customers *had* to invest in building their own infrastructure layer to support software functionality. With IaaS, they no longer have to build their own infrastructure, although they still can if they choose to. Similar options are available with PaaS and SaaS. The key is to keep flexibility over time. A customer focusing *today* on innovating though a specific PaaS solution might find over time that their innovation is being hampered by their cloud platform provider as they are unable to move to another PaaS vendor due to proprietary roadblocks.

Software as a service (SaaS)

Software as a service (SaaS) is probably the most familiar kind of cloud service, where the focus is more on what software functionality is *delivered* to a customer as opposed to how it is implemented. For example, a cloud-based productivity suite can edit documents online instead of requiring the purchase of a proprietary, licensed product. There are advantages to both provider and consumer here: the provider enjoys more consistent, regular revenue streams and the customer no longer has to build and manage the infrastructure necessary to use the software; the customer no longer needs to manage multiple copies of the application software, with all the maintenance and updating/patching processes, and hence reduces costs.

SaaS offerings are dependent upon a network by their very nature. The Internet itself which serves as the network is much more ubiquitous now than it was when the idea of online software first came about. Also, web technologies have evolved to create a robust, consistent platform that is universally deployed across the gamut of potential client devices via the web browser. This removes the need for software vendors to have to create a separate, custom software client for every function, although vendors can still write specialized and enriched client-side applications such as for mobile and smartphone environments. Finally, a multitude of functions can be consumed not only by humans, but also by machines (e.g. other technology environments) and this provides unprecedented opportunities for enterprises to be more flexible and agile.

Infrastructure as a Service (IaaS)

Infrastructure as a Service (IaaS) focuses on the delivery of physical or virtual computing resources, such as disk drives, on which software can execute. To put it another way, IaaS is about providing processing, storage, networks and other fundamental computing resources. Why is this so important to cloud computing? Now customers can create new virtual compute, disk-drive or physical storage resources in a matter of minutes, on demand, at a fraction of the cost without needing to own the physical resources. Dynamic provisioning based on immediate, changing needs means you don't have to build and own equipment to meet peak demand or highly volatile workloads.

To appreciate the value of this, you have to consider what customers have had to do in the past. Traditionally, customers had to purchase all necessary hardware resources, install and configure all the hardware and software resources and plan for the subsequent management of them. This often resulted in very time-consuming and error-prone tasks; just ordering and receiving hardware might take weeks or even months. Further, to support the largest potential workloads desired, systems had to be over-provisioned or built to support much more capacity than they used on average. This is a costly venture. IaaS has emerged as a way to shorten timescales, decrease costs and aid planning.

Now with IaaS, customers only have to rent (either literally in a public cloud context, or figuratively in a private cloud) whatever hardware resources they need at any given moment, and therefore only pay for what they consume. IaaS resources can be allocated in mere minutes, and on-demand, resolving the delays associated with hardware acquisition and installation. With this model, cloud platform providers offer hardware resources such as CPU, network and storage devices on call.

However, although this model allows great flexibility and solves some hardware acquisition problems, customers still need to configure and manage the individual components that comprise IaaS platforms. This management overhead can be non-trivial since each instance of a cloud service running on an IaaS platform, as well as all the resources it requires, must still be actively maintained. Management overhead scales up directly with the number of resources in use; more virtual instances require more management overhead. Since IaaS focuses on the fundamental level of systems-management, IaaS platforms usually target IT professionals (and, to a lesser extent software developers). Yet this may not solve the issue that enterprises need to access mission critical information. Bear in mind the risk that the mechanical digger could rip up your communications lines. Cloud does not necessarily mean everything is online, rather it is an important part of a holistic enterprise solution.

Platform-as-a-Service (PaaS)

This brings us to the third service type: PaaS.

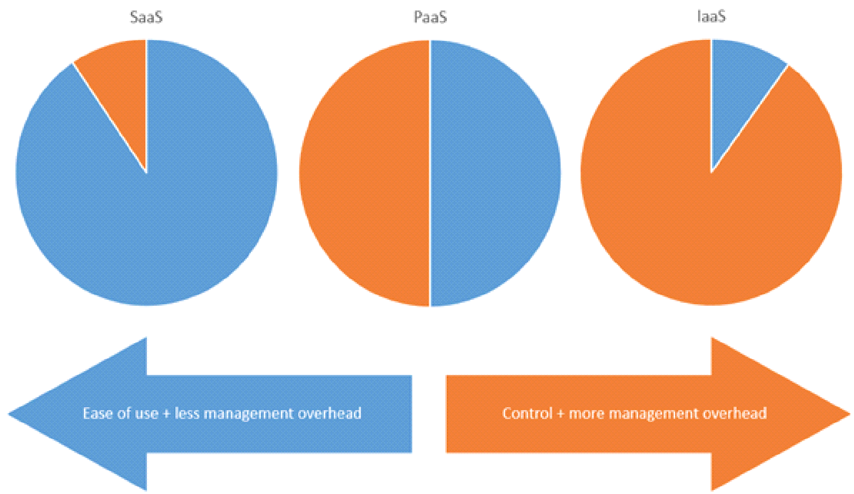

Figure 1: SaaS/PaaS/IaaS compared

PaaS allows software to be deployed without the cost and complexity associated with acquiring, configuring and managing every instance of the underlying hardware and software. It provides operating systems, middleware and software development frameworks. PaaS platforms automate the allocation of resources necessary to support *all* of the services running on it dynamically. They allow developers to focus on the functionality that their service is providing, and not worry about the underlying plumbing that IaaS focuses on. In PaaS platforms, applications can either be delivered in-house or can be purchased, acquired and then deployed to provide services. PaaS can automatically scale up or scale down services by increasing or decreasing the amount of resources available to them, so they can adapt to changes in workloads. Traditionally this scaling was accomplished via a manual process of adding more resources (e.g. hardware systems), which added to the management burden for each service. This automation and abstraction allows developers to improve their productivity – they can write or deploy software more quickly and with more agility.

Software developers build applications and services that access the features of PaaS platforms in a similar way to how they used traditional operating systems over the past decades and more recently via middleware: by calling interfaces to the platform. However, instead of installing software on specific systems for end users, PaaS developers simply push their software into the cloud without necessarily knowing - or even caring- which physical systems will actually run it. To a developer, a PaaS platform looks like one large unified system as opposed to a group of separate systems that are connected to each other.

The power of PaaS is that it can standardize and streamline an entire development workflow including the deployment of an application into production, through its ongoing maintenance and up to its retirement.

Finally, it should be noted that these different types of services - SaaS, IaaS and PaaS - can be combined and merged. For example, a company could build a cloud using disk space provided by an IaaS platform and deliver that to end-users via the SaaS model.

2.3 THE IMPORTANCE OF INTERFACES

Cloud computing uses specific, well-defined interfaces to create an abstraction of the underlying implementation of the service which provides the supported functionality. Conceptually, an interface represents the *only* mechanism by which the cloud service can be accessed - it is a specification that a cloud service publishes to customers. As long as the interface itself does not change, the underlying implementation of a cloud service can be modified by a vendor without affecting any existing clients. For example, by using a consistent interface, a cloud service vendor may choose to build a completely new version of a cloud service that has significant performance improvements over the original. This new service can be deployed without requiring any changes to existing clients. Additionally, if an interface to a cloud service is an open standard, one implemented by multiple vendors, a customer can choose to access any of those offerings without making major changes to their clients. Like the pieces of a puzzle, interfaces define the size and shape of cloud services and how they are expected to interact with other parts of the overall solution. If the interfaces supported are identical, different cloud services are *compatible* and can be interchanged at will. The choice is up to the customer: whether they find another cloud service that is faster, more reliable or cheaper, as long as the interface to it is identical the customers is free to select whichever implementation is best for their immediate needs. It is important to understand that different users will place more or less value on different interfaces. For example, end-users that consume a SaaS offering will place the most importance on the user interface of an offering and what features it supports. Software developers on the other hand will focus on the specific Application Programming Interfaces (APIs) that a PaaS platform supports. IT Pros will focus on what hypervisors, virtual machines and operating systems are supported by an IaaS platform. The value chain for any cloud platform is directly proportional to the *compatibility* and *interoperability* of the interfaces it supports.

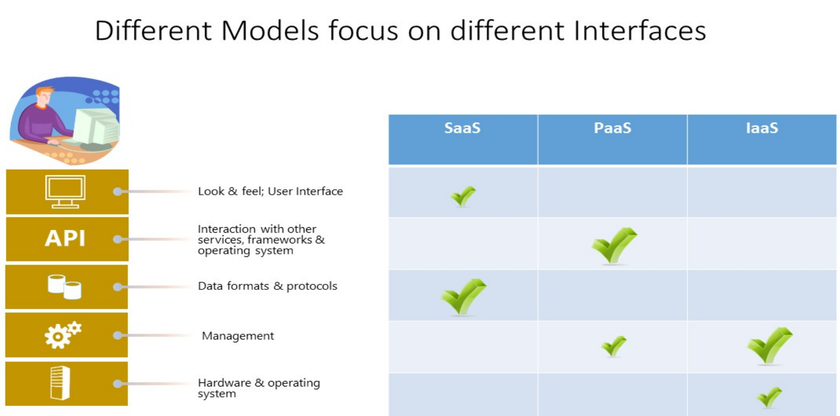

Figure 2 : Different focus on interfaces

The following table compares the different cloud models and what elements drive competition as well as what may hinder competition.

Model	Main interface(s) level	What drives competition	What hinders competition
SaaS	Human/consumer	Feature set, Open file/document formats, consistent user interfaces, ability to “export” data to competitive offering	Proprietary formats & protocols, dependencies on proprietary services, lack of data “export” capability
PaaS	Software frameworks	Open Source/Open standards; choice of programming languages, tools, scaling	Proprietary APIs, tools, frameworks; dependence upon proprietary platforms
IaaS	Operating Systems (Hardware resources)	Open Source/Open standards; cost, functionality	Proprietary APIs, proprietary hypervisors, limited operating system support

Table 1: Cloud models compared

2.4 Different types of services have different “Interfaces of Interest”

At some level, *every* cloud service might be considered as software that is running *somewhere*, providing some level of functionality to *someone*. Simply put, end-users likely won’t (and shouldn’t) even care what architectural type(s) have been used by any given product they purchase or consume. However, *to those enterprises that are building services and solutions, or utilizing those provided by third parties, for cloud computing* there are distinct advantages and disadvantages to different models. To represent the value offered by each given type, there are several industry definitions that can aim to define what functions the interfaces are providing, i.e. what the focal point is from an implementation’s perspective.

Type / Model	Main interface(s) (function)	Value provided	Target audience (“user”)	Attraction
SaaS	Human/consumer “use the features, don’t manage the app”	Automates provisioning, deployment and management of application functionality	End-users	Pay for functionality you actually need/use
PaaS	Software frameworks “manage the service or the app, not the server”	Automates provisioning, deployment, management and scaling of underlying operating system and hardware	Developers	Focus on business needs; no overhead for managing OS, hardware, network, storage, etc.

IaaS	Operating Systems (Hardware resources) “instantly/dynamically align resources needs with demand” manage the service, the app and and the virtual hardware	Automates provisioning and deployment of physical hardware resources for Virtual Machines	IT Pros, Developers	CapEx->OpEx, no need to pay for hardware you rarely use (“overprovision”), agility

Table 2: Cloud architectural types and interfaces

Simply put, the issues, concerns and value of any given cloud service are best measured by the direct user of the service. There are different audiences, with different needs, depending upon where they focus within the overall cloud computing architectural stack.

One additional important point is that the value chain of a service is dependent upon all the other services upon which it relies. Thus, if a vendor creates a SaaS solution that depends on a PaaS offering, they are by proxy restricted by the limitations of that PaaS offering. As you can imagine, the linking of dependencies can quickly become complicated in a cloud service; thus it only takes one bad apple to spoil the entire barrel of cloud services. Conversely, improvements in the value of any function by proxy improve the value of all other features that depend upon it.

Clearly, as the momentum around cloud computing continues to transform the industry the benefits of *increasing* value will continue to grow exponentially.

2.5 Data Formats

It is important to consider the role of data formats in this discussion. In addition to interfaces, application programs and software packages acquire, store, and process data using structures which are optimized by the software developer. Following the principle that compatible interfaces are important, in a cloud environment, two implementations of the same service do not have to be created in the same way and may store data very differently. They may well also store derived and implementation specific data differently. Without proper definitions of import and export formats, which you might well expect to be behind an interface, a set of data from one implementation will probably be meaningless when imported into another implementation. This problem clearly has an impact on interoperability and data portability between clouds. If we want to avoid complete lock-in of data at this level, there needs to be agreement on interchange data formats. This is not a straightforward task. There are multiple different application areas for software. Therefore data interchange formats will need to be defined in a manner appropriate for each specific market such as for office suites, finance, healthcare, etc. This issue is of particular concern at the SaaS level.

3 Different Issues with the three types of cloud service

3.1 Issues at IaaS level

Interfaces at the IaaS level are defined very low in the architectural stack. At its core, the IaaS model presents a virtual hardware machine interface and a management interface. In order to be compatible with existing products, most IaaS platforms support a virtual PC interface which supports operating systems that run on off-the-shelf hardware. Although most virtualization platforms operate from this conceptual model, there are several different technologies on the market. Some of these are open and others are proprietary, while some are a combination of the two. It is important to note that regardless of the IaaS cloud vendor, the workloads that run on top of their platform may still be proprietary or open in nature.

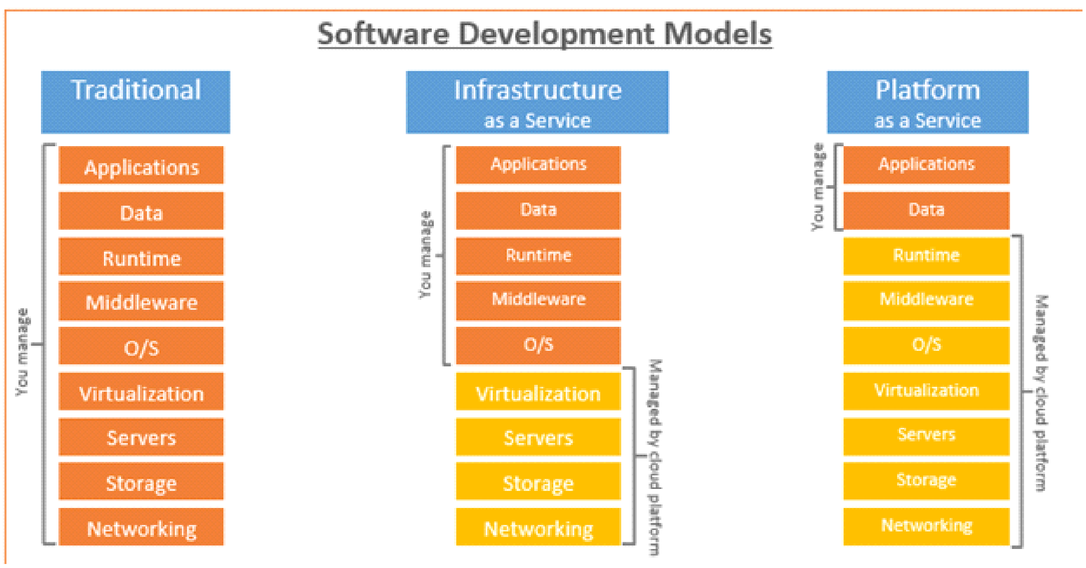

Figure 3: Software stacks and management

The most important interface provided by the IaaS model is the IaaS management interface. This is the mechanism that controls how cloud computing implementations are created, deployed, initiated and stopped. In this way, IaaS can be seen as a layer that can be added to any existing or legacy (non-cloud) software application, such as an existing software package that currently runs on a dedicated server inside a corporate network. To date, IaaS has been the most popular cloud computing model in the market with one player occupying the leading position in terms of market share. Platforms may use an open virtual machine; however, the *management interface* – perhaps the most critical feature -- may still be proprietary. As a result, the majority of cloud services today are driven by a proprietary platform. An investment in using those interfaces may be sacrificed when an application is moved to a different IaaS vendor.

There are open alternatives. Open Source efforts have gained significant momentum in developing a cloud computing platform over the past few years, for example, OpenStack. It is an IaaS platform that has an open interface for management and many large vendors are rapidly enhancing their offerings based on OpenStack.

Due to their different histories and goals, different IaaS platforms can have dramatically different interfaces. Existing workloads from one cloud platform can communicate and inter-operate with services running on another, but the ability to *migrate* such workloads between cloud platforms can be very limited. Open Source projects, such as OpenStack, hold great promise at improving this situation, since they provide open source solutions for the management of cloud service workloads.

3.2 Issues at PaaS level

Although the PaaS layer likely has the least amount of hype at the moment, it has strong potential to lock users in to to proprietary platforms. The interface between software applications and the platform they run on is known as the Application Programming Interface (API). It is the mechanism by which an application actually uses the features and services provided by the platform. For example, a software application that wishes to save some information in a document uses the API provided by the platform to request that the information be saved to a disk drive. Modern operating systems provide thousands of functions to software applications via their API, and most operating systems- especially proprietary ones- have different APIs.

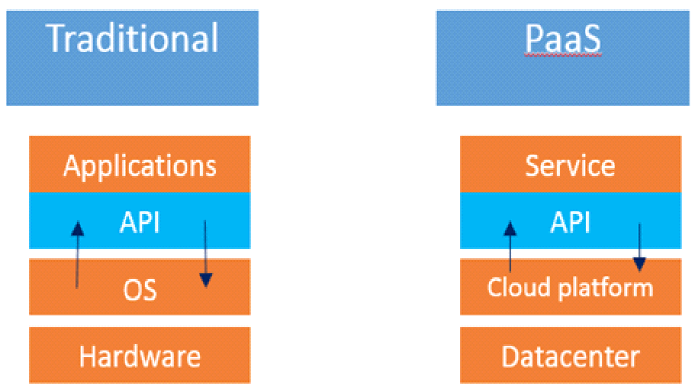

Figure 4: Traditional application architecture and PaaS

Much in the way that software running on a specific operating system use that operating system via the interface to its functionality, cloud services running on a PaaS platform are also connected to the underlying platform via its interfaces. If the PaaS platform is proprietary, services running on top of it will run exclusively (have high dependencies) on that platform. If the PaaS platform is open and offered by multiple vendors, customers that run cloud services on that platform can move their services across cloud platforms providing the same PaaS interfaces. This provides a very similar choice that has previously been provided by open middleware platforms. Conversely, as is the case with IaaS investments made in building software using any proprietary PaaS platform interfaces will be sacrificed when you need to move your applications to a different PaaS platform vendor.

Put another way, for a customer looking to build its own cloud, or use a third party's offering, the issue is whether a PaaS allows you to write applications and move them to another platform, including back to a more traditional platform. Now, of course, in the latter, the more open that additional functionality is, the less lock-in there will be. However, in the case of OpenShift (or CloudFoundry), this PaaS solution uses standard languages, frameworks, and tools and provides a platform on which and through which they can be used.

Management in a PaaS system is mostly concerned with the deployment, configuration and operation of applications, as opposed to the management of underlying compute and storage resources at the IaaS level. Applications that rely on proprietary packaging, deployment and run-time management interfaces will be restricted in their ability to migrate to other PaaS systems; fortunately work is underway in both standards organizations and in open source communities to define open application management.

3.3 Issues at SaaS level

The largest issue for SaaS workloads is likely based on the formats of documents, data files and protocols used by the services being provided. Since the SaaS model is more focused on individual end-user needs, there is already a huge variety of SaaS offerings in the market; each focused on a specific customer base and specific functionality. However, some of these offerings are more 'general' in nature: for example office productivity application services. Most traditional businesses have relied on well-known proprietary, productivity suite software to handle these needs, but the future is squarely aimed at the cloud. Leading vendors have shifted their marketing strategy away from their traditional disk-based software offering and towards their new SaaS version. Still however, proprietary formats are common. A service may be accessed and used by a wide variety of clients, including mobile phones, Linux desktops and even Apple iPads. However the information created and consumed by those services can still be limited to a single vendor if a proprietary document format is used. Further, interfaces to the functionality may be proprietary, and thus any solution that is built to leverage the functionality provided cannot be easily migrated to a competitive cloud service offering.

This illustrates a significant potential pitfall for considering SaaS offerings: although customers might be able to access and *use* the services from a variety of clients, the ability to move seamlessly from one vendor to another may be difficult because of other dependencies such as different data formats. If a cloud service is limited by *either* accessibility or by data compatibility, its value is reduced.

SaaS service	Data format	Protocols	Open/Proprietary	Comments
Email	Messages, MIME	SMTP, IMAP, POP	Mostly open	Some email systems have proprietary formats and protocols, but supports interoperability with open standards
Office Productivity	Specific proprietary and open document formats	Web, XML, Proprietary	Mixed	Supports open clients, open and proprietary documents

Calendar	iCalendar	CalDAV	Open	Supported by most modern clients and collaboration services

Table 4: SaaS examples

4 Conclusions

Cloud computing is evolving rapidly to meet the extremely wide variety of today's market demands. Customers can benefit from significant advances in agility, flexibility and cost efficiencies by adopting cloud computing solutions. However, there are also significant issues to consider when making a choice about a cloud model and implementation. Without meaningful, accurate and up-to-date information, enterprises are likely to replicate the same vendor lock-in that characterizes traditional IT investments. Education is very important because there are many aspects of cloud computing solutions that need to be given adequate consideration in order to fully understand the potential advantages - and consequences - of choosing a specific platform or solution. With the increasing complexity and pace of technological evolution, this task may seem quite difficult. However, the ability to rapidly experiment with cloud-based technologies can lead to better decision-making and encourage business model evolution.

By evaluating a cloud offering from the perspective of a variety of stakeholder positions, appropriate interfaces and data formats can be assessed to help evaluate their value and its potential for lock-in.
By understanding what *choice* those interfaces and formats enable- and by extension the level of compatibility with alternate offerings - the potential for a thriving and competitive market can be better understood[2]. And these choices hold, whether one is looking at a private, hybrid or public cloud approach.

As the global economic landscape continues to change, enterprises must be able to react more quickly in order to remain competitive in the global market. Many enterprises will need to be able to easily track and migrate information and functionality based on the ever-changing business climate. Reliance on a single proprietary vendor and their technology infrastructure may not be able to support these abilities.

When considering any decision involving the use of a cloud-computing service it is important to understand the answer to the questions: "how easily can I move my data and functionality to another location or platform should I need to do so - for any reason?" This includes focus on an effective **exit and migration strategy.**

[2]Although the shift from software to services enables increased agility and flexibility in buying, building and deploying solutions, it has introduced challenges since individual users and enterprises expect to be able to bring-your-own-device, including laptops, tablets, and smart phones, and use it with their cloud services. Because cloud computing is a worldwide phenomenon businesses and consumers also expect seamless high performance access to their cloud services at work, home, on the road, even internationally. There are boundaries that businesses have to deal with – leaving from behind the secure corporate firewall – as well as crossing country boundaries – that change the information available to the consumer and how it is delivered in a secure manner, if at all.