The Oil and Gas Company in a Box

Planning a Future in the Cloud

Oil and Gas Company in a Box
14 February 22

EnergySys Limited
Hudson House, 8 Albany Street
Edinburgh EH1 3QB

t: +44 1224 433 493
e: sales@energysys.com
www.energysys.com

Table of Contents

1 Introduction

1.1 Overview

In our accompanying paper (Black, 2017), we discussed the historic precedents for the migration to cloud, why it is happening now, and why it is essential for every oil and gas company to embrace it. In this paper, we provide guidelines for adopting the cloud, and present our Cloud Checklist, explaining the four essential things that you must demand of every potential cloud provider.

We will discuss the challenges facing the developers of software for oil and gas, and the difficulties in establishing a viable economic model that delivers economies of scale that benefit customer and vendor. We will provide the questions that a company must ask of itself, and its potential vendors, around adoption, security, resilience, and data management.

1.2 An Extinction Event

Even though the advent of cloud represents a potential extinction event for many companies, there is institutional resistance to change. There is a lack of understanding of the technologies underpinning cloud and how traditional applications can be delivered in this modern environment.

This resistance is not limited to users, who struggle to understand the best model for adoption, but extends to vendors, many of whom offer solutions that cannot easily be migrated to the cloud. They engage in widespread “cloud washing”: simply adding the word “cloud” to product names as an attempt to generate market interest. They promote private cloud and hybrid cloud models, without any clear or consistent definition of these terms.

The pace of innovation in oil and gas software has stalled, and low-cost, widely available consumer software vastly exceeds the quality and usability of most traditional business applications. Data managers are frequently engaged in activities whose sole purpose is to remedy the deficiencies of traditional applications, identifying and cataloguing data carelessly left behind by incontinent software. This must, and will, change. The cloud revolution has already started.

1.3 Defining Cloud

For people struggling with the nature of cloud, it is often easiest for them to try to think of it in terms of technology. They think about servers, and storage, and networks, and virtualisation, and try to define the essential technological distinctiveness about cloud. This is a mistake. Cloud is not, in and of itself, about technology. It is fundamentally about a different approach to the consumption of compute services; one in which services are provided on a utility basis. Compare it with electricity generation, where the details of production are unimportant and we only measure reliability and cost per unit.

In their 2011 paper, members of the US National Institute of Standards and Technology (Mell, et al., 2011) provided their definition of cloud computing. Their five essential characteristics of cloud are not about technology, but about the nature of the service.

- *On-demand, self-service*. This means that consumers (or end users) can provision computing capabilities, like server time and storage or features, without needing to contact a service provider. In other words, the power is in the hands of the user and doesn't mandate third-party involvement.
- *Broad network access*. This characteristic implies that the access to services is over standard mechanisms, and that it doesn't require specialist or proprietary technology. You can use laptops, or phones or whatever device is appropriate for your work.
- *Resource pooling*. The resources, like compute power or network storage, are shared among multiple users, or tenants. There is no requirement to understand how this is achieved, or to know the location of the compute resources providing the service. Obviously, security and governance concerns must be addressed.
- *Measured service*. Usage of compute resources is measured and controlled to benefit the provider and the consumer. There is transparency, and users generally pay only for their metered use of resources. Charging can reflect demand, thereby controlling consumption and increasing efficiency.
- *Rapid elasticity*. This is one of the most important characteristics. The resources that I provision as a user, in a self-service model, can be scaled up or down to meet demand in an apparently unlimited fashion. In accordance with the measured service characteristic, I will only be charged for the resources I use.

While the NIST paper does define a private cloud deployment model, it is important to note that very few organisations will have the demand or the ability to deliver a service to their internal users with the characteristics above. Resource pooling and multi-tenancy are essential in delivering elasticity in an economic fashion, and this demands a scale of operation that few companies possess. Further, significant effort is required to maintain the infrastructure to offer metered usage of resources that are dynamically provisioned by users in a self-service model. For most companies, IT provision is a necessity, not a strategic investment. Public cloud should almost certainly be the first choice for every oil and gas company.

An interesting corollary is that true cloud applications must be built to support a multi-tenancy, self-service, metered model, or the economies of scale are lost. If this is not the case, "moving applications to the cloud" simply becomes an exercise in shifting servers from one data centre to another. As we'll discuss later, this will require fundamental changes to the nature of software

development for oil and gas software. More broadly, we'll explain why much software produced for oil and gas is mediocre, and why it lags at least ten years behind comparable industries.

2 Cloud Software is Better Software

2.1 A Tradition of Mediocrity

Most users would struggle to name an oil and gas software package that they use in-house that they love. They would find it equally hard to identify one that delivers an attractive, professionally designed, user interface. They would struggle to describe a positive experience with upgrades and maintenance, and many would curse both the cost and quality of support.

The truth is, the software products industry for oil and gas is, for the most part, broken. It compares desperately badly with the quality and care associated with many consumer-oriented smartphone apps, where people have taken the time and trouble to think carefully about the way the product is used. Over the years, users in oil and gas have learned to accept what they are given. They ignore the faulty bits and the clunky user interfaces, and work around issues to get the job done.

Worse than this, users have endured, but also encouraged, a culture of custom software development. The mistaken belief that a problem is unique or that a solution will provide competitive advantage is used to justify the investment of huge sums of money for programmers to develop substandard software. What's more, these sums are dramatically underestimated because people rarely calculate the true total cost of ownership (TCO), a topic to which we will return later. Even some so-called "product" software requires programmers to tailor it for an individual customer's requirements, which means that upgrades are a major problem.

Software development is hard. You need time and incremental updates to achieve a high-quality deliverable that does what users want. You need a sizeable base of users that will provide feedback and revenues sufficient to justify the on-going investment in a product development and support team. That user base must be using the same product, or the support and development costs escalate and there are no economies of scale. Upgrades to the product must be regular, easy to deploy, and adopted readily.

In contrast, many software packages in oil and gas have extremely limited user numbers. They are complicated to install and frequently unreliable. Support costs are borne by the small number of users, so they are significant. Any development is typically done on demand and paid for by the current customers or customer. It is painful to deploy new versions, so little is done unless it is absolutely necessary.

In summary, then, there are three problems. First, there is insufficient focus on the creation of true products that are architected to ensure the maximum user base possible. Second, the on-premises distribution model is a truly dreadful one when user numbers are small, particularly for complex technical products. Finally, the sclerotic pace of development is worsened by the habit of demanding payment for any new features that customers request.

The result is high licensing costs and high support costs, combined with disappointing quality and limited feature sets.

2.2 Cloud Economics

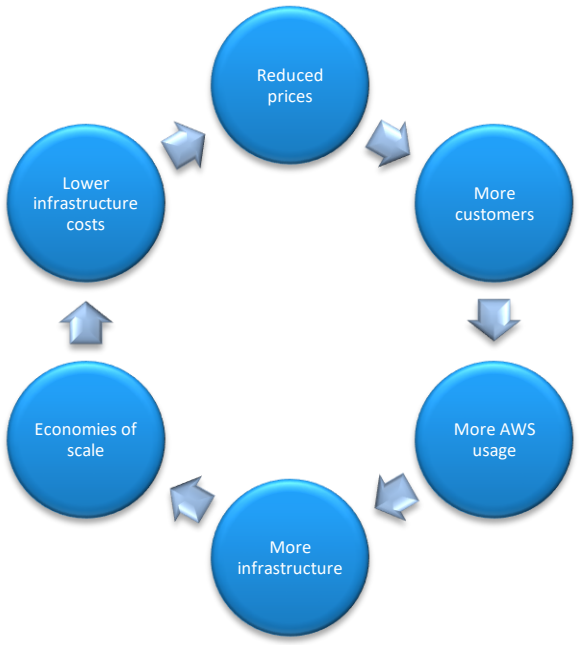
Cloud software addresses many of the core problems with the economics of on-premises solutions that were described in the previous section. Even with a smaller user community, it is

significantly easier to develop, deliver and support a limited number of cloud instances of software. There are no release processes or installation scripts, and no need to deal with procurement and configuration of hardware and software for every customer. Even with virtualised on-premises environments, it is still not straightforward to distribute software efficiently and ensure that it delivers optimum performance. In the cloud, when we implement encryption of data and failover and backup and enhanced security, everyone benefits at once. It's not difficult to add resources, like CPU or storage, when customer demand peaks.

Most of this is simply the real and obvious practical benefits of cloud production and supply, both for developer and customer. However, there are also differences in the way that software is designed and developed for cloud that reflects a different philosophy, as captured in the NIST characteristics.

Building a multi-tenant service that is configurable by the end user implies a dramatic reduction in complexity. All customers are on the same software platform, and new releases are delivered to every client at the same time. This dramatically simplifies support and development and allows real economies of scale. Shared costs provide the opportunity to fix or even reduce costs year on year, so that every customer benefits from growth in product use.

If the design of software is such that much change can be accomplished by the users themselves, then the need for costly development and releases can be avoided. Clearly, there are limits to this, and the core product will require extension and new features to deliver functionality not in the control of the user. However, the development and delivery model for cloud allows an increased cadence of releases. In short, cloud delivery allows the development of software at a higher quality and lower cost, with more features over time.



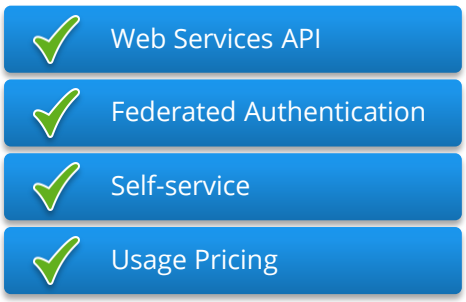
This philosophical difference is one of the clear reasons why simply migrating server workload to a third-party data centre is not cloud. There may be real benefits from using cloud infrastructure, but the true win will only come if the software is built for cloud.

All the above is neatly captured in the virtuous cycle model promoted by Amazon Web Services (AWS) for its cloud infrastructure, as shown in the diagram above.

3 The Cloud Checklist

3.1 Four Essentials

We believe there is a need for a "cloud checklist" that represents the minimum set of standards that should be applied when selecting cloud providers. Note that this is not intended to be a comprehensive list of considerations. It's a core set that we consider essential to assure customers of minimum standards for security and interoperability, as shown the diagram.



We will discuss these four items in more detail in the following sections. However, note that it does not include a requirement for multi-tenancy, even though we highlighted its importance previously. Having an architecture that supports multi-tenancy allows the realisation of the benefits of the economies of scale that cloud promises. However, many legacy applications will require investment to make this transformation, and it does not imply that they cannot be useful in the meantime, so long as they adhere to the checklist.

3.2 Web Services API

The first part of this initial requirement is that the cloud application has an application programming interface (API). An API is more than simply an interface for exchanging data. It implies an ability to control the behaviour of an application. In many ways, it would be ideal if it were possible to do everything programmatically via the API that is possible in the application's user interface.

The second part is that it is a web service, which implies that it communicates using standard web protocols. Unfortunately, this does not fully describe the content or the protocol, and there are competing standards. As an example, XML-RPC uses XML as an encoding, HTTP as a transport mechanism, and a remote procedure call as a model. This mechanism evolved into the SOAP standard (Wikipedia, 2017), which attempted to provide a comprehensive definition of all components of the technology stack.

While SOAP might still have its uses, we argue that it should be avoided. SOAP layers an object RPC model on top of HTTP, creating a hybrid mess that adds little of value. You cannot use a SOAP interface without some form of manual, and every SOAP API will behave differently. It is also likely to be stateful, so not hugely scalable, and is particularly poor in a distributed environment. While an RPC model is acceptable for internal systems, where network performance and reliability are generally assured, it is less successful as a generalised mechanism for integrating cloud and on-premises systems.

Our preference is for a REST-based architecture (Fielding, 2000). REST (representational state transfer) describes the way the web works, or its architectural style, and is a term defined by Roy Fielding in his PhD dissertation from 2000. It is deceptively simple. There is no "out-of-band" information, or things you need to know or read before you get started. It is just hyperlinks and resources. The REST architecture describes the principles that makes the web work and have

made it possible to create the largest inter-connected network of information that the world has ever seen.

However, it is also rather complicated to build a high-quality REST service, particularly around the definition of resources. For that reason, we recommend the OData standard (OASIS) as a good basis for a REST service. Originally developed by SAP and Microsoft, it is now a full international standard that is embedded in many products, including the Microsoft suite of products. Use of OData enhances scalability and interoperability. Like the traditional web, there's no out-of-band information, and all a programmer needs is a starting web address.

It is important to recognise that much enterprise software, and software generally, is not written this way. Programmers require manuals that define the information needed to work with the software, and the functions or methods that we call. This is all "out-of-band" information and makes the task of linking disparate systems extremely difficult.

None of this implies that OData is beyond criticism, or that there are no other options, but it is likely to be substantially better than defining a new protocol or service from scratch.

3.3 Federated Authentication

Centralised management of users and single sign-on (SSO) is essential as cloud adoption increases.

Federated authentication means that you can designate a single directory to use for authentication and establish a trust relationship with any other directories in use, including those in the cloud. This provides a single point at which to establish password rules, and to add and remove users. This federation can bridge on-premises and cloud, so it is possible, for example, to use an on-premises Active Directory as the primary repository, and to federate with any cloud directories.

This is a clear benefit both for the management of users, and for user password management. In an ideal situation, a user will have a single password (and possibly an authentication token) that will allow them access to every system that an organisation uses, both on-premises and in the cloud. This should extend to VPN servers, where it is common to find organisations with different directories for VPN access.

3.4 Self-service

Self-service is one of the characteristics of the cloud identified by NIST (see section 1.3). However, they frame it in terms of provisioning of virtual servers and storage and so on. For cloud application users, this is better described as the ability to do all required configuration of the service without needing any help from the vendor or from centralised IT. From user provisioning to access controls, to application setup, to reporting, to the provision of a test service, and so on. In general, nothing should require the intervention of the vendor, and users should be able to make all changes themselves.

Most importantly, it must be possible for users to control the parameters of the cloud that determine the charges. So, if the metric is users, it must be possible to add and delete users at will. If the metric is storage, then the ability to centrally control storage for all users must be available.

3.5 Usage Pricing

The pricing model for cloud is fundamentally different from traditional on-premises solutions. It is not simply a matter of switching from up-front licensing costs, with annual support and maintenance, to a model of annual subscriptions. While this minimises up-front costs, it is not the real benefit of cloud. The requirement is to charge based on a metric that reflects real usage and value derived from the cloud system. If it is a cloud mail service, then users are an obvious metric. If it is human resources software, it might be employee numbers. For EnergySys, it is based on a forecast of annual production or number of assets.

It must also be possible to increase or decrease these usage metrics, ideally on a month-by-month basis. Thus, as a company grows or shrinks its use, the costs grow and shrink in line. In general, it is only straightforward to offer this type of pricing in the cloud.

4 Broader Considerations

4.1 Cloud vs On-premises

The checklist items we considered in the previous section are the minimum standard an organisation should apply when evaluating cloud solutions. However, when choosing and contracting for a cloud service, there are several other topics to consider. It is worth noting, that many of the following comments could apply equally to on-premises systems. Thinking holistically about disaster recovery, user education, compliance, security and so on, is essential. One survey of US companies (PwC, 2014) suggests that 28% of cybercrime incidents are initiated by insiders, and 32% of the survey's respondents felt that insider events were likely to be significantly more damaging.

4.2 Costs

The assessment of costs of a transition to the cloud is likely to invite a comparison of costs with on-premises solutions. If such a thing is to be done, and we will argue later that it misses the point, then it is essential to ensure that the comparison is of total cost of ownership.

For example, in many companies with organisation siloes, central services are charged as overhead, and the costs are hidden to a degree. Also, oil and gas companies are often poor at tracking the total cost of systems over time, with initial development or acquisition treated separately from the costs of ongoing maintenance. This masking of true costs can lead to mistaken judgements and poor choices. Finally, time spent by staff in managing and maintaining internal applications must be counted, including that invested by end users. Some of this may be time spent in working around deficiencies in a traditional solution.

Notwithstanding the above, costs are important, but not the key factor in moving to the cloud. The true goal is transformation of the business. To be able to realise the benefits afforded by focus. To be able to respond flexibly and quickly to change. To be able to increase and decrease costs as circumstances dictate. To build a truly agile business.

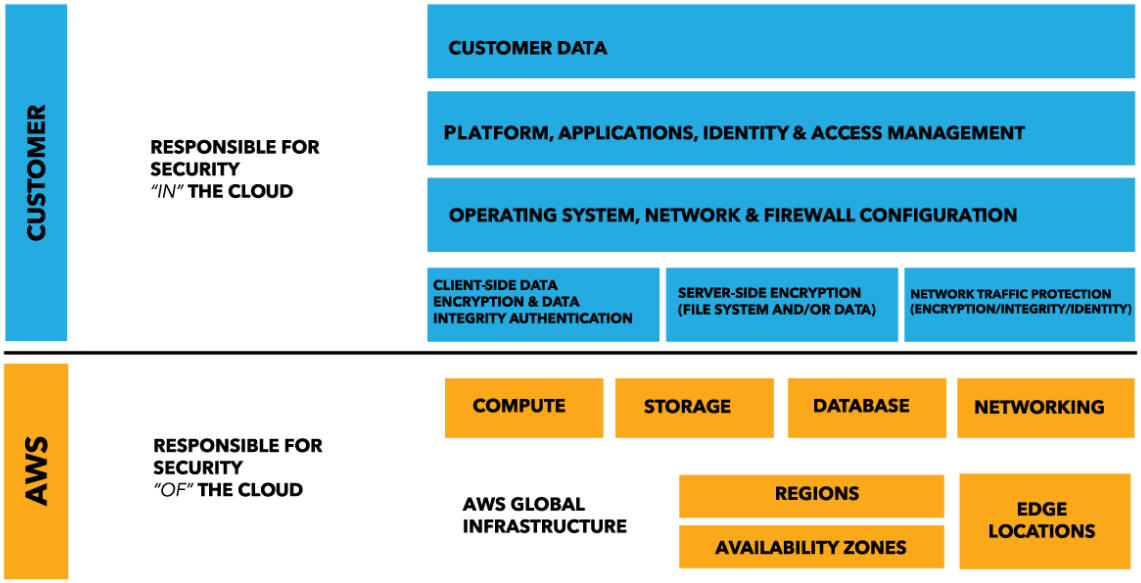
4.3 Security

Security is frequently cited as one of the top concerns of those considering cloud solutions (IDG Enterprise, 2016). However, the reality is that cloud providers have made security their top priority, and they are well-placed to attract the highest quality talent to support their goals. In comparison with many on-premises servers and networks, the technology capability is immense, and the pace of development is staggering. They provide secure role-based access that assists with least privilege models of application management. Easy access to features like disk encryption or network access logging make it practical to deliver extremely secure services. Clearly, such things are possible on-premises, but the levels of effort and expertise required would demand very large investments of the kind not typically seen in even the largest organisations.

None of this eliminates the requirement for users to manage and enforce security standards and procedures. However, it does make this job significantly easier as many aspects of security are the responsibility of the cloud provider. It is important to understand where responsibility for

security lies for each element of the service. If you provision standard applications in the Microsoft Office365 suite, then Microsoft is responsible for the full stack, from infrastructure to application. However, this is unlikely to be the case for many cloud application providers, most of whom will host on one of the major infrastructure providers like Microsoft Azure or AWS.

In this case, the responsibility for security is shared. The diagram below illustrates this, and is taken from the AWS web site (https://aws.amazon.com/compliance/shared-responsibility-model/).



This distinguishes between the "security of the cloud", which is the infrastructure that AWS provides, and the "security in the cloud", which is the security that is the responsibility of the cloud application provider. An almost identical picture would be applicable for Microsoft Azure or Google Cloud. If a vendor is running on AWS or Azure, it does not mitigate the need to ask questions about security and privacy procedures of the application itself.

With this in mind, and in addition to the checklist in Section 3, we suggest the following actions:

Develop a security policy that defines priorities and objectives for all aspects of the company's data, both cloud and on-premises. The latter is important, as the general belief in, and reliance on, physical security has generated a measure of complacency. In fact, with few exceptions, it is almost impossible to prevent a determined individual from bypassing all security if they can obtain physical access to a server. Despite this, many companies regard their locked front door as the only barrier required.

Educate users on the policy. Help them understand the goals, and how they can make changes to achieve compliance.

The security policy should apply equally to APIs. For application-to-application integration, PKI certificates should be considered mandatory, to avoid issues with password management.

The policy should mandate the universal encryption of data at rest. If the data is particularly sensitive it is possible, at a cost, to have a device on-premises that stores the encryption keys used by the cloud applications, making it impossible for the cloud service provider to decrypt the data.

The policy should specify a least privilege model, in which each user has a role that strictly defines the extent of their permission set. This role should ideally not be too generic or unspecific, such as "Administrator", as this will lead to "permissions leaks". Rather, it should suggest the business functions that the user can carry-out.

The policy should specify that Federated Security should extend to all cloud applications, and to on-premises systems too. The advantages of SSO and centralised management must extend across the enterprise, and not just be afforded to users of cloud software.

A least privilege model does not remove the need to have individuals with higher administrative privileges, but it becomes possible to minimise the need for such powers. To ensure that higher-level privileges are used appropriately, the policy should ensure that a complete audit log of every action on every service is maintained and monitored. Alarms should be set up to flag suspicious activity.

Application providers should confirm that they have a regular schedule of penetration testing. It might also be reasonable for potential users of the service to execute their own tests, with the vendor's permission.

The policy should specify that multi-factor authentication (MFA) is mandatory for every application. This move dramatically improves security, with only minor inconvenience, and remedies many of the failures of password usage. The classic paradigm for authentication assumes authentication based on something you know (a password, for example), or something you have (like a secure ID card), or something you are (like your fingerprint). MFA enforces the use of at least two of these.

In preparing policies, it is worth reviewing the NIST guidelines on Digital Identity (National Institute of Standards and Technology, 2017), particularly 800-63B on Authentication and Lifecycle management.

It has traditionally been common in oil and gas to ask for evidence of quality standards, and adherence or certification to an international standard like the ISO 9000 series. ISO/IEC 27001:2013 is the equivalent standard for information security, and companies like Amazon offer their certificate as a download (Amazon, 2016). The standard is not prescriptive and is based on a demonstration that a management and monitoring system is in place, based on a risk assessment and appropriate controls. While it is worthwhile investigating whether a cloud provider complies with or is certified to ISO 27001, it is important to be aware of the hierarchy of responsibilities discussed above, and the scope of the certification.

4.4 Contracts and Terms

4.4.1 Service Levels

Service levels are a key component of any agreement, covering not only the cloud service but the associated support. Again, this is not dissimilar from an on-premises solution from a traditional provider.

Clarity is essential. Customers need to understand the duration of potential outage that is implied by the percentage uptime target in any service level agreement (SLA). Where scheduled maintenance is excluded, understanding the timing and duration of outages in a year is also important. If the support hours are restricted, what time zones do they cover? Is the infrastructure support different from the application support?

It is not uncommon for agreements to include a measure of reparation, generally in credits for subscriptions, if agreed targets are not met. This is not particularly useful, as the goal is to ensure that agreed service levels are achieved, and not to make money from a vendor's failure to hit the targets. As with many other issues, the correct response is to have contingency plans in exactly the way you would with on-premises systems. As an example, can processes be run manually for a few days?

As with all legal matters, the intent must be clear. Contract terms and conditions are not the place to exact revenge.

4.4.2 Data Ownership

The legal agreement with a cloud provider must be explicit about ownership of data. It may seem obvious, but all customer data should remain the property of the customer, including any generated data. It should not be used by the provider in any way and, indeed, it should not be viewed by the provider unless explicit permission has been given. This applies equally to any custom code developed to run on a platform, and it is necessary to clearly distinguish provider IP and customer IP.

One of the many benefits to cloud is the promise of "easy entry, easy exit". The ease of that exit would be substantially reduced if data is not returned promptly, or if significant exit fees are applied. The agreement should be clear on ownership, and that data will be returned on termination, with any associated charges and timescales.

4.4.3 Data Locality

Data location is another issue that arises frequently in oil and gas. Most of the concerns in the EU and US are about personal data, and its security and an individual's privacy, but the concern in oil and gas is less well-defined. Indeed, it is sometimes difficult to identify any regulations controlling data location, and rules appear to be enforced and repeated based on habit not regulation. While some countries do enforce the location of data within their borders, it is often not clear if this is a concern over data loss, or to avoid outsourcing of skilled work to overseas teams. Nonetheless, there is increasing opportunity to ensure that data is stored and processes in a region of choice: this is not always a specific country, although it could be. For example, specifying that processing must occur in the US or UK, or EU is certainly possible.

4.4.4 Provider Failure

If the vendor is established and secure, the potential for business failure may not be a significant concern. However, due diligence is required to ensure that the contracting entities are clear, and that the provider is commercially viable. It makes sense to ensure that the provider has some mechanism that allows regular export of data for backup purposes. In extreme cases, it might be worth negotiating an escrow agreement that allows access to the software in the event of a company failing. However, this must go beyond a simple store of source code. It must be a service that ensures the provided code builds and that the software will work, and this must be done with every release. This is not a trivial cost, and providers will likely pass the costs to customers if they insist on it. It should be considered only if the risks are substantial.

Finally, there may be multiple parties involved in delivering the service, and while your contract is with the main provider, it is worth establishing if there are any obligations with third-party providers that might impact your service.

4.4.5 Liabilities

In the event of data loss or corruption, or a service failure, it is important to be clear where liability lies. Most contracts will exclude consequential damages and limit the liability to the total contract value. Further, given the shared responsibility model described above it must be clear where the responsibility lies, and who will be liable in the event of failure.

Unfortunately, if a company has lost all its data and is now struggling to operate, the compensation available is unlikely to be of consequence. In truth, therefore, the mitigation is not legal, unless negligence can be demonstrated. Instead, the same processes and procedures that are applied for on-premises systems should be in place. If a key staff member has permissions sufficient to delete all data and backups, then it is sensible to have a secondary backup in a secured location that is inaccessible to that individual or group. This is true for both cloud and on-premises, if the data is mission critical. Adequate risk assessments, with appropriate mitigation, are much more effective than legal recourse.

5 Envisioning the Future

5.1 The Oil and Gas Company in a Box

For many years, we have had a singular vision: The Oil and Gas Company in a Box. This is not an application platform from a single vendor. It is a vision of a patchwork of cloud applications that can be added as required by an oil and gas company. At the start, it might be office productivity applications, document management, finance and reserves management. To that we might add reservoir simulation and seismic interpretation. There is no need to invest in on-premises databases or software licenses, and costs are tied to use.

As the company evolves, we can add applications for drilling, and later for production data management. We might also have metering systems or historians. We can augment the core with analytics platforms to interpret and report data, and CRM software for marketing and management of partners. Asset tracking allows us to maintain a record of all owned assets, and their location, to support current operations and future decommissioning.

All these applications would obey the cloud checklist, which renders it unnecessary to consider details of how the data is stored, or which cloud infrastructure provider is in use. Accessing the web services API will be identical across different applications, different providers, and different platforms. This is particularly true if OData is used. It eliminates the requirement for a data warehouse, as all data can be accessed from its native repository, using web services to navigate from platform to platform.

This is a compelling vision, but, in truth, there is a shortage of quality cloud applications for oil and gas, driven by a lack of demand, that creates a vicious circle. However, many services do exist, and more would appear if the industry demanded it.

5.2 Making the Vision Reality

A significant, and increasing, percentage of North Sea production data is now stored in the EnergySys Cloud. This is a dramatic change from ten years ago, when it was routine to be told that “no oil and gas company would ever store its data in the cloud”. What is more, they are doing it at a cost that is a fraction of a traditional system. Every aspect of hydrocarbon accounting and production reporting can now be done in the cloud.

However, this is not the only service available to companies who want to take their first steps with cloud application. They can choose from cloud applications for:

- Finance
- CRM
- Helpdesk
- Action tracking
- Metering uncertainty
- Metering historians
- Production accounting
- Environmental reporting

- Reserves management
- Document management
- Personnel tracking
- Asset management
- Environmental reporting
- Workflow

Ultimately, the transformation in IT that we envisage is captured by our three tenets of cloud computing.

- Pay for service, not software or hardware.
- Pay for value delivered, not value promised.
- Be in control, not without control.

These tenets are not about slightly better technology, or about different deployment models, but about an entire new way of buying computing. It is fundamentally about service, not software. They define what we think is different, and better, about the cloud model of computing.

The Oil and Gas Company in a Box is becoming more real every day.

6 Bibliography

Arkes Hal R and Blumer Catherine The Psychology of Sunk Cost [Article] // Organizational Behavior and Human Decision Processes. - [s.l.] : Academic Press, Inc., 1985. - Vol. 35. - pp. 124-140.

Black Benjamin EC2 Origins [Online] // Benjamin Black. - 25 January 2009. - http://blog.b3k.us/2009/01/25/ec2-origins.html.

Black Peter The End of Corporate Computing: Revisited [Report]. - 2017.

Carr Nicholas G IT Doesn't Matter [Online] // Harvard Business Review. - May 2003. - https://hbr.org/2003/05/it-doesnt-matter.

Carr Nicholas G The End of Corporate Computing [Online] // MITSloan Management Review. - MITSloan Management Review, 15 April 2005. - http://sloanreview.mit.edu/article/the-end-of-corporate-computing/.

Edstrom Dave MTConnect: To Measure is to Know [Book]. - Ashburn : Virtual Photons Electrons, LLC, 2013.

Kahneman Daniel Thinking Fast and Slow [Book]. - [s.l.] : Farrar, Straus and Giroux, 2011.

Mell Peter and Grance Timothy The NIST Definition of Cloud Computing [Report] / National Institute of Standards and Technology, US Department of Commerce. - 2011. - NIST Special Publication 800-145.

Seely Sam Sam Seely [Online] // The Amazon Flywheel: Part 1. - 2 May 2016. - http://www.samseely.com/blog/2016/5/2/the-amazon-flywheel-part-1.

Snyder Neil H, Dowd Jr, James J and Houghton Dianne Morse Vision, Values, and Courage: Leadership for Quality Management [Book]. - [s.l.] : The Free Press, Simon & Schuster, 1994.

Tykocinski Orit E, Pittman Thane S and Tuttle Erin E Inaction Inertia: Foregoing Future Benefits as a Result of an Initial Failure to Act [Journal] // Journal of Personality and Social Psychology. - [s.l.] : American Psychological Association, Inc., 1995. - 5 : Vol. 68. - pp. 793-803.