Cloud Providers Viability: How to Address it from an IT and Legal Perspective?

Cesare Bartolini[1(✉)], Donia El Kateb[1], Yves Le Traon[1], and David Hagen[2]

[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT),
Université du Luxembourg, Luxembourg, Luxembourg
{cesare.bartolini,donia.elkateb,yves.letraon}@uni.lu
[2] Commission de Surveillance du Secteur Financier (CSSF),
Luxembourg, Luxembourg
david.hagen@cssf.lu

Abstract. A major part of the commercial Internet is moving towards a cloud paradigm. This phenomenon has a drastic impact on the organizational structures of enterprises and introduces new challenges that must be properly addressed to avoid major setbacks. One such challenge is that of cloud provider viability, that is, the reasonable certainty that the Cloud Service Provider (CSP) will not go out of business, either by filing for bankruptcy or by simply shutting down operations, thus leaving its customers stranded without an infrastructure and, depending on the type of cloud service used, even without their applications or data. This article attempts to address the issue of cloud provider viability, proposing some ways of mitigating the problem both from a technical and from a legal perspective.

Keywords: Cloud computing · Provider bankruptcy · Law

1 Introduction

In November 2013, an alarm rang. Nirvanix, a major provider of cloud storage services operating in California, closed down its business with very little anticipation to allow its customers to recover their cloud-stored assets. This sudden event wroke havoc in the cloud community, and revealed a significant weakness of the cloud-based business model: once an enterprise outsources services and data to a cloud provider, it is no longer in control of them, and can suffer from adverse conditions occurring to the cloud provider.

While this new business model is definitely the direction where the market is heading, it introduces new and significant challenges. While some of them are simply a new way of addressing well-known issues, such as performace issues, resource availability, service security and reliability, others are strictly related to the outsourcing of parts of the business to the Cloud Service Provider (CSP).

The risk of the CSP suddenly going out of business, either through a "soft" cessation (i.e., filing for bankruptcy), or in a more abrupt way (simply ceasing the

J. Altmann et al. (Eds.): GECON 2015, LNCS 9512, pp. 281–295, 2016.
DOI: 10.1007/978-3-319-43177-2_19

operations and removing the assets) concerns the uncertainty about the stability of the CSP. If the CSP faces a bad financial situation, it might decide to go out of business, and this could happen more or less abruptly. For example, the CSP might file for bankruptcy, leaving its customers to entertain their business with the trustee instead of the regular CSP management board. Even worse, the CSP might not go through the legal shutdown, and instead simply cease to do business, shutting down all operations and leaving its customers stranded and unable to use their required infrastructure anymore. And all this might occur without any forewarning to the CSP's customers.

This is the problem of the so-called *long-term viability* of CSPs, one of the main factors to take into account when moving to a cloud platform. Normally, the migration to a cloud is a one-way door. Unless there is a major change in the size of the enterprise's business (in which case there might be pressing needs to adopt an in-house architecture), the choice to rely on a CSP to run its business, both as a starting choice or after migrating pre-existing applications, is a long-term one. The relationship between the client enterprise and the CSP is likely to last until either one goes out of business. But there is a lot of difference between the client enterprise and the CSP going out of business. In the former case, the CSP would lose a customer, which might have serious repercussions from an economic point of view, but this fact does not *per se* hinder the CSP's operations. On the other side, the CSP going out of business would seriously hamper the client's operations, regardless of the client's business status. Even if the enterprise's activity were solid and growing, the CSP's end of business would block all of its IT activities, because the underlying infrastructure would not be accessible anymore.

In fact, the CSP suddenly going out of business can be a serious problem for enterprises relying on it. Of course, the client can always switch to another CSP or set up an in-house architecture to solve the problem. However, this might take some time, and a prolonged inactivity might seriously damage the client, in terms of loss of income, customers, and reputation. This might be fatal to small enterprises. Additionally, the CSP might also be used by the client as a storage resource, meaning that it might have a large amount of data used by the client, and its loss (or unavailability for a prolonged period) might be a significant damage itself.

This paper addresses the issue of CSP long-term viability, by evaluating what solutions (mostly not specifically designed for viability) the CSP and the client can adopt to avoid the consequences of the CSP suddenly going out of business, and what further problems emerge from these solutions. We discuss the issue from two perspectives, namely an IT and a legal point of view, and propose a list of technical and legal measures to mitigate the problem. In particular, Sect. 2 delves into the concept of cloud computing and illustrates the problem with respect to the various types of cloud; in that section, some real-world scenarios in which this problem was manifest are summarized. Section 3 is a survey of literature on cloud viability and some approaches that could partially address the problem. Section 4 tries to build on existing research to propose solutions or mitigators based on a technical (Subsect. 4.1) or legal (Subsect. 4.2) approaches, or mixed solutions (Subsect. 4.3). Then, Sect. 5 summarizes and evaluates the solutions in a preliminary cost analysis. Finally, Sect. 6 tries to summarize the findings and suggests future research directions.

2 Problem Statement

The impact of the problem of long-term viability increases with the pervasiveness of the cloud and depends on cloud layers. In what follows, we explore the problem of cloud providers viability with respect to the cloud delivery models and we highlight some real-life scenarios that illustrate the impact of providers going out of business.

2.1 Providers Viability and Cloud Delivery Models

Depending on how much has been outsourced, the effect of the CSP going out of business can be much different [6]. When dealing with IaaS or PaaS CSPs, the main assets (application source code and data) are under the control of the customer, who can migrate them (with different technical difficulties between IaaS and PaaS); in a SaaS environment, on the other hand, the customer normally has no access to the application source code or the datacentre where its data are stored, plus it might also lack the technical skills to perform a migration. In particular, with respect to the most common types of cloud models:

- The loss of IaaS platforms[1] would require the client to find or set up a suitable hardware architecture;
- PaaS clouds would require it to also set up the basic software platform for running its applications. This configuration is very common for those enterprises which offer web-based services to their end customers. In general, the customers of PaaS clouds[2] are application developers who offer their applications through the cloud;
- Finally, in the SaaS paradigm, the client (the end user of the IT service) normally only uses a web browser (thin client) to access the service, so a CSP going out of business completely cuts its clients off their IT applications and data.

The problem might depend not only on the CSP but also on other third-party partners going out of business in case the CSP has in turn outsourced some of its services [6], with a potential cascading effect.

Long-term viability is one of the major risks that must be taken into account when choosing to rely on a CSP to support one's software applications.

[1] The reference example is Amazon Elastic Compute Cloud (EC2) (http://aws.amazon.com/ec2/).

[2] Such as Google App Engine (https://appengine.google.com/start) or the Salesforce1 platform (http://www.salesforce.com/platform/solutions/connect-integrate/).

The money that can be saved by relying on an external infrastructure might be counterweighted by the risk of having the business stalled for some time, or the software and data lost due to the shutdown of the CSP.

The problem changes with the size of the CSP. Relying on major CSPs, large enterprises with an insignificant risk of abruptly shutting down, mitigates the problem. But the type of service offered by these CSPs might not be suited for the needs of the would-be customer. Additionally, while the potential customer enterprise might enjoy some negotiating power against smaller CSPs, so as to adjust the cloud functionality to its own needs, no such power will be available against major corporations, leaving the enterprise the sole options to accept or reject the standard offer. The enterprise might have political reasons to prefer a small CSP over a large one (for example nationality); the reasons might be economic (favorable offers by the small CSP); or the choice might be pushed by the need for interoperability with partners relying on a specific cloud. However, small, start-up cloud providers are more subject to market fluctuations, and might at some time decide to stop providing hosting services to their customers, or to move to another business model. This might dramatically impact cloud customers who can consequently face operational interruption of their services, business losses, reputation degradation, etc.

Long-term viability is in the interest both of the client enterprise and the CSP: the former has an interest that the cloud infrastructure will be available and able to guarantee its service, whereas the CSP wants to be seen as a reliable entity which enterprises can rely upon, so both parties might be willing to undertake some measures against the risk of a sudden disappearance of the CSP (bankruptcy, acquisition by third parties and subsequent termination, forced closure by public authorities and so on).

2.2 Cloud Providers Viability: Real-Life Scenarios

Several incidents have been reported over the last years. Probably the most outstanding example so far is the storage provider Nirvanix [5], one of the major stakeholders in cloud storage services, which went out of business in September 2013, and customers had to recur to insourcing or migrate to other cloud providers. However, Nirvanix only gave its customers two weeks before the final shutdown. The customers, hosting terabytes of data on Nirvanix services, had to deal with a considerable interruption time that resulted into a halt of their running services. The impact of this incident on some major enterprises, including IBM (who used Nirvanix's cloud storage technology) and Dell (who had some agreements with Nirvanix) has not been disclosed.

In 2012, Megaupload, one of the pioneering companies in providing storage services, has been shut down by the US Department of Justice which started an investigation against its employees [1], and its founder Kim Dotcom created a new storage service known as MegaCloud. In late 2013, this service disappeared as well [14], possibly due to an NSA blockade. Users have suddenly lost access to the areas where they had stored the files.

The case of the 2e2 company highlights other interesting consequences. After 2e2 went bankrupt, it first asked its customers (such Vodafone and Kellog) a large amount of funding to keep the service up and running [20]. Shortly after, it was discovered that the company had outsourced most of its customers' data to third-party services. After the bankrupt, the datacentres were acquired by a company that guaranteed that it would keep providing the service to customers.

Cloud providers' viability has raised several criticism over the last years. According to Gartner, one of the major risks posed by cloud adoption is the uncertainty related to providers viability, and one out of four cloud providers could go out of business in 2015 due to some reasons such as bankruptcy or acquisition [18].

In an attempt to raise the attention on the problem of cloud providers' long-term viability issues, this paper explores some research directions with the objective of setting up trustworthy cloud environments and increasing cloud services adoption.

3 Related Work

To the best of our knowledge, there does not appear to be a lot of literature addressing the issue of CSP viability. Most studies only mention it as one of the potential problems that should be accounted for when migrating to clouds, or try to address more common problems that have some connection with viability.

Reliability is seen as a major concern, and steps have been taken to address the problem. [11] surveys a number of examples in which cloud reliability became a concern. Several reliability models and approaches have been proposed (e.g., [2]). However, reliability generally refers to short-term problems, in the sense of a continuous availability of the service without interruptions or outages. While there is clearly some overlapping with the problem of long-term viability, the solutions for short-term failures are not suited for addressing the final termination of the CSP, and the other way around approaches to achieve long-term viability might be unable to solve a service downtime lasting a few hours.

The main reference to the problem of CSP long-term viability resides in a popular Gartner analysis [4] which highlights seven critical features that a business should evaluate when moving to cloud; among these is long-term viability. The report is considered a milestone concerning cloud risks. A different risk classification is provided in [19], highlighting the problem of stored data in the event of bankruptcy.

Also [15] considers the various risks associated with the migration from an insourced infrastructure to a CSP, and the set of risks includes long-term viability. The author puts it in the perspective of contractual issues that have to be addressed but again does not provide a lot of insight into the subject.

A slightly more detailed analysis is offered by [3]. This work introduces the distinction among several scenarios in which the CSP undergoes a soft shutdown (i.e., filing for bankruptcy), merges or is acquired, and abruptly ceases its business. Apart from a brief overview on software escrows (described in Subsect. 4.3), the book does not explore other solutions.

Perhaps the only work which actually attempts to address the issue of a CSP going bankrupt or ceasing its business is [12]. However, this work exclusively focuses on legal solutions, without discussing their technical implications, and moreover it puts too much emphasis on the source code of services, most likely because, despite the initial statements, it only addresses SaaS environments. Some of the solutions proposed in [12] will be discussed in this paper.

4 Addressing the Problem

This section analyses the different methods and approaches that can be used to anticipate the needs for building trustworthy cloud platforms. In what follows, these approaches are presented taking into account an information technology perspective, a purely legal perspective, and finally an approach that combines both the IT and law perspectives.

4.1 Discussion of Possible IT Solutions

The risk of a CSP failure can be mitigated by means of some preemptive technical solutions that cannot prevent the CSP from ceasing its activity, but can help avoid the consequences of such an event, especially that of preventing the cloud customer to carry on its business. Although approaches based on redundancy and standardization will involve some additional costs, they might offer a viable trade-off between expenses and benefits.

Backing Up. As obvious as it may seem, the most immediate solution to protect against the sudden disappearance of a CSP is to regularly back up one's software and data. By means of an in-house duplication of all the assets that are outsourced to the cloud, an enterprise combines the advantages of both solutions: on one side, it does not require to maintain the high-performance, 24/7-reliable infrastructures to deliver its service to its customers; on the other side, it constantly has an up-to-date version of its assets, ready for migration to another CSP (or for insourcing).

Data backup is also advisable if the service delivery is in-house. In particular, it has been suggested [14] to use a cloud backup when delivering in-house, and to use an in-house backup when the services and data are outsourced.

On the down side, backing up the software and the data has its costs, but these will reasonably be smaller than those required for in-house operation. Also, this approach does not solve every problem, because the CSP might use technologies, protocols and Application Programming Interface (APIs) that are not compatible with an in-house operation, or for migration to a different CSP. So, backed-up data might still require a lot of work to restore full business operations.

That is, unless the CSPs are based on some standard technology or structure (as described in the next paragraph).

Cloud Platforms Portability and Standardization. In the last years, several PaaS platforms have appeared as an initiative to create and manage scalable cloud-based software. When a PaaS provider goes out of business, cloud customers should be able to migrate their application to others sites.

A cloud platform should provide standard connectors, so that applications can be moved to another platform adopting the same standards. In the last few years, several PaaS tools have been proposed by different PaaS market players. Some of these tools are based on standard programming languages, however they still suffer from the lack of standard APIs in terms of interfaces and applications connectors, thus exposing customers to vendor lock-in risks. With the lack of interoperability between platforms, customers' workload migration from one cloud provider to a new one becomes a tedious task, since the customer has to adapt its migrated applications to the proprietary components of the new provider's platform in order to migrate the software and the data. To ensure high interoperability and portability between PaaS, standardization initiatives have to be developed to provide standard application dependencies mechanisms and common interfaces to ease the transfer of applications between heterogeneous PaaS platforms.

A recent initiative to enable interoperability between platforms and to ease the migration of applications is illustrated by the containers model, whose basic idea is to have an isolated packaging of an application embedded with its dependencies. Using this approach, it becomes easier to move the overall application when its hosting environment encounters some problems. "Containerization" has been promoted by Linux Dockers[3], which adds an additional layer of abstraction to virtual machines, allowing to have isolated features within the same Linux instance.

Some authors [13] have presented some proposed standard interfaces such as DMTFs Open Cloud Standards Incubator (OCSI) for resources management, the Open Cloud Computing Interface Working Group (OCCI-WG) for IaaS specification, and Cloud Data Management Interface (CDMI) for the manipulation of data elements. Others [9] have discussed some standardization challenges related to credentials and network. Another initiative to achieve cloud interoperability has been presented in the Interoperability (LISI) Maturity Model [8] published by the Department of Defense (DoD). LISI explores four levels of system interoperability related to procedures, applications, infrastructure, and data.

By defining standards for the alignment of cloud development tools and platforms, the migration from one cloud to another becomes more flexible. Standardization can have a huge impact on the costs of migration, both in money and in time, when services and data need to be transferred from one cloud provider to another one.

[3] https://www.docker.com/.

Service Level Agreement. A Service Level Agreement is, in its essence[4], "an agreement between the service provider and its customers quantifying the Minimum acceptable service to the customer" (capital not added) [10].

Long-term viability would clearly be one requirement that is not fit for formal specification or monitoring. Nonetheless it could be addressed in a SLA, for example by addressing issues such as disaster recovery, data portability and an exit strategy. In particular, with respect to data portability, the parties might benefit from clauses specifying:

- The use of some interface which would ease migration to other providers, to protect the customer in case of data loss or business failure of the CSP;
- Strict conditions and terms under which the customer is entitled to enact the migration, to avoid damages to the CSP due to abuse on part of the customer.

Recently, the European Cloud Select Industry Group on Service Level Agreements (C-SIG SLA)[5] has released a set of guidelines toward a standardisaztion of SLAs. Among the service level objectives that should be covered by the SLA, the guidelines suggest including a termination process [7], with steps to enable the customer to retrieve their data. The problem with this clause, however, is that it might be difficult to apply in a short time in the case of a sudden bankruptcy of the CSP.

Service Choreographies. Cloud-based services are intricate by design as they are composed of a number of software services that maintain several features which depend on different stakeholders. To maintain these dynamic interactions between stakeholders, an abstraction layer that models the different flows of communications between cloud-based software entities is needed.

Choreographies [17] have been widely used in the context of web services to specify how services interact with each other. They commonly rely on the orchestration of every entity that can be involved in this choreography. An orchestration specifies from a central view the behaviour of collaborating parties and the flow of message exchanges between the different entities. A choreography gives more visibility about the interactions between the different collaborators, since it enables each collaborator in the choreography to observe all message exchanges involved in the different.

Choreographies can be revisited in the cloud context to provide a high-level view of the interactions between the different stakeholders involved in maintaining cloud services. Maintaining the status of the different collaborations between the different entities involved in the deployment, delivery and usage of a service eases the tracking of the partners taking part in the management of cloud-based applications when the provider goes out of business.

[4] Other definitions of Service Level Agreement (SLAs) exist. According to [22], SLAs are "a common way to formally specify the exact conditions (both functional and non-functional) under which services are or should be delivered".

[5] Information about the group can be found at https://ec.europa.eu/digital-agenda/en/cloud-select-industry-group-service-level-agreements.

4.2 Legal Perspective and Solutions

Contractual agreements can help the stakeholders of a cloud environment avoid the worries of the CSP's bankruptcy or cessation of business. If the problem is not addressed beforehand, the customer might incur into serious trouble, because *ex post* remedies can be quite ineffective.

Once the customer's assets fall into the bankrupt estate, recovering them might prove uncertain, expensive, and long enough to seriously damage the customer's business. In addition, the liablility of a CSP for not being able to provide its service in continuity is rather blurred (whereas provisions exist concerning liability for IP violations, failure in data protection, or negligence in security [21]), and CSPs normally operate by means of non-negotiable contracts that make frequent use of liability waivers. Customers are not guaranteed in case of a prolonged inactivity of the CSP which cascades upon them.

When filing for bankruptcy, the management of the company is transferred from the CEO or board to a trustee, under the surveillance of a tribunal. Licensors or third-party providers cannot take action against the bankrupt CSP without a judicial authorization, so customers have a few days to react. However, this protection can't be relied upon if human intervention is needed, because employees cannot be guaranteed to provide customer support [6]. This problem would also impact differently depending on the size of the CSP: the customers of a major provider would have an easier opportunity to recover their data, than those of a small enterprise.

If the CSP simply ceases its activity without filing for bankruptcy, the situation can be much more complicated. The customer will simply have the normal legal remedies such as filing a suit for a breach of contract, but this does not aid in recovering the assets, especially when manual intervention would be required on the CSP's side.

An insurance of the CSP in favor of the customer can reduce the risks by covering the losses suffered by the customer, although it would not ease the recovery of the assets. Also, a simpler form of private insurance might imply the CSP requesting a few months' payment in advance from the customer; this fee would have to be bound to ensuring the temporary continuation of the service (including manual operation) even in case of bankruptcy. This solution, combined with some agreement which would grant the customer access to its assets, would allow the customer to recover the data and prepare for the migration or insourcing of the services.

CSPs might use part of their revenues to establish a special guarantee fund devoted to keep the services up and running for several months, giving their customers the opportunity to take appropriate measures (a SaaS-guarantee fund has been suggested [23], but it can be applied to other paradigms as well). The fund could also be set up in a collaboration between several CSPs, but this would have the drawback that when an individual CSP in the consortium disappears, the others might feel like they are paying for its expenses, and this would be an inhibitor against setting up such a fund. At any rate, the authors acknowledge that no such fund exists at the moment.

In 2013, the Luxembourgish Parliament tried to address the problem of a CSP going bankrupt. The Parliament introduced provisions[6] that allow owners of intangible non-fungible goods held by a bankrupt to reclaim those data at their own expenses, provided they are separable from all other assets. The provision, according to the parliamentary discussion, was introduced specifically to address the issue of a CSP going bankrupt, although its application is not limited to cloud environments.

The Luxembourgish approach is definitely a step toward addressing the problem of CSP viability. However, it only allows to recover the assets, but does not help in relocating to some other CSP or insourcing, unless the services have been designed with portability in mind. Moreover, it does not work well with SaaS platforms, because the customer will not be able to recover the software, which is a shared platform and not a proprietary asset of the customer.

The imaginative solutions suggested in [12] place too much emphasis on the source code, and appear to address the legal issues but not the technical implications. A split copyright between the CSP and the customer consists in transferring part of software copyrights to the user, whereas a joint copyright transfers a share of the ownership; in both cases, the copyright of the whole source code does not fall into the bankrupt estate. The software code could be assigned in usufruct to the customer, or IP rights transferred to a foundation or association of customers. Notwithstanding the inefficiency of these ideas, which would hinder the CSP's ability to evolve or improve the services, they place too much emphasis on the source code. They may allow its partial recovery, but not ensure the continuity of the services, as the customer might not have the technical resources to use those assets.

Data ownership, on the other hand, can be detailed in the contract, so that the customer's data do not fall into the bankrupt estate. Again, this does not help if the CSP simply shuts down the operations, as the customer will not be able to recover the data even if it has the right to do so. Therefore, the contract should either guarantee some maintenance even after the business closure, at least for a short period, or provide the customer with some means of accessing the data without the intervention of the CSP personnel.

A CSP suffering bad financial conditions might not file for bankruptcy or cease its business, but rather be acquired or merged by a financially stronger enterprise. Under these conditions, the customer is relatively safe, because the acquiring enterprise will inherit the existing contracts of the CSP, giving the customer time to renegotiate the terms or migrate to other providers.

However, there might be some issues related to the regulations on personal data, due to the change of the data controller, something that is more and more frequent. Currently, European legislation on data protection does not explicitly address the issue. However, some national rulings have filled in the gaps, by requiring that the companies involved in the merger operation notify their customers about the terms of the operation.

[6] Luxembourgish Code of commerce, Article 567.

4.3 Mixed Approaches

Software escrows [16] are a part of intellectual property licenses that define necessary terms to maintain a product, including software code. A software escrow is based on a relationship between a licensor and licensee, set up through an escrow agent (a trusted third party). The agent delivers the software to the licensee and manages it according to an agreement. To ensure continuity of the service, software management is delegated to the licensee in case the licensor is not able to guarantee the software service anymore, due to bankruptcy or disaster. The licensee will then provide the service continuity.

Software escrow techniques must be revisited for cloud-based services. As the hosting of data and/or software is outsourced, the availability of services is not guaranteed by securing a traditional escrow (which only covers source code), but lies in the hands of the CSP, which is a key entity to ensure continuity of the services.

Software escrow solutions raise issue concerning software synchronization and the viability of software escrow agents, which is one of the major factors that reduces the assurance of service continuity in a software escrow service. Additionally, software escrows are mainly able to address a SaaS services model, whereas solutions for IaaS and PaaS providers require mirroring the infrastructure.

Dedicated escrow solutions for cloud-based software applications offer a mirroring of the whole application execution environment [23]. To maintain cloud-based software available when the provider goes out of service, the hardware resources, application platform, dependencies and on-line status execution are continuously mirrored. Companies like Iron Mountain and EscrowTech have developed some dedicated services for SaaS systems, offering business continuity options based on backup sites mirroring the services and the data. This solution is ideal since it enables to restore the whole provider's context when it goes out of business, assuring the continuity of services. In case of a service failure, the escrow agreement enables these backup sites to ensure the continuity of services. However, it is quite expensive for both the cloud customer and the provider because each service that runs in the cloud environment must be cloned.

Recently, an escrow alliance has described the essential building blocks for a cloud escrow solution[7]:

- The contract that states relevant elements related to the source code of applications and their underlying platforms, the SLAs and the users' context;
- An online deposit as a backup environment to ensure service availability;
- Verification mechanisms to control the backup and restoration of processes.

Both cloud providers and users can benefit from software escrow techniques, but there is still a lot of effort to be done by cloud stakeholders in the domain of software escrows, particularly with respect to standard agreements.

[7] http://www.escrowalliance.nl/en/escrow-solutions/cloud-escrow-solutions/.

Table 1. Summary of solutions for long-term viability.

Solution	Type	Applies to	Notes
Back-ups	IT	All	Some in-house infrastructure is required, but less expensive as the one needed to deliver the service. Ready duplicate of software and data for fast redeployment
Cloud portability	IT	IaaS, partly PaaS	Requires standards for cloud support which currently do not exist. Also requires the existence of a different CSP offering the same type of service
Service Level Agreement	IT	All	A SLA can define both functional and non-functional requirements, but metrics are required for enforcement. Such metrics might help assess the CSP's financial stability
Service choreographies	IT	All	Standard interfaces between services make it easier to replace an actor in case of a cessation of business. A governing board can also remove an actor if it doesn't provide enough guarantees
Contractual remedies	Legal	All	It might be hard to obtain the fulfilment of the CSP's obligations and damage restoration once it ceases business, and damage restoration does not help the customer reenact its business in a short time
Insurance	Legal	All	Should be combined with a contractual agreement to recover the customer's own assets, without the CSP's cooperation
Recovery of own data	Legal	PaaS, SaaS	Only applicable in Luxembourg so far
Split copyright	Legal	SaaS	Problems in splitting copyright over multiple customers and about the ownership of the data. Only allows the recovery of the source code which is not the key feature of modern cloud environments
Software escrow	Mixed	All	The agreement with the original CSP must address the transmission of data to the escrow. Additionally, having a mirrored service involves additional costs

5 Summary of Proposed Solutions

Table 1 offers a quick overview of the possible approaches that can be used to tackle the problem of long-term viability described in the previous pages.

The point, then, is what solutions a CSP should adopt, and what features an enterprise should look at when selecting a CSP, to have a guarantee of long-term viability. Due to space limitations, only some preliminary considerations will follow, while a detailed cost analysis of the technical and legal solutions is out of the scope of this paper and will be reserved for a future work.

Interoperability techniques and standards for cloud portability are easy and cheap improvements if adopted early on, while they become more complex and expensive as the business size and the number of customer grows. As it is unlikely that a large CSP disappears in a short time, redesigning a non-interoperable platform to allow easy portability of the services might be an appealing solution only for small and medium CSPs.

SLAs and service choreographies are more practical solutions, because they do not involve redesigning the services but only providing a guaranteed quality of service and adequate interfaces. They are fit for both small and large CSPs, and adopting a service choreography model can improve the quality of the single actor and of the overall service composition.

From a legal standpoint, an insurance that guarantees that the services will keep running for some months after the CSP goes out of business would help medium and large providers, while for small ones it might have too high a cost against the benefits it offers. The insurance cost would be charged back on the customers, and this might be possible only in an economy of scale.

Finally, software escrows and mirroring provide huge benefits to viability, but their cost might be prohibitive. Small enterprises might not be able to afford the costs of maintaining a duplicate of their services and data at an escrow provider, whereas large CSPs might have such a technological asset that doubling the resources might be a non-feasible task. In short, escrow services seem to provide a positive benefit-cost ratio only for medium-sized CSPs.

6 Conclusion

This work analyses the current cloud landscape by focusing on the challenges related to CSPs' long-term viability. One of the main risks associated to the cloud computing model has been pointed out, analysing the current challenges behind the continuity of services when the provider goes out of business.

Several approaches can be taken in advance, to mitigate the problems into which the customer can incur in case the CSP actually (and maybe suddenly) goes out of business. These approaches can ensure that the service keeps running steadily, albeit under a different legal entity (software escrow, but this is also what happens in the case of a merger); keep the services running for some time, to allow the customer to retrieve its assets (insurances and advance payments);

allow to quickly migrate the outsourced assets (standards and federated clouds); or allow recovery of the assets in a bankrupt procedure (under Luxembourgish law). Only the first category is long-term, while all others require a fast response on the customer's side.

The viability risk is minimal for big corporations; but when outsourcing to a small enterprise, appropriate measures are needed in case the CSP vanishes from one day to the next one. The problem of cloud providers' long-term viability has no easy solution, since it is unpredictable if and when a CSP will go out of business. Most of the approaches proposed in the context of this paper were not designed to address this problem specifically but can be tailored to mitigate it. The choice of a CSP can also be based on the degree of viability offered (such as specific contractual clauses). For this reason, the above measures serve not only to protect the customer against the loss of the means of doing its business, but also to increase the trust inspired by the CSP.

The problem is largely unexplored and would need further investigation. As a possible follow-up to this work, the authors envision a detailed cost analysis of the proposed solutions, and metrics to measure their effects.

References

1. Anthony, S.: Megaupload's demise: what happens to your files when a cloud service dies? January 2012. http://www.extremetech.com/computing/114803-megauploads-demise-what-happens-to-your-files-when-a-cloud-service-dies
2. Bauer, E., Adams, R.: Reliability and Availability of Cloud Computing, 1st edn. Wiley-IEEE Press, Hoboken (2012)
3. Bowen, J.A.: Legal issues in cloud computing. In: Buyya, R., Broberg, J., Goscinski, A.M. (eds.) Cloud Computing: Principles and Paradigms, 1st edn, pp. 593–613. Wiley, Hoboken (2011). Chap. 24
4. Brodkin, J.: Gartner: seven cloud-computing security risks. Technical report, Gartner, July 2008
5. Butler, B.: The best time to prepare for getting data out of the cloud isbefore you put it in there, June 2014. http://www.networkworld.com/article/2173255/cloud-computing/cloud-s-worst-case-scenario-what-to-do-if-your-provider-goes-belly-up.html
6. Caplan, D.S.: Bankruptcy in the cloud: effects of bankruptcy by a cloudservices provider. Technical report, Law Offices of David S. Caplan, 1289 FordhamBlvd., Suite 345 Chapel Hill, NC 37514, USA, August 2010. http://ftp.documation.com/references/ABA10a/PDfs/3_3.pdf
7. Cloud Select Industry Group on Service Level Agreements (C-SIG SLA), Brussels: Cloud Service Level Agreement Standardisation Guidelines, June 2014. http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?action=display&doc_id=6138
8. Dowell, S., Barreto, A., Michael, J.B., Shing, M.T.: Cloud to cloud interoperability. In: Proceedings of the 6th International Conference on System of Systems Engineering (SoSE), pp. 258–263. IEEE, Albuquerque, June 2011

9. Harsh, P., Dudouet, F., Cascella, R.G., Jegou, Y., Morin, C.: Using open standards for interoperability - issues, solutions, and challenges facing cloud computing. In: Proceedings of the 8th International Conference on Network and Service Management (CNSM) and 6th International DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standards and the Cloud (SVM), pp. 435–440. IEEE, Las Vegas, October 2012
10. Hiles, A.: Service Level Agreements: Winning a Competitive Edge for Support & Supply Services. Rothstein Associates Inc., Brookfield (2000). Rothstein Catalog on Service Level Books
11. Hu, F., Qiu, M., Li, J., Grant, T., Tylor, D., McCaleb, S., Butler, L., Hamner, R.: A review on cloud computing: design challenges in architecture and security. J. Comput. Inf. Technol. **19**(1), 25–55 (2011)
12. Louwers, E.J.: Continuity in the cloud: new practical solutions required. In: iTech-Law 2013 European Conference, October 2013
13. Machado, G.S., Hausheer, D., Stiller, B.: Considerations on the interoperability of and between cloud computing standards. In: 27th OpenGrid Forum (OGF27), G2C-Net Workshop: From Grid to Cloud Networks, OGF, Banff, Canada, October 2009. http://dx.doi.org/10.5167/uzh-24316
14. McKendrick, J.: What to do in case your cloud provider falls off the grid, November 2013. http://www.forbes.com/sites/joemckendrick/2013/11/04/what-to-do-in-case-your-cloud-provider-falls-off-the-grid/
15. Mills, L.H.: Legal issues associated with cloud computing, May 2009. http://www.secureit.com/resources/Cloud%20Computing%20Mills%20Nixon%20Peabody\%205-09.pdf
16. Pappous, P.A.: The software escrow: the court favorite and bankruptcy law. St. Clara High Technol. Law J. **1**(2), 309–326 (1985)
17. Peltz, C.: Web services orchestration and choreography. Computer **36**(10), 46–52 (2003)
18. Thibodeau, P.: One in four cloud providers will be gone by 2015, December 2013. http://www.computerworld.com/article/2486691/cloud-computing/one-in-four-cloud-providers-will-be-gone-by-2015.html
19. Van Hoboken, J., Arnbak, A., Van Eijk, N.: Obscured by clouds or how to address governmental access to cloud data from abroad. In: Proceedings of the 6th Annual Privacy Law Scholars Conference (PLSC), June 2013
20. Venkatraman, A.: 2e2 datacentre administrators hold customers' data to 1m ransom, February 2013. http://www.computerweekly.com/news/2240177744/2e2-datacentre-administrators-hold-customers-data-to-1m-ransom
21. Weber, R.H., Staiger, D.N.: Cloud computing: a cluster of complex liability issues. Web J. Curr. Leg. Issues **20**(1) (2014). http://webjcli.org/article/view/303
22. Wieder, P., Butler, J.M., Theilmann, W., Yahyapour, R. (eds.): Service Level Agreements for Cloud Computing. Springer Science+Business Media, LLC, New York (2011)
23. van de Zande, T., Jansen, S.: Business continuity solutions for SaaS customers. In: Regnell, B., van de Weerd, I., De Troyer, O. (eds.) ICSOB 2011. LNBIP, vol. 80, pp. 17–31. Springer, Heidelberg (2011)