Architectural Patterns for the Secure Multicloud Architectures

Ishwarappa Kalbandi [1], Anuradha J[2], P.P.Halkarnikar[3]
[1, 2, 3] Dept. of Computer Engineering
[1, 3] Dr. D. Y. Patil Institute of Engineering, Management & Research Akurdi, Pune, India
[2] Vellore Institute of Technology Vellore, TamilNadu,India

Abstract- *Security challenges are still among the biggest obstacles when considering the adoption of cloud services. This triggered a lot of research activities, resulting in a quantity of proposals targeting the various cloud security threats. Alongside with these security issues, the cloud paradigm comes with a new set of unique features, which open the path toward novel security approaches, techniques, and architectures. This paper provides a survey on the achievable security merits by making use of multiple distinct clouds simultaneously. Various distinct architectures are introduced and discussed according to their security and privacy capabilities and prospects*

Keywords- Cloud, security, privacy, multicloud, application partitioning, tier partitioning, data partitioning, multipart computation

I. INTRODUCTION

CLOUD computing offers dynamically scalable resources provisioned as a service over the Internet. The third party, on-demand, self-service, pay-per-use, and seamlessly scalable computing resources and services offered by the cloud paradigm promise to reduce capital as well as operational expenditures for hardware and software.

Clouds can be categorized taking the physical location from the viewpoint of the user into account. A public cloud is offered by third-party service providers and involves resources outside the user's premises. In case the cloud system is installed on the user's premise—usually in the own data center—this setup is called private cloud. A hybrid approach is denoted as hybrid cloud. This paper will concentrate on public clouds, because these services demand for the highest security requirements but also as this paper will start arguing includes high potential for security prospects.

In public clouds, all of the three common cloud service layers (IaaS, Paas, SaaS) share the commonality that the end-users' digital assets are taken from an intra organizational to an inter organizational context. This creates a number of issues, among which security aspects are regarded as the most critical factors when considering cloud computing adoption. Legislation and compliance frameworks raise further challenges on the outsourcing of data, applications, and processes. The high privacy standards in the European Union, e.g., and their legal variations between the continent's countries give rise to specific technical and organizational challenges.

One idea on reducing the risk for data and applications in a public cloud is the simultaneous usage of multiple clouds. Several approaches employing this paradigm have been proposed recently. They differ in partitioning and distribution patterns, technologies, cryptographic methods, and targeted scenarios as well as security levels. This paper contains a survey on these different securities by multicloud adoption approaches. It provides four distinct models in form of abstracted multicloud architectures. These developed multicloud architectures allow to categorize the available schemes and to analyze them according to their security benefits. An assessment of the different methods with regards to legal aspects and compliance implications is given in particular. The rest of this paper is organized as follows: Section 2 motivates the need for effective cloud security countermeasures by briefly reviewing the current state of play. The observations further lead to the fact that most of the research and development work is currently devoted to dedicated security schemes, which do not consider the specific properties of the cloud itself. Only recently some proposals on making use of multiple distinct clouds at the same time to realize security goals started to appear. To provide a formal ground to categorize and analyze these proposals, we propose a set of four distinct multicloud architectures. These multicloud architectures are introduced in Section 3 and each of them is further discussed in Sections 4, 5, 6, and 7, including case studies. Section 8 provides a consideration of legal and compliance aspects. Finally, in Section 9, an assessment and comparison of the presented approaches is given.

II. CLOUD SECURITY ISSUES

Cloud computing creates a large number of security issues and challenges. A list of security threats to cloud computing is presented in [5]. These issues range from the required trust in the cloud provider and attacks on cloud interfaces to misusing the cloud services for attacks on other systems.

The main problem that the cloud computing paradigm implicitly contains is that of secure outsourcing of sensitive as well as business-critical data and processes. When considering using a cloud service, the user must be aware of the fact that all data given to the cloud provider leave the own control and protection sphere. Even more, if deploying data-processing applications to the cloud (via IaaS or PaaS), a cloud provider gains full control on these processes. Hence, a strong trust relationship between the cloud provider and the cloud user is considered a general prerequisite in cloud computing.

Depending on the political context this trust may touch legal obligations. For instance, Italian legislation requires that government data of Italian citizens, if collected by official agencies, have to remain within Italy. Thus, using a cloud provider from outside of Italy for realizing an E-government service provided to Italian citizens would immediately violate this obligation. Hence, the cloud users must trust the cloud provider hosting their data within the borders of the country and never copying them to an off-country location (not even for backup or in case of local failure) nor providing access to the data to entities from abroad.

An attacker that has access to the cloud storage component is able to take snapshots or alter data in the storage. This might be done once, multiple times, or continuously. An attacker that also has access to the processing logic of the cloud can also modify the functions and their input and output data. Even though in the majority of cases it may be legitimate to assume a cloud provider to be honest and handling the customers' affairs in a respectful and responsible manner, there still remains a risk of malicious employees of the cloud provider, successful attacks and compromisation by third parties, or of actions ordered by a subpoena.

In [6], an overview of security flaws and attacks on cloud infrastructures is given. Some examples and more recent advances are briefly discussed in the following. Risten part et al. [7], [8] presented some attack techniques for the virtualization of the Amazon EC2 IaaS service. In their approach, the attacker allocates new virtual machines until one runs on the same physical machine as the victim's machine. Then, the attacker can perform cross-VM side channel attacks to learn or modify the victim's data. The authors present strategies to reach the desired victim machine with a high probability, and show how to exploit this position for extracting confidential data, e.g., a cryptographic key, from the victim's VM. Finally, they propose the usage of blinding techniques to fend cross-VM side-channel attacks.

In [9], a flaw in the management interface of Amazon's EC2 was found. The SOAP-based interface uses XML Signature as defined in WS-Security for integrity protection and authenticity verification. Gruschka and Iacono [9] discovered that the EC2 implementation for signature verification is vulnerable to the Signature Wrapping Attack [10]. In this attack, the attacker—who eaves dropped a legitimate request message—can add a second arbitrary operation to the message while keeping the original signature. Due to the flaw in the EC2 framework, the modification of the message is not detected and the injected operation is executed on behalf of the legitimate user and billed to the victim's account.

A major incident in a SaaS cloud happened in 2009 with Google Docs [11]. Google Docs allows users to edit documents (e.g., text, spreadsheet, presentation) online and share these documents with other users. However, this system had the following flaw: Once a document was shared with anyone, it was accessible for everyone the document owner has ever shared documents with before. For this technical glitch, not even any criminal intent was required to get unauthorized access to confidential data. Recent attacks have demonstrated that cloud systems of major cloud providers may contain severe security flaws in different types of clouds (see [12], [13]). As can be seen from this review of the related work on cloud system attacks, the cloud computing paradigm contains an implicit threat of working in a compromised cloud system. If an attacker is able to infiltrate the cloud system itself, all data and all processes of all users operating on that cloud system may become subject to malicious actions in an avalanche manner. Hence, the cloud computing paradigm requires an in-depth reconsideration on what security requirements might be affected by such an exploitation incident.

For the common case of a single cloud provider hosting and processing all of its user's data, an intrusion would immediately affect all security requirements: Accessibility, integrity, and confidentiality of data and processes may become violated, and further malicious actions may be performed on behalf of the cloud user's identity.

These cloud security issues and challenges triggered a lot of research activities, resulting in a quantity of proposals targeting the various cloud security threats. Alongside with these security issues, the cloud paradigm comes with a new set of unique features that open the path toward novel security approaches, techniques, and architectures. One promising concept makes use of multiple distinct clouds simultaneously.

III. SECURITY PROSPECTS BY MULTICLOUD ARCHITECTURES

The basic underlying idea is to use multiple distinct clouds at the same time to mitigate the risks of malicious data manipulation, disclosure, and process tampering. By integrating distinct clouds, the trust assumption can be lowered to an assumption of noncollaborating cloud service providers. Further, this setting makes it much harder for an external attacker to retrieve or tamper hosted data or applications of a specific cloud user.

The idea of making use of multiple clouds has been proposed by Bernstein and Celesti [14], [15]. However, this previous work did not focus on security. Since then, other approaches considering the security effects have been proposed. These approaches are operating on different cloud service levels, are partly combined with cryptographic methods, and targeting different usage scenarios.

In this paper, we introduce a model of different architectural patterns for distributing resources to multiple cloud providers. This model is used to discuss the security benefits and also to classify existing approaches. In our model, we distinguish the following four architectural patterns:

Replication of applications allows to receive multiple results from one operation performed in distinct clouds and to compare them within the own premise (see Section 4). This enables the user to get an evidence on the integrity of the result.

Partition of application System into tiers allows separating the logic from the data (see Section 5). This gives additional protection against data leakage due to flaws in the application logic.

Partition of application logic into fragments allows distributing the application logic to distinct clouds (see Section 6). This has two benefits. First, no cloud provider learns the complete application logic.

Second, no cloud provider learns the overall calculated result of the application. Thus, this leads to data and application confidentiality.

Partition of application data into fragments allows distributing fine-grained fragments of the data to distinct clouds (see Section 7). None of the involved cloud providers gains access to all the data, which safeguards the data's confidentiality.

Each of the introduced architectural patterns provides individual security merits, which map to different application scenarios and their security needs. Obviously, the patterns can be combined resulting in combined security merits, but also in higher deployment and runtime effort. The following sections present the four patterns in more detail and investigate their merits and flaws with respect to the stated security requirements under the assumption of one or more compromised cloud systems.

IV. REPLICATION OF APPLICATION

How does a cloud customer know whether his data were processed correctly within the cloud? There is no technical way to guarantee that an operation performed in a cloud system was not tampered with or that the cloud system was not compromised by an attacker. The only kind of guarantee is based on the level of trust between the cloud Customer and the cloud provider and on the contractual regulations made between them such as SLAs, applicable laws, and regulations of the involved jurisdictional domains. But even if the relation and agreements are perfectly respected by all participants, there still remains a residual risk of getting compromised by third parties. To solve this intrinsic problem, multiple distinct clouds executing multiple copies of the same application can be deployed (see Fig. 1). Instead of executing a particular application on one specific cloud, the same operation is executed by distinct clouds. By comparing the obtained results, the cloud user gets evidence on the integrity of the result. In such a setting, the required trust toward the cloud service provider can be lowered dramatically. Instead of trusting one cloud service provider totally, the cloud user only needs to rely on the assumption that the cloud providers do not collaborate maliciously against themselves. Assume that $n > 1$ clouds are available (like, e.g., Clouds A and B in Fig. 1). All of the n adopted clouds perform the same task. Assume further that f denotes the number of malicious clouds and that $n \_f > f$ the majority of the clouds are honest. The correct result can then be obtained by the cloud user by comparing the results and taking the majority as the correct one. There are other methods of deriving the correct result, for instance using the TurpinCoan. Algorithm [16] for solving the general Byzantine

Agreement problem. Instead of having the cloud user performing the verification task, another viable approach consists in having one cloud monitoring the execution of the other clouds. For instance, Cloud A may announce intermediate results of its computations to an associated monitoring process running at Cloud B. This way, Cloud B can verify that Cloud A makes progress and sticks to the computation intended by the cloud user. As an extension of this approach, Cloud B may run a model checker service that verifies the execution path taken by CloudA on-the-fly, allowing for immediate detection of irregularities. This architecture enables to verify the integrity of results obtained from tasks deployed to the cloud. On the other hand, it needs to be noted that it does not provide any protection in respect to the confidentiality of data or processes. On the contrary, this approach might have a negative impact on the confidentiality because—due to the deployment of multiple clouds—the risk rises that one of them is malicious or compromised. To implement protection against an unauthorized access to data and logic this architecture needs to be combined with the architecture described in Section 5.

The idea of resource replication can be found in many other disciplines. In the design of dependable systems, for example, it is used to increase the robustness of the system especially against system failures [17]. In economic business processes—and especially in the management of supply chains—single-source suppliers are avoided to lower the dependency on suppliers and increase the flexibility of the business process [18]. In all these cases, the additional overhead introduced by doing things multiple times is accepted in favor of other goals resulting from this replication.

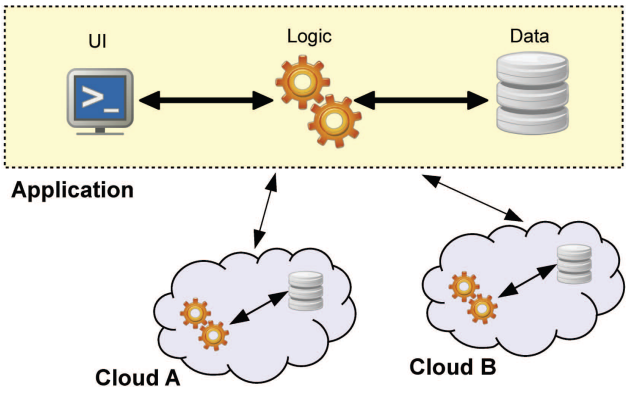

Figure 1. Replication of application systems.

V. PARTITION OF APPLCATION SYSTEM INTO TIERS

The architectural pattern described in the previous Section 4 enables the cloud user to get some evidence on the integrity of the computations performed on a third-party's resources or services. The architecture introduced in this section targets the risk of undesired data leakage. It answers the question on how a cloud user can be sure that the data access is implemented and enforced effectively and that errors in the application logic do not affect the user's data? To limit the risk of undesired data leakage due to application logic flaws, the separation of the application system's tiers and their delegation to distinct clouds is proposed (see Fig. 2). In case of an application failure, the data are not immediately at risk since it is physically separated and protected by an independent access control scheme. Moreover, the cloud user has the choice to select a particular—probably specially trusted—cloud provider for data storage services and a different cloud provider for applications It needs to be noted, that the security services provided by this architecture can only be fully exploited if the execution of the application logic on the data is performed on the cloud user's system. Only in this case, the application provider does not learn anything on the users' data. Thus, the SaaS-based delivery of an application to the user side in conjunction with the controlled access to the user's data performed from the same user's system is the most farreaching instantiation

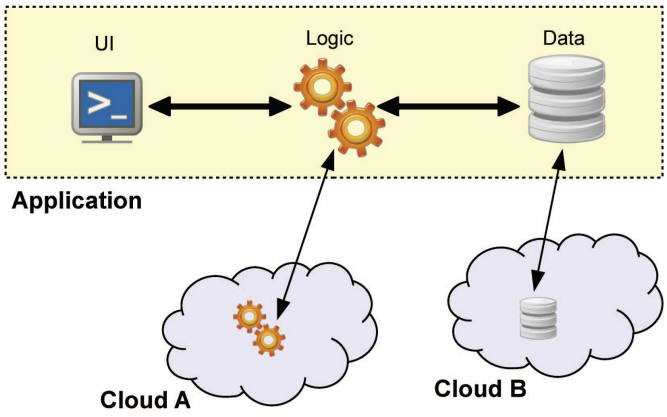

Figure 2.

Besides the introduced overhead due to the additionally involved cloud, this architecture requires, moreover, standardized interfaces to couple applications with data services provided by distinct parties. Also generic data services might serve for a wide range of applications there will be the need for application specific services as well. The partitioning of application systems into tiers and distributing the tiers to distinct clouds provides some coarsegrained protection against data leakage in the presence of flaws in application design or implementation. This architectural concept can be applied to all three cloud layers.

VI. PARTITION OF APPLICATION LOGIC INTO FRAGMENTS

This architecture variant targets the confidentiality of data and processing logic. It gives an answer to the following question: How can a cloud user avoid fully revealing the data or processing logic to the cloud provider? The data should not only be protected while in the persistent storage, but in particular when it is processed.

The idea of this architecture is that the application logic needs to be partitioned into fine-grained parts and these parts are distributed to distinct clouds (see Fig. 3). This approach can be instantiated in different ways depending on how the partitioning is performed. The clouds participating in the fragmented applications can be symmetric or asymmetric in terms of computing power and trust. Two concepts are common. The first involves a trusted private cloud that takes a small critical share of the computation, and an untrusted public cloud that takes most of the computational load. The second distributes the computation among several untrusted public clouds, with the assumption that these clouds will not collude to breakthe security.

Obfuscating Splitting

By this approach, application parts are distributed to different clouds in such a way, that every single cloud has only a partial view on the application and gains only limited knowledge. Therefore, this method can also hide parts of the application logic from the clouds. For application splitting, a first approach is using the existing sequential or parallel logic separation. Thus, depending on the application, every cloud provider just performs subtasks on a subset of data.

An approach by Danezis and Livshits [20] is build around secure storage architecture and focusing on online service provisioning, where the service depends on the result of function evaluations on the user's data. This proposal uses the cloud as a secure storage, with keys remaining on client side, e.g., in a private cloud. The application is split in the following way: The service sends the function to be evaluated to the client. The client retrieves his necessary raw data and processes it according to the service needs. The result and a proof of correctness are given back to the service providing public cloud. In the cloud, the remaining functionality of the service is offered based on the aggregated input of the clients. This architecture protects the detailed user data, and reveals only what the cloud needs to know to provide the service. Similarly, the FlexCloud approaches [21] is based on interconnecting local, private computing environments to a semitrusted public cloud for realizing complex workflows or secure distributed storage. This approach utilizes multiple resource-constrained secure computation environments ("private clouds") to form a collaborative computing environment of similar trust level, a trustworthy "community cloud." A difficult challenge of obfuscating splitting in general is the fact that there is no generic pattern for the realization. Careful analysis where the application can be split into fragments must be performed regarding its confidentiality, i.e., checking if the information that the participating cloud providers receive is really innocuous

Homomorphic Encryption and Secure

I. MULTIPARTY COMPUTATION

Homomorphic encryption and secure multiparty computation both use cryptographic means to secure the data while it is processed. In homomorphic encryption, user encrypts the data with his public key and uploads the ciphertexts to the Cloud. The cloud can independently compute on the encrypted data to obtain an encrypted result, which only the user can decrypt. Therefore, in our scenario, homomorphic encryption uses an asymmetric fragmentation, where the user (or a small trusted private cloud) manages the keys and performs the encryption and decryption operations, while the massive computation on encrypted data is done by an untrusted public cloud. The possibility of fully homomorphic encryption supporting secure addition and multiplication of ciphertexts was first suggested in [22]. However, for a long time all known homomorphic encryption schemes supported efficiently only one operation [23], [24]. Therefore, the recent discovery of fully homomorphic encryption by Gentry [25],

Asharov et al. [26] had a tremendous impact on the cryptographic community and revived research in this field. In the case of homomorphic encryption, the cloud has the main share of work, as it operates on the encrypted inputs to compute the encrypted output. However, the algorithms are far from being practical, so the vision of clouds based on homomorphic encryption seems unreal for the foreseeable future. In addition, the applicability is limited, as for services that go beyond the outsourcing of computation, intermediate or final results need to be decrypted. This requires either interaction with the entity that holds the key (e.g., a private cloud) or the key is shared among several clouds who then assist in decrypting values that are needed in clear with a threshold encryption scheme [27]. The idea of secure multiparty computation was first presented in [28] as a solution to the millionaires' problem: Two millionaires want to find out who is richer without disclosing any further information about their wealth. Two main variants of secure

multiparty computation are known: Based on linear secret sharing [29] or garbled circuits [30]. Schemes based on a linear secret sharing scheme work as follows: The user computes and distributes the shares to the different clouds. The clouds will jointly compute the function of interest on these shares, communicating with each other when necessary. In the end, the clouds hold shares of the result which is sent back to the user who can reconstruct the result. At least three clouds are necessary for this scheme and no two of them should collude. The approach of garbled circuits works as follows: One cloud generates a circuit that is able to compute the desired function and encrypts this circuit producing a garbled circuit, which is however still executable. Then, this cloud assists the users in encrypting their inputs accordingly. Another cloud needs now to be present to evaluate the circuit with the user's inputs. Thus, this scheme requires in general only two clouds. Although the ideas of multiparty computation are old, it is ongoing research to reduce the overhead by multiparty computation. A recent improvement, e.g., on equality and comparison of values, has lead to the constructions of programming frameworks, which can already be considered practical [31], [32]. An example architecture that uses garbled circuits is the TwinClouds approach [33] that utilizes a private cloud for preparation of garbled circuits. The circuits itself is then evaluated within a high-performance commodity cloud of lower trust level-without lowering the security guarantees for the processes outsourced to the public cloud. In all cases, using secure multiparty computation in distinct clouds guarantees the secrecy of the input data, unless the cloud providers collude to open shares or decrypt inputs. Assuming that the cloud provider itself is not malicious, but might be compromised by attacks or have single malicious employees, this collusion is hard to establish so that a good protection is given. A multiparty computation between clouds makes it possible to compute a function on data in a way that no cloud provider learns anything about the input or output data

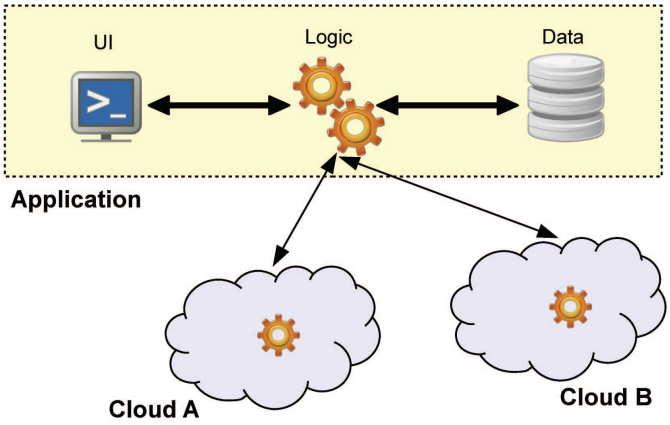

Figure 3. Partition of application logic into fragments.

VII. PARTITION OF APPLICATION DATA INTO FRAGMENTS

This multicloud architecture specifies that the application data is partitioned and distributed to distinct clouds (see Fig. 4).The most common forms of data storage are files and databases. Files typically contain unstructured data (e.g., pictures, text documents) and do not allow for easily splitting or exchanging parts of the data. This kind of data can only be partitioned using cryptographic methods (see Section 7.1). Databases contain data in structured form organized in columns and rows. Here, data partitioning can be performed by distributing different parts of the database (tables, rows, columns) to different cloud providers (see Section 7.2). Finally, files can also contain structured data (e.g., XML data). Here, the data can be splitted using similar approaches like for databases. XML data, for example, can be partitioned on XML element level. However, such operations are very costly. Thus, this data are commonly rather treated using cryptographic data splitting.

1. Cryptographic Data Splitting

Probably, the most basic cryptographic method to store data securely is to store the data in encrypted form. While the cryptographic key could remain at the user's premises, to increase flexibility in cloud data processing or to enable multiuser systems it is beneficial to have the key available online when needed [38]. This approach, therefore, distributes key material and encrypted data into different clouds. For instance, with XML data, this can, e.g., be done inside the XML document by using XML encryption [39]. A similar approach is taken by several solutions for secure Cloud storage: The first approach to cryptographic cloud storage [40] is a solution for encrypted key/value storage in the cloud while maintaining the ability to easily access the data. It involves searchable encryption [41], [42] as the key component to achieve this. Searchable encryption allows keyword search on encrypted data if an authorized token for the keyword is provided. The keys are stored in a trusted private cloud whereas the data resides in the untrusted public cloud (see Section 6.2). One example of a relational database with encrypted data processing is CryptDB [43]. The database consists of a database server that stores the encrypted data and a proxy that holds the keys and provides a standard SQL interface to the user. The data are encrypted in different layers with schemes such as order-preserving encryption [44], homomorphic encryption [23], searchable encryption [41], and a standard symmetric encryption system, such as the AES. For every SQL query, the proxy identifies and provides only the necessary keys to the server, so that exactly this query can be answered. Obviously, this implies that the database server may

learn more keys with every new query. Hence, security against persistent attackers is certainly limited here. The advantage of cryptDB lies in the fact that the database part is a standard MySQL database, and in that its efficiency is only decreased marginally, as compared to unencrypted data storage. Another option is to compute a secret sharing of the data. In a secret sharing protocol, no single cryptographic key is involved. Instead, secret sharing splits the data into multiple shares in such a way that the data can only be reconstructed if more shares than a given threshold are collected. This method integrates well with multiparty computation, as presented in Section 6.2. As discussed, multiparty computation often operates on such shares, so that the clouds that form the peers of the multiparty protocol can store their shares permanently without any loss of security.

2. Database Splitting

For protecting information inside databases, one has to distinguish two security goals: confidentiality of data items (e.g., a credit card number) or confidentiality of data item relationships (e.g., the items "Peter" and "AIDS" are not confidential, but their relationship is). In the first case, data splitting requires a scenario—similar to other approaches presented before—with a least one trusted provider (or additional encryption; see below). However, very often only the relationship shall be protected, and this can be achieved using just honest-but-curious providers. A typical way of database splitting is pseudonymization: One provider receives the data with some key fields replaced by a random identifier, and the second provider receives the mapping of the identifier to the original information. This approach is used, for example, in a commercial cloud security gateway [45]. For splitting a database table, there are two general approaches: Vertical fragmentation and horizontal fragmentation [46]. With vertical fragmentation, the columns are distributed to cloud providers in such a way that no single provider learns a confidential relationship on his own. A patient health record, for example, might be fragmented into two parts, e.g., (name, patient number) and (patient number, disease).

This way, the individual providers only learn noncritical data relations. However, for real-world applications, it is a nontrivial task to find such a fragmentation. First, new relations can be learned by performing transitive combination of existing ones. Second, some relations can be concluded using external knowledge. If, in the example above, the first provider additionally learns about the relation (patient number, medication), he has technically still no knowledge about the patient's disease. However, someone with pharmaceutical background can derive the disease from the medication. Further, new relations can also be derived by combining multiple data sets. For instance, using again the relation of (patient number, medication), the knowledge of a combination of medications can ease the guessing of the patient's disease. Thus, also on a row level, database splitting might be required. This is called horizontal fragmentation. Finally, database splitting can also be combined with encryption. Using key management mechanisms like mentioned before, some database columns are encrypted. The combination of encryption and splitting protects confidential columns and still allows querying database entries using plain text columns

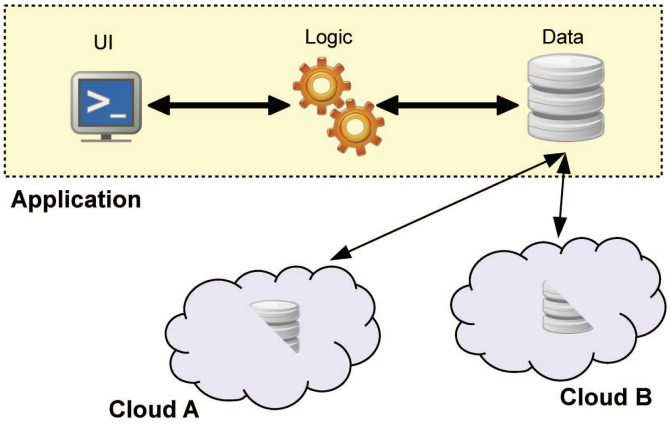

Figure 4. Partition of application data into fragments

VIII. LEGAL COMPLIANCE WITH MULTICLOUD ARCHITECTURES

Since legislation traditionally only slowly copes with technological paradigm shifts, there are few to none cloud specific regulations in place by now. Therefore, for cloud computing the same legal framework is applicable as for any other means of data processing. Generally, legal compliance does not distinguish between different meansof technologies but rather different types of information. For instance, enterprises are facing other legal requirements for the lawful processing of their tax information than for the lawful processing of their Customer Relationship Management. A one-cloud-fits-all approach does not reflect these differing compliance requirements. Multicloud architectures may be a viable solution for enterprises to address these compliance issues. Hence, this section gives a coarse-grained legal analysis on the different approaches, and their flaws and benefits in terms of compliance and privacy impact. The conflict between cloud computing and the world of laws and policies results from the borderless nature of clouds in contrast to the mostly national scope of legal frameworks. The most successful cloud service providers operate their clouds across national borders in multiple data centers all over the globe. Hence, they can offer high availability even in case of regional failure as well as reduced costs because of their choice of location. In contrast, the cloud customer is subject to its

national legal requirements, and faces the problem to ensure legal compliance to national laws in a multinational environment. This conflict is neither new nor unique to cloud computing, but the highly dynamic and virtualized nature of clouds intensifies it as the applicability of laws relate to physical location. The legal uncertainties of cloud computing, especially in Europe with its strict data protection laws, are subject to an ongoing discussion. Nevertheless, legal experts agree that lawful cloud computing is possible as long as the adequate technical, organizational, and contractual safeguards for the specific type of information to be processed are in place.

Before deciding on which cloud service type to use, be it public, private, or hybrid, IaaS, PaaS, or SaaS, the enterprise needs to conduct a risk assessment. This risk assessment is not only the best practice but also sometimes legally mandatory, e.g., in form of a Privacy Risk Assessment in the proposed European Data Protection Regulation [48]. A proper risk assessment before "going cloud" means to identify one's internal processes and the relevant information involved in these processes, a risk and threat analysis, as well as identifying legal compliance requirements that have to be met and the necessary safeguards to be installed. European Economic Area to outsource the processing of personal identifiable information has to adhere to the EU Data Protection Directive [53]. This includes mandatory contractual safeguards for the export of personal data, including mandatory contractual safeguards such as Standard Contractual Clauses and Binding Corporate Rules (see [54], [55]). Furthermore, much national legislation requires specific information to stay within the national borders of the country. This typically applies to information regarding national security, but also to information of public authorities or electronic health records.

Potential cloud customers are facing several of these requirements for security controls, standards, and certifications, probably even varying per process. Identifying one cloud service provider to offer all of these options like a modular system seems impossible. Multicloud approaches may help addressing these issues. As discussed next, the compliance benefits and drawbacks of the identified multicloud architectures, in general, seem auspicious.

1. Replication of Application

This approach appears to have the fewest benefits regarding legal compliance as it multiplies the necessity to identify and choose a cloud service provider perfectly tailored for the requirements of the relevant process and information. Since this could mean negotiating and concluding individual contracts with several cloud service providers, replicating a highly sensitive process or an application seems to unreasonably tie up personnel and financial resources. Therefore, this approach has its value for information and processes with low sensitivity but high availability and soundness requirements.

2. Partition of Application System into Tiers

The separation of logic and data offers the possibility to store the data in the cloud with compliant controls and safeguards and to outsource the processing logic to a not specifically certified cloud with favorable price. It also allows for storing the data in a national cloud while the application logic is outsourced to a multinational one. A drawback of this approach is that the compliant separation of logic and data is only possible if the application provider does not receive the customer's data in any case. The processing needs to take place in an environment as secure and certified as the chosen storage

cloud. This can either be the customer's own premise, an approach that almost annihilates the benefits of outsourcing, cost reduction, and seamless scalability of using cloud computing, because the customer needs to provision sufficient and compliant resources by himself. Alternatively, the application logic can also take place in a different tier of the compliant storage cloud, or on a different cloud with similar compliance level. The drawback of this approach obviously is that the customer has to fully trust those cloud service providers that receive all information, logic, and data. This somewhat contradicts the initial motivation of this multicloud approach.

3. Partition of Application Logic/Data

a) Obfuscating Splitting and Database Splitting

These approaches are especially valuable for dealing with personal identifiable data. Segmenting personal identifiable data—if realized in a reasonable way—is a viable privacy safeguard. Best practice would be to separate the data in a way that renders the remaining data pseudonymous. Pseudonymity itself is a privacy safeguard (see [56, Section 3a]). Therefore, outsourcing seudonymized information, which is unlinkable to a specific person, does require considerable less additional safeguards as ompared to nonpseudonymized information. Pseudonymization based on the Obfuscated Splitting approach could be used, e.g., in Human Resources or Customer Relationship Management. A potential cloud customer would have to remove all directly

identifying data in the first place, like name, social security number, credit card information, or address, and store this information separately, either on premise or in a cloud with adequately high-security controls. The remaining data can still be linked to the directly identifying data by means of an unobvious identifier (the pseudonym), which is unusable for any malicious third parties. The unlinkability of the combined pseudonymized data to a person can be ensured by performing a carefully conducted privacy risk assessment.

These assessments are always constrained by the assumptions of an adversary's "reasonable means" [53Recital 26]. The cloud customer has the option to outsource the pseudonymized data to a cloud service provider with fewer security controls, which may result in additional cost savings. If the customer decides to outsource the directly identifiable data to a different cloud service provider, she has to ensure that these two providers do not cooperate, e.g., by using the same IaaS provider in the backend 8.3.2 Cryptographic Data Splitting and Homomorphic Encryption.

As of today, this approach appears to be the most viable alternative, both from the technical and economical point of view. State-of-the-art encryption of data with adequate key management is one of the most effective means to safeguard privacy and confidentiality when outsourcing data to a cloud service provider. Nevertheless, at least in the European Union, encryption is not considered to relieve cloud customers from all of their responsibilities and legal obligations. Encrypted data keep the nature it has in its decrypted state; personally identifiable information in encrypted form is still regarded as personally identifiable information (see [57]). Encryption is considered as an important technical security measure; however, some additional mandatory legal safeguards still apply. For personally identifiable data, this means that, e.g., adequate contracts for the export of data to countries outside of the European Economic Area have to be in place.

IX. ASSESSMENT OF MULTICLOUD ARCHITECTURES

Given the vast amount of specific approaches for realizing each of the presented multicloud architectures, it is not feasible to perform a general assessment adequately covering all of them. Furthermore, many approaches are only suitable in very special circumstances, rendering each comparison to other approaches of the same domain inadequate. However, in this section we perform a high-level assessment of all multicloud approaches presented above, focusing on their capabilities in terms of security, feasibility, and compliance, as shown in Fig. 5. Therein, the security considerations indicate an approach's general improvements and aggravations in terms of integrity, confidentiality, and availability of application logic or data, respectively. For instance, the n clouds approach is highly beneficial in terms of integrity (every deviation in execution that occurs at a single cloud provider only can immediately be detected and corrected), but quite disadvantageous in terms of confidentiality The feasibility aspect covers issues of applicability, business readyness, and ease of use. Herein, applicability means the degree of flexibility of using one approach to solve different types of problems. Business-readyness evaluates how far the research on a multicloud approach has progressed and if it is ready for real-world applications, whereas ease of use indicates the complexity of implementing the particular approach. As an example, the approaches of secure multiparty computation may be of high benefits in terms of security, but only solve a very specific type of computation problem (i.e., are limited in applicability), and are quite complex to implement (i.e., not easy to use) even if they can be applied reasonably. The compliance dimension provides a high-level indication of the impact of each approach to the legal obligations implied to the cloud customer when utilizing that approach. Application of the dual execution approach, for instance, may be favorable in terms of security and feasibility, but requires complex contractual negotiations between the cloud customer and two different cloud providers, doubling the workload and legal obligations for the whole cloud application. Equivalently, the use of more than two different cloud providers (n clouds approach) improves on integrity and availability, but also requires n contract negotiations and risk assessments, amplified by the necessity to assess the risks associated with automated detection and correction of irregularities within the n parallel executions. Based on the observations subsumed in Fig. 5, we can conclude that there is no such thing as a "best" approach. From a technical point of view, the use of multiple cloud providers leads to a perceived advantage in terms of security, based on the perception of shared—and thus mitigated—risks. From a compliance point of view, however, many of these advantages do not sustain, and may even lead to additional legal obligations—and hence to higher risks. The few approaches that would be beneficial in terms of both security and compliance tend to be quite limited in feasibility of application, and are not business ready yet or rather nontrivial to use in real-world settings.

X. CONCLUSION

The use of multiple cloud providers for gaining security and privacy benefits is nontrivial. As the approaches investigated in this paper clearly show, there is no single optimal approach to foster both security and legal compliance

in an omniapplicable manner. Moreover, the approaches that are favorable from a technical perspective appear less appealing from a regulatory point of view, and vice versa. The few approaches that score sufficiently in both these dimensions lack versatility and ease of use, hence can be used in very rare circumstances only. As can be seen from the discussions of the four major multicloud approaches, each of them has its pitfalls and weak spots, either in terms of security guarantees, in terms of compliance to legal obligations, or in terms of feasibility. Given that every type of multicloud approach falls into one of these four categories, this implies a state of the art that is somewhat dissatisfying. However, two major indications for improvement can be taken from the examinations performed in this paper. First of all, given that for each type of security problem there exists at least one technical solution approach, a highly interesting field for future research lies in combining the approaches presented here. For instance, using the n clouds approach (and its integrity guarantees) in combination with sound data encryption (and its confidentiality guarantees) may result in approaches that suffice for both technical and regulatory requirements. We explicitly do not investigate this field here—due to space restrictions; however, we encourage the research community to explore these combinations, and assess their capabilities in terms of the given evaluation dimensions. Second, we identified the fields of homomorphic encryption and secure multiparty computation protocols to be highly promising in terms of both technical security and regulatory compliance. As of now, the limitations of these approaches only stem from their narrow applicability and high complexity in use. However, given their excellent properties in terms of security and compliance in multicloud architectures, we envision these fields to become the major building blocks for future generations of the multicloud computing paradigm.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing, Version 15," Nat'l Inst. Of Standards and Technology, Information Technology Laboratory, vol. 53, p. 50, http://csrc.nist.gov/groups/ SNS/cloud-computing/, 2010.

[2] F. Gens, "IT Cloud Services User Survey, pt.2: Top Benefits & Challenges," blog, http://blogs.idc.com/ie/?p=210, 2008.

[3] Gartner, "Gartner Says Cloud Adoption in Europe Will Trail U.S. by at Least Two Years," http://www.gartner.com/it/page. Jsp? Id=2032215, May 2012.

[4] J.-M. Bohli, M. Jensen, N. Gruschka, J. Schwenk, and L.L.L. Iacono, "Security Prospects through Cloud Computing by Adopting Multiple Clouds," Proc. IEEE Fourth Int'l Conf. Cloud Computing (CLOUD), 2011.

[5] D. Hubbard and M. Sutton, "Top Threats to Cloud Comput- in g V 1.0," C lo ud Sec u rity an l lia NC e, h ttp: / / w w w. cloudsecurityalliance.org/topthreats, 2010.

[6] M. Jensen, J. Schwenk, N. Gruschka and L. Lo Iacono, "On Technical Security Issues in Cloud Computing," Proc. IEEE Int'l Conf. Cloud Computing (CLOUD-II), 2009.

[7] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third- Party Compute Clouds," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 199-212, 2009.

[8] Y. Zhang, A. Juels, M.K.M. Reiter, and T. Ristenpart, "Cross-VM Side Channels and Their Use to Extract Private Keys," Proc. ACM Conf. Computer and Comm. Security (CCS '12), pp. 305-316, 2012.

[9] N. Gruschka and L. Lo Iacono, "Vulnerable Cloud: SOAP Message Security Validation Revisited," Proc. IEEE Int'l Conf. Web Services (ICWS '09), 2009.

[10] M. McIntosh and P. Austel, "XML Signature Element Wrapping Attacks and Countermeasures," Proc. Workshop Secure Web Services, pp. 20-27, 2005.

[11] J. Kincaid, "Google Privacy Blunder Shares Your Docs without Permission," TechCrunch, http://techcrunch.com/2009/03/07/ huge-google-privacy - blunder- shares-your-doc s -witho ut- permission/, 2009.

[12] J. Somorovsky, M. Heiderich, M. Jensen, J. Schwenk, N.Gruschka, and L. Lo Iacono, "All Your Clouds Are Belong to Us: Security Analysis of Cloud Management Interfaces," Proc. Third ACM Workshop Cloud Computing Security Workshop (CCSW '11), pp. 3-14, 2011.

[13] S. Bugiel, S. Nu¨ rnberger, T. Po¨ ppelmann, A.-R. Sadeghi, and T. Schneider, "AmazonIA: When Elasticity Snaps Back," Proc. 18th ACM Conf. Computer and Comm. Security (CCS '11), pp. 389-400, 2011.

[14] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond,

and M. Morrow, "Blueprint for the Intercloud—Protocols and Formats for Cloud Computing Interoperability," Proc. Int'l Conf. Internet and Web Applications and Services, pp. 328-336, 2009.

[15] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to Enhance Cloud Architectures to Enable Cross-Federation," Proc. IEEE Third Int'l Conf. Cloud Computing (CLOUD), pp. 337-345, 2010

[16] R. Turpin and B.A. Coan, "Extending Binary Byzantine Agree ment to Multivalued Byzantine Agreement," Information Proces- sing Letters, vol. 18, no. 2, pp. 73-76, 1984.

[17] I. Koren and C.M.C. Krishna, Fault-Tolerant Systems. Morgan Kaufmann, 2007.

[18] J.D.J. Wisner, G.K.G. Leong and K.-C. Tan, Principles of Supply Chain Management: A Balanced Approach. South-Western, 2011.

[19] N.A.N. Lynch, Distributed Algorithms. Morgan Kaufmann, 1996.

[20] G. Danezis and B. Livshits, "Towards Ensuring Client-Side Computational Integrity (Position Paper)," Proc. ACM Cloud Computing Security Workshop (CCSW '11), pp. 125-130, 2011.

[21] S. Groß and A. Schill, "Towards User Centric Data Governance and Control in the Cloud," Proc. IFIP WG 11.4 Int'l Conf. Open Problems in Network Security (iNetSeC), pp. 132-144, 2011.

[22] R. Rivest, L. Adleman, and M. Dertouzos, "On Data Banks and Privacy Homomorphisms," Foundations of Secure Computation, vol. 4, no. 11, pp. 169-180, 1978.

[23] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.

[24] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," Proc. 17th Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT '99), pp. 223-238, 1999.

[25] C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., 2009.

[26] G. Asharov, A. Jain, A. Lo´ pez-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs, "Multiparty Computation with Low Communication, Computation and Interaction via Threshold Fhe," Proc. 31st Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '12), pp. 483-501, 2012.

[27] Y. Desmedt, "Some Recent Research Aspects of Threshold Cryptography," Proc. First Int'l Information Security Workshop, pp. 158-173, 1998.

[28] A.C.A. Yao, "Protocols for Secure Computations," Proc. IEEE 23rd Ann. Symp. Foundations of Computer Science (FOCS '82), pp. 160- 164, 1982.

[29] M. Ben-Or, S. Goldwasser, and A. Wigderson, Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation," Proc. 20th Ann. ACM Symp. Theory of Computing (STOC '88), pp. 1-10, 1988.

[30] O. Goldreich, S.M.S. Micali, and A. Wigderson, "How to Play Any Mental Game," Proc. 19th Ann. ACM Symp. Theory of Computation (STOC '87), pp. 218-229, 1987

[31] I. Damga˚ rd, M. Geisler, M. Krøigaard and J.B.J. Nielsen, "Asynchronous Multiparty Computation: Theory and Implemen- tation," Proc. 12th Int'l Conf. Practice and Theory Public Key Cryptography (PKC '09), pp. 160-179, 2009.

[32] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos, "SEPIA: Privacy-Preserving and Statistics," Proc. USENIX Security Symp, pp. 223-240, 2010

[33] S. Bugiel, S. Nu¨ rnberger, A.-R. Sadeghi, and T. Schneider, "Twin Clouds: Secure Cloud Computing with Low Latency," Proc. 12th IFIP TC 6/TC 11 Int'l Conf. Comm. and Multimedia Security (CMS'11), pp. 32-44, 2011.

[34] P. Bogetoft, D.L.D. Christensen, I. Damgard, M. Geisler, T.P.T. Jakobsen, M. Kroigaard, J.D.J. Nielsen, J.B.J. Nielsen, K. Nielsen, J. Pagter, M.I.M. Schwartzbach, and T. Toft, "Secure Multiparty Computation Goes Live," Financial Cryptography and Data Security, R. Dingledine and P. Golle, eds., pp. 325-343, Springer-Verlag, 2009.

[35] "DEMONS Deliverable D2.4: Preliminary Implementation of the Privacy Preservation Techniques,"

Giuseppe Bianchi, eds., et al., DEMONS Deliverable D2.4, 2011.

[36] J.-M. Bohli, W. Li, and J. Seedorf, "Assisting Server for Secure Multi-Party Computation," Proc. Sixth IFIP WG 11.2 Int'l Conf. Information Security Theory and Practice: Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems (WISTP '12), pp. 144-159, 2012.

[37] O. Catrina and F. Kerschbaum, "Fostering the Uptake of Secure Multiparty Computation in E-Commerce," Proc. IEEE Third Int'l Conf. Availability, Reliability and Security (ARES '08), pp. 693-700, 2008.

[38] F. Pagano and D. Pagano, "Using In-Memory Encrypted Data- bases on the Cloud," Proc. First Int'l Workshop Securing Services on the Cloud (IWSSC), pp. 30-37, 2011.

[39] J. Somorovsky, C. Meyer, T. Tran, M. Sbeiti, J. Schwenk, and C. Wietfeld, "SeC2: Secure Mobile Solution for Distributed Public Cloud Storages," Proc. Second Int'l Conf. Cloud Computing and Services Science (CLOSER), pp. 555-561, 2012.

[40] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptography and Data Security, pp. 136- 149, 2010

[41] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Con- structions," Proc. 13th ACM Conf. Computer and Comm. Security, pp. 79-88, 2006.

[42] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable Encryption Revisited: Consistency Properties, Relation to Anon- ymous IBE, and Extensions," Proc. 25th Ann. Int'l Conf. Advances in Cryptology (CRYPTO '05), pp. 205-222, 2005

[43] R. Popa, C. Redfield, N. Zeldovich and H. Balakrishnan, "CryptDB: Protecting Confidentiality with Encrypted Query Processing," Proc. 23rd ACM Symp. Operating Systems Principles, pp. 85-100, 2011.

[44] A. Boldyreva, N. Chenette, Y. Lee, and A. Oneill, "Order- Preserving Symmetric Encryption," Proc. 28th Ann. Int'l Conf. Advances in Cryptology: The Theory and Applications of Cryptology (EUROCRYPT '09), pp. 224-241, 2009.

[45] J. Vijayan, "Vendors Tap into Cloud Security Concerns withNew Encryption Tools," http://www.cio.com.au/article/376252/vendors_t a p_ into_clo ud_secur ity_ concerns_new_ encryption_tools/, 2013.

[46] L. Wiese, "Horizontal Fragmentation for Data Outsourcing with Formula-Based Confidentiality Constraints," Proc. Fifth Int'l Work- shop Security (IWSEC '10), pp. 101-116, 2010.

[47] H. Rajasekaran, L. Lo Iacono, P. Hasselmeyer, J. Fingberg, P.Summers, S. Benkner, G. Engelbrecht, A. Arbona, A. Chiarini, C.M.C. Friedrich, M. Hofmann-Apitius, K. Kumpf, B. Moore, P.Bijlenga, J. Iavindrasana, H. Mueller, R.D.R. Hose, R. Dunlop, and A. Frangi, "@neurist—Towards a System Architecture for Advanced Disease Management through Integration of Hetero- geneous Data, Computing, and Complex Processing Services," Proc. IEEE 21st Int'l Symp. Computer-Based Medical Systems (CBMS '08,) pp. 361-366, 2008.

[48] European Commission, "Proposal for a Regulation of the European Parliament and of the Council on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data (General Data Protection Regulation),"http://ec.europa.eu/justice/data-protection/ document/review2012/com_2012_11_en.pdf, 2012.

[49] US Congress, "U.S. Health Insurance Portability and Account- ability Act," 1996.

[50] PCI Security Standards Council, "Payment Card Industry (PCI) Data Security Standard - Requirements and Security Assess- ment Procedures," https://www.pcisecuritystandards.org/ documents/pci_dss_v2.pdf, 2010.

[51] US Congress, "Federal Information Security Management Act," http://csrc.nist.gov/drivers/documents/FISMA-final.pdf, 2002.

[52] US General Services Administration, "Federal Risk and Author- ization Management Program," http://www.gsa.gov/portal/ category/102371, 2012.

[53] European Union, "Directive 95/46/EC on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data," http://eur-lex.europa.eu/

LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML,1995.

[54] European Commission, "Commission Decision of 5 February 2010 on Standard Contractual Clauses for the Transfer of Personal Data to Processors Established in Third Countries under Directive 95/46/EC of the European Parliament and of the Council," Official J.European Union, vol. L39, pp. 5-18, 2010.

[55] EU Article 29 Working Party, "Standard Application for Approval of Binding Corporate Rules for the Transfer of Personal Data," Recommendation 1/2007, WP 133, http:// ec.europa.eu/justice/data-protection/article-29/documentation/ opinion-recommendation/index_en.htm, 2007.

[56] Fed. Republic of Germany, "German Federal Data Protection Act