Department: Internet of Things, People, and Processes
Editor: Schahram Dustdar, dustdar@dsg.tuwien.ac.at

COM-PACE: Compliance-Aware Cloud Application Engineering Using Blockchain

Gagangeet Singh Aujla
Newcastle University

Masoud Barati and Omer Rana
Cardiff University

Schahram Dustdar
Technische Universität Wien

Ayman Noor
Newcastle University
Taibah University

Jose Tomas Llanos and Madeline Carr
University College London

Davit Marikyan and Savvas Papagiannidis
Newcastle University Business School

Rajiv Ranjan
Newcastle University

***Abstract*—The COVID19 Pandemic has highlighted our dependence on online services (from government, e-commerce/retail, and entertainment), often hosted over external cloud computing infrastructure. The users of these services interact with a web interface rather than the larger distributed service provisioning chain that can involve an interlinked group of providers. The data and identity of users are often provided to service provider who may share it (or have automatic sharing agreement) with backend services (such as advertising and analytics). We propose the development of compliance-aware cloud application engineering, which is able to improve transparency of personal data**

Digital Object Identifier 10.1109/MIC.2020.3014484
Date of current version 26 October 2020.

use – particularly with reference to the European GDPR regulation. Key compliance operations and the perceived implementation challenges for the realization of these operations in current cloud infrastructure are outlined.

With the increasing demand of externally hosted services (from government, finance, e-commerce/retail, and entertainment), often hosted over cloud computing infrastructure, there is a realization that online electronic services can involve an interlinked set of providers. Users of these services only interact with a Web interface rather than the larger, distributed service ecosystem that is often hidden behind the user interface. Users often endow (or entrust) their data and identity without realizing that the service provider may share their data (or a subset of it) with several backend services (e.g., cloud-hosted analytics, user profiling, and advertising services). To overcome this, the General Data Protection Regulation (GDPR) was proposed to ensure that nonexpert users can make knowledgeable decisions about their privacy and, thereby, give "informed consent" to use, store, share, and reprocess their personal data. However, there are several challenges that need to be address to realize this, both for individuals (data owners) who need to provide consent and for data controllers who need to obtain it.

One of the major challenges in the aforementioned context is the confusion of three terms: *Security*, *Privacy*, and *Compliance*. In general, these terms are interrelated but have distinct semantics that make them different in practical use. *Security* refers to the freedom (resilience) from potential harm or damage (such as disruption or misdirection of services) caused by others (such as attackers, malware, etc.). *Privacy* relates to any entity or information that is secluded from an individual or a group. It is linked directly to the sensitivity of data as any information that is private to an individual tends to be sensitive. *Compliance* refers to an *act of obeying*, i.e., any conduct that is based or bounded on (by) a specific rule, policy, order, or request. In other words, compliance signifies the conformance to a rule or guidelines (like a standard, legislation, or law). Compliance is based on different principles, for example, the GDPR legislation sets out seven key principles: Lawfulness, fairness and transparency, purpose limitation, data minimization, accuracy, storage limitation, integrity and confidentiality (security), and accountability.

Commercial cloud providers, such as Amazon Web Services, Google Cloud, and Microsoft Azure, provide limited or no support (restricted to security and limited privacy) for compliance adherence (specifically for GDPR). These cloud providers offer ready to use stacks (Serverless computing, Function-as-a-Service, CloudFormation, etc.) but their key focus remains on elastic provisioning of infrastructure rather than compliance. For instance, Function-as-a-Service provides a serverless platform (AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, etc.) where users can deploy, run, and manage their applications without worrying about infrastructure complexity and management (which is handled by the cloud provider). Similarly, AWS CloudFormation provides a configurable platform using the CloudFormation template, support for testing it locally or at Amazon S3, the use of APIs/ browser console/ command line tools. At infrastructure level, the cloud providers are ameliorating horizontally as well as vertically with respect to speed, scale, and quality of service. However, the threat of data breaches or loss of sensitive/personal data still causes concern among organizations that make use of cloud services, and users while using cloud infrastructure. A summary of potential data breaches that can occur when using cloud services, and mechanisms for auditing this capability is provided by Rahulamathavan *et al.*[1]

Although cloud providers continue to provide mitigation strategies to limit/avoid data loss, significant challenges still remain, especially when multiple providers need to work together. The liabilities of any unauthorized access or usage of personal data can have a significant impact on cloud provider revenue, and more importantly cloud provider market reputation (and perception). The GDPR requires organizations to report any data breaches within 72 h and hefty fines (4%

of annual global turnover or 20 million Euros) are applicable.[2] For example, Facebook had to pay $5 billion due to *2018 Cambridge Analytica scandal* wherein 50 million profiles were accessed to target advertisement during the 2016 presidential election campaign.* Another example is where British Airways and its parent International Airlines Group were fined US $230 million in connection with a data breach that took place in 2019—affecting 500K customers.**

One of the biggest questions for cloud providers is to understand the sensitivity of the data entrusted upon them by users. Another problem lies with the varied compliance guidelines across different geographic locations. This makes it hard to understand any applicable laws and monitor data flows across different geographic boundaries. Users often entrust their privacy to an organization, but when the organization relies on different cloud providers located across different locations, then different privacy and compliance conditions arise. Here lies the risk of data leakage. This raises a number of questions, which are as follows.

- *What data is labeled/ classified as "personal data" by a cloud provider*?
- *How do we monitor compliance in a useful and effective way*?
- *How are terms identified in data privacy regulations related to the monitoring of compliance*?
- *How do we verify compliance and then ensure the "right to be informed" clause through enforcement*?
- *How do we develop a shared agreement for compliance provisioning, monitoring, and verification between a cloud user and provider*?

COM-PACE ARCHITECTURE FOR HOSTING SERVICES IN A MULTICLOUD ENVIRONMENT

To address the aforementioned questions, we describe a specific compliance aware approach in the form of *compliance-aware cloud application engineering* (COM-PACE). Figure 1 shows the overview of COM-PACE architecture in a distributed service ecosystem. To help understand this landscape, we characterize COM-PACE in a multilayered compliance-aware service stack [see Figure 1(a)]. This architecture is realized over a traditional Infrastructure-as-a-Service (IaaS) layer composed of multiple CSPs, which are managed through virtualization [using virtual machines (VM) or containers]. This article focuses on embedded compliance checking and verification through the introduction of the COM-PACE layer.

The application architecture decides how, when, and which compliance operations should be executed in the cloud. The deployment of such compliance-aware architecture is challenging as cloud application composition involves dependencies among the heterogeneous resources (software, hardware, and VM/container). Figure 1(b) depicts the high-level architecture of a compliance-aware application comprising different software resource layers, including data layer, application logic, and compliance engine. Here, the different compliance operations are programmed to coordinate and control the application and compliance resources (at run time and design time) required to support compliance enforcement. To follow our COM-PACE architecture, cloud application developers and deployment teams have to follow three programming steps and operations [shown in Figure 1(c)] discussed in the following.

Provisioning Compliance (at Design and Runtime): User data access requirements are analyzed alongside the organizations software resources to realize the compliance requirements according to the applicable data protection regulations. After this, the compatible hardware resources are selected for instantiating the compliant trust services and configuring them to handle the interoperability and communication with other software resources in the multitier web application. The amalgamation of compliance server or manager with the database server can be seen in Figure 1(b).

Monitoring Compliance (Runtime): The monitoring of operations performed on data helps track the events that can be checked for GDPR compliance or violation (such as data leakage, profiling, or advertising). Event information

*[Online]. Available: https://eu.usatoday.com/story/tech/news/2019/07/24/facebook-pay-record-5-billion-fine-u-s-privacy-violations/1812499001

**[Online]. Available: https://techcrunch.com/2019/07/08/uks-ico-fines-british-airways-a-record-183m-over-gdpr-breach-that-leaked-data-from-500000-users

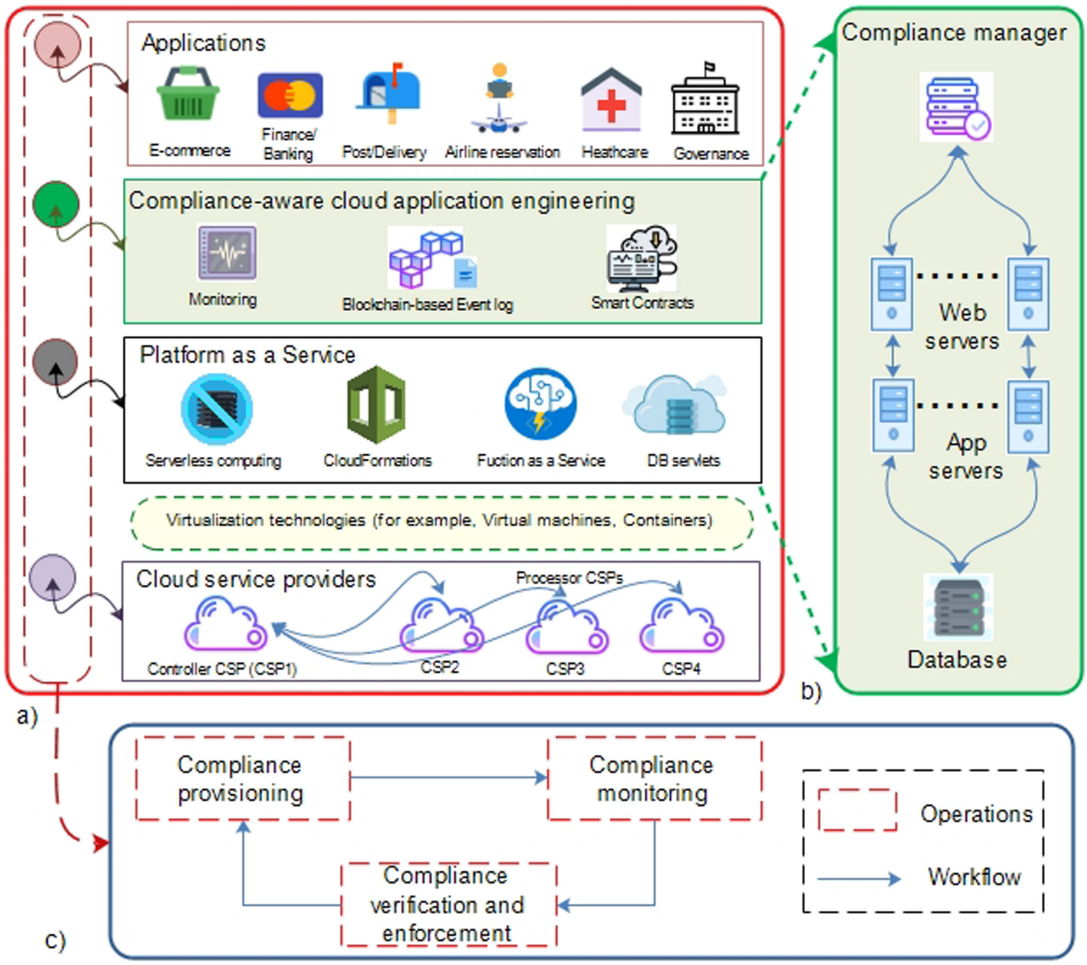

Figure 1. Overview of compliance-aware cloud application engineering (COM-PACE). (a) Interlinked compliance-aware service stack that provides a layered architecture utilizing different attributes and granularity. (b) Conceptual architecture associated with a cloud application engineering methodology comprising of a compliance manager, web servers, application servers, and database server(s). (c) Abstract view of the life cycle of compliance engineering operations (CSP: Cloud service provider).

generated by resources deployed for running a cloud application (such as user location) can be used to understand the applicable data protection rules and relate it to the possible violation event recorded by the monitoring engine. A monitoring executor can initiate the data operations recording and subsequent submission to a blockchain network for auditing purposes.

Verifying and Enforcing Compliance (runtime): Based on the compliance contract, the blockchain manager verifies compliance through the query executor that analyzes the behavior of the events in accordance with the GDPR. Upon verification, an event (such as disclosure or processing of personal data for advertising, etc.) can trigger the violation alert, and thereafter, the data controller can initiate the corrective actions (such as reporting the violation to the authorities) without disturbing the runtime system.

Compliance Provisioning

Cloud SDKs provide a contemporary way of hosting web application components and provisioning data and services. However, the current APIs available for handling risk, governance, and compliance in these Cloud SDKs are not fully capable of provisioning compliance to the extent required. Some of the current cloud APIs (such

as Cloud Elements,[3] AWS compliance programs,[4] and RedLock[5]) enable secure access to user assets (or personal data), but without adhering or with limited adherence to the GDPR compliance principles. For example, Cloud Elements[3] is an API integration platform as a service (PaaS) hosted on Amazon AWS that provides stringent security provisions and practices, but without any GDPR compliance control and enforcement. Moreover, Cloud Elements utilize and integrates different security solutions (for file integrity, multifactor authentication, etc.), but it fails to provide a multilevel-trusted environment for compliance provisioning. In such a solution, even if stringent security practices are adopted, the data usage audit trail and the operations performed on the data still cannot be tracked.

Microsoft Azure provides built-in compliance tools, but once the services have been provisioned, the responsibility of operating the security and privacy policies rests on the user (who often is not fully trained to do so).[6] Amazon also provides a shared responsibility-based compliance cloud APIs under which the role of management, operation, and verification of security and privacy policies lies with the user, and AWS handles systems operational management.[7] For example, in Amazon EC2 service, the security configuration and management (including the configuration of the AWS-provided firewall) is handled by the customer or user when they deploy an Amazon EC2 instance. For other services such as Amazon S3 and Amazon DynamoDB, a customer is responsible for data management (including encryption), asset classification, and setting of relevant permissions.[4] Other vendors, e.g., AlertLogic also utilize this shared responsibility approach. Palo Alto Networks' RedLock service provides automatic redress and compliance reporting, along with ease of control in a multipublic cloud.[5] RedLock uses the APIs of the major public CSPs (such as Amazon Web Services, Microsoft Azure, and Google Cloud) to provide an agentless multipublic Cloud PaaS–IaaS security environment for the handling of sensitive data. However, it is involved only at network level and not at the application level.

For a fully compliance-aware environment, the need for end-to-end compliance provisioning is the first step to act upon. The current cloud APIs should be normalized to connect to many endpoints (such as Overleaf, Dropbox, etc.). Compliance provisioning should enable the connection with the endpoint, and thereafter, the data can be streamed directly to a user's application. During this process, any pass-through data from services should not be stored and the entire end-to-end transmission should take place through HTTPS. Finally, the data stored at the endpoints are encrypted using encryption (e.g., 256-b AES). Compliance provisioning acts as the backbone for delivering security and privacy solutions throughout the application life cycle. It should ensure the continuous scanning of the whole of the application and components (build or purchased), even covering all the frameworks, application types, and so on.

Compliance Monitoring

Compliance monitoring is mainly responsible for affirming that an amenable framework is being adhered to as a watchdog for unwarranted operations or events, and acts as an autonomous process, operating in the second line of defence.[8] The base of compliance monitoring can be assumed from the process execution events that follow up. Most common frameworks for monitoring and tracking complex and significant events and submitting them to the monitoring tier can be accumulated from real low-level execution events.[9] The monitoring tier ensures observance of the adhered rules. With regard to the stakeholder's requirements, the visualization of results is carried out in the reporting tier, which in turn, gains input from the monitoring tier.

Currently, various monitoring frameworks (such as docker stat,[†] cAdvisor,[‡] DataDog,[¶] Amazon cloud watch,[¶¶] and CLAMS[10]) are available to observe the applications running in the cloud. However, these monitoring frameworks are either cloud-provider-oriented (Microsoft Azure Fabric Controller) or virtualization-architecture-oriented (cAdvisor)[11] and, hence, fail to meet the monitoring requirements in a complex multiple cloud environment. Several studies[10,11] have

[†][Online]. Available: https://www.docker.com
[‡][Online]. Available: https://github.com/google/cadvisor
[¶][Online]. Available: https://www.datadoghq.com
[¶¶][Online]. Available: https://aws.amazon.com

been carried out, all of which focusing on comprehensive performance-based monitoring in the cloud. Yet, to be GDPR compliance-specific, the exact data processing event during individual stages of the process cannot be portrayed by the overall metrics. There remains a dearth of GDPR metrics and an acute need for an intelligent monitoring framework. An extensive investigation of both the real-time monitoring overhead and the framework scalability is also required. Hence, the monitoring of "what" and "how" appears to be the primary challenge of multicloud event monitoring framework where the event logs have to be stored in a blockchain.

To address these challenges, we put forth a proposal for an extensive real-time compliance monitoring framework that can be used to monitor the processing of personal data (in line with GDPR) in multicloud systems. Using the daemon process and log analysis, the data operations are obtained by the framework in real time. The dimensions of compliance monitoring are elaborated ahead.

Monitoring Granularity A wide variety of technologies (VMs and Docker containers) have been used to increase the stack of elements that must be managed for application creation, including the use of containers to run the software, Web servers, or big-data processing. Although typically only hardware and software structuring components (servers, databases, or proxies) need be controlled, monitoring at the lower rates, i.e., cloud systems, microservices, and APIs, even internally used methods or functions, is increasingly needed. The purpose of the COM-PACE architecture is to provide an automated management framework for the different layers used in applications decomposed in microservice architectures and container clusters. Not only on a single host but in many container clusters can a container-based application be deployed. There are several nodes for each container cluster (hosts), and there are several containers for each node. Output management data can be obtained from various code layers (e.g., node layer, database layer, and server layer) for applications implemented in container-based environments. The system seeks to mitigate that issue by introducing a multilayer monitoring system for applications that are broken up in and deployed in containers in the multicloud environment (hardware metrics from operational monitoring, software metrics by monitoring server processes and database processes, as well as internal metrics such as method latency or processing rates of an API call).

Monitoring Topology The event capturing defined by the metrics is performed through monitoring agents, condensed in the conceptual representation of SmartAgent (SA) or GDPRAgent (GA). These are deployed at each CSP container and are responsible for monitoring all the events taking place in the relevant container. In the COM-PACE architecture, we have considered four types of agents connected to SA/GA. The agents in charge of monitoring the metrics related to READ, WRITE, TRANSFER, and PROFILE operations are considered in the COM-PACE architecture. Here, the biggest concern is related to the selection of an appropriate monitoring agent in the considered topology to make sure the compliance-related events are captured successfully. Another concern involves the selection of suitable compliance-related events to be monitored across the entire topology based on a tradeoff between monitoring overhead, response time, scalability, and compliance coverage.

Monitoring Challenges The challenges for monitoring frameworks in multicloud environments are discussed ahead.

Volume of Events and Alert Overload: The effect of the increasing resources of microservices, cloud, and virtualization on the monitoring needs is not always determined beforehand by a significant number of organizations, and consequently, segregating the unexpected from the expected can be difficult for containers, which are ephemeral and dynamic. Conventional monitoring systems may have difficulty in coping, or may run out of event log resource allocation due to an overwhelming increase in event activity caused by the easy scaling of resources by application owners. Tackling the problem with tooling expansion involves an incoherence, as the concept of legacy environment is not familiar with modern tools, while containerized

environments are not familiar with conventional monitoring systems.

Monitoring Rule Complexity: The monitoring of operations can become arduous as complex changes in the environment are expected to be notified to users by the tools. Disregarding many probable real events and consideration of false alerts by the monitoring agents can cause problems. This kind of situation occurs mostly when a lot of alerts arise due to imprecise rules and large volume, which makes monitoring by monitoring agents difficult.

Architecture of Complex Systems: Problems arise for monitoring systems due to systems and networks with varying degrees of trust, as communication is necessary within and beyond the boundaries of trust. Such problems may be solved by using a unique trusted configuration for the monitoring system. Additionally, several monitoring systems can also be used to communicate with each other to compare the information attained. The installation of such a monitoring system depends on the level of sensitive data that is being dealt with, and also on the engineers who install the monitoring system in a controlled environment.

Auditing Issue in Clouds: The strategies for auditing and monitoring the fulfillment of compliance duties are not integrated into current public offerings of IaaS. In case of deployment of the application in a public cloud, it is difficult to follow the specifications of storage and location-based processing since the cloud properties maintain the underlying details abstractly. The deployment of business applications is no longer possible due to the inability to fulfill compliance requirements as the cloud cannot be monitored. This inability to satisfy the terms of compliance may cause large fines or cancellation of business licenses. Hence, IP administration and VM scale operations should be issued with auditing logs and monitoring facilities by IaaS clouds. To ensure the fulfillment of the audit logs, stern logging conditions are required. Using resources of the cloud in a traceless way should not be allowed even when the administrative account is used for logging in. Observance by the IaaS of the service provider constraints should also be aimed at. Any IaaS should act in accordance with regulatory restrictions and legal requirements, and this should be ensured by the service providers.

Application Migration in Multiclouds: Another problem that arises is related to the active organization of the containers or applications of the CSPs. Fixing the procedure and timing for the migration of applications from different CSPs and determining the properties that impact the migrations are a few examples to be noted. Application parts are managed and deployed in different ways, as different CSPs deal with the process. It is difficult to manage the application components overall as a whole entity as the CSPs maintain a heterogeneous nature. A major part is played by monitoring in recognizing the timing of migration of particular applications or containers.

Compliance Verification and Enforcement

Here, the data (event) logs can be queried to verify and enforce compliance using the blockchain and smart contracts. The smart contracts are used to digitally verify or enforce compliance, as required by the contract. This helps verify credible transactions without third-party intervention. There are two major challenges: first, to select an appropriate blockchain platform, and second, to select the events that should be queried or verified to avoid additional overhead.

Figure 2 presents an abstract model for connecting the parties—including the user, CSPs, and the arbiter—to the blockchain in order to use the smart contracts supporting GDPR requirements. The model enables the audit trail of service providers that can have the roles of data controller or data processor. It makes use of the blockchain to record the operations (e.g., READ, COPY, etc.) carried out by providers on user data. Furthermore, the model checks whether or not the executed operations comply with the GDPR. The blockchain-based VM is an open blockchain platform (e.g., Ethereum VM) providing an environment for the parties to run smart contracts and create a blockchain network. The VM involves four smart contracts that provide the basis for the verification of providers following a set of GDPR obligations. The smart contracts are data purpose contract, confirmation

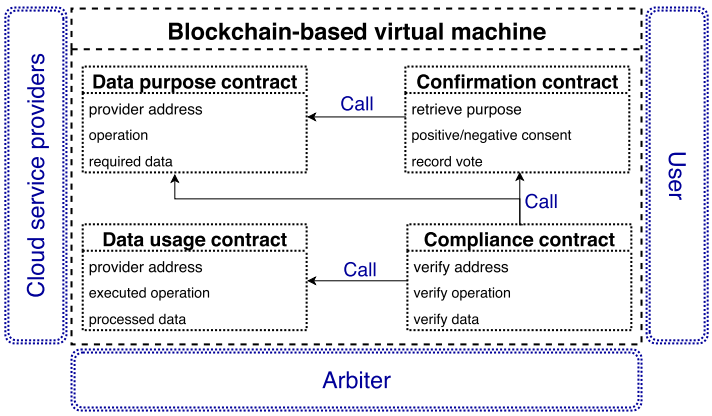

Figure 2. Proposed smart contracts.

contract, data usage contract, and compliance contract.

Data purpose contract captures the purpose of data processing of cloud providers. The purpose of data processing can be specified with several typical operations (i.e., READ, COPY, TRANSFER, PROFILING, etc.) carried out by controllers/ processors on personal data. The contract enables the providers to store their addresses (e.g., Ethereum accounts) and the operations that will be executed by them on user data in a blockchain. The activators of the contract's transactions are cloud providers. This contract gives effect to Arts. 5(1)(b) and 30(1)(a) and (b) of the GDPR, under which the purpose of data processing and the address of the service provider should be clarified in advance.

Confirmation contract enables users to confirm or deny consent for data processing recorded in the blockchain. The contract can contain two functions: one for retrieving records identifying data processing purposes declared by providers; another for sending the vote (accept/ reject) of users into a blockchain. The former permits users to verify the purpose of data processing before sharing their personal data with cloud providers. Through the latter function, a user can specify whether the execution of an operation on their personal data should be allowed or not. The activators of such functions are users. The contract is based on Art. 6(1)(a) of the GDPR legislation, under which the consent of the data subject (user) legitimizes (i.e., authorizes) the processing of his/her personal data.

Data usage contract records all operations of providers carried out on personal data and records this in a blockchain. Such operations are recorded by the container executing on the provider, and used to track the processes of the service provider on personal data. The contract can involve a function activated by the container to store the provider's address and the executed operation. This information is sent to the blockchain to enable future verification of operations executed by providers. The contract enables users to track and be aware of the history of data movement between cloud providers. Such a capability supplied by the contract aligns with Art. 15 of the GDPR legislation, which identifies a data subjects' right to be informed about what and where their personal data are processed.

Compliance contract verifies the blocks created by the aforementioned smart contracts to detect GDPR violations. The contract is deployed and executed by the arbiter entrusted third-party connected to the blockchain VM to report a cloud provider violating GDPR rules. The following verification is undertaken through the contact to automatically expose violators:

1) whether or not the addresses of the service providers recorded by the data usage contract conform to those recorded via the data purpose contract;
2) whether or not the operations of each service provider recorded by the data usage contract conform to those recorded via the data purpose contract;
3) whether or not the operations of each provider recorded by the data usage contract were already confirmed by the data subject.

CONCLUSION

Privacy and consent are two key requirements of the European GDPR regulation, aiming to improve the transparency and fair use of personal data in the cloud. For this reason, current cloud application engineering and implementation must be reconsidered, to ensure that user data is not shared without consent. COM-PACE can act as a potential enabler for such "compliance-aware" cloud engineering through the three operations: compliance provisioning, monitoring and

verification, suggested in this article. We describe how this can be realised through a service stack that can be combined with other operations made available alongside other application specific services. These programming operations have to handle several intertwined dependencies from heterogeneous resources, performance barriers, to usability (to meet particular GDPR compliance requirements).

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council funded project PACE: Privacy-Aware Cloud Ecosystems under Grant EP/R033293/1 and Grant EP/R033439/1.

REFERENCES

1. Y. Rahulamathavan, M. Rajarajan, O. F. Rana, M. S. K. Awan, P. Burnap, and S. K. Das, "Assessing data breach risk in cloud systems," in *Proc. 7th IEEE Int. Conf. Cloud Comput., Technol., Sci.*, Nov. 30–Dec. 3, 2015, pp. 363–370.
2. G. Strawbridge. "5 examples of security breaches in 2018," 2018. [Online]. Available: https://www.metacompliance.com/blog/5-examples-of-security-breaches-in- 2018/
3. "Cloud elements platform security and compliance," 2020. [Online]. Available: https://cloud-elements.com/security-compliance/
4. "Amazon web services: Risk and compliance," 2017. [Online]. Available: https://d0.awsstatic.com/whitepapers/compliance/AWS_Risk_and_Compliance_Whitepaper.pdf
5. "Continuous security and compliance for multi-cloud deployments by Palo Alto," Accessed: Jul. 2020. 2019. [Online]. Available: https://www.digitalmarketplace.service.gov.uk/g-cloud/services/89478468 4094117
6. R. Waggoner, "Achieving trust and compliance in the Microsoft Azure Cloud," 2017. [Online]. Available: http://blog.mycloudit.com/achieving-trust-and-compliance-in-the-microso ft-azure-cloud
7. "Shared responsibility model," Accessed: Jul. 2020. [Online]. Available: https://aws.amazon.com/compliance/shared-responsibility-model/
8. D. Loreti, F. Chesani, A. Ciampolini, and P. Mello, "A distributed approach to compliance monitoring of business process event streams," *Future Gener. Comput. Syst.*, vol. 82, pp. 104–118, 2018.
9. L. T. Ly, S. Rinderle-Ma, D. Knuplesch, and P. Dadam, "Monitoring business process compliance using compliance rule graphs," in *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*. Berlin, Germany: Springer, 2011, pp. 82–99.
10. K. Alhamazani *et al.*, "CLAMS: Cross-layer multi-cloud application monitoring-as-a-service framework," in *Proc. IEEE Int. Conf. Services Comput.*, 2014, pp. 283–290.
11. A. Noor *et al.*, "A framework for monitoring microservice-oriented cloud applications in heterogeneous virtualization environments," in *Proc. IEEE 12th Int. Conf. Cloud Comput.*, 2019, pp. 156–163.