Giving Customers Control over Their Data: Integrating a Policy Language into the Cloud

Jens Hiller[*], **Maël Kimmerlin**[†], **Max Plauth**[‡], **Seppo Heikkilä**[§],
Stefan Klauck[‡], Ville Lindfors[¶], Felix Eberhardt[‡], Dariusz Bursztynowski[‖],
Jesus Llorente Santos[†], Oliver Hohlfeld[*], Klaus Wehrle[*]
[*]Communication and Distributed Systems, RWTH Aachen University, Germany
[†]School of Electrical Engineering, Aalto University, Finland
[‡]Hasso Plattner Institute for Digital Engineering, University of Potsdam, Germany
[§]Helsinki Institute of Physics, CERN, Geneva, Switzerland
[¶]F-Secure Oyj, Finland [‖]Orange Polska S.A., Poland **equal contribution**
firstname.lastname@{comsys.rwth-aachen.de, aalto.fi, hpi.uni-potsdam.de, cern.ch, f-secure.com, orange.com}

***Abstract*—Cloud computing offers the potential to store, manage, and process data in highly available, scalable, and elastic environments. Yet, these environments still provide very limited and inflexible means for customers to control their data. For example, customers can neither specify security of inter-cloud communication bearing the risk of information leakage, nor comply with laws requiring data to be kept in the originating jurisdiction, nor control sharing of data with third parties on a fine-granular basis. This lack of control can hinder cloud adoption for data that falls under regulations. In this paper, we show in six use cases how cloud environments can be enriched with policy language support to give customers control over cloud data. Our use cases are based on realizing policy language support in all three cloud environment layers, i.e., IaaS, PaaS, and SaaS. Specifically, we present policy-aware resource management (with OpenStack) and dynamic network configuration. With CERN's big data storage and the in-memory database Hyrise, we show realization for storage and further exemplify policy-aware cloud processing by network function virtualization which enables Orange to offload customer home gateways to the cloud. Finally, we discuss benefits of policy support in F-Secure's Security Cloud. These use cases show the feasibility of realizing customer control with policy support in the cloud. Thus, our work enables customers with regulated data to tap cloud benefits and significantly broadens the market for cloud providers.**

I. Introduction

Cloud computing drastically changed the IT landscape by providing means to (rapidly) offload functionality to highly available, scalable, and elastic cloud environments. This offloaded functionality ranges from data storage and processing tasks up to complete applications (e.g., network function virtualization). The offered flexibility thus enables rapid prototyping of IT products and the adaptive scaling of IT resources.

Despite the offered benefits and the current wide adoption of cloud computing, its further growth is severely hindered by the limited control of customers over offloaded data. That is, it is currently not (always) transparent to cloud users where offloaded data is stored and processed. This is particularly challenging for federated cloud scenarios in which data is offloaded to one cloud but processed by multiple clouds in the background. This lack of control is highlighted in a survey by the Intel IT Center, where 78% of 800 IT professionals need to comply with regulations that affect cloud usage and 78% are concerned that public clouds cannot meet corresponding requirements [1]. As a consequence, 57% refrain from using the cloud and 55% especially reported *lack of control over data* among the three major security concerns regarding cloud usage [1]. Respective regulations affect personal data of customers but also apply to financial, communication, and governmental data [2]. The lack of control is caused by the use of cloud provider selected, static policies to specify data handling, enabling only limited control by cloud users. Consequently, cloud customers cannot sufficiently negotiate their own requirements and lack fine-grained, data specific control [3], [4]. Thus, enabling control over cloud data is a major challenge to tap cloud benefits for many business cases.

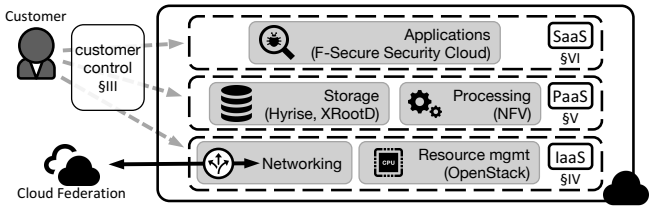

Figure 1. We put customers back into control of data that they offloaded to the cloud. Therefore, we show realization of policy support across all layers of the cloud (from IaaS up to SaaS) by enabling it for representative services.

Leveraging cloud benefits for regulated data requires enabling the negotiation of data handling requirements between cloud provider and customer. Demands for control range from location of storage and processing over guaranteed data deletion, up to enforcing communication or storage security levels. Realizing these demands is addressed by first academic attempts to design policy languages [3]–[11], each providing means to express data handling requirements to cloud providers. Yet, these languages vary in their expressiveness and lack experience in realizing concrete deployments.

In this paper, we close this gap by applying CPPL, a recent policy language that is specifically tailored to cloud scenarios [4], to realize a wide spectrum of industry-driven

Accepted at IC2E 2018, https://doi.org/10.1109/IC2E.2018.00050.

cloud use cases covering all layers of a cloud stack, i.e., Iaas, PaaS, and SaaS. We show an overview of our use cases in Figure 1, including realization of policy-aware IaaS with policy language support in OpenStack as state-of-the-art cloud resource management middleware and policy-controllable network connection configuration (§IV). We further realize policy language support in CERN's big data storage (XRootD), in Hyrise as major in-memory database system, and Orange Poland's telco NFV deployment to enable policy-aware PaaS (§V). Finally, we exemplify compliance with policies on the SaaS layer based on F-Secure's Security Cloud (§VI). Our use-case realizations show that the chosen policy language is widely applicable to cover a wide spectrum of use-cases, whose implementation requires minimal to medium changes to the target system. Beyond the state of the art of policy languages, we further introduce *policy decision points* to realize compliance with policies that limit the set of cloud servers that are allowed to receive data, e.g., due to location regulations (§III). The contributed policy decision points can be further used to realize policy-awareness in federated cloud scenarios, which is not supported by current policy languages. By showing how easily existing architectures can be made policy-aware, we aim to pave the way for bringing policy-aware cloud computing into practice.

II. Related Work

Regarding related work, we distinguish policy negotiation and realization of compliance with policies in cloud ecosystems.
Policy Negotiation. S4P [5] focuses on matching of user expectations and provider policies thereby neglecting performance requirements. XACML [6] is an XML based access control policy language. PPL [7] extends it with user expectations and the A4Cloud project adds accountability with A-PPL [8], [9]. However, large memory footprints of these XML-based policies increase overhead especially for fine-grained per data policies. A4Cloud also surveyed requirements and further tools for an accountable cloud [12], [13]. C^2L [10] limits its policies to control placement and migration of virtual machines and thus cannot cover all layers of the cloud. CES [11] focuses on end-to-end communication enabling transparent policy negotiations via broker gateways. FLAVOR [14] introduces the idea to specify actions in case of policy breaches. CPPL [4] is a recent policy language specifically tailored to the cloud. It thus provides a promising building block to realize customer control over cloud data. We further extend CPPL with policy decision points and use it to realize a wide spectrum of industry-driven use cases to contribute experience in realizing concrete deployments.
Policy Support. PRADA [3] realizes policies for the distributed storage system Cassandra. However, the mere focus on storage does not provide policy support for all cloud layers. CryptDB [15] realizes database queries on encrypted data to enable customers to offload a limited set of tasks to the cloud. Similarly, BLOOM [16] uses homomorphic encryption to securely offload search for specific genome sequences to the cloud. However, approaches that realize processing on encrypted data yet add substantial performance overheads and even those are limited to specific use cases as general processing on encrypted data is not yet feasible [16]. Furthermore, regulations such as restrictions on location or guaranteed deletion cannot be achieved with encryption and require further negotiation. Henze et al. survey possibilities privacy-aware cloud usage for handling data gathered by cyber-physical systems [17]. Betgé-Brezetz et al. [18] present policy support restricted to IaaS and limited to policies that do not require integration into services. We show strategies to realize efficient and comprehensive policy support across all cloud layers also including service specific policies. MIP [19] provides efficient and accurate real-time cloud security assessment and thus enables comparison of security levels of different cloud providers. To check actions taken by cloud services, Anisetti et al. propose an certification framework which they exemplify for OpenStack [20]. Alternatively, several approaches [21]–[24] employ trusted computing such as Intel SGX or ARM TrustZone to enable attestation of server behavior which cloud providers could employ to prove their adherence to negotiated policies.

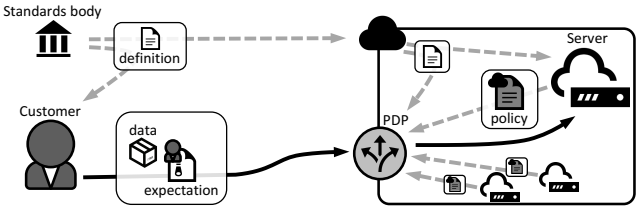

Figure 2. CPPL considers customer expectations and provider policies to derive instructions that enable handling of regulated data in the cloud. Policy decision points (PDPs) realize policies that limit suitable service end-points.

III. Negotiation of Customer Expectations

To realize compliance with customer policies, customers must be enabled to express their expectations to the cloud. To enable this negotiation, we adopt the recent ***C**ompact **P**rivacy **P**olicy **L**anguage* (CPPL) [4], which is specifically tailored for cloud use cases. CPPL (i) enables cloud customers to express their *expectations* on data handling towards the cloud. To incorporate cloud server abilities and *policies* of cloud providers, CPPL (ii) provides an automated process to match the customer expectations with cloud provider policies, i.e., checks if the cloud server is able and the provider is willing to adhere to the expectations. As result of this matching procedure, CPPL (iii) provides the cloud with concrete instructions for data handling, e.g., to delete data after three months.

The available expressions that make up customer expectations and provider policies are specified in *policy definitions*. Experts use domain knowledge to create and tailor policy definitions to specific domains. This enables CPPL to (i) express data handling requirements for various domains (expressibility) and even adapt to future, yet unforeseen data handling requirements and cloud services (extensibility). Furthermore, pre-distribution of policy definitions enables (ii) efficient compression of expectations to reduce costs for transfer and storage and, thus, enables fine-grained policies per data item or network packet. Finally, an efficient parsing and evaluation methodology based

on policy definition information enables (iii) fast matching of customer expectations and provider policies.

The general procedure for handling regulated data in the cloud is depicted in Figure 2. In an initial deployment step (gray dashed lines), customers as well as providers receive the policy definitions relevant for their use cases, e.g., from a standardization body. A customer expresses her expectations based on the policy definition and compresses it for efficient later use. Similarly, for each cloud server, cloud providers create a policy that specifies which expressible expectations the system is able, and the cloud provider is willing, to fulfill. The expectations can range from restricted location over guaranteed data deletion and notification instructions up to configuration of security properties for storage or network connections.

Given this initial setup, a customer attaches the compressed expectations as *data annotation* to regulated data that she sends to the cloud (black line). The cloud service checks the received expectations against its policy. The result of this *matching* can be negative, i.e., if the service cannot comply with the expectations it does not handle the request. Otherwise, the service obtains instructions on how to handle the data, e.g., to delete it after three months. We demonstrated the applicability of this negotiation in a detailed performance analysis [4]. Still, after negotiation, the cloud service must handle the data according to the derived instructions. In this paper, we present realization strategies for all cloud layers to show feasibility of such a comprehensive policy support in the cloud.

Policy Decision Point. Yet, CPPL does not consider that some expectations must be evaluated before data reaches the service end-points, e.g., location requirements must be checked before data travels to forbidden places. To address this challenge, we introduce *policy decision points* (PDPs) which evaluate such expectations at the edge of the cloud (cf. Figure 2). Upon reception of data, a PDP selects a service that is able to comply with the requested expectations. To this end, PDPs regularly retrieve policies from cloud services and match them with received expectations[1]. Thus, PDPs enable customers, e.g., to offload location regulated tax information [2], but also enable controlled sharing with third parties, e.g., in federated clouds.

Still, most customer expectations (likewise denoted as *policies* in the following) must be addressed at the service end-points itself, e.g., data deletion. In the following sections, we thus show strategies to comprehensively realize this policy support at all cloud layers based on representative services.

IV. Providing Policy-Aware Infrastructure

As outlined in Figure 1, the IaaS layer requires measures to realize policy-aware resource management. For example, it realizes location restrictions during server bootstrapping or ensures availability of requested hardware features such as trusted computing or computational power. Similarly, customers need control over network connections that transmit regulated data, e.g., to control connection security among cloud servers or across clouds in cloud federations. In the following, we exemplify policy-aware resource management by our modifications to OpenStack and present a mechanism for policy-aware network connection setup and packet routing.

[1]Customers check the PDP's compliance with their expectations by retrieving the PDP's policy for local matching, prior to sending data to the cloud.

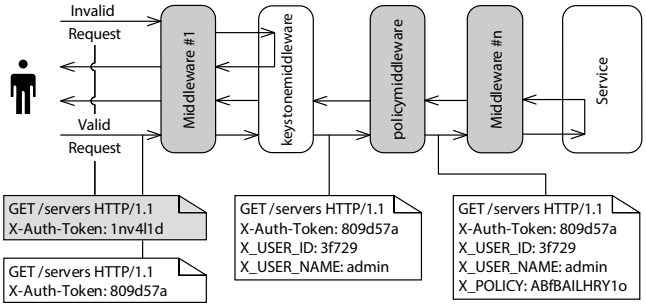

Figure 3. As *keystonemiddleware*, the *policymiddleware* transparently annotates requests with policy information to make them available to all components.

A. Realizing Customer Control on Resource Management

One primary benefit of policy-aware resource management is the provision of elasticity for regulated data: Many policies restrict the set of systems allowed to handle the affected data, e.g., location or hardware security requirements. Consequently, systems that fulfill often requested but rarely offered attributes face high load [3]. Policy-aware resource management addresses this challenge by enabling clouds to elastically bootstrap new instances *with specific attributes*, e.g., based on statistics on used policies (gathered by PDPs) and current load of services.

***OpenStack* integration.** To exemplify policy-aware resource management, we integrated policy-support in OpenStack. Specifically, we enabled customers to control the location for a new virtualized machine, support configuration of volume encryption with policies, and enable control on replication strategies. This required us to make customer expectations available to all components that contribute to resource management. Our approach comprises two major components, the *policymiddleware* and the *policyextension*-framework. With a detailed documentation being available [25], brief descriptions are provided hereinafter.

***Policymiddleware*-Component.** To make policy information transparently available to arbitrary OpenStack components, we introduce a new middleware component called *policymiddleware*, which is based on the general concepts of the *keystonemiddleware* component (see Figure 3). The *policymiddleware* validates incoming CPPL annotations and deposits the policy information in *Keystone*, from where the *policyextension*-framework can retrieve it.

***Policyextension*-Framework.** As a second component, we introduce the *policyextension*-framework, which enables developers to implement policy support through *PolicyExtensions*, which do not rely on potentially missing extension facilities but rather inject their logic through *monkey patching* mechanisms.

To implement new *PolicyExtensions*, developers create a new class containing a list of functions to be modified alongside with the methods implementing the modifications. The *policyextension*-framework handles the entire patching process and

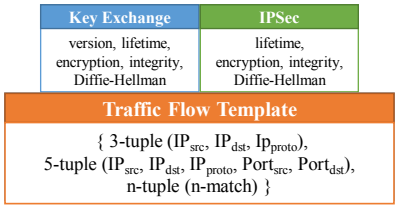

Figure 4. Controllable security parameters and available traffic flow patterns.

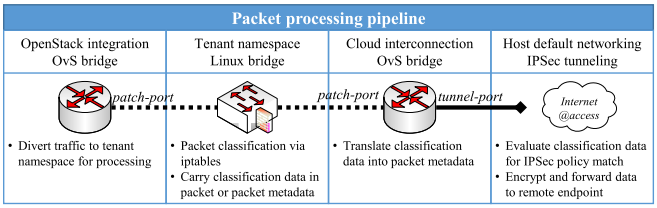

Figure 5. Packet Pipeline Classification.

provides convenience methods for accessing the arguments of the original function and makes policy annotations available from arbitrary locations in the service implementations. With external requests to OpenStack APIs resulting in a large number of internal requests among the individual services, the compact representation of CPPL evades any bloating effects caused by annotating all internal requests with policy information.

Based on the two introduced components, our design demonstrates the versatile applicability of CPPL by enabling policy-aware resource management and thus realizing, e.g., elasticity for regulated data in the cloud.

B. Enabling Policy-Aware Network Connection Setup

Beyond policy-aware resource management with OpenStack, clouds with multiple datacenters and especially federated clouds require control on network connection, e.g., to fulfill user-defined encryption requirements for inter-cloud communication. Current technologies do not offer frameworks to define security policies for such traffic. As cloud providers may not encrypt inter-cloud or inter-datacenter traffic if they have a dedicated connection, this opens possibilities of eavesdropping [26] in case of low security or misconfigurations from the users. Mitigating this risk, policy support enables customers to express their security requirements with a flow granularity. The particular challenge to realize such a fine-grained control lies in associating specific traffic patterns with their corresponding security policies. In the following, we tackle this challenge by introducing a policy-aware traffic classification mechanism.

In previous work [27], we presented a cloud federation agent for OpenStack, which is configured by the system administrator. The agent enables the expansion of tenant virtual networks across federated clouds, providing isolation and encryption. If the links between the federated clouds are not secured, IPSec tunnels could be established to ensure that tenant traffic never leaves the cloud unprotected. However, we lacked the granularity of parallel IPSec tunnels and mapping of tenant traffic with a specific tunnel. In this work, we enhance the capabilities of our system by supporting parallel tunnels together with fine-grained classification and mapping of tenant traffic. A tenant can now define its own CPPL-based security policy for traffic, i.e., express required connection security on a per-flow granularity.

To realize transmission over IPSec tunnels that comply with the traffic's policy, we (i) need to map traffic to its CPPL policy and (ii) must identify the tunnels that comply with the policy. To address the first challenge, i.e., map traffic to policies, we bundle each CPPL policy with a traffic flow template (TFT). A TFT identifies traffic with an n-tuple iptables match and thus specifies to which traffic the bundled policy applies. The policy is used to specify parameters for key exchanges and the corresponding IPSec tunnels, and thus allows tenants to control the level of security, e.g., by selection of available ciphers, interval until re-keying, and suitable key length. To set up a TFT and its CPPL policy, the cloud federation agent offers tenants a REST API. The currently available parameters for CPPL policies and TFTs (cf. Figure 4) are derived from the strongSwan reference configuration [28]. This enables sanity checks of user input.

When receiving a TFT and its CPPL policy, the cloud federation agent needs to identify IPSec tunnels that comply with the policy (cf. (ii)). To this end, it checks for each tunnel if it complies with the received policy using efficient CPPL matching (cf. Section III). Being able to identify suitable tunnels, we now only need efficient traffic classification based on TFTs to send corresponding traffic through the identified tunnels, which we detail in the following.

Traffic classification. We devise a traffic classification subsystem that is plugged in the standard OpenStack OpenvSwitch (OvS) integration bridge, benefiting from the underlying MAC learning capabilities of the switching fabric. This method allows for transparent bridging with the remote clouds. Additional network namespaces are used for enforcing the TFT rules, which perform the packet classification for later IPSec policy matching. An overview is depicted in Figure 5.

A tenant virtual network is identified within an OpenStack host by a unique VLAN id. From the OvS integration bridge it is straightforward to divert traffic to the respective tenant namespace via an internal port. The namespace contains a Linux bridge with the corresponding iptables rules defined by the TFT. Traffic classification is performed upon matching on these rules and carried with the packets. Afterwards, packets leave the namespace and continue towards the OvS cloud interconnection bridge. The classification metadata is translated before the tenant traffic is encapsulated by the tunneling ports. The unique VLAN id is mapped to a unique network id and carried in the encapsulated tunnel id. The encapsulation process maintains the translated classification metadata intact, which is later used for finding the corresponding IPSec XFRM policy. Classified packets are assured at least the requested level of security, whereas the unclassified packets are assigned a default security policy. Furthermore, the use of network namespaces allows us to establish resource limitations via *cgroups*.

We have implemented two variants for traffic classification: compatible and enhanced modes. Our *compatible mode* uses

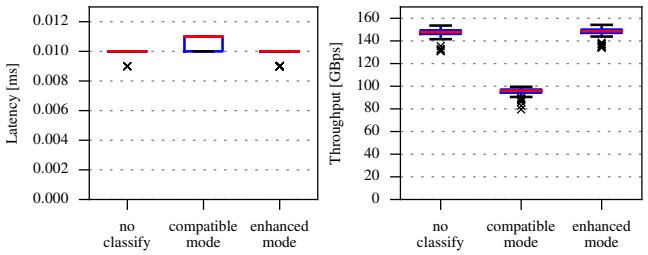

Figure 6. **Traffic classification performance:** The enhanced mode offers customers controllable networking with almost no impact on performance.

a virtual ethernet pair, resembling a network pipe. The classification metadata is carried within each packet, encoded in the VLAN PCP (priority code point - 3 bits) field. This is achieved with a combination of iptables rules using the *CLASSIFY* target and Linux Traffic Control (tc) classfull and hierarchical queuing disciplines with VLAN rewriting. However, this requires complex tc rules and filters to perform this rewriting, with a strong performance impact. The received PCP value is further translated into the packet mark upon reception by the cloud interconnection OvS bridge. This operation mode is supported across BSD and Linux OSes.

The *Enhanced mode* uses an OvS Internal Port instead of virtual ethernet pair. The classification metadata is carried alongside each packet as the packet mark (32 bit integer). This is achieved using a combination of iptables rules with the *MARK* target. The advantage of this method consists on the skb packet mark not being scrubbed while traversing the network namespace. This constitutes the most straightforward operation, albeit it is only available from OvS 2.5.x versions.

Performance evaluation. To evaluate our design, we set up an OpenStack replica of the virtual networking for a single tenant, enabled traffic classification and, in view of high traffic load handled by the connections, evaluate the network performance by means of throughput and latency following the scenario depicted in Figure 5.

We obtained throughput performance using 100 iterations of *iperf* (-t 10 -P 8) and measured latency with 100 iterations of *ping* (-f -c 100000). Figure 6 shows the results of our measurements. The performance degradation seen in the compatible mode is due to the inherent complexity of the hierarchical queuing disciplines and filters required by tc. In contrast, the enhanced mode benefits from the least computational requirements needed for traffic classification yielding similar performance as without classification, i.e., without policy support. The enhanced mode thus enables efficient CPPL-based control of network connection properties for traffic flows without latency or throughput overhead.

As our design applies to any system interconnection, it also covers intra-cloud communication. We thus enable customers to control communication security between datacenters and across clouds in a federation, hence, enabling customers to prohibit transfer of sensitive data over unsecured links, and achieve this without additional cost.

V. Controllable PaaS: Tapping Cloud Benefits for Regulated Use Cases

The PaaS layer needs to address expectations that affect storage as well as processing of services. A major challenge to be addressed at this layer is compliance with location restrictions for storage as well as processing. In contrast to IaaS, services are already running and data must not reach forbidden locations which is ensured by our policy decision points (cf. Section III), to which cloud servers report their location properties. Still, also the services must be aware of location restrictions as they often interact with services in other locations. Another challenge is the realization of reporting mechanisms, e.g., cloud customers can negotiate that file access is logged for later analysis. Furthermore, requesting secure network connection usage is vital to ensure data privacy when data is passed among different services. Finally, especially storage services face the need for guaranteed data deletion to set customers into control of the lifetime of data in the cloud, e.g., privacy regulations often require deletion of data after specific time periods. In the following, we discuss policy-aware storage and processing by means of big data management at CERN, the Hyrise in-memory research database, and Orange's use of network function virtualization to virtualize and offload customer premise equipment (e.g., residential home gateways).

A. Policies Simplify Big Data Management at CERN

To exemplify policy-aware storage, we integrated policy-support to XRootD which is used to enable big data analysis at CERN. Specifically, XRootD is used to store data from high-energy particle collisions to study the nature of elementary particles. These collisions produce petabytes of data, which has to be accessible by researchers spread around the world. High energy physics (HEP) is thus a data intensive field of science and consequently storage and retrieval of data objects is one of the core tasks. To provide this data access, HEP data objects are stored in large ROOT files, which are accessed through data servers such as XRootD. Typically, HEP analysis jobs read small parts from several large ROOT files.

This scenario can be relevant beyond XRootD, e.g., to address challenges when dealing with research data obtained under Non Disclosure Agreements (NDAs). Here, policy-aware storage is needed when processing such sensitive data in conjunction with cloud setups.

To address such demands, we exemplify policy-aware storage by realizing access logging and guaranteed data deletion for XRootD. For access logging, we log IP address, file name, and time stamp upon file creation and access while data deletion enforces a limited storage time. Traditionally, these policies are followed by system administrators rather informally by changing system configurations and manually running scripts. To automate and stabilize adherence to policies, we enhanced XRootD with policy support and annotate HEP data files with policies. To this end, each upload or download of a file incorporates checking the policy. For both actions, if the policy enforces access tracking, we log the IP address, file name and time stamp. For upload actions of files that request limited

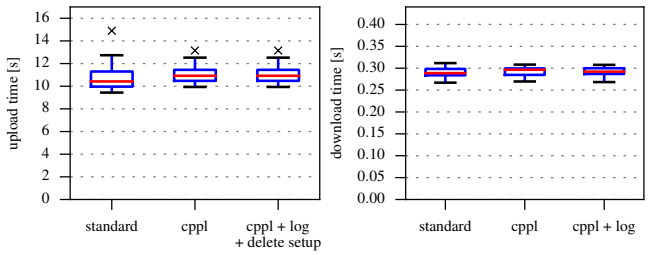

Figure 7. **Performance of policy-aware XRootD:** Our results show negligible overhead for policy negotiation as well as accompanied actions.

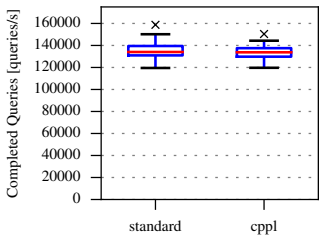

Figure 8. **Performance of policy support in Hyrise-R:** For high-throughput, transactional workloads, the policy evaluation incurs no notable overhead.

storage time, we furthermore set up a time-based trigger for data deletion according to the lifetime specified in the policy.

Performance evaluation. To evaluate the effect of policy-support on XRootD, we measured upload and download times for a standard and a policy-aware XRootD server. The server was deployed in an OpenStack VM (1vCPU and 2GB RAM) at Aalto University and accessed by clients from an identical VM located at CERN.

Derived from real usage patterns, we uploaded 300 MB files for a period of five minutes with one process. For our download test, we instructed 10 parallel processes to retrieved 10 kB files over the course of one minute. We measured execution times for each up- and download and repeated each test 30 times. Figure 7 shows the results. For the CPPL enabled case, we present two results, one including only checking of the CPPL policy and one that also measures execution of the derived instructions. The results show negligible overhead for policy negotiation, as CPPL provides space-efficiency (with compression) and fast matching (cf. Section III), as well as for accompanied actions. Thus, our design and realization enable customer control on storage for big data analysis. Applying policy-aware request processing to further cloud storage systems, hence realizes large scale, efficient policy-aware storage, e.g., to enable NDA-compliant offloading of research data to the cloud, but also to realize general policy-compliant storage of regulated data.

B. Realizing Policy-Aware Storage in In-Memory Databases

Beyond policy-aware storage for XRootD, we also investigated policy integration strategies for distributed in-memory databases, as these offer an essential form of data storage in business-oriented use-cases. However, database administration is known to be a demanding task, since expert knowledge is required to properly set up and tune databases to provide good performance. Hence, outsourcing the operation of databases to corresponding PaaS offerings is becoming increasingly popular. Especially for PaaS-based database offerings, strict policy adherence is vital, as databases often hold crucial business assets. To not impede the substantial performance gains of *In-Memory Databases* (IMDB), it is necessary that policy adherence mechanisms do not tax the overall performance of PaaS-based *IMDB* offerings.

To study and enable efficient realization of policy support in *IMDBs*, we augmented the Hyrise [29] open source in-memory research database with policy adherence mechanisms based on CPPL. Hyrise uses replication mechanisms to support cloud-based scale-out deployment [30], elasticity [31], as well as high availability features. Currently, Hyrise-R implements full replication, where the entire data is stored at every node in the database cluster. Using this approach, every node can process every query. However, expressive policy adherence mechanisms enable more fine-grained replication and data distribution mechanisms in cloud scenarios compared to maintaining full copies. For example, some data may be restricted to be stored in specific geographic locations. Other data may demand a minimum replication rate for high availability. In these cases, partial replication enables increased flexibility and improves resource utilization, as every database node only has to maintain a subset of the entire database. Using partial replication, a policy annotation is crucial, if the user wants to specify what data fragments are allowed to be stored on which nodes.

As an initial prototype on the way towards achieving partial replication, we implemented policy adherence support for the properties location and replication rate based on the CPPL policy language. First, we adapted the query dispatcher to only forward queries to replica nodes located in a permissible location, storing all queried data fragments. Second, integrating policy support also affected the synchronization mechanism of replicas, i.e., the transmission of data manipulation messages only to permissible nodes storing related data fragments.

Performance evaluation. Performance measurements were conducted by simulating a high-throughput transactional workload. No-op queries were used instead of actual operations to exclude confounding effects of actual operations as potential decelerating components that do not contribute to the integration of CPPL. To retrieve a sufficiently meaningful dataset, we performed 30 repeated measurements of both the standard and the CPPL-enabled implementations of the Hyrise-R dispatcher, which is backed by two Hyrise replica instances. The results depicted in Figure 8 indicate negligible overhead for policy evaluation. Thus, based on the compact representation and the efficient matching of policies in CPPL, cloud-optimized scale-out deployments of in-memory databases can offer policy support without notable performance degradation.

C. Benefits of Controllable Processing – Enabling Orange to Virtualize and Offload Customer Hardware to the Cloud

Apart from cloud storage, also processing of data in the cloud can be affected by regulations. To exemplify corresponding requirements on the PaaS layer, we focus on the attempt of Orange Polska, a large ISP, to virtualize customer premise

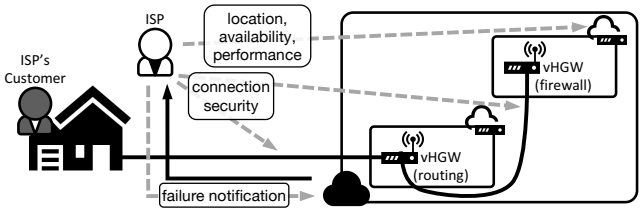

Figure 9. Policy-aware PaaS enables tapping cloud benefits for regulated cloud processing, e.g., ISPs can offload network functionality to the cloud.

equipment functionality (e.g., routing or firewall filtering in home gateways) and offload processing to the cloud as virtualized home-gateway (vHGW). These vHGWs are envisioned to replace today's heterogeneous deployments comprised of various HGW types that offer differing functionality which makes operation, administration, and maintenance complex especially at a large scale. Instead, offloading functionality to the cloud as vHGWs allows different devices to offer unified functionality, reduces capital and operational expenditures of ISPs, and enables fast deployment of new functionality. To realize this offloading, several regulations and customer expectations must be addressed. These range from privacy of customers, over security of network connections, up to availability and performance demands. While we attribute the vHGW to the SaaS layer, we still choose it to discuss policy realization for PaaS which realizes most of the corresponding requirements on policy support.

Typically, ISPs need to follow privacy regulations that apply to customer communication. As depicted in Figure 9, they need to restrict the location of a vHGW, e.g., to the country of the customer [2]. To realize this location support we can instruct policy decision points (PDPs) to enforce location regulations for vHGW bootstrapping and relocation (cf. Section III). Similarly, ISPs must ensure confidentiality of data transmissions between customers and their vHGWs. Specifically, today's local area security mechanisms must be replaced with end-to-end security mechanisms when offloading HGWs to the cloud. Furthermore, different functionality of a vHGW such as routing, DHCP, and firewall can be distributed to different cloud servers. These functional blocks must likewise securely communicate with each other. To ensure suitable connection security in all these cases, we can adapt our design used to realize control on network setup at the IaaS layer (cf. Section IV-B).

Beyond privacy and security demands, vHGWs must provide similar or better performance as today's deployed hardware. Thus, the PaaS layer must meet performance metrics, especially regarding delay, jitter, throughput, and packet loss as well as datacenter availability. Prior to vHGW deployment, PDPs use information on server attributes, which they receive similar as location information, to match the requirements with the capabilities of cloud servers. Compared to today's SLAs, the use of policies enables more dynamic control, e.g., each vHGW can request a specific performance level based on end-user traffic patterns. After deployment, constant monitoring of performance metrics can enable adjustments or relocation on demand.

Moreover, policy support enables configurable reporting to customers, e.g., to enhance monitoring capabilities of ISPs for their vHGWs: Today, failure of HGWs can be detected within the ISP's own network. Contrarily, failures in the cloud require the cloud to notify the ISP on occurrence and details of a problem. To this end, customer expectations negotiate instructions on error notification, e.g., how and where the cloud has to report errors to the ISP. This enables ISPs to handle vHGW failures automatically even though they run on foreign cloud infrastructure. For example, ISPs can instruct the cloud to provide information that enable the ISP to trigger actions like relocation of the vHGW. Negotiating specific error handling instructions even allows the cloud to handle issues automatically without interaction with the ISP at time of error. Finally, realizing data deletion with policies enables ISPs to comply with data storage regulations, i.e., the requirement to store data for a certain amount of time for inspection but also to ensure deletion of data according to privacy laws.

Although missing confidence in foreign clouds limits ISP's plans to the usage of their own cloud infrastructure today, we hope that our work establishes increasing trust such that also offloading of mission critical data and services to foreign clouds and cloud federations becomes viable for vHGWs, as well as for processing in regulated contexts in general.

VI. Policy-Aware SaaS: New Business Cases, Broader Customer Base, Simplified Configuration

To complete comprehensive policy support for the cloud, we now show that adherence to customer expectations at the SaaS layer enables enhanced and new business cases. To this end, services at the SaaS layer can draw upon policy-support of the IaaS and PaaS layer (cf. Sections IV, V). Still, there are a lot of very service specific expectations of customers on the handling of their data, e.g., restriction regarding data aggregation, usage for automated processes such as machine learning, or sharing with third parties. In the following, we exemplify such expectations by means of the Security Cloud offer of F-Secure which provides security solutions to their customers. Making this service policy-aware and thus putting customers into control of handling of their data greatly enhances confidence of customers into the service. Thus, a policy-aware SaaS has the potential to significantly broaden the customer base, increasing cloud provider profit but also enabling usage of the service for customers with regulated data.

A. Adressing Customer Expectations in F-Secure's Security Cloud Improves Service Quality and Security of Customers

F-Secure products range from protecting an endpoint, (laptop, phone, server) to protecting an organization from threats posed by user provided content in cloud based SaaS services, and lastly detecting breaches in an organization. The core of F-Secure's Security Cloud is a knowledge base of digital threats that is constantly growing and evolving as data is gathered from client applications and accumulated through automatic threat analysis. Centralizing the information used to combat digital threats into a cloud service provides many benefits. New knowledge about threats can be utilized faster, and the system

can consolidate data from a large range of clients and maintain a picture of the global threat situation.

For analysis, F-Secure retrieves new, not yet analyzed files for analysis from their customers. In the past, however, customers often excluded specific types of files, e.g., text documents, from cloud analysis due to worries about security and privacy and the need to comply with regulations, e.g., the *General Data Protection Regulation*. This broad exclusion of files from security analysis puts customers at significant security risks as documents that exploit zero day vulnerabilities became one of the main attack vectors in high profile targeted attacks. Hence, to protect against security threats but also comply with security and privacy requirements, F-Secure's customers have demand for fine-grained control on data handling in the cloud. For F-Secure, realizing policy-support thus enables new business cases based on regulated and confidential data. Furthermore, the increased amount of analyzed data results in a better trained knowledge base which increases the quality of security analysis.

Based on customer requests, F-Secure identified a set of relevant policy attributes: As content submitted for analysis is often user generated or confidential, e.g., emails or documents, customers request F-Secure to delete data after analysis or want to restrict analysis to automated processes, i.e., exclude manual analysis by humans. Other regulations limit the location of the analysis, e.g., customer's tax information must not leave the originating jurisdiction [2]. Even more, customers would like to control the use of third parties involved in security analysis, or even limit analysis to servers within the customer's own organization, e.g., to prevent information leakage. Furthermore, some organizations such as governments, military, or insurance providers affected by the *Health Insurance Portability and Accountability Act* request that their data is handled separately from other organizations. Finally, F-Secure also incorporates meta-data, e.g., creator or origin of a document, in the analysis. While this data is valuable to train the knowledge base, and can thus improve service quality, it is often of sensitive nature. Thus, customers require means to express availability or exclusion of (meta-)data for knowledge base training.

To address these requirements of customers, F-Secure can annotate data with corresponding policies which travel with the data (as any other meta-data). In contrast to specifically tailored realizations, the annotation provides each component with a uniform representation of customer requirements, thus, significantly easing policy integration. CPPL's features enable an efficient realization of this concept: Compression ensures low overhead for passing on policies with the data and fast matching incorporates negligible overhead for policy checking at different components (cf. Section III). Beyond expressing customer requirements, also applications can express their requirements, thus, addressing F-Secure's need to specifically shape the security analysis for different appliances, e.g., cloud or smartphone protection. Finally, it enables F-Secure to easily adapt to changes posed by law or regulatory changes as well as setting up new standard data handling procedures based on changing customer perceptions regarding security and privacy.

Summing up, customers already actively express their demand for the features of a policy-aware SaaS layer as it enables them to use cloud services for regulated and confidential data. For cloud providers, establishing this support does not only yield new business cases and broadens customer base but also eases management and extension of their cloud services.

VII. Conclusion

To tap highly available, scalable, and elastic resources of clouds for regulated use cases, customers must be enabled to control data handling in the cloud. Extending upon the recent CPPL policy language that is specifically tailored for cloud scenarios, we show that a wide spectrum of industry-driven cloud use cases covering all layers of a cloud stack can be realized. Specifically, we enable elasticity for regulated data by realizing policy-aware resource management (with OpenStack) and set customers into control of intra- and inter-cloud communication, thus, enabling them to, e.g., mitigate information leakage due to transfer over unsecure links. Exemplified by big data management at CERN, we showed that realizing policy-aware storage comes with negligible overhead for data upload and access which, as we showed based on our realization for the in-memory database Hyrise, even holds for high-throughput transactional workloads. Drawing upon this support, we demonstrated means to realize policy-aware processing exemplified by Orange Poland's attempts to offload network functionality to cloud environments to significantly reduce costs. For applications at the top of a policy-aware cloud stack, as demonstrated for F-Secure's Security Cloud, policy-awareness enables new business cases, broadens the customer base and simplifies cloud service management.

Moreover, we extended CPPL with policy decision points that realize compliance with policies that limit the set of cloud servers that are allowed to receive data, e.g., due to location regulations, and also enables policy-awareness in federated cloud scenarios. Beyond feasibility of comprehensive policy realization in the cloud, our results also witness the necessity of key performance features of CPPL for cloud computing based on a wide spectrum of industry-driven cloud use cases.

As part of future work, we plan to enable cloud providers to technically prove their compliance with expressed customer requirements, e.g., using trusted computing technology. We hope that our positive experience and results motivate further integration of policy-support into cloud services to foster support for tapping highly available, scalable, and elastic cloud resources for regulated data. CPPL is publicly available [4].

Acknowledgements

This paper has received funding from the European Union's Horizon 2020 research and innovation programme 2014-2018 under grant agreement No. 644866 (SSICLOPS). It reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains. We would like to thank the German Research Foundation DFG for the kind support within the Cluster of Excellence "Integrative Production Technology for High-Wage Countries".

REFERENCES

[1] Intel IT Center, "Peer Research: What's Holding Back the Cloud?" Intel, Tech. Rep., 2012.

[2] N. Cory, "Cross-border data flows: Where are the barriers, and what do they cost?" 2017, accessed 24.10.2017. [Online]. Available: http://www2.itif.org/2017-cross-border-data-flows.pdf

[3] M. Henze, R. Matzutt, J. Hiller, E. Mühmer, J. H. Ziegeldorf, J. v. d. Giet, and K. Wehrle, "Practical data compliance for cloud storage," in *IEEE IC2E*, April 2017.

[4] M. Henze, J. Hiller, S. Schmerling, J. H. Ziegeldorf, and K. Wehrle, "Cppl: Compact privacy policy language," in *ACM WPES*, 2016, source code available at https://github.com/SSICLOPS/cppl.

[5] M. Y. Becker, A. Malkis, and L. Bussard, "A practical generic privacy language," in *ICISS*, 2010.

[6] "eXtensible access control markup language (XACML) version 3.0," OASIS Standard, 2013.

[7] L. Bussard, G. Neven, and F. S. Preiss, "Downstream usage control," in *POLICY*, 2010.

[8] M. Azraoui, K. Elkhiyaoui, M. Önen, K. Bernsmed, A. S. Oliveira, and J. Sendor, "A-PPL: An accountability policy language," in *DPM*, 2014.

[9] R.-A. Cherrueau, R. Douence, H. Grall, J.-C. Royer, M. Sellami, M. Südholt, M. Azraoui, K. Elhhiyaoui, R. Molva, M. Önen, A. Garaga, A. S. Oliveira, J. Sendor, and K. B ernsmed, "Policy representation framework," A4Cloud Consortium, Tech. Report, 2013.

[10] J. Poroor and B. Jayaraman, "C2L: A formal policy language for secure cloud configurations," in *ANT*, 2012.

[11] R. Kantola, J. Llorente Santos, and N. Beijar, "Policy-based communications for 5g mobile with customer edge switching," *Security and Communication Networks*, vol. 9, no. 16, 2016.

[12] M. G. Jaatun, I. A. Tøndel, N. B. Moe, D. S. Cruzes, K. Bernsmed, and B. Haugset, "Accountability requirements for the cloud," in *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Dec 2017, pp. 375–382.

[13] M. G. Jaatun, S. Pearson, F. Gittler, R. Leenes, and M. Niezen, "Enhancing accountability in the cloud," *International Journal of Information Management*, 2016.

[14] R. Thion and D. Le Metayer, "Flavor: A formal language for a posteriori verification of legal rules," in *IEEE POLICY*, 2011, pp. 1–8.

[15] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: Protecting confidentiality with encrypted query processing," in *ACM SOSP*, 2011.

[16] J. H. Ziegeldorf, J. Pennekamp, D. Hellmanns, F. Schwinger, I. Kunze, M. Henze, J. Hiller, R. Matzutt, and K. Wehrle, "Bloom: Bloom filter based oblivious outsourced matchings," *BMC Medical Genomics*, vol. 10, no. 2, p. 44, 2017.

[17] M. Henze, J. Hiller, R. Hummen, R. Matzutt, K. Wehrle, and J. H. Ziegeldorf, *Network Security and Privacy for Cyber-Physical Systems*, in *Security and Privacy in Cyber-Physical Systems*. Wiley, 2017, pp. 25–56.

[18] S. Betgé-Brezetz, G. B. Kamga, M. P. Dupont, and A. Guesmi, "End-to-end privacy policy enforcement in cloud infrastructure," in *IEEE CloudNet*, Nov 2013, pp. 25–32.

[19] J. Modic, R. Trapero, A. Taha, J. Luna, M. Stopar, and N. Suri, "Novel efficient techniques for real-time cloud security assessment," *Computers & Security*, vol. 62, pp. 1 – 18, 2016.

[20] M. Anisetti, C. A. Ardagna, E. Damiani, F. Gaudenzi, and R. Veca, "Toward security and performance certification of open stack," in *2015 IEEE 8th International Conference on Cloud Computing*, 2015, pp. 564–571.

[21] R. Pires, D. Gavril, P. Felber, E. Onica, and M. Pasin, "A lightweight mapreduce framework for secure processing with sgx," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, ser. CCGrid '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 1100–1107. [Online]. Available: https://doi.org/10.1109/CCGRID.2017.129

[22] L. V. Silva, R. Marinho, J. L. Vivas, and A. Brito, "Security and privacy preserving data aggregation in cloud computing," in *Proceedings of the Symposium on Applied Computing*, ser. SAC '17. New York, NY, USA: ACM, 2017, pp. 1732–1738. [Online]. Available: http://doi.acm.org/10.1145/3019612.3019795

[23] F. Kelbert, F. Gregor, R. Pires, S. Köpsell, M. Pasin, A. Havet, V. Schiavoni, P. Felber, C. Fetzer, and P. Pietzuch, "Securecloud: Secure big data processing in untrusted clouds," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, March 2017, pp. 282–285.

[24] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'Keeffe, M. L. Stillwell, D. Goltzsche, D. Eyers, R. Kapitza, P. Pietzuch, and C. Fetzer, "SCONE: Secure linux containers with intel SGX," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, 2016, pp. 689–703. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/arnautov

[25] M. Plauth, M. Bastian, and A. Polze, "Facilitating Policy Adherence in Federated OpenStack Clouds with Minimally Invasive Changes," in *Proceedings of the Fifth HPI Cloud Symposium "Operating the Cloud"*, 2017, (to appear).

[26] S. Landau, "Highlights from making sense of snowden, part ii: What's significant in the nsa revelations," *IEEE Security Privacy*, vol. 12, no. 1, pp. 62–64, Jan 2014.

[27] M. Kimmerlin, P. Hasselmeyer, S. Heikkilä, M. Plauth, P. Parol, and P. Sarolahti, "Network expansion in OpenStack cloud federations," in *EuCNC*, June 2017.

[28] strongSwan, "ipsec.conf Reference," accessed 24.10.2017. [Online]. Available: https://wiki.strongswan.org/projects/strongswan/wiki/ConnSection

[29] M. Grund, J. Krüger, H. Plattner, A. Zeier, P. Cudré-Mauroux, and S. Madden, "HYRISE - A Main Memory Hybrid Storage Engine," *PVLDB*, vol. 4, no. 2, 2010.

[30] D. Schwalb, J. Kossmann, M. Faust, S. Klauck, M. Uflacker, and H. Plattner, "Hyrise-R: Scale-out and Hot-Standby through Lazy Master Replication for Enterprise Applications," in *IMDM*, 2015.

[31] S. Klauck, "Scalability, Availability, and Elasticity through Database Replication in Hyrise-R," in *Proceedings of the Fourth HPI Cloud Symposium "Operating the Cloud"*, 2016.