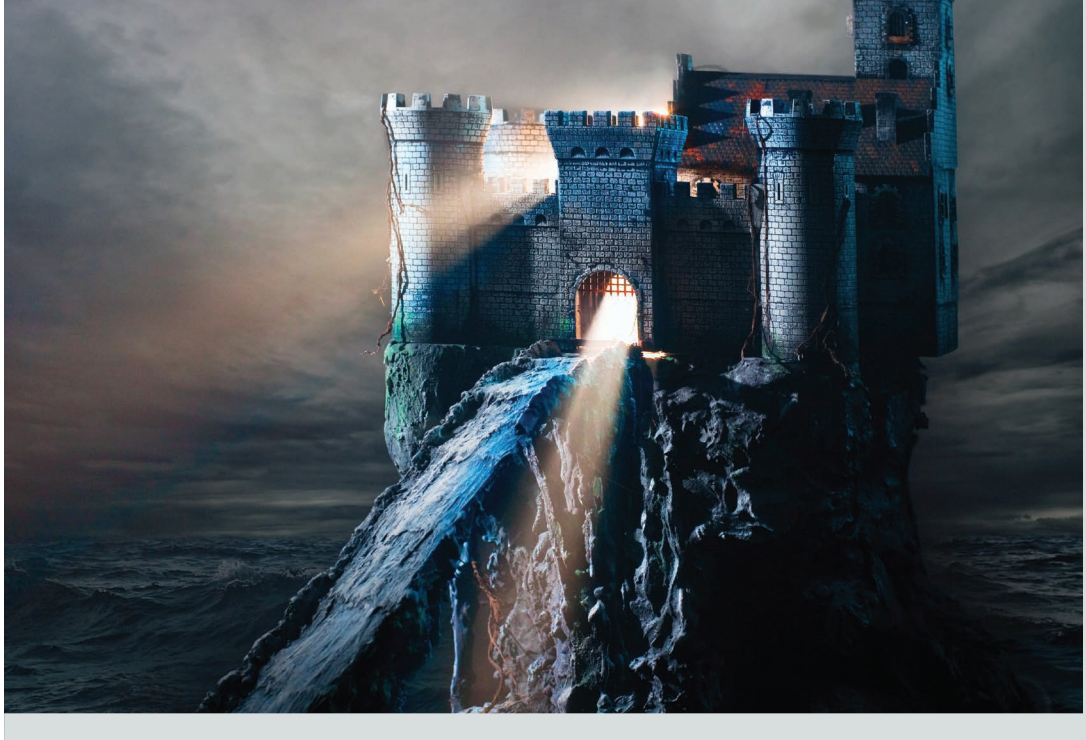

User-Centric Security and Dependability in the Clouds-of-Clouds

Marc Lacoste, Orange Labs
Markus Miettinen, Technische Universität Darmstadt
Nuno Neves and Fernando M.V. Ramos, University of Lisbon
Marko Vukolić, IBM Research
Fabien Charmet and Reda Yaich, Institut Mines-Telecom
Krzysztof Oborzyński and Gitesh Vernekar, Philips Healthcare
Paulo Sousa, Maxdata Software

Secure Supercloud computing aims to provide security and dependability management of distributed clouds. This approach is both user-centric and self-managed, enabling users to achieve provider independence for security management.

The high maintenance costs of private datacenters and disaster-recovery requirements are causing cloud architectures to go distributed. Virtualization is expanding outside a single datacenter for compute, network, storage, and devices. Resource-specialized clouds are becoming federated, evolving from centralized to fully distributed infrastructures across heterogeneous resources—a cloud-of-clouds—and away from the datacenter to the edge.[1,2]

These new architecture paradigms present key benefits:

- better user performance (for example, lower end-to-end latency) due to fine-grained geodistribution,
- lower costs by choosing best-of-breed cloud providers in terms of pricing model,[3] and
- improved resilience to avoid wide-area outages due to single points of failure.

Nevertheless, distributed cloud computing raises several concerns,[4] mainly due to these systems' high complexity and the current lack of interoperability between heterogeneous, often proprietary, infrastructure technologies.

In practice, distributed cloud computing has remained highly provider-centric, and multicloud integration remains a challenge. Adoption also suffers from vendor lock-in, with services tightly coupled to providers. Lack of interoperability stems mainly from the heterogeneity of technologies (for example, different hypervisors), and from service-resources mappings that are incompatible across providers, hampering, for instance, uniformity in service-level agreements (SLAs). User control is also limited by monolithic infrastructures, preventing fine-grained cloud customization by the customer (for example, hypervisors hide specific hardware capabilities).

Multicloud infrastructures also raise several security and dependability challenges. First, infrastructure layers, which include customer virtual machines (VMs), provider hypervisors, and services, are extremely vulnerable to attacks, in part due to new virtualization technologies,[5] so the infrastructures can't be trusted. Second, interoperability and unified control of security across providers is mostly absent. Policy heterogeneity among providers facilitates the introduction of more vulnerabilities because of mismatching APIs and workflows. Finally, security administration challenges for such complex infrastructures clearly prohibit a manual approach. Automation of security management is required, but still lacking, in the multicloud.

In today's provider-centric clouds, service specification, security, dependability, pricing, and SLAs are beyond users' influence. To tackle the security and dependability challenges in a multicloud, we need new infrastructure management paradigms that are both user-centric and self-managed. The former means enabling self-service of cloud-of-clouds, where customers define their own protection requirements and can avoid technology and vendor lock-ins. The latter means reducing the administration complexity of cloud-of-clouds through automation techniques.

Secure Supercloud Computing

This article introduces the notion of Supercloud, a new architectural concept that follows the vision of user-centric distributed cloud security and dependability management.[6] Supercloud can be understood as a security distribution layer, providing an end-to-end interface between user-centric and provider-centric views of multiple clouds.

Supercloud deploys several user-centric clouds (or U-Clouds). A U-Cloud is a set of computation, data storage, and communication services that lets individual Supercloud users run their applications and services over a distributed cloud. U-Clouds can be implemented on top of resources from several providers. However, strict U-Cloud is guaranteed using data encryption and dedicated U-Cloud-specific VMs for computation.

Supercloud addresses the interoperability challenge by providing a resource abstraction layer spanning multiple cloud providers, decoupling resource production by cloud providers from their consumption by users. It also addresses the control challenge by enabling customers to deploy clouds with self-service security, ranging from software as a service (SaaS) to full infrastructure as a service (IaaS), independent of the underlying providers. In addition, it offers unified control for automated management of security and resilience across different clouds.

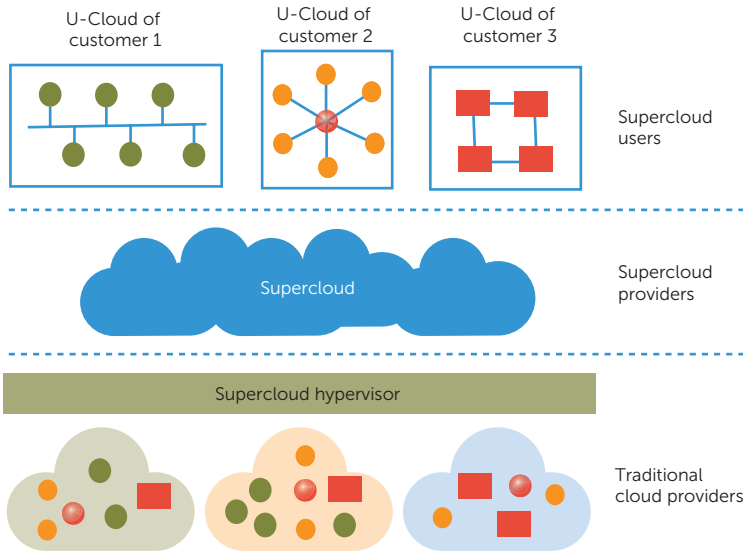

FIGURE 1. The Supercloud concept, which includes users, Supercloud providers, and traditional cloud providers. Cloud resource consumption (by users) is separated from cloud resource production (by cloud providers) thanks to the Supercloud layer, thus enabling it to overcome vendor lock-in.

This approach has several benefits. First, independence from the provider means lower infrastructure operation overhead and faster service deployment, but also increased homogeneity. Second, increased customizability can also be expected, as the customer can choose which virtualized services (such as for security) to deploy, resulting in fully à la carte clouds. Third, it can create new business opportunities and ecosystems.[7]

In a nutshell, Supercloud is a provider-agnostic distributed virtualization infrastructure for running U-Clouds, leveraging compute, data, and network resources from both public cloud providers and private cloud infrastructures (see Figure 1). This heterogeneity impacts the level of infrastructure visibility and control that can be achieved for services running in U-Clouds.

On one end, public clouds operated by commercial cloud service providers (CSPs) give only limited visibility and control over the hypervisor and the network. Public CSPs are typically big players, offering commercial cloud services to their customers at massive scale, allowing them to take advantage of cost savings and elastic resources. They provide well-defined high-level service APIs with few possibilities for individual customers to customize the deployment details of their cloud service instances.

On the other end, in private clouds, where the datacenter belongs to the user, full infrastructure access can be reached. Private CSPs are typically entities such as a corporation's IT department, supporting tailored cloud services for its own organization. These services aren't limited to a narrow service API, but can also have lower-level control over specific deployment details (of the computational, storage, and networking resources). This enables the U-Cloud to provide user-centric system services (USS), extending user control into the lower layers of the infrastructure to enforce flexible, secure, and dependable computing behaviors (for example, firewalling, introspection, or live migration).

Supercloud is a step away from provider-centric cloud interoperability approaches (such as hybrid/federated clouds) that rely on business, interface, or protocol agreements between providers. The Supercloud approach is closer to customer-centric solutions (such as multicloud and broker-based aggregation), but focuses on security: interoperability is transparent to providers, using an adaptation layer or third-party operation. (See the sidebar for a discussion of other work in this area.)

Requirements

To meet these challenges, the Supercloud architecture should address the following objectives:

- *Self-service security*: Users should be able to specify their own protection requirements and manage the corresponding security and privacy policies autonomously, to control their resources' security in a fine-grained manner.
- *Self-managed security*: The architecture should automatically and seamlessly manage the distributed cloud's security over compute, storage, and network layers, and across provider domains to ensure compliance with user-defined security policies.
- *End-to-end security*: The architecture should guarantee SLAs (for example, for isolation) for multiple compute clouds, data protection in a multiprovider setting, and secure network interconnection.
- *Resilience*: Resource management should provide robust composition of provider-agnostic resources, leveraging primitives from multiple providers.

This leads to the following requirements.

First, the Supercloud architecture must enable provider independence and isolation. It should offer a distributed cloud infrastructure that lets users deploy cloud applications and services in specific cloud instances (that is, U-Clouds) in a transpar-

RELATED WORK IN MULTICLOUD INFRASTRUCTURE SECURITY

Many distributed virtualization infrastructures (for example, microhypervisors, nested virtualization, container platforms, and library operating systems), isolation and trust management technologies, and protection automation techniques have tackled security challenges related to multiprovider interoperability and vulnerable software layers, but without meeting requirements for user control, low attack surface, interoperability, and legacy compatibility. The Supercloud hybrid virtualization architecture enables flexible but efficient user-centric tradeoffs in terms of both interoperability and security for the multicloud.

Many solutions, such as Google Drive and Dropbox, allow users to store their own data on the cloud. However, most don't permit user-centric data encryption. In current cloud-based data storage solutions, no commercial product uses advanced cryptographic tools for data confidentiality protection, such as those we propose to use in Supercloud. Most major infrastructure-as-a-service providers offer replication solutions across multiple datacenters to support dependability; however, this remains limited to proprietary protocols and single administrative domains.

Software-defined networking-based virtualization solutions allow cloud providers to offer complete network virtualization.[1] They give tenants the freedom to specify their network topologies and addressing schemes, while guaranteeing the required level of isolation. These platforms, however, have been targeting the datacenter of a single cloud provider with full control over the infrastructure. In Supercloud, we extend this concept, supporting the creation of virtual networks spanning multiple datacenters that might belong to distinct cloud providers, while including private facilities owned by the tenant. The novelty of our solution arises mainly from tackling the challenges of using multiple clouds, including public clouds on which we have very limited control.

Reference

1. T. Koponen et al., "Network Virtualization in Multi-tenant Datacenters," *Proc. 11th USENIX Symp. Networked Systems Design and Implementation* (NSDI), 2014, pp. 203–216.

ent and user-configurable manner. Individual U-Clouds must be strictly separated, preventing, for instance, misbehaving U-Clouds from impacting other U-Clouds.

The architecture must also support interoperability at the infrastructure and platform levels. It should support a distributed cloud with flexibility and control levels similar to those in a single-provider scenario—for example, in terms of usage or migration of resources across providers. In particular, it should enable the deployment of legacy applications and management tools in the distributed cloud infrastructure.

Third, it should enable user-controlled security. It should allow users to define fine-grained security settings to control the protection level of their cloud resources. For instance, to meet legal requirements that prohibit transfer of particular data types across jurisdictional boundaries, users might need to control where their U-Cloud data is physically stored and processed. It must also protect user privacy by preventing cloud providers from accessing user data without the user's explicit consent.

Finally, the architecture should guarantee integrity and availability of services and data. It should allow specification and enforcement of measures related to integrity, redundancy, and disaster recovery of data resources as part of a user-provider SLA. Performance guarantees might also be required, namely on response times for critical accesses to some data resources.

System Architecture

We now describe the architecture of the Supercloud, both statically (that is, its components) and dynamically (that is, how these components interact to guarantee overall security).

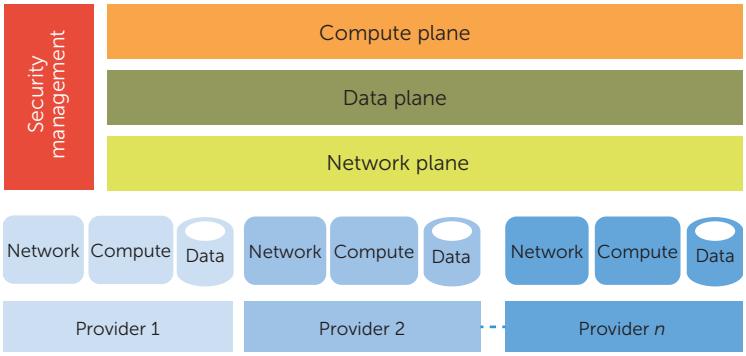

FIGURE 2. High-level overview of the Supercloud architecture, including compute, data, and network planes, and security management framework.

Static Architecture

The Supercloud architecture allows customers to instantiate U-Clouds that run on the underlying infrastructure. Figure 2 shows the three abstraction planes, each addressing a particular aspect of the Supercloud system. Each plane is realized with resources from the underlying CSPs.

The *compute* plane enables users to instantiate computational nodes regardless of physical servers hosting computations. The *data* plane realizes an abstract cloud data storage service transparent to providers and data resources providing the physical storage space. The *network* plane provides the connectivity between computational and storage resources regardless of the networking infrastructure realizing physical connectivity between servers hosting computational nodes and data storage. A security management framework provides fine-grained control to users over protection of any computational, data, and networking resources in the abstraction planes.

This layered design minimizes interface complexity between planes, clearly defining interdependences between architectural components. Users can deploy computational nodes and storage resources in the Supercloud system easily and flexibly, regardless of the specific technical requirements of individual CSPs' resource platforms: orchestration of resources for computation, storage and networking is handled by respective abstraction planes.

Interplay between the three planes allows flexible and efficient attack mitigation. A key risk of federated clouds is that a malicious component be present in a U-Cloud, considered as a service composition.[8] Each plane provides relevant countermeasures: enforcing VM isolation, attesting service trustworthiness, guaranteeing data availability, or sanitizing the network environment. Such mechanisms can be orchestrated with the security management plane to prepare and enforce a relevant security response.

Compute plane and security self-management. Figure 3a shows a simplified computing view of two U-Clouds: one (U-Cloud A) spans different providers, while another (U-Cloud B) is confined to a single provider.

The virtualization infrastructure is a distributed abstraction layer for computing resources across multiple providers. Nested virtualization is a core U-Cloud technology because it offers interoperability and security benefits to guarantee VM protection despite untrusted virtualization layers.[9] The provider controls the lower virtualization layer, called L0. Public clouds usually run general-purpose hypervisors (for example, Xen for Amazon). In private clouds, more modular hypervisors enable users to take control on a part of L0 in the form of infrastructure services for deep, fine-grained customization of U-Cloud security. The upper virtualization layer, called L1, provides the necessary facilities for users to instantiate execution environments forming layer L2, using VMs or containers that are under users' control. A horizontal orchestration component typically realizes distributed execution or migration of L2 environments connecting multiple L1 instances.

Supercloud mainly addresses security at the infrastructure level, for example, to guarantee isolation among system computation units such as VMs or containers and enforce a U-Cloud boundary. Security partitioning of applications across clouds is also important to users,[10] and can be achieved on top of the Supercloud layer as in single clouds. Provider heterogeneity is hidden within the U-Cloud, already a secure, distributed environment for application deployment.

The self-management infrastructure implements autonomic configuration and management of security aspects for the distributed cloud. Such automation means simpler, faster, and more efficient detection and response to threats, minimizing overall human intervention. U-Cloud-specific components also let users control their U-Cloud's security settings. An overall component arbitrates between such settings and provider security requirements. The security response to a threat is elaborated by orchestration of multiple autonomic security loops across infrastructure layers and providers. Other services include flexible isolation, trust management, configuration compliance for auditability, and authorization.

Data plane. Figure 3b shows several types of storage entities in the data plane. *Clients* represent

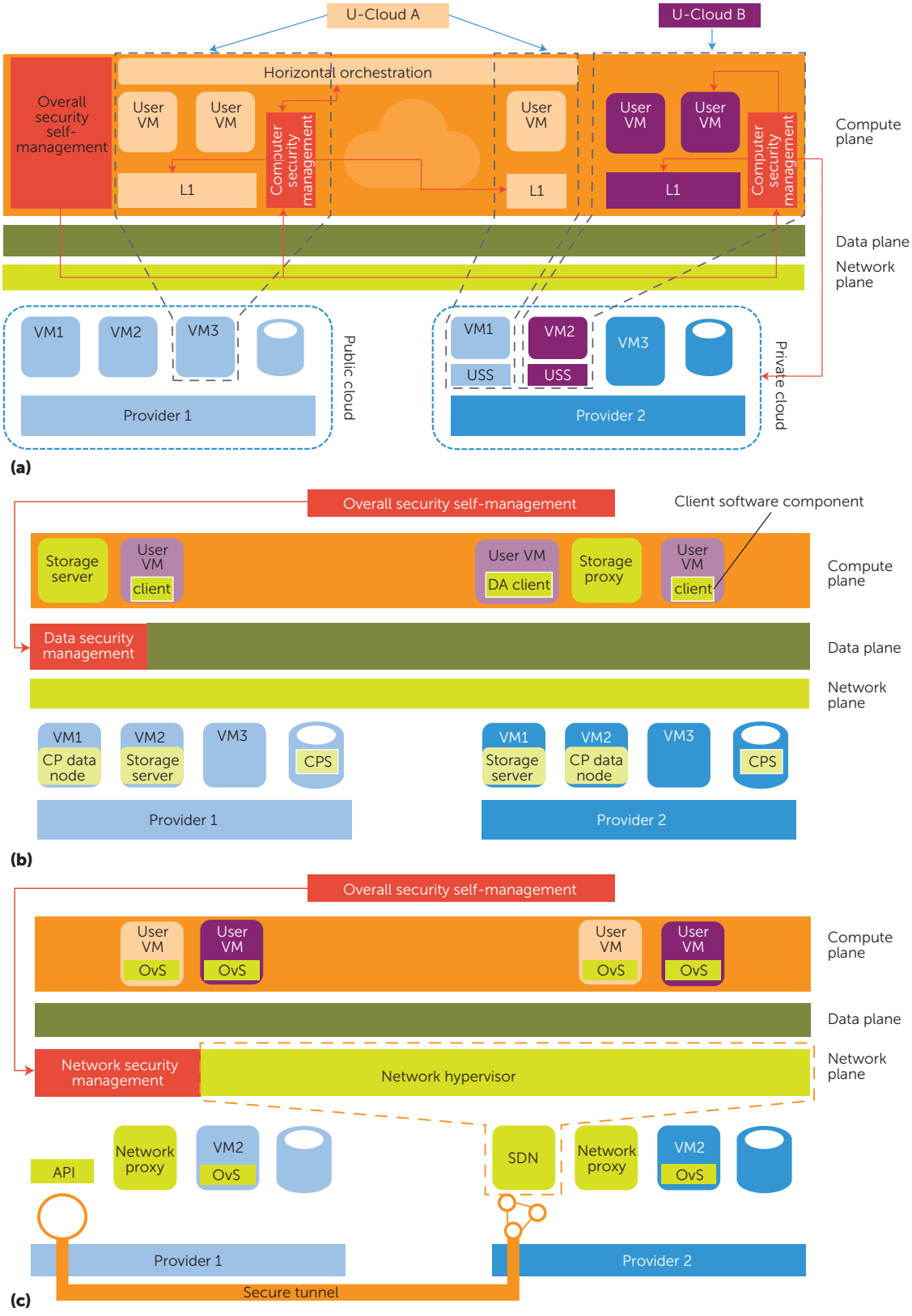

FIGURE 3. Detailed view of the Supercloud architecture: (a) computing plane, (b) data plane, and (c) network plane. Each figure shows detailed subcomponents for computation, data management, and networking, and interplay with security self-management. (OvS: Open vSwitch; SDN: software-defined network)

users of the Supercloud storage infrastructure. Ordinary clients interact transparently with the data plane via storage proxies. This requires minimal changes to clients without installing additional libraries. In contrast, *direct accessor clients* run Supercloud-specific logic as a client library and can interact and access storage servers and L1 cloud provider services directly. Direct accessor clients can also have certain features of storage servers built-in. Such clients could thus also be independent of storage servers.

Proxies, typically L2 VMs, facilitate client access to Supercloud storage and data management offerings, such as for encryption and secure deduplication. They're usually stateless and can be easily added dynamically to the system.

Servers, typically stateful L1 or L2 VMs, perform housekeeping of critical portions of metadata vital to the Supercloud data plane's operation, such as metadata for storage, data integrity, or configuration management. Cloud provider services (CPSs) are L1 cloud storage services that direct accessor clients or proxies can directly access. They expose different APIs, notably object storage and block storage. Examples include OpenStack Swift and Amazon's Simple Storage Service (S3) and Elastic Block Store. Cloud provider data nodes are L1 VMs in the distributed provider infrastructure. Complementing CPS, they can perform computation and have locally mounted L1 block storage for Supercloud user data.

Security self-management components allow arbitration between provider and user data security settings.

Network plane. Figure 3c illustrates the Supercloud network virtualization architecture. Its main design goals are network controllability; full network virtualization to guarantee isolation between users, while enabling them to use their desired addressing schemes and topologies; and VM snapshotting and migration for availability and flexibility.

To fulfill these objectives, the architecture leverages software-defined networking (SDN),[11] which provides logically centralized control over forwarding and configuration state of the software switches running in the Supercloud VMs. OpenFlow and Open vSwitch (OvS) technologies provide fine-grained control of packet forwarding and of switch configurations, respectively. Logical centralization of control facilitates isolation, for example, through flow rule redefinition at the network edge, with translation of physical to virtual events. Availability goals extend well-proven techniques to the multi-cloud setting.

For each user, a specific set of network applications that control the virtual network will run on top the Supercloud network hypervisor that maps the virtual and physical resources. These include an address translator (to offer L2 and L3 address virtualization), a topology abstraction module (for topology virtualization), and a resource isolation application (to slice network resources among tenants, such as switch CPU and forwarding tables). The network hypervisor controls and configures the OvS switches that are installed in all VMs. An SDN controller will establish secure connections with each OvS switch to control the forwarding plane.

The network hypervisor is built as an application that runs in the Supercloud SDN controller. Each cloud will host a specific VM, the network proxy, where secure tunnels are set up to all other clouds. In a distributed configuration, each proxy will host an instance of the SDN controller.

Security management is facilitated through the interplay of overall security self-management and network security management components, which enable Supercloud users to specify user-specific settings for network configurations inside their U-Clouds.

Dynamic Architecture

Figure 4 illustrates two typical workflows between some key Supercloud architecture components. User 1 interacts with its VM (u1VM*x*) through a set of APIs. Providers 1 and 2 host compute (VM*x*), networking (NVM*x*), and storage management (DVM*x*) VMs. Provider 1 also hosts a physical storage service. Supercloud considers a nested architecture—that is, u1VM*x* runs inside VM*x*.

Supercloud users interact through four interfaces to deploy their applications in the cloud. The network plane interface, typically the network hypervisor, interacts with the SDN controller and network proxies, hosted in the NVM*x* machines, to handle communication and establish secure tunnels with other clouds. The data plane interface, typically storage proxies, interacts with the providers' DVM*x* VMs to ensure access to the user's private data. The compute plane interface, typically the L1 hypervisor, interacts with providers' VM*x* machines to provide memory and CPU resources.

We describe the interfaces between Supercloud elements in several scenarios. The first scenario relates to requesting data from the cloud storage; the second relates to establishing communication between two VMs hosted in the Supercloud. The last example shows how the Supercloud security management interfaces enable to deploy security services.

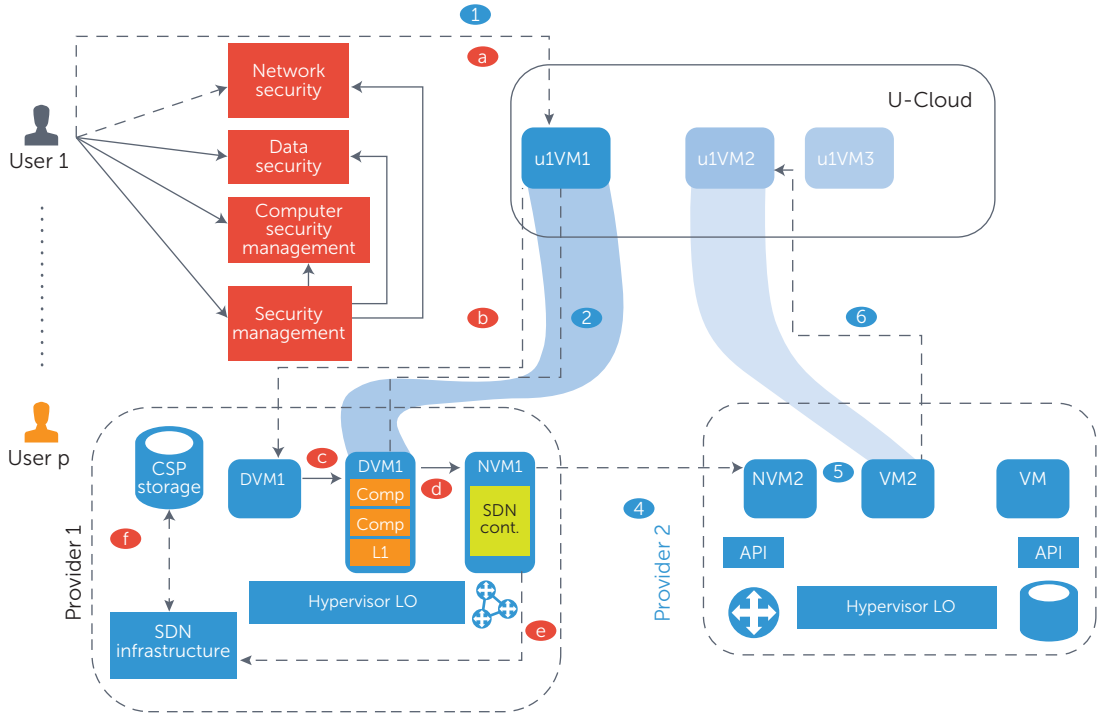

FIGURE 4. Sample Supercloud workflows. Shown are the Supercloud interfaces (computation, data, networking, self-management) to deploy applications in several scenarios: access to cloud storage, establishing communications between VMs, and self-management of security.

Access to cloud storage. In this scenario (steps a–f in Figure 4), during a request to the data layer (step a), the user VM (uVM) sends a request to the data management VM (DVM) (step b). The DVM is aware of the resource's physical location inside the cloud infrastructure. It provides this information to the hypervisor hosting the uVM (step c), which will ask the network management VM (NVM) (step d) to establish a connection between resource and uVM through the SDN network (steps e and f).

Establishing communication between user VMs. In this scenario (steps 1–6 in Figure 4), after receiving a request from User 1 (step 1), uVM1 sends a communication request through the hypervisor (step 2). The hypervisor forwards the request to the NVM (step 3), which establishes the SDN rules for the path to communicate with NVM2 (step 4). If the destination VM is hosted on a different CSP, the NVM forwards the request to the NVM of the other CSP, hence setting up the connection. Finally, NVM2 shares the request with VM2 (step 5), which is hosting uVM2 (step 6). Each component (VM*x*, DVM*x*, and NVM*x*) is accessed independently of the provider owning the physical resource.

Security management. Users and providers also interact with a security management plane interface (see Figures 2 and 3) to deploy, orchestrate, enforce, and monitor security requirements. Such requirements are specified and negotiated through SLAs during the cloud service discovery and brokering phases. This distributed protection plane is realized through interplay of several security self-management components spread across the Supercloud abstraction planes.

The resource management components are self-management agents (SMAs) responsible for delivering atomic security services such as enforcement, detection, reaction, and monitoring. These components operate on a particular architecture abstraction plane and are dedicated to a specific security service (such as intrusion detection, authorization enforcement, or trust management). Some security services might require multiple SMAs across multiple planes and/or providers. For instance, intrusion detection might require the collaboration of multiple (cross-provider) SMAs to collect, aggregate, and process activity logs.[12]

Aggregation components provide a unified and uniform view of multiple SMAs to the orchestrator. They abstract the heterogeneity of provider security

mechanisms, meeting platform independence and interoperability requirements.

Orchestration components are decision-making components providing security services. Each component is a manager for a specific security service such as authorization and access control, intrusion detection and prevention, and trust management. In addition, an overall orchestrator coordinates the actions of all security managers; a planner generates plans to reach and/or maintain security objectives; and a storage manager guarantees persistence and delivery of the knowledge needed for self-management of security. Orchestration components are also responsible for retrieving user security requirements from SLAs, converting them into policies and configurations to be enforced, and detecting and managing conflicts between tenants, users, and/or providers.

Use Cases

To illustrate how the Supercloud architecture can be mapped to real-world use cases, we use examples from the healthcare domain.

Hospital Imaging Archive

The amount of diagnostic imaging data is quickly increasing, imposing great challenges on hospital archive infrastructures, which must ensure high data availability, security, and regulatory compliance. A cloud-based solution can help address these challenges.

Such a solution's architecture should minimize the risk of security breaches and privacy violations, including unprivileged access to data (both at rest and during processing) with regard to defined policies. These policies might include hospital-specific policies context, legal country boundaries, and user groups. In terms of performance, robust data processing with low latency is desired, especially across different clouds.

Hospitals can store their clinical data as well as their imaging studies in on-premises private cloud storage. Archiving in the cloud helps simplify the data management and hospital archive infrastructure—especially due to high-volume imaging studies that are often as large as 1 Gbyte. Since on-premises storage can be limited, it makes sense to store this data securely in public cloud storage. For example, a hospital might store data from the last six months in the private cloud's on-premises storage, while storing older data (10 years or more) in the public cloud. Figure 5a shows a sample Supercloud implementation of such a solution.

In Figure 5a, three hospitals (A, B, and C) share a private cloud to store and manage their clinical data. A VM on the compute plane is dedicated to each hospital for its operations. Whenever a hospital VM wants to store or retrieve clinical data (for example, MRI imaging data), it communicates with a picture archiving and communication system (PACS) VM interfacing with the data plane. The data plane provides an abstraction to the user VM, making all underlying storage directly accessible (including encrypting stored data). Data older than six months is stored on the public cloud, whereas recent data is kept in the private cloud's on-premises storage, providing instantaneous access to it. Here, the network plane is responsible for handling all communications across different clouds and VMs. Hospitals can also define their security policies, such as how other hospitals can access their data. Components that are dedicated to data security and security management across the L1 hypervisor and compute VMs will prevent any unprivileged access to data based on security policies defined by each hospital.

Healthcare Laboratory Information System

This use-case demonstrates the impact of the Supercloud architecture for Maxdata Software, a healthcare software vendor that aims to deploy its software on the cloud as SaaS while enforcing the security requirements of different healthcare institutions.

The CLINIdATA®LIS healthcare laboratory information system (LIS) is a cross-platform Web application in which server components can run on any common operating system and relational database. The CLINIdATA®LIS must integrate with dozens of other clinical and nonclinical information systems (such as intensive care units, patient identification, billing, and regional health portals). It includes a set of real-time interfaces with physical electronic equipment (automated analyzers). The solution consists of three components on the server side: a stateless application, a database engine, and database data. The Supercloud approach allows each healthcare institution to define the U-Cloud that best fits its needs. Concrete deployment on physical cloud providers is then automated. The considered setting is a large hospital cluster that employs thousands of professionals, processes tens of millions of transactions per day, and is located in a country where personal data protection must be guaranteed.

In a typical U-Cloud specification,

- the application and database engine are replicated across several VMs on the compute plane (fault tolerance and load balancing);
- data is split among different storage nodes in the data plane (offering confidentiality, even if one storage node is compromised);

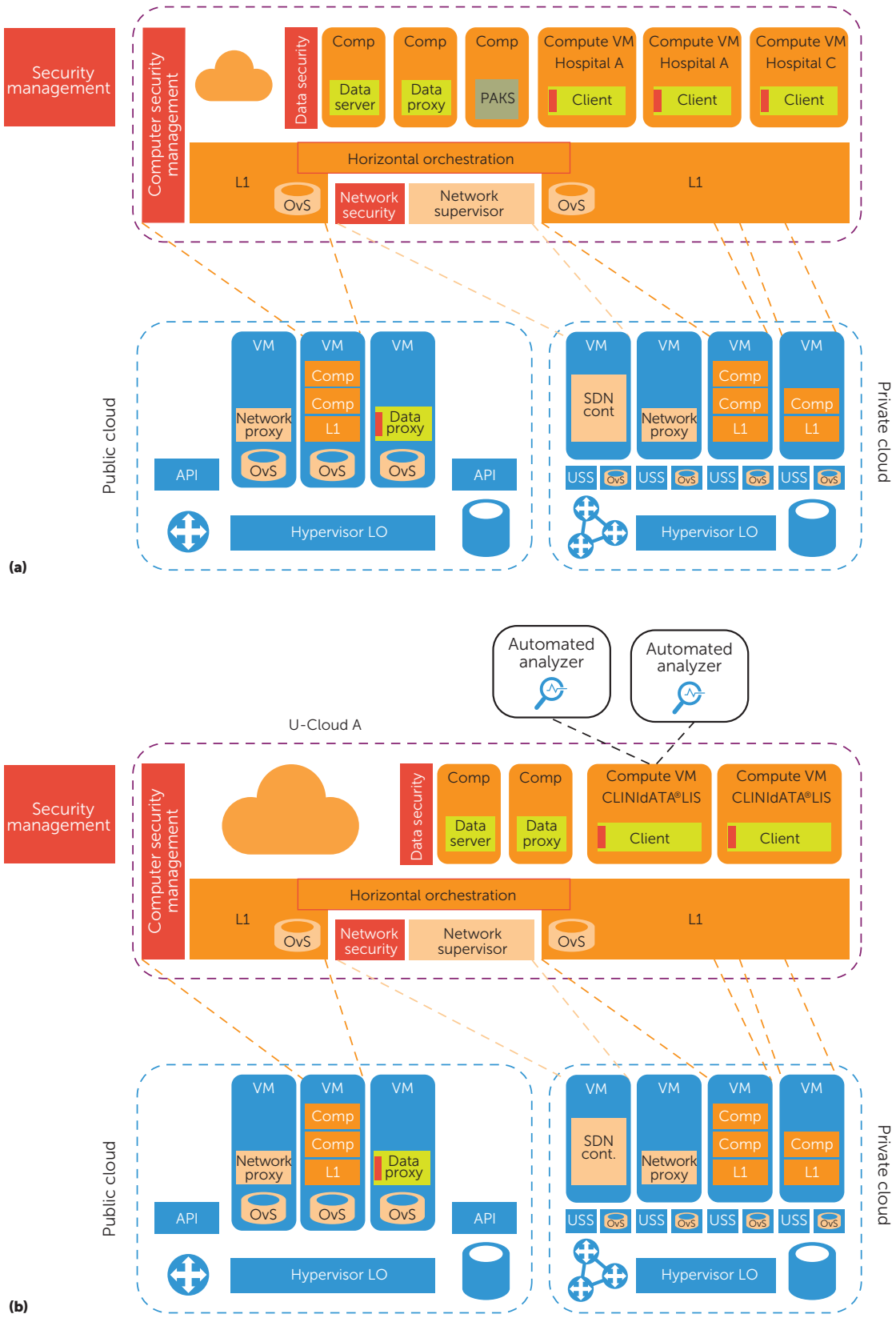

FIGURE 5. Supercloud practical deployments: (a) high-availability storage and disaster recovery, and (b) healthcare laboratory information system software as a service (SaaS). (USS: user-centric system service)

- a set of networks connect application VMs to automated analyzers running on hospital premises and to database engine VMs, which in turn are connected to storage nodes;
- VMs on the compute plane ensure confidentiality, integrity, and 99.99 percent availability;
- storage nodes on the data plane ensure data integrity and 99.99 percent availability; and
- data may be processed and stored only in a predefined set of countries.

As Figure 5b shows, a Supercloud infrastructure can then deploy the VMs on a trusted private cloud to ensure confidentiality on the compute plane, instantiate the storage nodes on a set of public cloud providers running security mechanisms (such as encryption and secret sharing) to ensure confidentiality, and connect the different components using virtual networks provided by the network plane. Deployments consider the locations or countries specified by the healthcare institution. Replicated instances of the CLINIdATA®LIS application run on VMs on the compute plane. These instances then connect to the database engine running on a different VM linked with the data plane.

In case of regulatory, economic, or other type of change, healthcare institutions can update U-Cloud requirements and/or features. The Supercloud infrastructure automatically redeploys the solution accordingly, enabling quick adaptation to context changes. It also prevents vendor lock-in.

We're implementing the different components of the Supercloud architecture to gradually achieve integrated proof of concepts. The solution is currently at an advanced stage of implementation. Several results are already available (see https://Supercloud-project.eu/publications-deliverables). However, we're still integrating the various components. Preliminary performance results have shown relatively modest overheads, giving good indications about the potential for the solution (such as for network virtualization[13]). Our next step is to validate the approach through testbed integration. Other foreseen applications domains include network function virtualization or smart home security. Results will be disseminated to promote open source cloud technologies and will be contributed to major standardization bodies.

Acknowledgments

This work is supported by the European Union Supercloud Project (Horizon 2020 Research and Innovation Program, grant 644962) and by the Swiss Secretariat for Education, Research and Innovation (contract 15.0091). It is based on contributions from the entire Supercloud consortium.

References

1. F. Bonomi et al., "Fog Computing and Its Role in the Internet of Things," *Proc. 1st Workshop Mobile Cloud Computing* (MCC), 2012, pp. 13–16.
2. F. Manco et al., "The Case for the Superfluid Cloud," *Proc. 7th USENIX Workshop Hot Topics in Cloud Computing* (HotCloud), 2015.
3. L. Zheng et al., "How to Bid the Cloud," *Proc. ACM Conf. Special Interest Group on Data Comm.* (SIGCOMM), 2015, pp. 71–84.
4. R. Los, D. Shackleford, and B. Sullivan, *Notorious Nine Cloud Computing Top Threats in 2013*, tech. report, Cloud Security Alliance, 2013.
5. D. Sgandurra and E. Lupu, "Evolution of Attacks, Threat Models, and Solutions for Virtualized Systems," *ACM Computing Surveys*, vol. 48, no. 3, 2016, pp. 1–38.
6. D. Williams, H. Jamjoom, and H. Weatherspoon, "Plug into the Supercloud," *IEEE Internet Computing*, vol. 17, no. 2, 2013, pp. 28–34.
7. A. Ludwig and S. Schmid, "Distributed Cloud Market: Who Benefits from Specification Flexibilities?" *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 43, no. 3, 2015, pp. 38–41.
8. K. Bernsmed et al., "Thunder in the Clouds: Security Challenges and Solutions for Federated Clouds," *Proc. IEEE 4th Int'l Conf. Cloud Computing Technology and Science* (CloudCom), 2012, doi:10.1109/CloudCom.2012.6427547.
9. M. Ben-Yehuda et al., "The Turtles Project: Design and Implementation of Nested Virtualization," *Proc. 9th USENIX Conf. Operating Systems Design and Implementation* (OSDI), vol. 10, 2010, pp. 423–436.
10. P. Watson, "Application Security through Federated Clouds," *IEEE Cloud Computing*, vol. 1, no. 3, 2014, pp. 76–80.
11. D. Kreutz et al., "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, 2015, pp. 14–76.
12. S.T. Zargar et al., "DCDIDP: A Distributed, Collaborative, and Data-Driven Intrusion Detection and Prevention Framework for Cloud Computing Environments," *Proc. 7th Int'l Conf. Collaborative Computing: Networking, Applications and Worksharing* (CollaborateCom), 2011, pp. 332–341.
13. M. Alaluna, F. Ramos, and N. Ferreira Neves, "(Literally) above the Clouds: Virtualizing the

Network over Multiple Clouds," *Proc. IEEE Conf. Network Softwarization* (NetSoft), 2016, pp. 112–115.

MARC LACOSTE *is a senior research scientist in the Security Department of Orange Labs, and technical leader of the H2020 Supercloud Project. His main research interests include security architecture, cloud computing security, self-protecting systems, and open security kernels. Lacoste has a PhD in computer science from the University of Grenoble, France. He's a member of ACM. Contact him at marc.lacoste @orange.com.*

MARKUS MIETTINEN *is a researcher in the System Security Lab at the Department of Computer Science at Technische Universität Darmstadt, Germany. His research interests include contextual security, data analysis-based security enablers, and security management in new computation environments such as the Internet of Things. Miettinen has an MSc in computer science from the University of Helsinki, Finland. Contact him at markus.miettinen@trust.tu-darmstadt.de.*

NUNO NEVES *is an associate professor in and head of the University of Lisbon's Department of Computer Science, where he leads the Navigators research group and is on the executive board of the Large-Scale Informatics Systems Laboratory (LaSIGE) research unit. His research interests include the security and dependability aspects of distributed systems. Neves has a PhD in computer science from University of Illinois Urbana-Champaign. Contact him at nuno@di.fc.ul.pt.*

FERNANDO M.V. RAMOS *is an assistant professor in the Department of Computer Science the University of Lisbon. His research interests include network programmability and network virtualization. Ramos has a PhD in computer science and engineering from the University of Cambridge, UK. Contact him at fvramos@ciencias.ulisboa.pt.*

MARKO VUKOLIĆ *is a research staff member at IBM Research, Zurich. His research interests are in distributed algorithms and systems, including fault-tolerance, blockchain and distributed ledgers, cloud computing security, and distributed storage. Vukolić has a PhD in distributed systems from Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Contact him at mvu@zurich.ibm.com.*

FABIEN CHARMET *is a research engineer in the Samovar Lab (Centre National de la Recherche Scientifique) in Télécom SudParis, Institut Mines-Télécom. His research interests include penetration testing, network security, and software-defined networking. Charmet has an MSc in enterprise architecture and an MSc in network and security from the University of Lille 1, France. Contact him at fabien.charmet@telecom-sudparis.*

REDA YAICH *is a researcher in the LabSTICC Laboratory (Centre National de la Recherche Scientifique) in Télécom Bretagne, Institut Mines-Telecom. His research interests include the specification and enforcement of trust and security policies over open, distributed, and decentralized systems. Yaich has a PhD in computer science from Ecole des Mines of Saint-Etienne, France. Contact him at reda.yaich @telecom-bretagne.eu.*

KRZYSZTOF OBORZYŃSKI *is a software architect at Healthcare Informatics Services and Solutions, Clinical Platforms, Philips Healthcare. His research interests include systems reliability, performance, and serviceability. Oborzyński has a PhD in computer science at the Institute of Computing Science, Poznań University of Technology, Poland. Contact him at Krzysztof.Oborzynski@philips.com.*

GITESH VERNEKAR *is a senior manager at Healthcare Informatics Services and Solutions, HealthSuite Digital Platform, Philips Healthcare. His research interests include delivering innovative and repeatable software solutions in healthcare, banking, and high-tech industries, and technology-enabled operations and services. Vernekar has an international MBA in entrepreneurship and strategy management from the Rotterdam School Management, Erasmus University, The Netherlands. Contact him at Gitesh.Vernekar @philips.com.*

PAULO SOUSA *is chief executive officer at Maxdata Software. His research interests include real-time systems, intrusion tolerance, and security. Sousa has a PhD in computer science from the University of Lisbon. Contact him at paulo.sousa@maxdata.pt.*