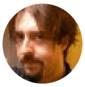
An Architecture for Federated Cloud Computing

CHAPTER · JANUARY 2011

DOI: 10.1002/9780470940105.ch15

14 AUTHORS, INCLUDING:

Alex Galis
University College London
159 PUBLICATIONS **1,046** CITATIONS



Fermín Galán
Telefónica, S.A
41 PUBLICATIONS **668** CITATIONS

CHAPTER 15

AN ARCHITECTURE FOR FEDERATED CLOUD COMPUTING

BENNY ROCHWERGER, CONSTANTINO VÁZQUEZ, DAVID BREITGAND, DAVID HADAS, MASSIMO VILLARI, PHILIPPE MASSONET, ELIEZER LEVY, ALEX GALIS, IGNACIO M. LLORENTE, RUBÉN S. MONTERO, YARON WOLFSTHAL, KENNETH NAGIN, LARS LARSSON, and FERMÍN GALÁN

15.1 INTRODUCTION

Utility computing, a concept envisioned back in the 1960s, is finally becoming a reality. Just as we can power a variety of devices, ranging from a simple light bulb to complex machinery, by plugging them into the wall, today we can satisfy, by connecting to the Internet, many of our computing needs, ranging from full pledge productivity applications to raw compute power in the form of virtual machines. Cloud computing [1], in all its different forms, is rapidly gaining momentum as an alternative to traditional IT, and the reasons for this are clear: In principle, it allows individuals and companies to fulfill all their IT needs with minimal investment and controlled expenses (both capital and operational).

Cloud computing enables companies and individuals to lease resources on-demand from a virtually unlimited pool. The "pay as you go" billing model applies charges for the actually used resources per unit time. This way, a business can optimize its IT investment and improve availability and scalability.

While cloud computing holds a lot of promise for enterprise computing, there are a number of inherent deficiencies in current offerings such as:

Cloud Computing: Principles and Paradigms, Edited by Rajkumar Buyya, James Broberg and Andrzej Goscinski Copyright © 2011 John Wiley & Sons, Inc.

- **Inherently Limited Scalability of Single-Provider Clouds.** Although most infrastructure cloud providers today claim infinite scalability, in reality it is reasonable to assume that even the largest players may start facing scalability problems as cloud computing usage rate increases. In the long term, scalability problems may be expected to worsen as cloud providers serve an increasing number of on-line services, each accessed by massive amounts of global users at all times.
- **Lack of Interoperability Among Cloud Providers.** Contemporary cloud technologies have not been designed with interoperability in mind. This results in an inability to scale through business partnerships across clouds providers. In addition, it prevents small and medium cloud infrastructure providers from entering the cloud provisioning market. Overall, this stifles competition and locks consumers to a single vendor.
- **No Built-In Business Service Management Support.** Business Service Management (BSM) is a management strategy that allows businesses to align their IT management with their high-level business goals. The key aspect of BSM is service-level agreement (SLA) management. Current cloud computing solutions are not designed to support the BSM practices that are well established in the daily management of the enterprise IT departments. As a result, enterprises looking at transforming their IT operations to cloud-based technologies face a non-incremental and potentially disruptive step.

To address these issues, we present in this chapter a model for business-driven federation of cloud computing providers, where each provider can buy and sell, on-demand, capacity from other providers (see Figure 15.1).

In this chapter we analyze the requirements for an enterprise-grade cloud computing offering and identify the main functional components that should be part of such offering. In addition, we develop from the requirement the basic principles that we believe are the cornerstone of future cloud computing offerings. The remainder of this chapter is organized as follows: In Section 15.2 we will present use cases and requirements, and in Section 15.3 we expand on the principles of cloud computing derived from these requirements. In Section 15.4 we will present a model for federated cloud computing infrastructure and provide definitions of the concepts used and in Section 15.5 we describe the seurity considerations for such system. We conclude with a summary in Section 15.6.

15.2 A TYPICAL USE CASE

As a representative of an enterprise-grade application, we have chosen to analyze SAP™ systems and to derive from them general requirements that such application might have from a cloud computing provider.

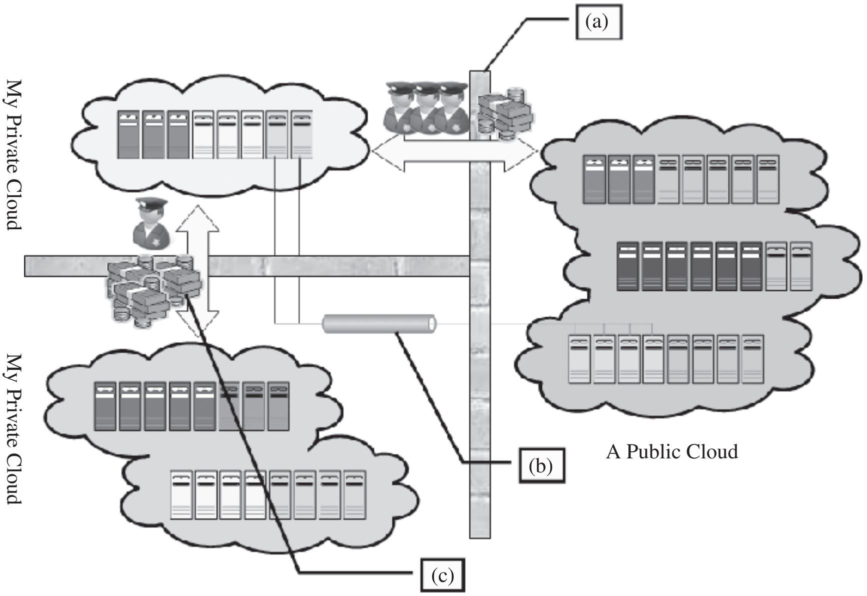


FIGURE 15.1. Model for federated cloud computing: (a) Different cloud providers collaborate by sharing their resources while keeping thick walls in between them; that is, each is an independent autonomous entity. (b) Applications running in this cloud of clouds should be unaware of location; that is, virtual local networks are needed for the inter-application components to communicate. (c) Cloud providers differentiate from each in terms of cost and trust level; for example, while a public cloud maybe cheap, companies will be reluctant to put in there sensitive services.

15.2.1 SAP Systems

SAP systems are used for a variety of business applications that differ by version and functionality [such as customer relationship management (CRM) and enterprise resource planning (ERP)]. For a given application type, the SAP system components consist of generic parts customized by configuration and parts custom-coded for a specific installation. Certain SAP applications are composed of several loosely coupled systems. Such systems have independent databases and communicate asynchronously by message with each other.

An SAP system is a typical three-tier system (see Figure 15.2) as follows:

- Requests are handled by the SAP Web dispatcher.
- In the middle tier, there are two types of components: multiple stateful dialog instances (DIs) and a single central instance (CI) that performs central services such as application-level locking, messaging, and registration of DIs. The number of DIs can be changed while the system is running to adapt to load.
- A single database management system (DBMS) serves the SAP system.

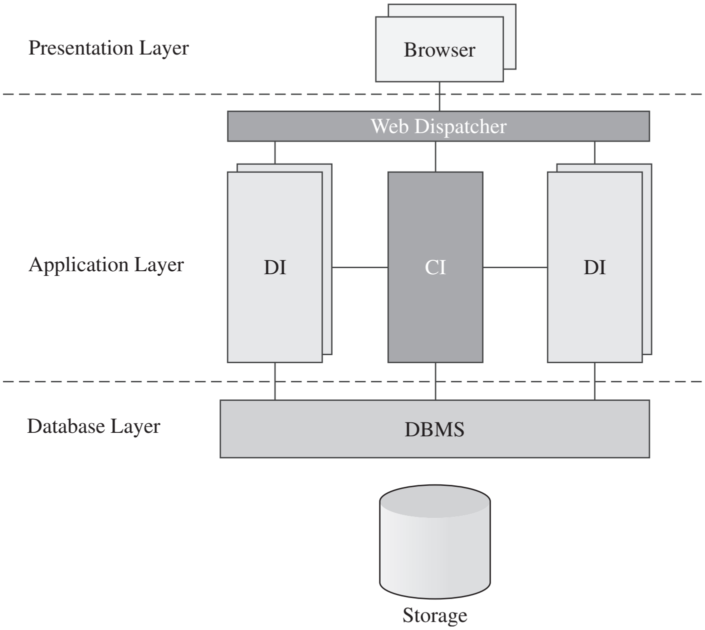


FIGURE 15.2. Abstraction of an SAP system.

The components can be arranged in a variety of configurations, from a minimal configuration where all components run on a single machine, to larger ones where there are several DIs, each running on a separate machine, and a separate machine with the CI and the DBMS (see Figure 15.3)

15.2.2 The Virtualized Data Center Use Case

Consider a data center that consolidates the operation of different types of SAP applications and all their respective environments (e.g., test, production) using virtualization technology. The applications are offered as a service to external customers, or, alternatively, the data center is operated by the IT department of an enterprise for internal users (i.e., enterprise employees).

A special variation that deserves mentioning is when the data center serves an on-demand, Software as a Service (SaaS) setup, where customers are external and where each customer (tenant) gets the same base version of the application. However, each tenant configures and customizes the application to suit his specific needs. It is reasonable to assume that a tenant in this case is a small or medium business (SMB) tenant.

We briefly mention here a few aspects that are typical of virtualized data centers:

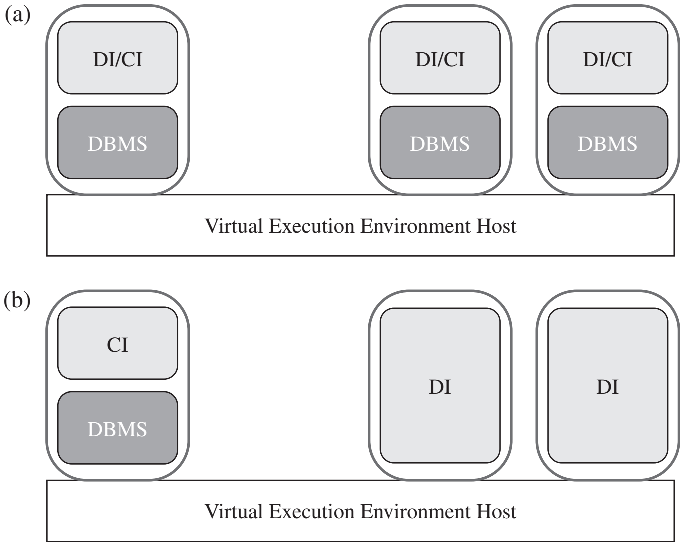

(a)
DI/CI
DBMS
DI/CI
DBMS
DI/CI
DBMS
Virtual Execution Environment Host
(b)
CI
DBMS
DI
DI
Virtual Execution Environment Host

FIGURE 15.3. Sample SAP system deployments. (a) All components run in the same virtual execution environment (represented as rounded rectangles); (b) the large components (CI and DBMS) run each on a dedicated virtual execution environment. The virtual execution environment host refers to the set of components managing the virtual environments.

- The infrastructure provider must manage the life cycle of the application for hundreds or thousands of tenants while keeping a very low total cost of ownership (TCO). This includes setting up new tenants, backing up the databases, managing the customizations and configurations of tenants, and getting patches and newer versions of the software from SAP (the service provider).
- Setting up a new tenant in the SaaS for SMBs case is completely automated by a Web-based wizard. The new tenant runs through a series of configuration questions and uploads master data items (e.g., product catalog and customer lists). Following these steps, the tenant is up and running, typically using a trial version. The provisioning of the resources (storage, database, and application server) is part of this automated setup.
- The customers are billed a fixed monthly subscription fee or a variable fee based on their usage of the application.
- There are several well-known approaches to multi-tenancy of the same database schema [2]. Regardless of the approach taken, multi-tenancy calls for flexible virtualization schemes where, for example, the DBMS component and the storage system are shared between multiple tenants. The main reason for this sharing is to keep the TCO per tenant at a minimum.

In summary, the key challenges in all these use cases from the point of view of the infrastructure provider are:

- Managing thousands of different service components that comprise a variety of service applications executed by thousands of virtual execution environments, on top of a complex infrastructure that also includes network and storage systems.
- Consolidating many applications on the same infrastructure, thereby increasing HW utilization and optimizing power consumption, while keeping the operational cost at minimum.
- Guaranteeing the individual SLAs of the many customers of the data center who face different and fluctuating workloads.

15.2.3 Primary Requirements

From the use case discussed in the previous section, we derived the following main requirements from a cloud computing infrastructure:

- **Automated and Fast Deployment.** The cloud should support automated provisioning of complex service applications based on a formal contract specifying theinfrastructure SLAs. The same contract should be reused to provision multiple instances of the same application for different tenants with different customizations.
- **Dynamic Elasticity.** The cloud should dynamically adjust resource allocation parameters (memory, CPU, network bandwidth, storage) of individual virtual execution environments seamlessly. Moreover, the number of virtual execution environments must be dynamically and seamlessly adjusted to adapt to the changing load.
- **Automated Continuous Optimization.** The cloud should continuously optimize alignment of infrastructure resources management with the high-level business goals.

15.3 THE BASIC PRINCIPLES OF CLOUD COMPUTING

In this section we unravel a set of principles that enable Internet scale cloud computing services. We seek to highlight the fundamental requirement from the providers of cloud computing to allow virtual applications to freely migrate, grow, and shrink.

15.3.1 Federation

All cloud computing providers, regardless of how big they are, have a finite capacity. To grow beyond this capacity, cloud computing providers should be able to form federations of providers such that they can collaborate and share their resources. The need for federation-capable cloud computing offerings is also derived from the industry trend of adopting the cloud computing paradigm internally within companies to create *private clouds* and then being able to extend these clouds with resources leased on-demand from *public clouds*.

Any federation of cloud computing providers should allow virtual application to be deployed across federated sites. Furthermore, virtual applications need to be completely location free and allowed to migrate in part or as a whole between sites. At the same time, the security privacy and independence of the federation members must be maintained to allow competing providers to federate.

15.3.2 Independence

Just as in other utilities, where we get service without knowing the internals of the utility provider and with standard equipment not specific to any provider (e.g., telephones), for cloud computing services to really fulfill the computing as a utility vision, we need to offer cloud computing users full independence. Users should be able to use the services of the cloud without relying on any provider-specific tool, and cloud computing providers should be able to manage their infrastructure without exposing internal details to their customers or partners.

As a consequence of the independence principle, all cloud services need to be encapsulated and generalized such that users will be able to acquire equivalent virtual resources at different providers.

15.3.3 Isolation

Cloud computing services are, by definition, hosted by a provider that will simultaneously host applications from many different users. For these users to move their computing into the cloud, they need warranties from the cloud computing provider that their stuff is completely isolated from others. Users must be ensured that their resources cannot be accessed by others sharing the same cloud and that adequate performance isolation is in place to ensure that no other user may possess the power to directly effect the service granted to their application.

15.3.4 Elasticity

One of the main advantages of cloud computing is the capability to provide, or release, resources on-demand. These "elasticity" capabilities should be enacted automatically by cloud computing providers to meet demand variations, just as electrical companies are able (under normal operational circumstances) to automatically deal with variances in electricity consumption levels. Clearly the behavior and limits of automatic growth and shrinking should be driven by contracts and rules agreed on between cloud computing providers and consumers.

The ability of users to grow their applications when facing an increase of real-life demand need to be complemented by the ability to scale. Cloud computing services as offered by a federation of infrastructure providers is expected to offer any user application of any size the ability to quickly scale up its application by unrestricted magnitude and approach Internet scale. At the same time, user applications should be allowed to scale down facing decreasing demand. Such scalability although depended on the internals of the user application is prime driver for cloud computing because it help users to better match expenses with gain.

15.3.5 Business Orientation

Before enterprises move their mission critical applications to the cloud, cloud computing providers will need to develop the mechanisms to ensure quality of service (QoS) and proper support for service-level agreements (SLAs). More than ever before, cloud computing offers challenges with regard to the articulation of a meaningful language that will help encompass business requirements and that has translatable and customizable service parameters for infrastructure providers.

15.3.6 Trust

Probably the most critical issue to address before cloud computing can become the preferred computing paradigm is that of establishing trust. Mechanisms to build and maintain trust between cloud computing consumers and cloud computing providers, as well as between cloud computing providers among themselves, are essential for the success of any cloud computing offering.

15.4 A MODEL FOR FEDERATED CLOUD COMPUTING

In our model for federated cloud computing we identify two major types of actors: *Service Providers (SPs)* are the entities that need computational resources to offer some service. However, SPs do not own these resources; instead, they lease them from *Infrastructure Providers (IPs)*, which provide them with a seemingly infinite pool of computational, network, and storage resources.

A *Service Application* is a set of software components that work collectively to achieve a common goal. Each component of such service applications executes in a dedicated VEE. SPs deploy service applications in the the cloud by providing to a IP, known as the *primary site*, with a *Service Manifest*—that is, a document that defines the structure of the application as well as the contract and SLA between the SP and the IP.

To create the illusion of an infinite pool of resources, IPs shared their unused capacity with each other to create a *federation cloud.* A *Framework Agreement*

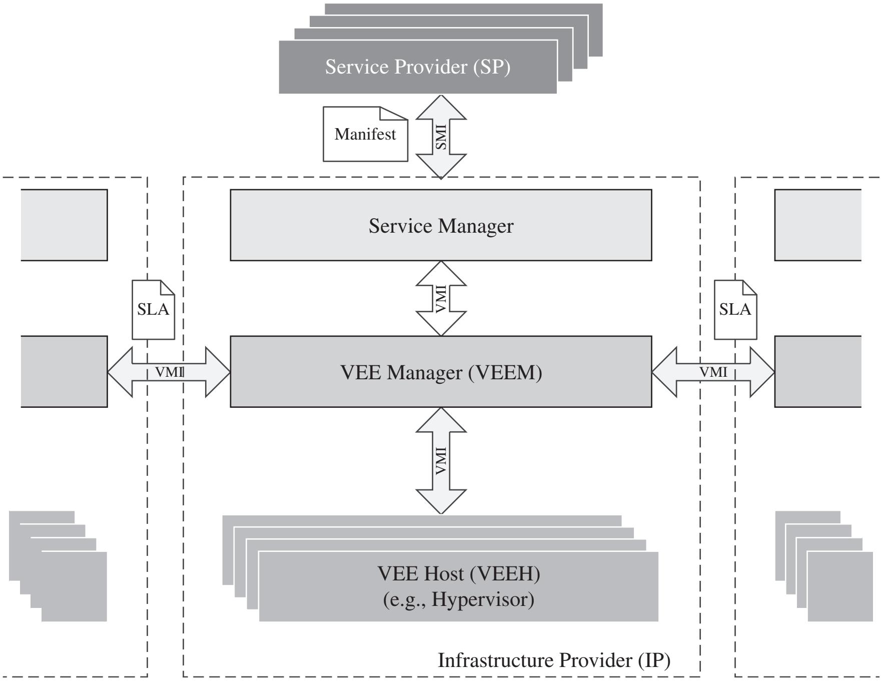


FIGURE 15.4. The RESERVOIR architecture: major components and interfaces.

is document that defines the contract between two IPs—that is, it states the terms and conditions under which one IP can use resources from another IP.

Within each IP, optimal resource utilization is achieved by partitioning physical resources, through a virtualization layer, into *Virtual Execution Environments (VEEs)*—fully isolated runtime environments that abstract away the physical characteristics of the resource and enable sharing. We refer to the virtualized computational resources, alongside the virtualization layer and all the management enablement components, as the *Virtual Execution Enviroment Host (VEEH)*.

With these concepts in mind, we can proceed to define a reference architecture for federated cloud computing. The design and implementation of such architecure are the main goals of the RESERVOIR European research project. The RESERVOIR architecture [3], shown in Figure 15.4, identifies the major functional components needed within an IP to fully support the cloud computing paradigm. The rationale behind this particular layering is to keep a clear separation of concerns and responsibilities and to hide low-level infrastructure details and decisions from high-level management and service providers.

- **The Service Manager** is the only component within an IP that interacts with SPs. It receives Service Manifests, negotiates pricing, and handles billing. Its two most complex tasks are (1) deploying and provisioning VEEs based on the Service Manifest and (2) monitoring and enforcing SLA compliance by throttling a service application's capacity.

- **The Virtual Execution Environment Manager (VEEM)** is responsible for the optimal placement of VEEs into VEE Hosts subject to constraints determined by the Service Manager. The continuous optimization process is driven by a site-specific programmable utility function. The VEEM is free to place and move VEEs anywhere, even on the remote sites (subject to overall cross-site agreements), as long as the placement satisfies the constraints. Thus, in addition to serving local requests (from the local Service Manager), VEEM is responsible for the federation of remote sites.
- **The Virtual Execution Environment Host (VEEH)** is responsible for the basic control and monitoring of VEEs and their resources (e.g., creating a VEE, allocating additional resources to a VEE, monitoring a VEE, migrating a VEE, creating a virtual network and storage pool, etc.). Given that VEEs belonging to the same application may be placed on multiple VEEHs and even extend beyond the boundaries of a site, VEEHs must support isolated virtual networks that span VEEHs and sites. Moreover, VEEHs must support transparent VEE migration to any compatible VEEH within the federated cloud, regardless of site location or network and storage configurations.

The layered design stresses the use of standard, open, and generic protocols and interfaces to support vertical and horizontal interoperability between layers. Different implementations of each layer will be able to interact with each other. The Service Management Interface (SMI) with its service manifest exposes a standardized interface into the RESERVOIR cloud for service providers. The service provider may then choose among RESERVOIR cloud providers, knowing that they share a common language to express their business requirements. The VEE Management Interface (VMI) simplifies the introduction of different and independent IT optimization strategies without disrupting other layers or peer VEEMs. Furthermore, VMI's support of VEEM-to-VEEM communication simplifies cloud federation by limiting the horizontal interoperability to one layer of the stack. The VEE Host Interface (VHI) will support plugging-in of new virtualization platforms (e.g., hypervisors), without requiring VEEM recompilation or restart. RESERVOIR's loosely coupled stack reference architecture should promote a variety of innovative approaches to support cloud computing.

15.4.1 Features of Federation Types

Federations of clouds may be constructed in various ways, with disparate feature sets offered by the underlying implementation architecture. This section is devoted to present these differentiating features. Using these features as a base, a number of federation scenarios are defined, comprised of subsets of this feature set.

The first feature to consider is the framework agreement support: Framework agreements, as defined in the previous section, may either be supported by the architecture or not. If framework agreements are not supported, this implies that federation may only be carried out in a more ad hoc opportunistic manner. Another feature is the *opportunistic placement support*. If framework agreements are not supported by the architecture, or if there is not enough spare capacity even including the framework agreements, a site may choose to perform opportunistic placement. It is a process where remote sites are queried on-demand as the need for additional resources arises, and the local site requests a certain SLA-governed capacity for a given cost from the remote sites.

One interesting feature to take into account is the *advance resource reservation support*. This feature may be used both when there is an existing framework agreement and when opportunistic placement has been performed. Both types of advance reservations are only valid for a certain time, since they impact the utilization of resources at a site. Because of this impact, they should be billed as actual usage during the active time interval.

The ability to migrate machines across sites defines the *federated migration support*. There are two types of migration: cold and hot (or live). In cold migration, the VEE is suspended and experiences a certain amount of downtime while it is being transferred. Most modern operating systems have support for being suspended, which includes saving all RAM contents to disk and later restoring the runtime state to its prior state. Hot or live migration does not allow for system downtime, and it works by transferring the runtime state while the VEE is still running.

Focusing on networks, there can be *cross-site virtual network support*: VEEs belonging to a service are potentially connected to virtual networks, should this be requested by the SP. Ideally, these virtual networks will span across sites. However, this requires substantial effort and advanced features of the underlying architecture. In the same line, the federation can offer *public IP addresses retention post cross-site migration*. With fully virtualized networks, this may be a directly supported feature; but even if virtualized networks are not available, it may still be possible to maintain public IP addresses by manipulating routing information.

Information disclosure within the federation has also to be taken into account. The sites in the federation may provide information to different degrees (for instance, the information exchange between sites may be larger within the same administrative domain than outside it). Information regarding deployed VEEs will be primarily via the monitoring system, whereas some information may also potentially be exposed via the VMI as response to a VEE deployment request.

The last identified feature useful to define scenario is the *VMI operation support*: Depending on the requirements of the federation scenario, only a subset of the VMI operations may be made available. Which operations are required may be related to the amount of information that is exposed by the remote sites; access to more information may also increase the possibility and need to manipulate the deployed VEEs.

15.4.2 Federation Scenarios

In this section, a number of federation scenarios are presented, ranging from a baseline case to a full-featured federation. These scenarios have various requirements on the underlying architecture, and we use the features presented in previous section as the basis for differentiating among them.

The *baseline federation* scenario provides only the very basic required for supporting opportunistic placement of VEEs at a remote site. Migration is not supported, nor does it resize the VEEs once placed at the remote site. Advanced features such as virtual networks across site boundaries are also not supported. The baseline federation should be possible to build on top of most public cloud offerings, which is important for interoperability. The *basic federation* scenario includes a number of features that the baseline federation does not, such as framework agreements, cold migration, and retention of public IP addresses. Notably missing is (a) support for hot migration and (b) cross-site virtual network functionality. This scenario offers a useful cloud computing federation with support for site collaboration in terms of framework agreements without particularly high technological requirements on the underlying architecture in terms of networking support. The *advanced federation* scenario offers advanced functionality such as cross-site virtual network support. The feature most notably missing is hot migration, and the monitoring system also does not disclose VEE substate metadata information. The *full-featured* federation scenario offers the most complete set of features, including hot migration of VEEs.

15.4.3 Layers Enhancement for Federation

Taking into account the different types of federation, a summary of the features needed in the different layers of the RESERVOIR architecture to achieve federation is presented.

Service Manager. The *baseline federation* is the most basic federation scenario, but even here the SM must be allowed to specify placement restrictions when a service is deployed. Deployment restrictions are associated to an specific VEE (although the restriction expression could involve other VEEs, as can be seen in the affinity restrictions above) and passed down to the VEEM along with any other specific VEE metadata when the VEE is issued for creation through VMI. They specify a set of constraints that must be held when the VEE is created, so they can be seen as some kind of "contour conditions" that determine the domain that can be used by the placement algorithm run at VEEM layer. Two kinds of deployment restrictions are envisioned: First, there are *affinity restrictions*, related to the relations between VEEs; and second, there can be *site restrictions*, related to sites.

In the *basic federation* scenario, federation uses framework agreement (FA) between organizations to set the terms and conditions for federation. Framework agreements are negotiated and defined by individuals, but they are encoded at the end in the service manager (SM)—in particular, within the business information data base (BIDB). The pricing information included in the FA is used by the SM to calculate the cost of resources running in remote systems (based on the aggregated usage information that it received from the local VEEM) and correlate this information with the charges issued by those remote sites. The SM should be able to include as part of the VEE metadata a "price hint vector" consisting on a sequence of numbers, each one representing an estimation of the relative cost of deploying the VEE on each federated site. The SM calculate this vector based on the FA established with the other sites.

Given that the *advanced federation* scenario supports migration, the placement restrictions have to be checked not only at service deployment time but also for migration. In addition, the SM could update the deployment restrictions during the service lifespan, thereby changing the "contour conditions" used by the placement algorithm. When the VEE is migrated across sites, its deployment restrictions are included along with any other metadata associated with the VEE. On the other hand, no additional functionality is needed from the service manager to implement the *full-featured federation*.

Virtual Execution Environment Manager. Very little is needed in the *baseline federation* scenario of the VEEM. The only requirement will be the ability to deploy a VEE in the remote site, so it will need a plug-in that can communicate with the remote cloud by invoking the public API. This will satisfy the opportunistic placement requirement. For the different features offered by the *basic federation* scenario, the VEEM will need framework agreement, since it is necessary that the VEEM implement a way to tell whether it can take care of the VEE or not, attending to the SLAs defined in the framework agreement. The best module in the VEEM for the SLA evaluation to take place is the admission control of the policy engine. Also, cold migration is needed; therefore the VEEM needs the ability to signal the hypervisor to save the VEE state (this is part of the VEEM life-cycle module) and also the ability to transfer the state files to the remote site. Additionally, the VEEM need to be able to signal the hypervisor to restore the VEE state and resume its execution (also part of the VEEM life-cycle module). Regarding advance resource reservation support, the policy engine must be capable of reserving capacity in the physical infrastructure given a timeframe for certain VEEs.

In the *advanced federation* scenario, the ability to create cross-site virtual networks for the VEEs has to be achieved using the functionality offered by the virtual application network (VAN) as part of the virtual host interface API. Therefore, the VEEM needs to correctly interface with the VAN and be able to express the virtual network characteristics in a VEEM-to-VEEM connection.

In the full-featured federation scenario the live migration feature offered by this scenario will need to be supported also in the VHI API. The VEEM will just need to invoke the functionality of live migration to the hypervisor part of the VHI API to achieve live migration across administrative domains.

Virtual Execution Environment Host. The ability to monitor a federation is needed. The RESERVOIR monitoring service supports the asynchronous monitoring of a cloud data centers′ VEEHs, their VEEs, and the applications running inside the VEEs. To support federation, the originating data center must be able to monitor VEEs and their applications running at a remote site. When an event occurs related to a VEE running on a remote site, it is published and a remote proxy forwards the request to the subscribing local proxy, which in turn publishes the event to the waiting local subscribers. The monitoring framework is agnostic to type and source of data being monitored and supports the dynamic creation of new topics.

No further functionality is required for the *basic federation* in the VEEH apart from the features described for the baseline scenario. On the other hand, for the *advanced federation* one, several features are needed. First, it must have the ability to implement federated network service with virtual application network (VANs), a novel overlay network that enables virtual network services across subnets and across administrative boundaries [8,9]. VANs enables the establishment of large-scale virtual networks, free of any location dependency, that in turn allows completely "migratable" virtual networks. (1) The offered virtual network service is fully isolated, (2) it enables sharing of hosts, network devices, and physical connections, and (3) hides network related physical characteristics such as link throughputs, location of hosts, and so forth. Also, the ability to do federated migration with non-shared storage service is required. RESERVOIR enhances the standard VM migration capability typically available in every modern hypervisor with support for environments in which the source and the destination hosts do not share storage; typically the disk(s) of the migrated VM resided in the shared storage.

Regarding the *full-featured federation* scenario, hot migration is the functionality that affects the most what is demanded from VEEH in this scenario. RESERVOIR's separation principle requires that each RESERVOIR site be an autonomous entity. Site configuration, topology, and so on, are not shared between sites. So one site is not aware of the host addresses on another site. However, currently VM migration between hosts require that the source and destination hypervisors know each other's addresses and transfer a VM directly from the source host to the destination host. In order to overcome this apparent contradiction, RESERVOIR introduces a novel federated migration channel to transfer a VEE from one host to another host without directly addressing the destination host. Instead of transferring the VEE directly to the destination host, it passes through proxies at the source site and destination site, solving the unknown hypervisor location problem.

15.5 SECURITY CONSIDERATIONS

As previously reported, virtualized service-oriented infrastructures provide computing as a commodity for today's competitive businesses. Besides cost-effectiveness, they also ensure optimized use of system and network resources, reduced carbon footprints, and simplify management of their underlying resources. Businesses around the world are therefore giving enormous attention to virtualized SOI technology nowadays [4]. The capability of using virtual resources across the Internet is making up throughout a new kind of computation infrastructures. These platforms presented an unspecified environment where it is possible to run any type of VEEs. However, the salient features of these virtualization infrastructures give rise to a number of security concerns. These security threats are now emerging as the biggest obstacle in the widespread deployment of virtual infrastructures for cloud computing. Security concerns are multiplying with an increasing number of reported cloud computing incidents and other on-line services incidents such as the Kaminsky DNS vulnerability [5]. According to a survey results published in the Guardian newspaper, cloud computing security was the foremost concern for the year 2009 [6]. The higher stakes and broader scope of the security requirements of virtualization infrastructures require comprehensive security solutions because they are critical to ensure the anticipated adoption of virtualization solutions by their users and providers. The conception of a comprehensive security model requires a realistic threat model. Without such a threat model, security designers risk wasting time and effort implementing safeguards that do not address any realistic threat.

Or, just as dangerously, they run the risk of concentrating their security measures on one threat while leaving the underlying architecture dangerously exposed to others. Threats of large-scale cross-border virtualization infrastructures are broadly classified into two major categories, namely, *external threats* and *internal threats*, so as to complement the Dolev–Yao threat model [4].

15.5.1 External Threats

The Internet represents the same origin of threats for the communication across the RESERVOIR sites (VMI interfaces) and outside the RESERVOIR sites both for the SMI interface and service interface (SI—interface for service user on Internet). Some threats, related to communication, can be classified as: *men-in-the-middle, TCP hijacking (spoofing), service manifest attacks (malicious manifest/SLA format injection), migration and security policies and identity theft/impersonation (SP* or *RESERVOIR site pretends to be someone else),* and so on. The main goals of these threats are to gain *unauthorized access* to systems and to impersonate another entity on the network. These techniques allow the attackers to eavesdrop as well as to change, delete, or divert data. All the interfaces could be instead exposed to the following attacks: *denial of service (DoS* or *distributed DoS), flooding, buffer overflow, p2p-attacks,* and so on.

These kind of threats are aimed toward provoking a *system crash*, leading to the inability to perform ordinary functions. All the interfaces (SMI, VMI and SI) are affected by the same issues, but we have to underline that the solutions in some cases are different. Considering the VMI interfaces, the RESERVOIR system administrator has the full capability to manage security policies and to apply them on both the sides (endpoints of site A and site B). It is possible for each RESERVOIR site to select its own security framework; however, in the case of communication between SM and SP (SMI), the RESERVOIR cloud has to use a common security framework shared with many different partners (SPs). All threats related to SI are managed through a simple monitoring, because no action can be performed.

15.5.2 Internal Threats

Each RESERVOIR site has a logical representation with three different layers, but these layers can be compounded by one or more hardware components. Figure 15.5 gives an overview of these entities and relative mapping with a simplified view of the hardware. It is possible to split the site in two different virtual zones: *control and execution zone*; in the *control zone* the components are: Service Manager (SM), VEEM (in bridge configuration between control

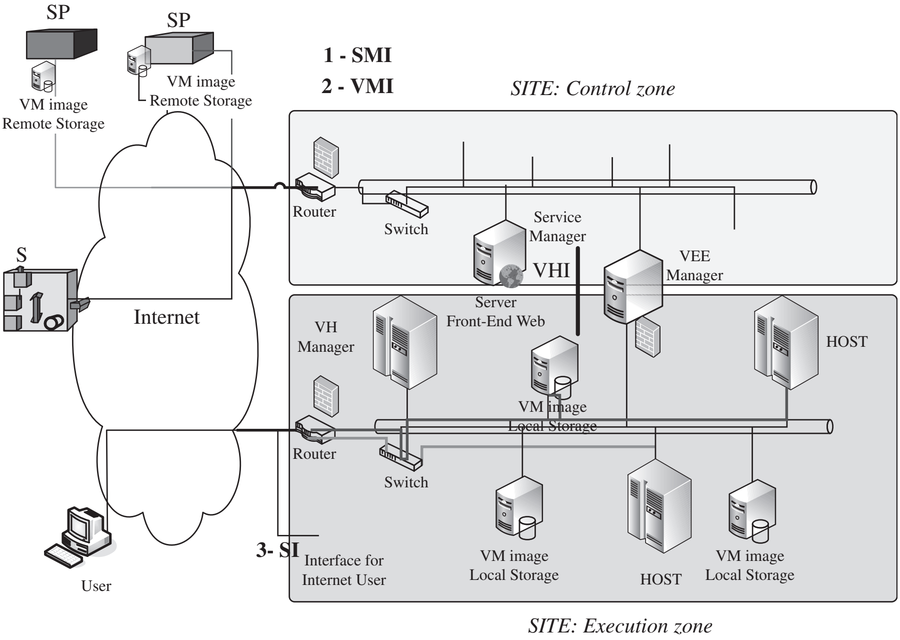


FIGURE 15.5. RESERVOIR site: internal representation.

and execution zone), network components (router, switch, cable, etc.), SMI/VMI interfaces, and VHI internal interface.

In the *execution zone* instead there are: VEEH, VEEM (in-bridge configuration between control and execution zone), VHI internal interface, network components (router, switch, cable, etc.), network storage (NAS, databases, etc.), and SI (user access interfaces).

The *control zone* can be considered a trusted area. Some threats can appear through the SMI and VEEM interfaces, since they fall into the same cases of external threats. The firewall located next to the router increases the trustworthiness. In this zone the weak ring of the system is represented by the VEEM. It is the bridge between two areas, and it allows the exchange of data among the zones. Figure 15.5 shows a firewall close to the VEEM, added to prevent any attacks from the execution area. The zone with a high level of risk is represented by the *execution zone*. This area shares all the hardware components. The hypervisor (VEEH) uses the network, storage, CPU, and ram (host) to load and execute all the VEEs. To better explain the role of each component, it can be useful to evaluate chronologically all the phases necessary to execute a virtual execution environment (VEEH); once all the requirements from the VEEM are received, it downloads the VM image from the SP, stores the image into the NAS, performs the setup configuration, and executes the VM. The internal threats related to these phases can be classified as follows: (1) threats linked to authentication/communication of SPs and other RESERVOIR site; (2) threats related to misbehavior of service resource allocation—to alter the agreement (manifest) during the translation between service manager and VEEM malicious component on SM; (3) data export control legislation—on an international cloud or between two clouds; (4) threats linked to fake command for placement of VEEs and compromising the data integrity of the distributed file system (NFS, SAMBA, CIFS); (5) storage data compromising (fake VEE image); (6) threats linked to compromise data privacy; (7) threats linked to the underlying hypervisor and OS (VEE could break hypervisor/underlying OS security and access other VEE); and (8) data partitioning between VEE.

To avoid any fraudulent access, the VEEH has to verify *authentication/communication* of SPs and other RESERVOIR sites. Thus, the same behavior is analyzed for all the communications in external threats.

Relatively to the latter group of threats (3,4,5–6,7,8), the RESERVOIR site has to guarantee different types of isolation—that is, *runtime isolation, network isolation,* and *storage isolation.*

Runtime isolation resolves all the security problems with the underlying OS. The hypervisor security mechanisms need to be used to provide the isolation.

Network isolation is addressed via the dynamic configuration of network policies and via virtual circuits that involve routers and switches.

To avoid fake VEE image loading and do not compromise data privacy, *storage isolation* has to be performed and secure protocols has to be used. Protocols like NFS, SAMBA, and CIFS are not secure.

Virtual execution environment, downloaded from any generic SP, can expose the infrastructure toward back door threats, spoofing threats and malicious code execution (virus, worm, and Trojan horse). The RESERVOIR site administrator needs to know at any time the state of threats, with a strong monitoring of the *execution zone*, through the runtime intrusion detection.

15.6 SUMMARY AND CONCLUSIONS

Cloud computing as a new computing paradigm has the potential of dramatically changing the way we use computers. Just as in the early days of the power grid, nobody could have imagined fully automated robotic production plants, or the high-definition TVs in our houses, today we can't really predict what will happen once the computing utility dream becomes a reality. As this new paradigm becomes prevalent, there are many exciting opportunities: Cloud computing providers will probably achieve levels of efficiency and utilization that seem imaginary just a few years ago, while consumers of cloud computing services will be able to free resources and focuses on their business. However, along the way there are many challenges that the industry needs to deal with. First of all, just in the case of the power grid, interoperability between cloud providers and standardization are a fundamental need. Second, cloud providers will need to build mechanisms to ensure the service levels; without proper warranties on the levels of reliability, serviceability, and availability, companies are going to be reluctant to move any of the more critical operations to the cloud. Last, but not least, the need to build trust is essential and probably the hardest because it is not a technical issue only.

In this chapter we presented the RESERVOIR model for cloud computing that deals with these issues and extended on federation and security. RESERVOIR's work on business orientation management is left for future publications.

15.6.1 Acknowledgments

We would like to thank the following people Irit Loy and Shimon Agassi from IBM, Juan Caceres and Luis Vaquero from Telefónica I&D, Stuart Clayman from UCL, and Erik Elmroth and Johan Tordsson from Umeå; without their help, this work would not have been possible.

The research leading to these results is partially supported by the European Community's Seventh Framework Programme (FP7/2001-2013) under grant agreement 215605.

REFERENCES

1. M. Armbrust et al., *Above the Clouds: A Berkeley View of Cloud Computing*, Technical Report, University of California, Berkeley, 2009.