Cloud@Home on Top of RESERVOIR

Vincenzo D. Cunsolo, Salvatore Distefano, and Antonio Puliafito*

University of Messina,
Contrada di Dio, S. Agata, 98166 Messina, Italy
`{vdcunsolo,sdistefano,apuliafito}@unime.it`

Abstract. Cloud computing is the emerging technology in distributed, autonomic, service-oriented, on-demand, trusted computing. The fact that several Cloud solutions have been implemented so far, such as Amazon EC^2 and S^3, IBM's Blue Cloud, Sun Network.com, Microsoft Azure Services Platform, etc., is evidence of the great success already achieved by this computing paradigm. On the other hand, an increasing number of research projects focus on Cloud (Nimbus, OpenNEbula, Eucalyptus, OpenQRM, RESERVOIR, etc.) thus confirming that the topic is really hot, attracts investments and funds, and involves more and more researchers.

Our idea of Cloud has been synthesized into *Cloud@Home*, a computing paradigm that supports both open and commercial communities. Starting from the contribution philosophy at the basis of the Volunteer computing paradigm, we imagine a Cloud built on off the shelf, independent, network-connected resources and devices owned and managed by different users. Such users can both sell and/or buy their resources to/from Cloud providers or, alternatively, they can share them with other users establishing open interoperable Clouds.

Being aware of the crucial and driving role played by the RESERVOIR project in defining and implementing a reference architecture for Cloud computing, in this paper we focus on how to adapt and use the results of such project in the Cloud@Home specification. Starting from the RESERVOIR architecture, we discuss and detail how the Cloud@Home paradigm can be implemented on top of it, individuating components and modules to be integrated in a new reference architecture which allows to extend RESERVOIR towards the Volunteer contributing paradigm, improving SLA management and federation issues and, at the same time, enhancing virtualization and resources management in Cloud@Home.

Keywords: Cloud computing, Volunteer computing, cross-platform interoperability, RESERVOIR.

1 Introduction and Motivation

Cloud computing is a *distributed/network* computing paradigm that mixes aspects and goals of several other paradigms such as: *Grid computing* ("... hardware

---

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

D.R. Avresky et al. (Eds.): Cloudcomp 2009, LNICST 34, pp. 41–56, 2010.

and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities" [1]), *Internet computing* ("... a computing platform geographically distributed across the Internet" [2]), *Utility computing* ("a collection of technologies and business practices that enables computing to be delivered seamlessly and reliably across multiple computers, ... available as needed and billed according to usage, much like water and electricity are today" [3]) *Autonomic computing* ("computing systems that can manage themselves given high-level objectives from administrators" [4]), *Edge computing* ("... provides a generic template facility for any type of application to spread its execution across a dedicated grid, balancing the load ..." [5]) *Green computing* (a new frontier of *Ethical computing* starting from the assumption that in next future energy costs will be related to the environment pollution [6]) and *Trusted computing* ("... a Trusted platform is a computing platform that has a trusted component, probably in the form of built-in hardware, which it uses to create a foundation of trust for software processes." [7]).

Cloud computing is a distributed computing paradigm derived from the *service-centric perspective* that is quickly and widely spreading on the IT world. From this perspective, all capabilities and resources of a Cloud (usually geographically distributed) are provided to users *as a service*, to be accessed through the Internet without any specific knowledge of, expertise with, or control over the underlying technology infrastructure that supports them. Cloud computing provides *on-demand service provision*, *QoS guaranteed offer*, and *autonomous system* for managing hardware, software and data transparently to users [8].

In order to achieve such goals it is necessary to implement a level of abstraction of physical resources, uniforming their interfaces and providing means for their management, adaptively to user requirements. The development and the success of Cloud computing is due to the maturity reached by the hardware and software *virtualization* and Web technologies.

A great interest on Cloud computing has been manifested as demonstrated by the numerous projects proposed by both industry and academia. In commercial contexts, among the others we highlight: Amazon Elastic Compute Cloud [9], IBMs Blue Cloud [10], Sun Microsystems Network.com [11], Microsoft Azure Services Platform [12], Google App Engine [13], Dell Cloud computing solutions [14]. Some scientific activities worth of mention are: RESERVOIR [15], Nimbus-Stratus-Wispy-Kupa [16], Eucalyptus [17], OpenQRM [18] and OpenNEbula [19]. All of them support and provide an on-demand computing paradigm: a user submits his/her requests to the Cloud that remotely processes them and gives back the results. This client-server model well fits aims and scopes of commercial Clouds: the business. But, on the other hand, it represents a restriction for scientific Clouds, that have an open view [20,21], closer to that of *Volunteer computing*. Volunteer computing (also called *Peer-to-Peer computing*, *Global computing* or *Public computing*) uses computers volunteered by their owners as a source of computing power and storage to provide distributed scientific computing [22]. It is behind the *"@home"* philosophy of sharing/donating network connected resources for supporting distributed scientific computing.

In [23] we introduced *Cloud@Home*, a more "democratic" form of Cloud computing in which the resources of the users accessing the Cloud can be shared in order to contribute to the computing infrastructure. The proposed solution allows to overcome both hardware and software compatibility problems of Volunteer computing and, in commercial contexts, it can establish an *open computing-utility market* where users can both buy and sell their services. Since the computing power can be described by a "long-tailed" distribution, in which a high-amplitude population (Cloud providers and commercial data centers) is followed by a low-amplitude population (small data centers and private users) which gradually "tails off" asymptotically, Cloud@Home can catch the *Long Tail* effect [24], providing similar or higher computing capabilities than commercial providers' data centers, by grouping small computing resources from many single contributors.

In order to make real such vision of Cloud, we decide to base a possible implementation on a riper architecture. Since from the infrastructure point of view one of the most important activity on Cloud is carried on by the RESERVOIR project, as above introduced, we choose to start from such architecture in order to develop the Cloud@Home infrastructure. More specifically, in this paper we investigate how to implement Cloud@Home starting from the RESERVOIR architecture, mainly building an extra layer on top of it.

Thus, in section 2 we describe the architecture of both the RESERVOIR and the Cloud@Home infrastructures, comparing the two architectures in the following section 3. Section 4 describes the implementation of Cloud@Home on top of RESERVOIR. Finally, section 5 summarizes the paper also discussing about challenges and future work.

2 Background

In this section we summarize the RESERVOIR (subsection 2.1) and the Cloud@ Home (subsection 2.2) projects and the corresponding architectures.

2.1 RESERVOIR

RESERVOIR (*REsources and SERvices VirtualizatiOn wIthout baRriers*) [15,25] is an European Union FP7 funded project that will enable massive scale deployment and management of complex IT services across different administrative domains, IT platforms and geographies. The project will provide a foundation for a service-based online economy, where - using virtualization technologies - resources and services are transparently provisioned and managed on an on-demand basis at competitive costs with high quality of service.

The RESERVOIR vision is to enable on-demand delivery of IT services at competitive costs, without requiring a large capital investment in infrastructure. The model is inspired by a strong desire to liken the delivery of IT services to the delivery of common utilities. It starts from the consideration that no single provider can serve all customers at all times, thus, next-generation

Cloud computing infrastructure should support a model where multiple independent providers can cooperate seamlessly to maximize their benefit. In their vision, to truly fulfill the promise of Cloud computing, there should be technological capabilities to *federate* disparate data centers, including those owned by separate organizations. Only through *federation* and *interoperability* infrastructure providers can take advantage of their aggregated capabilities to provide a seemingly infinite service computing utility. This view is totally shared by the Cloud@Home project.

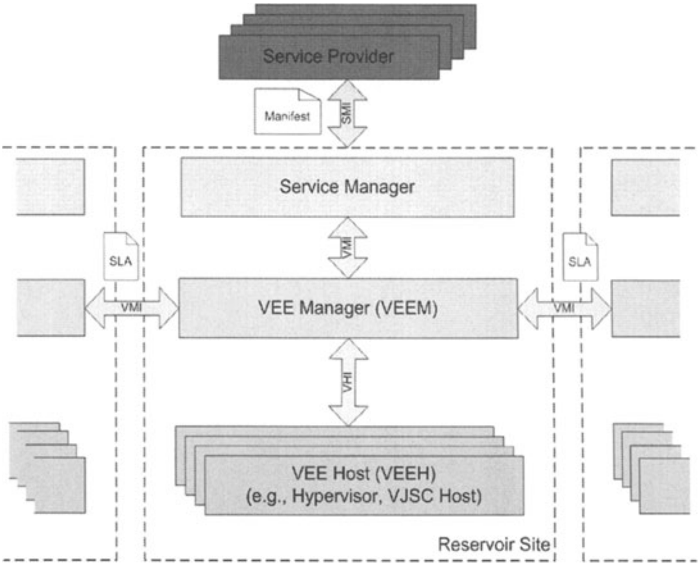

Fig. 1. RESERVOIR Architecture

The RESERVOIR architecture depicted in Fig. 1 is designed to provide a clean separation of concerns among the layers operating at different levels of abstraction. 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 highest level of abstraction, interacting with the service providers to receive their *Service Manifests*, negotiate pricing, and handle 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 Service Manager is also responsible for monitoring the deployed services and adjusting their capacity, i.e., the number of VEE instances as well as their resource allocation (memory, CPU, etc.), to ensure SLA compliance and alignment with high-level business goals (e.g., cost-effectiveness). Finally, the Service Manager is responsible for accounting and billing.

The *Virtual Execution Environment Manager* (VEEM) is the next level of abstraction, interacting with the Service Manager above, VEE Hosts below, and VEE Managers at other sites to enable federation. The VEEM is responsible for the optimal placement of VEEs into VEE hosts subject to constraints determined by the Service Manager. 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. At the VEEM level a service is provided as a set of inter-related VEEs (a VEE Group), and hence it should be managed as a whole.

The *Virtual Execution Environment Host* (VEEH) is the lowest level of abstraction, interacting with the VEE Manager to realize its IT management decisions onto a set of virtualization platforms. The 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.). Each VEEH type encapsulates a particular type of virtualization technology, and all VEEH types expose a common interface such that VEEM can issue generic commands to manage the life-cycle of VEEs. The receiving VEEH is responsible for translating these commands into commands specific to the virtualization platform being abstracted.

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. Further, 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.

2.2 Cloud@Home

Cloud@Home intends to reuse *"domestic"* computing resources to build voluntary contributors' Clouds that can interoperate each other and with external commercial Clouds, such as Amazon EC^2, IBM Blue Cloud, Microsoft Azure Services Platform, and so on. With Cloud@Home, anyone can experience the power of Cloud computing, both actively providing his/her own resources and services, and passively submitting his/her applications.

In Cloud@Home both the commercial/business and the volunteer/scientific viewpoints coexist: in the former case the end-user orientation of Cloud is extended to a collaborative two-way Cloud in which users can buy and/or sell their resources/services; in the latter case, the Grid philosophy of few but large computing requests is extended and enhanced to *open* Virtual Organizations. In both cases QoS requirements could be specified, introducing both in the Grid and in the Volunteer philosophy (*best effort*) the concept of quality.

Cloud@Home can be also considered as a generalization and a maturation of the @home philosophy: a context in which users voluntarily share their resources without any compatibility problem. This allows to knock down both hardware (processor bits, endianness, architecture, network) and software (operating systems, libraries, compilers, applications, middlewares) barriers of Grid and Volunteer computing, into a service oriented architecture.

On the other hand, Cloud@Home can be considered as the enhancement of the Grid-Utility vision of Cloud computing. In this new paradigm, users' hosts are not passive interfaces to Cloud services, but they can be actively involved in computing. Single nodes and services can be enrolled by the Cloud@Home middleware, in order to build own-private Cloud infrastructures that can (for free or by charge) interact with other Clouds.

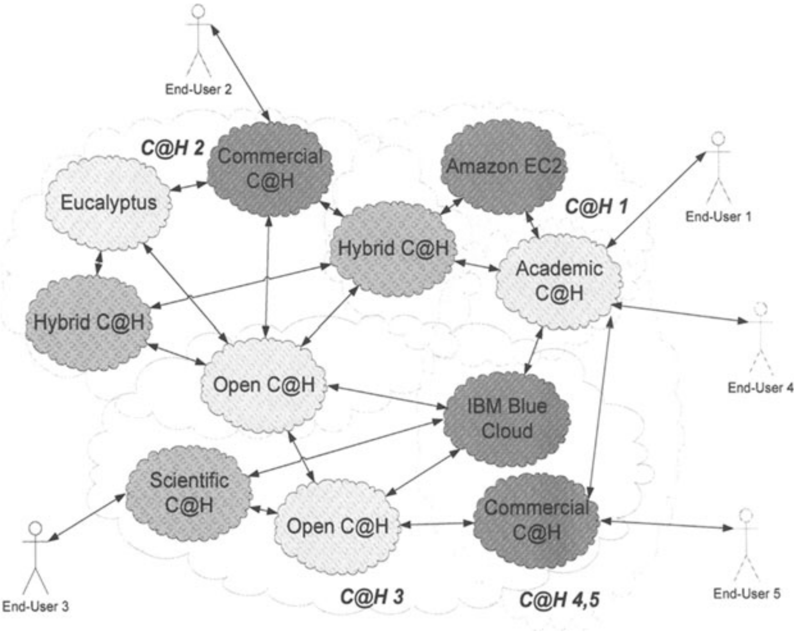

Fig. 2. Cloud@Home Scenario

The key points of Cloud@Home are on one hand the *volunteer contribution* and on the other the *interoperability* among Clouds. Well-known problems for the parallel, distributed and network computing communities have to be addressed regarding security, QoS, SLA, resource enrollment and management, heterogeneity of hw and sw, virtualization, etc. All of them must be contextualized into an highly dynamic environment in which nodes and resources can frequently change state, instantaneously becoming available/unavailable. Problems that are also partially shared and faced by the RESERVOIR project, in particular with regard to virtualization, resource management and interoperability. This motivates our choice of developing the Cloud@Home architecture on top of the RESERVOIR architecture above introduced.

The Cloud@Home idea can be pictorially depicted in Fig. 2, where several different Clouds, also built on volunteered resources (open Clouds), can interact and can provide resources and services to the other federated Clouds. They are characterized as: *open* if identify open environments operating for free Volunteer computing; *commercial* if they represent entities or companies selling their computing resources for business; *hybrid* if they can both sell or give for free their services. Both open and hybrid Clouds can interoperate with any other Clouds, also commercial, while these latter can interoperate each other if and only if the two commercial Clouds are mutually *recognized*. In this way it is possible to make *federations* of Clouds working together on the same project. Thus, a user interacting with a specific Cloud can use resources from different other Clouds, implementing different access points for a unique, global computing infrastructure. Such a form of computing, in which workloads and requests can be spread among different interoperable Cloud infrastructures, can be ideally associated to a fluid, giving rise to a new concept of computing we can identify as *fluid computing*.

The Cloud@Home logic architecture [23] by which we try to implement such idea is shown in Fig. 3, where three hierarchical layers can be identified:

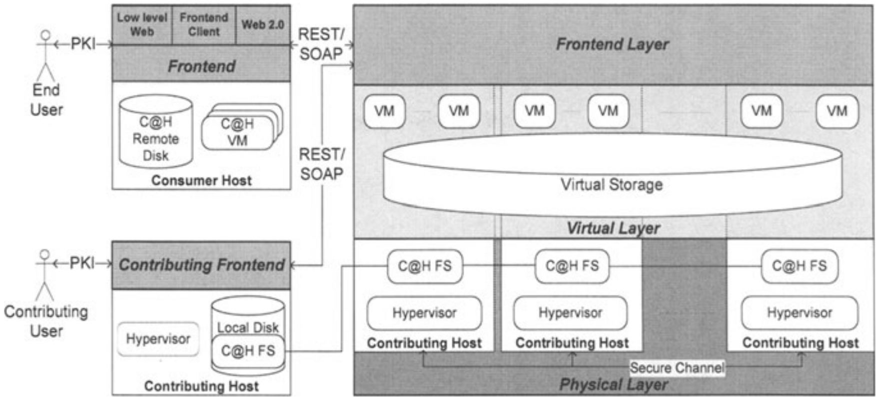

Fig. 3. Cloud@Home Architecture

- The *Frontend Layer* that globally manages resources and services (coordination, discovery, enrollment, etc), implements the user interface for accessing the Cloud (ensuring security reliability and interoperability), and provides QoS and business models and policies management facilities.
- The *Virtual Layer* that implements a homogeneous view of the distributed Cloud system offered to the higher frontend layer (and therefore to users)

in form of two main basic services: the *execution service* that allows to set up a virtual machine, and the *storage service* that implements a distributed storage Cloud to store data and files as a remote disk, locally mounted or accessed via Web.

- The bottom *Physical Layer* that provides both the physical resources for elaborating incoming requests and the software for locally managing such resources.

According to this view the Cloud is composed of several *contributing hosts* that share their resources. A user can interact with the Cloud through the *consumer host* after authenticating him/herself into the system. One of the main enhancement of Cloud@Home is that a host can be at the same time both contributing and consumer host, establishing a symbiotic mutual interaction with the Cloud.

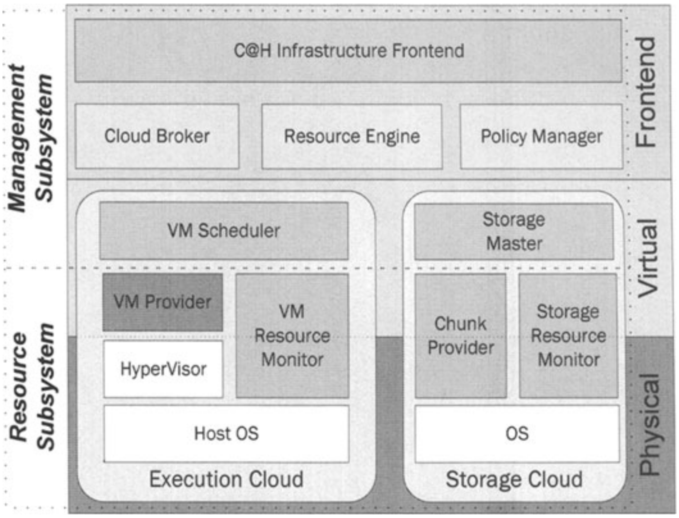

Fig. 4. Core Structure of a Cloud@Home Server

The blocks implementing the functional architecture of Fig. 3, are depicted in the layered model of Fig. 4, that reports the core structure of the Cloud@Home server-side, subdivided into *management* and *resource subsystems*:

- *Management subsystem* - is the backbone of the overall system management and coordination composed of six blocks: the *C@H infrastructure frontend*, the *Cloud broker*, the *resource engine*, the *policy manager*, the VM scheduler and the storage master.
- *Resource subsystem* - provides primitives for locally managing the resources (distributed operations), offering different services over the same resources: the *execution Cloud* and the *storage Cloud*.

The two subsystems are strictly interconnected: the management subsystem implements the upper layer of the functional architecture, while the resource subsystem implements the lower level functionalities.

The infrastructure frontend provides tools for Cloud@Home-service provider interactions, forwarding the incoming requests to the lower level blocks. The Cloud broker collects and manages information about the available Clouds and the services they provide (both *functional* and *non-functional* parameters, such as QoS, costs, reliability, *request formats' specifications* for Cloud@Home-foreign Clouds translations, etc). The policy manager provides and implements the Cloud's access facilities. This task falls into the security scope of identification, authentication, permission and identity management.

The resource engine is the hearth of Cloud@Home. It is responsible for the resources' management, the equivalent of a Grid *resource broker* in a broader Cloud environment. To meet this goal, the resource engine applies a hierarchical policy. It operates at higher level, in a centralized way, indexing all the resources of the Cloud. Incoming requests are delegated to *VM schedulers* or *storage masters* that, in a distributed fashion, manage the computing or storage resources respectively, coordinated by the resource engine. In order to manage QoS policies and to perform the resources discovery, the resource engine collaborates with both the Cloud broker and the policy manager at higher level, locally monitored and managed by schedulers and masters through the hosts' resource monitors.

The VM provider, the resource monitor and the hypervisor are responsible for managing a VM locally to a physical resource of an execution Cloud.

Chunk providers physically store the data into a storage Cloud, that are encrypted in order to achieve the confidentiality goal.

3 RESERVOIR vs. Cloud@Home

In order to adapt the Cloud@Home architecture to the RESERVOIR one, it is necessary to in depth investigate the two architectures, individuating points in common and differences. Let's start with the points in common. The first regards the architecture. Both RESERVOIR and Cloud@Home specify layered architectures decomposed in three levels, but the decomposition approach applied in the two contexts differs. In RESERVOIR the decomposition resulting in Fig. 1 is made on implementative issues. Specifically, in the RESERVOIR architecture there is a correspondence between layers and physical nodes implementing them. In Cloud@Home, the layered model of Fig. 3 describes a more abstract functional characterization, whose implementation, detailed in Fig. 4, does not establishes a direct 1:1 correspondence between functions, blocks and physical nodes. In order to implement Cloud@Home starting from the RESERVOIR architecture it is necessary to adapt the former architecture to the latter, and so to establish the correspondence between layers and blocks to physical nodes.

Another important point in common to both projects are the federation and the interoperability goals. Both projects share these goals providing different architectural solutions: RESERVOIR implements Cloud federations by providing vertical interoperability to service providers through a standardized SMI interface, and limiting the horizontal interoperability to one layer of the stack, the VEEM, achieving VEEM-to-VEEM communication through VMI. Due to the choice of defining a logical-functional architecture, Cloud@Home unifies both vertical and horizontal interoperability into a unique block specifically conceived and devoted to interoperability and federation tasks: the Cloud broker.

As in RESERVOIR, we believe that the best solution to achieve interoperability among different Clouds is the standardization way, opinion validated by several significant initiatives and efforts towards Cloud standardizations [20,26,21]. It is needed a clear, unambiguous and widely accepted standard allowing automatic Cloud discovery and communications setup. But, since at now Cloud infrastructures are mainly commercial, the question wether the involved corporations will accept to conform to a standard is an open problem not so obvious. So we think it could be necessary to provide means for bridging or translating between different interfaces in order to reach the interoperability goal in Cloud. The Cloud broker accomplishes this task with regard to Cloud@Home.

With regard to interoperability, another important problem to face is the *Cloud discovery*: how a Cloud knows about the existence of other Clouds and the services they provide? While RESERVOIR not so clearly identifies such topic problem, Cloud@Home deals with the Cloud discovery by delegating such task to the Cloud broker. Both centralized and distributed solutions are possible for addressing the Cloud discovery task, but we retain to follow a trade-off between the two approaches in order to take advantage from both [23].

A significant difference between RESERVOIR and Cloud@Home regards resource management. RESERVOIR concentrates all the resource management functions into the VEEM. This centralized solution allows to simplify the resource management but, on the other hand, it cannot easily manage great quantities of hosts (VEEH) implementing the Cloud infrastructure, since a unique manager does not scale when the number of hosts increases. Cloud@Home instead proposes a hierarchical approach, by which the resource management is coordinated at high level by a resource engine, and implemented at lower level by schedulers or masters that could be also hierarchical. This solution allows to reduce the workload incoming to the resource engine moving it toward the VM schedulers. A distributed-hierarchical approach is further motivated by the fact that the context in which Cloud@Home operates includes volunteer contributions. Such environment is highly dynamic, since resources can be "plugged" in or out the infrastructure autonomously, therefore the system must be able to manage such dynamics, quickly adapting to variations. For this reason to address the problem we think about autonomic approaches [23], able to quickly reconfigure after changes occur.

With regard to SLA and QoS issues, RESERVOIR splits the task of SLA in two parts: the *vertical SLA* towards Service Provider is managed by the Service Manager; the *horizontal SLA* among VEEM of different infrastructures due to the dynamic federation of infrastructure providers. The functional architecture of Cloud@Home individuates a specific block to which assign QoS and SLA challenges, the policy manager. In combination with the resource engine, the policy manager manages the vertical SLA with service providers, locally monitoring the resources through a resource monitor active for each host. The resource information are kept locally to the corresponding VM scheduler or storage master, accessed by the resource engine in the SLA discovery and checking/monitoring phases. The policy manager also provides tools for the horizontal SLA. In such case the SLA process is managed in combination with the Cloud broker that performs the Cloud discovery.

An important topic to adequately take into the right consideration is security, particularly felt in high dynamic and interoperable-distributed environments. Security issues are only partially covered into RESERVOIR, mainly delegated to underlying technologies such as virtualization isolation and OpenNEbula security. Cloud@Home faces several security issues in its architecture. Authentication is implemented through PKI infrastructure and X509 certificates, and it is managed by the policy manager. Starting from the Grid experience, credential delegation and Single Sign-On (SSO) mechanisms can be used in order to manage the identity into a Cloud. The problem of identity management in Cloud@Home is further complicated by the interoperability goal, since it is necessary that interoperable Clouds must mutually trust each other. Also in such case it is strongly recommended to specify and use widely accepted standards in the topic of authentication and identity management.

In the context thus individuated, we think it is necessary to build up an *identity provider* which provides tools and mechanisms for univocal/single-users and mutual-Clouds authentications. In order to implement such identity provider we think about a distributed technique as the *eXtensible Resource Identifier* (XRI) [27] and the OpenID [28] approaches.

Information security in Cloud@Home is achieved through encryption techniques [29]. The information stored in a Cloud@Home infrastructure are always encrypted, while information in clear are transferred through a secure channel such as SSH, TSL, IPSEC, SSL, XMPP, etc.

4 Synthesis: Implementing Cloud@Home on Top of RESERVOIR

The differences between RESERVOIR and Cloud@Home detailed in the previous section highlight that, in the corresponding architectures, there are parts in common and parts riper or better covered in one of them rather then in the other. This motivates our efforts in combining the two approaches into a Cloud architecture resulting as a trade-off between the existing ones.

From the above considerations we can observe that the main difference between the two approaches is that Cloud@Home adopts a higher abstraction level than RESERVOIR in the architecture specification. This impression is validated by the two architecture's implementations: RESERVOIR better focuses on low level aspects such as virtualization and centralized resource management, while Cloud@Home privileges higher level aspects mainly concerning the management of distributed resources, SLA and QoS, security and interoperability, maybe not yet well focused into RESERVOIR. Moreover, since the context of Cloud@Home, also including the volunteer contribution, can be wider than the RESERVOIR one, and also due to the experience and the knowhow reached by this latter project, we retain really practicable and feasible the idea of building a Cloud@Home architecture starting from the RESERVOIR one.

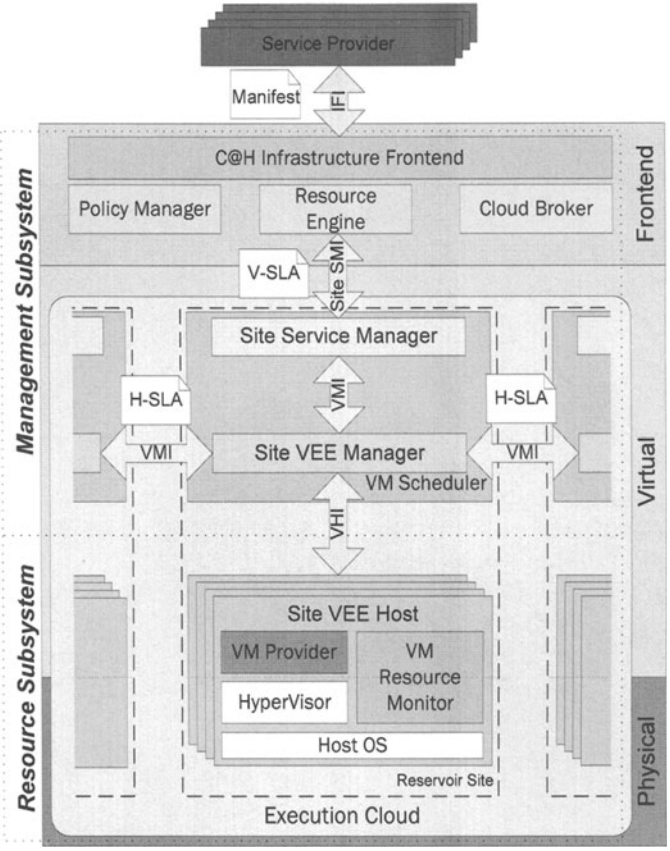

Fig. 5. Cloud@Home Architecture on top of RESERVOIR

More specifically, according to such interpretation, being RESERVOIR focused on lower level aspects than Cloud@Home, it is reasonable to think about an implementation of Cloud@Home on top of RESERVOIR. Such idea is formally represented into the architecture shown in Fig. 5, where concepts and parts of both the corresponding architectures are merged and integrated.

From a functional/higher-level perspective, the hierarchical distributed resources management, the interoperability among different Clouds and the high level security management are drawn from Cloud@Home. With regard to the resource management, at lower level, each site is organized according to the

RESERVOIR architecture, with a Site Manager that manages a pool of distributed network-connected resources, the Site VEEHs, constituting the site. In order to implement an adaptive and easy-to-scale solution, each site can manage a limited finite number of resources. Thus, the sites are hierarchically coordinated by the specific subsystems of the frontend layer (resource engine, policy manager and Cloud broker). This solution allows to also manage volunteer contributions: each time a new resource is offered to the infrastructure and must be enrolled into the Cloud, the resource engine has to select a site to which associate the resource. If no sites are available a new site is built up by aggregating the resources that are not yet associated to a site with the ones selected from other different sites, applying load balancing principle in the selection in order to avoid overloaded sites and resources.

In this new architecture, the SLA and QoS management solution is derived from both the original architectures: the characterization made in RESERVOIR, distinguishing between high level, vertical SLA (VSLA) and low level, horizontal SLA (HSLA) has been inherited by the new architecture. The high level VSLA is subdivided into two parts: the former between the service providers and the frontend, the latter between the frontend layer blocks and each site. The HSLA has the aim of making adaptive the infrastructure to external solicitations. Before asking to resource engine and policy manager, the single VEEM can autonomously try to discover resources when they cannot locally (on-site) satisfy the requirements, by asking to other VEEM. Otherwise, they recur to resource engine and policy manager, that must be always updated also in case of lower level reconfigurations. Such goal can be pursued by exploiting autonomic computing techniques.

Let's jump into details. Following a top-down approach, the service providers interact with the Cloud@Home infrastructure frontend through a specific *infrastructure frontend interface* (IFI) that forwards their service manifests to the lower level blocks. The information specified in the service manifests are translated into the local Cloud format by the Cloud@Home infrastructure frontend and therefore forwarded to the lower level blocks, as done in Cloud@Home. Thus the resource engine, in collaboration with the policy manager and, if required, with the Cloud broker, perform the VSLA with the service provider. This task requires the interposition of the infrastructure frontend, from one side, and of the site through the specific SMI interface from the other side.

Through the frontend, we can also adapt the SLA to interact (by the policy manager and the resource engine) with the VM Scheduler, which includes two RESERVOIR components: the Site Service Manager and the Site VEE Manager. According to the Cloud@Home architecture, the VM Scheduler uses and interacts with the VM Provider. To integrate this behavior within RESERVOIR, we can place the VM provider inside a Site VEEH, allowing the resource monitor to directly interact with VM scheduler.

Such requests are managed on-site by the site service manager, that negotiates the site SLA interacting with the lower VEEM layer, which manages the site resources and therefore monitors their status. Both such components implement the functions associated to the original Cloud@Home VM scheduler and therefore in Fig. 5 are encapsulated in this latter component.

A Cloud@Home site is also composed of a pool of VEEH physical nodes. Each VEEH contains a Cloud@Home VM provider and a VM resource monitor, and obviously has its own hypervisor and host OS, such as the one typically used in RESERVOIR (XEN, KVM hypervisors and Linux OS). A goal of Cloud@Home is to implement a cross-platform interface independent of hypervisor and host OS. This is a mandatory requirement in case Clouds interoperability is needed. Since this is not satisfied by the RESERVOIR architecture, we need to extend the RESERVOIR infrastructure in order to support other hypervisors. The best solution is the specification of a unique, standard VM format [26]. Another requirement is that the hypervisors have to be interoperable, independent of the host OS. Our idea to overcome this latter specific OS constraints, waiting for a standard VM format, is to include the support of VirtualBox [30] in the architecture.

5 Conclusions

Cloud computing provides *on-demand service provision*, *QoS guaranteed offer*, and *autonomous system* for managing hardware, software and data transparently to users. To such context, Cloud@Home adds the possibility of enrolling volunteer contributing resources merging aims and scopes of both Cloud and Volunteer computing paradigms. In order to implement Cloud@Home, instead of starting from scratch, we decided to exploit the existing work produced by the RESERVOIR project which is building a Cloud computing framework without barrier in a federated way for implementing large data center.

In this paper we propose how to merge the two approaches to introduce flexibility in RESERVOIR, improving SLA management and federation issues better covered in Cloud@Home. Moreover, the volunteer contribution feature allows to extend RESERVOIR Clouds with new available resources from academic, open communities and commercial organizations. On the other hand, Cloud@Home benefits from RESERVOIR, exploiting its riper infrastructure in terms of virtualization and site resources management.

References

1. Foster, I.: What is the grid? - a three point checklist. GRIDtoday 1(6) (July 2002)
2. Milenkovic, M., Robinson, S., Knauerhase, R., Barkai, D., Garg, S., Tewari, A., Anderson, T., Bowman, M.: Toward internet distributed computing. Computer 36(5), 38–46 (2003)
3. Ross, J.W., Westerman, G.: Preparing for utility computing: The role of it architecture and relationship management. IBM System Journal 43(1), 5–19 (2004)
4. Kephart, J.O., Chess, D.M.: The vision of autonomic computing. Computer 36(1), 41–50 (2003)

5. Davis, A., Parikh, J., Weihl, W.E.: Edgecomputing: extending enterprise applications to the edge of the internet. In: WWW Alt. 2004: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, pp. 180–187. ACM, New York (2004)
6. Murugesan, S.: Harnessing green it: Principles and practices. IT Professional 10(1), 24–33 (2008)
7. Pearson, S.: Trusted Computing Platforms: TCPA Technology in Context. Prentice Hall PTR, Upper Saddle River (2002)
8. Wang, L., Tao, J., Kunze, M., Castellanos, A.C., Kramer, D., Karl, W.: Scientific Cloud Computing: Early Definition and Experience. In: HPCC 2008, pp. 825–830. IEEE Computer Society, Los Alamitos (2008)
9. Amazon Inc.: Elastic Compute Cloud [URL]. Amazon (2008), `http://aws.amazon.com/ec2`
10. IBM Inc.: Blue Cloud project. IBM (2008), `http://www-03.ibm.com/press/us/en/pressrelease/22613.wss/`
11. Sun Microsystem.: Network.com (SUN), `http://www.network.com`
12. Co., M.: (Azure services platform), `http://www.microsoft.com/azure/default.mspx`
13. Inc., G.: (Google application engine), `http://code.google.com/intl/it-IT/appengine/`
14. Dell: (Dell cloud computing solutions), `http://www.dell.com/cloudcomputing`
15. RESERVOIR Consortium: RESERVOIR Project (2009), `http://www-03.ibm.com/press/us/en/pressrelease/23448.wss/`
16. University of Chicago-University of Florida-Purdue University-Masaryk University: Nimbus-Stratus-Wispy-Kupa Projects (January 2009), `http://workspace.globus.org/clouds/nimbus.html/`, `http://www.acis.ufl.edu/vws/`, `http://www.rcac.purdue.edu/teragrid/resources/#wispy`, `http://meta.cesnet.cz/cms/opencms/en/docs/clouds`
17. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The Eucalyptus Open-source Cloud-computing System. University of California Santa Barbara Computer Science (2009), `http://open.eucalyptus.com/`
18. OpenQRM: Open Source Data Management Platform (2009), `http://www.openqrm.com/`
19. Distributed Systems Architecture Research Group: OpenNEbula Project Universidad Complutense de Madrid (2009), `http://www.opennebula.org/`
20. Open Cloud Manifesto Organization: The Open Cloud Manifesto (Spring 2009), `http://www.opencloudmanifesto.org/`
21. Distributed Management Task Force, Inc.: Open Cloud Standards Incubator (2009), `http://www.dmtf.org/about/cloud-incubator`
22. Anderson, D.P., Fedak, G.: The computational and storage potential of volunteer computing. In: CCGRID 2006, Washington, DC, USA, pp. 73–80. IEEE Computer Society, Los Alamitos (2006)
23. Cunsolo, V.D., Distefano, S., Puliafito, A., Scarpa, M.: Volunteer Computing and Desktop Cloud: the Cloud@Home Paradigm. In: Proceedings of the 8th IEEE International Symposium on Network Computing and Applications (IEEE NCA 2009), July 9-11. IEEE, Los Alamitos (2009)
24. Anderson, C.: The Long Tail: How Endless Choice Is Creating Unlimited Demand. Random House Business Books (2006)

25. Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I.M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W., Galan, F.: The reservoir model and architecture for open federated cloud computing. IBM Journal of Research and Development 53(4) (2009)
26. VMWare Inc., XEN Inc.: The Open Virtual Machine Format Whitepaper for OVF Specification (2007), `http://www.vmware.com/appliances/learn/ovf.html`
27. OASIS Extensible Resource Identifier (XRI) TC: Extensible Resource Identifier (XRI) (2009), `http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xri`
28. Reed, D., Chasen, L., Tan, W.: OpenID identity discovery with XRI and XRDS. In: IDtrust 2008: Proceedings of the 7th symposium on Identity and trust on the Internet, pp. 19–25. ACM, New York (2008)
29. Cunsolo, V.D., Distefano, S., Puliafito, A., Scarpa, M.: Implementing Data Security in Grid Environment. In: Proceedings of the IEEE Workshop on Emerging Technologies for Next Generation GRID (IEEE ETNGRID 2009), June 9 - July 11. IEEE, Los Alamitos (2009)
30. Sun Microsystems Inc.: VirtualBox (2009), `http://www.virtualbox.org/`