2010 Second International Conference on Advances in Future Internet

Three-Phase Cross-Cloud Federation Model: The Cloud SSO Authentication

Antonio Celesti, Francesco Tusa, Massimo Villari and Antonio Puliafito
Dept. of Mathematics, Faculty of Engineering, University of Messina
Contrada di Dio, S. Agata, 98166 Messina, Italy.
e-mail: {acelesti,ftusa,mvillari,apuliafito}@unime.it

***Abstract*—Cloud federation aims to cost-effective assets and resources optimization among heterogeneous environments where clouds can cooperate together with the goal of obtaining "unbounded" computation resources, hence new business opportunities. This paper describes an architecture for the federation establishment, where clouds that need external resources ask to federated clouds the renting of extra physical resources. Our architecture introduces a new module named Cross-Cloud Federation Manager including three agents (Discovery, Match-making and Authentication). In this work, we specifically focus on the authentication agent, which is responsible for a secure federation. To address such problem we propose a technical solution based on the IdP/SP model along with the SAML technology.**

***Keywords*-Cloud Computing; Cross-Cloud; Heterogeneous Systems; Federation; Security, Trustiness; SAML.**

I. INTRODUCTION

Cloud computing brings a new level of efficiency in delivering services, representing a tempting business opportunity for IT operators of increasing their revenues. Services are classified as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), or Software as a Service (SaaS) [1], whereas clients might range from other clouds, organizations, and enterprises to single users.

Currently, the cloud scenario includes hundreds of independent, heterogeneous, private/hybrid clouds but, many business operators have predicted that the process toward an interoperable federated cloud scenario will begin shortly. In [2], the evolution of the cloud computing market is hypothesized in three subsequent stages: stage 1 "Monolithic" (now), cloud services are based on proprietary architectures - islands of cloud services delivered by megaproviders (this is what Amazon, Google, Salesforce and Microsoft look like today); stage 2 "Vertical Supply Chain", over time, some cloud providers will leverage cloud services from other providers. The clouds will be proprietary islands yet, but the ecosystem building will start; stage 3 "Horizontal Federation", smaller, medium, and large providers will federate horizontally themselves to gain: economies of scale, an efficient use of their assets, and an enlargement of their capabilities.

This work describes how to make up an interoperable heterogeneous cloud environment in a "Horizontal Federation" configuration, where clouds can cooperate together accomplishing trust contexts and providing new business opportunities, such as cost-effective assets optimization, power saving, and on-demand resources provisioning. In particular, we carry out what the involved issues are, analyzing a resource provisioning scenario and proposing an architectural solution. For simplicity, in the rest of the paper, with terms such as "cross-cloud federation", "federation in cloud computing", or "cloud federation" we will refer to the above mentioned "Horizontal Federation".

In this work we propose a three-phase cross-cloud federation model where, the federation establishment between a cloud needing external resources and a cloud offering resources, passes through three main phases: *discovery*, the cloud looks for other available clouds; *match-making*, the cloud selects between the discovered clouds the ones, which fit as much as possible its requirements; *authentication*, the cloud establishes a trust context with the selected clouds. According to our three-phase model, we designed a module named Cross-Cloud Federation Manager (CCFM) including three agents (Discovery, Match-making and Authentication) responsible to accomplish the aforementioned three phases. More specifically, we focus on the authentication agent, which is responsible for a secure federation. The authentication phase poses many serious problems in a cross-cloud federation establishment due to the need for each cloud of managing a huge number of credentials depending on the security mechanisms employed in each infrastructure. Instead, in our opinion, a cloud should be able to authenticate itself with other heterogeneous clouds regardless their security mechanisms, performing the log-in once, gaining the access to all required resources. We identify this issue as: *Cloud Single-Sign On (SSO) Authentication*. To address such problem, we propose a technical solution based on the Security Assertion Markup Language (SAML) technology [3]. More specifically, we designed a new SAML profile named Cross-Cloud Authentication Agent SSO (CCAA-SSO), which defines the steps needed for a secure cloud SSO authentication to be performed by the authentication agents of the involved clouds.

The paper is organized as follows. Section II describes the state of the art of cloud federation. After planning a resource provisioning scenario of cooperating clouds, in Section III, we provide a detailed analysis of cloud federation requirements, introducing the concept of *home cloud* and *foreign cloud*. In section IV, we introduce our three-phase federation

978-0-7695-4091-7/10 $26.00 © 2010 IEEE
DOI 10.1109/AFIN.2010.23

model. In addition it is presented the Cross-Cloud Federation Manager (CCFM), a module deployable on each cloud middleware, responsible to accomplish the three federation phases by means of three autonomous software agents. In Section V, we focus on the authentication phase and in particular on the cloud SSO authentication problem presenting our solution, the new SAML CCAA-SSO profile. A detailed description of the messages exchanging flow is also provided with practical examples. Conclusions and lights to the future are summarized in Section VI.

II. Related Work

Cloud computing is generally considered as one of the more challenging research field in the IT world. Interesting research areas are strictly related to cloud computing such as security, privacy, trustiness and federation. Considering the federation perspective, new terms are also been coined as Intercloud ("Think of the existing cloud islands merging into a new, interoperable Intercloud where applications can be moved to and operate across multiple platforms..." [4]) or Cross-cloud ("For the benefit of human society and the development of cloud computing, one uniform and interoperable Cross-cloud platform will surely be born in the near future..." [5]).

A few works are available in literature related to cloud federation. The main reason is that several pending issues concerning security and privacy still have to be addressed, and a fortiori, is not clear what cloud federation actually means and what the involved issues are [6]. Nowadays, the latest trend to federate applications and service oriented architectures (SOAs) over the Internet is represented by the Identity Provider/Service Provider (IdP/SP) model [7]. Examples are the aforementioned SAML, OpenID [8], Shibboleth [9] and Cardspace [10]. Such solutions, considered alone, do not solve the cloud federation issues. In fact, the federation problem in cloud computing is greater than the one in traditional systems. The main limit of the existing federation solutions is that they are designed for static environments requiring a priori policy agreements, whereas clouds are high-dynamic and heterogeneous environments, which require particular automatic security and policy arrangements. Keeping in mind the cloud federation perspective, several security issues are already picked out. Interoperability in federated heterogeneous cloud environments is faced in [5], in which the authors propose a trust model where the trust is delegated between trustworthy parties, which satisfy certain constrains. Instead, the data location problem is treated in [11] where it is proposed a privacy manager to solve the problems of the data compliance to the laws of different jurisdictions.

Nevertheless, such works do not fully clarify what it is really meant with the term cloud federation. Basically, it is not fully evaluated when, why, and how a cloud federation should be established and what the impact over the existing infrastructure, the involved architectural issues, and the security concerns are. Therefore, we think a cloud federation model addressing architectural and security issues, also with implementation practice compliant with existing cloud infrastructures, is strongly needed.

III. Cross-Cloud Federation Analysis: our Reference Scenario

In this Section we try to clarify ideas concerning the general concept of cross-cloud federation. In order to identify requirements and goals, we propose a possible resource provisioning scenario where clouds might benefit of federation advantages. Cloud Computing relies its computational capabilities exploiting the concept of "virtualization". This technology has re-emerged in recent years as a compelling approach of increasing resource utilization and reducing IT services costs. The common theme of all virtualization technologies is hiding the underlying infrastructure by introducing a logical layer between the physical infrastructure and the computational processes. The virtualization is being possible thanks to Virtualization Machine Monitors (VMMs commonly known as "hypervisors"), i.e. processes that run on top of a given hardware platform, control and emulate one or more other computer environments (virtual machines). Each of these virtual machines, in turn, runs its respective "guest" software, typically an operating system, executed as if it is installed on a stand-alone hardware platform.

Private clouds hold their own virtualization infrastructure where several virtual machines are hosted to provide services to their clients. In a scenario of "cross-cloud federation", each cloud operator is able to transparently enlarge its own virtualization resources amount (i.e., increasing the number of instantiable virtual machines) asking computing and storage capabilities to other clouds. Consequently, the cloud operator will be able to satisfy any further service allocation request sent by its clients.

According to our analysis, within the above mentioned scenario we distinguish two types of cloud: home cloud and foreign cloud. Home cloud is a cloud provider which is unable to instantiate further virtual machines as the capability of its virtualization infrastructure is saturated and consequently, forwards federation requests to foreign clouds with the purpose to host its own virtual machines on their virtualization infrastructures. Instead, foreign cloud is a cloud provider which leases part of the storage and computing capabilities of its virtualization infrastructure to home clouds for free or by charge. A cloud provider could be at the same time both home cloud and/or foreign cloud.

In order to better explain such idea, we consider the reference scenario depicted in Figure 1. When a home cloud realizes that its virtualization infrastructure has saturated its capabilities, in order to continue providing services to its clients (i.e., other clouds, enterprises, generic end users, etc), it decides to federate itself with foreign clouds A and

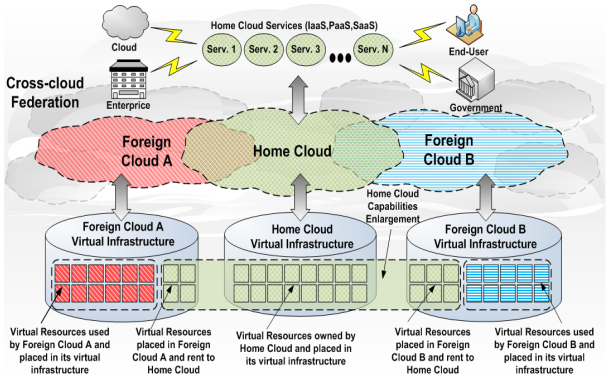

Figure 1. Cross-Cloud Scenario: basic for heterogeneous and federated clouds.

B. The home cloud, besides hosting virtualization resource inside its own virtualization infrastructure, is also able to hosting virtual machines inside the foreign clouds A and B virtualization infrastructures, enlarging the amount of its available virtualization resources (See Figure 1, bottom part). Therefore, although the virtualization resources rent to the home cloud are physically placed within the virtualization infrastructures of foreign clouds A and B, they are logically considered as resources indeed hosted within the home cloud virtualization infrastructure.

Despite the obvious advantages, the implementation of such cross-cloud federation scenario is not at all trivial. The main reason is that clouds are more complicated than traditional systems and the existing federation models are not applicable. In fact, while clouds are typically heterogeneous and dynamic, the existing federation models are designed for static environments where it is needed an a priori agreement among the parties to make up the federation. Keeping in mind the aforementioned scenario, we think cloud federation needs to meet the following requirements: *a) automatism and scalability*, a home cloud, using discovery mechanisms, should be able to pick out the right foreign clouds which satisfies its requirements reacting also to cloud changes; *b) interoperable security*, it is needed the integration of different security technologies, for example, permitting a home cloud to be able to join the federation without changing its security policies. In the "interoperable security" context we identify: *1) SSO authentication*, a home cloud should be able to authenticate itself once gaining the access to the resources provided by federated foreign clouds belonging to the same trust context without further identity checks; *2) digital identities and third parties*, each home cloud should be able to authenticate itself with foreign clouds using its digital identity guaranteed by a third party. This latter feature is more challenging because it implies a cloud has to be considered as a subject uniquely identified by some credentials.

IV. Cross-cloud federation: Architectural Overview

In Section III, we described the concept of federation and its bindings with the cloud. In the following, we provide a detailed description of the approach used to address the cross-cloud federation issues. Considering the requirements of automatism, scalability and interoperability previously stated, our solution tries to answer all such issues. Describing the federation process we point out three main different phases: *discovery*, *match-making* and *authentication*. These phases are opportunely explained in the following.

A. The Three-Phase Cross-Cloud Federation Model and the Cross-Cloud Federation Manager

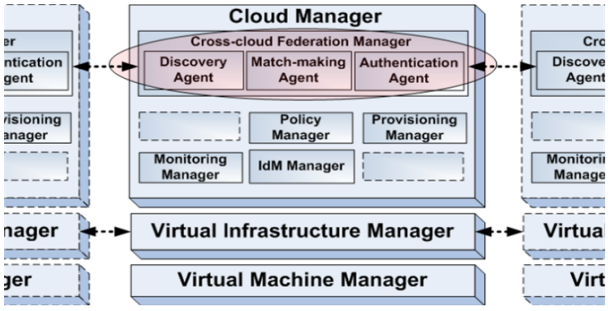

Figure 2. CCFM for the management of the cross-cloud federation inside the general three-layers cloud architecture.

In order to identify the main components constituting a cloud and better explain the federation idea on which our work is based, we are considering the internal architecture of each cloud as the three-layered stack [12] presented schematically in Figure 2. Starting from the bottom, we can identify: *Virtual Machine Manager*, *Virtual Infrastructure (VI) Manager* and *Cloud Manager*. VI Manager is a fundamental component of private/hybrid clouds acting as a dynamic orchestrator of Virtual Environments (VEs), which automates VEs setup, deployment and management, regardless of the underlying Virtual Machine Manager layer (i.e., Xen, KVM, or VMware). The Cloud Manager layer is instead able to transform the existing infrastructure into a cloud, providing cloud-like interfaces and higher-level functionalities for security, contextualization and VM disk image management.

In a cloud architecture designed according to the aforementioned three-layered stack, all the cloud components and their respective functions are clearly defined and separated, thus introducing simplicity and efficiency when the cloud middleware has to be modified or new functionalities have to be added. In our work, we exploited such modular characteristics of the layered cloud architecture, and introduced a new component within the Cloud Manager layer (depicted in the top part of Figure 2), named *Cross-Cloud Federation Manager* (CCFM). The CCFM has been conceived for

enabling each cloud to perform all the operations needed to pursue the target of the federation establishment.

The cross-cloud scenario we are considering can be seen as an highly dynamic environment: new clouds, offering different available resources and different authentication mechanisms could appear, while others could disappear. Taking into account such dynamism, when a home cloud needs to "lease" external resources from a foreign cloud, the first step the home cloud will perform refers to the *discovery* (phase 1) of the foreign cloud, which properly *matches* (phase 2) its requirements (both in terms of available resources and supported authentication mechanisms). Once these two steps have been performed, and the best foreign cloud has been found, in order to establish a secure interaction between the home cloud and the selected foreign cloud, an *authentication* (phase 3) process will begin.

The CCFM module represents the main "actor" in our three-phase federation model. In our design, it consists of three different subcomponents (agents) each addressing a different phase of the federation model:

- The *discovery agent* manages the discovery process among all the available clouds within the dynamic environment. Since its state is pretty flexible and dynamic, the discovery process has to be implemented in a totally distributed fashion: all the discovery agents must communicate exploiting a p2p approach.
- The *match-making agent* accomplishes the task of choosing the more convenient foreign cloud, evaluating all the parameters regarding the QoS, available resources and available authentication mechanisms. By means of specific algorithms, this agent is able to evaluate from all the available (discovered) clouds, the ones that best "fit" the requirements (e.g. the load capacity in terms of resources leasing and the supported authentication methods) of its home cloud.
- The *authentication agent*, cooperating with third parties trusted entities, takes part in the creation of a security context between home and foreign clouds. When the authentication phase begins, the home cloud authentication agent contacts its "peer" on the foreign cloud: the authentication process between such agents (and thus the clouds) will be lead exchanging authentication information in form of meta-data, also involving trusted third parties in the process. The Authentication Agent communicates both with other peers and third parties via web service interfaces.

The accomplishment of the authentication process, carried out by the authentication agents of both home and foreign clouds, leads to the establishment of a secure and direct connection between the related VI Manager Layer of the same clouds. As consequence, the home cloud will be able to instantiate (or migrate) Virtual Resources (VMs) on the Foreign Cloud in a secure environment. The concept of migration can be seen as the opportunity to move the Virtual Machines not only in intra-site domain but also to transfer them on federated inter-site domains. In this case the migration might occur across subnets, among hosts that do not share storage and across administrative boundaries.

Although in Section IV-B we'll describe the three phases needed to pursue the cloud federation, the main scope of this paper refers to the solution of the cloud SSO authentication problem.

In our work, the practical solution to overcome the authentication problem is the introduction the well-known concept of Identity Provider (IdP) along with a new SAML profile (further details are presented in Section V).

B. The Three-Phase Cross-Cloud Federation Process: a Concrete Scenario

In this Section, we provide a more detailed description of the three-phase cross-cloud federation model considering the scenario represented in Figure 3. As depicted, such scenario includes both home clouds and foreign clouds, which are represented according to the layered model already discussed. The highest stack level of these clouds also comprises the CCFM module, which comes with its own agents (discovery, match-making and authentication).

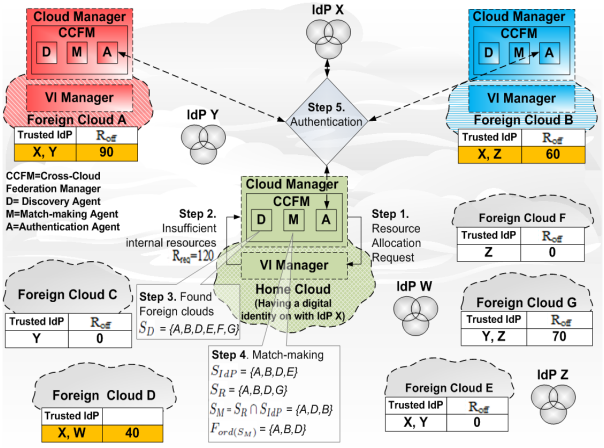

Figure 3. Example of cross-cloud federation establishment.

We remarked the need of providing a global authentication mechanism exploitable from all the entities belonging to the cloud federation. In Figure 3, together with home clouds and foreign clouds, IdPs are also depicted. An IdP is a provider of digital identity representing a trusted third party, which provides authentication services to its clients. In such scenario we assume each home cloud must have one digital identity at least on one IdP (even though many cloud digital identities may exist on different IdPs), whereas each foreign cloud must be trusted or compliant with one or more IdPs. Before explaining our motivation to the introduction of IdP

within our scenario, we provide a description of the three phases needed to achieve the cloud federation.

In the scenario of federation establishment depicted in Figure 3, during the step 1, the home cloud manager layer receives a request for services from its clients and sends a resource allocation request (i.e. virtual machines) to the underlying VI manager layer. In step 2 the home cloud VI manager, evaluating its instantaneous workload, replies to the request notifying it has not enough resources. In step 3 (the discovery phase) the home cloud manager decides to ask for resources to foreign clouds: the resource request is forwarded to the CCFM, which, by means of its discovery agent, will begin the *discovery* process to obtain a list of all the available foreign clouds. The discovery phase can exploit whatever p2p approach to achieve the complete list of cloud providers. Each discovered foreign cloud is associated to a set of meta-data describing several cloud information: the amount and type of the resources available for leasing, the offered SLA level and the supported IdP(s). In this particular example, the agent has found the set of discovered foreign clouds $S_D = \{A, B, C, D, E, F, G\}$.

In step 4 the *match-making* phase begins: the match-making agent of the home cloud selects from the set of discovered foreign clouds S_D the ones, which fits its requirements. The adopted criteria to perform the selection is based on two different evaluation tasks: in the first one, starting from the foreign clouds set S_D, a new subset $S_R = \{A, B, D, G\}$ is obtained considering the foreign clouds better satisfying the home cloud request in terms of resources availability (CPU, RAM, storage) and QoS. In the second evaluation task, starting from the discovered foreign clouds set S_D, the match-making agent selects the subset of foreign clouds $S_{IdP} = \{A, B, D, E\}$, having trusted relationship(s) with the IdP(s) on which the home cloud already has a digital identity. In this example foreign clouds A, B, D, and E are trusted with the IdP X, which provides authentication services to the home cloud guaranteeing for its digital identity. The subsequent operation accomplished by the match-making agent refers to the definition of the set of match-made clouds $S_M = S_R \cap S_{IdP}$.

We now define the metrics R_{req} and $R_{off}(F_i)$ representing respectively a measure of the resources requested by the home cloud, and a measure of the resources offered by the foreign cloud F_i. The value of the metric is obtained evaluating different parameters such as CPU, RAM, storage and QoS for both R_{req} and $R_{off}(F_i)$. In order to identify which foreign clouds fit the home cloud requirements, the match-making agent achieves a list of preferred foreign clouds $F_{ord(S_M)} = \{F_1, F_2, \ldots, F_n\}$ considering the set S_M and ordering its element by the R_{off} value, in a descending order.

Considering the example depicted in Figure 3, $S_M = \{A, D, B\}$ and $F_{ord(S_M)} = \{A, B, D\}$. The match-making agent has to consider the resources provided by the first k foreign clouds of $F_{ord(S_M)}$ to satisfy the condition $R_{req} \leq \sum_{i=1}^{k} R_{off}(F_i), 1 \leq k \leq n$ (in the scenario depicted in Figure 3, we assume $k = 2$ and consequently both foreign clouds A and B will be chosen to establish the federation).

In step 5 (authentication phase), in order to establish a federation with foreign cloud A and B, a cloud SSO *authentication* process has to be started by the home cloud. Such process will involve: the authentication agent of the home cloud, the corresponding peers of the foreign cloud A and B and the IdP X (trusted with A and B, on which the home cloud has a digital identity) where the home cloud performs a SSO log-in. Once the home cloud and foreign cloud A authentication agents establish a trust context, their respective underlying VI manager layers setup a low-level trust context allowing the cross-cloud resource provisioning. Therefore, the home cloud VI manager will be able to instantiate virtual resources on the foreign cloud VI manager. Even if cross-cloud federation has to be established also with foreign cloud D, no further authentication tasks would be needed because foreign cloud D has already a trusted relationship with IdP X.

As can be perceived, the employment of the IdPs presents some advantages well fitting our cross-cloud federation scenario: even though each cloud has its internal security mechanisms, whatever the *foreign cloud* is, regardless of its authentication mechanisms, by means of IdPs a *home cloud* will be able to authenticate itself with other foreign clouds already having a trust relationship, exploiting the well-known concept of SSO. The resource provisioning in cross-cloud federation may be solved establishing trust relationships between the clouds using several IdPs containing the credentials of the cloud asking for resources. Section V better describes the steps involved in phase 5, pointing out the technologies employed to implement the authentication and the set of information exchanged between the involved entities. The same Section describes our new SAML profile designed to accomplish the cloud SSO authentication in a federated scenario.

V. Authentication Practice Using SAML

In this Section, after a brief description of the SAML standard, we focus on the authentication phase of our three-phase cloud federation model performed by the Authentication Agent. More specifically, using the SAML technology we propose a new *Cross-Cloud Authentication Agent SSO Profile*, which describes the messages exchanging flow between a home cloud, foreign clouds and IdPs during the establishment of a trust context.

In order to explain the authentication process (step 5 of Figure 3), we consider the SAML technology. SAML is an XML-based standard for exchanging authentication and authorization assertions between security domains, more specifically, between an identity provider (IdP) (a producer

of assertions) and a generic service provider (SP) (a consumer of assertions). SAML consists of: a subject, a person or a software/hardware entity that assumes a particular digital identity and interacts with an online application, composed of several heterogeneous systems; a SP or relying party, a system, or administrative domain, that relies on information supplied to it by the Identity Provider; an IdP or asserting party, a system, or administrative domain, that asserts information about a subject. In literature, such model is also referred as IdP/SP.

SAML combines four key concepts: assertion, binding, protocol and profile. Assertion consists of a package of information that supplies one or more statements (i.e., authentication, attribute, and authorization decision) made by the IdP. Authentication statement is perhaps the most important meaning the IdP has authenticated a subject at a certain time. A Protocol (i.e., Authentication Request, Assertion Query and Request, Artifact Resolution, etc) defines how subject, service provider, and IdP might obtain assertions. More specifically, it describes how assertions and SAML elements are packaged within SAML request and response elements. A SAML binding (i.e., SAML SOAP, HTTP Redirect (GET), HTTP POST, etc) is a mapping of a SAML protocol message over standard messaging formats and/or communications protocols. A profile (i.e., Web Browser SSO, Enhanced Client or Proxy (ECP), Single Logout, Attribute, etc) is a technical description of how a particular combination of assertions, protocols, and bindings defines how SAML can be used to address particular scenarios.

A. Our Testbed Overview

In order to solve the cloud SSO authentication problem, pointed out in Section IV-A, we prearranged a testbed involving three clouds where OpenQRM [13] has been employed as VI manager. Although the cloud manager layer depicted in Figure 2 includes several modules performing different high level features, the software system deployed on our testbed just implements the authentication agent of the CCFM exposing both SOAP web services for the communication with other peers, and software interfaces for the communication with the underlying layer (OpenQRM). Along with such software implementation, we also extended the OpenQRM capabilities, enabling web service communications between different OpenQRM platforms (feature not supported natively), in order to permit the interaction between different clouds VI managers.

The authentication agent has been designed both to manage the digital identity of the home cloud and to perform authentication tasks sending/receiving authentication requests to/from foreign clouds, interacting with their respective peer modules. More specifically the authentication agent does not directly manages the digital identity of the cloud, but uses one or more trusted IdPs acting as guarantor when the agent likes to authenticate the home cloud with other foreign clouds during the federation establishment. As far as regards the authentication phase, the agent implements our own SAML profile (described in Section V-B), which defines the messages exchange flow between the home cloud, the foreign clouds, and the IdP, solving the cloud SSO authentication problem. More specifically, the implementation of the profile has been accomplished using and extending the java libraries of the OpenSAML project [14] for both the authentication agents and the IdP. In order to accomplish such tasks, the agent has been developed exposing a web service interface using the SOAP [15] technology, but nothing prevents the adoption of other web service technologies such as REST, JAX-RPC, or XML-RPC.

B. The SAML Cross-Cloud Authentication Agent SSO (CCAA-SSO) Profile

To address the cloud SSO authentication problem, during the cross-cloud federation establishment we developed a new SAML profile named Cross-Cloud Authentication Agent SSO (CCAA-SSO). Such profile was designed to enable a home cloud to perform SSO authentication on several foreign clouds both having a trusted relationship with the home cloud's IdP and regardless their security mechanisms. Such authentication process is fundamental for the subsequent establishment of a secure channel between the home cloud VI manager and one or more foreign cloud VI manager(s). Once the secure channel has been established the home cloud is able to gain the access to the required resources offered by the foreign cloud.

In a CCAA-SSO profile use case, both the home cloud and the foreign cloud, by means of their own Authentication Agents, represent respectively the subject and the relying party, whereas the IdP acts as the third party asserting to a foreign cloud the trustiness of the home cloud identity. The CCAA-SSO profile has been designed as a combination of the following SAML elements: an assertion including an authentication statement, a request-response protocol, and a SAML SOAP Binding.

Considering the scenario already pointed out in Section IV-B, in the following we describe the authentication process previously marked as phase 5 keeping in mind our SAML CCAA-SSO profile. In Figure 4 is shown the flow of messages exchanged between the home cloud, the foreign cloud A and the IdP X, putting aside for the time being foreign cloud B. More specifically, inside each cloud both Authentication Agent and the VI Manager are involved in the process.

In step 5.1 the Authentication Agent, on behalf of the home cloud manager, forwards to the corresponding peer of the foreign cloud A a SOAP request for a set of virtual resources by means of a XML document. In the SOAP request message reported below, such document is embedded inside the <ResourceType> element and is not depicted for briefness.

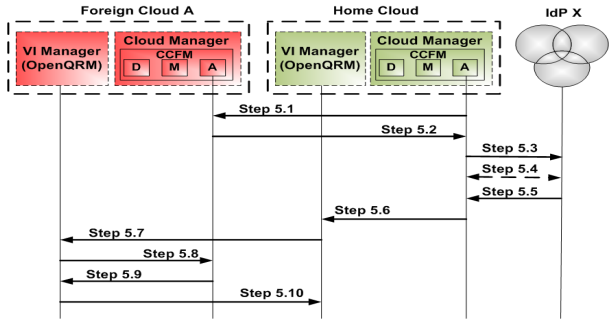

Figure 4. Sequence diagram describing the steps of the CCAA-SSO profile during the authentication of the home cloud with the foreign cloud A by means of the IdP X.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/
    envelope/">
  <S:Header/>
  <S:Body>
    <ns2:AA-ForeignCloud-A-ResReq xmlns:ns2="https://
        cloudA.net/SAML2/">
      <ResourceType>"XML resource description
          document"</ResourceType>
    </ns2:AA-ForeignCloud-A-ResReq>
  </S:Body>
</S:Envelope>
```

In step 5.2 the Authentication Agent of the foreign cloud A responds to the home cloud with a SAML authentication request containing an authentication query. Considering the underlying SAML/SOAP response, the authentication request is provided by means of the element <samlp:AuthnRequest...>.

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/
    envelope/">
  <S:Body>
    <ns2:AA-ForeignCloud-A-ResReqResponse xmlns:ns2="
        http://webservices/">
      <return>
        <samlp:AuthnRequest xmlns:samlp="urn:oasis:names
            :tc:SAML:2.0:protocol" xmlns:saml="urn:
            oasis:names:tc:SAML:2.0:assertion" ID="dfa6
            " Version="2.0" IssueInstant="2010-01-12T18
            :34:42Z" AssertionConsumerServiceIndex="0">
          <saml:Issuer>https://cloudA.net/SAML2</saml:
              Issuer>
          <samlp:NameIDPolicy
           AllowCreate="true"
           Format="urn:oasis:names:tc:SAML:2.0:nameid-
               format:transient"/>
        </samlp:AuthnRequest>
      </return>
    </ns2:AA-ForeignCloud-A-ResReqResponse>
  </S:Body>
</S:Envelope>
```

In step 5.3 the Authentication Agent of the home cloud unpacks the authentication request received at step 5.2 and forwards it via SAML/SOAP to the IdP X, making a SSO request. Since a valid trust context does not exist, in step 5.4 the IdP X authenticates the home cloud using a given security technology (the independence from the security technology used by each cloud is accomplished). In step 5.5, since the home cloud identity is verified, the IdP X responds to the authentication request by means of the following SAML/SOAP response, identified by the element <samlp:Response...>. Such element contains an assertion (see element <saml:Assertion...>) with an authentication statement (see element <saml:AuthnStatement...>) and has been signed by the IdP X (see elements <saml:Issuer> and <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" >).

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/
    envelope/">
<S:Body>
<ns2:IdpX-SSO-ServiceResponse xmlns:ns2="http://
    webservices/">
<return>
  <samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML
      :2.0:protocol" xmlns:saml="urn:oasis:names:tc:
      SAML:2.0:assertion" ID="7d46" InResponseTo="dfa6"
       Version="2.0" IssueInstant="2010-01-12T18:35:23Z
      " Destination="https://cloudA.net/SAML2/SSO/SOAP
      ">
    <saml:Issuer>https://idpx.net/SAML2</saml:Issuer>
    <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:
        status:Success"/>
    </samlp:Status>
    <saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML
        :2.0:assertion" ID="f8ab" Version="2.0"
        IssueInstant="2010-01-12T18:35:23Z">
      <saml:Issuer>https://idpx.net/SAML2</saml:Issuer>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
          xmldsig#">
       mgQpzcz4tfZDPCOZfGhyodpRrYHbk4Le/i+
           iynUjpW2uAgCvPJTSweVTofRcy8tHrvz6h5g2KOdB
       8XY9+h/4euIVxg5vXuD6PldBqWgKYtY84+910IP7TXQJS/
           cblOCIf2TdMo55vR0QGDYdBt2yRXd1
       wCUfbWNBy97ODoEvTtptJtpjp9NNkZS7g9w0TJFKlI/
           OJUOO93dtaSAF6WVid55JE4oraYFEFMfO
       hdttW0jOIazNLSIr8qp7mt0C8jWLBrRsIChVGDML4s+
           xEyyN4hrCEvz2hIcLYA5Q4B1HTKryMCw5
       PIJt0eaTeMicjAyrNLuJnLmMmDbE50KsRoo+yA==
      </ds:Signature>
      <saml:Subject>
        <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:
            nameid-format:transient">
          5a42edc7-6439-4de9-12d2-836a74df279c
        </saml:NameID>
        <saml:SubjectConfirmation Method="urn:oasis:names:
            tc:SAML:2.0:cm:bearer">
          <saml:SubjectConfirmationData InResponseTo="dfa6"
              Recipient="https://cloudA.net/SAML2/SSO/
              SOAP" NotOnOrAfter="2010-01-12T18:40:23Z"/>
        </saml:SubjectConfirmation>
      </saml:Subject>
      <saml:Conditions NotBefore="2010-01-12T18:30:23Z"
          NotOnOrAfter="2010-01-12T18:40:23Z">
        <saml:AudienceRestriction>
          <saml:Audience>https://cloudA.net/SAML2</saml:
              Audience>
        </saml:AudienceRestriction>
      </saml:Conditions>
      <saml:AuthnStatement AuthnInstant="2010-01-12T18
          :35:17Z" SessionIndex="73d8">
        <saml:AuthnContext>
          <saml:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:
                PasswordProtectedTransport
          </saml:AuthnContextClassRef>
        </saml:AuthnContext>
      </saml:AuthnStatement>
    </saml:Assertion>
  </samlp:Response>
</return>
</ns2:IdpX-SSO-ServiceResponse>
</S:Body>
</S:Envelope>
```

In step 5.6 the Authentication Agent of the home cloud unpacks the authentication assertion received in step 5.5 and forwards it to the underlying VI Manager. In step 5.7 the VI manager of the home cloud sends the authentication assertion via SAML/SOAP to the corresponding peer of the foreign cloud A. In step 5.8 the VI manager of the foreign cloud B forwards the received authentication statement to its authentication agent, which verifies its correctness. In step 5.9 the VI manager of the foreign cloud B receives a notification about the authentication assertion validity and allocates the resources requested by the home cloud at step 5.1. In step 5.10 the VI manager of the foreign cloud contacts its peer on the home cloud notifying where and how to access the requested resources, for example establishing a secure communication channel.

The authentication process of the home cloud with the foreign cloud B is analogous to the one already described for foreign cloud A, with one important difference: since the home cloud has already performed the authentication on the IdP X in phase 5.4, no further authentication is needed because a trust context already exists (the SSO is thus accomplished). Therefore, the SAML CCAA-SSO profile combines both security and flexibility ensuring cloud SSO authentication in cross-cloud federation environments between clouds, using different security technologies representing a possible solution for secure federated cloud interactions.

VI. Conclusions and future works

In this paper we tackled the cross-cloud federation problem performing an in depth analysis and proposing a three-phases model. An architectural solution including the Cross-Cloud Federation Manager (CCFM) has been designed and described. Furthermore, an implementation practice of the Authentication Agent using a SAML CCAA-SSO profile has been developed and some examples are also proposed. In future, we plan to study the performances of such cross-cloud federation scenario, evaluating the amount of authentications and IdP enrollments needed, either employing real testbeds or by means of a simulated environment, including hundreds of clouds dynamically joining and leaving federations.

Acknowledgements

The research leading to the results presented in this paper has received funding from the European Union's seventh framework programme (FP7 2007-2013) Project RESERVOIR under grant agreeement number 215605.

References

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE '08*, pp. 1–10, 2008.

[2] T. Bittman, "The evolution of the cloud computing market," *Gartner Blog Network, http://blogs.gartner.com/thomas_bittman/2008/11/03/the-evolution-of-the-cloud-computing-market/*, November 2008.

[3] SAML V2.0 Technical Overview, OASIS, http://www.oasis-open.org/committees/ download.php/11511/sstc-saml-tech-overview-2.0-draft-10.pdf.

[4] Sun Microsystems, Take your business to a Higher Level - Sun cloud computing technology scales your infrastructure to take advantage of new business opportunities, guide, April 2009.

[5] W. Li and L. Ping, "Trust model to enhance security and interoperability of cloud environment," in *Cloud Computing*, pp. 69–79, November 2009.

[6] N. Leavitt, "Is cloud computing really ready for prime time?," *Computer*, pp. 15–20, January 2009.

[7] Liberty Alleance Project, http://projectliberty.org.

[8] OpenID Authentication 2.0, OpenID Foundation, http://openid.net/specs/openid-attribute-exchange-2_0.html, 2007.

[9] The Shibboleth system standards, http://www.openqrm.com.

[10] Microsoft Windows Cardspace, http://netfx3.com/content/WindowsCardspaceHome.aspx.

[11] S. Pearson, Y. Shen, and M. Mowbray, "A privacy manager for cloud computing," in *Cloud Computing*, pp. 90–106, November 2009.

[12] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *Internet Computing, IEEE*, vol. 13, pp. 14–22, September 2009.

[13] OpenQRM, "the next generation, open-source Data-center management platform", http://www.openqrm.com/.

[14] OpenSAML, "Open source libraries in Java and C++ providing core message, binding, and profile classes for implementing applications based on SAML 1.0, 1.1, and 2.0", http://saml.xml.org/internet2-opensaml.

[15] "Web services security: Soap message security 1.0, oasis, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf."