Scalable and Privacy-preserving Access Mechanism for Dynamic Clouds

Uttam Thakore
Spring 2012
Summa Cum Laude
Bachelor of Science in Computer Engineering

Advisor: Dr. Shigang Chen

Table of Contents

1. Introduction

Cloud computing, as defined by NIST, is a model for enabling always-on, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., storage, applications, services, etc.) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. It allows users to "rent" computing resources in a pay-as-you-go fashion without needing to outright purchase the hardware and software, which eliminates the need for a large upfront capital investment. Cloud computing also allows for massive scalability and greater flexibility than a traditional IT service model for a relatively constant price. For example, a cloud user can provision 1000 hours of computational power on a single cloud instance for the same price as 1 hour of computational power on one thousand cloud instances for the same price, and can vary its usage based on its immediate needs [2].

As the number of organizations using cloud services increases, the issue of security of the data stored in the cloud and transmitted between the cloud provider and cloud consumer becomes a growing concern. In fact, in the Cloud Computing Services Survey done by IDC IT group in 2009, the number one issue for adoption of cloud computing, cited by over 87% of those surveyed, was security [3]. Security is comprised of many aspects, including auditing and compliance with regulations, information flow control, fault tolerance, intrusion detection and prevention, and management of identity and access control. The most fundamental and most challenging of these is identity and access management.

Identity and access management (IAM) is the process of assigning identities to users and controlling their access to a computer system. IAM can be broken down into three components: identity management, authentication, and authorization. Identity management refers to the creation and management of identities for users. Authentication refers to the verification of a user's identity. Authorization refers to the verification that a user is permitted to perform a particular action. In cloud systems, in which a cloud consumer likely uses multiple cloud services on different cloud service providers, users must be identified at each of the providers and their access to the cloud systems must be controlled for each service and resource they attempt to access.

In order to provide access to a system, private information about a user must be exchanged. For example, a typical bank login page requests an answer to a security question, the answer to which often contains information that could be used to identify the user. Other services require that a user be above a certain age, which requires that the user's date of birth be checked during authorization. In a cloud environment, this presents a threat to privacy, as this sensitive information must be passed to each cloud system and service to which a user is requesting access. Often, a cloud provider is opaque about its handling of such information, so a user may not know who could have access to the information throughout her use of the cloud service. As a result, it is necessary to consider the privacy of user identity information as it is used during the IAM process.

One of the major advantages to cloud computing is the flexibility and dynamicity it provides in provisioning computing resources as needed. An organization only needs to request as much computing power as it currently needs, instead of provisioning for its peak requirements, which significantly decreases its cost of using the resources. As organizations adopt the cloud, they will push for even more dynamicity in cloud offerings. Organizations will want to use public cloud services as an extension to their own applications, rapidly forming a cloud when they are overextended and destroying the cloud when their need diminishes. They will also desire the ability to rapidly scale up and down the number of cloud services they use as their need for those services changes. Furthermore, organizations will desire the flexibility to choose from various cloud offerings and switch between different cloud service providers as prices fluctuate without significant vendor lock-in or transitional overhead. It is therefore important to examine IAM within the context of dynamic cloud computing.

In this thesis, I propose a scalable mechanism for IAM in dynamic clouds that preserves the privacy of the cloud user. The key tenets of my approach are the centralization of all identity and authorization information, management, and decision making; the elimination of the need to provision user identity or authorization policy information at each cloud service provider; and the use of temporary identifiers to identify users at cloud service providers. The rest of my thesis is organized as follows. In section 2, I define the terms and ideas that are necessary to understand the IAM process within the context of dynamic clouds. In section 3, I present research that has been done in the area and analyze the suitability of the approaches presented for scalability and preservation of privacy in dynamic clouds. In section 4, I propose my approach for solving the aforementioned problem. Finally, in section 5, I conclude my thesis and identify areas of future work.

2. Background

2.1. Service Oriented Architecture (SOA)

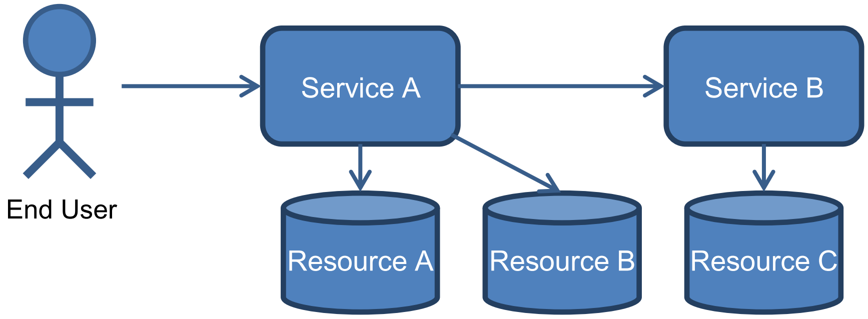

Figure 1: Service Oriented Architecture

Cloud computing is an extension of the Service Oriented Architecture (SOA) model of computing. In a SOA system, applications and business functions are packaged into independent, modular components called *services*. A service is a self-contained, discoverable resource that executes a repeatable task. Services can be invoked individually, or may be combined and orchestrated to perform a more complex task in a *composite application*.

There are two types of parties in a service-oriented system:

1. Parties that offer services are called *service providers*.
2. Parties that request and use these services are called *service requestors*.

In Figure 1, shown above, the presentation component used by the End User is a service requestor of Service A, and the organization providing Service A is the service provider. Similarly, Service A is a service requestor of Service B, and Service B is provided by a service provider.

There are many advantages to service orientation. Take the example of a bank system, which may provide many functions to its users, such as account creation, depositing, withdrawal, and checking, each of which encapsulates some business logic. In a non-service oriented system, the bank would combine the functions into a monolithic application and provide the application to its customers. This approach is inflexible and makes it difficult to separately execute the different functions or to easily put together individual functions to perform an operation not supported by the bank's application. If the bank were to implement a service oriented system, each of these functions would be provided to the bank's customers as separate services, which the customers could then integrate with other services to meet their individual business process needs.

While services provide business logic, the actual state or functionality needed to fulfill the responsibilities of the services is provided by what are called *service resources*. Resources can be data (stored in a database, for example), a back-end system, or other services. In Figure 1 above, Service A accesses Resources A and B and Service B, which in turn accesses Resource C. The resources of Service A are therefore Resources A and B and Service B.

2.2. Cloud computing

Cloud computing, as defined by NIST, is a model for enabling always-on, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., storage, applications, services, etc.) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. At the core of cloud computing is a datacenter that uses virtualization to isolate instances of applications or services being hosted on the cloud. The datacenter provides cloud users the ability to rent computing resources at a rate dependent on the datacenter services being requested by the cloud user.

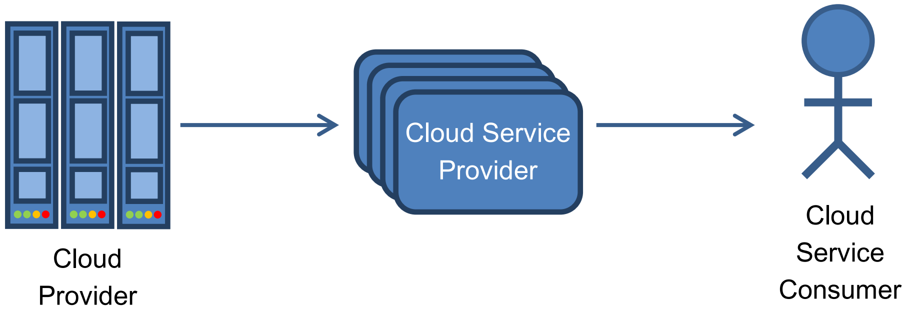

Figure 2: Cloud computing parties

It is important to define the roles of the parties involved a cloud computing solution. Figure 2 illustrates the relationship between each party. The organization providing the datacenter and related management services is the cloud provider. The organization or entity using the cloud to host applications, in the form of services, for use by other individuals and/or organizations is called the cloud service provider. The individuals and/or organizations using the services hosted on the cloud are called the cloud service consumers. At times, the cloud provider may also be the cloud service provider, such as in the case where the service being used by the cloud user is in fact the infrastructure provided by the cloud provider for hosting applications.

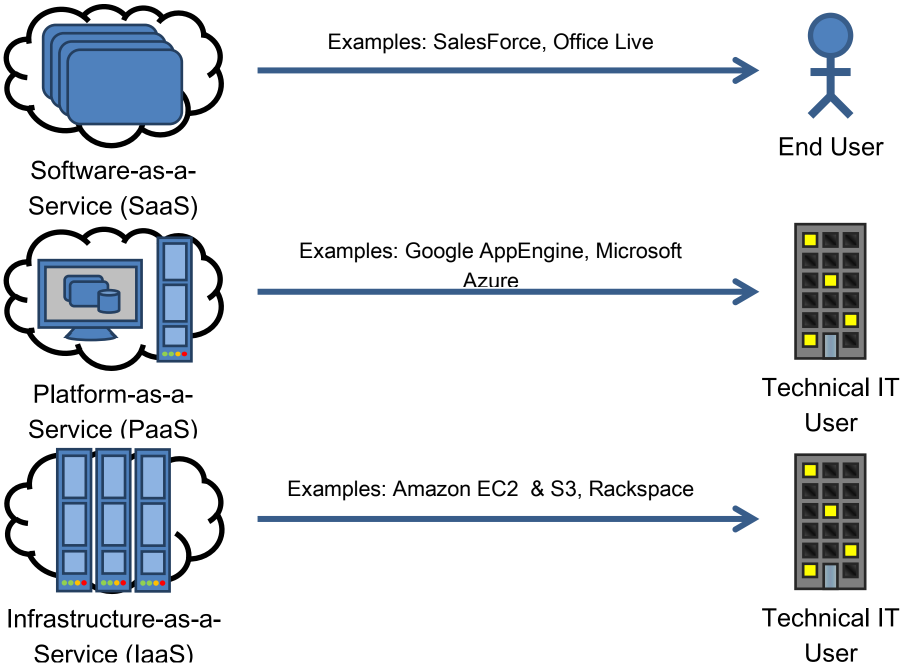
Software-as-a-Service (SaaS)
Examples: SalesForce, Office Live
End User
Platform-as-a-Service (PaaS)
Examples: Google AppEngine, Microsoft Azure
Technical IT User
Infrastructure-as-a-Service (IaaS)
Examples: Amazon EC2 & S3, Rackspace
Technical IT User

Figure 3: Cloud computing service models

Cloud service providers can be classified into three *service models* based on the types of services they provide. They are (as defined by NIST):

- Software as a Service (SaaS) – The cloud service provider provides the cloud consumer with either the capability to deploy an application on a cloud infrastructure, or use of an existing application. The application is available to users through a network interface, such as a thin-client website. The cloud consumer does not control the underlying infrastructure except for user-specific settings [1]. Examples of SaaS offerings are given in Figure 3.
- Platform as a Service (SaaS) – The cloud service provider provides the cloud consumer with the capability to develop and deploy applications on a cloud infrastructure using tools, runtimes,

libraries, and services supported by the cloud service provider. The cloud consumer does not control the underlying infrastructure, such as the hardware, virtualization, operating systems, or storage, but may have control over the application hosting environment and settings for the tools available for development and deployment [1]. Examples of PaaS offerings are given in Figure 3.

- Infrastructure as a Service (SaaS) – The cloud service provider provides the cloud consumer with essentially a virtual machine. The cloud consumer has the ability to provision processing, storage, networks, etc., and to deploy and run arbitrary software supported by the operating system run by the virtual machine. The cloud consumer has control over the operating system, storage, applications deployed on the VM, and potentially some networking settings (such as firewalls), but it does not control the underlying infrastructure of the cloud provider, such as the hardware or virtualization hypervisor [1]. Examples of IaaS offerings are given in Figure 3.

Other literature about cloud computing defines other X-as-a-Service, such as Hardware-as-a-service (HaaS), Storage-as-a-Service, Databases-as-a-Service, or Business-Processes-as-a-Service, but these terms are not as commonly accepted or used, and can often be classified into the SPI (Software, Platform, and Infrastructure) service models [1]. All cloud offerings can be classified into one of the service models mentioned above.

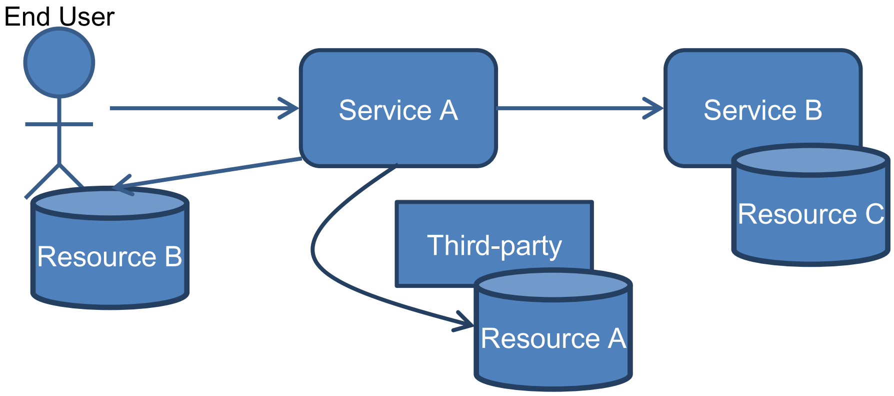

Figure 4: Cloud resource ownership

Cloud consumers often use multiple cloud services, potentially from multiple service providers, to complete a single transaction. Services within a single business process may even invoke multiple services from different cloud service providers. The resources accessed by these cloud services may be located with the cloud service providers, with a third party, or even with the cloud consumer itself. In order to ensure secured access, cloud systems must allow authorized users access to the right resources, and restrict access to all others.

Figure 4 shows the same scenario as in Figure 1, except in a cloud setting. In this setting, Resource A is owned by a third-party, Resource B is owned by the End User itself, and Resource C is owned by Cloud Service B. To provide service results to the End User, Cloud Service A goes to the Third-party to get access to Resource A, accesses Cloud Service B, and goes to the End User to get access to Resource B.

2.3. Dynamic Clouds

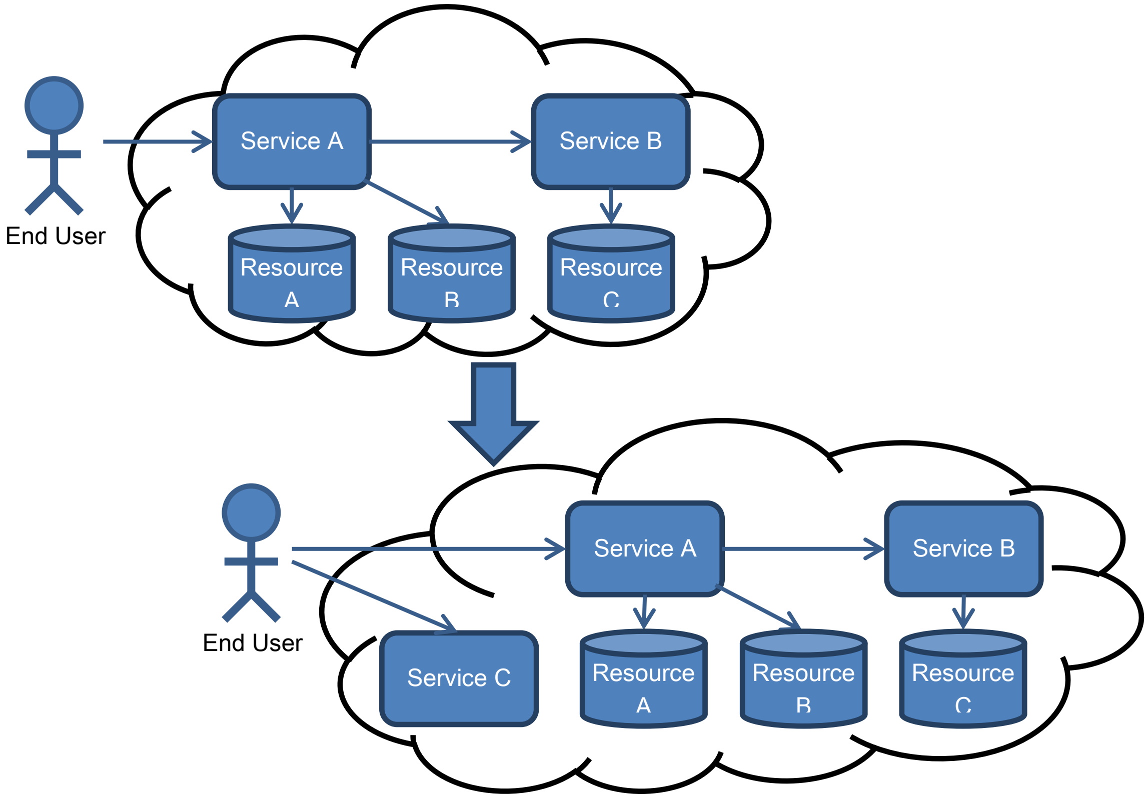

Figure 5: Dynamic cloud formation

As the cloud computing industry matures, I believe that customers will start to demand quick provisioning and deprovisioning of cloud services that support specific business processes. Organizations will want to use public cloud services as an extension to their private clouds when they are overextended, such as during peak times. If a cloud service provider cannot meet or is not meeting desired service levels, or if another cloud service provider is providing the same service at a cheaper rate, organizations will want the flexibility to switch between different cloud service providers without large overhead or vendor lock-in.

For example, in Figure 5 above, if the End User wants to provision access to Service C for one month to handle higher traffic for the holiday season, it should be able to form a new cloud containing Cloud Service Provider C without having to go through cumbersome trust and business relationship agreements or a lengthy user provisioning process. Service C should be provisioned and ready to use within the course of a few minutes to hours, depending on the time constraints of the End User.

This highly dynamic approach will only be successful if the clouds can be formed rapidly and access can be provisioned on-the-fly, so it will be important that cloud providers be able to support rapid dynamicity in the formation of clouds and in the provisioning of services. While dynamic clouds may involve quality-of-serve and distributed computing issues [15,16,17] in a network-wide scope where cloud computing is implemented in a distributed manner, this thesis will focus the security aspect of such a computing paradigm.

2.4. Securing Access in a Cloud Computing Environment

To best explain the concepts within cloud computing access control, I will illustrate a typical access scenario with a simple example. The organizations within this scenario include:

- Example University – a university that has online components to its classes. The university has partnered with BBCo to provide students access to a student blackboard web service through which they can access assignment and course information, lecture videos, and a course discussion board for each of their courses.
- BBCo – a company that provides cloud-based customized student blackboard service solutions for universities that want to provide online components to their classes. BBCo has its own databases for the discussion board and course information, but uses e-mail services provided by its client universities and a video streaming service provided by LiveClassroom.
- LiveClassroom – a cloud-based video streaming service for universities that provides the ability to upload recorded classroom lectures and stream them to students.

Example University wants to provide its students with a student blackboard service, through which they can view course information and documents, communicate with other students using a course discussion board, and view recorded course lectures. Example University partners with BBCo to provision access to BBCo's student blackboard service for its students. BBCo's student blackboard uses a local database resource to store the course information and documents and course discussion board content. Within its service, it contains a portlet to LiveClassroom, a cloud-based video streaming service, to stream course videos to the students. It also uses an e-mail service provided by Example University to allow students to send e-mails to the professors, TAs, and other students in their courses.

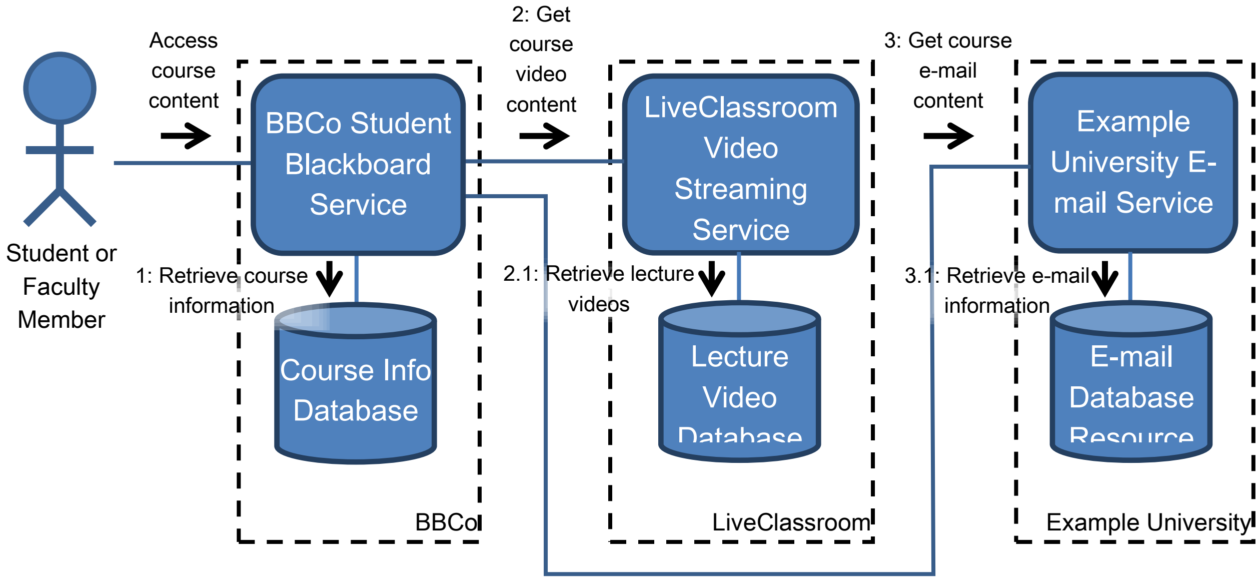
Student or Faculty Member
Access course content
BBCo Student Blackboard Service
1: Retrieve course information
Course Info Database
BBCo
2: Get course video content
LiveClassroom Video Streaming Service
2.1: Retrieve lecture videos
Lecture Video Database
LiveClassroom
3: Get course e-mail content
Example University E-mail Service
3.1: Retrieve e-mail information
E-mail Database Resource
Example University

Figure 6: Student blackboard service collaboration diagram

The collaboration diagram shown in Figure 6 above provides a model of the interactions between the parties in the scenario given above. Students and faculty members of Example University access the blackboard system directly through BBCo's web interface. BBCo directs requests for the course video streams to LiveClassroom's cloud video streaming service, and directs requests to send e-mails to Example University's e-mail service. Requests for all other course information are handled directly by BBCo through an access of BBCo's database resource for course information.

Notice that in the scenario above, resources used by BBCo's student blackboard cloud service are owned by multiple parties. The course information database is owned and managed by BBCo, the lecture video database is owned by LiveClassroom, and the student e-mail database is owned by Example

University. BBCo's student blackboard service also accesses cloud services other than its own – in this case, it uses LiveClassroom's lecture video streaming service to provide content to Example University's students.

Security Policies

Access rules for services and resources are defined by security policies. *Security policies* refer to the set of mechanisms by means of which an organization can define and achieve its security objectives. Security policies are used at all levels of security in the scenario above. For example, at an organizational level, a security policy may define the roles and levels of access students, faculty, and administrators of Example University are granted to BBCo's blackboard service. At a transport level, a security policy may define the secure communication protocol used between BBCo and LiveClassroom when transmitting authentication data. At the data level, a security policy may define the encryption method used when storing data in BBCo's course information database.

Identity and Access Management

In a dynamic cloud with multiple cloud services offered by different cloud service providers, such as the scenario given above, resources can be scattered throughout the cloud and be accessed from a plurality of the cloud services. Ensuring easy and quick provisioning (and deprovisioning) and efficient access to the various cloud systems, their resident services, and the resources they use is a challenge that must be addressed. This challenge can be broken into three primary tasks: identity management, authentication, and authorization [4].

Identity Management

Identity is the set of data that uniquely defines a user and distinguishes them from others. All users in a system must have an identity so they can be given access to the resources and services within the system in a secured manner.

For example, in the student blackboard scenario above, all students in the system would need identities in the blackboard system to be able to access the course content specific to them. In addition, each service would need an identity to access other services; for example, for the student blackboard service to gain access to LiveClassroom's video streaming service, it would need to be able to identify itself as BBCo's blackboard service.

Identity attributes are the individual pieces of information about a user that define the user and its interactions with other users. Identity attributes can be something a user has, such as a name or address, something a user knows, such as a password or PIN, or something the user is, such as a retinal scan or fingerprint. In the case of most cloud systems, identity attributes can be classified into the first two categories.

Identity attributes often contain information that can personally identify a user, such as a Social Security Number or name and address. Such information is called *Personally Identifiable Information*, or *PII*. Protection of PII from unwanted or unauthorized dissemination is often a legal obligation, and falls in the realm of privacy protection. *Privacy* is the right of individuals to determine how, when, and to what extent their personal information is shared with other parties. As cloud service consumers have little knowledge about the inner workings of a cloud service provider's system, data privacy concerns are common, so considerable emphasis must be placed on privacy protection.

Identity management refers to the creation, modification, and deletion of identity objects. Identity management provides the first line of access control for a system. For a user to be able to access the system, the system must first be able to identify the user and determine whether the user should be given access. To do this, the user must be given credentials, in the form of an identity object. The credentials associated with the identity object could be any uniquely identifying combination of identity attributes, such as a username and password or ID number. The purpose of an identity management system is to allow a business to create an identity object for a user, manage the identities and their properties (e.g., to what groups a user belongs, when a user’s password should automatically expire, etc.), and delete the identities once the user no longer needs to access the system.

In the scenario above, in order to provide secure access to the blackboard service, Example University must maintain an identity for all of the students who will be using the blackboard service. BBCo and LiveClassroom must also maintain some identity information for all of the students in order to provide customized information to the students.

Identity Management in a Cloud Environment

In a dynamic cloud environment, a single transaction often involves multiple cloud services, and hence, passes through multiple cloud service providers. This requires that each provider maintain some level of identity information for each of the users of the cloud service. It is imperative that these identities are managed throughout the service providers in such a way that a user’s identities across the cloud are linked together. This is referred to as *federated identity management* [5]. The cloud service providers taking part in the federation are collectively referred to as an *identity federation*.

The two most important roles in a federation are the identity provider and the service provider [6].

- An *Identity Provider* (IdP) is the party in an identity federation that provides assurances of the identity of a user to other parties in the federation. The IdP is responsible for management of users and their identities, issuance of credentials, and handling user administration.
- A *Service Provider* (SP) is the party in a federation that consumes the identity information provided by the identity provider and provides access to a service based on the asserted identity. The SP is responsible for controlling access to services based on asserted identity, validation of the asserted identity information, and management of locally relevant user attributes as needed to perform its other roles.

In the scenario given above, Example University serves as the IdP, and BBCo and LiveClassroom serve as SPs.

Trust

In order to be able to place credence in the identity information being provided by the IdP, and more generally, for any of the parties in the above scenario – Example University, BBCo, and LiveClassroom – to be able to have faith in an interaction with any other party, trust must first be established between the parties. This is usually done using a business agreement that establishes liabilities for improper use of information or falsification of results in interactions. Trust is further established through the designation of security policies for information flow that allow the parties in the interaction to have confidence that information in transit will not be leaked.

Authentication

Once identities have been created, when a user attempts to access a service, the service provider must be able to validate the identity of the user. *Authentication* is the process of identifying a user, ensuring that the user is whom it claims to be. Authentication serves as the first level of defense against illegitimate access to a system by permitting only those users who have identities on the system to access it.

Authentication takes place when a user attempts to access a service. The user is redirected to the IdP, which prompts the user for an identity credential. The IdP then attempts to validate the credential, and if the user’s credentials match with those stored by the IdP, issues an identity token to the user. An *identity token* is a digitally signed object that asserts the identity of a user. The user can then pass this identity token to the SPs, which verify the authenticity of the token using the digital signature and provide access to their services.

Authentication in the Cloud – Federated Single Sign-On

In a dynamic cloud computing environment, a user may need to access multiple cloud services provided by multiple cloud service providers over the course of a single transaction. If the user is required to authenticate at each of the service providers, authentication can become cumbersome and does not scale well with the size of the cloud [7].

A solution to this problem is Federated Single Sign-On, or federated SSO. Within federated SSO, a user can authenticate once with the IdP and provide the identity token given to her by the IdP to all SPs to which she wants to authenticate. The SPs in the federation trust the IdP to authenticate the user and can verify the validity of the identity token generated by the IdP.

Figure 7 below illustrates federated SSO in the student blackboard scenario given above. Federation of identity is performed using temporary, randomly generated identifiers, called *pseudonyms*, to identify the user [8]. There are other methods for performing federated SSO, but in this scenario, I focus on the method that best preserves the privacy of the user.

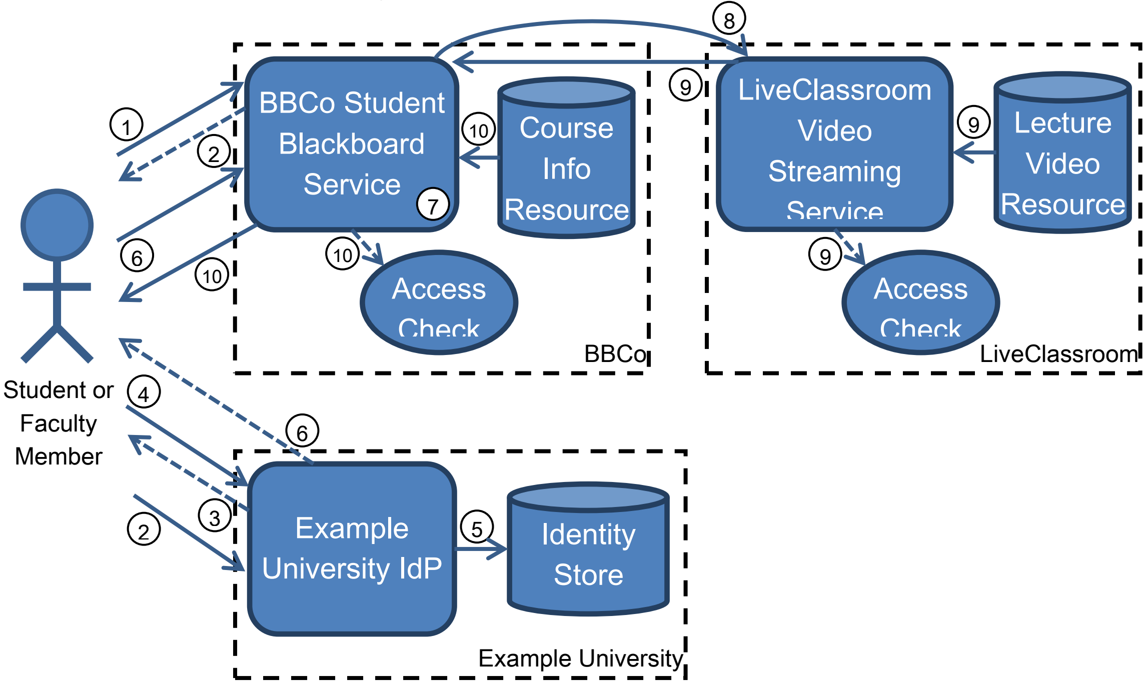

Figure 7: Student blackboard service cloud authentication

The steps in the scenario are as follows:

1. A student attempts to access BBCo's student blackboard service.
2. As the user has not yet authenticated with the service, she is sent a response redirecting her to the Example University to be authenticated. The page being requested is temporarily saved by BBCo.
3. Example University prompts the student to provide her university username and password.
4. The student provides her credentials, which are validated by Example University.
5. Example University looks up the user in its identity store and creates a transient, or temporary, pseudonym to return to the SP. This transient pseudonym will be used by the three cloud service providers throughout the session.
6. Example University generates an identity token using the transient pseudonym, signs it, and sends it to the user to pass along to BBCo.
7. BBCo validates the identity token and dynamically creates a session for the user using the pseudonym.
8. BBCo forwards the identity token to LiveClassroom, which also dynamically creates a session for the user.

9. LiveClassroom performs an access check, as discussed in the next section, and provides BBCo's blackboard service access to the lecture video stream if it is successful.
10. BBCo then performs an access check and provides the user access to the student blackboard service if it is successful, embedding the content from LiveClassroom.

In the scenario above, when a student from Example University accesses BBCo's student blackboard service, BBCo's use of SSO to pass identity information along to LiveClassroom alleviates the student of Example University of the need to re-authenticate with LiveClassroom. This also eliminates the need for the student to remember multiple credentials, as Example University can keep track of the credentials required by each service provider and provide them upon an authentication request.

The use of pseudonyms in the above scenario also enhances the privacy of the student. When a student authenticates with BBCo, BBCo is not provided full identity information about the student. Instead, BBCo receives a token with a transient pseudonym that cannot be used to identify the user outside of the session, and just enough identity information to provide the appropriate access to the student. Similarly, LiveClassroom receives only a token with a transient pseudonym and enough identity information to grant access to the video streaming service. Since neither BBCo nor LiveClassroom are given personally identifiable identity information, and neither need to store such information locally to provide authentication, the privacy of the student is preserved.

Authorization

Once a user has been authenticated, she must be authorized to gain access to the services and resources provided by the SP. *Authorization* is the act of granting users permissions to services and resources by specifying access rights. Access rights are defined in terms of authorization policies. Authorization policies specify the conditions under which the user is permitted to access a resource. For example, access to a resource or service may be limited to a particular user group, to users above 21 years or age, or to users living within a specific location.

Authorization involves the interplay of the following components:

- The *authorization policy store* holds the authorization policies that define access rights for the resource or service managed by the authorization system.
- The *Policy Decision Point* (PDP) takes an authorization request, retrieves the appropriate policy information from the policy database, and evaluates the request based on the user's identity attributes, service or resource properties, and authorization policy. It returns a yes/no response indicating whether or not a user should be granted access to the service or resource specified.
- The *Policy Enforcement Point* (PEP) accepts authorization requests and forwards the requests to the PDP along with the user's identity attributes and properties of the service or resource to which authorization is being requested. It returns the results of the authorization requests given by the PDP.
- The *resource manager* receives requests for a resource or service, sends authorization requests to the PEP, and grants or denies access to the resource based on the result received from the PEP.

Authorization takes place when a user attempts to access a service or resource, after the user has authenticated. The resource manager receives the request for the service or resource and generates an authorization request to send to the PEP. The PEP gathers necessary user identity and system attributes and forwards the request to the PDP, which retrieves the authorization policies and evaluates the authorization request using the information provided to it by the PEP. The PDP then returns its decision to the PEP, which forwards it to the resource manager to grant or deny access to the service or resource.

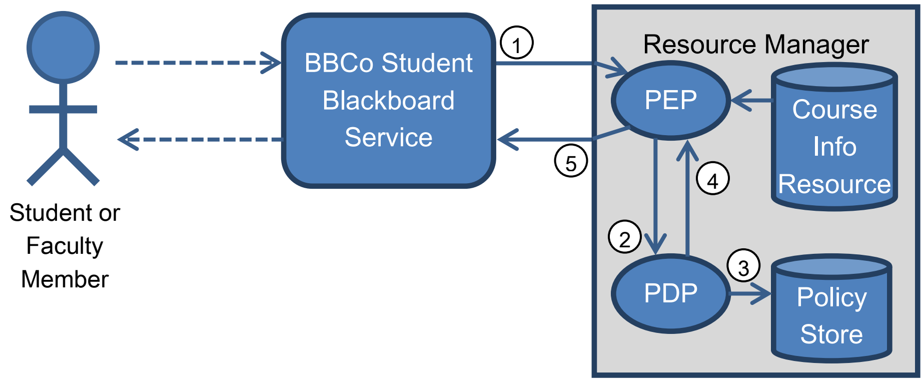

Figure 8: Student blackboard service non-cloud authorization

For example, in the scenario above, when a student accesses BBCo's student blackboard service, the service must retrieve discussion board posts and course information from its course information database resource. BBCo must first verify that the student is authorized to gain access to the database resource, which it does by comparing the student's ID number with a course enrollment list. Figure 8 shows the interaction of the authorization components within BBCo's blackboard service. After the student authenticates to the service, the blackboard service sends an access request to the resource's resource manager. The resource manager then sends an authorization request to the blackboard service's PEP, which gathers the student's ID number, the course number, and properties of the information being requested, and passes the request along to the PDP. The PDP looks up the authorization policy information from the policy store, evaluates the request, and returns the response back up through the PEP to the resource manager. Notice that in order to perform the authorization, BBCo required the student's ID number, which is PII.

Authorization in a Cloud Environment

In a cloud environment, a user may consume many services provided by a multitude of service providers within a single transaction. These services may access resources owned by numerous parties. In order for these services and resources to support authorization, the user must provide each service and resource with policy information.

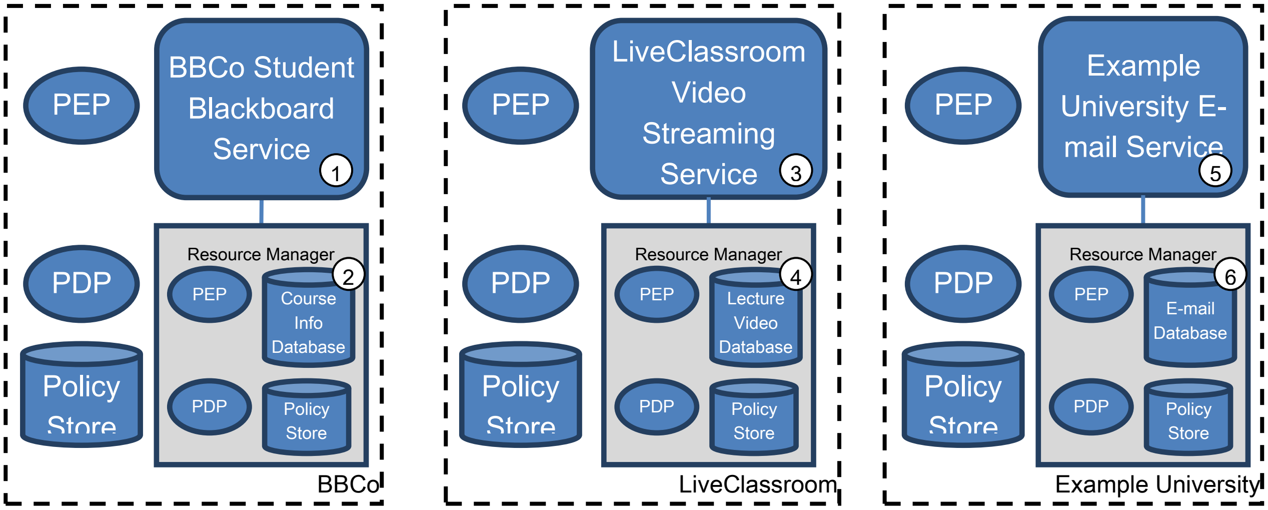

Figure 9: Student blackboard service cloud authorization

In our university scenario above, when Example University provisions access to the blackboard service for each of its students, it must communicate authorization policies to each resource manager in Figure 9 above. When a student accesses BBCo’s student blackboard service, she must be authorized to the following services and resources in the order shown in the figure:

1. BBCo’s blackboard service
2. BBCo’s database resource in order to retrieve the relevant course information
3. LiveClassroom’s video streaming service
4. LiveClassroom’s lecture video database resource to retrieve the lecture video stream
5. Example University’s e-mail service
6. As necessary for retrieving e-mails, Example University’s e-mail database resource

At each of these points, authorization occurs as described in the previous section.

Provisioning and Deprovisioning

Once the identity of a user has been established, the identity and authorization policies must be provisioned. *Provisioning* is the process of providing users with identities appropriate access to services and resources [4]. *Deprovisioning* is the process of revoking users’ access to services and resources when their access rights have expired.

Provisioning takes place at two stages in the identity and access management process:

1. *Identity provisioning* refers to the linking of identity objects with services to provide users the ability to authenticate with those services. Identity provisioning is needed to support federated SSO, since the IdP must be able to link the identity credentials of a user to an identity local to the SP to perform SSO.
2. *Access provisioning* refers to the creation and transmission of authorization policies to a service provider to define the access rights of users accessing the services and resources provided by the SP. Access provisioning is needed to support authorization to the various services and resources in a cloud environment.

In the scenario above, identities for each of Example University’s students must be provisioned to BBCo’s student blackboard service, LiveClassroom’s video streaming service, and to Example University’s e-mail service, as well as all of their database resources. Example University must also provision access rights for all of its students to BBCo’s, LiveClassroom’s, and its own services and resources.

3. Survey of existing research

A number of researchers have already investigated scalability, privacy preservation, and dynamicity in identity and access management in cloud computing.

In order to address scalability concerns in management of trust relationships for federated identity management, researchers at IBM Research – China [9] suggest using a brokered trust model, in which a third-party broker server is used to establish the trust with a cloud service user. The business agreement between a cloud service provider and the identity broker allows the CSP to place trust in the broker, allowing it to act as an agent for the CSP to establish trust with other parties, such as organizations using cloud services [9]. The organizations can then take advantage of their own identity federation services to relay credential information for authentication with the cloud service.

Such an approach reduces the CSP's cost of establishing multiple trust relationships with multiple service users. It also pushes complexity to the trust broker, which can handle support of more forms of federated identities. From the consumer's perspective, if multiple CSPs utilize same trust broker, establishing trust with multiple different types of services can be done by establishing trust with a single trust broker. While this approach addresses scalable management of trust relationships for federated identity management and authentication, it does not address the challenge of scalability of authorization in large dynamic clouds.

Eric Olden [10] also addresses the issue of scalability of federated identity management in large clouds. In a one-to-one model of identity management, the number of credentials required for users of an organization to utilize cloud services increases multiplicatively. If an organization has an extensive set of users and uses a number of cloud services, such a model rapidly becomes unmanageable. Olden introduces the idea of an identity fabric, which preintegrates with multiple cloud services to allow an organization to federate once to gain access to all of the services. With an identity fabric model, the number of credentials required for users to utilize multiple cloud services increases additively, which is much more scalable than a one-to-one model.

In addition to introducing a mechanism for scalable identity management, Olden identifies scalability issues in clouds regarding audit and compliance, authentication and federated SSO, and authorization and access control. He mentions the increased complexity in authorization in cloud computing that results from the need to provide authorization decision across security domains for a large number of users. To increase scalability of authorization, he suggests a federated model that separates authorization responsibilities into policy management, decision making, and enforcement and distributes these responsibilities across the cloud. However, he provides no detailed approach for implementing such a model.

Rohit Ranchal et. al. [11] address the issue of preservation of privacy in identity management on untrusted cloud hosts. Most cloud providers today are very opaque in their handling of identity information, and of particular importance, personally identifiable information (PII). Identity management and authentication with these cloud providers therefore presents inherent privacy risks, as cloud consumers do not know who has access to their data. Ranchal suggests the use of "active bundles" when passing PII to the cloud providers – an encapsulation of identity and other sensitive data in a virtual machine that can restrict access to the data on an untrusted host based on disclosure policies stored within the bundle.

The approach suggested by Ranchal et. al. allows for authentication on an untrustworthy cloud provider without the need to enclose unencrypted PII. It also forgoes the need for a trusted third-party to evaluate and handle requests for sensitive data. However, their paper does not deal with the privacy risks involved in authorization of access to resources on untrustworthy hosts.

There has also been research conducted in the area of secure, privacy-preserving authorization in cloud computing. Researchers at the University of Waterloo [12] address the issue of protecting privacy of users during authorization on untrusted cloud hosts by proposing a user-centric authorization scheme based on the OAuth standard called AAuth. The approach works by encapsulating files with symmetric-key encryption and adding a header that contains a digital signature secured using attribute-based encryption. To gain access to a file on a cloud server, a user must decrypt the header with an attribute-based encryption token, verify the digital signature, and decrypt the data with a public key provided by the owner of the data.

This approach allows authorization to be decoupled from the cloud provider, since the attribute-based encryption keys and public key are managed and distributed by the owner of the data, a cloud consumer. However, it requires a significant amount of computational overhead to encrypt and decrypt the data for every use, which may be overkill if the data being secured is not of critical importance. Additionally, the user of the data must support the infrastructure to request the keys from the owner of the data and must support multiple methods of decryption, which can be prohibitive in a lightweight, dynamic cloud.

Researchers at Hewlett-Packard labs [13] address the issue of scalability in multi-tenant authorization in cloud computing. Cloud service providers support multiple cloud applications that may belong to different organizations. These multi-tenant applications must have controlled access to resources, which is provided by an authorization system. It is often the case that a single service provider is host to applications that are participants in a federation and should therefore have the ability to access resources belonging to other applications in the federation. Furthermore, the authorization system(s) used to control this access should be scalable as the number of federations and services increase.

To address this issue, the researchers propose the use of a centralized authorization system that provides authorization for all services and resources within a single cloud provider’s system. The authorization system performs the role of the policy decision point (PDP), trust manager, and policy store, and pushes the enforcement of the policies to the individual cloud services accessing the resources. The HP researchers, however, do not provide any details addressing the issue of privacy, and do not address authorization across multiple cloud service providers. Additionally, the entire authorization system is tenant on the cloud provider’s system, which may be untrustworthy.

4. My approach

My proposed approach is to provide identity and access management in dynamic clouds while preserving privacy.

4.1. Issues with existing approaches to IAM

To address the problems that exist in applying current approaches to dynamic clouds, it is important to first define the characteristics of dynamic clouds that set them apart from regular clouds. Dynamic clouds are characterized by the following attributes:

- In a dynamic cloud, cloud consumers should have the flexibility to change the service providers they are using and make the new service providers operational for all of their users very quickly. These services should be available for use in a matter of hours to days.
- Cloud consumers should be able to rapidly provision or deprovision cloud services or resources as necessary to handle their computing needs. These services must also be available for use in a matter of anywhere from minutes to days.
- Cloud consumers should be able to rapidly form and destroy clouds as their computing needs change and make the cloud operational for all of their users very quickly. The cloud should be operational in a matter of hours to days.

As adoption of cloud computing increases, the demand for dynamic cloud computing will increase significantly. Cloud consumers would like to establish dynamic clouds for a number of reasons:

- In dynamic clouds, a consumer has more flexibility in selecting a service provider for a particular service or business process.
- Changing service providers does not require significant overhead in terms of reprovisioning, so the cloud consumer is not locked into a particular service provider once it has started using the service.
- Dynamic clouds allow the cloud consumer to utilize cloud service providers as and when needed to deal with short-term spikes in demand without lock-in to providers already being used at the time.

In the traditional paradigm of identity and access management, as discussed in the background section, in order for a cloud to be used, users must be provisioned across the cloud. This entails provisioning user identities to all of the service providers in the cloud and provisioning of access, i.e., authorization policies, to all of the resources used by the service providers and to all of the services provided by the service providers. Further, when use of a service is discontinued, the identities and authorization policies must be deprovisioned from all of these points as well. Any changes made to users or their permissions must be propagated throughout the cloud to ensure that access control is up-to-date.

In a dynamic cloud, in which a consumer may use tens of services and hundreds of resources to support a single business process, and provisioning and deprovisioning must be done for potentially thousands of users, the aforementioned identity and access provisioning process can take a significant amount of time. This is prohibitive for a dynamic cloud, since it hinders the flexibility of a consumer to rapidly replace service providers or scale up the number of services it is using.

Additionally, the issue of privacy is compounded in dynamic clouds. As discussed in the background section, in order to authorize access to resources, a cloud service provider must gather PII from the Identity Provider and use it for authentication and authorization of the user. Most cloud service providers today are not very transparent in their handling of PII, so consumers do not know with whom their information may be shared in the process of authentication and authorization [14]. In a dynamic cloud, as the consumer provisions and deprovisions access to a number of service providers over a period of time, its users' PII will have been made available to a large number of service providers. This presents a significant privacy risk, as the chances of the PII being abused, leaked, or otherwise compromised throughout the identity and access management process increases with each service provider used. This, in turn, makes cloud consumers hesitant to move sensitive information to the cloud or to use a cloud service that requires sensitive information for authentication and/or authorization.

Another problem with the traditional paradigm of identity and access management in a dynamic cloud setting is the difficulty of auditing. In a dynamic cloud, the audit and non-repudiation logs for the execution of a single business process are maintained by multiple service providers in several locations. This fragmentation increases the management overhead for audit and non-repudiation, as the logs must be retrieved from all of the cloud service provider audit systems and correlated for reporting purposes.

4.2. My approach:

My approach attempts to solve the abovementioned problems. The key features of my approach are as follows:

- Centralization of identity and authorization information

  In my approach, identity management and authorization policy management are centralized either with the cloud consumer or with a third party trusted by the cloud consumer. Identity management and authentication are performed, as described in the background section, by an Identity Provider, or IdP. I define the entity that performs authorization policy management and authorization of access to resources and services as the Access Provider, or AP. The AP acts as a central authority for authorization policy management, and remains constant throughout the lifecycle of the dynamic cloud.
- Centralization of the authorization policy decision making for the entire cloud

  In addition to managing authorization policies for the entire cloud, the access provider serves as the policy decision point (PDP) for all of the resources and services in the cloud. In order to authorize access to a resource or service, a CSP sends the request to the AP, which evaluates the access request based on the user's credentials and resource or service identifier and returns the policy decision to the CSP to be enforced.
- Elimination of the need to provision user identity and authorization policies to the various cloud providers during cloud formation or addition of a CSP

  By centralizing the identity and authorization policy management, my approach eliminates the need to provision user identities and authorization policies to the CSPs. Upon formation of a cloud, the CSPs communicate all resource and service information to the AP. All provisioning to resources and services is then performed by the cloud consumer at the AP. As the CSPs no longer need to make authorization policy decisions, they do not require persistent identity information to provide access to their services. As a result, the cloud consumer does not need to communicate identity information or authorization policy information to the CSPs.

  A consequence of the elimination of the need to provision identities and authorization policies is the elimination of the need for deprovisioning when a service is no longer used, the cloud is destroyed, or a user is retired from the consumer's system.
- Use of transient pseudonyms for identification

  In my approach, since the CSPs do not require persistent identity information to control access to their services and resources, the IdP can use transient pseudonyms to identify a user throughout a session. The CSP remains ignorant of the actual identity of the user, and simply uses the transient pseudonym to forward authentication and authorization requests to the IdP and AP.

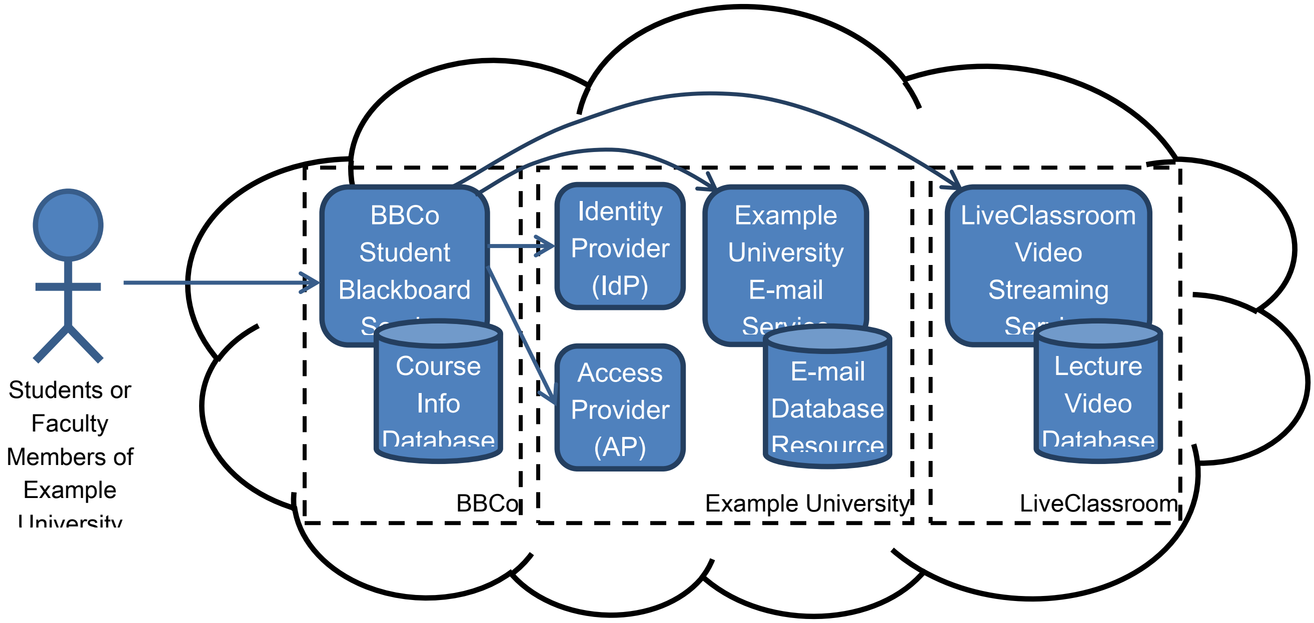

Figure 10: Student blackboard service scenario under my approach

In the university blackboard service scenario provided in the background section, Example University is the cloud consumer. It uses cloud services and resources provided by CSPs BBCo and LiveClassroom. Example University also acts as its own IdP and AP, providing BBCo and LiveClassroom authentication capabilities for its students and granting access to the blackboard and lecture video stream resources. In addition, Example University acts as a CSP, providing access to an e-mail resource that is accessed by BBCo's student blackboard service. The users of the cloud formed by these three parties are the students of Example University.

I explain my approach by detailing the key actions that are performed when utilizing dynamic clouds.

4.2.1. Cloud Formation

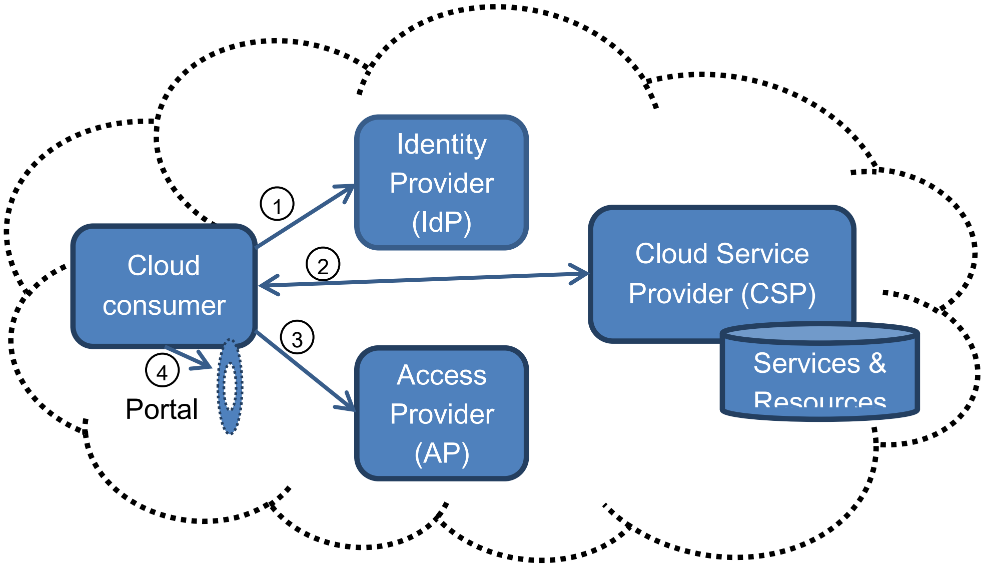

Figure 11: Cloud formation

1. The cloud consumer provisions all of its users to the cloud by communicating user identity information to its IdP.
2. The cloud consumer establishes business agreements with the CSPs for the use of their services and resources. This entails communicating the URLs for the consumer's IdP and AP to the CSPs to provide a location to which the CSPs should send authentication and authorization requests. Furthermore, the cloud consumer establishes with the CSP a unique identifier to use to identify each service and resource provided by the CSP.
3. The cloud consumer provisions access to the cloud services and resources at the AP. To do this, the consumer sends the AP its authorization policies, and uses the authorization policies, user identity information provided by the IdP, and service and resource IDs established with the CSPs to provision access for its users.
4. The cloud consumer establishes a portal through which its users will access the cloud services. Upon access of the cloud services by a user, the portal will provide the cloud consumer's identity (not to be confused with the user's individual identity) to the CSP, which will indicate to the CSP to use the IdP and AP provided by the consumer in step 2 for all authentication and authorization requests.

For example, in the blackboard service scenario, when Example University enters an agreement with BBCo and LiveClassroom to provide its students with a student blackboard service, it initiates the formation of a cloud. As Example University is its own IdP, it does not need to perform step 1. It first establishes an agreement with BBCo and LiveClassroom to allow its students to use the blackboard and lecture video services, and then sends both CSPs the URLs of its IdP and AP. As part of establishing an agreement with BBCo and LiveClassroom, Example University also establishes the identifiers it will use for the services and resource managed by the CSPs. Then, Example University provisions access to the services and resources at its internal AP based on the class enrollment of its students. Finally, Example University sets up a portal site for access of the blackboard service that, when used by its students, will inform BBCo to use Example University as the AP and IdP.

There are many advantages to this approach. First, the cloud provider needs only to provision identity and access centrally at the IdP and AP instead of provisioning at every CSP. This approach is beneficial for the cloud consumer, as it requires less time to commission and provision all services and resources, as identity and authorization policy information only needs to be transmitted once to the IdP and AP. Reduced cloud creation time is essential for the use of dynamic clouds. This approach is also beneficial for the CSPs, as they do not need to deal with large amounts of identity or authorization policy information or with the systems needed to manage such information. The CSPs can reclaim the resources otherwise needed for identity and access management, which reduces the cost of providing cloud services.

In addition, this approach preserves the cloud consumer’s privacy, as the CSP never needs to be given the full identity information of the user. Additionally, PII is only known by the IdP and AP, both of which are trusted parties. As a result, privacy risks associated with the transmission and use of PII for authentication and authorization are significantly reduced.

4.2.2. Cloud Destruction

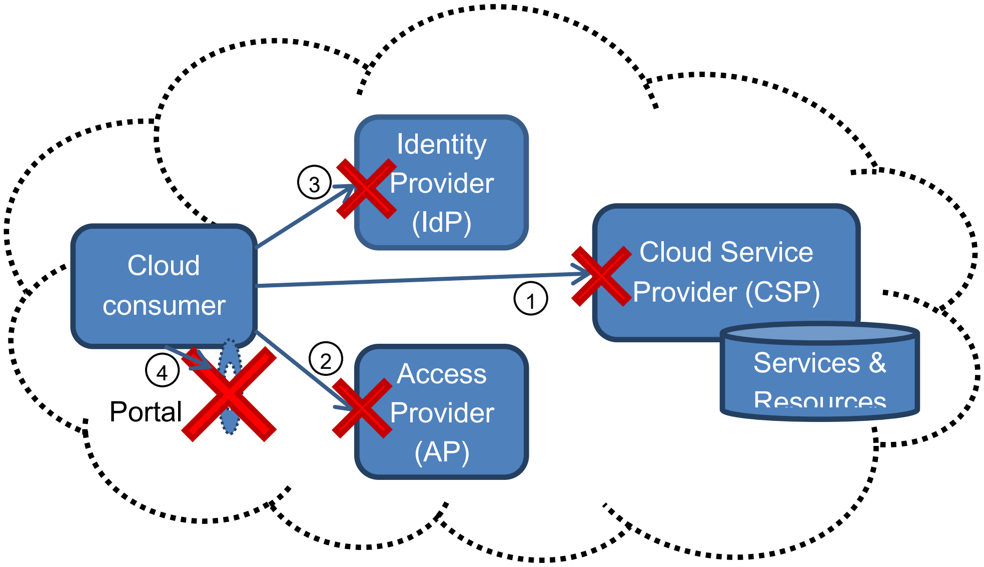

Figure 12: Cloud destruction

1. The cloud consumer decommissions the services and resources at the CSPs it was using and terminates its business agreements.
2. The cloud consumer deprovisions all authorization policies and access rights at the AP for all of the cloud services and resources previously used in the cloud.
3. The cloud consumer deprovisions all identity information for the users of the cloud service at the IdP.
4. As necessary, the cloud consumer retires the portals to the cloud services and reclaims resources used to provide the portals.

In our university example, cloud destruction might take place if Example University migrates to a new blackboard service. In such an event, Example University would first decommission its use of the blackboard and video streaming services at BBCo and LiveClassroom, respectively. It would then deprovision any associated access rights at its internal AP and take down its portal to the blackboard service to free up the resources used by the infrastructure used to support the service. If the university were instead shutting down, it would also deprovision all identity information of the users from its internal IdP to complete destruction of the cloud.

Since the deprovisioning takes place centrally at the IdP and AP instead of at each CSP, the time required to deprovision is substantially lower using this approach than using the traditional approach discussed in the background section. The deprovisioning can even take place out-of-band once the CSPs have been decommissioned, allowing the cloud consumer to rapidly destroy the cloud without the need to wait for deprovisioning to complete.

Additionally, because the CSPs were not given any identity information, there is no threat of lingering PII stored by the CSP posing a threat to user privacy. The cloud consumer does not need to place trust in the thoroughness of the CSP's data deletion methods to ensure privacy of its users' information after destruction of the cloud.

4.2.3. Addition of a User

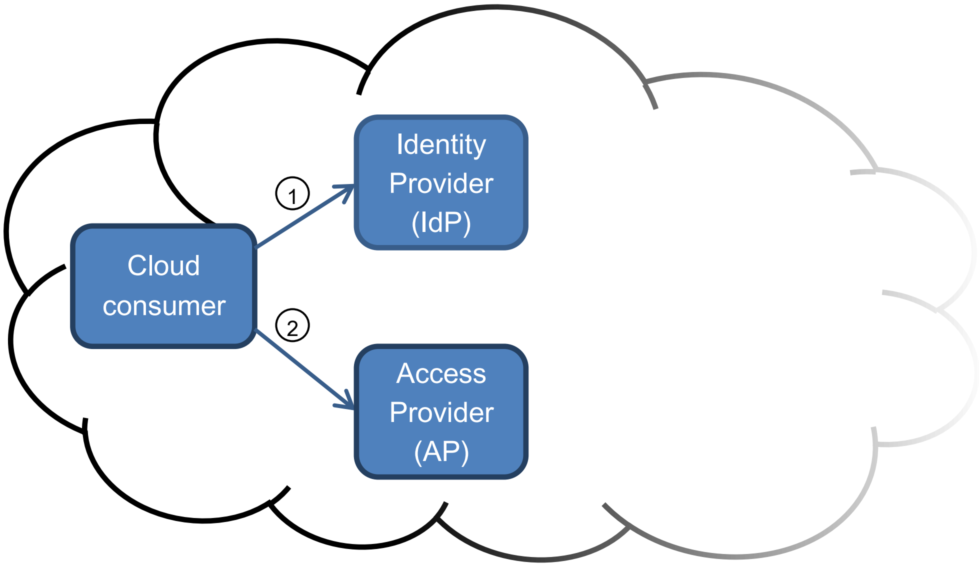

Figure 13: Addition of a user

1. The cloud consumer adds the user's identity information to the IdP's user directory.
2. As necessary, the cloud consumer provisions access to cloud services and resources for the user at the AP.

Such a scenario would be a common occurrence for Example University, where students would be admitted each semester and would need appropriate access to the blackboard service for the classes in which they are enrolled. The university would first create the identity object for the student's access to all university systems and services, and then provision the student's account to be able to access the blackboard service for his classes.

The primary advantage to my approach within the context of addition of a user lies in privacy. Since identity and access management are performed centrally, the entire process is invisible to the cloud provider – the cloud provider is unaware that any changes have been made to the users of the system. This protects the cloud consumer's privacy, as the CSP cannot attempt to glean information about the cloud consumer, such as reorganization or increase in hiring, through the changes in its users. Another advantage of my approach is the increase in efficiency of adding users, as a new user only needs to be added and provisioned in one place as opposed to at all of the CSPs.

4.2.4. Addition of a Resource or Service

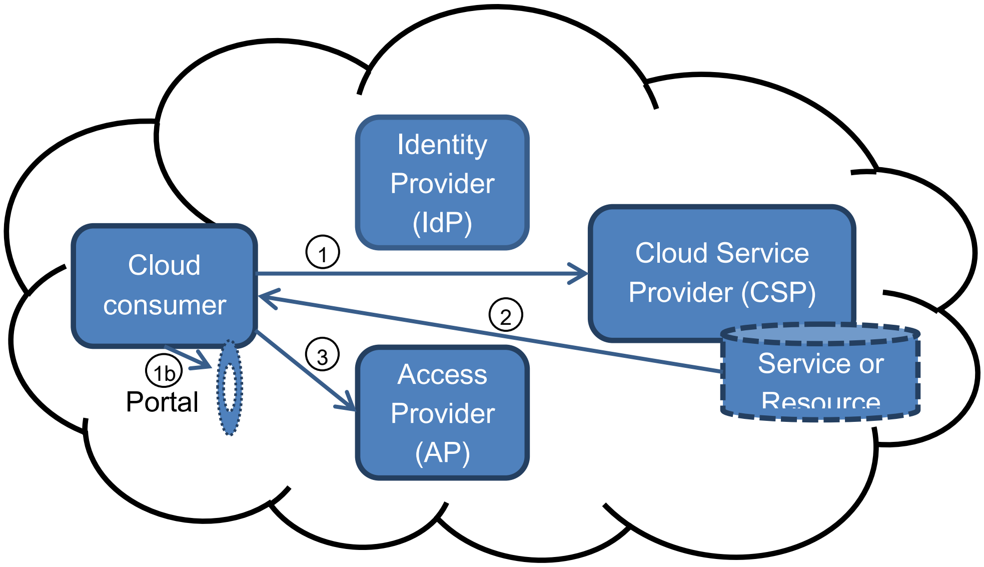

Figure 14: Addition of a resource or service

1. The cloud consumer establishes a business agreement with the CSP for the use of the additional resource or service.
a. If the consumer is using a new CSP, it also communicates the URLs for its IdP and AP to the CSP.
b. If the consumer is using a new CSP, it also creates a new portal or modifies its existing portal for its users to access the resource or service at the new CSP.
2. The CSP communicates its resource or service ID to the cloud consumer.
3. The cloud consumer provisions access to the resource or service for all of its users at its AP.

In our university example, such a scenario would occur if the blackboard service were integrating a new service, such as an Internet whiteboard service, to which Example University wanted its students to have access. Assume the provider of this service is a new CSP called Connect .In this case, Example University would negotiate an agreement with Connect to grant its students access to the whiteboard service and provide Connect with the URLs for its IdP and AP. In order to allow for authorization to the blackboard service, Example University and Connect would establish an ID for the whiteboard service, which Example University would then use to provision access to the service for all of its students.

Since the CSP does not need to receive user identity or authorization policy information for the new resource or service, the addition of the new service can take place extremely rapidly, which is necessary for a dynamic cloud. All provisioning of access takes place at the AP, which already has information about user identities and can therefore provision user access rights simply through transmission of the authorization policies for the new resource or service.

4.2.5. Modification or Retirement of a User

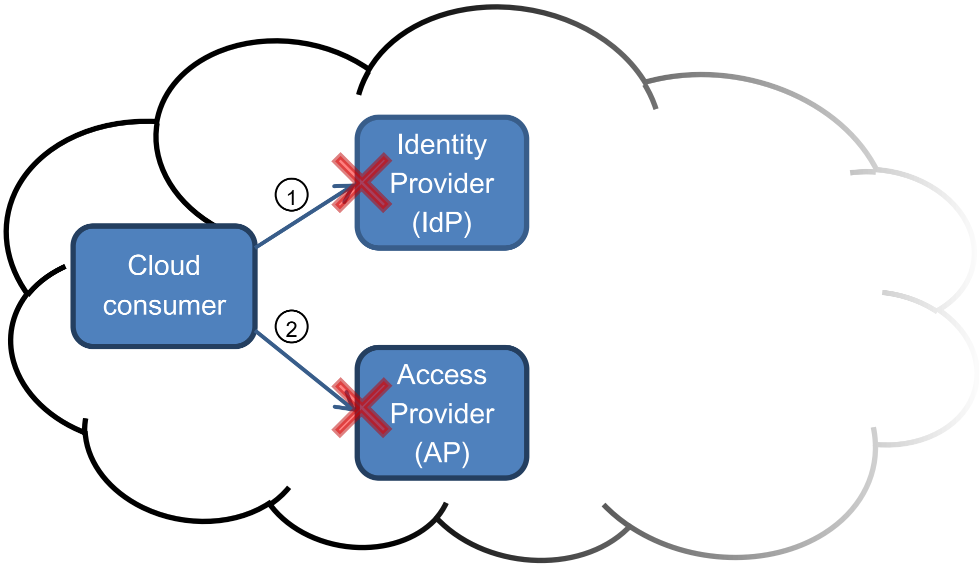

Figure 15: Modification or retirement of a user

1. If necessary, the cloud consumer modifies the user's identity information at the IdP. If the user has been retired, the user's identity information is removed from the IdP.
2. The cloud consumer modifies or deprovisions the user's access rights at the AP.

Such a scenario would again be common in our university example, as every semester, students register for and drop classes and graduate or drop out, thus changing their access rights to the blackboard service. In the situation in which a student changes his class registration, Example University would provision the student access to the blackboard site for the new class and possibly deprovision access to classes taken in the previous semester. In the situation in which a student graduates or retires, Example University would deprovision the user's identity from its IdP and deprovision the user's rights from its AP.

As was the case with the addition of a user, my approach preserves the privacy of the cloud consumer when modifying a user's identity information and access rights by hiding changes to the cloud consumer's users from the CSP. Additionally, as the user's information is managed centrally, it only needs to be changed in two locations instead of at every CSP, which increases the efficiency of modifying users. In a dynamic cloud, users may be changed frequently, so the ease of modification provided by my approach is essential to identity and access management in a dynamic cloud.

4.2.6. Termination of Use of a Resource or Service

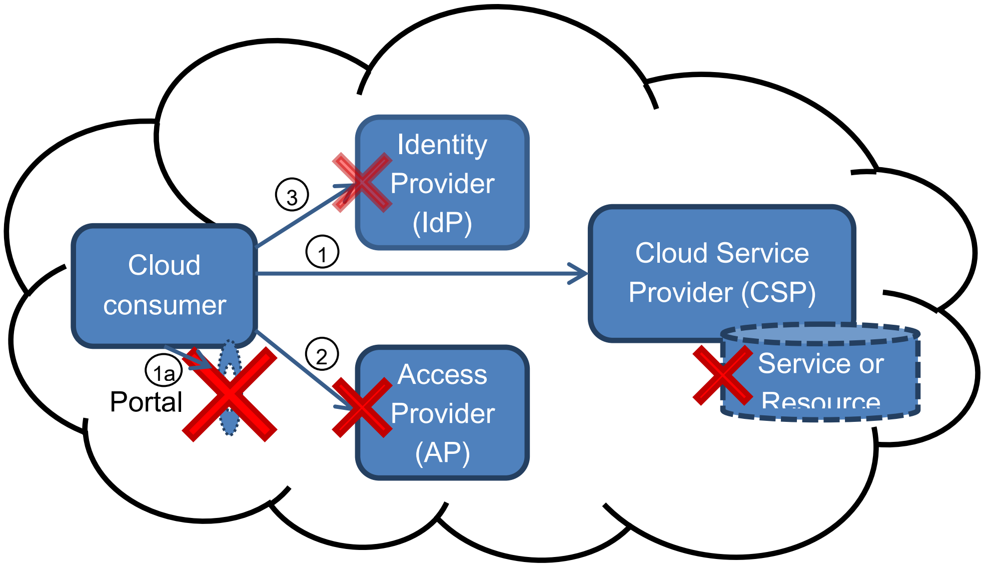

Figure 16: Termination of use of a resource or service

1. The resource or service is decommissioned at the CSP.
a. If necessary, the consumer retires the portal to the cloud service.
2. Access rights for the resource or service are deprovisioned at the AP.
3. If any users need to be retired as a result of decommissioning the resource or service, they are retired at the IdP.

In our university example, if Example University decided to terminate its use of LiveClassroom's lecture video streaming service, it would enter the scenario given above. In such a scenario, Example University would first decommission its use of LiveClassroom's service, and then deprovision all access rights for the video streaming service at its AP. Since the students would still need to use the blackboard service, Example University would not need to retire any of its users.

The advantages offered by my approach during termination of a service or resource are the same as those offered when destroying the cloud. Since the deprovisioning takes place centrally, it can be done extremely rapidly. The deprovisioning can even take place out-of-band once the resource or service has been decommissioned, allowing the cloud consumer to rapidly terminate use of a resource or service without the need to wait for deprovisioning to complete. Also, there is no threat to the cloud consumer's privacy resulting from lingering identity information being stored by the CSP after termination.

4.2.7. Authentication

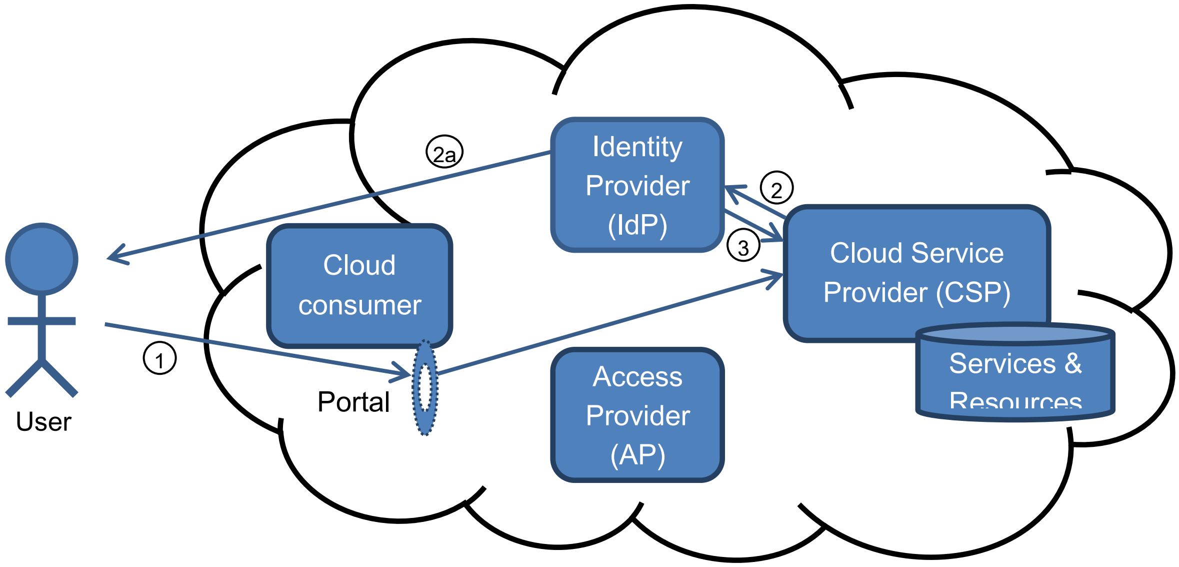

Figure 17: Authentication

1. The user accesses a service through the cloud consumer's portal to the CSP. The portal sends the CSP the ID of the cloud consumer (not the user at first) to indicate to the CSP which IdP and AP to use.
2. The CSP sends a request to the IdP to get the user's transient pseudonym in order to authenticate the user. If the user has not yet authenticated at the IdP, the IdP performs the authentication as follows:

a. The IdP requests authentication credentials from the user.
b. The IdP verifies the validity of the credentials provided, and if they are validated, generates a transient pseudonym to identify the user for the duration of the session.

3. The IdP returns the pseudonym to the CSP, which the CSP uses to authenticate the user.

For example, in our university scenario, to access the blackboard service, a student would visit Example University's portal for the service, which would direct the user to BBCo's site. BBCo would then redirect the student to Example University's IdP page to get the student's pseudonym. If the user were not logged in at the time, the IdP would redirect the user to a login page, where the user would authenticate using his university username and password. The IdP would then generate a transient pseudonym for the user and redirect the user to BBCo's service page, pushing the user's transient pseudonym.

The primary advantage of my approach during authentication is the preservation of the user's privacy. Throughout the entire session, the CSP is never given any permanent identity information that it could then leak to other parties or misuse. It uses a transient pseudonym only known by it, the IdP, and the AP to identify the user for the duration of the session, which even if leaked, could not be used to personally identify the user.

4.2.8. Authorization

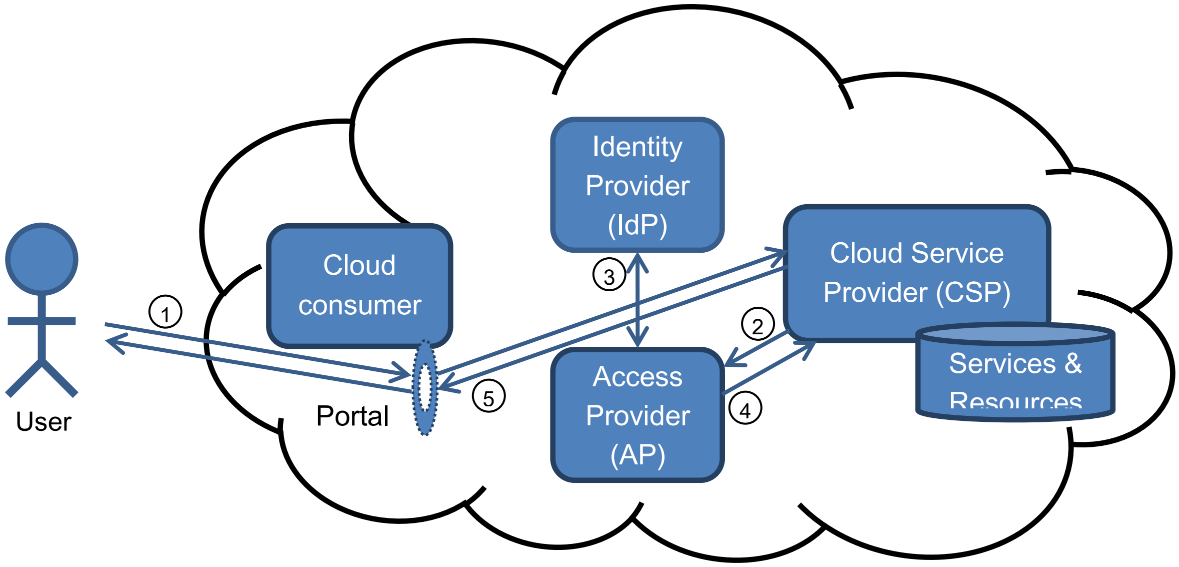

Figure 18: Authorization

1. A user requests access to a resource or service.
2. The CSP sends a request to the AP with the resource or service ID and the user's transient pseudonym asking whether it should authorize the request.
3. The AP determines the user's ID:

a. If the AP is the same party as the IdP, it is already aware of the user's identity from the transient pseudonym.

b. If the AP is a separate entity from the IdP, it requests the identity attributes it needs for authorization from the IdP.

4. The AP, acting as the policy decision point (PDP), evaluates the authorization request and returns its response to the CSP.
5. The CSP, acting as the resource manager and policy enforcement point (PEP), enforces the decision made by the AP and returns the result of the resource request to the user.

For example, in our university example, after the student authenticates to BBCo's service, the blackboard service would need to retrieve the discussion board information from its own database and the lecture video from LiveClassroom's service, and might potentially need to use Example University's e-mail service to send e-mails on behalf of the student. To do so, the blackboard service would send a request to Example University's AP with the student's transient pseudonym, the respective resource or service ID for the resource or service being accessed by the blackboard service, and the information being requested. The AP would then look up the student's identity from Example University's IdP, determine if the student should have access to the resource or service being requested, and return its decision to the blackboard service. The blackboard would then process the results of the authorization request and return the student its personalized blackboard page.

The primary advantage offered by my approach during authorization is again preservation of the cloud consumer's privacy. The CSP is not required to know the cloud consumer's authorization policies or the PII required to enforce these policies. All PII that is needed to perform authorization checks is passed between the IdP and AP, which are both trusted parties, so the threat to privacy is drastically reduced.

Another advantage offered by my approach is greater flexibility in choosing authorization policies. The AP is free to implement authorization policy decision making as it chooses, so the cloud consumer is not reliant on methods supported by the CSP. As a consequence, the CSP is also freed from needing to support the widest range of authorization checks used by its most demanding customer; in fact, it can support none!

4.2.9. Audit logging

1. When users are created, retired, or modified, or when an authentication request is received by the IdP, the IdP makes a log entry in the cloud consumer's central audit and non-repudiation logs. In order to correlate with CSP logs as necessary for reporting and investigation, the IdP also logs the transient pseudonyms generated for each user.
2. When access is provisioned or deprovisioned for a user, services are provisioned or deprovisioned, or authorization requests are received by the AP, the AP makes a log entry in the cloud consumer's central audit and non-repudiation logs.
3. Upon any interaction with the cloud consumer, the CSPs also log the details of the interaction using the transient pseudonyms to identify the users.

In our example university scenario, audit logging would take place every time a student accesses the blackboard service or authenticates at the IdP, every time the blackboard service makes a request for access to the LiveClassroom service, e-mail service, or its own discussion board resource, every time users are provisioned, deprovisioned, added, or deleted, and every time use of resources or services is started or terminated. All of these would be logged at Example University's IdP and AP, but BBCo and LiveClassroom would also keep their own logs of these activities for their own audit and non-repudiation purposes.

In my approach, all requests for access pass through the IdP and AP, and are hence logged there. This makes reporting much simpler, as the cloud consumer can simply query the central audit and non-repudiation log repository to gain log information for all CSPs instead of needing to retrieve and correlate log information from all CSPs.

Additionally, since only transient pseudonyms are known by the CSP, the CSP cannot use log information to mine information about a consumer's use of resources and services, which protects the privacy of the cloud consumer's users.

4.3. Potential Improvements

A potential improvement to the approach given above is to employ precomputation of authorization responses. Since the AP knows what resources and auxiliary services are associated with a particular service, when a CSP requests authorization for the service, the AP can compute a priori the authorization responses for the various resources and services used by the service. The AP could also potentially push such information to the CSP in anticipation of its subsequent requests for authorization. This would only be performed for those responses the AP could precompute at the time of the first request. Such an approach would decrease the latency in requesting access to a service that utilizes a wide range of resources and services, as the policy decision making for the auxiliary requests would not need to take place in-band and the number of round trips required to gain authorization would decrease.

5. Conclusions and Future Work

Due to their numerous business benefits, dynamic clouds are critical to the examination of cloud computing. Furthermore, in order to facilitate the movement of organizations to dynamic clouds, ensuring the privacy of their identification data is of utmost importance. In this paper, I presented an approach to identity and access management in dynamic cloud computing that was scalable and preserved privacy. My approach centralized all identity and authorization information, management, and decision making at an identity provider and access provider, parties trusted by the cloud consumer, in order to increase scalability and dynamicity and decrease the risk of loss of privacy. It also eliminated the need to provision user identity and authorization policy information at the cloud service providers by redirecting the cloud service provider to the cloud consumer's identity and access providers, thereby substantially reducing privacy risks associated with the provisioning of such information and decreasing time to provision. Lastly, my approach used transient pseudonyms for the identification of users throughout a session in order to eliminate the need to store identity information outside the identity provider. I illustrated my approach by detailing scenarios related to the lifecycle of dynamic clouds, which included the formation of the cloud, the destruction of the cloud, the addition of a user, the addition of a resource or service, the modification or deletion of a user, the termination of use of a resource or service, authentication, authorization, and audit logging.

The approach I presented offers the benefits of increased scalability, preservation of privacy, decreased time to provision and deprovision, and ease of auditing. Because identity and access are managed centrally instead of at each cloud service provider, addition of new services and users can be done with little provisioning, which increases scalability and dynamicity. Since the cloud service providers do not need to be provided any identity or authorization policy information, there is no threat of PII being compromised due to maliciousness or mishandling of the data by the CSP. By centralizing the identity and authorization data, provisioning is done in one place, so clouds can be created and destroyed rapidly. Lastly, audit logging is much more manageable, since all audit logging occurs at the identity and access providers, eliminating the need to retrieve and correlate audit logs from each cloud service provider.

Because of the abovementioned benefits, my approach enables users to participate in dynamic clouds much more easily, thereby facilitating the adoption and use of dynamic clouds.

As an extension to this work, potential schemas for detailing the structure of the dynamic cloud and effective methods for communicating this structure to the parties in the cloud as and when the structure changes could be explored. There may also be situations in which resources used by the cloud service are private and completely managed by the cloud service provider instead of the cloud consumer. Extensions of the approach I have given to address such scenarios can be further explored. Additionally, the precomputation techniques mentioned at the end of my approach for improving performance can be implemented and tested for different cloud types and industry-specific scenarios in order to determine their effectiveness and suitability.

6. References

[1] National Institute of Standards and Technology, *NIST Definition of Cloud Computing*, Sept 2011.

[2] Armbrust, M. et. al., (2009), “Above the clouds: A Berkeley view of Cloud Computing”, *UC Berkeley EECS*, Feb 2010.

[3] Ramgovind, S.; Eloff, M.M.; Smith, E., "The management of security in Cloud computing," *Information Security for South Africa, 2010*, vol., no., pp.1-7, 2-4 Aug. 2010.

[4] IBM Corporation, *Enterprise Security Architecture Using IBM Tivoli Security Solutions*, Aug 2007.

[5] “Federated identity management.” Internet: http://en.wikipedia.org/wiki/Federated_identity_management, [Apr. 10, 2012].

[6] Cloud Security Alliance, *Security Guidance for Critical Areas of Focus in Cloud Computing V2.1*, 2009.

[7] Celesti, A.; Tusa, F.; Villari, M.; Puliafito, A., "Security and Cloud Computing: InterCloud Identity Management Infrastructure," *Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE), 2010 19th IEEE International Workshop on*, vol., no., pp.263-265, 28-30 June 2010.

[8] OASIS Security Services TC, *Security Assertion Markup Language (SAML) V2.0 Technical Overview*, 25 Mar 2008.

[9] He Yuan Huang; Bin Wang; Xiao Xi Liu; Jing Min Xu, "Identity Federation Broker for Service Cloud," *Service Sciences (ICSS), 2010 International Conference on*, vol., no., pp.115-120, 13-14 May 2010.

[10] Olden, E., "Architecting a Cloud-Scale Identity Fabric," *Computer*, vol.44, no.3, pp.52-59, March 2011.

[11] Ranchal, R.; Bhargava, B.; Othmane, L.B.; Lilien, L.; Anya Kim; Myong Kang; Linderman, M.; , "Protection of Identity Information in Cloud Computing without Trusted Third Party," *Reliable Distributed Systems, 2010 29th IEEE Symposium on*, vol., no., pp.368-372, Oct. 31 2010-Nov. 3 2010.

[12] Tassanaviboon, A.; Gong, G., “OAuth and ABE based authorization in semi-trusted cloud computing: AAuth,” *Data intensive computing in the clouds, 2011 Second international workshop on*, vol., no., pp.41-50, 2011.

[13] Calero, J.M.A.; Edwards, N.; Kirschnick, J.; Wilcock, L.; Wray, M., "Toward a Multi-Tenancy Authorization System for Cloud Services," *Security & Privacy, IEEE*, vol.8, no.6, pp.48-55, Nov.-Dec. 2010.

[14] Angin, P.; Bhargava, B.; Ranchal, R.; Singh, N.; Linderman, M.; Ben Othmane, L.; Lilien, L., "An Entity-Centric Approach for Privacy and Identity Management in Cloud Computing," *Reliable Distributed Systems, 2010 29th IEEE Symposium on*, vol., no., pp.177-183, Oct. 31 2010-Nov. 3 2010.

[15] Shigang Chen, Meongchul Song, Sartaj Sahni, Two Techniques for Fast Computation of Constrained Shortest Paths, IEEE/ACM Transactions on Networking, vol. 16, no. 1, pp. 105-115, February 2008.

[16] King-Shan Lui, Klara Nahrstedt, Shigang Chen, Hierarchical QoS Routing in Delay-Bandwidth Sensitive Networks, in Proc. of IEEE Conference on Local Area Networks (LCN’2000), pp. 579-588, Tampa, FL, November 2000.

[17] Shigang Chen, Yi Deng, Attie Paul, Wei Sun, Optimal Deadlock Detection in Distributed Systems Based on Locally Constructed Wait-for Graphs, in Proc. of 16th IEEE International Conference on Distributed Computing Systems (ICDCS’96), Hong Kong, May 1996.