NIST SP 500-XXXX

The NIST Cloud Federation Reference Architecture

Craig A. Lee
Robert B. Bohn
Martial Michel

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.XXXX

National Institute of Standards and Technology
U.S. Department of Commerce

NIST SP 500-XXXX

The NIST Cloud Federation Reference Architecture

Craig A. Lee
Information Systems and Cyber Division
The Aerospace Corporation

Robert B. Bohn
NIST Cloud Computing Program
Advanced Networking Technologies Division
Information Technology Laboratory
NIST

Martial Michel
Data Machines Corporation

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.XXXX

July 2019

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Walter Copan, NIST Director and Undersecretary of Commerce for Standards and Technology

National Institute of Standards and Technology (NIST) Special Publication 500-XXX
nn pages (July 2019)

Comments on this publication may be submitted to:

National Institute of Standards and Technology Attn: Advanced Networking Division, Information Technology Laboratory 100 Bureau Drive (Mail Stop 8920) Gaithersburg, MD 20899-8920

All comments are subject to release under the Freedom of Information Act (FOIA).

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at NIST promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology (IT). ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems. This document reports on ITL's research, guidance, and outreach efforts in IT and its collaborative activities with industry, government, and academic organizations.

Abstract

This document presents the NIST Federated Cloud Reference Architecture model. This actor/role based model used the guiding principles of the NIST Cloud Computing Reference Architecture to develop an 11 component model which are described individually and how they function as an ensemble. There are many possible deployments and governance options which lend themselves to create a suite of federation options from simple to complex. The basics of cloud federation can be described through the interactions of the actors in a layered three planes representation of trust, security, and resource sharing and usage. A discussion on possible future standards and use cases are also described in great detail.

Key words

Federation; Identity; Resources; Authentication, Authorization, Cloud Computing.

Table of Contents

List of Tables

List of Figures

Acknowledgements

This document reflects the contributions and discussions by the membership of the NIST Federated Cloud Public Working Group (FC-PWG), co-chaired by Robert Bohn (NIST ITL), and Craig A. Lee (The Aerospace Corporation).

NIST would like to acknowledge the specific contributions[1] to this document made by the following FC-PWG members:

Victor Danilchenko *Schneider Electronics*	**Alexander Rebo** *US Department of Treasury*	**Robert Bohn** *NIST*
Craig A. Lee *The Aerospace Corporation*	**Bryan Ward** *UniSys*	**John Messina** *NIST*
Martial Michel *Data Machines Corporation*	**Khalil Yazdi** *Yazdi Associates*	

[1] "Contributors" are members of the NIST Federated Cloud Public Working Group who dedicated great effort to prepare and gave substantial time on a regular basis to research and development in support of this document. All opinions are the authors' own.

Executive Summary

The adoption of cloud computing into the US Government (USG) and its implementation depend upon a variety of technical and non-technical factors. NIST has developed and described fundamental starting points such as a definition of cloud computing and a cloud computing reference architecture. NIST has also produced a roadmap for "USG Cloud Computing Standards and Technology" Roadmap (NIST SP 500-293, 2014), which discusses and highlights a set of high priority requirements for the adoption of cloud computing. Requirement 5 of this document states a need for "*Frameworks to support seamless implementation of federated community cloud environments".* Industry and the USG need to develop frameworks to support seamless implementation of federated community cloud environments.

In Community Cloud deployments, infrastructure is shared by organizations that have common interests (e.g. mission, security requirements, and policy). In the case where a Community Cloud deployment model is not implemented in one (private cloud or public) environment, which accommodates the entire community of interest, there is a need to clearly define and implement mechanisms to support the governance and processes, which enable federation and interoperability between different Cloud Service Provider (CSP) environments, in order to form a general or mission-specific, federated Community Cloud.

We also wish to emphasize that cloud computing -- and what CSPs provide – is becoming far broader than just basic infrastructure, i.e., compute, storage or networking. These broader capabilities include databases on demand, microservices such as Functions-as-a-Service, workflow managers, edge caches, and a host of other capabilities that reside higher up in the system stack. Such capabilities from different providers could also be shared across a set of remote users. This could also be done for arbitrary, application-level services at the Software-as-a-Service level. Harmonization of access, capabilities, and resources are important when working with heterogenous clouds; a multi-cloud approach is possible when common exchange mechanisms are available for services.

The importance of the Community cloud was clearly identified in the NIST-hosted Reference Architecture public working group. The architecture anticipated potential multi-cloud configurations such as Hybrid cloud or those topologies involving a Cloud Broker. It did not address the generalized notion of a federated Cloud Community. USG agencies, the National Security Telecommunications Advisory Committee, and the Open Grid Forum are examples of potential cloud adopters which have identified this matter as a high priority. The concept has been developed in earlier IT models such as the "grid," where public and private sector research labs and universities make up a community of High-Performance Computing scientists. Federation techniques have been applied across grids, data centers, and countries to create a "multi-grid community logical grid."

A fundamental reference point, based on the NIST definition of Cloud Computing, is needed to describe an overall framework that can be used government-wide. This document presents the NIST Cloud Federation Computing Reference Architecture (CFRA) and Taxonomy that will accurately communicate the components and offerings of cloud computing. The principles adhered to in creating this CFRA were to:

1) Use the original NIST Cloud Computing Reference Architecture as a guide,

2) Develop a vendor-neutral architecture that is consistent with that reference architecture,
3) Develop a federation reference architecture that does not stifle innovation by defining a prescribed technical solution, and
4) Identify the unique features of this reference architecture.

The resulting reference architecture and vocabulary for cloud computing was developed as an Actor/Role-based model that lays out the central elements of cloud computing for Federal CIOs, Procurement Officials, and IT Program Managers. The cloudscape is open and diversified, and the accompanying taxonomy provides a means to describe it in an unambiguous manner.

The Architectural Components of the CFRA describe the important aspects of service deployment and service orchestration. The overall service management of the cloud is acknowledged as an important element in the scheme of the architecture. Business Support mechanisms are in place to recognize customer management issues like contracts, accounting, and pricing, and are vital to cloud computing. A discussion on Provisioning and Configuration points out the requirements for cloud systems to be available as needed, to be metered, and to have proper SLA management in place. Portability and Interoperability issues for data, systems, and services are crucial factors facing consumers in adopting the cloud, and are also undertaken here. Consumers need confidence in moving their data and services across multiple cloud environments.

As a major architectural component of the cloud, Security and Privacy concerns need to be addressed, and there needs to be a level of confidence and trust to create an atmosphere of acceptance in the cloud's ability to provide a trustworthy and reliable system. Security responsibilities, security consideration for different cloud service models, and deployment models are also discussed.

1 Introduction

1.1 Background

NIST defines a *Community Cloud* as supporting organizations that have a common set of interests (e.g. mission, security, policy [1]). When that community cloud cannot be implemented in one public or private cloud, "there is a need to clearly define and implement mechanisms to support the governance and processes which enable federation and interoperability between different cloud service provider environments to form a general or mission-specific federated Community Cloud." This is the core of *Requirement 5: Frameworks to Support Federated Community Clouds* in the NIST *US Government Cloud Computing Technology Roadmap, Volume I* [2].

What is federation? In the simplest terms, federation means to enable interaction or collaboration of some sort. Federation is an overloaded term with different meanings to different stakeholders. What does it entail in this context and with regard to the cloud computing model? What is the scope of capabilities it can or must support? Of course, this can mean very different things in different use cases, in different application domains, and at different levels in the system stack. In some situations, federation is used to mean identity federation. This means being able to ingest identity credentials from external identity providers. This can be used to provide single sign-on (SSO) – a very useful capability. SSO allows a single authentication method to access different systems within external identity providers based on mutual trust. We will demonstrate that identity federation (also referred to as Federated Identity Management) is a necessary component that enables the federation of clouds.

In this document, we shall refer to “federation” as synonymous with cloud federation, i.e. getting two or more cloud providers to interact or collaborate [3]. The term multi-cloud has been used when cloud provider capabilities are "integrated" by defining a separate interface layer for each “back-end” provider whereby a single, common interface can be presented to the user [4]. This approach achieves cloud interoperability by using the rich feature set of the cloud capabilities, but integrates them very shallowly, if at all. Another approach is to use a "lowest common denominator" approach. Here, some minimal feature set across all providers is used, e.g. VMs, and the "integrated" infrastructure system is built on top using, for example, Docker, Kubernetes, OpenStack, or various DevOps solutions. This approach provides portability across cloud providers by avoiding use of any of their differentiating capabilities.

Along these lines, the ISO/IEC Cloud Computing Reference Architecture [5] defines the concept of an inter-cloud with inter-cloud providers. Here, different cloud service providers peer to one another to offer cloud services to a larger set of cloud service consumers. This peering is done through federation, intermediation, aggregation, and arbitrage of existing cloud provider services. While these are important concepts, this ISO/IEC document does not go into any further detail about what federation or these other activities entail and require. We investigate those issues here.

In the case of a Community Cloud deployed by a single Cloud Provider, the cloud PaaS layer can be used by developers to create applications. If developers establish common technical policies and credentials within that Community Cloud, they can use tools and management systems from different vendors, and connect applications to others using common PaaS facilities. However, in a federated multi-cloud environment with diverse cloud implementations and policies, the modules may need manual intervention to function together. Technical policies, credentials, namespaces, and trust infrastructure must be harmonized to support a Community Cloud that spans multiple service providers' physical environments.

The NIST Cloud Federation Reference Architecture (CFRA) is presented in ten parts: a complete overview of the actors and their roles, and the necessary architectural components for managing and providing cloud services such as service deployment, service orchestration, cloud service management, security and privacy. The Taxonomy is presented in its own section and appendices are dedicated to terms and definitions and examples of cloud services.

The CFRA describes six actors with their roles & responsibilities using the associated Federated Cloud Computing Taxonomy and operating under specific administrative and regional domains. The six major participating actors are the Federated Cloud Consumer, Federated Cloud Provider, Federated Cloud Operator, Federated Cloud Broker, Federated Cloud Auditor, and Federated Cloud Carrier. These core individuals have key roles in the landscape of federated cloud computing.

Although, the NIST CCRA (NIST SP 500-292) and this current CFRA share some certain actors & functionalities, there are some significant differences. Principle differences lie between the roles of the Cloud Broker in the CCRA and the CFRA. There are new actors and responsibilities, which appear in the CFRA, that have no counterparts, such as the Federated Cloud Operator and a subservient entity, the Federated Cloud Administrator. In addition, the Cloud Federation Manager is an indispensable piece of the federation machinery where the specific federation instance is instantiated. This new architecture depicts the Administrative Domains and Regulatory Environments under which the federated cloud operates.

1.2 Report Production

The NIST federated cloud computing reference architecture project team has surveyed and completed an initial analysis of federated models. Based on available information, the project team developed a strawman model of architectural concepts. This effort has leveraged the collaborative process from the NIST Federated Cloud Computing Reference Architecture working group that was active between August 2017 and June 2019. This process involved from the industry, academic, and government agencies. The working group has iteratively revised the reference model by incorporating comments and feedback. This document reports the first edition of the NIST Federated Cloud Reference Architecture and Taxonomy.

1.3 Report Structure

Following the introductory material presented in Section 1, the remainder of this document is organized as follows:

- Section 2 introduces the essence of federation and a three-plane model to describe the basic functionality of a federation.
- Section 3 introduces the NIST Federated Cloud Architecture and describes its components.

- Section 4 presents a discussion of federation governance, which describes how the pieces of the architecture of a federation operates, works together, and interacts and the essential characteristics of a federation.
- Section 5 presents a systematic look at federation deployment models, i.e. implementation approaches and trade-offs and how they affect simplicity, generality, performance, governance, trust relationships, and scalability.
- Section 6 describes the requirements and options of deployment governance which carry a large number of trust implications.
- Section 7 describes the large number of possible federation deployment models and their increasing scalability and complexity.
- Section 8 gives a discussion on relevant existing tools and standards on federated cloud.
- Section 9 describes areas of possible federation-specific standards that could be derived from this work.
- Section 10 concludes the discussion and makes some final observations.

2. The Essence of Federation

In its most general sense, federation could support the sharing of arbitrary resources, from arbitrary application domains with arbitrary consumer groups across multiple administrative domains. These could be data-sharing services, e.g. international "big science" collaborations, disaster response, supply chain management, or medical information systems. Any type of organizational collaboration could be facilitated by a secure method to selectively share data with specific partners. This could be said for sharing any type of functional or analytical service under a set of resource-sharing rules and conditions. This was the goal of the *Virtual Organization (VO*) concept developed in the grid computing community [6].

Given this wide applicability and fundamental impact of federation, it is critical to understand the essence of what federation entails. This is described in Figure 1 and Figure 2. Figure 1 illustrates how authentication and authorization are done in modern systems. Here, a *User* is issued some form of *identity credentials* by an *Identity Provider (IdP)* (1,2). When the User requests service from some Service Provider (SP), the User must also present their credentials (3). Before responding, the Service Provider will *validate* the User's credentials with the IdP (4). After a response from the IdP (5), the SP will make an *access decision* to either *honor* or *decline* the service request (6) based on the validity of the User's identity credential, and the *roles* or *attributes* associated with it.

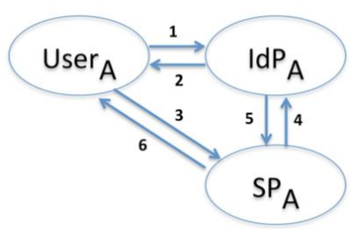


Figure 1. Ordinary authentication and authorization.

To enable different organizations to collaborate, we must enable this same kind of process among the collaborating organizations. This fundamental requirement is illustrated in Figure 2.

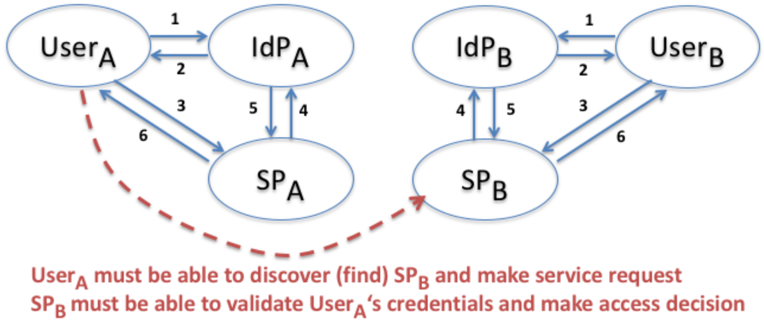


Figure 2. Federated authentication and authorization.

Hence, a federation is essentially an environment wherein:

1. Users in Organization A can discover and invoke services in Organization B, and
2. Service Providers in Organization B can validate credentials from Organization A and make the proper access decisions.

With this understanding of federation "in a nutshell", we can identify the necessary fundamental federation design principles.

2.1 Fundamental Federation Design Principles

The fundamental requirements of a federated environment can be expressed in the following fundamental design principles:

- A federation is a security and collaboration context that is not "owned" by any one user or organization.
- Since only specific users, sites, and organizations may wish to collaborate for specific common goals, it can be said these participating entities have *membership* in a specific federation.
- Participating members can jointly agree upon the common *goals* and *governance* of the federation.
- That joint governance is expressed by the *policies* governing the *roles* and *responsibilities* of membership, resource discovery, and resource access.
- There are *roles* and *attributes* on which these policies are based that are well known.
- There is an *administration role* whereby federation membership, resource discovery, and resource access can be granted or revoked according to governance policy.
- Sites can participate in a federation by selectively making some of their resources discoverable and accessible by other federation members.
- While the purpose of a federation is to collaborate and share resources, resource owners retain ultimate control over their own resources. A resource owner can unilaterally change their discovery and access policies. However, a resource owner should have good reason to since such unilateral policy changes could adversely affect the other federation members.

With these design principles, it is important to realize that a federation can be considered a *Virtual Administrative Domain*. A federation is an administrative and security domain wherein users and resources are consistently managed, like any other administrative domain. In a federation, however, that domain is virtual – it is logically comprised of multiple parts of different sites or organizations.

We can make further observations at this point that will become clear as the reference architecture is developed. It can be colloquially said that the federations require *identity federation* on the front end, and *resource federation* on the back end. Federation Members and Service Providers must have a common understanding of the identity credentials being issued by IdPs along with their attribute semantics. Resource owners (service providers) may wish to control or limit who in a federation can discover and use their resources through policies based on the identity credentials and attributes that are meaningful within a given federation. This implies that trust relationships must be established among all federation participants.

Different federation instances (or simply federations) could be created for different collaboration purposes and goals, even among the same participants. Collaborations can be managed at any level in the system stack. That is to say, we could manage federations of cloud infrastructure services, or we could manage federations of arbitrary business functions.

The notion of invoking services between two organizations and administrative domains is directly relevant to the cloud deployment models defined in NIST's Definition of Cloud Computing [1]. The *hybrid cloud* and *community cloud* deployment models could be considered specific use cases of a more general federation model that enables two or more organizations to collaborate [7]. That is to say, this federation reference architecture will actually clarify what is necessary to support these two use cases that were previously identified as deployment models.

The goal of this document is to first organize all of these principles into a coherent reference architecture. As a conceptual model, all fundamental *federation entities (actors)* will be identified, along with their *functional behaviors* and *interactions*. The necessary *governance* at each stage in the *lifecycle* of a federation instance (or simply federation) will be identified. After establishing this baseline, we will examine *federation deployment models*. Here we will describe how the actors and interactions in the Reference Architecture could be realized using different implementation approaches. These different approaches will have different ramifications with regards to ease of implementation and deployment, fault tolerance, and scalability. Across these different deployment models, we will identify relevant, existing standards that will support standardized federation environments. Just as important, though, we will identify areas of necessary or desirable areas of *federation-specific* standardization that need to be addressed.

2.2 Describing Federation: A Three-Plane Representation

Before introducing the reference architecture in detail, we will present a preview of the concepts. While this will require a number of *forward references* to the terminology used in the reference architecture, this should nonetheless give the reader an intuitive, visual understanding of what the reference architecture is actually enabling. The reader should then be able to better understand the reference architecture as it is developed in the following sections.

As we emphasize throughout this document, the reference architecture identifies fundamental, functional capabilities that could be used with a range of different deployment and governance models. It endeavors to organize the *federation design space*. It identifies how federations can be organized and used, but does not dictate how any of this must be done. That is determined by the requirements of the specific federation instance, as defined by the federation members.

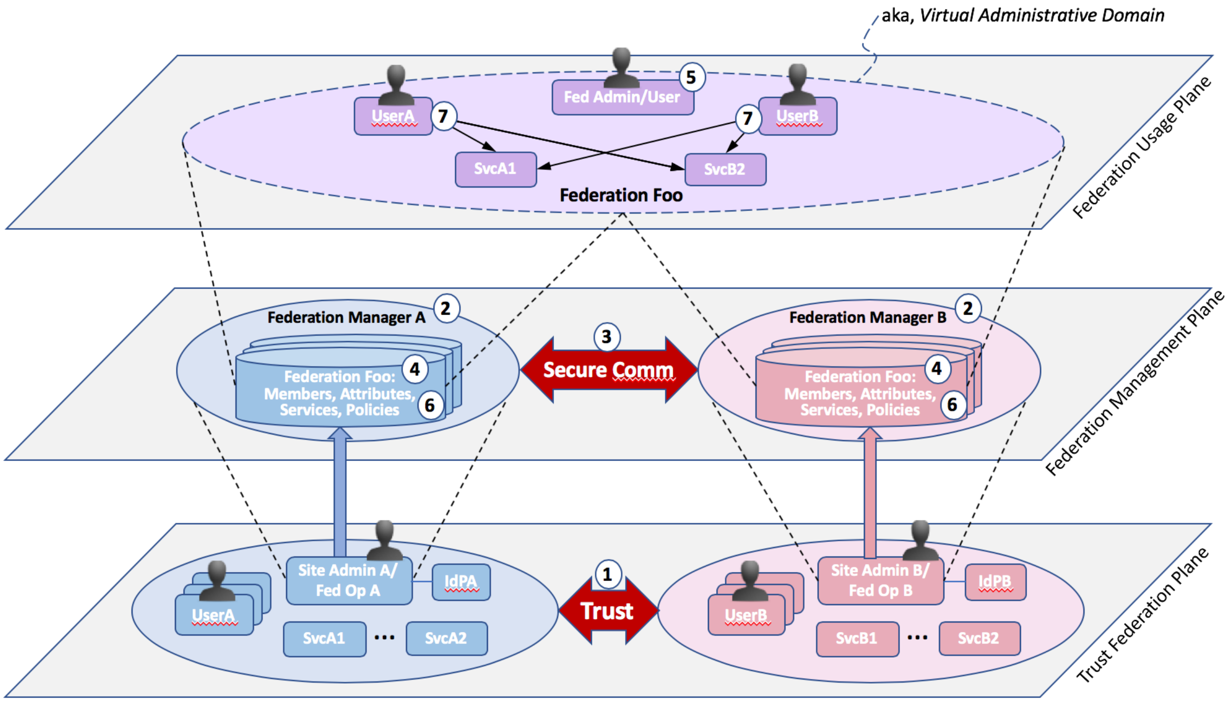


Figure 3. A Three-Plane Illustration of the CFRA.

Figure 3 gives a three-plane illustration of the CFRA using a *peer-to-peer* deployment of two *internal Federation Managers (FMs)* between two sites, *A* and *B*. An FM is the entity that provides the necessary federation functions. The FMs here are called internal since each site is deploying and operating their own FM. We emphasize that this is just one possible deployment and governance model allowed by the CFRA, and it is being used just for the purposes of this preview. Initially, both sites are operating independently. We describe the federation steps as follows:

(1) Sites *A (blue)* and *B (red)* realize that they would like to collaborate for a specific purpose to accomplish specific, joint business goals. Hence, they decide to establish a federation. This must begin with the two sites establishing a *trust* relationship. What constitutes trust is determined by the sites. Part of this trust relationship is the exact structure and governance of the federation they wish to create. We can say this occurs in the *Trust Management Plane*.

(2) Once this is done, each *Site Admin* or *Federation Operator* deploys a *Federation Manager*. Initially, these FMs are "empty" since they are not yet hosting any federations. They can be call *internal* since they are deployed and operated internally to each site.

(3) Once deployed, *secure communications* must be established between the FMs in any way suitable to ensure that their communications are not susceptible to eavesdropping or interception. This is necessary since the FMs must exchange information concerning the management of federations that is valid and trusted. This could be called the *Federation Management Plane*.

(4) Once this secure communication has been established, the two *Site Admin*s can create a common federation. In this example, this is called Federation *Foo*. When initially created, Federation *Foo* is "empty" or unconfigured. What is important is that both FMs maintain a consistent state for *Foo* over its lifetime. From a practical perspective, one Site Admin

may invite another Site Admin to join through their FM, or one site may ask the other to be allowed to join. For this example, how this happens is not critical.

(5) Once Federation *Foo* has been created across all participating FMs, what has actually been created is a *Virtual Administrative Domain*. This is illustrated in the *Federation Usage Plane*. In this plane, Federation *Foo* is neither blue nor red – it is *purple*. Initially Federation *Foo* is also "empty" or unconfigured. However, immediately after creation, a federation's first member would typically be a *Federation Admin*. We note there could be one or more Fed Admins that are users from Site A or B.

(6) Once Federation *Foo* has been created and its management is in place, it can be populated with members and services to accomplish its business goals. The Fed Admin(s) could grant membership and authorizations to other users. Resource/service owners from Site A and B could make services available in Foo by registering their service endpoints and defining their associated discovery and access policies. These users, services, policies, authorizations, etc. could change dynamically over the course of the federation's lifetime.

(7) Finally, when "up and running", the federation logically consists of users and services from either site. These users can discover and use those services. That discovery and use is governed by the specific policies that are associated with those services for this federation. This is possible since Federation *Foo* is a *Virtual Administrative Domain*.

We emphasize again that this is just one deployment and governance model that is possible. The range of such models will be discussed at length later.

3. The Cloud Federation Reference Architecture

We now more formally introduce the NIST Cloud Federation Reference Architecture that captures these fundamental aspects of federated authentication and authorization. This is done by extending the concepts defined in the NIST Cloud Computing Reference Architecture [8], where possible, to include the functions necessary to establish and manage collaborative federations. At this time, we emphasize the following points:

> *This Reference Architecture is a **conceptual model**. The goal of this conceptual model is to identify the fundamental federation functions that **may** be important to different participating stakeholders in different application domains. The subsequent sections of this document identify different governance and deployment models that are possible. We emphasize that there is a wide spectrum of possible federation deployments. This can range from very simple federations where many of the elements of this conceptual model are simply not needed, to very large federations that will require extensive governance machinery to be in place. The use case scenario(s) given in Appendix B are intended to show how this conceptual model can be mapped to more concrete implementations, possibly using existing tools and standards that are augmented to accomplish the general federation behavior described here.*

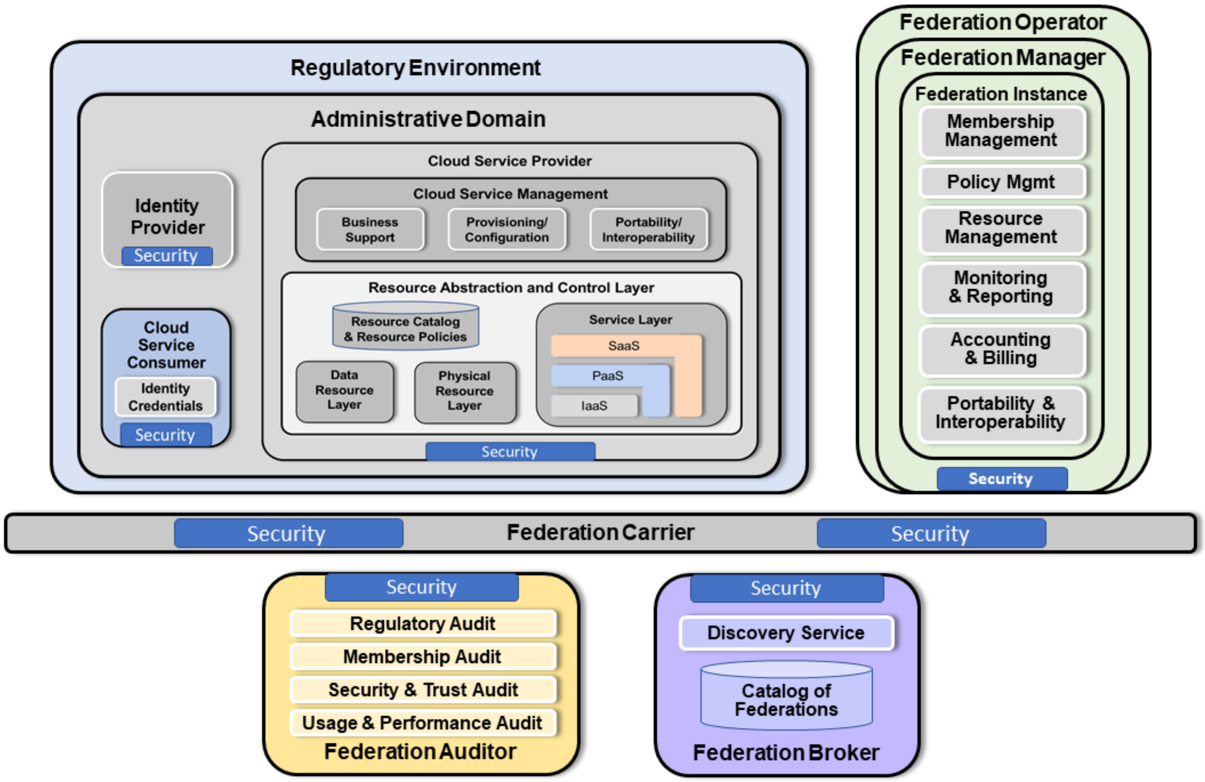


Figure 4. The NIST Cloud Federation Reference Architecture Actors.

Figure 4 identifies the following components that are similar counterparts to SP 500-292:

- Cloud Service Consumer
- Cloud Service Provider
- Federation Manager
- Federation Operator
- Federation Auditor
- Federation Carrier
- Federation Broker

By analogy, these components define the *anatomy* of a federation – simply how it is structured. Despite the numerous similarities, there are some important distinctions and additions to the model that we will be drawing attention to in the discussion. For example, it is necessary to develop the concepts of Administrative Domains (AD) and Regulatory Environments (RE) and it shall be shown how they are fundamental in this model of cloud federation. Cloud federations may be composed of entities that are widely geographically dispersed and exist under jurisdictions that frequently span multiple national and local domains. Furthermore, this model also incorporates two new actors, namely, the Federation Operator and Federation Manager. These actors are central to the operation and management of the federation. Their roles and responsibilities are distinct, but there is a dependence on the federation governance model. This will also be described later on.

We begin by describing the two additional concepts of Administrative Domains and Regulatory Environments in Figure 4 that are central to managing federated environments. We will then describe each of the relevant actors in turn.

3.1. Administrative Domains

The basic, non-federated authentication and authorization process described in Figure 1 above exists within an *Administrative Domain (AD)* that is essentially comprised of:

- An Identity Provider (IdP),
- A Cloud Service Provider (SP), and
- A Cloud Service Consumer, or simply User.

As described above, an IdP issues *identity credentials* to a *Service Consumer*, or *User*. When a User makes a service request, the SP validates the User's credentials with the IdP, and then makes an access decision.

ADs typically operate as independent, autonomous environments. The domain administrators will issue identity credentials, deploy services, and define the policies for who can access what. For example, the IT department at a large corporation will issue credentials to employees that enable them – based on company policies – to use email accounts, and access shared internal websites, databases, collaboration tools, etc.

These independent, autonomous environments are de facto *identity silos* outside of which a user's credentials have no useful meaning. There is no easy, convenient way to securely manage the sharing of specific information and resources among such silos. An organization can stand up a website that is accessible by the general public to make information available. However, to *control* access, general users must be given accounts at that site that determine which resources they can access. How can two or more organizations make the same *kinds of data* available to select subsets of their users? Requiring users to have different accounts at each site is simply not scalable or manageable. Even if users have different accounts at each site, there is no coherent, consistent way for the sites to manage which resources the users can access for a common purpose or project. Federation enables the bridging of these identity silos whereby the participants can jointly define, agree upon, and enforce resource discovery and access policies. A federation could be considered a *Virtual* Administrative Domain.

Often, Federated Identity Management through IdP offers a service akin to Identity as a Service (IDaaS) solutions, where a set of cloud Users are recognized within another cloud using authentication tokens (using OAuth or SAML to provide SSO). Federation within AD goes beyond the identity conversation, adding services and resources.

3.2. Regulatory Environments

All administrative domains exist within some *Regulatory Environment*. That is to say, all users and service providers exist within the jurisdiction of some set of governmental entities, and must observe all relevant regulations defined by those entities. There could be multiple governmental entities at the national, state, and local levels. The users and service providers must observe the union of the regulations defined therein. The Federation Governance Body determines the baseline compliance requirements and defines the strategies for identity and access to data and services in their Regulatory Environment. This must be done through the identity and authorization credentials that are associated with users, and the access policies that are defined for any given resource.

3.3. Identity Provider

Identity Providers (IdPs) are a central part of an AD. There are, of course, many different types of IdP and many different types of identity credentials that they issue to Users. In the simplest form, an identity may simply be an account name and password stored in a local data structure or database. Cryptographically signed bearer tokens may also be issued to users. Public Key Infrastructure (PKI) X.509 certificates could also be issued that are signed by Certificate Authorities. An early form of credentials for distributed, networked environments were Kerberos tickets, where an Authentication Server would issue a Ticket Granting Ticket. These tickets could be exchanged for session keys that could be used to access a resource. Without going into an exhaustive survey of identity provisioning, in all cases, a User's identity is associated with a number of *roles* or *attributes*. Resource access policies can be defined based on these roles or attributes. Generally speaking, an attribute is associated with a specific, narrowly-defined authorization. On the other hand, a role may denote a set of authorizations. Attribute-Based Access Control (ABAC) enables fine-grained access control, while Role-Based Access Control (RBAC) can be easier to manage. RBAC and ABAC define rules that determine access based on a user's roles or attributes for Identity and Access Management (IAM) which provide systems with dynamic methods for controlling access to proprietary resources. Roles or Attributes are, in turn, turned into permissions to "access" functionalities within the federation. These allow users to control and define the lifecycle of a user's access to resources.

3.4. Cloud Service Consumer

For the purposes of federation, a Cloud Service Consumer (CSC) or User is considered to be part of an Administrative Domain. As with ordinary Cloud Service Consumers, they "represent a person or organization that has a business relationship with, and uses the services from, a Cloud Service Provider" [8]. The Cloud Service Consumer has one or more *identity credentials*. At least one credential is typically issued by the local IdP with a User's *home domain*. However, a CSC may also have additional federated identity credentials, possibly issued by the local IdP or a federated IdP (see the Federation Manager).

Similarly, a CSC may browse the *Resource Catalog* of its local Cloud Service Provider. In the context of a specific federation, however, there may also be a *federation-specific* Resource Catalog that the CSC may browse. In both cases, there may be *resource discovery* and *access policies* that the *Resource Owners* may wish to define and enforce.

As with ordinary CSCs, federated CSCs may access resources at any level in the system stack. That is to say, local and federated resources may be at the infrastructure level (IaaS), platform level (PaaS), and the software level (SaaS). Hence, resources can range from instantiating VMs and storage buckets to arbitrary, application-level business functions. When done in a federated environment, this means that resources at any level can be shared among sites.

3.5. Cloud Service Provider

The Cloud Service Provider (CSP) includes all of the components as in the Cloud Computing Reference Architecture [8].

3.5.1. Cloud Service Management

Cloud Service Management is broken down into *Business Support*, *Provisioning/ Configuration*, and *Portability/Interoperability* functions. For convenience, we review each of these areas here.

- **Business Support**
    - *Customer management:* Manage customer accounts, open/close/terminate accounts, manage user profiles, manage customer relationships by providing points-of-contact and resolving customer issues and problems, etc.
    - *Contract management:* Manage service contracts, set up/negotiate/close/terminate contract, etc. Inventory Management: Set up and manage service catalogs, etc.
    - *Accounting and Billing:* Manage customer billing information, send billing statements, process received payments, track invoices, etc.
    - *Reporting and Auditing:* Monitor user operations, generate reports, etc.
    - *Pricing and Rating:* Evaluate cloud services and determine prices, handle promotions and pricing rules based on a user's profile, etc.
- **Provisioning and Configuration**
    - *Rapid provisioning:* Automatically deploying cloud systems based on the requested service/resources/capabilities.
    - *Resource changing:* Adjusting configuration/resource assignment for repairs, upgrades, and joining new nodes into the cloud.
    - *Monitoring and Reporting:* Discovering and monitoring virtual resources, monitoring cloud operations and events, and generating performance reports.
    - *Metering:* Providing a metering capability at some level of abstraction appropriate to the type of service (e.g. storage, processing, bandwidth, and active user accounts).
    - *SLA management:* Encompassing the SLA contract definition (basic schema with the QoS parameters), SLA monitoring and SLA enforcement according to defined policies.
- **Portability/Interoperability**
    - *Data Portability:* The ability of customers to move their data or applications across multiple cloud environments at low cost and with minimal disruption.
    - *Service Interoperability:* The ability of cloud consumers to use their data and services across multiple cloud providers with a unified management interface.
    - *System Portability:* Allows the migration of a fully-stopped virtual machine instance, machine, or container image from one provider to another, or migrates applications and services and their contents from one service provider to another.

As we shall see, all of these same business functions will eventually need to be addressed when we discuss the management of federations.

3.5.2. Resource Abstraction and Control Layer

All clouds need to manage a set of resources. The current state of all of these resources needs to be maintained within some type of registry or catalog. In traditional infrastructure clouds, this includes keeping track of virtual machines that have been instantiated on various physical servers, which storage containers that have been allocated from physical storage, etc.

The identities of the consumers of these virtualized resources need to be established and the usage of the resource needs to be monitored. The CSP, or resource owner, may have resource policies concerning the discovery and access of resources by potential consumers.

A CSP may manage resources at different levels in the system stack, i.e. at the infrastructure level (IaaS), at the platform level (PaaS), and also at the software level (SaaS). What this means is that a CSP can manage not only infrastructure services, but also arbitrary application-level services, i.e. arbitrary business functions.

We can also make some further important distinctions in the types of resources to be managed. Managing access to physical resources is certainly a fundamental part of what clouds do. However, another very fundamental capability is simply managing access to data resources. Since this capability underlies many application domains, this is identified as its own resource layer.

The result is that the resource abstraction and control layer must provide an abstraction for all types of resources that enables it to effectively manage the resources, and while also providing a uniform interface for overall cloud resource management.

Now, as we shall see, when participating in a federated environment, the CSP will need to keep track of resources that are actually coming from other CSPs. VMs or storage buckets may be physically allocated at another site while being used by local users. A remote data owner may wish to make specific data sets discoverable and accessible to a select set of collaborators. This means that the CSPs must be able to agree on, and jointly enforce, the appropriate discovery and access policies.

3.6. Federation Operator

A Federation Operator is an entity that enables the overall operations of a Federation Manager or Managers. This entity has the capability to manage, maintain, and oversee multiple Federation Managers (described in next section). This entity is depicted as superior to the Federation Manager and Federation Administrator. At sites that participate in multiple separate and distinct federations, a Federation Operator will coordinate the activities of the Federation Managers and provide administrative support and maintenance by collecting, processing, and resharing individual federation metadata while following the common policies and legal frameworks shared between federations. However, not all cloud federations have a need for a Federation Operator. In simpler instances, the Federation Manager may be as simple as a server that does the simple management of a federation.

3.7. Federation Manager

At this point, we have introduced the essential concept of what federation entails, and the cloud actors that are similar to their non-federated counterparts. We now introduce the *Federation Manager*. The Federation Manager (FM) is the conceptual entity that provides the essential management functions over the lifespan of a federation. An FM can support multiple *Federation Instances*, or simply *federations*, that can span multiple Administrative Domains.

The Federation Manager occupies a place that is unique to this model and has no counterpart in the original NIST Cloud Computing Reference Architecture. The Federation Manager establishes and operates a federation across multiple sites. It is required to perform a number of critical management functions over the lifespan of a federation instance.

In practical deployments, the FM is not necessarily a single, separate third party. Federated environments may consist of one or more FMs, each of which are operated by an *FM Operator*, but a single FM Operator may operate multiple FMs. FMs may exist in centralized or decentralized deployments. As the scale and magnitude of the federation increases, the presence and activities of the FM Operator will become more pronounced. These are all, however, *deployment issues*. A detailed discussion of deployment issues will be given in Section 5.

We must also make a clear distinction between the Federation Manager and the Federation Instances that "ride" on top of it. While each FM has an FM Operator, each Federation Instance will have a *Federation Owner* that will manage that federation. However, ownership of a federation instance is a *governance* issue. A detailed discussion of governance issues will be given in Section 4.

At this point, we will stay at the conceptual level as we describe the functional components of Federation Instances.

3.7.1. Federation Membership Management

A federation is intended to be a security and collaboration context wherein the participants can define, agree to, and enforce joint resource discovery and access policies. Clearly, there is a need for the notion of *federation membership*, i.e. keeping track of who is actively participating. This also means that there must be some process for vetting and on-boarding new members, i.e. granting membership. The entity associated with this process could be called the *Federation Administrator*, or simply the *Fed Admin*. We note that while individual Cloud Service Consumers could have federation membership, it could also be possible for entire organizations to have a *site membership* (these issues will be discussed in more detail in Section 4.4).

The notion of membership in a federation implies some notion of *identity* within that federation. While some federations may simply rely on a member's identity credentials from their home institution, this may be limiting since managing a federation may be much more effective if a member's identity were associated with a number of *federation-specific roles* or *attributes*. Hence, a *federated identity credential* could possibly be derived from a member's home institution credentials, or could be a separate credential issued by the Federation Manager acting as an IdP. In any case, the semantics associated with these federation-specific roles or attributes would depend on the federation's business needs, and would have to be well-known to all participants or participating sites. Likewise, any federated identity credential should only be meaningful and useful within the context of the federation for which it was issued.

Another fundamental issue that must be mentioned is the *release of identity attributes*. Identity Attributes relate to Digital Identities, as described in ISO/IEC 24760-1, such that they allow for the assessment and the authentication of a user interacting with a system without requiring the involvement of human operators. Identity Attributes are the digital representation of a set of claims or characteristics about a given user within a certain context of the federation (attributes can be as simple as combinations of name, roles, location, or age). Authorization and Authentication reflect on those identities. Authentication is a key component of the trust-based identity attribution system; providing a codified assurance of the identity of one entity to another. Authorization reflects the understanding that an authenticated user can access a set of resources.

Any federation member may have multiple identity attributes from their home institution and within any given federation. When requesting service from an SP, only a subset of a member's identity attributes may be necessary to enable a proper authorization decision. Some federations may wish to limit the release of identity attributes to that minimal set of attributes. For other federations, this may not be an issue.

Finally, we again note that the necessary extent and strictness of membership management is dependent on the requirements of any given federation. Some federation may have very lenient membership requirements, i.e. any user or site could self-identify and join the federation. Other federations may have very strict membership vetting and on-boarding requirements, with very tightly controlled federated identity credentials.

3.7.2. Federation Policy management

A federation may have to observe a number of policies. As illustrated in Figure 4, each AD participating in a federation exists within the jurisdiction of some regulatory environment. This regulatory environment could involve national, state, and local regulations that must be observed. Clearly, a Federation Manager may have to reconcile the different regulatory requirements of all participants, or at least, enforce the regulations local to each participant.

Participation in a federation may also involve some degree of expectations as a condition of membership. Generally speaking, each resource owner will retain complete, unilateral control over their resources. However, to realize the benefits of collaboration, the resource owner may need to agree to provide access to their resources based on the roles and attributes governing the actions of members within the federation. These expectations could possibly be expressed formally in a *contractual agreement*, and possibly be codified in policies. As an example, a resource owner may need to agree to provide data of a certain type to federation members that possess the necessary authorization attributes for that data type. As another example, a resource owner may have to agree not to unilaterally change their access policy unless specific conditions occur, e.g. an intrusion has been detected.

This is also relevant when members of a federation are located within different geographical jurisdictions that span multiple national and local domains. Some regional variations will exist due to the specific laws or government services, which require specific federation-to-federation agreements (policies) to be put in place for the different services provided by each federation to have an agreed-upon level of equivalency and access.

Within a same region, often, access for education and research purposes exists (for example, InCommon in the United States of America; also, the international roaming service, eduRoam, for researchers visiting institutions). All such research and education specific to federation provide access to the terms of their Federation Policy, as well as to additional documents such as participant agreements, privacy policies, expectations, dispute resolutions, trust relationships, and more. In addition, research federation providers maintain and publish a registry of organizationally valid metadata that is vetted, signed with a cryptographic key (often requiring two human actors), and published periodically at well-known public locations. Metadata processes are also controlled using the Metadata Registration Practice Statement (MRPS), which covers the lifecycle of registration, management, and generation of the metadata. The Security Assertion Markup Language (SAML) is an open standard for exchanging authentication and authorization data between entities. It is often used to represent the relationships between IdP and SP.

Resource usage may also be governed by Service Level Agreements (SLAs). Again, to realize the desired federation benefits, some services may need to meet certain throughput, latency, and availability requirements. From the resource owner’s perspective, the owner may wish to meter or throttle access to certain resources. For example, a resource owner may wish to limit access to a given percentage of the resource’s total capacity.

3.7.3. Federation Resource Management

In any federation, there must be some mechanism whereby members can find the resources that are available within that federation. This implies some type of *catalog* and *discovery services* for the resources that federation participants are making available. This, in turn, implies that resource owners must *register* their resources with the catalog/discovery service. There is clearly a variety of ways that such a catalog/discovery service could be implemented, but this is out of scope for this discussion. We do note, however, that resource discovery presents a fundamental semantic interoperability challenge: How can the semantics of a resource be represented and understood, such that a proper selection decision can be made? In the simplest cases, this can be addressed by a type system that is defined and well known beforehand. In more general cases, however, more extensive sets of metadata will need to be associated with resource descriptions.

Not all federation members may be authorized to use – or discover – all resources within a federation. Either by federation-wide agreement, or by individual resource owner requirements, there may be a *resource discovery policy* associated with any given resource. When invoking the discovery service, a federation member's roles and attributes could determine which resources the member can discover. A member should only be able to discover those resources for which they have authorization to use in some capacity.

Once a member invokes a known federation resource, some type of access control may still be desired based on the member's roles or attributes. We note that a resource owner may wish to limit or meter the amount of the resource capacity that is being consumed, either by the specific federation member, or by the federation, as a whole. Again, how this is implemented is outside of the scope of the current discussion; but it is clear that such management requirements are associated with specific federations and should be coordinated with the Federation Manager.

In the original NIST Cloud Computing Reference Architecture, the Cloud Broker provided three distinct capabilities beyond those of a Cloud Provider:

- Service Intermediation: Enhancing a given service by improving some specific capability and providing value-added services to cloud consumers.
- Service Aggregation: Combines and integrates multiple services, possibly from different providers, into one or more new services.
- Service Arbitrage: Similar to service aggregation, service arbitrage means a broker can choose services from multiple providers.

These functions all support the concept of an environment in which a User goes through a single Broker to get access to resources, rather than going to multiple providers directly. A Federation Manager could provide these same capabilities, yet its critical function is to enable various *federation governance models* to be jointly defined and enforced by the participants in a federation.

Multi-clouds derived from commercial, infrastructure cloud providers have relatively narrow governance requirements. Commercial cloud services are discoverable by anyone, and the only authorization credential that a user really needs is a valid credit card number. This could be considered a simple form of federation with a very simple governance model. However, general federations must enable the federation participants to jointly define resource discovery and access policies that are driven by goals of the specific federation, writ large. This is the function of the Federation Manager.

3.7.4. Federation Monitoring & Reporting

Monitoring is a basic function that supports many other functions. This includes usage, performance, health and status, and security. Besides being able to collect the necessary metrics at the appropriate places, this data must be reported to where it can be used. In many cases, simply keeping such data in system log files will be sufficient. In other time-critical cases, however, *event* communication may be necessary, i.e. communication that must be acted on immediately and cannot be buffered in any way. Security incident reporting falls in this category. Cybersecurity, in particular, is a necessary component to keeping federation service safe. Proactive FSPs often aim to detect breaches and vulnerabilities early to secure access to resources. Reporting, additionally, allows an FSP to understand the resource usage of its users; these metrics are important for the purpose of billing.

3.7.5. Federation Accounting & Billing

Virtually, all federations will want to keep track of resource usage on their systems by their members. For many federations, there may also be a need to associate this usage with a pricing or cost schedule where sites or members can be billed for payment. This will be increasingly necessary as federations increase in size, and non-trivial amounts of resources are consumed in support of collaborative federations. As a simple example, if a federation participant is serving data to other participants, this may incur direct costs from the serving partner's cloud provider. The serving partner may need to recoup these costs from the partners that are requesting and consuming the data. Billing processes information received from Reporting; as metrics are collected and aggregated, they are then processed through different rating modules. It is because the Monitoring is able to determine the User's access to resources and services. This telemetry relates in general to the data, networking, and compute usage.

3.7.6. Federation Portability & Interoperability

Federated environments will have many of the same portability and interoperability issues that non-federated environments have. Even if a partner makes data available within a federation, *data portability* would be needed to enable consumers to access and retrieve data with reasonable cost, and understand the data format. Different federation partners from different sites may offer the same type of data or services. Ideally, these should have a unified management interface; but in practice, these may have been deployed at different times with divergent interfaces. In this case, some type of service mediation that presents a more unified Application Programming Interface (API) may achieve better *service interoperability*. Likewise, moving images (containers, virtual machines, disks) or containers among federation partners to achieve *system portability* is desirable.

Federation, by itself, does not address these issues. A federated environment will, however, define the "scope" in which portability and interoperability may be needed. When forming a federation to address joint goals, an initial set of partners may also be able to define their portability and interoperability requirements. By constraining the necessary scope, a federation may be able to make these requirements more tractable.

3.8. Federation Auditor

In the broadest sense, a Federation Auditor will be an independent third party that can assess compliance for any type of policy associated with a federation. While a Federation Auditor may address compliance assessment issues similar to those of an ordinary Cloud Auditor, we note some significant differences:

- *Usage & Performance Audit:* Some federations may wish to audit for usage and performance, perhaps in support of evaluating Service Level Agreements associated with the federation.
- *Membership Audit:* Federation membership may come with a set of expected behaviors as a condition of membership. A Federation Auditor could assess whether members are complying.
- *Security & Trust Audit:* This encompasses all security issues but with the added concern that a federation must rely on a number of trust relationships. Security and trust could be based on auditing for acceptable configuration, privacy, confidentiality, minimal release of identity attributes, etc. In the same way that members may have requirements, Federation Admins may have similar requirements that may be audited.
- *Regulatory Audit:* Since federations may span different regulatory environments, a Federation Auditor may be required to assess whether joint and local regulations are being observed. The Federation Policy Management and enforcement relies on a review of these documents and how those affect the adherence to both Membership, and Security & Trust.

This is but a cursory overview of possible, federation-specific auditing requirements. A more thorough examination of relevant security controls could be done to apply the controls identified in NIST SP 800-53 to include federation-specific security.

3.9. Federation Broker

When federations become a widely accepted method of managing collaborations, many federations may be active at the same time. While some federations may wish to be known to only a select set of members, other federations may wish to be *discoverable* by potential members. This need could be addressed by a *Federation Broker*. This would provide the traditional function of a broker to connect “buyers and sellers”. This implies that there must be some type of *Catalog of Federations*, along with a *Discovery Service*. This Discovery Service would need to be able to categorize federations based on specific properties, such that appropriate potential members could identify federations they may wish to join. Federations may choose to release as much or as little detailed information to the Federation Broker to limit discovery of their catalog of services and data.

Extending from the Federation Policy management description of research federation providers, the metadata exchange mechanism needs to be common for the participants of a given federation, with their schema definition easily accessible and available. Furthermore, as in similar directory services, such as the Domain Name Resolution (DNS), the hierarchy of Federation Service Provider (FSP) must contain a root level with an accessible, vetted, and signed registry of metadata published at a publicly know location. This will allow Federation Managers to confirm the origin and authentication of the metadata exchange and its integrity, making it more akin to Domain Name System Security Extensions (DNSSEC). The reasoning being the need for the signing of this metadata information is to prevent what is commonly referred to as “cache poisoning”, where metadata content is spoofed (corrupted) within a copy of the metadata. Because the public signing keys are known and published, a broker user is able to confirm the validity of this content.

It is recommended to update the metadata's content following a known schedule as to enhance the broker's role as a facilitator for the Discovery Service, such that the lifecycle of the federation participants within a metadata provider provides information on registration, management – including removal from the federation – and updates to services and resources provided within the federation.

Beyond its role enabling Discovery and Cataloguing, a Federation Broker provides additional capabilities:

- Service Intermediation: provides value-added service. In this case, the knowledge of the available components (resources and services) hastens the User's access to the resources needed – i.e. compute, data, and networking – with enhanced access to those as locally as possible for efficiency and, in case of billing, cost worthiness.
- Service Aggregation: combines and integrates multiple services, possibly from different providers, into one or more new services. This optimization step can take many forms, but one of the key broker roles is to provide information from one federation to another using the metadata model of said federation participant. In particular, this provides a compatibility matrix for communication protocols supported – at minimum – by each federation (for example, security requirements for a given federation member to communicate with another member). The value added, in this case, can be described simply as providing the results of the Transport Layer Security (TLS) Cipher suite negotiation to each participant of the federation.
- Service Arbitrage: similar to the service aggregation but with a flexible dynamic choice. In practice, when used, this function might prefer, for similar characteristics, federation participants that follow the User's choice, be it to save money, to be more local, or other User criteria.

Federation cloud brokers allow Users to decide between multiple federations. Users benefit from their service arbitrage capabilities. In order for these capabilities to be useful to Users, brokers need to continuously update their metadata, as well as have relationships with members of the federation to be able to match changes to protocols and provide accurate information.

3.10. Federation Carrier

In much the same way as the Cloud Carrier in non-federated environments, the Federation Carrier will provide "connectivity and transport of cloud services between cloud consumers and cloud providers" [8]. While this may include providing communication with a given SLA, and providing secure connections between cloud service providers and consumers, this could be taken a step further in federations.

The notion of a federation as a collaboration and security context could be enhanced by isolating its traffic at the network level. Software-Defined Networks (SDNs) could be used to define a communication environment that supports just the members of a federation. This SDN would have to be dynamically reconfigured whenever a member joins or leaves a federation. Hence, the SDN API would have to be integrated into the appropriate Federation Manager(s), such that any necessary reconfiguration could be done at the appropriate time. This layer supports components such as migration, i.e. the capability to move VMs, containers, or disk images from one federation member to another. While this would probably not be a trivial endeavor, it offers interesting possibilities for pushing some of the federation management machinery into the network level.

3.11. Security

Security can cover the areas of identity/authentication, authorization/policy, integrity, privacy/confidentiality, and nonrepudiation. It is clear that the actors in this reference architecture squarely address the issues of federated identity, authentication, policy and authorization. Security negotiations are the steps taken to establish a minimum level of trust for the exchanges between federation members. The purpose of the *Security* function shown here on each of the actors, on the simplest level, is to secure the communications among them. This means that the source and destination for any communication must be able to determine each other's identity, and that the information communication has integrity and perhaps privacy. A number of standards and tools exist for securing such communications which will be discussed in Section 8: Relevant Existing Standards.

However, as the discussion of the other actors should have made clear, the establishment and management of federated environments is, at its essence, the establishment of a security and collaboration context wherein all necessary security requirements can be met. In the context of a federation, this means (a) being able to establish the identity of federation participants; (b) being able to specify which resources are to be shared within that context; (c) being able to define the discovery mechanism and policy associated with any resource, such that only the authorized users with a given federation can discover a resource; (d) ensuring that only authorized users access a resource; and (e) ensuring that all such interactions are done with information integrity and privacy. We shall examine these security requirements in more depth as we examine the lifecycle governance requirements of a federation.

4. Federation Governance: Requirements and Options

In Section 3, the conceptual architecture for a federated cloud was presented. In this section, we present a discussion of federation governance which describes how the pieces of the architecture of a federation operates, works together, and interacts. Hence, we discuss its governance with its requirements and possible options.

The first step in federation governance is to clearly define what is being governed. Therefore, there must be a clear set of essential characteristics for what a federation instance is. Once we have a clear understanding of what constitutes a federation – its essential characteristics – then we can examine the governance necessary for when a federation is instantiated or created, and for each step in the rest of the federation's lifecycle.

4.1. Essential Characteristics of a Cloud Federation

Every federation has a specific configuration of the federation instance elements identified in Section 3. However, all federations have a set of essential characteristics that they share. These characteristics are found in all federations, large or small, and may be instituted or implemented according to the federation participants or governing body.

- *Resources to be shared and their metadata*. While the types of resources (data and services) to be shared might be open-ended, each federation has certain resource types that are commonly shared to meet the goals of the federation. These data and services will need to be clearly identified and described with some well-known metadata. Therefore, this represents a potential semantic interoperability requirement that will typically be addressed by standardized schemas and ontologies. Any working federation environments could leverage work done in this area, as well as work related to the Internet of Things [9].

- *Roles & Attributes*. Federations will have a set of roles or attributes that are associated with the actors in Section 3. These roles and attributes define the responsibilities that different members have, or what actions they can take and use to make various policy decisions governing the operation of the federation. The meaning of these roles and attributes also needs to be well known to all members.
- *Resource Discovery*. After a federation is instantiated, various member resources will need to be made available to and accessible by the other members. There needs to be a mechanism in which members can discover available resources and services. This implies that there needs to be some type of resource catalog and discovery service as described in Section 3.9. The details of how these catalog and discovery services and their semantics are implemented can be federation-specific. Likewise, the resource discovery policies associated with the cataloged resources can be federation-specific, and based on federation-specific metadata attributes and roles or attributes associated with any member that is searching the catalog.

  In some circumstances, the federation members may jointly agree to define the discovery policy for the different types of available sources. This may be desirable and necessary for the federation members to achieve the goals of the federation. In other situations, however, the resource owner may wish to define the discovery policies for their own resources. These policies would nonetheless have to be based on the resource metadata, roles, and attributes defined within the federation. If a federation only involves a small, fixed set of services that each member must offer to any other member, then the resource catalog and discovery process become very simple. In the more general case, however, there will be a definite need for resource metadata and service discovery policies.

  The availability of a metadata store to list and describe the federation resources supports the federation members by sharing vetted information about said resources and services, providing such metadata information for a given federation in a persistent shared location. Cryptographic signing of this metadata prevents its unauthorized modification.
- *Federation Membership*. A federation consists of a set of users that are members, for some definition of membership. Each federation may define its membership based on a set of requirements. Some federation may allow users to self-identify and join with essentially no identity proofing or new member vetting. Other federations may have strict requirements in this regard. Some federations may have definite expectations or conditions of membership that each member is expected to observe. Joining a federation may also require specific legal agreements concerning how a member is expected to support the goals of the federation and not abuse any federated resources. In practice, we also note that a distinction could be made between *individual memberships* and *organizational memberships*. This type of distinction can have great impact on the federation's governance model.
- *Federation Member Identity Credentials:* Federation members can have a type of federation-specific identity credentials. As stated above, what exactly constitutes a "member" is to be determined by the organization; hence the exact form of the identity credentials of a member is to be determined as well. The form these credentials take, and how they are related or traced back to a member's identity in their "home" institution when they were granted membership in the federation, are also a matter for governance.

- *Has a process to grant or revoke federation membership:* Assuming that members are not allowed to simply self-identify and join, then there must be a mechanism which allows granting and revoking memberships, some entity that has the authorization to grant or revoke membership. This authorization could be a role or attribute granted to specific federation members. As part of this role, the Fed Admin would have the responsibility to enforce new member identity proofing or vetting policies, if any, such that an authorized and authenticated user could access a set of resources. If the federation has any conditions that require membership revocation, then the Fed Admin has the responsibility to execute the revocation. The Fed Admin may also have the responsibility to monitor, detect, and verify when such conditions have occurred.
- *Has a process to grant or revoke member roles or attributes:* Assuming that not all federation members are "equal" and can access all shared resources equally, then there must be some method of distinguishing among what different members can do. Clearly, this is done by assigning different roles or attributes to members. Hence, there must be an entity that has the authorization to grant and revoke member roles or attributes. This entity will typically be a Fed Admin.
- *Governance:* Other aspects of how a federation is to be governed, managed, and operated are captured in a larger discussion of the federation ecosystem. This section will be covering many of those aspects.

A cloud federation ecosystem is a specific configuration of semantics and governance. The formality of the ecosystem depends on the needs of the federation participants. A single individual or organization could create and own a particular federation definition type. Probably more common, though, an initial set of federation participants will agree to define a federation definition type that supports the participant's goals for creating a federation.

Once created, the participants will want the foundational aspects of the federation type to be static and immutable, but flexible enough to accommodate the dynamic aspects of a working federation. For example, the semantics and certain aspects of the governance are items that can remain static. Having a stable federation type that is well known by all participants will certainly facilitate all other aspects of governance. On a practical level, the dynamic quality will affect a change in requirements and, thus, in the federation. If the federation is created by a single individual or organization, then conceivably they could unilaterally change it and force all participants to adjust. It is possible that a single federation owner could be a *Federation Provider* that instantiates different types of commercially useful federations in a marketplace of federation consumers. While the single owner could have the authority to change a federation type, any potential changes would have to be weighed against the potential impacts (positive and negative) to their federation revenues. A more common scenario is that a committee of federation participants will decide on the necessary change, and introduce it into operations throughout the federation in an orderly fashion that causes the least disruption. To make an analogy with software engineering, modifications within the federation parameters should be reflected in means that are interpretable by the federation systems, such that a level backward compatibility is possible. Unless the changes are necessary to reflect complex changes in the policies and procedures of the federation membership, evolutive changes should be reflected. If this is not possible, access to certain resources or services might be unavailable for previously authorized Users.

4.2. Federation Instantiation

Once the formality of creating and establishing a federation type is complete, how does the federation become instantiated and operating? Who has the proper authorization to instantiate a federation according to its ecosystem configuration? The answer to this question depends on the formality of the ecosystem configuration ownership. It is conceivable that formal federation types could become intellectual property. Using such a federation type could require obtaining a license, paying for a subscription, or agreeing to some other type of revenue scheme. Others could be open source or in the public domain. Simple federations could be informally defined by individuals or small groups that have no particular legal status. The upshot is that assuming the appropriate federation machinery exists, anybody could instantiate a federation, but only as constrained by the configuration ownership.

Once a federation is instantiated, we can say that whoever created the original instance owns it. This entity could be called the *Federation Instance Owner*, or simply the *Federation Owner*. Depending on how the federation system works, whenever a federation is instantiated, it is considered empty and has to be populated. It could be populated with roles, attributes, members, resources, policies, etc. to get the federation operational. If such background information is well known, then it can be used as a basis for a federation constructor that instantiates the federation with all the supplied parameters. However, given that federations and federation systems could be (and probably will be) inherently distributed across multiple administrative domains, having a completely automated instantiation process may be problematic. In the near term, it will be more likely that federations will have to be created by humans-in-the-loop at each of the participating administrative domains.

Governance must be properly handled after instantiating a federation. While a new federation may have an owner, it could be considered to have zero members. To properly handle all subsequent governance, the first member of a new federation must be the *Federation Administrator*, or simply *Fed Admin*. Most commonly, the Fed Admin will be the Federation Owner. However, it is completely possible that the Fed Owner could immediately grant membership to a new member, and then transfer or delegate the Fed Admin role to that new member. In either case, once the federation has been instantiated with a Fed Admin, that Fed Admin manages all granting and revoking of membership, authorizations, etc. From a practical perspective, given how integral an administrator is to a federation, it should be possible to specify the Fed Admin as a configuration parameter to the instantiation process.

The sharing of roles, and the capability to have more than one entity with a given role within this federation facilitate its functionality within the FSP, Federation Operators, and underlying Cloud Service Provider. In some systems, a quorum is used to control each role, with means to replace entities from their roles with enough votes. This prevents the risk of single point of failures for certain federation roles.

4.3. Federation Discovery

Once a federation has been instantiated, potential members will need to know that it exists. In general, new members can be added by (a) the Fed Operator extending an invitation to potential new members, or (b) potential new members requesting membership. How a Fed Operator identifies potential new members, or how potential new members identify federations they wish to join, could certainly be done by traditional methods, e.g. word of mouth, or other modes of communication outside of the federation itself.

While some federations may wish to be known only to a select set of members, other federations may wish to be more readily discoverable by potential members. Making federations more discoverable could be supported by a Federation Discovery Broker service. Such a broker service could be separated from the federation itself and implemented in a variety of ways. A federation owner that wishes to make an existing federation more discoverable could register information with the Federation Broker. This information could be the data on the federation ecosystem data and metadata. It should, however, include a Point of Contact for the federation. This should be a Fed Operator that has authorization to vet potential members and grant membership. As financial considerations are also part of access within the federation, the billing and accounting for proposed resources need to be listed, such that potential members are able to make a reasoned choice as to the use of certain resources and services within the federation. Often, cost calculators are part of the additional services provided by such federation brokers, and potential members are able to compare the use of given clouds for common resources. The Federation Operator may wish to make their federation discoverable only by certain types of potential members. Hence, similar to resource discovery within a federation, the Federation Operator may wish to specify some discovery policy that the Broker must enforce. How federation discovery is supported, or not, is an important aspect of governance.

4.4. Federation Membership

Like any other human collaboration, the success of a federation depends on its goals and the participants that choose to join and make it work. While a federation may have an initial set of members, this group may not be static for the entire lifecycle of the federation – members may come and go over time. Hence, at this point, we will assume that at least one conceptual Fed Operator exists that can grant and revoke federation membership, and keep track of members that leave the federation.

4.4.1. Membership Criteria and Requirements

Any federation may have a set of criteria that a potential new member must satisfy as a condition for granting and retaining membership. Some federations may have essentially no criteria where any user can self-identify and join the federation, but many federations will have specific criteria that are deemed necessary or desirable to achieve the goals of the federation. Such criteria might include:

- *Be a recognized stakeholder in the federation's goals.* Members should have a recognized need to know or use the data/resource expected to be available in the federation. Members that own data or resources that are recognized to be directly useful to the federation may be expected to make these resources available.
- *Reasonable cooperation.* While most resource owners will want to retain complete and ultimate control over their resources, if a resource owner joins a federation, there may be some expectation that they will share their resources in a reasonable manner to support the goals of the federation.
- *Acceptable Use.* Members are expected not to abuse the available resources, e.g., not to exceed a level of usage for a given service. Such expectations could be codified in an *Acceptable Use Policy*.
- *Security Policy.* Members are expected to control access to resources and service, to prevent the proliferation of online threats such as data loss, or unauthorized access. Auditing is part of the tools available to the federation to confirm the member conforms to its *Terms of Service*.

- *Operational Support.* A member may be expected to support the federation by supporting the monitoring and reporting of resource usage, perhaps as part of accounting and billing. There could also be incident reporting requirements for events that may be important for the other federation members to know about. Some federations could even require a legal agreement as a condition of membership that clearly defines a member's responsibilities and liabilities.
- *Active Participation.* Members that are idle for a long time and not contributing to the federation may be asked to leave or have their membership revoked.

To reiterate, these criteria are some possible ones that can be used. Some federations may be very informal while others may have very strict membership criteria and requirements. In all cases, though, any such criteria and requirements should be clearly defined.

4.4.2. New Member On-boarding Process

Assuming new member criteria are well-defined, how are prospective new members vetted? This can also be called *identity proofing* or *identity verification*. Again, this aspect of federation management could be addressed with varying levels of formality and process. This could include:

- Simple self-identification
- Recommendation from current member
- Known reputation
- In-person interviews
- Verification of identity credentials by employer/host organization

Generally speaking, the Federation Owner could be able to decide the desired or necessary vetting process. However, this could also be decided by some type of governing board for a given federation. We also note that a Federation Provider may or may not have guidelines or requirements for new member vetting. Hence, the CFRA identifies new member on-boarding as a requirement but does not mandate any specific approach.

As an illustration of different member on-boarding requirements, consider the following example. A set of data catalog provider which to federate to present a federated catalog to their consumers. The catalog providers may have strict requirements concerning the identify verification of a new catalog provider that wishes to join and become a member of the catalog federation. However, this catalog federation may wish to serve catalog data to the widest possible consumer base. Hence, to become a user of the federated catalog may have very lax requirements. A user may be allowed to simply self-identify, or log-in with some pre-existing social media credentials. While such users can be technically considered to members of the federation, they have very limited authorizations. This is another example of the range of deployment and governance models that are possible for federations.

4.4.3. A Member's Federation Identity

What constitutes a federation member's identity? A federation member must have some type of identity credential whereby their actions within the federation can be governed by policy (if any). A federation credential could be very simple. It could be identical to the member's credentials when their membership was granted. In many cases, however, a member's federation credentials will be derived from their original credentials. This will be especially true when the federation has a set of federation-specific roles and attributes. There must be some way to associate these roles or attributes with different members. Being able to make such associations is what identity credentials are used for.

This implies that a Federation Manager may act as an Identity Provider to issue federation-specific identity credentials. A Federation Manager could also simply act as an Attribute Authority to issue identity assertions concerning federation-specific roles or attributes.

In general, the notion of managing a federation member's identity can be called federated identity management. A Federation Manager may need to ingest various kinds of identity credentials from different IdPs and map them by some means to a credential that is meaningful within the federation. This is related to the notion of Single Sign-On where one credential can be used for multiple services or sites. For example, being able to log-in with one's Google ID or Facebook ID is another example where a service provider is relying on these external identity providers to make an access decision. This is performed using a technical solution using the OAuth open standard for secure access delegation, which allows third-party to access and retrieve selected information in order to authenticate users. For the kinds of federations being considered here, though, more comprehensive and federation-specific methods for managing identities and authorizations will be needed.

4.4.4. Individual and Organizational Memberships

Another important distinction that could be made concerning federation membership is that of *individual* versus *organizational* memberships. It is common to think of a user as an individual entity that has authorizations and uses resources. However, users will also be commonly part of some larger organization. Hence, the notion of an organizational membership in a federation will have great utility and, in fact, may be the most common way that federations are used.

The difference between individual and organizational membership has clear implications for federation governance. When granting membership to an organization, what are the membership criteria and requirements? What is the on-boarding process for an organization? All of the considerations discussed above for these concerns would still be relevant, but there could be additional specific requirements when the entity being on-boarded is an organization.

Does an organization have a specific identity credential within the federation? While this might be possible, another perspective is that an organization will have a federation member with special roles or authorizations. This special member might be called a *Federation Site Administrator*, or simply *Site Admin*. As the name implies, a Site Admin is a type of Fed Admin, only with an administrative scope that is limited to the local site. A Site Admin could have the authorization to:

- Grant/revoke federation membership to local individual users within that organization or administrative domain,
- Grant/revoke roles or attributes to those local individual members, or
- Grant/revoke authorization for a Service Owner to register their service(s) in a federation, and define access policies based on the federation attributes.

This notion of a Site Admin implies that multiple trust relationships must exist among the Federation Owners, the Federation Manager(s), and the other Site Admins. On a practical level, it may be very common for federations to occur among organizations that wish to collaborate. As such, it may be very common for the necessary governance to be achieved by special members such as Site Admins. It is conceivable that other types of organizational memberships could be possible that would need to be supported by other types of special membership roles.

4.5. Federated Resource Availability and Discovery

Once a federation has been instantiated and members inducted (individual or organizational), these members must decide which resources they wish to share within the context of a specific federation. Without loss of generality, we can say that every resource or service will have a Resource or Service Owner. Regardless of whether this owner is an individual or organizational member of a federation, they should retain ultimate control over their resource(s). Nonetheless, joining a federation implies some support for the goals of a federation, along with some expectation of the specific types of resources to be shared. Hence, Resource Owners must decide which resources (services) they wish to make available within the context of a federation. That is to say, the Resource Owner must decide to register their resource(s) with the federation.

Once Resource Owners have decided to make their resource available within a specific federation, there must be some mechanism whereby other members can discover the existence of those resources. This implies that the Federation Manager must provide some type of resource catalog along with a resource discovery mechanism based on that catalog. While all resources within a federation could possibly be available to all members of a federation, in general, there may be some resource discovery policy that governs which federation members may discover and use which shared resources. These discovery policies would typically be based on the commonly known federation attributes. Discovery of information is also dependent on the access level of the federation member. When probing the discovery mechanism for available resources, validation of access level should be performed such that only authorized content is returned. This intersection operation between the federation member’s known attributes and the federation resources’ available attribute is important when needing to control limited or controlled access resources.

An outstanding issue is who gets to define discovery policy. One possibility is that the federation ecosystem includes the resource discovery policies for the types of resources that are expected to be shared within the federation. Of course, these resource types and associated attributes would have to be commonly understood. Another possibility is that the Resource Owner gets to define the discovery policy for their resources. In this case, the Resource Owner would have to understand how to define the desired policy based on the attributes that are commonly understood across the federation.

There are many ways that resource catalogs and discovery services could be implemented such that discovery policies are enforced. This will be discussed at more length in the next section on Deployment Models.

One other concept to present concerning resource availability is that of symmetric and asymmetric federations. When two (or more) administrative domains join in a federation, a common use case is that there will be users and resources in each domain that become part of that federation. This can be called a symmetric federation since authorized users in either domain can use the resources being offered by the other domain. However, it is also possible that some federations may be asymmetric, in which case an administrative domain that joins a federation may provide authorized users or resources, but not both. This may be the case for a data provider in a specific application domain. That data provider may wish to provide data to selective groups of external users for specific projects. While a useful property to recognize, whether a federation is symmetric or asymmetric does not fundamentally change how resource discovery or access must be managed.

4.6. Federated Resource Access

Once a federation member has authenticated to a federation instance, identity credentials of some sort have been established, and resources of interest have been identified, how are those resources invoked? Clearly, when invoking a desired service, the federation member must also provide their authorization credentials whereby the Resource Provider can (a) validate the member's credentials, and (b) make an access decision based on the access policy defined by the Resource Owner. While such access policies may be based on common (non-federation-specific) roles or attributes, some federations may wish to define federation-specific roles or attributes on which policies can be based.

We note that resources may include traditional cloud infrastructure services -- allocating compute, storage and networking resources -- up to arbitrary, application-level services. The policies involved could manage consumption limits or common create, read, update, delete (CRUD) operations on the resources involved. Some of these policies may be part of a larger set of Acceptable Use Policies that a federation defines as a condition of membership.

Again, we note that there could be many ways to implement the validation of credentials, how access decisions are made, and where they are enforced. Different implementations approach will have different implications concerning security and necessary trust relationships. Such topics will be directly covered in the next section on Deployment Models.

4.7. Monitoring, Reporting, Accounting, Auditing, and Incident Response

During a federation's lifecycle, the Federation Operator, Federation Manager and the members should be prepared to perform monitoring and reporting of relevant conditions and events. Such reporting may cover routine operations, such as resource usage. Such reporting could possibly be kept in various member log files, but could also be reported to some centralized or consolidated logging facility. Such reporting could be used for accounting and billing among federation members, and possibly a Federation Provider. Federation Auditors may also need access to such log files to verify that the information reported is valid and that the necessary policies have been observed.

Another important function for monitoring and reporting is to support incident response. If any unexpected or malicious events are detected, the federation may wish to take some form of corrective action. If a federation member determines that some unexpected or malicious event has occurred, for example unauthorized data exfiltration, the member may unilaterally change the access policy for their resources. In extreme cases, the member could disallow access to any of their resources. Similarly, if the Federation Manager observes an unexpected or malicious event, it may decide that unilateral action is necessary. Such unilateral action may include suspension or revocation of a member's access, suspension of resource discovery, or putting a member or site in some sort of quarantine. In extreme cases, unilateral action could even include suspension or termination of an entire federation.

Although, usage is monitored, and in some case limited (for example, a compute job limited to a certain time slot), unexpected behaviors might present themselves and be more noticeable to other users. Capabilities to enable third party reporting, such as abuse email addresses are tools that should be made available to federation members in case of deterioration of access due to other federation members use of shared resources.

4.8. Termination

While a Federation Operator would certainly have the authority to unilaterally terminate a federation they created, a federation may wish to define conditions or policies concerning an orderly termination, or even a panic termination. Since federation members may become dependent on federation resources, it is reasonable that there should be some commonly known understanding or policy that governs when those shared resources might become unavailable.

Members should have the right to leave a federation at any time. They could renounce their membership and withdraw any resources shared with the federation. If membership in a federation falls below a given threshold, this might trigger its termination. Similarly, if a federation is just not being used – if the members are too inactive – this could also trigger termination. For federations where accounting and billing is essential to maintain economic viability, termination might be triggered if the federation is failing to support itself. If a federation has simply fulfilled its purpose and is no longer needed, then it could be terminated.

These situations could be considered part of the natural lifecycle of a federation. If termination becomes inevitable, then notice should be provided to members. If there is any disagreement about the necessity to terminate, a dispute and resolution process could be defined to resolve the disagreement.

These scenarios all concern orderly terminate. Disorderly or panic terminations may also be necessary, as noted above concerning incident response. While such actions are undesirable, we must recognize their possibility as part of the reference architecture.

During the Federation Instantiation steps, “Federation Operating Practices and Policies” and “Community Dispute Resolution Process” documents might have been produced that should cover those terminations cases.

5. Deployment Models

In the preceding sections, we presented a reference architecture that identifies all necessary and possible functional components and their interactions for cloud federation and federation, in general. In doing so, we remained at the conceptual level (as much as possible) and did not examine or discuss implementation issues. In this section, we take a systematic look at federation deployment models, i.e., implementation approaches and trade-offs and how they affect simplicity, generality, performance, governance, trust relationships, and scalability. We also emphasize that we will be examining the spectrum of possible federation deployments – from the very simplest, bare-bones federations that could be quite useful yet need very little of the functionality identified in the Reference Architecture, to the most fully-functional, industrial-grade federations that could operate at a global scale.

We note that these federation deployments are inherently distributed. As such, these deployment models will inherit the fundamental properties and challenges of distributed computing systems. Different implementation approaches have different issues concerning data replication, data consistency, communication latency, the management of a federation in the presence of stale or incomplete data, fault tolerance, semantic interoperability, etc. We will not discuss here how these issues could be addressed, but rather will focus on identifying when they may be a concern.

The following deployment model diagrams are based on different deployments of Sites and Federation Managers (FMs). These deployment models will embody common and different fundamental properties of:

- Internal vs. external Federation Managers,
- Centralized vs. distributed deployments,
- Federation topology, and
- Infrastructure governance.

These basic deployment and individual federation instances have similar and significant governance requirements We subsequently discuss larger deployments, and conclude this section with a discussion of Auditor and Broker deployments.

5.1. Basic Site and Federation Manager (FM) Deployments

In the reference architecture, the FM is depicted as a single item, however in principle, its location and logical relationship to the federation partners is crucial to the deployment of the system. As such, we introduce the concept of internal vs. external Federation Managers (see Figure 5) in these basic deployment models. An internal FM is operated by a site that also participates in one or more federations that are hosted by the FM. An external FM is operated by a site that is not participating in any federations that are hosted by the FM. As such, these external FM Operators could be considered a Federation Provider since they provide a federation capability to a set of clients. As we shall discuss in Section 6, this distinction between internal and external, and the number of FM Operators, has direct implications concerning necessary trust relationships and governance.

The notions of centralized vs. distributed deployments and topology are also very important. Federations could be supported by a single FM in a centralized deployment. Distributed deployments could be supported by two or more FMs that exist in some communication topology. Centralized, single FM deployments will certainly be limited in their scalability, but will nonetheless be much simpler, and easier to deploy and operate (since they need not communicate with any other FMs). As such, they will serve the federation needs of a large segment of application domains. Larger deployments will require multiple FMs in some topology. While many graph structures are possible, for this discussion we will only address hierarchical and Peer-to-Peer (P2P) topologies since these represent fundamentally different topology classes and are likely to find practical use.

We begin by describing centralized deployments for both internal and external FM. We next discuss pair-wise deployment for both internal and external FMs, and also introduce hierarchical and P2P topologies. We then progress to larger internal and external hierarchical deployments. This is followed by larger P2P deployments. We conclude this sub-section by a brief discussion of mixed internal/external deployments.

For all models, we will not go into details on their expected functionalities, but here will list commonalities to be expected from such federation of cloud components, most of which have been discussed before.

- Security: Negotiation (for example Cipher, including reaching a minimal level of trust between parties), non-repudiation.
- Membership: Identity and Organization; registration, proof of membership, authentication mechanisms.

- Governance: including policies.
- Resources: data access but also specific access to compute, orchestration, specialized hardware.
- Telemetry: for Accounting and Billing, but also Auditing capabilities, incident reporting,
- Network: access to subset of resource, ingress and egress rules, separation of information.

5.1.1. Centralized FM Deployments

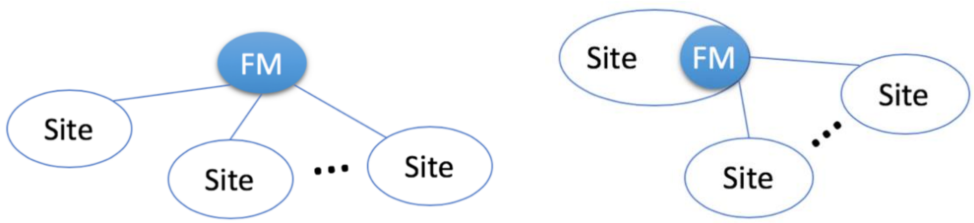


Figure 5. Centralized FM Deployments exhibiting external and internal FMs.

Centralized deployments have exactly one FM as shown in Figure 5. Figure 5 (left) is a single, external FM. This can also be called a centralized, third-party deployment, since the FM Operator is a third-party to the participating sites. Figure 5 (right) is a single, internal FM operated by one of the sites and participates in one or more federations with the other sites. A single FM interacts with all the Sites and must be done through a well-defined *FM-Site API.* This API provides access to information about the participants within the federation, and at the same time authorizes new members to join because of a pre-existing relationship of trust: either through a pre-seeding of cryptographic information to prospective members or exposition of the federation capabilities and manager information in a centralized location.

In the external FM case, all participating sites must trust the FM and its operator to manage the federations properly. In the internal case, the sites must also trust the FM operator, but the FM operator is one of the participants. From a practical perspective, this could be an important distinction.

5.1.2. Pair-wise FM Deployments

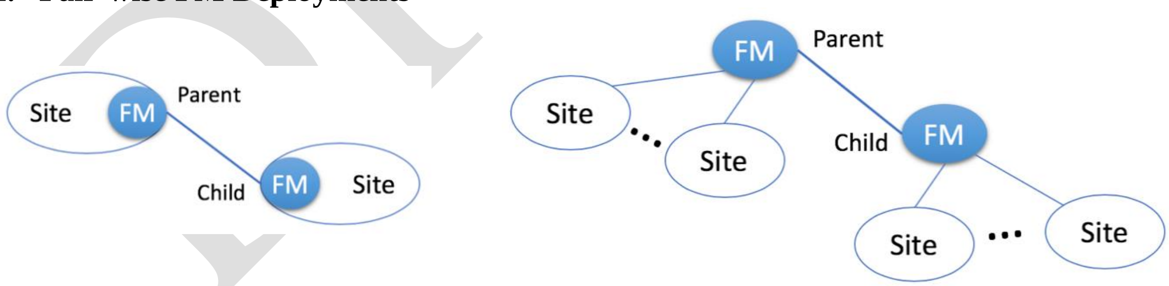


Figure 6. Pair-wise, Hierarchical FM Deployments.

Here we describe pair-wise FM deployments. Figure 6 illustrates pair-wise, hierarchical internal (left) and external (right) deployments. Here the two FMs exist in a *parent-child* relationship that can be utilized in governing the FMs and their federations. The parent FM Operator could define governance for the child FM Operator. Resource discovery and access policies could flow down from parent to child. Inheritance could be used to manage how this is done. A key distinction here is that with two FMs, they must also support a *Hierarchical FM-to-FM API* whereby the parent-child relationship can be established and used to manage resource discovery and access.

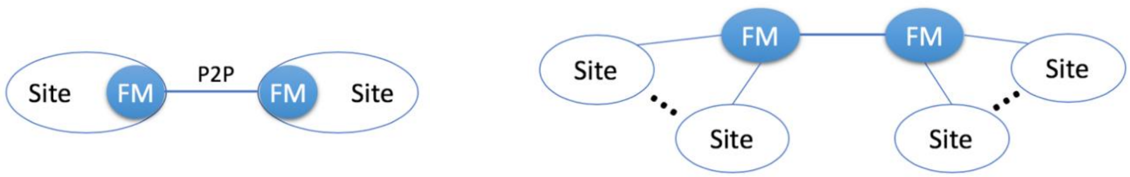


Figure 7. Pair-wise, P2P FM Deployments.

Figure 7 illustrates pair-wise, P2P internal (left) and external (right) deployments. Here the two FMs are obviously peers to one another. There is no graph property that can be used to define governance and federated resource management. However, a P2P approach could leverage existing concepts and tooling for defining a *P2P FM API* for building and operating P2P-based federated environments.

In this simplest, pair-wise deployment, the two Site Admins could manually configure their FMs to establish a *trust relationship* between the two sites and enable federation-related communication. Since this relationship is established using out-of-band knowledge, then there is no federation discovery or brokerage requirement. As a simple, informal federation, there may also be no requirement for any auditing or accounting functions. Going even further, if the two sites are very similar in function and business goals, the types of services each has to offer the other may be the same. In this model, the topology of communication is that of a distributed application architecture, where the peers are directly available to other peers, without the need for a central coordination by brokers. That is to say, there may be no requirement for resource discovery.

5.1.3. Larger FM Deployments

The deployment models shown above are the fundamental, base cases for centralized, hierarchical and P2P deployments. These can, however, be used in larger deployments. For illustrative purposes, Figure 8 and Figure 9 show larger deployments of internal and external hierarchical FMs, respectively. Figure 10 illustrates larger P2P deployments, with internal FMs on the left, and external FMs on the right.

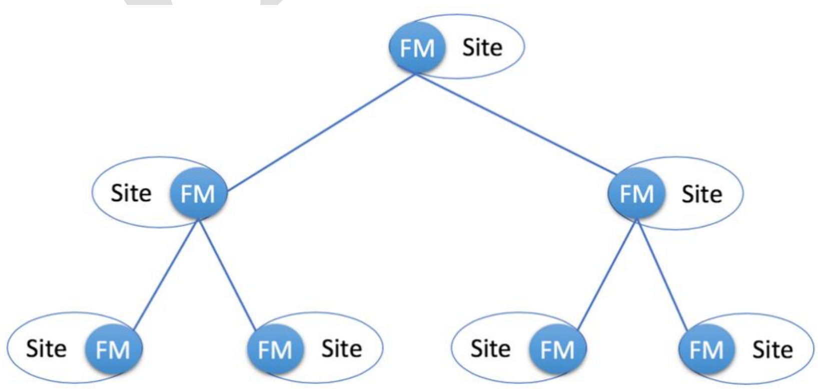


Figure 8. Larger Hierarchical Internal FM Deployments.

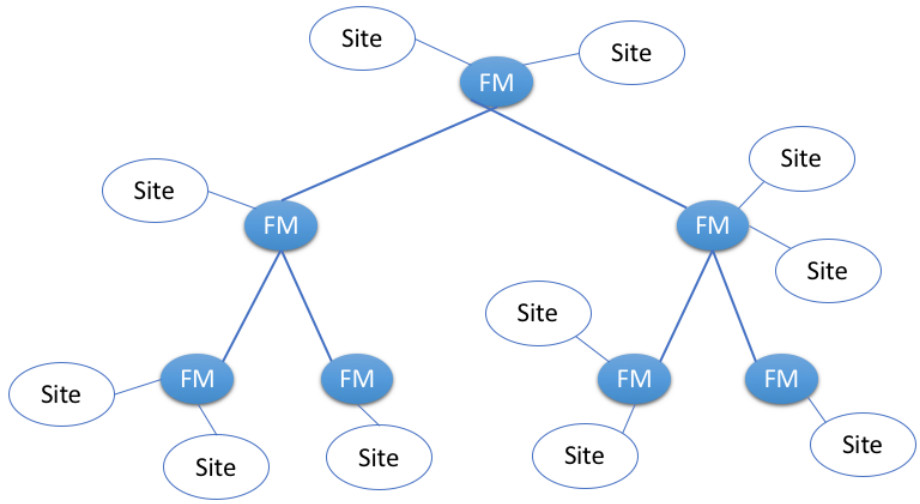


Figure 9. Larger Hierarchical External FM Deployments.

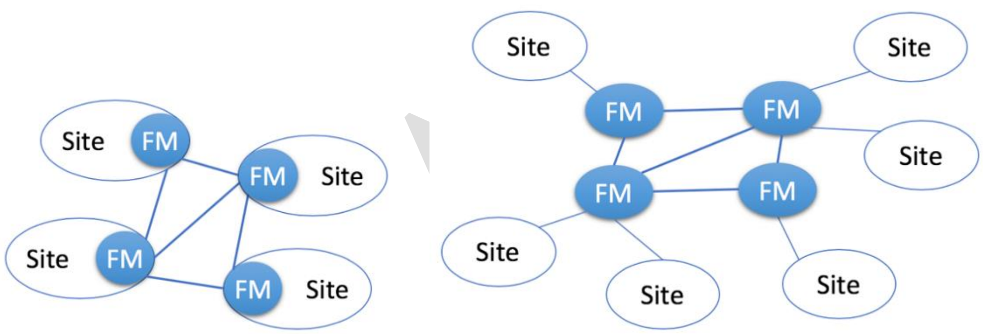


Figure 10. Larger P2P FM Deployments; Internal (left) and External (right).

5.1.4. Mixed Internal/External FM Deployments

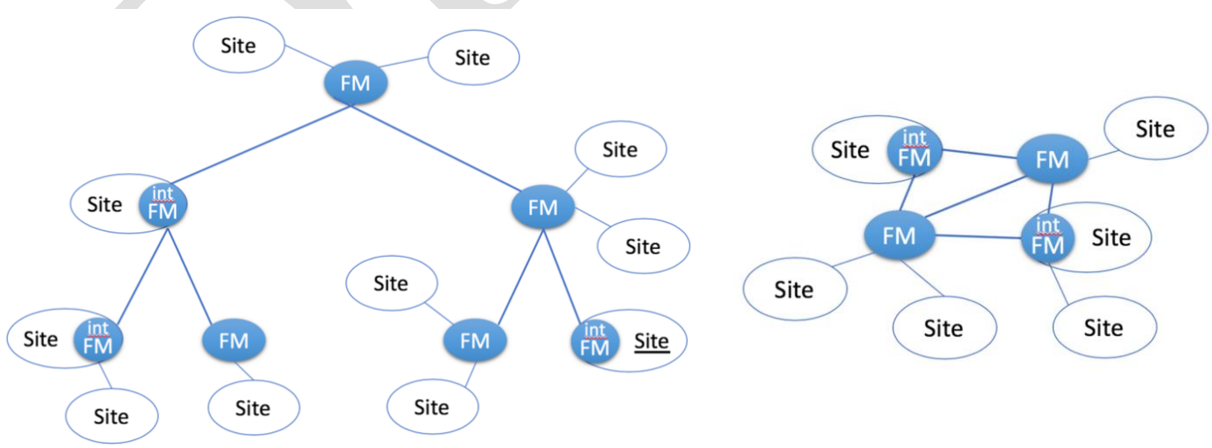


Figure 11. Mixed Internal/External FM Deployments.

While internal and external FMs have direct implications with regards to trust relationships between FMs and the Sites that operate or use them, we note here that there is no inherent reason why internal and external FMs cannot be mixed in the same deployments. This is illustrated in Figure 11. The implied trust relationships are different in a mixed deployment, but the different FMs nonetheless compatible.

5.2. Federation Auditor Deployments

The traditional audit function is an independent, third-party assessment of compliance to policies, contracts or other agreements among interested parties. Under this traditional arrangement, Federation Auditors would be separately deployed from any Sites or FMs. An independent Fed Auditor could be deployed as a single, centralized service, or as a distributed service – in essentially the same topologies described above in Section 5.1. In all cases, the Fed Auditors and FMs must establish each other's identity and the communication among them must be secure since the FAs will be requesting information that the FMs may consider is sensitive. In a distributed auditing service, the communication among the Fed Auditors must likewise be secure.

This traditional approach describes an *external* audit. We note, however, that *internal* audits are also possible. In many cases, an organization may wish to conduct an internal audit prior to any external audits. In the case of internal audits, it would be possible for the Fed Auditors to be co-located with a set of FMs – in essentially the same topologies described in Section 5.1. Because the different members of the federation might have different requirements or access level (including classification level), audit information must have different level of access and content. The common admin activity, data access and system event audit logs are to be recorded at different security and compliance levels to perform regulatory risk assessments. To do so, the use of an immutable log storage with access API is recommended. By identifying oneself with the log and audit server, a user is given an access limited to its access level and role in the federation. In all cases, the information that is considered auditable would have to be clearly understood by all parties. Such information would have to be collected and maintained according to audit requirements.

5.3. Federation Broker Deployments

In much the same way as Federation Auditors, Federation Brokers could be deployed in the same types of external communication topologies. The difference, of course, is that the Fed Brokers are providing a federation discovery service. As discussed above, the Fed Brokers would have to maintain a catalog of discoverable federation along with all necessary metadata about those federations. Whether this catalog and discovery service are centralized or distributed across a topology of Fed Broker servers is a deployment choice. If a set of FMs are being operated externally as a Federation Provider, then in principle, this same set of servers could host a set of Fed Brokers. Clearly if this Federation Provider is operating a large number of federations, then it might want to offer a discovery service for these federations. On the other hand, it is also possible that a Fed Broker service may be completely separate and independent, and attempt to catalog all possible known federations, regardless of who is operating them.

We also note that Fed Brokers could also be co-located with internal FMs. A set of internal FMs could also be hosting a set of federations that may wish to be discoverable. Of course, the scope of discoverable federations would typically be correlated to the number of FMs in a given federated infrastructure.

The federation broker can provide many components to facilitate access from potential federation members to one or more federation under its knowledge. Of note, knowledge of a given federation does not entail membership within a federation. Although the broker's role is one of central point for discovery service, it is also akin to simply a repository of information. For example, a federation broker can be as simple as being a web page with a repository of information of the metadata schema, location, policy and cryptographic requirements of existing federations. More complex federation brokers can act as the root in a deployment similar to a pair-wise hierarchical FM topology: each new child provides its information to the root node; as such the resource list of the federation broker organically increases. This model works in peer-to-peer models as well; new FMs added to the graph of connection are aware of the others, and using metadata propagation, the federation broker will, at some point in time, have a complete understanding of the existing graph of FMs. In hybrid situations, an alternative situation is for the federation broker to offer itself as a known FM tracker: when new FMs join, they contact the broker and offer it the information about the resources they are sharing. In this model, the growth of the broker is natural, but the broker needs to have a trust relationship with the federation owner.

Federation brokers become more powerful when they grow and are trusted by additional federation owners. The federation broker is then able to provide information about the topologies of the FMs in relationship to their federation and its resources. An analogy to this model can be that the stars in a constellation are the FMs, the constellation itself one of the Federation that is part of the known universe of federations as seen by the federation broker and its users.

Information propagation is key to keeping resource information pertinent for the federation broker users. In the case that the federation broker has a trust relationship with the federation owner and is able to query the FMs, periodically the broker should probe FMs for updates. FMs metadata should be cryptographically signed to prevent content spoofing, available at a known persistent location, and have a tag information available to allow differentiation from version to version, enabling the federation broker to update its content securely. Furthermore, if the FM is able, it can push its modification to the federation broker service.

Communication of metadata between FMs under a given federation follows a common format. This is not ensured for communication of metadata between disparate federations. The federation broker's role as such is additionally harder, and requires it to transform metadata information from federation A to federation B. In such case, the broker might need to provide additional API compatibility layers between federation A and federation B. It is in the interest of the FSP of each federation to publish their API, so that mechanisms can be written to support the use of the resources from different federations. This is a complex technical problem, beyond the scope of this discourse, but some entities are working to enable this support, one cloud at a time. Often, the first level of access is done using Federated Identities using a single set of credentials using one of the three major protocols for federated identity (OpenID, SAML and OAuth).

As a final comment, it would be also possible for Fed Brokers to also act as a security gateway to the FMs themselves. The Fed Broker service could vet a user or site that is searching for a given federation, according to a candidate membership policy defined by the FMs hosting the given federation. Whether it would be advantageous for FMs to delegate this responsibility to the Fed Brokers is an issue that will be resolved with further experience.

6. Deployment Governance: Requirements and Options.

In all deployment models, two or more entities wish to interact. This desired interaction carries a number of important implications concerning *trust* and *governance*. In Section 4, considerable discussion was devoted to the trust and governance of individual federations. This trust and governance directly depend on the trust and governance among the Sites and FMs themselves. This section addresses this issue.

6.1. Trust Federations

Any federation will be comprised of two or more Sites and will be hosted by one or more FMs. Any such set of Sites and FMs that interoperate to support application-level federations will be called a *Trust Federation* since these Sites and FMs must have established trust relations. The identity of each Site and FM must be well-known and trusted by those it interacts with. Admitting a malicious entity into a federation must be avoided.

We note that the FMs are responsible for one static, fixed function: faithfully providing the component functions of a Federation Manager, as described in Section 3. Once these functions are available in a trusted environment, any number of application-level federations with arbitrary functionality can be realized. A key question, then, is how to establish a Trust Federation? The following section examine the issues of how to “boot” a Trust Federation and admit new Sites and FMs.

6.2. Establishing Trust Federations

When creating a Trust Federation, any one Site or Federation Provider can deploy a single FM that could be considered *ab initio* a Trust Federation of one. Clearly though, to be useful, additional Sites and FMs must be added. When on-boarding a new Site or FM, we can say without loss of generality that one entity is part of the established trust and the other entity is the potential new-comer. On-boarding a new Site or FM essentially requires establishing a *trust relationship*. For any specific Trust Federation, the specific criteria for establishing trust may vary. This will be discussed shortly.

As per the deployment models in Section 5.1, FMs can be internal or external. Hence, the deployment models can be characterized by their *Site-to-FM* or *FM-to-FM* trust relationships:

Deployment Model	Type of Trust Relationships
Pair-wise	All of these deployment models have FM-to-FM trust relationships since each Site is operating their own FM.
Internal Hierarchical	
Internal Peer-to-Peer	
Centralized, Third-Party	Since there is only one FM in this deployment model, all trust relationships are Site-to-FM.
External Hierarchical	Since all FMs are external to the Sites, there are Site-to-FM trust relationships. However, since there are multiple FMs, there are also FM-to-FM trust relationships.
External Peer-to-Peer	

Table 1: Deployment Models and Trust Relationships

When adding a new Site or FM, we do not want to admit any malicious entities. Hence, there must be some process and policies for vetting and admitting new Sites and FMs. Likewise, some entity must have the authorization to conduct the vetting process and grant or deny admission. This entity can be called the *Trust Federation Administrator*. We will examine this for Site-to-FM and FM-to-FM trust relationships.

6.2.1. On-boarding New Site Members – Establishing Site-to-FM Trust

When establishing Site-to-FM trust, it will generally be the case that the FM is part of an established trust and the Site is requesting access as a member. Without loss of generality, we can say that this Site will communicate with the FM through some type of *Federation Site Client* that understands the necessary and compatible federation APIs and protocols. This client will be managed by some type of *Federation Site Administrator*. This Site Administrator will not be responsible for any particular federation, but rather just for the operation of the client itself.

The new Site Client and Administrator must be fundamentally trusted by the FM, and vice-versa. This trust would be established by:

- Use of acceptable federation tooling, i.e., a compatible Site Client that secures the communication between the Site and FM,
- Proper configuration and management of that tooling,
- The Site Administrator has been vetted to the FM, and
- The FM has been vetted to the Site Administrator.

What type and degree of identity proofing (vetting) and on-boarding issues that must be addressed could vary from one trust to another? Generally speaking, the Site may have to demonstrate they have a genuine need to join the trust, or bring resources needed by other members to the federation. Agreement on policies, communication models and negotiations of minimum level of services are part of this step. The FM could require an audit of the Site to verify that the client being run is an acceptable version, has the right patches, and the Site Administrator has the right process in place to ensure that the client stays up-to-date.

While on-boarding may commonly be focused on vetting the Site (and Site Administrator) to the FM, we note that the Site may also need to validate the identity of the FM. In much the same way that browsers validate the identify of a website, Sites could use *extended validation certificates* to validate the identity of an FM. Because vetting is important, the use of cryptographic signatures is recommended to ensure authentication and integrity of data exchanged between the parties joining the federation.

Finally, the issue of who has authorization to admit a new Site to an existing FM trust is discussed. While it may be common for the admission of a new Site to be managed by the FM it will directly interact with, this decision may be made by other entities in the trust. A trust may have one centralized authority or administrator that makes admission decisions for the entire trust. A middle-ground option is that a specific subset of FMs (and their administrators) have the authorization to admit new Sites. The opposite extreme is to give every FM the authorization to admit new Sites.

6.2.2. On-Boarding New Federation Managers – Establishing FM-to-FM Trust

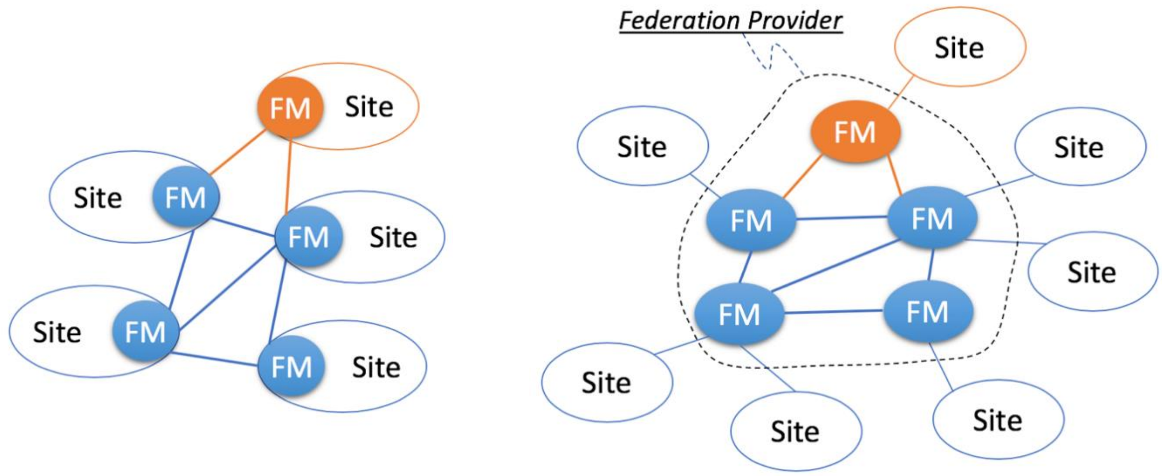


Figure 12. On-boarding a new FM.

When establishing FM-to-FM trust, many of the same issues will exist. In many cases, an FM will be associated with a single Site that is joining an established trust, as illustrated in Figure 12 (left). In general, the new Site FM must satisfy all the requirements to be considered trustworthy by the other FMs. The exact criteria that define trustworthiness may vary among different infrastructures. The two FMs that will interact may wish to begin by verifying each other's identity. They must then verify that they support the same intra-FM APIs and protocols, and that the communication between them is secured. Both sides may also wish to verify that the opposite FM is being maintained and operated in an acceptable manner.

When all FMs are external and operated by the same FM Operator, as illustrated in Figure 12 (right), the FM Operator can be called a *Federation Provider*. In this case, the deployment trust issues become much simpler. The Federation Provider can ensure that all configuration and trust issues are addressed when adding a new FM. We note that a Federation Provider may wish to add a new FM to enable a new Site to join a federation.

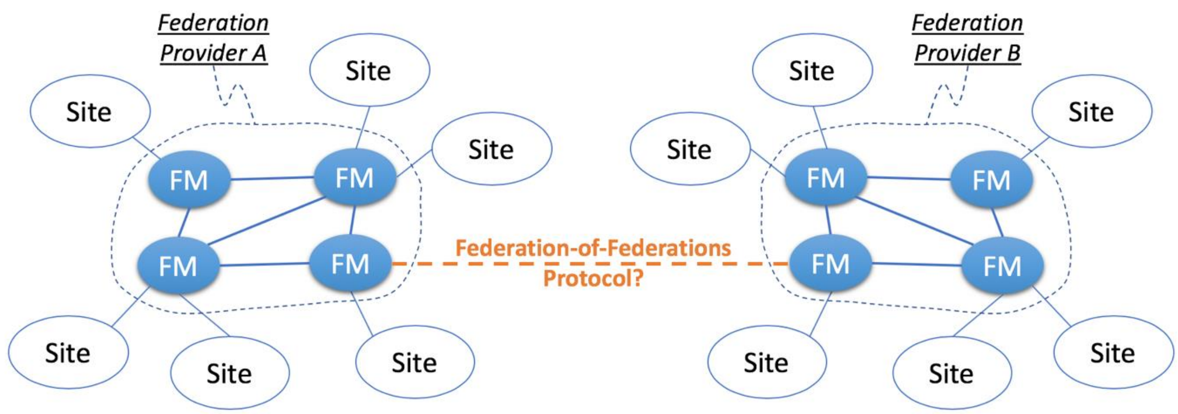


Figure 13. A Federation of Federations.

In other cases, each FM may be associated with a separate, established trust. In this case, this could be considered a *federation merger* or *federation of federations* as illustrated in Figure 13. Each FM could be operated by different Federation Providers. As such, these FPs could agree to *peer* to one another through these FMs. In addition, there could be a hierarchical relationship between the two FPs, or the FPs could agree to peer for only certain types of federations. The exact constraints would be defined by the business goals of the FPs.

6.3. Transitivity and Delegation of Trust

While these cases have been described as establishing a *pair-wise* trust relationship between two FMs, they will more often occur between sets of FMs, i.e., federation trusts. Clearly there will be issues of the *transitivity of trust* and the *delegation of trust*. If FM_A in $Trust_A$ establishes a pair-wise trust relationship with FM_B in $Trust_B$, will all other FMs in $Trust_A$ trust FM_B?

This is another fundamental governance issue. If a single FP is simply adding an FM to their existing set of FMs, then the on-boarding process can be relatively simple since this is essentially one trust environment. However, if two trusts are being bridged, then the transitivity and delegation issues must be addressed.

If trust is not completely transitive, then each FM in $Trust_A$ will have to establish their own trust relationship with FM_B for it to be admitted to $Trust_{A.}$ While such admission by consensus may be desirable in some deployments, it will quickly become unsustainable. To avoid such unsustainable scalability issues, *Trust Federation Administrators* will have to *delegate the authorization* to establish trust relationships with new Sites and FMs to a smaller set of FAs.

At one extreme, there is exactly one entity – a Trust Federation Administrator -- that has the authorization to establish a trust relationship with another trust through a specific FM. This requires that trust is completely transitive – every $FM_A \in Trust_A$ must trust the newly admitted FM. This represents another scalability issue since this one FA may become a bottleneck and requires complete transitivity. At the other extreme, we have the admission by consensus case noted above. A middle ground is to authorize a small set of FMs that have the authorization to establish trust relationships with external Trusts and FMs. This addresses the scalability issue of a single point of authorization, while reducing the required degree of trust transitivity. Such an arrangement may also enable a trust topology to be used as part of the governance model.

6.4. Federations and Trust Federations at Scale

Up to this point, we have made the implicit assumption that all users, Sites, FMs, federations and trust federations will operate in a well-known, deterministic way, however in practical deployments, this will not be the case at all times. Federation deployments will inherit all aspects and challenges of general distributed computing. As the scale of a federation increases, having perfect information about the entire federation at any given point in time will become increasingly difficult, and ultimately impossible. At some point, federation systems will have to cope with using stale or incomplete information in the management of federations and trusts.

Clearly the typical methods developed for distributed computing could be applied here, e.g., replication, caching, pipelining, estimation, etc. There will also be reliability and fault tolerance issues. Concepts for network protocols could be relevant here, e.g., the use of alternate routing and soft state. When the number of services available in any given federation becomes too large to manage in a single catalog, that catalog could be distributed. When that distributed catalog becomes large enough, the use of something like a WWW search engine might be useful to find services of a desired type.

Federation brokers, as well, are tools to help with the discovery and cataloguing of the known elements and resources of their known federations. Those will, as well, need to update their information at interval to be able to contain relevant resource information. Allowing vetted FMs to list at a known endpoint their metadata update will enable the propagation to take place by periodic pulls. Many adaptive data propagation algorithms have been used in networking and database solutions palliate this staleness problems.

Aside from using established distributed computing techniques to deal with large environments, another possibility is to use more distributed governance models. As mentioned above, more distributed methods for delegating managing functions to more Sites and FMs could be used (which brings in the transitive trust issues). This could include the use of *Friend-of-Friend* relationships to essentially establish *Webs of Trust*. Such social trust mechanisms could also include *rating* and *reputation systems*.

Such mechanisms, of course, achieve scalability by allowing error (or malicious actors) to creep into the system. The Byzantine Generals Problem captures the extreme of this condition: a set of Byzantine generals are planning an attack and every general does not trust messages sent by the other generals. How does a general determine where the truth lies and successfully plan the attack? This kind of *establishing trust in an otherwise untrusted world* can be done by the use of *distributed consensus* methods, e.g., *blockchain*. Blockchain[26] is an algorithmic means to agree on the state of a system, even when there is no pre-existing trust between parties. It relies on multiple trusted arbiter to validate its history and determine its next state, such that the starting state and the history of states prove the current state. This process relies on the use of a distributed ledger; this ledger is decentralized, peer-to-peer, synchronized through consensus, and tamper evident and resistant. Blockchains store their information in "chained" "blocks": transactions are recorded in a sequence of blocks. Blocks are cryptographically chained together using a hash chain, such that 1) a change in "Block YY" will prevent the hash validation of "Block YY+1", as such breaking the chain and providing tamper evidence 2) a broadly distributed chain provides a strong mean of validation, providing tamper resistance. Blockchain technology utilizes proven computer science mechanisms and cryptographic primitives (cryptographic hash functions, asymmetric-key cryptography, digital signatures) with append-only ledgers for record keeping. While blockchain methods may have their own scalability issues, their use in inherently distributed, federated environment is directly relevant.

7. A Catalog of Deployment Properties

We have presented the CFRA and the associated federation governance models. We have also examined the possible deployment and governance models of the CFRA actors to support application-level and organizational-level federations. As illustrated by Figure 14, these federation deployments can range from very simple, bare-bones deployments that are manually managed with informal agreements, to very large-scale federations that provide a full set of accounting and auditing services, along legal agreements concerning federation membership.

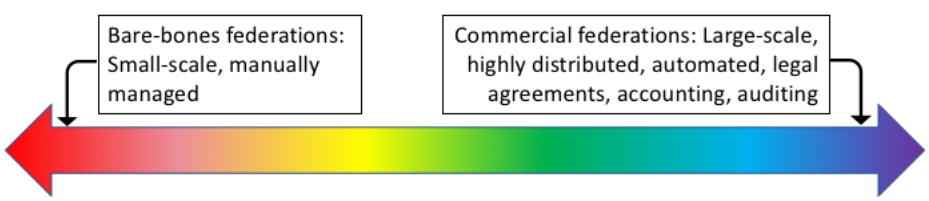


Figure 14. A Spectrum of Deployment Properties and Options.

In this section, we catalog this range of deployment properties and their options. Many of these deployment issues are optional, in that some deployments could assume and rely on many factors being known previously or simply not needed. Here we catalog these options to identify deployment options that could be chosen by various application domains. We note that these deployment properties can be broadly partitioned into the areas of *Deployment/Scale* and *Governance*. The Governance area is by far where most of the simplifying options can be found.

- Deployment/Scale
  - *Internal vs. External FMs:* Having a small set of internal FMs in a manually managed federation is certainly simpler than having a large set of external FMs. The trust relationships are easier to manage and less extensive.
  - *Centralized vs. Distributed FMs:* Having one centralized FM is certainly simpler than having a large number of FMs that effectively operate as a large distributed FM.
  - *Simple vs. Large/Arbitrary Communication Topologies:* Simple, pair-wise, or point-to-point federation topologies that are manually managed are certainly simpler than large, essentially arbitrary topologies that may be built-up from many disparate sites that wish to join a federation.
  - *Homogeneous vs. Heterogeneous Deployments*: Deployments can be significantly simpler if the same code is deployed everywhere. However, only relatively small deployments will be able to have this luxury. The larger a deployment that encompasses more disparate organizations becomes, the more probable it becomes that the deployment will involve heterogeneous FM implementations.
- Governance
  - *Implicit vs. Explicit Trust Relationships:* Whenever two or more FMs interact, there is either an implicit or explicit trust relationship. This trust can be implicit if the FM Operators know each other through informal or pre-established methods. However, as federations grow in scale, these informal methods will become impractical and ore formal methods will have to be used for establishing trust. (See the brief discussion of [10] in Section 8.)
  - *Vetting/On-Boarding New FMs:* Vetting a new FM for inclusion in a set of trusted FMs can also be done through informal methods. This is tantamount to establishing a trust relationship. Specifically, this could involve determining that the FM is the correct version, is configured properly, and has all the necessary patches.
  - *Federated Identity:* There must be some way of establishing identity within the context of a federation. As discussed in Section 3.7.1, this could involve mapping between arbitrary types of identity credentials, or mapping to a separate federated identity. If the federation relies on the same identity credentials being used everywhere, then the deployment and governance would be greatly simplified.
  - *Roles/Attributes:* All federation must have some set of roles or attributes whose semantics is commonly known. Smaller federations that have a relatively small, fixed set of roles or attributes can establish this common understanding through informal methods. Larger federations, however, may need a more formal or automated way of establishing this common understanding. This could involve establishing ontologies or mappings of the role/attribute namespaces among sites.

- *Resource Discovery:* If the services being managed in a federation are a relatively small, static set of services (such as basic cloud infrastructure services), these could be established informally. In a general federation where any number of application-level services may need to be managed, there would need to be a more compete resource cataloging and discovery services.
- *Resource Discovery Policies:* Again, if a relatively small, static set of services is being used with a set of commonly known roles or attributes, then the resource discovery policies associated with those resources could be relatively static and established informally. More general federations could make use of a policy language and policy engines to enforce discovery policies.
- *Resource Access Policies:* As a recurring option, if the resources being accessed is a relatively small, static set, then a common understanding of their access policies could be established by pre-established methods. However, as the resources being managed and their access policies become more general, more automated methods of defining and disseminating jointly agreed-upon access policies will be needed.
- *New Federation Member Vetting/On-Boarding:* Once a trust federation has been established and a specific federation has been created, there must be a way to vet and on-board new federation members. Establishing the true identity and need-to-know for a potential federation member could be an informal process. In other application domains, more formal processes may be needed. (See the brief discussion of [11] in Section 8.) Becoming a federation member may involve some agreement to follow the rules and support the overall goals of the federation.
- *Accounting/Auditing:* Small, informal federations will seldom need accounting and auditing functions. Any exchange of value may not need to be quantified by accounting, and compliance to policies or agreements may not need to be verified by auditing. As federations become larger and more formal, such practices will be needed. Accounting and auditing approaches will have their own range of implementations.
- *Federation Discovery:* Finally, the existence of many federations will be disseminated by out-of-band methods. This will be especially true when the federations are smaller, and the members can adequately manage the federation. However, as federations become larger and more numerous, they may wish to make their existence discoverable by potential new members. Hence, federations may wish to register with a federation discovery service that potential new members can use.

These deployment and governance properties can be used to compare different federation deployments. Further experience will determine which options are the most common and widely used across application domains.

8. Relevant Existing Tools and Standards

The goal of this section is to identify current IT standards that are directly relevant and not to provide an extensive review. The *Federated Cloud Engineering Report* [12] produced as part of the Open Geospatial Consortium's Testbed-14 contains more of a survey, along with discussion of how the systems, tools and standards covered there relate to the Cloud Federation Reference

Architecture presented here. Additional comparative discussion can be found in [7, 13, 14]. For purpose of identification, relevant standards can be categorized as follows:

- *Securing the communication:* These standards are relevant to all distributed systems, which includes federated systems. That is to say, the communication among members, sites and FMs must be secured against all possible malicious efforts. Relevant standards include:
    - SSL/TLS
    - HMAC
- *Identity, Authorization, Policy:* Identity is established by issuing a credential that can be associated with one or more authorization attributes. Discovery and access policies can be defined over these identity and authorization attributes. Relevant standards include:
    - Account name and password
    - Public Key Infrastructure (PKI) and PKI Proxy Certs
    - Kerberos
    - Shibboleth
    - Grid Security Infrastructure (GSI)
    - SAML and XACML
    - OpenID, OAuth, and OpenID Connect
    - UMA
- *Catalogs and Discovery:* Cataloging and discovery services are an integral part of all distributed systems, including federations. Relevant standards include:
    - Lightweight Directory Access Protocol (LDAP)
    - Active Directory and Active Directory Federation Services
    - Web Service API Gateways
    - DNS/DNSSEC
    - OWL-S
- *Trust and Governance:* While much trust and governance may be established out-of-band, we recognize that there are tools for establishing trust in an otherwise untrusted environment that relevant for federated systems. Relevant tools include:
    - Blockchain
    - Consensus Algorithms, e.g., Proof-of-Work, Raft, PAXOS

We note that FICAM (the Federal Identity, Credential and Access Management Architecture) [10] covers a number of USG federal policies, standards, and guidance concerning all of the above topics. This includes guidance as defined in the NIST *Digital Identity Guidelines* [11] for *Identity Assurance Levels*, *Authenticator Assurance Levels*, and even *Federation Assurance Levels*. Notably the Federation Assurance Levels define the strength of assertions made between an IdP and a Relying Party in a federated environment. A more complete discussion of this topic is out-of-scope for the current document. Additional NIST guidance is available for security and privacy controls [15], and managing Personally Identifiable Information (PII) [16]. When deploying a federation infrastructure or instantiating a federation, the stakeholders should decide which concerns are relevant or necessary.

9. Areas of Possible/Needed Federation-Specific Standards

In developing the NIST Cloud Federation Reference Architecture, we have developed a conceptual model of general federation. In doing so, we have identified the fundamental actors and their interactions. While we've reviewed a number of existing standards and tools that are relevant to these general federation functions, additional federation-specific standards are needed to make federations truly general and easy to use.

9.1. Federation Manager Protocols and API Standards

A critical part of the NIST Cloud Federation Reference Architecture is clearly the Federation Manager. This is the entity that manages all the pre-established relationships, i.e., the *virtual administration domain*, among federation members. How FMs interact with Users, Sites, Admins, and other FMs is a definite area of standardization. Each of these entities could define a segment of the overall FM API:

- *FM Admin API:* When an FM is booted, there will be an owner and an administrator for it. This administrator will have the authorization to manage how the FM is configured and operated. This administrator will have the authorization for creating new federation instances. When a new federation is instantiated, the FM administrator has the authorization to create the first member who will be the Federation Administrator.
- *FM Federation Admin API:* Each instantiated federation will have at least one admin that can grant/revoke federation membership and roles/attributes.
- *FM-Site Admin API:* In some governance models, there will be a *Federation Site Admin* that will have the authorization register service endpoints for specific federations. There may also be a federation-specific discovery policy associated with a service endpoint.
- *FM-User API:* An ordinary user that is a federation member must be able to authenticate to an FM for a specific federation. Upon successful authentication, the user must be able to discover and invoke the services that they are authorized to use, in some capacity, within the context of that federation.
- *FM-FM API:* In centralized deployment, a single FM must only communicate with member Sites and Users. This greatly simplifies their API. In larger deployments, multiple FMs must clearly communicate among themselves through an FM-to-FM API. This API must enable FMs to exchange information about specific federations, e.g., which services are being made available, what their discovery policies are, current site members, etc.

  If FMs exist in a known graph topology, then the API should reflect this fact. In a hierarchical deployment, the API should clearly enable parent-child relationships to be utilized. In a P2P deployment, communicating with your nearest neighbors to eventually acquire all relevant information about a federation must be supported. Also, as a distributed system, such APIs should support operation in those environments, e.g., have support for fault-tolerance, achieving information consistency as quickly as possible, etc.

We note that these APIs could have different protocol bindings. A RESTful protocol binding is a likely candidate, but others, such as gRPC, are possible.

9.2. Federation Definition Standards

As discussed in Section 4.1, it should be possible to define a standard format for describing or defining a specific type of federation instance. Such formal descriptions could be used to enable federation discovery through a federation broker and also federation provisioning through a commercial federation provider. To briefly review, a standard format could include:

- Resources to be shared and their metadata
- Roles & Attributes
- Resource Discovery

- Federation Membership
- Federation Member Identity Credentials
- Authorization to grant or revoke federation membership
- Authorization to grant or revoke member roles or attributes
- Governance, policies, SLAs
- Security considerations

Such a standard description format could be called a *Federation Markup Language*, e.g., *FedML*. This could be completely XML-based or have pre-defined semantics for the terms that are used. A JSON binding could also be possible whereby objects and lists could be used in the formal description of a federation.

In addition, it would also be possible to define an *ontology* for federations. An *OWL-Fed* could be built on top of the *Web Ontology Language*. In much the same way that *OWL-S* is an ontology for web services, OWL-Fed could be an ontology for federations. That is to say, an OWL-Fed would provide a machine-interpretable set of classes and properties of a federation. This would define how the federation operates and how users interact with it.

9.3. Federation Discovery and Provisioning

As noted above, a standard, formal definition of a federation would be the linchpin of federation discovery through a Federation Broker. The Broker would offer an API whereby Federation Owners could register their federation descriptions. The Broker API would also provide a query API whereby potential new members could search for relevant federation based on information made publicly available.

Likewise, commercial federation providers could use such formal description to define what types of federations they can instantiate and operate on behalf of their clients. One could envision a federation provider with a drop-down menu of supported federation types. Each federation type could have a set of configuration parameters. Upon instantiation, the federation would be tailored to the client's requirements.

The API for any such Federation Broker or Federation Provider would need to rely on formal federation descriptions. While these particular use case scenarios will take a while to materialize in the marketplace, the benefits of having a formal description method for federations is unambiguous.

10. Final Observations

In this Reference Architecture document, we have posited a conceptual actor model for general federation. By starting from the most general interpretation of what federation entails (Figure 2), we were able to identify the fundamental capabilities that must go into this model. These fundamental capabilities were integrated into, and used to augment, the existing NIST Cloud Computing Reference Architecture. From this conceptual actor model, it was straight-forward to identify a possible spectrum of deployment and governance models. It was also possible to identify a number of possible areas for federation-specific standardization.

In this document, however, we have only scratched the surface. Many of the concepts presented here need to be examined in much more depth. The possible areas of standardization have only been described in very general terms. Not all areas have been given equal attention. Federation Auditors, for example, need to be flushed-out with regards to formal terms of compliance, and how audits would actually be done. Much more experience and specifics are needed.

Additional areas have not even been touched. Are trust description languages or trust modeling ontologies possible? What relevant work has been done in these areas? Is it possible to do an audit of trust relationships? We must leave such questions for other documents.

References

[1] Mell, P. and T. Grance, The NIST Definition of Cloud Computing, Special Publication 800-145, September, 2011. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[2] NIST, NIST US Government Cloud Computing Technology Roadmap, Volume I: High Priority Requirements to Further USG Agency Cloud Computing Adoption, SP 500-293, November, 2011. http://www.nist.gov/itl/cloud/upload/SP_500_293_volumeI-2.pdf.

[3] Assis, M. Luiz F. Bittencourt, Rafael Tolosana-Calasanz, and Craig Lee, Cloud Federations: requirements, taxonomy, and architectures, chapter in Developing Interoperable and Federated Cloud Architecture, Kecskemeti, Kertesz, Nemeth, eds., IGI-Global, pp. 1–42, April 2016, DOI10.4018/978-1-5225-0153-4.ch001.

[4] Bittencourt, Luiz F., Rodrigo Calheiros, and Craig Lee, Guest Editors, Middleware for Multiclouds, IEEE Cloud Computing, July-August 2017.

[5] ISO/IEC 17789, Information technology — Cloud computing — Reference architecture. https://standards.iso.org/ittf/PubliclyAvailableStandards/c060545_ISO_IEC_17789_2014.zip

[6] J. Cummings et al. *Beyond Being There: A Blueprint for Advancing the Design, Development, and Evaluation of Virtual Organizations.* http: //www.ci.uchicago.edu/events/VirtOrg2008/VO report.pdf, 2008. Final report on NSF workshop Building Effective Virtual Organizations.

[7] Craig A. Lee, Marcio Assis, Luiz F. Bittencourt, Stefano Nativi, and Rafael Tolosana-Calasanz, Big Iron, Big Data, and Big Identity, Chapter in *New Frontiers in High Performance Computing and Big Data*, G. Fox, V. Getov, L. Grandinetti, G. Joubert, T. Sterling, eds., Advances in Parallel Computing, Vol. 30, pp. 139-160, IOS Press, Nov. 2017.

[8] Liu, F., et al., *The NIST Cloud Computing Reference Architecture*, NIST Special Publication 500-292, September, 2011.

[9] J. Kiljander *et al*., "Semantic Interoperability Architecture for Pervasive Computing and Internet of Things," in *IEEE Access*, vol. 2, pp. 856-873, 2014. doi: 10.1109/ACCESS.2014.2347992

[10] NIST, Federal Identity, Credential and Access Management Architecture, https://arch.idmanagement.gov

[11] Grassi, Paul A., Michael E. Garcia and James L. Fenton, *Digital Identity Guidelines*, NIST Special Publication 800-63-3, https://pages.nist.gov/800-63-3

[12] Lee, C.A., *The Federated Cloud Engineering Report*, the Open Geospatial Consortium, OGC 18-090r1, March 2019. http://docs.opengeospatial.org/per/18-090r1.pdf

[13] Lee, C., Cloud Federation Management and Beyond: Requirements, Relevant Standards, and Gaps, IEEE Cloud Computing, v3n1, pp. 42–49, Jan-Feb 2016, doi:10.1109/MCC.2016.15.

[14] Chadwick, D.W., K. Siu, C. Lee, Y. Fouillat, D. Germonville, Adding Federated Identity Management to OpenStack, Journal of Grid Computing, Volume 12, Issue 1, pp. 3-27, March 2014. Also published on-line: http://rd.springer.com/article/10.1007/s10723- 013-9283-2.

[15] NIST, *Security and Privacy Controls for Federal Information Systems and Organizations*, NIST Special Publication 800-53, Revision 4, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

[16] McCallister, E., Tim Grance and Karen Scarfone, *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)*, NIST Special Publication 800-122, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-122.pdf

[17] Cloud Security Alliance, *Security Guidance*, v4.0, 2017, https://downloads.cloudsecurityalliance.org/assets/research/security-guidance/security-guidance-v4-FINAL.pdf.

[18] W3C, *Identity Credentials 1.0,* Draft Community Group Report, 03 April 2018, https://opencreds.org/specs/source/identity-credentials.

[19] ISO/IEC 27729:2012, *Information and Documentation -- International Standard Name Identifier (ISNI)*, https://www.iso.org/standard/44292.html.

[20] Pross, B. and C. Stasch, eds., *The Workflows Engineering Report*, the Open Geospatial Consortium, OGC 17-029r1, http://docs.opengeospatial.org/per/17-029r1.pdf

[21] Object Management Group, Business Process Model and Notation, http://www.bpmn.org

[22] IETF, *The OAuth 2.0 Authorization Framework*, RFC 6749, October 2012.

[23] The OpenID Foundation, *Welcome to OpenID Connect*, https://openid.net/connect

[24] WS02, *The WS02 Integration Agile Platform*, https://wso2.com/platform

[25] OASIS, *eXtensible Access Control Markup Language (XACML) Version 3.0*, http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

[26] Yaga, D, Mell, P, Roby, N., Scarfone, K; NIST IR 8202 “Blockchain Technology Overview” https://doi.org/10.6028/NIST.IR.8202

Appendix A. Cloud Federation Terms and Definitions

Here we collect and succinctly define the cloud federation terms that have been used in the Cloud Federation Reference Architecture. Since the CFRA was derived from the NIST Cloud Computing Reference Architecture, all efforts were made to maintain consistency with that vocabulary. All attempts were also made to find existing definitions for terms from other documents. These sources are referenced.

Term	**Definition**	**Comments**
Administrative Domain	An organization wherein a uniform set of discovery, access and usage policies are enforced across a set of users and resources based on identity and authorization credentials meaningful within that organization.	A set of resources under a single set of administrative policies.
Asymmetric Federation	A federation in which some participating sites provide only users or resources, but not both.	Compare with Symmetric Federation.
Attribute	*Derived from [17]: An identity property. Such properties may be relatively static, e.g., personal name, or may be dynamically granted or revoked, e.g., project membership.* An attribute can be termed an *authorization attribute* since possessing an attribute can be associated with possessing authorization for a specific action.	
Business Support	*Source [8]:* The set of business-related services dealing with clients and supporting processes. It includes the components used to run business operations that are client-facing.	
Cloud Auditor	*Source [8]:* A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.	
Cloud Broker	*Source [8]:* An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.	

Cloud Carrier	*Source [8]:* An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.	
Cloud Computing	*Source [8]:* A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.	
Cloud Consumer/Customer	*Source [8]:* A person or organization that maintains a business relationship with, and uses services from, Cloud Providers.	
Cloud Federation	A Federation of Cloud Providers.	
Cloud Federation Broker	See *Federation Broker.*	
Cloud Provider	*Source [8]:* A person, organization, or entity responsible for making a service available to interested parties.	
Cloud Service	A service that can be provided on-demand by a Cloud Provider. Such services may be at the Infrastructure-, Platform- or Software-as-a-Service levels.	
Cloud Service Consumer	See *Cloud Consumer/Customer.*	
Cloud Service Management	*Source [8]:* All service-related functions that are necessary for the management and operation of those services required by or proposed to cloud consumers.	
Cloud Service Provider	See *Cloud Provider.*	
Data Resource Layer	*Derived from [17]:* All computing resources used to provide data.	
External Federation Manager	A Federation Manager that an organization is using to participate in a federation, but is not being operated by that organization.	
Federated Environment	See *Federation*.	

Federated Identity	An identity that is meaningful and trusted within a federation.	
Federated Identity Management	*Source [17]:* The process of asserting an identity across different systems or organizations. This is the key enabler of Single Sign On and also core to managing IAM in cloud computing.	
Federated Resource Access	The process and policies governing the access to federated resources by federation members.	
Federated Resource Discovery	The process of discovering federated resources.	
Federated Resources	Resources that are being made available by the federation members such that discovery and access can be managed as part of the federation.	
Federation	An organization of self-governing entities that have common policies, administrative controls, and enforcement abilities governing the use of shared resources among members. A virtual administrative domain wherein multiple participating organizations/sites can define, agree upon and enforce resource discovery, access and usage policies for the sharing of a subset of their resources.	Alternate Names: Federation Instance, Federated Environment, Virtual Administrative Domain.
Federation Administrator (Instance)	The entity that has the authorization to configure and operate a Federation Instance. This entity may be distributed depending on the governance model.	
Federation Auditor	An entity that can assess compliance for any type of policy associated with a federation. This entity maybe internal or independent third-party.	
Federation Broker	An entity that enables new members to discover existing federations based on attributes made known during the brokering process.	
Federation Broker Administrator	(The entity that has the authorization to configure and operate a Federation Broker.	

Federation Carrier	The entity that provides connectivity and transport (a) among federation members, or (b) between federation consumers and federation providers.	
Federation Carrier Administrator	The entity that has the authorization to configure and operate a Federation Carrier.	
Federation Discovery	The capability and process of making a federation findable (discoverable) by potential new members.	
Federation Governance	All policies and semantics involved in managing every step and phase in a federation's lifecycle to achieve the federation's purpose.	
Federation Instance	See *Federation*.	
Federation Instance Owner	The entity that initially creates a federation. When initially created, a federation may be considered *empty* or have exactly one member: the *Federation Administrator*. The Federation Owner and Administrator may be the same entity.	
Federation Manager	The entity that provides the essential federation management functions described in the CFRA for potentially multiple federations over their lifespans.	
Federation Operator	The entity that deploys, configures and maintains one or more Federation Managers.	A Federation Operator may be a site that operates its own internal FM to collaborate with a set of federation partners. (Compare with *Federation Provider*.)
Federation Policy	The practices that govern the functioning of a federation.	

Federation Provider	A Federation Operator that makes federation services available to a community of consumers.	While a Federation Provider could be a site that operates a single Federation Manager to provide federation services to a set of federation partners, a Federation Provider could also operate a set of Federation Managers to provide federation services (perhaps commercially) to a community of users, while not participating in any federations itself.
Federation Resource Catalog	A systematic compilation of the resources being made discoverable and available within a federation.	
Federation Resource Management	Governance through the use policies for the discovery, access and usage of resources within a federation.	
Federation Service Provider	Any system entity that operates and provides a resource that is a service to the federation	
Federation Site	A member organization that contributes resources to a federation.	
Federation Site Administrator	The entity that has the authorization to manage a site's contributed resources.	
Governance	The establishment of policies and enforcement of compliance by the members of a governing body.	Derived from businessdictionary.com
Identity Attribute	See *Attribute*.	

Identity Credentials	*Source [18]:* A set of claims made by an entity about an identity.	An identity is a collection of attributes about an entity that distinguish it from other entities. Entities are anything with distinct existence, such as people, organizations, concepts, or devices. Some entities, such as people, are multifaceted, having multiple identities that they present to the world. People are often able to establish trust by demonstrating that others have made valuable claims about their identities. One way of doing this is by presenting a credential. A credential is **a set of claims made by an entity about an identity**. A credential may refer to a qualification, achievement, quality, or other information about an identity such as a name, government ID, home address, or university degree that typically indicates suitability.
Identity Federation	A federation that is exclusively concerned with managing federated identities.	
Identity Provider (IdP)	*Derived from [17]:* The source of the identity credentials in an Administrative Domain. The identity provider isn't always the authoritative source, but can sometimes rely on the authoritative source, especially if it is a broker for the process.	
Inter-Cloud	A concept of connected cloud networks, including public, private, and hybrid clouds. It incorporates a number of technology efforts put together to improve interoperability and portability among cloud networks.	

Internal Federation Manager	A Federation Manager that an organization is using to participate in a federation, and is also being operated by that organization.	
Interoperability	*Source [19]:* The ability of two or more systems or applications to exchange information and to mutually use the information that has been exchanged.	
Multi-cloud	Provisioning cloud resources from multiple Cloud Providers.	
Physical Resource Layer	*Source [17]:* All physical resources used to provide cloud services, most notably, the hardware and the facility.	
Portability	The ability to move an object from one system to another without the loss of functionality.	
Provisioning/Configuration	*Source [8]:* Automatically deploying resources based on the requested services or capabilities.	
Regulatory Environment	The legal regulations and laws imposed by any level of government that the actors in an Administrative Domain must observe. A federation, i.e., a Virtual Administrative Domain, may need to reconcile all relevant regulatory environments .	
Relying Party (RP)	*Source [17]:* The system that relies on an identity assertion from an Identity Provider.	
Resource	Any physical or virtual component within a computer system the access and consumption of which must be managed.	
Resource Abstraction and Control Layer	*Source [17]:* Software elements, such as hypervisor, virtual machines, virtual data storage, and supporting software components, used to realize the infrastructure upon which a cloud service can be established.	
Resource Discovery Policy	The policy governing the ability to find of resources within a federation.	

Resource Owner	The entity that is accountable and authorizes use and governance of a resource.	
Resource Provider (RP)	Any system entity that operates and makes the resource available.	
Role	*Derived from [17]: An identity property. A role is generally granted or revoked, and is associated with a set of authorizations or capabilities that constitute that "role" within an organization or domain. As such, a role may be associated with a set of authorization attributes.*	
Service Owner	The entity that is accountable and authorizes use and governance of a resource that is a service.	
Service Provider (SP)	Any system entity that operates and provides a resource that is a service.	
Symmetric Federation	A federation in which participating sites provide both users and services.	Compare with Asymmetric Federation.
Trust	A risk-based decision to consider a request, presented by another entity (a party or a system) within a given context, to be valid.	In IT systems, trust can be considered to be a binary decision based on performing a cryptographic "handshake" that reduces risk to acceptable levels. Trust can also be based on reputation systems that deal with a wider range of trust.
Trust Delegation	Trusting another entity to perform or validate your request.	This is different than Entity B *impersonating* Entity A. Under delegation, Entity B is *authorized* to act for Entity A, and is *known* to do so.
Trust Federation	An organization that defines how trust relationships can be created, and can manage their lifecycle -- from establishment and maintenance to termination.	
Trust Federation Administrator	The entity that has the authorization to manage a Trust Federation.	This should be distinct from the governance management.

Trust Relationships	The trust that is established among multiple entities in specific context.	
Trust Transitivity	If Entity A trusts Entity B, and Entity B trusts Entity C, then Entity A trusts Entity C. Transitivity implies delegatability, but not vice versa.	
Virtual Administrative Domain	See *Federation*.	

Table 2: Cloud Federation Terms and Definitions.

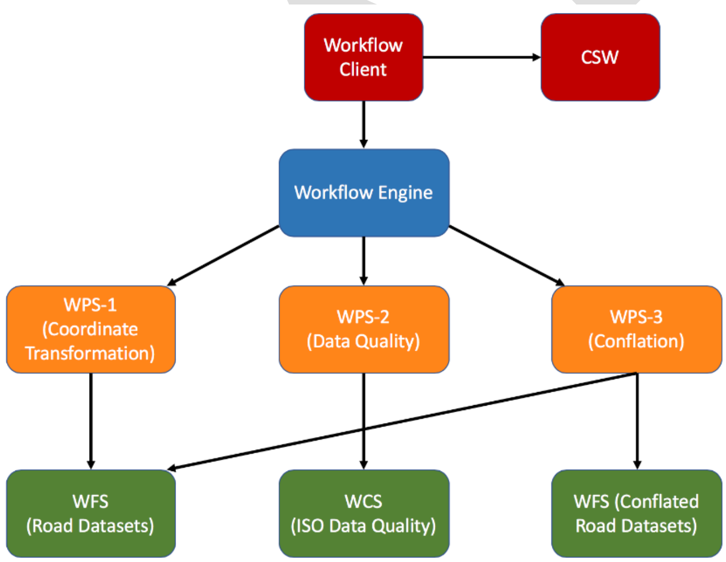
Appendix B. Example Use Cases

The Reference Architecture is, by its nature, conceptual. Its goal is to organize the entire design space for possible federation tools. As we have noted above, there are a number of possible deployment and governance models that affect how federation tooling can be implemented. The goal of this appendix is to show how the Reference Architecture can be mapped to something that is more concrete and implementable.

Many use case examples have been considered, including (a) scientific data sharing, (b) scientific computing sharing, (c) governmental public safety, (d) governmental disaster response, and (e) business supply chain management. All of these involve data sharing in one form or another. To be more specific, multiple stakeholders have discussed the need to execute workflows (a controlled sequence of operations) that must access data from different repositories that are owned by different organizations. The following example examines this use case in more detail.

B.1. The Conflated Road Dataset Workflow

The Open Geospatial Consortium (OGC) has investigated the use of workflows for geospatial applications. The OGC Testbed-13 *Workflows Engineering Report* [20] examines currently available workflow management tools, along with access control issues for the individual workflow services. This report uses the *Road Dataset Conflation* workflow as a test case.

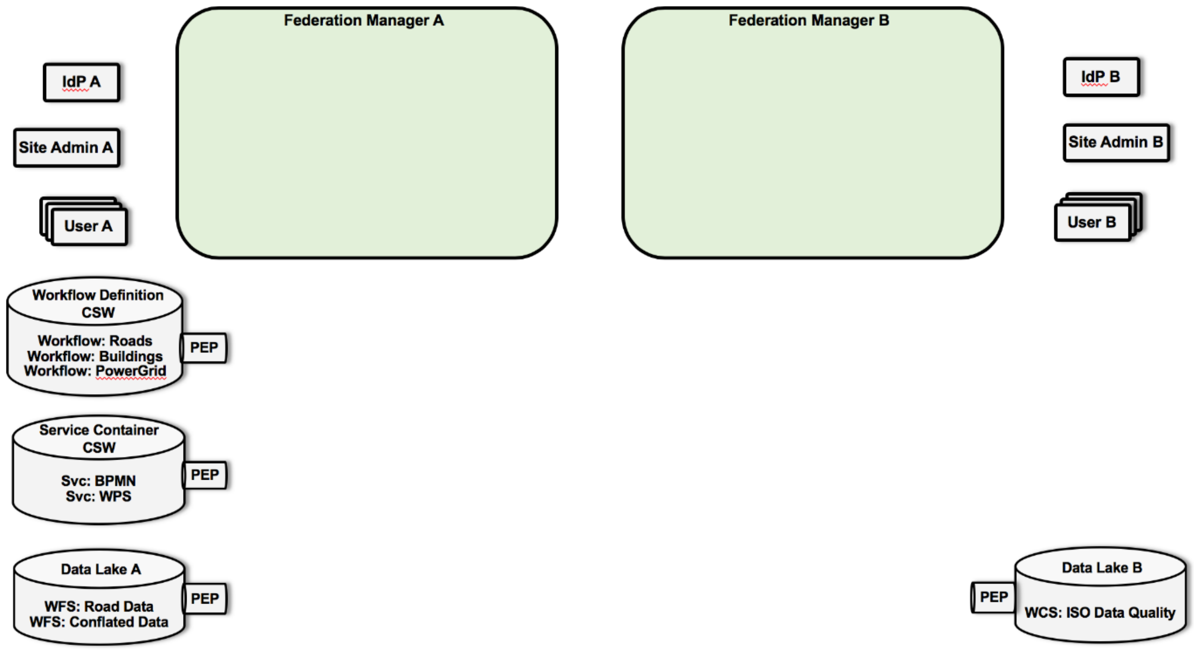


Appendix B.1 Figure 1. The Road Dataset Conflation Workflow.

This workflow is illustrated in Appendix B.1 Figure 1 (This is Figure 1 from [20] redrawn.) It leverages several standard OGC geospatial services. They are the *Catalog Service for the Web (CSW),* the *Web Processing Service (WPS),* the *Web Feature Service (WFS),* and the *Web Coverage Service (WCS).* As the names imply, CSW is an object catalog service and the WPS manages the execution of other services. The WFS serves map features, i.e., icons and other symbology that can be geolocated on a map. The WCS serves map coverages, i.e., raster data that covers an area on a map.

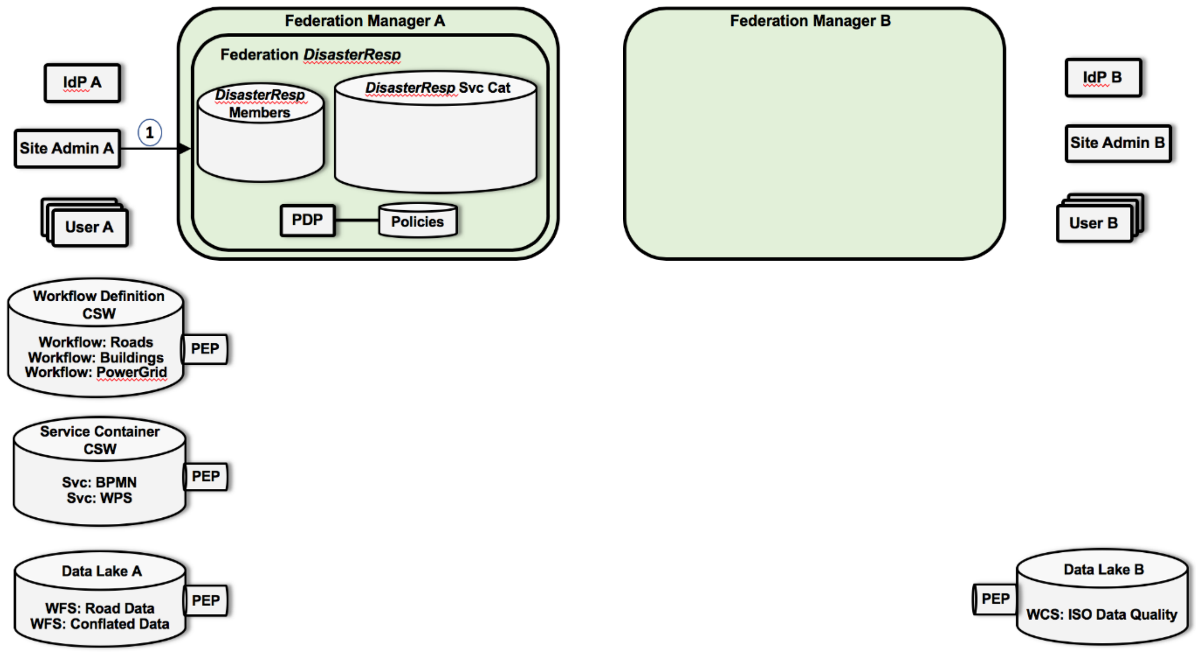
The example begins with the Workflow Client retrieving the workflow definition from a CSW. This definition is passed to a Workflow Engine which instantiates the workflow elements. This is a sequence of three WPSs. To start the workflow, the Client passes in parameters that identify the map region of interest and the Road Datasets to be used. The first workflow step, WPS-1, retrieves these target data sets from a WFS and performs any necessary coordinate transformations to ensure that all datasets of interest are in the same format. A reference to the target data then passes back through the Workflow Engine to WPS-2. WPS-2 contacts a separate WCS to determine the data's quality. For the purposes of the example, quality entails the positional accuracy of the data and any road discrepancies among the data sets. If the quality is insufficient, the workflow will be terminated. If the quality is sufficient, then the data references are passed to WPS-3. WPS-3 retrieves the road datasets and conflates them into one, merged dataset that is written back to the WFS. A reference to the final data product is returned to the Workflow Engine and the Client.

To cast this example into a federated environment, we will assume a specific deployment and governance model. We present this use case as two organizations that each operate their own *Data Lake*, i.e., a data repository, along with their own *Federation Manager*, in an *internal, pairwise P2P* deployment.



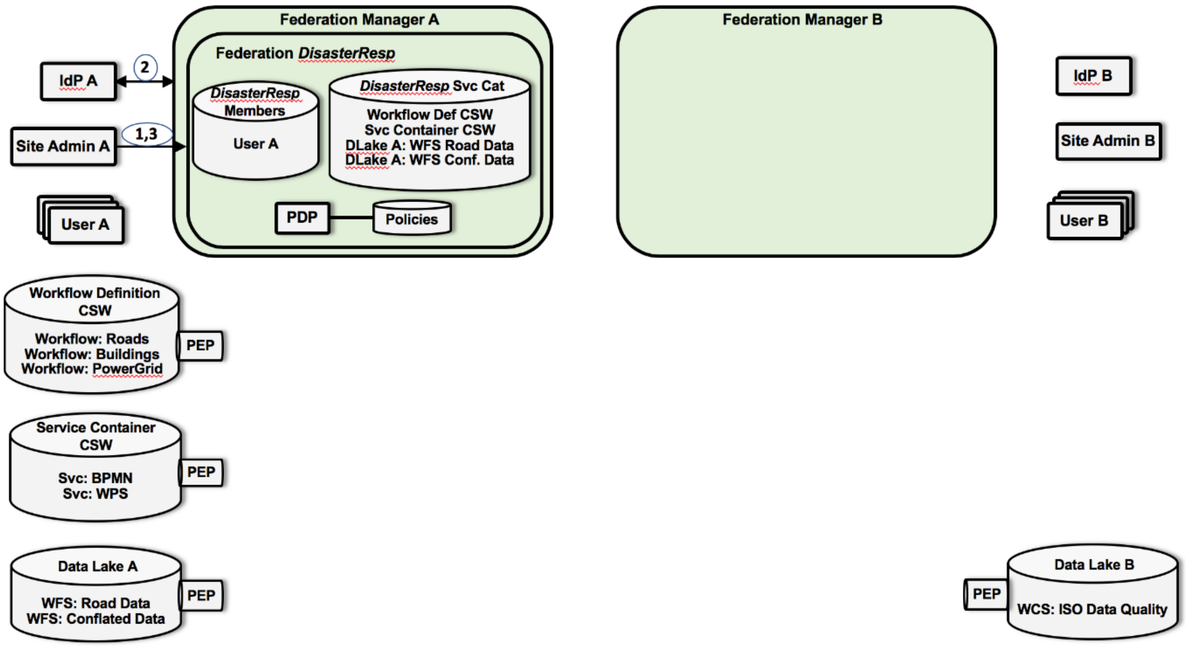
Appendix B.1 Figure 2. The System Components.

Appendix B.1 Figure 2 presents the system components of these two organizations, A and B. As independent identity silos, each organization has their own *IdP* and *Site Admin*. Each organization also has their own sets of *Users* and *services*. In each organization, the Site Admin has authorization to perform management operations on the local Federation Manager. In this example, Organization A operates a *Workflow Definition CSW*, a *Service Container CSW*, and finally a *Data Lake*. Furthermore, we assume that three workflows have already been defined and stored in the Workflow Definition CSW: *Roads*, *Buildings*, and *PowerGrid*. (Only the *Roads* workflow will be used here.) The Service Container CSW catalogs containerized services that can be instantiated as many times as needed. Also note, there is a *BPMN* service and a *WPS* service. BPMN is the *Business Process Model and Notation* [21] which has several, commercially available execution engines. The Data Lake is a large repository of data of disparate types. *Data Lake A* includes a *Road Data WFS* and a *Conflated Road Data WFS*. We note that in this example, Site Admin A is acting as the Service Owner for these services. While Organization B could operate many of the same types of services, Organization B operates its own *Data Lake B* which offers an *ISO Data Quality WCS*.



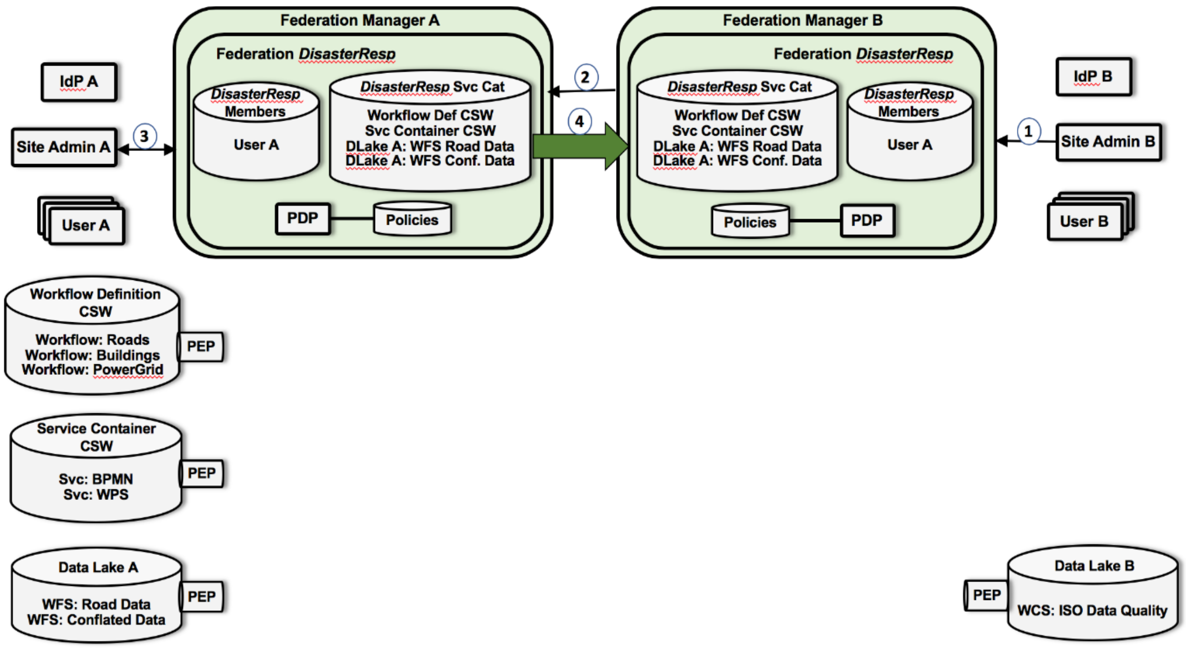
Appendix B.1 Figure 3. Fed Admin A instantiates Federation *DisasterResp* in Federation Manager A.

Appendix B.1 Figure 3 illustrates *Site Admin A* instantiating Federation *DisasterResp* in *Federation Manager A* in Step (1). In this example, *Site Admin A* acts as the *Federation Administrator (Fed Admin)* for the *DisasterResp* Federation. This federation contains a number of basic components. It keeps track of the members of *DisasterResp* and the federation attributes, project memberships, etc., they have been granted. *DisasterResp* maintains the *Service Catalog* of services that member sites have made available in this federation. Federation *DisasterResp* also maintains a *Policy Server* that is used in conjunction with a *Policy Decision Point (PDP).* In addition to policies, the Policy Server also maintains the set of federation-specific attributes on which the policies can be based.



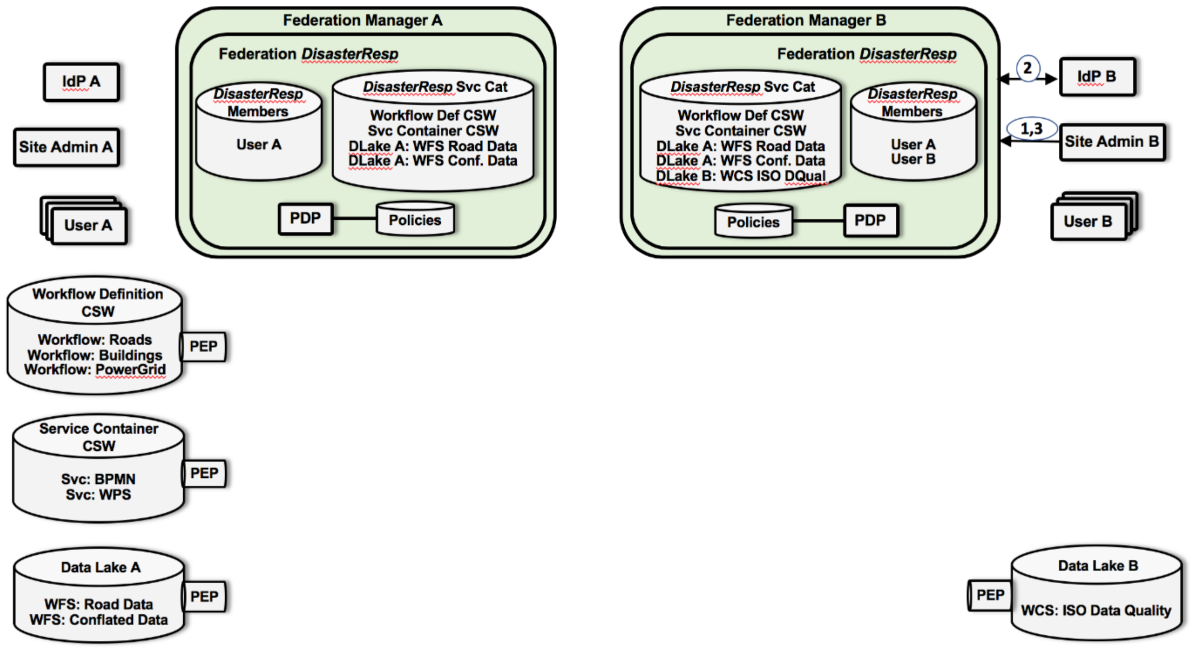
Appendix B.1 Figure 4. *Federation Admin A* populates Federation *DisasterResp*.

In Appendix B.1 Figure 4, having instantiated an empty federation, *Site Admin A* – acting as *Fed Admin A* -- begins to populate it with the necessary information. In Step (1), *Federation Admin A* grants *DisasterResp* membership to *User A*, whereby in Step (2), *IdP A* generates a *DisasterResp* credential for *User A*. In Step (3), Site Admin A – acting as the *Service Owner* -- registers four services in the *Service Catalog*: the *Workflow Definition CSW*, a *Service Container CSW*, and the *Road Data WFS* and *Conflated Road Data WFS* from *Data Lake A*. Hence, as part of Step (3) when registering services, the *Federation Admin A* can define and register resource discovery and access policies in the *Policy Server*. These policies are based on the authorization attributes that are known within the federation.



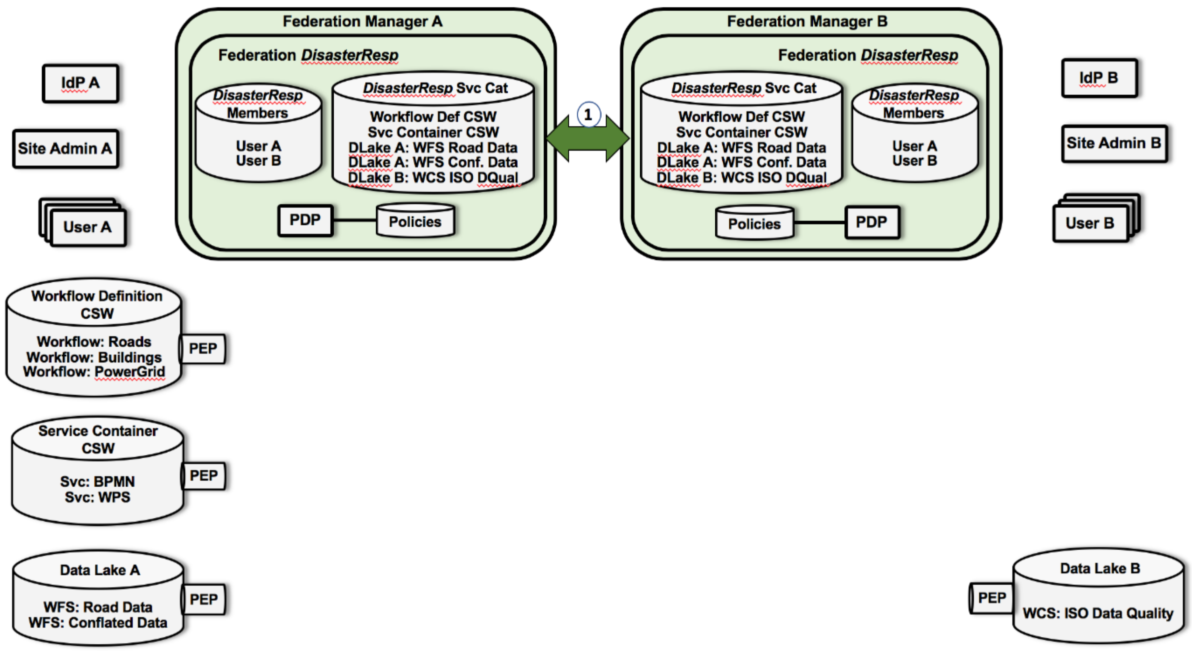
Appendix B.1 Figure 5. *Federation Admin B* decides to join Federation *DisasterResp*.

In Appendix B.1 Figure 5, *Site Admin B* has decided to join *DisasterResp*. In Step (1), *Site Admin B* makes a request to *Federation Manager B* to join the federation *DisasterResp*, which is managed by *Federation Manager A*. In Step (2), *Federation Manager B* makes this request to Federation Manager A who must establish or verify that a trust relationship exists between Organizations A and B. This is done in Step (3) by *Site Admin A* – acting as the *DisasterResp Fed Admin*. Assuming a trust relationship is in place, *Federation Manager B* receives a copy of the *DisasterResp* current state in Step (4).



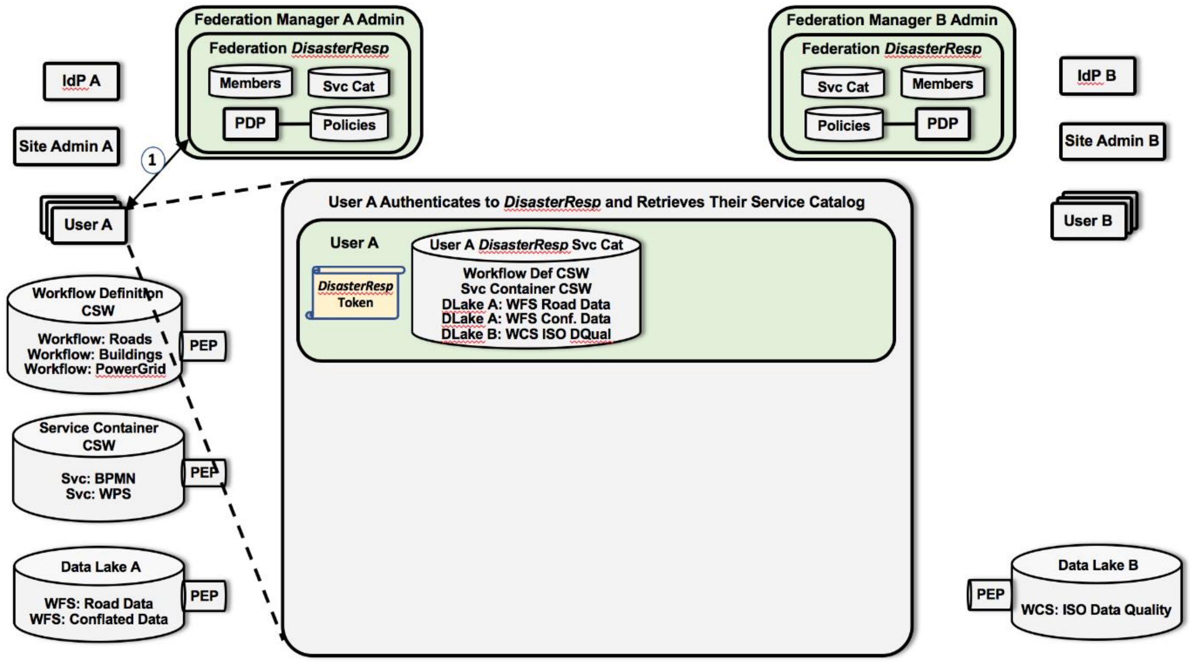
Appendix B.1 Figure 6, *Fed Admin B* populates Federation *DisasterResp* with their information.

As depicted in Appendix B.1 Figure 6, the *Federation Administrator B* adds similar types of user and service information to their local *DisasterResp* in Federation Manager B. In Step (1), *User B* is granted membership, and in Step (2), *IdP B* issues *User B* a *DisasterResp* credential. Likewise, in Step (3), *Federation Admin* B registers the *ISO Data Quality WCS* from *Data Lake B*, along with its discovery and access policies.



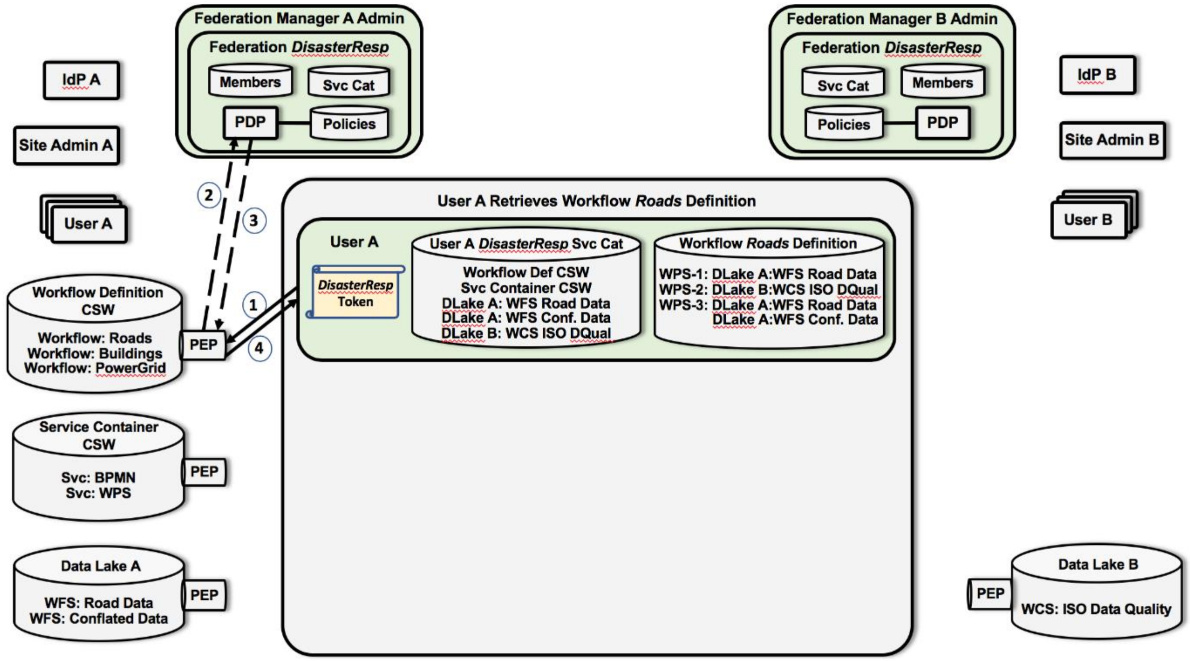
Appendix B.1 Figure 7. The Federation Managers achieve consistency.

In Appendix B.1 Figure 7, in Step (1), *Federation Managers A* and *B* eventually achieve consistency concerning Federation *DisasterResp*. We emphasize that a key function in P2P Federation Managers is to maintain such consistency. This is a fundamental requirement of the deployment and governance models in this example. Since the federation is being managed by multiple, P2P Federation Managers, any information that is changed in one Federation Manager must be propagated to all other Federation Managers involved in Federation *DisasterResp*. This is a fundamental issue within the realm of distributed computing that can be addressed using established methods.



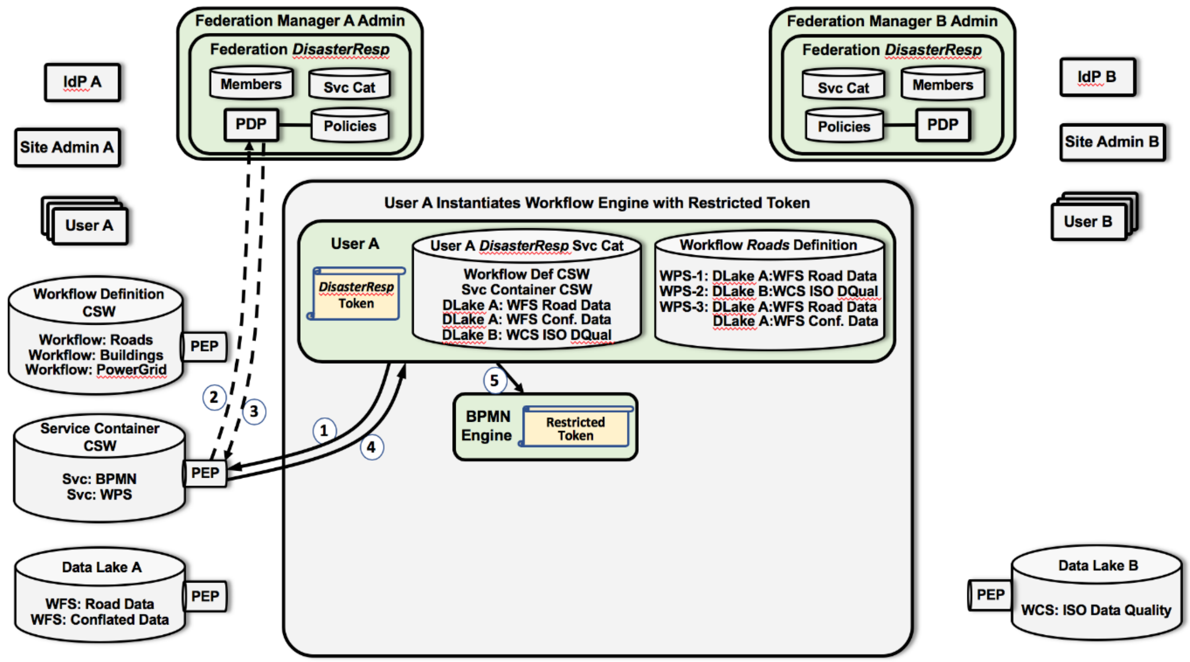
Appendix B.1 Figure 8. *User A* authenticates to Federation *DisasterResp*.

As illustrated in Appendix B.1 Figure 8, *User A* authenticates to Federation *DisasterResp* in Step(1). Upon successful authentication, *User A* has received their *DisasterResp Token*, and also their *DisasterResp Service Catalog*. Here we show the service catalog being returned as part of successful authentication. Alternatively, the Federation Manager can offer a *Service Discovery Service*. After authentication, a user could use their credential token to query the Federation Manager for the available services within the federation. We also note that a user's federation-specific service catalog may not contain all services registered within the federation. Based on a user's role within a federation, their service catalog may contain a subset of service for which they are authorized to use in some capacity. The service discovery policies are used to determine what service information is returned to the user. In this example, however, *User A*'s catalog contains all services.



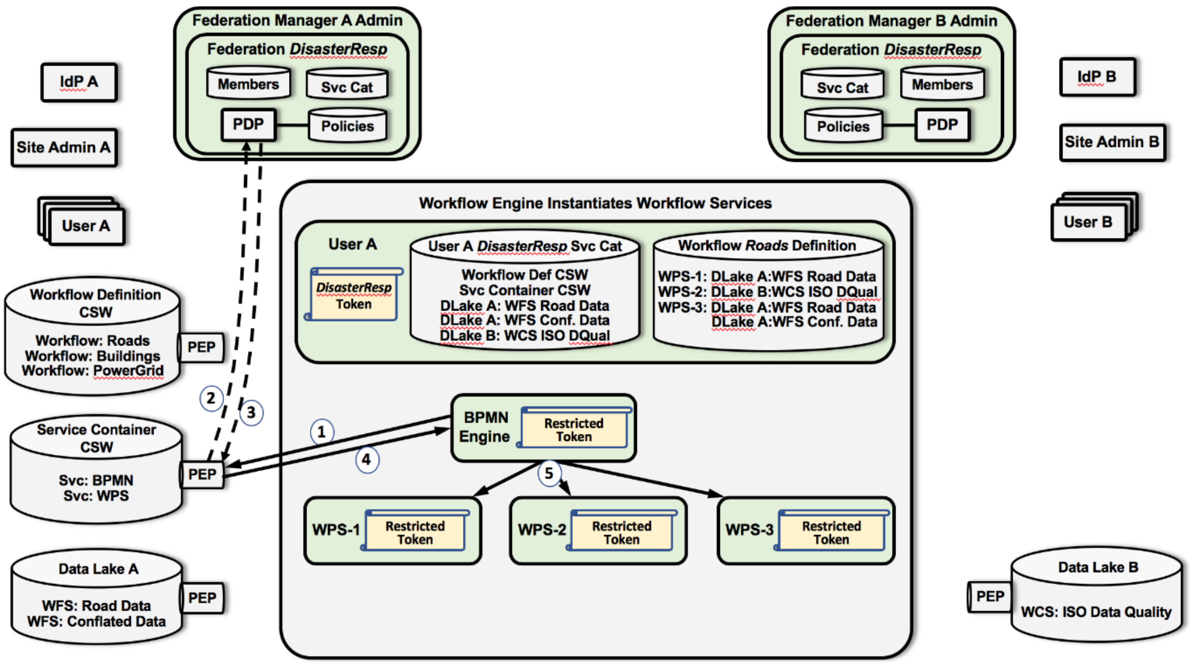
Appendix B.1 Figure 9. *User A* Retrieves the *Roads* Workflow Definition.

Appendix B.1 Figure 9 shows that *User A* can now begin the process of constructing a workflow. The first step is to retrieve the definition of the desired *Roads* workflow. In Step (1), *User A* invokes the *Workflow Definition CSW* using their *DisasterResp* token. This repository service is protected by a Policy Enforcement Point (PEP*)*. For requests involving federations, this PEP is configured to consult the Policy Decision Point (PDP) in Organization A's Federation Manager, as shown in Step (2). This PDP consults the Policy server and makes an access decision based on the requesting user's credentials and the access policy for the service being requested. The access decision is returned in Step (3), and upon success, the workflow definition is returned in Step (4). In general, this definition contains all necessary information about all services involved and the structure of their sequencing. This workflow consists of the execution of three WPSs. *WPS-1* will need to access the *Road Data WFS* in *Data Lake A*. *WPS-2* will need to access the *ISO Data Quality WCS* in *Data Lake B*. Finally, *WPS-3* will need to access both the *Road Data WFS* and the *Conflated Road Data WFS* in *Data Lake A*.



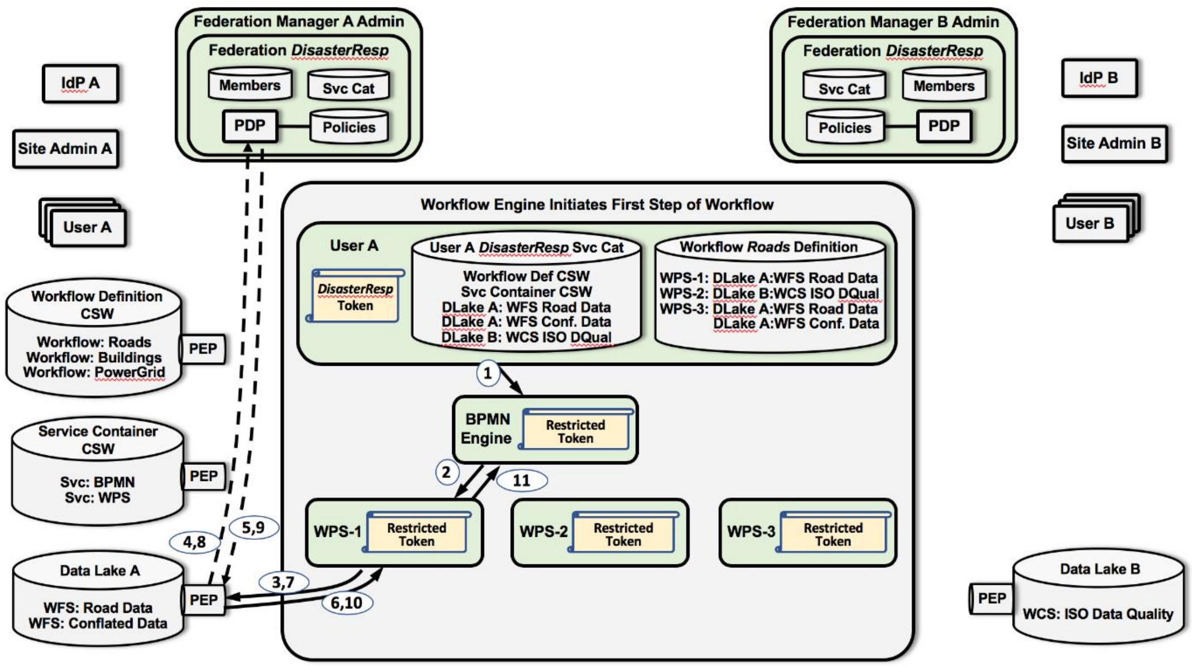
Appendix B.1 Figure 10. *User A* Instantiates the BPMN Workflow Engine.

In Appendix B.1 Figure 10, *User A* instantiates the BPMN Workflow Engine where the same sequence of authorization steps take place. All necessary services are containerized and stored in the *Service Container CSW*. Hence, in Step (1), *User A* requests that a BPMN container is started. Federation-specific authorization decisions are made in Steps (2) and (3). Upon success, the BPMN container information is returned in Step (4). In Step (5), the BPMN server is configured with a restricted authorization token derived from *User A*'s token, along with the necessary workflow information. While not explicitly illustrated, a Restricted Token could be produced by an OAuth 2 Client Credentials Authorization Grant [22].



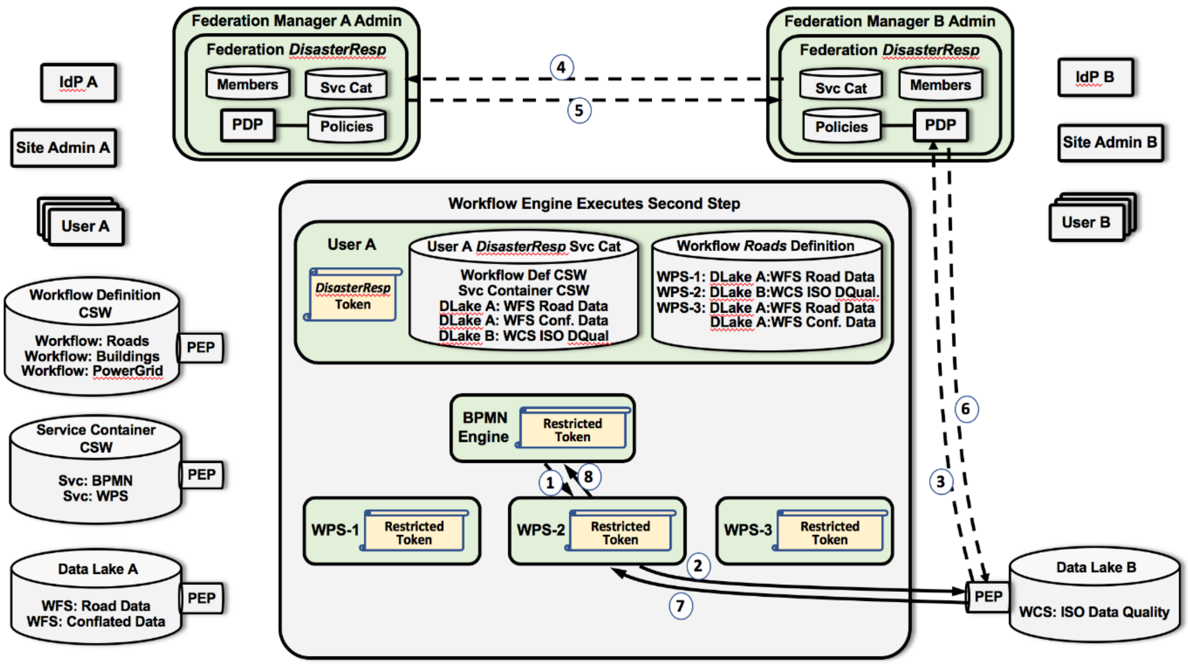
Appendix B.1 Figure 11. Workflow services are instantiated.

In Appendix B.1 Figure 11, using its restricted authorization, the *BPMN* service also accesses the *Service Container CSW*. The same authorization sequence in Steps (1), (2), (3), and (4) occurs as it does in Appendix B.1 Figure 9 and Appendix B.1 Figure 10, then three WPS service containers are spun-up in a Step (5). These services are also configured with restricted authorization tokens derived from *User A*'s *DisasterResp* token.



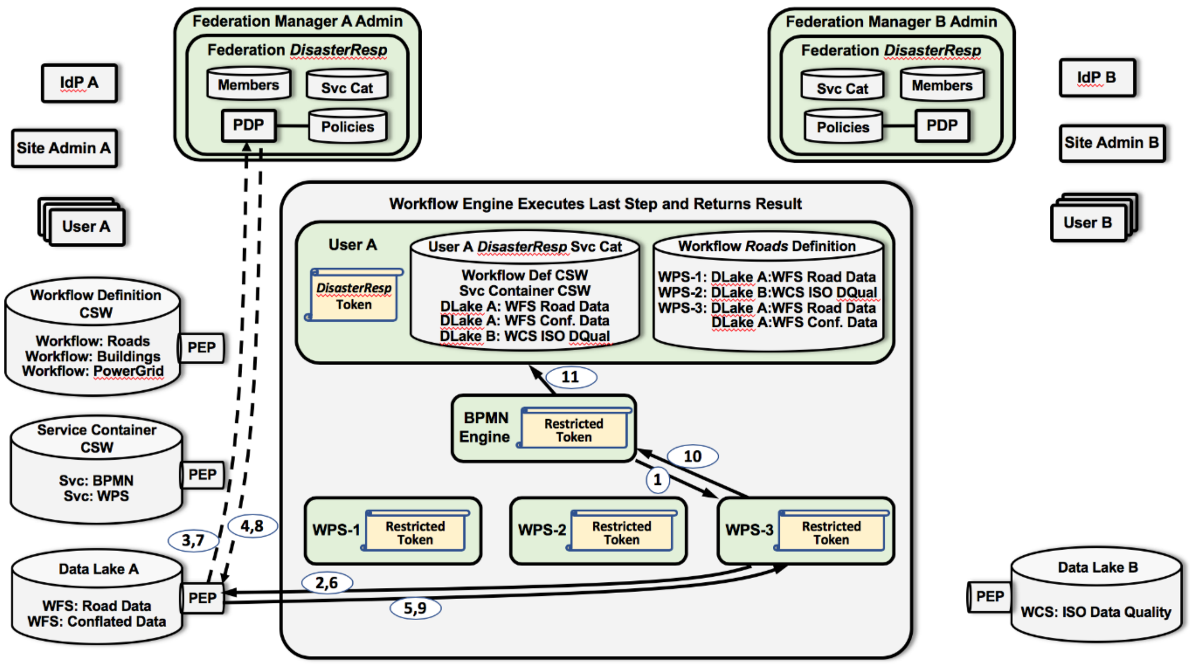
Appendix B.1 Figure 12. The workflow is initiated.

Appendix B.1 Figure 12 shows how *User A* starts the workflow in Step (1) by passing the geographical parameters for the desired Road Data to the *BPMN* service. In Step (2), *BPMN* executes *WPS-1*. This service needs to retrieve data from the *Road Data WFS* and perform coordinate transformations, if needed. The initial request is made in Step (3). Following the same sequence of operations, this request is validated with *the Federation Manager A PDP* in Steps (4) and (5). If successful, the data is returned in Step (6). Assuming some coordinate transformation had to be done, the transformed data is written back to the *Road Data WFS* in Step (7). Validation and authorization is done in Steps (8) and (9), with a final return message in Step (10). In Step (11), *WPS-1* passes a reference to the transformed data in *Data Lake A* back to the *BPMN Engine*.



Appendix B.1 Figure 13. The second workflow step is executed.

Appendix B.1 Figure 13 shows how in Step (1), the parameters of the desired Road Data are passed to *WPS-2*. *WPS-2* needs to assess the data's quality by contacting the *ISO Data Quality WCS* in Step (2). Here *Data Lake B PEP* contacts its local *Federation Manager PDP* to validate and authorize the request. *Federation Manager B* determines that the credentials associated with this request were issued by its trusted peer, *Federation Manager A*. In Steps (4) and (5), *Federation Manager B* asks *Federation Manager A* to make the validation and authorization decision, which is returned in Step (6). Upon success, the *ISO Data Quality WCS* does the quality checks and returns the results in Step (7). *WPS-2* makes a Go/No-Go decision and returns this result to the *BPMN Engine*. If the data quality is insufficient, the workflow is then terminated.



Appendix B.1 Figure 14. The last workflow step is executed and final results returned.

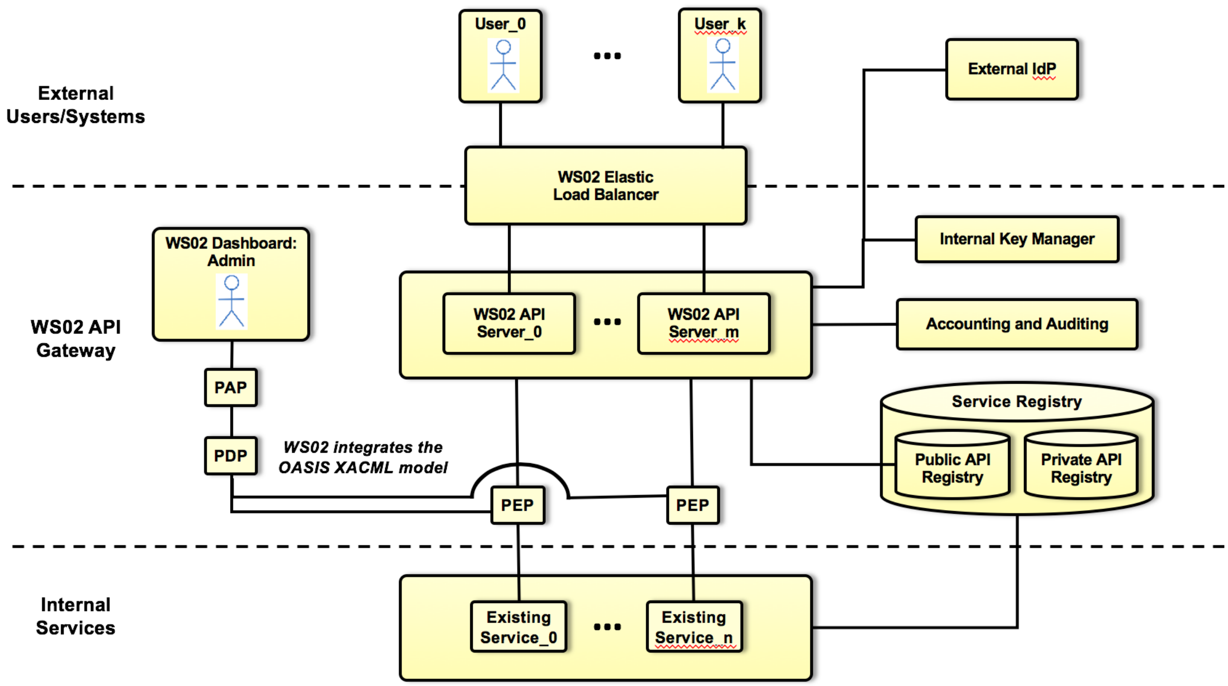
Appendix B.1 Figure 14 depicts the workflow if the data quality is sufficient, the *BPMN Engine* executes the last step. In Step (1), the reference to the transformed data in *Data Lake A* is passed to *WPS-3*. This sends a request to the *Road Data WFS* in Step (2). After validation and authorization in Steps (3) and (4), the data is returned in Step (5). After conflating the road data, the results are written to the *Conflated Data WFS* in Step (6). After validation and authorization in Steps (7) and (8), with a final return message in Step (9). *WPS-3* returns a reference to the final, conflated road data product to the *BPMN Engine* in Step (10). Since the workflow is complete, the reference to the final road data product is returned to *User A* in Step (11). At this point, the workflow could be run again, perhaps with different parameters, or the BPMN Engine and the WPSs could simply be terminated.

This use case example has illustrated how the Reference Architecture concepts, and specifically the Federation Manager, could be mapped to a more concrete deployment with a specific governance model. This was done by identifying a "real-world" workflow example and walking through the process of creating a federation and its use. By creating a virtual administrative domain, the Federation Managers were able to jointly enforce access policies for shared resources.

Clearly, though, there will be performance and scalability issues. Doing a remote credential validation and authorization on every call will be a significant overhead, especially for those that involve multiple Federation Managers. Establishing trust and basic communication security must also be addressed. For many application domains, trust will be established by traditional methods. In a service architecture, communication security could be accomplished using established tools, such as TLS.

B.2. The WS02-OpenID Connect Use Case

Gaining implementation experience of systems based on the Reference Architecture also means investigating how existing tools and standards could be re-purposed or augmented to provide the desired federated, resource-sharing capabilities. We have noted above that Web Service API Gateways are very relevant to the Federation Manager concept. They maintain a registry of externally visible services and apply service owner-defined policies on incoming requests. We have also noted above that OpenID Connect [23] might be used in managing access tokens used in a federation. In this use case, we explore how a Web Service API Gateway, specifically WS02, could be integrated with OpenID Connect to realize the semantic functionality of a Federation Manager.

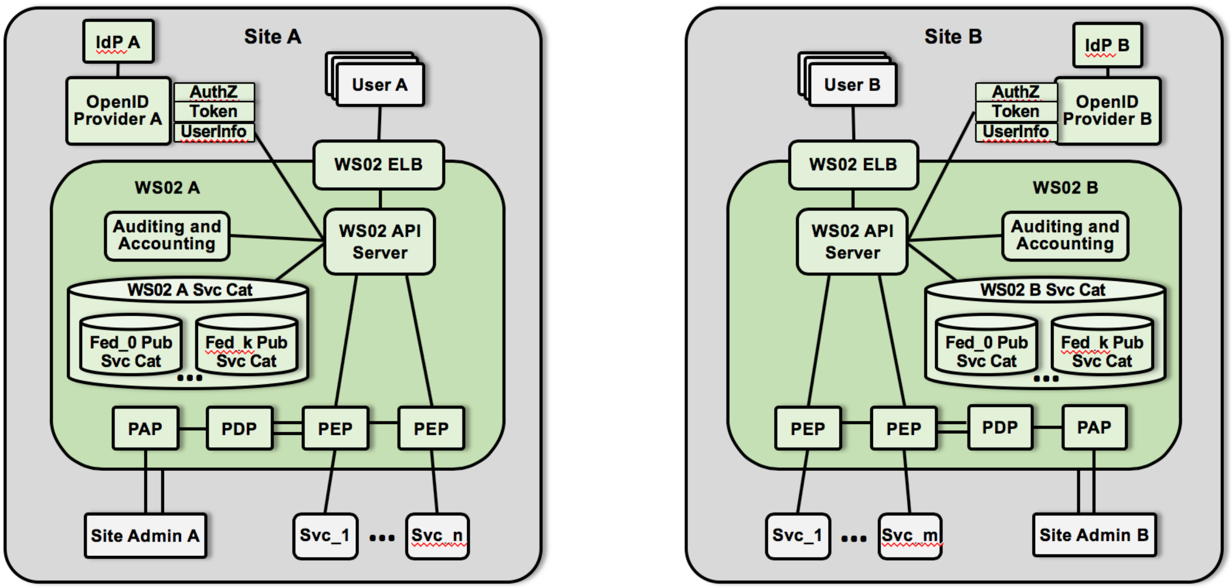


Appendix B.2 Figure 1. The WS02 Architecture.

Appendix B.2 Figure 1 presents the architecture of WS02 [24], a well-established, open source API Gateway. External users access services through a *Load Balancer* on the front-end to any number of *API Servers* necessary to meet throughput demands. These API Servers authenticate users through an *External IdP*. This enables WS02 to be integrated into existing enterprise environments, where the External IdP could be something like a corporate LDAP, Active Directory, or PKI Certificate Authority. The API Servers also log all necessary events for accounting and auditing.

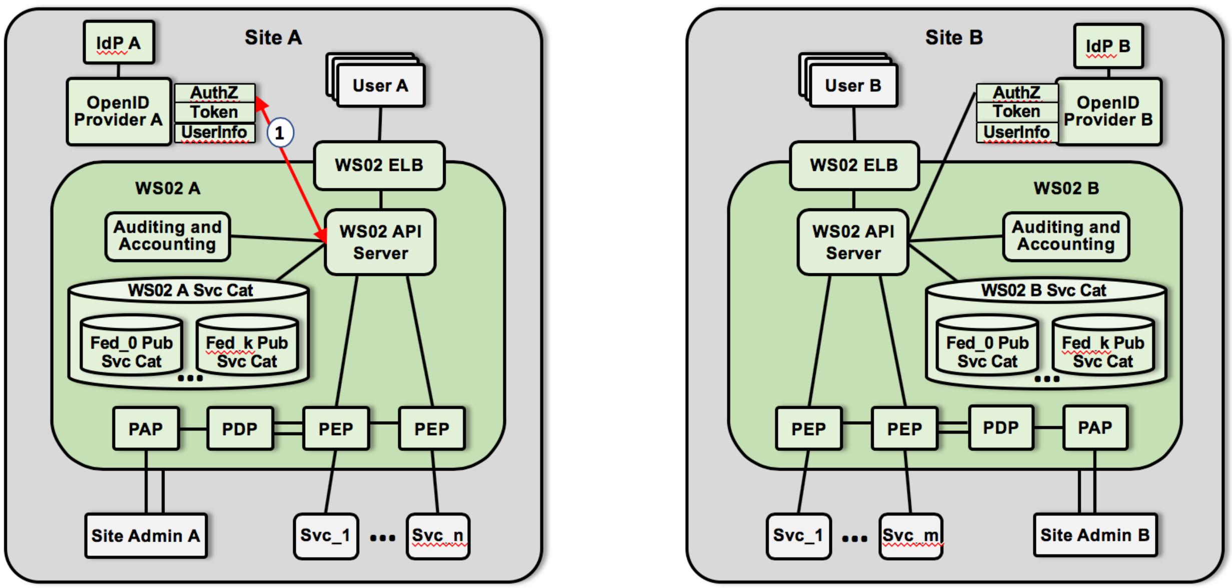
Existing internal services are registered with WS02. During development, a service can be registered with the *Private API Registry*. When ready, a service can be registered with the *Public API Registry*, at which time the service becomes discoverable by external users.

WS02 integrates the OASIS XACML model [25]. Every service is protected by a *Policy Enforcement Point (PEP)* which rely on a *Policy Decision Point (PDP)*. The WS02 Admin manages the service policies through a *Policy Administration Point (PAP).* We note that existing services do not have to be modified in any way to be managed by WS02.



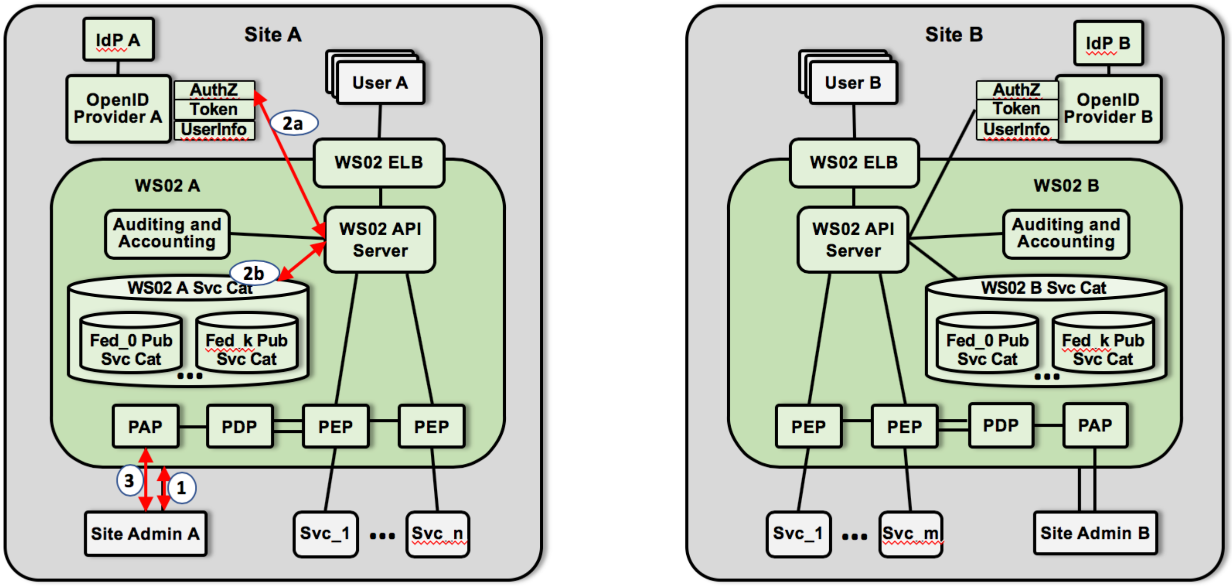
Appendix B.2 Figure 2. A Federation Manager based on WS02 and OpenID Connect.

Appendix B.2 Figure 2 illustrates two Federation Managers based on WS02 and OpenID Connect – one for *Site A* and one for *Site B*. Rather than just maintaining a private and public service catalog, each FM maintains a service catalog for each federation that it is supporting. The external IdP is interfaced through an *OpenID Provider* as specified in the OpenID Connect standard. The OpenID Provider has three endpoints – *AuthZ, Token,* and *UserInfo* – that are used for different functions. These will be described later. In this example, a peer-to-peer deployment of two internal Federation Managers is being illustrated.



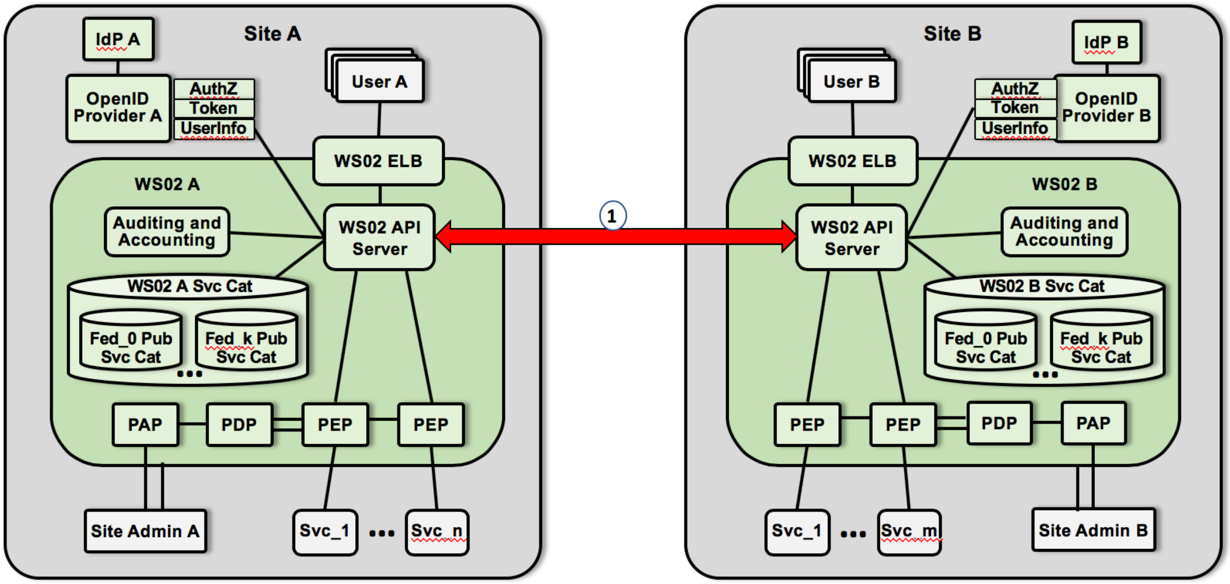
Appendix B.2 Figure 3. The WS02 API Server registers a redirection URI.

This example is based on using a form of the *Authorization Code Flow*. In Appendix B.2 Figure 3, when initially deployed, the WS02 API Server must register a redirection URI with the OpenID Provider through the AuthZ endpoint. (Shown as Step (1).) When the API Server is subsequently authenticating members through a redirection, the redirection URI being used must match the URI that was originally registered. This happens in both Site A and Site B.



Appendix B.2 Figure 4. Site Admin A does initial configuration of a Federation Foo.

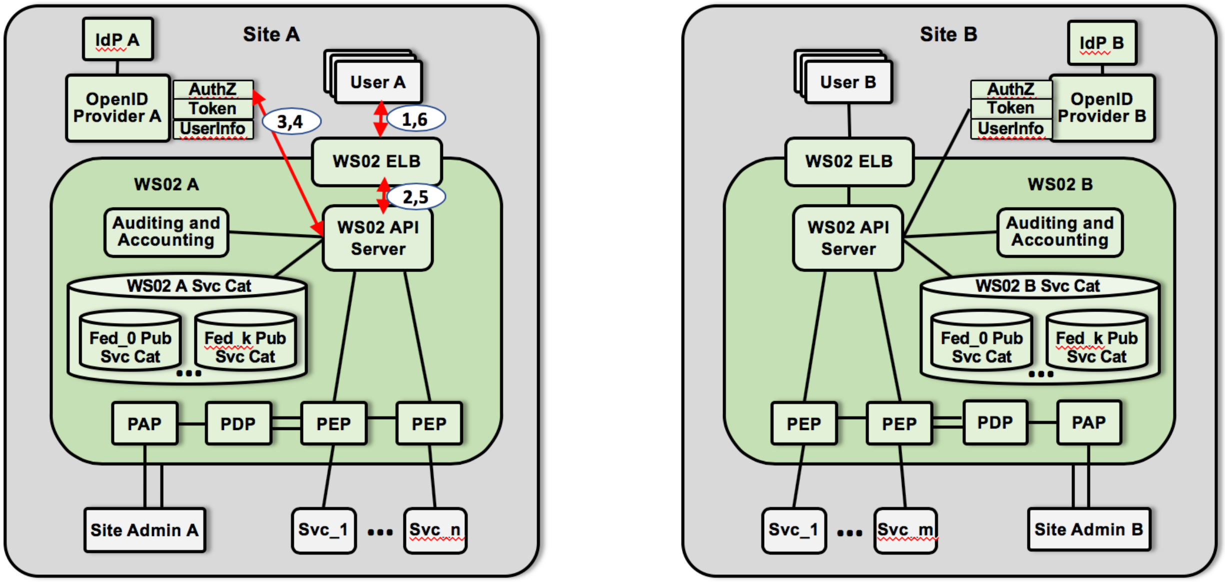
In Appendix B.2 Figure 4, after the Federation Manager itself is configured and running, Site Admin A can begin configuring federations. In Step (1), we can say an "empty" *Federation Foo* is created. In Step (2a), Federation Foo membership and authorizations are granted to local users by registering this information with the OpenID Provider. In Step (2b), local services are populated in the local service catalog for Federation Foo, along with their discovery policies. In Step (3), the access policies specific to these services in Federation Foo can be specified. This can happen in both Sites A and B.



Appendix B.2 Figure 5. WS02 API Servers exchange federation information.

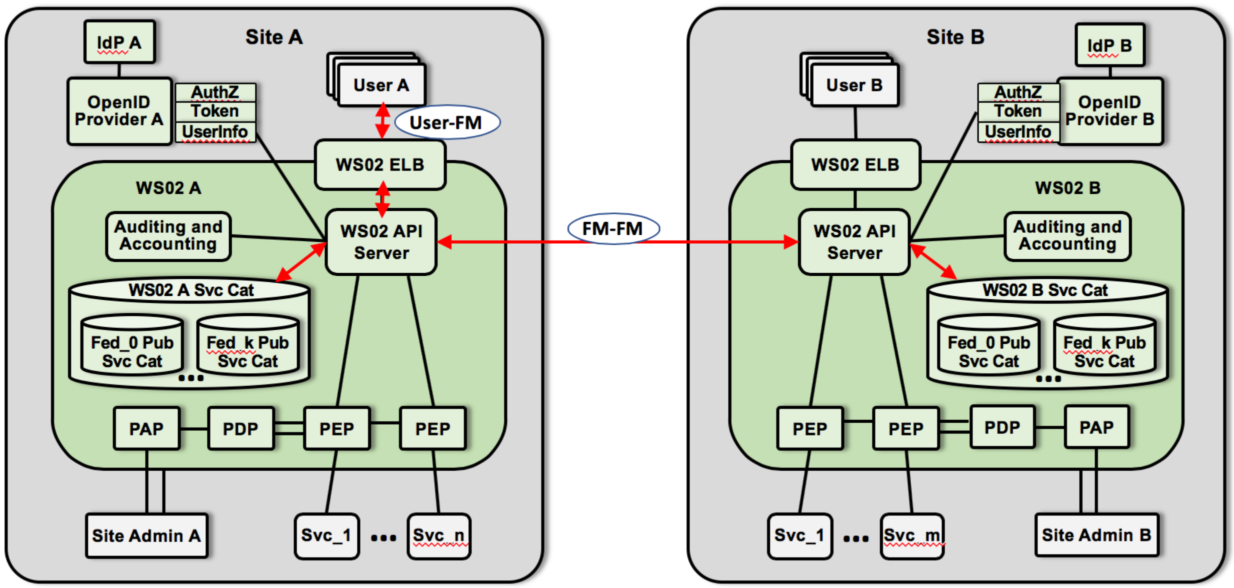
At this point, a trust relationship between Site A and Site B has already been established. Since this is a peer-to-peer deployment, the two Federation Managers must exchange information about the federations they are hosting. In Appendix B.2 Figure 5, since the trust relationship is in place, they can be configured to establish a secure, trusted communication channel between them. (Step (1).) The exact information that is exchanged, and how, can vary according to the desired governance model. Generally speaking, the FMs may need to exchange information about federation members and their identity attributes, information about a specific federation service catalog, or respond to authorization requests.

This communication can also be managed in different ways. While this is a P2P deployment, it could be managed simply in a static point-to-point topology. FMs could also forward requests through a topology of FMs using some routing algorithm. We note that even an eduROAM-like tree of RADIUS servers could be used. Here, a request to set-up a TLS session could be routed from the source FM to the destination FM. After the TLS session has been established, the secure transaction can take place. When that has been concluded, the TLS session is terminated.



Appendix B.2 Figure 6. User A authenticates to their local WS02.

After all the initial configuration has been done, User A can authenticate to its local WS02. In Appendix B.2 Figure 6's Steps (1), (2), and (3), an authentication request is sent to the OpenID Provider's AuthZ endpoint. OpenID Connect uses the notion of *scope* to manage the range of operations that a user is being authenticated for. Hence, User A can be said to be authorized for the *scope* of *Federation Foo*. After successful authentication, a *Client Identifier* is returned to User A in Steps (4), (5), and (6). We note that this is not an authorization token.



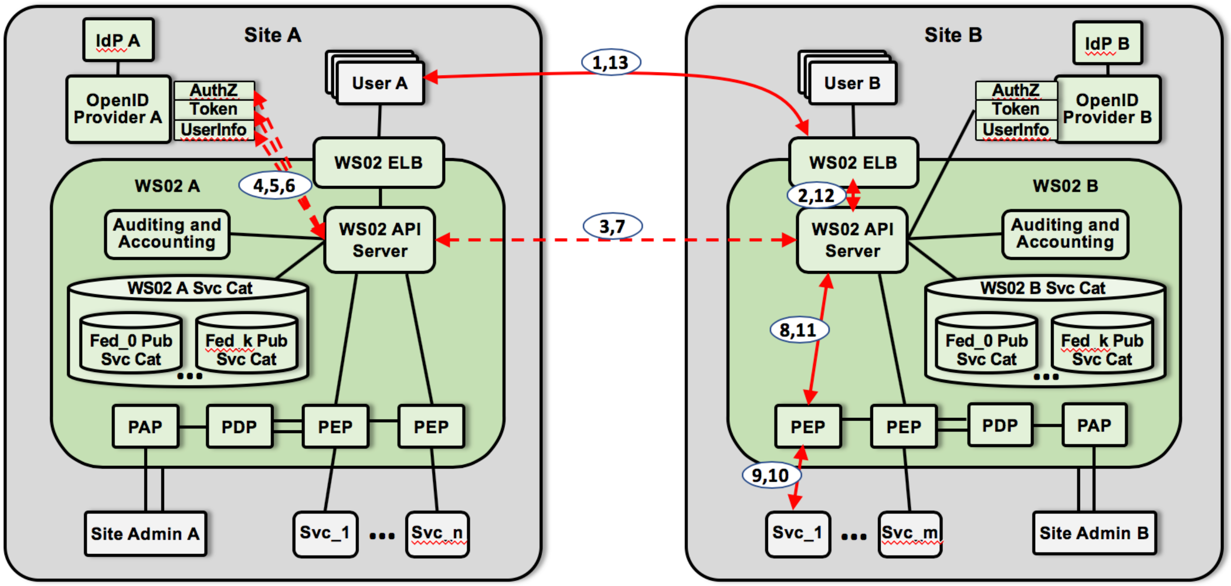
Appendix B.2 Figure 7. User A is authorized to do discovery on the Foo Service Catalog.

Once authenticated for Federation Foo, User A is authorized to discover services in Federation Foo, as constrained by the discovery policy for each service. Federation Foo can be said to have a *Service Catalog*. Since this is a P2P deployment, this service catalog could be physically distributed among the FMs involved. Hence, the discovery process could be *logical* and supported in many different ways.

Broadly speaking, the discovery process between User A and FM A could be done in an *eager* or *lazy* manner. (This could also be called *push* or *pull, respectively*.) Since this is a P2P deployment, the discovery process between FM A and FM B could likewise be done in a lazy or eager manner. Because of this, the actions in Appendix B.2 Figure 7 will not be labeled in a strict numerical sequence. We will instead itemize several options based on these properties:

- *Eager User-Eager FMs.* One approach is for all FMs to share catalog information in an eager, push manner. Whenever a service is added or deleted from the catalog at one site, that change is propagated to all other sites as quickly as possible for eventual consistency. Hence, each FM would be maintaining a replica of the entire Foo service catalog. With this approach, a complete catalog could be eagerly returned to User A as a result of successful authentication.
- *Lazy User-Eager FMs.* Here the FMs share information as before, but the User must query for catalog information after successful authentication. These queries could be based on different server metadata attributes. Since the FMs are maintaining complete replicas, all queries are satisfied locally.
- *Lazy User-Lazy FMs.* Here the FMs are not maintaining complete replica. When a User poses a query, a partial response could be produced from the local information. However, queries could also be propagated to other FMs to discover additional services. The service information retrieved could be cached for subsequent use.

We note that an eager user with lazy FMs is not a practical option. While local catalog information could be returned to a user on successful authentication, a user would need to make further queries anyway to discover federated services from other sites.



Appendix B.2 Figure 8. User A invokes a service in Site B.

Finally, as shown in Appendix B.2 Figure 8, after User A has authenticated and discovered a useful service, User A invokes that service in Step (1). This gets routed to the Site B WS02 API Server in Step (2). This API Server determines that this is a request from a different site, i.e., Site A. An authorization request then is routed to Site A in Step (3). The Site A API Server performs a series of actions. First, the API Server verifies that User A has already been authenticated by using the OpenID Provider's *AuthZ* endpoint in Step (4). This returns an *Authorization Grant*. The API Server can then exchange this grant for an *Authorization Token* by using the *Token* endpoint in Step (5). The API Server can also acquire additional *Claims* information about the user, i.e., identity and authorization attributes, by using the *UserInfo* endpoint in Step (6). The AuthZ Token and Claims are returned to Site B in Step (7) which are forwarded to the appropriate PEP in Step (8). Assuming that access is granted, the service is invoked in Step (9) and the results are returned to User A in Steps (10) through (13).