OGC Testbed-14

Federated Clouds Engineering Report

Table of Contents

Publication Date: 2019-03-05

Approval Date: 2019-01-22

Submission Date: 2018-12-01

Reference number of this document: OGC 18-090r1

Reference URL for this document: http://www.opengis.net/doc/PER/t14-D023

Category: Public Engineering Report

Editor: Dr. Craig A. Lee

Title: OGC Testbed-14: Federated Clouds Engineering Report

OGC Engineering Report

COPYRIGHT

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER’S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR’s sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to

indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Chapter 1. Summary

The geospatial community has had an on-going challenge with being able to share data and compute resources in dynamic, collaborative environments that span different administrative domains. For these types of requirements, the concept of *federation* has been developed. The near-term goal of the Federated Cloud task in OGC Testbed-14 is to demonstrate a specific data-sharing scenario among two or more administrative domains using existing security tooling, e.g., OpenID Connect and OAuth. The main details of this work are reported as part of the *OGC Testbed-14 Security Engineering Report* (ER) [1]. This *Federated Cloud Engineering Report (ER)* dovetails with the Security ER to:

- Coordinate across all federation-related tasks in Testbed-14, including the *Earth Observation Cloud* and *Workflow* tasks,
- Understand the overall federation design space,
- Analyze and critique the scope, trade-offs and limitations of the federation capabilities being built and demonstrated in Testbed-14,
- Identify and prioritize possible incremental development tasks for subsequent testbeds, and
- Liaison with groups external to OGC, such as the National Institute of Standards and Technology (NIST)/Institute of Electrical and Electronics Engineers (IEEE) Joint Working Group on Federated Cloud, to promote the further development and adoption of federated capabilities, and ultimately international standards.

1.1. Requirements & Research Motivation

The advent of the cloud computing era has fundamentally changed how people and organizations view computing — and more specifically how people and organizations interact with the resources that they care about, i.e., data and services. There is a popular notion that "everything" will be available from "anywhere" simply because it is "in the cloud".

However, the core capability of cloud computing is the *on-demand provisioning of compute resources*, e.g., compute and storage. This is completely orthogonal to the requirement of *managing access* to those compute and storage resources. All computing resources, including clouds, exist in some type of *administrative domain* wherein access management can be done. As long as resources are all in the same administrative domain, managing access is straight-forward.

However, with the continued development of our interconnected world, it is becoming increasingly common that the data and services desired by a user exists across different administrative domains. Organizations — each with their own administrative domain — that need to collaborate will also need a way to securely bridge those domains to selectively share data and services to achieve their joint organizational goals. Easily accessing resources distributed across different administrative domains is a problem. The naive approach is for an individual to maintain *n* different accounts and credentials for *n* different organizations. A more effective approach is *federation*.

Simply put, federation enables a set of participating organizations to selectively share data and resources for specific purposes. As an example for geospatial applications, this could mean

enabling a user to access a Web Feature Service (WFS) from different providers for a common purpose, using a single set of credentials. The goal is to make federated environments as seamless, transparent, and easy to use as a "normal" centralized environment. The Testbed-14 Federated Cloud tasks were structured around building and demonstrating a specific capability for managing credentials and access between two environments.

1.2. Prior-After Comparison

The Testbed-13 Cloud Engineering Report documented the efforts to enable a user associated with one cloud environment to access data from other cloud environments. The authentication and authorization approaches taken include the notion of a Distributed Access Control System (DACS) from Natural Resources Canada (NRCan) which is based on augmenting an Apache web server with a DACS module. The DACS can be used to act as a firewall for known users, IP addresses, etc. The DACS can also set environment variables, modify request headers with additional user information, or use an encrypted, client-side cookie to verify user identity and make access decisions. Another approach from CRIM was to use an encrypted cookie, but ran all requests through a Security Proxy. This Security Proxy could authenticate a user against an external IdP (e.g., LDAP), but maintained its own Access Control Lists for making access decisions.

Both of these approaches have fundamental shortcomings. While the DACS calls itself distributed, it is, in fact, not a distributed control system. As reported in this ER, the DACS are apparently independent of one another. If they are to be used to manage a distributed security environment, each must be manually configured to do so. A similar argument can be made for the CRIM Security Proxy. Also, neither approach has any support for resource (service) discovery, or being able to define and enforce resource discovery policies. Likewise, there is no notion of being able to define a uniform, consistent federated environment wherein such governance can be accomplished.

For all the reasons observed above in the Testbed-13 efforts, the computer science concept of federation was developed to enable secure, on-demand collaborations of all manner. A federation is a security and collaboration context wherein participants can define, agree upon, and enforce joint resource discovery and access policies. A federation is not necessarily owned by any one organization, or located at any one site. In the current cloud computing era, many federations will be cloud-based. However, since federations can be managed at any level in the system stack, it is certainly possible to address federation at the Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) levels. To manage SaaS-level federations means to manage arbitrary, application-level services. That is to say, federations can be used to manage collaborations for arbitrary business functions.

Hence, the goal for the Testbed-14 Federated Cloud task was to demonstrate a simplified federated, data-sharing scenario with the ultimate goal of enabling cross-domain collaborations for arbitrary geospatial services and applications. This was done by building and demonstrating a specific capability to access a service in one administrative domain using credentials issued in another administrative domain. This capability was enabled by the *Mediation Service* that was integrated with an *Authorization Server*. On a request from an external user, the Mediation Service would interact with the Authorization Service to create a local account and credential. The external user could then use this credential to successfully complete the service request.

While this is a valuable capability, there are additional functional capabilities that are necessary to

realize a general federation capability. More specific information is given in Sections 7 and 8. This gives rise to the following recommendations.

1.3. Recommendations for Future Work

The Federated Cloud tasks in Testbed-14 were constructed to address specific functional capabilities that were considered to be integral to enabling on-demand collaborations, i.e., the secure sharing of data among a known set of trusted participants. However, when evaluated against the NIST Federated Cloud Reference Architecture, there are a number of additional capabilities that are clearly necessary. Also, the established standards used in Testbed-14 were not intended nor designed to be used to support *virtual administrative domains*, i.e., federations. Hence, further coordination between OGC and the joint NIST/IEEE Federation Cloud efforts is needed to realize more complete implementations and validate the overall design approach. This will greatly facilitate industry adoption of scalable, general federations.

Based on these observations, we can make the following recommendations for future work as given in Section 9:

1. Clearly define and demonstrate how federated identity can be consistently managed and used.
2. Clearly define and demonstrate how the scope of attributes and authorizations can be used to consistently manage federated environments.
3. Clearly define and demonstrate how resource discovery and access can be consistently managed across all participating administrative domains.
4. Clearly define and demonstrate how federation administration is done.
5. Strategize on the development and use of federation deployment models.
6. Clearly identify and evaluate implementation trade-offs with regards to practical adoption issues, e.g., modifications to existing services.
7. Investigate and evaluate the benefits and necessary investment for developing purpose-built standards and tooling.
8. Develop awareness and understanding at the organizational level of the purpose and need for Trust Federations.

Complete information is presented in Section 9.

1.4. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contacts

Name	Organization
Craig A. Lee, editor	The Aerospace Corporation

1.5. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

This is an informational document. Many existing, relevant standards are discussed and possible areas of standardization are identified. While this document does make recommendations, they are not binding for any tools, systems, or organizations. The referenced standards include:

- OGC: Catalogue Services 3.0 - General Model, 12-168r6, http://docs.opengeospatial.org/is/12-168r6/12-168r6.html
- SNIA: Cloud Data Management Interface, https://www.snia.org/cdmi
- OGC: GeoXACML Implementation Specification, http://www.opengeospatial.org/standards/geoxacml
- IETF: RFC 4511, Lightweight Directory Access Protocol (LDAP): The Protocol, https://www.ietf.org/rfc/rfc4511.txt
- IETF: RFC 6749 - The OAuth 2.0 Authorization Framework, https://tools.ietf.org/html/rfc6749
- OpenID: OIDF, OpenID Authentication 2.0 - Final, https://openid.net/specs/openid-authentication-2_0.html
- OIDF: OpenID Connect Core 1.0, http://openid.net/specs/openid-connect-core-1_0.html
- OGF: Open Cloud Computing Interface, http://occi-wg.org/about/specification
- IETF: RFC 2865, Remote Authentication Dial In User Service (RADIUS), https://tools.ietf.org/html/rfc2865
- OASIS: Security Assertion Markup Language (SAML) V2.0 Technical Overview, http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html
- OASIS: SAML V2.0 Enhanced Client or Proxy Profile Version 2.0, http://docs.oasis-open.org/security/saml/Post2.0/saml-ecp/v2.0/saml-ecp-v2.0.html
- OASIS: Topology and Orchestration Specification for Cloud Applications (TOSCA), https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca
- IETF: RFC 5246, The Transport Layer Security (TLS) Protocol, Version 1.2, https://tools.ietf.org/html/rfc5246
- Kantara Initiative: User-Managed Access (UMA) Profile of OAuth 2.0, https://docs.kantarainitiative.org/uma/rec-uma-core.html
- OGC: OGC Web Services Security, 17-007, https://portal.opengeospatial.org/files/77693
- OGC: Web Feature Service 2.0 Interface Standard – With Corrigendum, 09-025r2, http://www.opengeospatial.org/standards/wfs
- 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
- W3C: eXtensible Markup Language (XML), https://www.w3.org/XML

Chapter 3. Terms and Definitions

This ER uses many terms from the *NIST Federated Cloud Reference Architecture* [2], many of which are based on the *NIST Cloud Computing Reference Architecture* [3]. For convenience, some of these terms are given here. Complete information, along with additional terms and definitions, can be found in [2] and [3].

- Administrative Domain

```
A system that augments a Security Environment by defining a desired set of policies
and governance for managing users and resources.  This includes defining an
authoritative source for policy concerning resource discovery and use.
```

- Cloud Service Consumer

```
A Cloud Service Consumer uses services provided by a Cloud Service Provider, as
managed within an Administrative Domain and Regulatory Environment.  See [1] and
[2] for more details.
```

- Cloud Service Provider

```
A Cloud Service Provider (CSP) is responsible for making cloud services (IaaS,
PaaS, or SaaS) available to Cloud Service Consumers, within a given Administrative
Domain and Regulatory Environments.  A CSP is comprised of a number of components
that are described in more detail in [1] and [2].
```

- Federation Auditor

```
A Federation Auditor will be an independent, third-party that can assess compliance
for any type of policy associated with a federation.
```

- Federation Broker

```
A Federation Broker enables potential federation members to find federations they
wish to join, and vice-versa.  There must be some type of Catalog of Federations
wherein federation owners can register salient information about their federations,
along with a discovery service that can be used by potential members.  See [1] for
more information.
```

- Federation Carrier

```
A Federation Carrier provides connectivity and transport of federation services
among federated sites, federation members, and Federation Managers.
```

- Federated Environment

```
The creation and management of a Virtual Administrative Domain whereby the same
kind of policies and governance can be used to manage users and resources within
the VAD that are, in fact, coming from an arbitrary number of non-federated
Administrative Domains.  This depends on a Federated Security Environment.
```

- Federation Manager

```
A Federation Manager (FM) is the entity that manages different, multiple Federation
Instances, or simply federations, among two or more Administrative Domains.
Multiple FMs may cooperate to form larger, distributed federation infrastructures.
```

- Identity Provider

```
An Identity Provider (IdP) manages a user's primary authentication credentials and
issues assertions derived from those credentials.  These credentials may be simple
account names and passwords, or they could be cryptographically signed documents.
The derived assertions involves roles or attributes on which Role-Based or
Attributed-Based Access Control can be built.
```

- Regulatory Environment

```
A regulatory environment is a governmental jurisdiction that can exist at the
local, state and national levels.  All Cloud Service Consumers or Cloud Service
Providers within that jurisdiction must observe all relevant regulations defined by
those governmental entities.
```

- Security Environment

```
A system that securely manages end-user information for the purpose of providing
Identity Management and Access Control.  These capabilities are usually achieved by
using cryptographic methods, secure network design, and observing data protection
regulations.
```

- Trust Federation

```
A set of Administrative Domains (Sites) and Federation Managers that have an
established set of trust relationships.
```

3.1. Abbreviated terms

- API Application Programming Interface
- AS Authorization Server
- AWS Amazon Web Services
- CA Certificate Authority
- FM Federation Manager
- IdP Identity Provider
- IGTF Interoperable Global Trust Federation
- KeyVOMS Keystone-based Virtual Organization Management System
- LDAP Lightweight Directory Access Protocol
- LOA Level of Assurance
- NIST National Institute of Standards and Technology
- OIDC OpenID Connect
- OP OIDC Provider
- P2P Peer-to-Peer
- RP Relying Party
- SAML Security Assertion Markup Language
- SP Service Provider
- TLS Transport Layer Security
- UMA User-Managed Access
- VO Virtual Organization
- XACML eXtensible Access Control Markup Language
- XSEDE Extreme Science and Engineering Discovery Environment

Chapter 4. Overview

NOTE	This Engineering Report must be read in conjunction with the Testbed-14 Security Engineering Report [1]. Sections 5 and 6 can be read by themselves, but Sections 7, 8, and 9 directly reference work reported in the Security ER. The reader may also wish to review the NIST Federated Cloud Reference Architecture document, but key concepts and information from that document are given here.

Section 5 provides a brief review of basic federation concepts derived from the draft NIST Cloud Federation Reference Architecture [2].

Section 6 provides a survey of the current *federation landscape*. There are many existing tools and systems that address different aspects of federation — typically for specific, narrow use cases or application domains. The capabilities of these tools and systems are evaluated based on the NIST Reference Architecture.

Section 7 summarizes the Federated Cloud tasks, based on the complete description given in the Security ER. The goal of this summary is to "tee-up" the analysis in Section 8.

Section 8 presents an analysis of the entire Federated Cloud effort based on [2]. This includes a critique of the demonstration, as it was done, a critique of the available tooling that was used, and finally discussion of next steps to incrementally build out a more complete federation capability.

Section 9 presents overall findings and recommendations. These recommendations are based on the current *state of the art* in the federation landscape and possible areas of collaborative development.

Appendix A reports on relevant systems, projects, and events.

Appendix B gives the revision history.

Appendix C contains all references cited in the document.

Chapter 5. A Brief Review of Federation

As noted above, this ER leverages the draft NIST Cloud Federation Reference Architecture [2] to enable a critique of this federated cloud demonstration. This reference architecture document is being produced as part of the joint NIST/IEEE Federated Cloud Working Group [4]. The goal of the NIST working group is to understand the *general federation design space* and document it in a reference architecture. The goal of the IEEE P2302 working group [5] is to then identify areas of needed *federation-specific* standardization and take these through the international standardization process.

Work in identifying areas of relevant standards and gaps in standardization has already been done [6] which was taken as the starting point for the NIST/IEEE effort. It is noted that the basic concepts from the NIST Reference Architecture for managing federations have also been prototyped in alternative network communication paradigms [7]. This raises confidence that the concepts developed in the NIST Reference Architecture are fundamental with wide applicability, and not specific to or dependent on any one communication paradigm.

The NIST Reference Architecture is characterized by a number of *deployment and governance properties* that each have a range of implementation options. As such, this reference architecture and deployment/governance properties will form the basis for the analysis and critique contained herein. Here the key concepts are summarized and placed into context for the reader.

5.1. Federation in a Nutshell

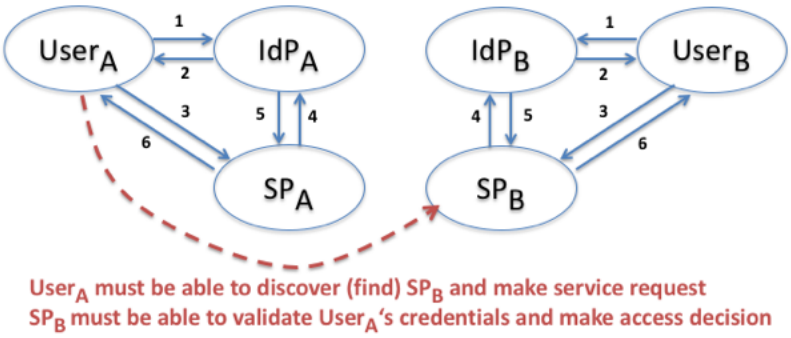

Figure 1. The Essence of Federation.

Figure 1 illustrates in a nutshell what federation actually entails. In a typical, stand-alone administrative domain, an *Identity Provider (IdP)* issues identity credentials to a *User*. When that User requests service from a *Service Provider (SP)*, that SP validates the User's credentials with the IdP. The User's request is either honored or declined based on the User's valid credentials.

A federated environment essentially crosses the boundary between administrative domains, e.g., A and B. Here a $User_A$ must be able to discover (find) any useful services that SP_B may wish to make available to a federation. When $User_A$ invokes SP_B, SP_B must have some way of validating $User_A$'s credentials and making a valid access decision. Supporting just these key functions has a large design space when one considers the issues of deployment models, governance models, and simply scale.

5.2. The NIST Federated Cloud Reference Architecture

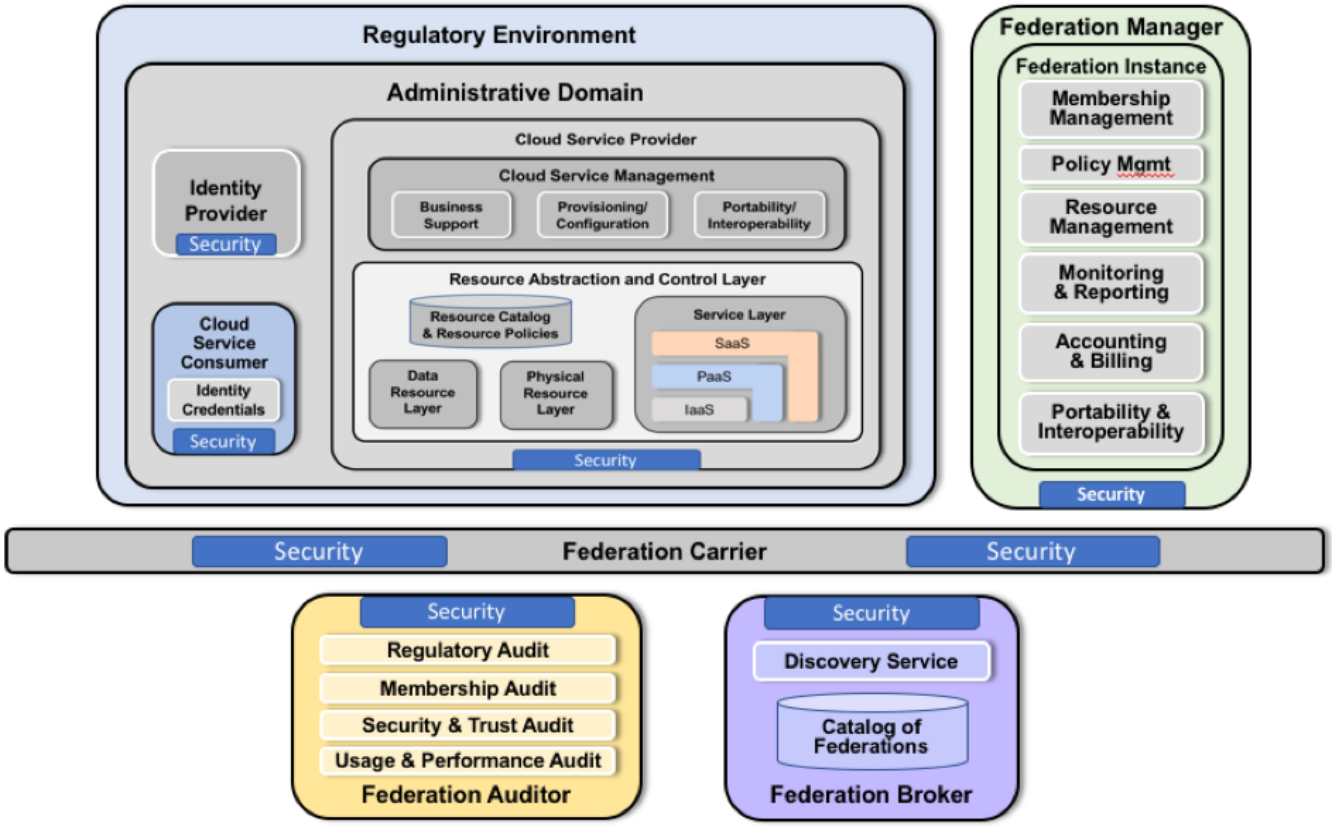

Figure 2. The NIST Federated Cloud Reference Architecture.

Figure 2 presents the draft NIST Federated Cloud Reference Architecture. This is a *conceptual model* and is not intended to be proscriptive of any specific implementation and deployment model. [2] provides extensive discussion of each of these actors and their interactions. This section will only review the major parts here.

A *Cloud Service Provider, Cloud Service Consumer*, and an *Identity Provider (IdP)* all exist within an *Administrative Domain*. An Administrative Domain will typically exist within some *Regulatory Environment*. Clearly there could be many instances of these elements.

Logically separate is a *Federation Manager (FM)*. A set of one or more FMs will manage the interactions — as per Figure 1 — over the lifecycle of a federation. Briefly this includes managing the federation members (both individual users and organizational members), the resources being shared within the federation, and the policies governing their discovery and use.

The reader interested in a more thorough discussion of these elements and their interactions is referred to [2].

5.3. Deployment Properties and Governance Functions

The NIST Cloud Federation Reference Architecture [2] also examines the range of possible *deployment* and *governance models*. These models can be expressed as a set of properties and functions, each of which has a range of implementation options:

- Deployment/Scale Properties
  - *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 becomes that encompasses more disparate organizations, the more probable it becomes that the deployment will necessarily involve heterogeneous FM implementations.
- Governance Functions
  - *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, out-of-band methods. However, as federations grow in scale, such informal methods will become impractical. More formal methods will have to be used for establishing trust.
  - *Vetting/On-Boarding New FMs:* Vetting a new FM for inclusion in a set of trusted FMs can also be done through informal, out-of-band 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 2.6.1, this could involve mapping between arbitrary types of identity credentials, or mapping to a separate federated identity. If the federation could rely on the same identity credentials being used everywhere, then the deployment and governance would be greatly simplified.
  - *Roles/Attributes:* All federations must have some set of roles or attributes whose semantics is commonly known. Smaller federations that perhaps have a relatively small, fixed set of roles or attributes could establish this common understanding through out-of-band 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 out-of-band. Clearly, 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 out-of-band. 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 are a relatively small, static set, then a common understanding of their access policies could be established by out-of-band 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, out-of-band process. In other application domains, more formal processes may be needed. Becoming a federation member may involve some agreement to "play by the rules" and support the overall goals of the federation. How strict these requirements are depends on the specific 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 through out-of-band methods. However, as federations become larger and more numerous, federations 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.

The critique of the current federated cloud demo will be based on these deployment properties and governance functions. This critique will, in turn, be used to form the recommendations for further work.

Chapter 6. The Federation Landscape

A tremendous amount of work has already been done concerning different aspects of federation. Rather than using a general approach, however, various systems and projects have built federation support for a very specific, narrow use case or application domain. This section reviews a number of those systems and projects to better understand what aspects of federation are being supported. The section also reviews some federation-related standards that have been widely used in these efforts. This will enable us to better situate the Testbed-14 Federated Cloud demo task in this landscape, and shape what kinds or recommendations are possible.

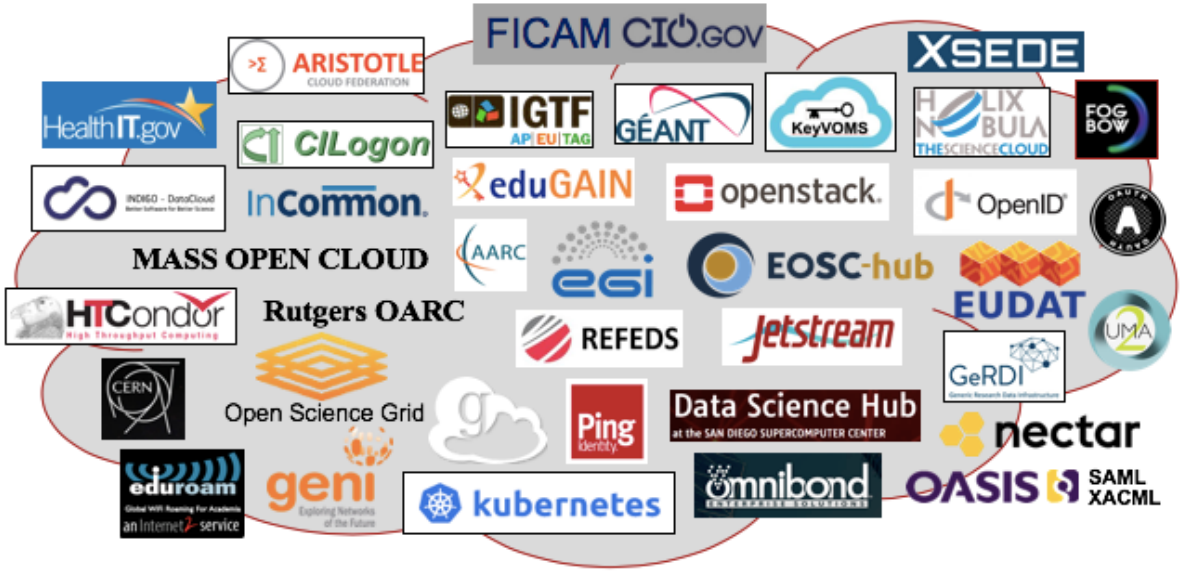

Figure 3. A Federation Logo Cloud.

The logos for these systems are collected here in the Figure 3 logo cloud. Some of these logos are for standards that are relevant to the functions and behaviors that federations must be able to support. Most of these logos are for systems or projects that address some aspects of federation, but for specific, narrow use cases or business functions. The following sections discuss these projects, systems and standards with the goals of:

a. Identifying what aspects of federation they do support,

b. How they could be used to support general federations, and

c. What functionality/capabilities would need to be added to achieve general federations.

6.1. Relevant Projects and Systems

6.1.1. EGI, EUDAT, the INDIGO-DataCloud, and the EOSC-hub

The European Open Science Cloud (EOSC), and specifically the EOSC-hub project [8], has the charter to use the European Grid Infrastructure (EGI) [9], the EUDAT Collaborative Data Infrastructure (EUDAT CDI) [10], and the INDIGO-DataCloud [11] to jointly offer data, services, and software for research across the European Union (EU). EOSC-hub will be a federated integration of these systems for EOSC, as a whole. Identifying how this integration can be done and achieving it will be a challenge.

The Enabling Grid for E-sciencE (EGEE) project received three rounds of funding from the European Commission (EC) before the creation of the EGI Foundation in 2010 to operate EGI in a sustainable manner. At its beginning EGI supported a number of grid computing software stacks to enable distributed e-science. However, as cloud computing and the on-demand provisioning of resources became an established business model, EGI has increasingly provided a cloud e-infrastructure. As EGI was already a collaboration of different resource providers, it was clear from early on that an *EGI Federated Cloud* was needed.

Rather than requiring all EGI cloud providers to run the same cloud software stack, EGI defined a *Cloud Management Stack (CMS)*. CMS provides standardized interfaces to common cloud infrastructure operations which the cloud providers must support. The Open Cloud Computing Interface (OCCI) [12] is used for Virtual Machine (VM) management, and the Cloud Data Management Interface (CDMI) [13] is used for object storage. Additional services are available for identity federation, accounting, information discovery, monitoring, etc. Identity federation is done by a Virtual Organization Management Service (VOMS) that operates as an Attribute Authority. The VOMS can issue augmented X.509 proxy certificates that are signed with the user's original certificate. Different Virtual Organizations can be set-up to support different research communities. Each Virtual Organization (VO), however, is expected to support a core set of VO services, e.g., a VOMS service for membership and authorization, a MyProxy service for managing proxy certs, and a Dashboard service, among others. While the EGI Federated Cloud primarily provides IaaS, other PaaS and SaaS-level tools are also available.

In contrast to EGI, EUDAT [10] has always focused on data management and access, and is technically not cloud-based. (It does not offer on-demand provisioning of any resources.) EUDAT provides a set of services whereby sites and users can make large data sets more discoverable, accessible, and manageable. Sites and research organizations can either *use* or *join* EUDAT. EUDAT users have a core set of services available:

- B2ACCESS: Authenticate to EUDAT using a variety of credentials
- B2SHARE: Publish, store and share scientific data sets using web-based tools
- B2STAGE: Transfer data into and out of EUDAT data nodes
- B2FIND: Search and access data in EUDAT
- B2DROP: Store, synchronize and exchange research data
- B2HANDLE: Use Persistent IDs (PIDs) to reference objects stored in EUDAT
- B2SAFE: Safely replicate data within the EUDAT network using handles
- B2HOST: Compute close to the data
- EUDAT Monitoring Service: Monitor health & status of the EUDAT services

Organizations can join the EUDAT CDI by installing and operating a core software stack. A key part of this stack is iRODS [14], a rule-based data management tool. Different iRODS installations can peer with one another and exchange data based on defined policies. Deploying the EUDAT CDI requires making *three principal federation connections* with other CDI installations:

- Federation with other data centers to enable B2SAFE data replications,
- Connection with the hierarchical B2HANDLE service to enable PID registration and resolution,

and

- Connection to the EUDAT metadata harvesting service to enable B2FIND.

EUDAT users are authenticated through B2ACCESS which can ingest a variety of credentials. B2ACCESS maintains different *service groups* of which users can be a member. Attributes are also maintained which can be granted to users. Through the attribute namespace, different attributes can be associated with specific EUDAT services. For example, if a user has an "eudat:b2stage:role=admin" attribute, they have administrative rights for the b2stage service.

While there is a B2ACCESS administration portal, the B2ACCESS service itself is based on *Unity ID Management* [15]. Unity ID is a logically centralized ID management site where users have a Unity ID. That ID is associated with an *organization* and a *project* within that organization. Unity organizations have a fixed set of roles: *owner, manager, user* and *guest*.

In contrast to EUDAT, the goal of the INDIGO-Datacloud (INtegrating Distributed data Infrastructures for Global ExplOitation) was to provide a Platform-as-a-Service layer for computing and data analysis. Based on an apparently known set of resource providers, TOSCA (Topology and Orchestration Specification for Cloud Applications) [16] specifications can be used to instantiate a set of computing resources across different OpenStack and OpenNebula providers. This can include the replication of data and the scheduling of data analytic containers in proximity to the data. All of these types of services can use the INDIGO Identity and Access Management (IAM) service. IAM provides an *identity hub* integrated with a Token Translation Service (TTS). The INDIGO Login Service can take SAML, X.509 (and OpenID Connect) credential information, and provide this on standard OpenID-Connect flows and endpoints. The TTS can convert this information into service-specific credentials for external services, such as Amazon's S3. This kind of data cloud PaaS layer can be used to support many different types of *Science Gateways*.

Usefully combining these systems will require extracting the best features of each of them and integrating them through a fundamental model that naturally supports the needed and desired integration. At a very top-level, we could say that the desired integration is to provide any number of federated environments that are managed separately but can share resources. Each of these federated environments could be considered a science gateway, much like the model for the INDIGO-Datacloud. Members of such environments will have access to data and computing resources appropriate for that science gateway community. Each of these environments could be deployed with access to a set of data management functions, much like EUDAT. Access to iRODS services for different purposes could be managed as a part of each environment. These services could be considered a *platform* and deployed on-demand. The allocation of compute resources to run the platform services and the access to the various data sets (owned by different owners) could all be managed as part of a virtual administrative domain. Actually achieving this kind of integration will certainly not be trivial, but the NIST federation model appears to be to support this goal. It is also general and extensible enough allow additional environments, platforms data sets, policies, governance, etc.

6.1.2. InCommon and eduGain

InCommon [17] is a federation-based collaboration systems that is run as part of Internet2 [18]. The core capability of InCommon is maintaining a set of *Identity Providers* and *Service Providers* across all institutions that are using InCommon to facilitate federation-based collaboration. These IdPs and

SPs are maintained in what is simply called the *metadata file*. The metadata file is a flat, eXtensible Markup Language (XML) listing of *Entity_Descriptors*, where an Entity_Descriptor is either an IdP or SP from any institution. (There are currently >6000 metadata entities.)

Several versions of the metadata file (called aggregates) are maintained for operational purposes. These include the *preview, main* and *fallback* aggregates. Clearly the *main aggregate* is considered the current operational metadata. The *preview* aggregate allows institutions to test new IdPs and SPs, while the *fallback* aggregate allows institutions to avoid using the *main* aggregate if that is causing any issues. A fourth aggregate is also available that is *IdP-only*.

Participating institutions can download the metadata file at any time. It is recommended that the file be downloaded once a day. The metadata files are not encrypted, but are all signed using the same metadata signing key and the SHA-256 digest algorithm. When downloading a metadata file, an institution must obtain an authentic copy of the InCommon Metadata Signing Certificate, whereby the signature on any downloaded aggregate can be verified.

InCommon maintains a strict process for adding new IdPs or SPs to the metadata. Submissions from institutions are vetted by a *Registration Authority*. This is done usually within one business day, but could take longer. InCommon also provides a set of traditional "human organizational" services, e.g., for resolving disputes between participating institutions regarding the operation or use of an IdP or SP.

By enabling institutions to share a flat file of IdP and SP information that is trusted, InCommon provides a valuable service that enables these institutions to collaborate in any way they choose. However, the use of a single metadata file where all participants can see and use all IdPs and SPs cannot provide key governance capabilities.

InCommon provides no way for participants to define any kind of discovery policies for the IdPs and SPs. There is no way for participants to share their endpoints (IdPs and SPs) with just a select subset of participants. There is no way for current federation participants to vet new members that will be allowed to discover and use the federation's resources. Essentially providing one centralized repository of metadata that must be periodically replicated also means that the responsiveness to change is quite slow, e.g., one business day or so.

While participating institutions may provide IdPs that they trust and will issue credentials to other institutions, there is no inherent commonality or uniformity of the identity credentials and attributes that are issued. That is to say, there is no inherent commonality or uniformity in how resource access decisions are made for institutions that wish to share similar resources for common purposes and goals.

Having a specific set of shared information for each specific federation enables all these governance issues to be addressed. Having a known set of identity credentials and attributes enables a known set of discovery and access policies to defined and agreed upon by the current federation partners. This directly facilitates the participant's common goals for the federation.

While InCommon does not directly provide such capabilities, there is nothing to prevent the participants from "rolling their own". This is precisely what has happened. An *ad hoc* ecosystem of discovery services, etc., have been built and deployed by individual institutions to provide some of the desired governance capabilities. While this represents a valuable data point, they are not built to any standards and may or may not be generalizable to other application domains.

InCommon is also part of a global *interfederation service* called *eduGain* [19] that is operated by the GÉANT network project in the EU. eduGAIN works by consolidating all of the national metadata registries into an international metadata registry. While it can be said that eduGAIN represents a truly global interfederation, it nonetheless has the same, basic business model — and drawbacks — as InCommon. It is noted that eduGAIN does maintain additional metadata whereby IdPs and SPs can be filtered according to their home federation.

6.1.3. eduRoam

eduRoam [20] is another service operated by Internet2. eduRoam essentially operates a specific type of *federated identity management* for one specific type of resource: wifi access at academic institutions around the world. As such, it is an important example of a limited type of federation for a very narrow purpose.

Each individual academic institution can manage wifi access for their students, faculty and staff using established, centralized authentication and authorization mechanisms. The goal of eduRoam is to enable users from one academic institution to travel to another eduRoam-enabled institution and get immediate wifi access. This is done by using a *tree* of *RADIUS servers* [21] that connects all eduRoam-enabled institutions. When a user from a remote institution connects a device to the local eduRoam service, the user is prompted for their home institution. A secure TLS session is then initiated with the home institution's wifi IdP over the tree of RADIUS servers. During this TLS session, the remote user authenticates to their home IdP. If this authentication home IdP is successful, the local institution grants local wifi access. It is noted that while the local institution sees the result of the authentication, at no time does the local institution see the remote user's identity credentials. These are handled directly between the user and the home IdP over the TLS session.

The eduRoam business model implies trust relationships between all institutions for the sole purpose of wifi access. Each institution must trust the IdPs at remote institutions to authenticate users. Since exactly one type of resource is being shared in the eduRoam use case, there is no need for a resource catalog or discovery policies. Also, since one type of resource is being managed, there is no need to have a richer set of authorization attributes whereby a richer set of access policies can be defined. While the tree of RADIUS servers can generally follow geospatial boundaries (e.g., nations and continents), eduRoam does not have a general mechanism whereby specific subsets of institutions can agree to grant wifi access among themselves, but not others. Nonetheless, eduRoam does represent one fundamental approach to the federated identity issue.

6.1.4. CILogon

CILogon [22] is one of those services that live within the InCommon ecosystem. CILogon enables InCommon users to obtain federated X.509 certificates based on their existing identifies at their home institutions. These certificates can then be used to access services across a much wider environment of research services at different institutions. CILogon currently supports 88 IdPs across different institutions. Initially when authenticating, a user would have to pick from a drop-down list of 88 IdPs to identify their desired IdP. This was avoided by using a text box supporting incremental search, along with cookies to remember prior selections.

The CILogon web front-end uses SAML and OpenID user authentication, whereby a standard PKCS12 file is downloaded containing their credentials. This certificate can be downloaded

completely outside the web browser by using the SAML Enhanced Client or Proxy (ECP) profile. CILogon operates three different Certification Authorities (CAs) that each support a different Level of Assurance (LOA) concerning identity. These LOAs are termed *Silver*, *Basic* and *OpenID*. (See [22] for details.) This arrangement enables relying parties to choose which types of certificates to accept based on their required LOA.

CILogon also uses OAuth to integrate external web applications. That is to say, a user can delegate a CILogon certificate to a web portal whereby it can act on the user's behalf. As an example, Globus (also known as Globus Online) can rely on specific CILogon certificate attributes to authorize large-scale GridFTP data transfers between research institutions.

CILogon works because the research institutions and projects involved have chosen to *trust* the CILogon certificates, and the specific attributes these certificates carry. This enables these organizations to manage geographically distributed users. As examples, XSEDE, the Open Science Grid (OSG), the Laser Interferometer Gravitational Wave Observatory (LIGO), and the Ocean Observatories Initiative (OOI) all use CILogon to make data and services available to their users, regardless of where their home institutions may be.

CILogon provides a valuable federated identity service that can be used to manage user bases that span different institutions and geographic areas. However, it is noted that this is all on the identity side, and not on the resource side. While different institutions that are participating in the same project could make services available to the same user base, there is no coordination about how this should be done. There is no discussion of resource discovery policy. There is also no discussion of the coordination of certificate attributes and their semantics across organizations. Attributes may need to be added to CILogon certificates to support what a specific project may need to do. This is evidently done on a case-by-case basis, rather than having a uniform way of managing different attribute sets for different "federated" environments.

6.1.5. REFEDS

REFEDS [23] (the Research and Education FEDerations group) is operated by GÉANT, the pan-European Research and Education Network. In a way similar to InCommon and eduGAIN, REFEDS enables federations of national federations. That is to say, national federations can make the IdPs and SPs available to the larger *interfederation*.

While enabling sharing and collaboration, this has led to the *Entity Type* problem. Different national federations will have different ways of expressing (or not) different services or policies. This is the *semantic interoperability* problem. To address this, REFEDS has added the notion of *Entity Categories*. The purpose of Entity Categories is to group entities that share common properties. Entity categories can also be used by IdPs to manage their Attribute Release policies. A set of attributes can be specified that are released to all SPs in a given category.

Each national federation must provide a *Federation Policy*. These Federation Policies define — at the human organization level — the *Obligations and Rights* of the *Federation Operator* and the *Federation Members*, where the members could be IdPs or SPs. These policies define things like how members may join or leave, what *level of assurance* identity credentials must have, what attributes must be released to SPs, and for what purposes SPs use the retrieved information.

An interesting aspect of REFEDS is the *Hide from Discovery* entity category. This is a special category

of IdPs that should not be shown on discovery searches. There are several reasons for this: (1) the IdP may be in test, and not be ready for production, (2) there may be a name collision or near-name collision with another IdP, or (3) the IdP may be limited in its network accessibility and is thus unsuitable to general use.

REFEDS has also looked at how to manage the distribution of federation metadata across the entire environment. (REFEDS calls this the *Metadata Flow* problem.) This can be broadly partitioned in *upstream* — how federation operators can publish their local entities to REFEDS and GÉANT — and *downstream* — how REFEDS and GÉANT can publish metadata to different members. This is managed allowing IdPs and SPs to either *Opt-In* or *Opt-Out* of pushing their metadata to REFEDS/GÉANT. This is a basic 2x2 decision matrix with four combinations: 1) Full Opt-In: entities are pushed to REFEDS/GÉANT only on specific request, 2) Full Opt-Out: entities are not pushed to REFEDS/GÉANT only on specific request, 3) IdP Opt-In, SP Opt-Out, and 4) IdP Opt-Out, SP Opt-In.

The Entity Categories in REFEDS are a step in the right direction, but are still not very general. Entity Categories are an addition to the REFEDS model. Discovery policies should be general, and not limited to Opt-in or Opt-Out. While federations could operate with disparate IdPs, enabling some degree of commonality in identity credentials could greatly improve semantic interoperability. The federation descriptions and policies should be formalized and automated as much as possible, so as to be manageable by Federation Managers. This should essentially define the *business model* for any given federation or virtual organization. As such, it should be possible to instantiate any given federation type on-demand.

6.1.6. CERN, Kubernetes Federation, and HTCondor

As a world-class, high-energy physics research institution, CERN (the European Organization for Nuclear Research) has enormous and varied computational requirements. To more flexibly support those requirements, CERN has been pursuing cloud computing. This enables CERN to allocate compute resources when they are needed. However, like many organizations, CERN has periods when resource demand exceeds capacity. In these cases, CERN will augment their own resources with commercial cloud resources.

CERN describes one approach to address this need [24], [25]. The Kubernetes container management system offers a basic Kubernetes cluster federation capability. Different Kubernetes clusters can be manually deployed on various commercial cloud providers. Using Kubernetes federation, these clusters can be jointly managed as a single entity. Once the initial cluster is deployed, the *kubefed init fed* command can be issued to start a kubernetes federation. When other clusters are started — potentially on other cloud providers — the *kubefed join* command can be issued with the appropriate arguments to join the previously initialized clusters. kubernetes ensures that the entire multi-cloud federation appears as a unified whole. Once the desired clusters have been created and connected, HTCondor [26] is used to run production workloads across the assembled infrastructure.

While this represents an interesting example of using multiple clouds to support world-class science, it is nonetheless a very simple, manually managed form of combining just compute resources that are then managed as a single administrative domain, using the usual kubernetes management tools. When a *kubefed join* command completes, there is an implicit trust relationship between the established multi-cluster and the joining cluster. The only services that are being managed are the kubernetes cluster services. Aside from the usual kubernetes authorizations, there

is no differential authorizations based on who is providing the resources, and who is asking to use them.

6.1.7. OpenStack

OpenStack is a well-funded, open source cloud software project [27]. While there are many teams working on various cloud-related services, OpenStack has a set of *core* services that any cloud software stack would have. This includes compute (Nova), storage (Swift) and networking (Neutron). *Keystone* is the security service that all other OpenStack services use to authenticate users and authorize what they can do, e.g., instantiate a VM, or create/read/write a storage container.

In Keystone v3, the Keystone team has built-out basic support for cloud federation. The goal of this work is specifically to support the *hybrid cloud* business model. This would enable one OpenStack installation to *cloud-burst* to another OpenStack installation.

To do this, the Keystone Application Programming Interface (API) was augmented with the OS-FEDERATION extension. This extension essentially added a few calls whereby a local Keystone administrator could explicitly configure their Keystone service to *federate-in* or *federate-out* with a specific remote OpenStack installation. That is to say, this establishes an explicit trust relationship between two Keystone.

Federation-out enables a local Keystone to act as an IdP for a remote Keystone. When doing so, Keystone issues PySAML2 assertions about the used owned by that Keystone. Federate-in allows users from a remote OpenStack to invoke services on the local OpenStack. When remote services are requesting identity assertions, SAML 2 or OpenID Connect can be used.

As part of OS-FEDERATION, the local Keystone administrator can define *mapping rules*. These mapping rules are used to map the attribute assertions issued by the remote user's IdP to a specific *project* and *group* within a *domain* in the local Keystone. This approach enables remote users to be granted the roles and authorizations of the group they were mapped to. These local group authorizations are then used to make local service access decisions. In this way, Keystone does not have to maintain *guest accounts* or do any type of dynamic account creation.

OpenStack represents an important example of cloud federation, but with some clear simplifying limitations. First, it is intended to manage a small, fixed set of common cloud infrastructure services, i.e., OpenStack services. It is not intended to manage arbitrary, application-level services. While Keystone v3 domains are semantically very close to virtual administrative domains or virtual organizations, and roles can be domain-specific, Keystone does not provide or enforce the common understanding or common state that is necessary to manage a distributed federation. As a case in point, the mapping rules that each Keystone administrator uses to map remote users to local groups are independently decided and implemented. Finally, it is noted that these Keystone federations are all manually managed, pair-wise federations.

6.1.8. KeyVOMS

KeyVOMS is another example of a centralized, third-party federation provider [28, 29]. KeyVOMS simply repurposed a stand-alone deployment of the OpenStack Keystone v3 server to function as a Keystone-based Virtual Organization Management Systems (KeyVOMS). Here, the concept of a

Virtual Organization (VO) developed in the grid computing era [30] was adapted to be essentially a federation instance.

Keystone v3 introduced the *domain* to its object model for managing users, projects, and access to all other OpenStack services. Domain could "own" users and projects. Users could be granted project membership by the domain administrator. The domain administrator could also grant different *roles* to users. Keystone maintains a *service catalog* for the entire installation, e.g., Nova, Swift, Cinder, etc. These service endpoints could be associated with projects, and thus made available to those project members. Whenever a user authenticated to Keystone for a specific domain and project, they would receive a cryptographically signed *AUTH_TOKEN* that included a *filtered service catalog* of just the services they were authorized to use.

Two key insights enabled the development of KeyVOMS: (1) a Keystone domain was tantamount to a VO, and (2) the Keystone service catalog could include arbitrary, application-level services — in addition to the other cloud-infrastructure services. Hence, any remote service owner that wished to participate in a VO could register a service endpoint with Keystone, and then associate it with a specific domain and project.

Whenever a user invoked one of these services (after authenticating and receiving their filtered service catalog), they needed to provide their AUTH_TOKEN, which included their roles, and also their domain (VO) and project memberships. Keystone services are built using Web Service Gateway Interfaces (WSGI). Hence, each service has a configurable pipeline of stages that can be used to condition service requests before actually handing them to the service for execution. For KeyVOMS, a *VO_AUTH* pipeline stage was added to any service that needed to be VO-enabled. Hence, whenever a request arrived from a VO member, the VO_AUTH stage would validate the user's token with KeyVOMS. The user's VO roles could then be used to make an access decision.

To make all this work, only the Keystone rule file was modified to add three roles. A *VOMS_Admin* role was added that was actually just a synonym for a Keystone *Admin*. A *VO_Admin* role enabled a user to manage anything within a single domain/VO federation. Finally, a *VO_Site_Admin* role was also added. These members were authorized to register services in KeyVOMS within a specific domain/VO.

A number of demonstrations were done using KeyVOMS. This involved VO-enabled services for managing RSS feeds, map data servers, and file servers. The demonstrations include international disaster response scenarios and managing access to resident space object tracking data from four continents.

KeyVOMS is a good example of a centralized, third-party, federation manager. Keystone was designed and built to be a service whose purpose is to manage access to service endpoints. As an open source project, it was straight-forward to leverage the Keystone object model for a much larger purpose, i.e., managing federations based on sharing service endpoints.

As a centralized system, member vetting is up to the discretion of the VOMS_Admin and VO_Admins. (On-boarding new FMs is a moot issue since KeyVOMS is a single, centralized FM.) Federated resource discovery policy was accomplished by using Keystone's endpoint filtering capability.

Initially Keystone v3 did not have domain-specific roles. All roles were visible across all domains. This is sufficient if only a small, fixed set of services are being managed across all domains — such

as a set of cloud infrastructure services. However, if arbitrary, application-level services are being managed, then each domain/VO should have its own set of roles. Domain-specific roles were finally implemented in the Pike release. Resource owners had unilateral authority to set the access policy for their resources. However, to participate in a VO, these policies would have to be based on the roles defined in a domain/VO. The Resource owner also had to explicitly trust the KeyVOMS server and VO_Admin to grant roles only to truly authorized users.

No specific accounting or auditing was done — other than the usual Keystone logging. Nonetheless, accounting and auditing could be supported. Since accounting and auditing is not really federation-specific, this could be addressed in the larger context of the OpenStack open source project.

Federation discovery was supported in a simple way — this only involved listing domains/VOs. Currently only the VOMS_Admin has authorization to list all domains/VOs. However, it would be straight-forward to allow domains/VOs to be discoverable by arbitrary users based on some set of attributes.

An important observation is that Keystone v3 could be generalized to a general-purpose, distributed P2P Federation Manager. The Keystone object model could be extended to include *virtual domains* that have additional semantics. Projects and service endpoints would be replicated among every Keystone installation that is a member of a federation, i.e., member of a virtual domain. Hence, service endpoints would eventually be discoverable by authorized domain/VO/project members. When a service invocation is made, the user's credentials could be routed back to their home Keystone for validation — in a manner similar to that used by eduRoam.

While performance and scalability issues would certainly exist, this approach is certainly feasible. It also means that internal FM capability could be realized as an optional part of a Keystone deployment. This could also be an external FM capability if the Keystone operator wished to make the federation services available to a wider set of users.

6.1.9. IGTF and the GÉANT Trusted Certificate Service

The Interoperable Global Trust Federation (IGTF) and the GÉANT Trusted Certificate Service both provide a valuable and necessary service for making federations work. Simply put, IGTF and the GÉANT TCS provide a trusted set of identity credential providers. This is done on a global scale by operating through three different regional *Policy Management Authorities*: (1) Asia-Pacific, (2) Europe, Middle East and Africa, and (3) the Americas. (By industrial standards, though, the actual resource demands by international science collaborations may be modest).

IGTF defines a minimum set of requirements and recommendations for the operation of PKI Certificate Authorities, attribute assertions, and attribute release. IGTF also maintains a set of *authentication profiles*. These specify the policies and technical requirements for classes of identity assertions, and the assertion providers that provide them. These authentication profiles are associated with different *Levels of Assurance* that balance cost and feasibility for IGTF identity providers. Details on these issues can be found at [31]. It is noted that IGTF is a peer organization to many other organizations, including the REFEDS.

The GÉANT Trusted Certificate Service (TCS) performs a similar function, but within the realm of GÉANT services. This is done by relying on a commercial Certificate Authority operator (DigiCert). Five different types of certificates are available for a variety of purposes, e.g., authenticating

servers and establishing secure sessions, identifying users, and signing code and documents. This is run in conjunction with the Trusted Academic CA Repository (TACAR). TACAR hosts the trust anchors of the Public Key Infrastructures (PKI) needed for GÉANT services such as eduRoam, eduGAIN and perfSONAR. The IGTF has accredited most of the CA root certificates hosted by TACAR.

IGTF and the GÉANT TCS both illustrate a critical function for federations to work: a set of trusted identity providers that are coupled with known governance. Organizations that wish to engage in federations must establish those common roots of trust. This could (and should) also include the common, known set of identity attributes that are integral to a federation's "business model". For federations to be truly instantiated on-demand, such trust infrastructures should be established prior to need.

6.1.10. GENI

GENI (Global Environment for Network Innovation) [32] provides a network environment that can support many different types of experimental networks at the same time. This capability has direct relevance to supporting federations.

GENI uses the notions of *projects* and *slices* to manage experimental environments. Every project has a *project lead* that can allocate one or more slices within the project. A slice is a set of *virtualized* resources that are used *in isolation* from other experiments and experimental communities. These resources (bare metal machines, virtual machines, small clouds) are owned and operated by different institutions. The resources at each institution are called an *aggregate*. Aggregates provide resources to the different slices and projects based on GENI-issued user and slice credentials.

GENI is organized with a separate *control plane* and *data plane*. When instantiating a slice, the control plane will create different Virtual Local Area Networks (VLANs) over the same physical links. The project that uses the slice cannot see any network traffic from any other experiment.

The allocation of aggregate resources is managed at each site through the *Aggregate Manager API*. This enables aggregates to advertise and allocate resources. The authorization to use these APIs is managed through three different GENI *clearinghouses*. These are operated by the GENI Program Office, EmuLab, and Planetlab. Clearinghouses issue GENI user and slice credentials, and manage project and slice membership. It is interesting to note that GENI users can authenticate to GENI using their InCommon credentials.

Aggregate owners must *federate* with one or more clearinghouses. In the GENI context, this means that aggregate owners must trust the credentials issued by these clearinghouses. It is evidently common for aggregate owners to federate with all three clearinghouses such that experimenters can get resources through any clearinghouse.

A key governance property here is that the GENI clearinghouses act as a federation *root of trust* for all entities within a GENI federation. That is to say, any member of a GENI project must trust the credentials signed by the clearinghouses. While this trust model may not work for all application domains, it does reduce the required number of trust relationships from $O(N^2)$ to $O(N)$.

GENI uses a federated environment to manage the networked aggregation of computing resources from different institutions for different experimental purposes. This is clearly not a general federation since only a narrow, fixed type of resources are being managed, i.e., networked

resources. Once a site makes aggregate resources available to GENI, there is no selective resource discovery policies. Any GENI user can look at the available resources and request their allocation.

Nonetheless, GENI represents a fundamental capability that has wide applicability. CloudLab [33] uses GENI to connect and access resources from different sites to entire, experimental clouds to be allocated on-demand. PlanetLab [34] has also looked at using GENI for similar purposes. Chameleon Cloud [35] is looking at using GENI for Identity Federation whereby users can login to either GENI or Chameleon environments. To do so, a GENI project association is necessary. Hence, the *GENI/Chameleon Federation* project was created that Chameleon users must be a member of [36].

6.1.11. Jetstream, XSEDE and Globus Auth

Jetstream [37] is a geographically distributed cloud that is hosted by Indiana University (IU), the Texas Advanced Computing Center (TACC), and the University of Arizona (UA). These sites have 100Gps connections provided by Internet2. While the operational Jetstream environment is based on OpenStack, Jetstream offers the *Atmosphere* browser-based interface. Atmosphere was originally developed to support scientific applications and workloads. Jetstream is not intended to be a high-performance computing environment, but rather to support interactive research and prototyping. Through this interface, users can launch resources (e.g., VMs) at IU or TACC. (UA provides a test environment.)

XSEDE (Extreme Science and Engineering Discovery Environment) is a distributed infrastructure, for large-scale scientific applications, that is funded by the National Science Foundation (NSF). While XSEDE nominally integrates resources across seven US institutions, it acts as a service broker across many more institutions than that. The XSEDE User Portal (XUP) provides users with a convenient, web-based interface to the XSEDE resources.

The important connection between these two systems is that Jetstream is accessible through XSEDE credentials issued by *Globus Auth* [38]. Globus Auth can broker authentication and authorization between users, identity providers and resource providers. This includes identity providers such as XSEDE, InCommon, and other commercial web service providers. A key concept is that a Globus Auth account can be a *set of identities* — a *primary* identity with one or more *linked* identities. Using this possible set of identities, Globus Auth can then act as an *authorization server* by issuing attribute assertions that enable a legitimate user to access a desired resource.

Globus Auth is implemented as an application on top of an AWS Reliable Data Service PostgreSQL database, with a web interface and API [39]. As such, Globus Auth is essentially a centralized, third-party Federation Manager. Hence, all identity providers and resource services that are managed by Globus Auth are registered in this central location. This makes it straight-forward for Globus Auth to manage *Globus groups* [40], which can even be hierarchical. Group membership can be used to manage access to data and services. A resource server can define different *scopes* for itself. This means that clients wishing to use that resource must request a *properly scoped access token* from Globus Auth.

Globus Auth relies on OpenID Connect and OAuth 2 to achieve these capabilities. This enables Globus Auth to exchange identity assertions with multiple Identity Providers, and also to delegate authorizations to third parties. As an example, Globus Auth can delegate tokens to *Globus Transfer* to accomplish third-party data transfers on behalf of a user.

Globus Auth is, again, a system that comes close to the NIST Conceptual Model. It is essentially an external federation provider with a centralized, third-party deployment model. It provides the group concept to manage sets of users and their authorizations. This is similar to a virtual domain or virtual organization, yet there are still differences. A group is something like a federation instance. A group can be created by any authenticated user. The group owner can then grant membership in the group, and determine the visibility of their group and subgroups.

Resource servers can also register with Globus Auth to use Globus Auth as an authorization server. Resources can define different scopes that are tantamount to different access policies, and they can also require that only specific IdPs can be used to grant access tokens. Groups can also be used to manage access to specific services, such as group wikis, but it is unclear how general this approach is. Globus Auth does support scientific data discovery but apparently not general resource discovery, i.e., to include arbitrary application-level services. There appears to be no general policy mechanism for managing resource discovery.

All in all, Globus Auth comes close to a general federation management tool. Its centralized implementation greatly simplifies many of its management requirements. While the group concept is close to a federation instance in the NIST Conceptual Model, it is not presented as a coherent, unified approach to managing federation instances, as in the NIST model. Also, Globus Auth is a centralized third-party federation provider. Generalizing it to a distributed implementation that can be deployed on-demand will be a major undertaking.

6.1.12. AARC

The Authentication and Authorisation framework for Research and Collaboration communities (AARC) [41] extends the capabilities of eduGAIN. A key function of eduGAIN is the maintenance of *the metadata* — a large, undifferentiated file of IdP and SP descriptors. AARC adds the notion of *groups* and actually uses the terminology of *Virtual Organizations*. The authorization to use a given service can be based on a user's specific group memberships and the group roles or attributes that have been granted to them. Recognizing the importance of semantic interoperability, work has been done to ensure that consistent syntax and semantics are maintained throughout a VO.

While a significant number of pilot projects have been done, a large part of AARC has been focused on high-level design work. A key aspect of AARC has been the choice and design of a *Token Translation Service*. Rather than taking the approach of converting "native" identity credentials into one common *VO credential* that is understood everywhere in a VO, the Token Translation Service can translate any member's credential type into any other type that is understood by a service the member wishes to invoke. That is to say, it can translate OpenID Connect into SAML assertions, or OpenID Connect into SSH keys. Even when no token type translation needs to be done, AARC defines User Attribute Services whereby *attribute enrichment* can be done, i.e., attributes can be added to a member's identity credential thereby enabling them to access desired services in the VO.

As a system built on top of eduGAIN and GÉANT, AARC adds key functionality for the flexible management of virtual domains — or VOs. AARC (and eduGAIN) are still an external federation provider, but with the added management flexibility of groups, roles and attributes, i.e., VOs. While services can base their access policies on group membership and granted attributes, there is no discussion of resource discovery policies associated with VOs. eduGAIN is based on maintaining an essentially flat metadata file of IdPs and SPs that is visible to all member institutions. In many situations, resource owners that want to make their resource available to a specific VO for a specific

purpose, may wish to define a discovery policy, such that only specific VO members with a genuine *need to know* can discover and access their resource.

The many AARC federation support services are presented in the conceptual *AARC Blueprint Architecture*. As a conceptual architecture, the functional capabilities can be allocated or colocated in different ways. The Token Translation Service (TTS), for example, could be (a) *embedded* in the AARC Proxy or with the End Services themselves, or (b) used in a *standalone* configuration. An Embedded TTS would only issue credentials for a single service or service invocation. A Standalone TTS, however, could issue credentials for many services across many organizations. In all cases, though, it appears that all member requests are proxied by AARC. This has performance latency implications but does mean that services can be made part of a VO w/o having to make any changes to the front-end of the services themselves.

While AARC does add valuable flexibility to the eduGAIN model, it continues to have some of the governance baked-in. While VOs could have one or more *VO Admins*, all VO member sites must trust these admins. As a centralized federation provider, AARC and eduGAIN can have no notion of a *site admin* that can manage their local users or resources. eduGAIN also manages the metadata, i.e., the admission of all new IdPs and SPs. Nonetheless, AARC has identified and demonstrates key capabilities in support of more general federations.

6.1.13. Fogbow

Fogbow [42] is a joint EU/Brazilian cloud infrastructure federation project. Rather than taking a multi-cloud or hybrid cloud approach, Fogbow is providing infrastructure federation across different cloud providers. Fogbow is cloud federation middleware implemented as a suite of microservices. This middleware is composed of three major components. The *Membership Service* controls which cloud providers belong to the federation, and manages network communication among them. The *Allocation Manager* at each cloud provides a well-defined common API to manage resources anywhere in the federation environment. Finally, the Allocation Manager uses an *Orchestration Manager* at each cloud provider to interact directly with that provider's API. Identity and authorization are handled by issuing *federation tokens*. When a user makes a service request that is routed to a remote provider, that federation token is translated into a local access token. Local token translation is unilaterally controlled by each cloud administrator. A *reverse tunneling service* is provided to enable users, user applications and Allocation Managers to communicate either locally or remotely, anywhere within the federated environment.

Fogbow demonstrates another interesting combination of possible federation capabilities. Using a network service to enable (and isolate) a federation's network traffic is an important part of providing communication security for a federation. For this reason, such network support will eventually be an integral part of federation management. It is not a coincidence that this network service is part of the Membership Service.

It is also important to note that the Fogbow project choose to use a common federation token that has meaning anywhere in the federation. Only when a service request gets to a local provider is this token translated into something that is locally understood. As federations become more widely used, it may be the case that federation tokens become more widely understood and trusted. Hence, such token translation may be become less of an issue.

Finally, it is noted that currently Fogbow is just an infrastructure federation. It is managing a small

set of core cloud services. However, a service catalog is under development. There may be no inherent reason why Fogbow could not be generalized to arbitrary, application-level services.

6.1.14. FICAM

The FICAM (Federal Identity, Credential and Access Management) initiative [43] is a program run by the Federal Chief Information Officers (CIO) Council. This initiative has defined a general, Federal FICAM Enterprise Architecture for managing identity and credentials to make authentication and authorization decisions at the enterprise scale. This includes a wide range of topics, such as identify proofing, credential distribution, and enterprise governance.

FICAM also identifies some federation use cases, where users from one security domain can access resources in another domain. These cases make the same assumptions as the standards they are built on. Namely that the user already knows about the service they are attempting to use. The assumption is that resource discovery has already been done by out-of-band methods. This assumes that the user and resource are operating in an *open environment*. Only after the initial user request is made does the system try to verify who is making the request, who might vouch for their identity, determine which authorizations the user may have that are relevant to the resource being requested. This also assumes that the SP and IdP involved understand the same identity and authorization attributes.

The NIST Reference Architecture has been developed to address all of these issues. This is done by establishing a virtual environment with a known boundary, i.e., a *closed environment*, wherein the necessary and desired resource management can be done by the participants. Resource discovery and discovery policies within the context of this virtual administrative domain can be customized, and are an integral part of the model. This also enables having a common understanding of identity and authorization attributes.

Aside from these federation-related issues, all of the enterprise-level identity and credential management practices defined by FICAM are nonetheless applicable for the NIST Reference Architecture. These FICAM issues should be systematically reviewed to determine how they should be used in federated environments. Unfortunately, this is out of scope for this document.

6.1.15. Ping Identity and Ping Federate

Ping Federate [44] is a commercial product of Ping Identity [45] that enables one company to provide specific corporate services to other companies. Such corporate services include things like email and travel arrangements. The key concept is that a client company continues to maintain the IdP that issues credentials for their employees (users) to use the corporate services being provided by the external company.

For example, when an employee starts their email client, that email client connects to the external email provider. The external email provider, however, redirects the employee to authenticate against their employer's IdP, e.g., an LDAP server. After successful authentication, the employee gets access to their corporate email, all of which is being managed by the external provider.

This arrangement means there is an explicit trust relationship between the external provider and a corporation's IdP. The external provider is trusting the corporate IdP to make authentication decisions. While this is a key example of a specific aspect of federation, it is noted that Ping

Federate must be configured and deployed to manage one fixed service between two fixed entities, e.g., email between a corporation and its email provider. Its main function is (a) to enable authentication decisions to be made, but without divulging employee's identity credentials to the provider, and (b) to secure the network communications between the two.

As such, Ping Federate is far from a general federation. An arbitrary number of arbitrary application-level services cannot be handled. As a result, a rich set of resource discovery and access policies are not needed. Since only one static type of service is being handled per installation, any necessary accounting or auditing becomes a much simpler issue. Nonetheless, Ping Federate is an important example of a commercially established product that addresses a key aspect of general federation.

6.1.16. And Many Others…

There are many other systems and projects that could be covered. With the efforts surveyed so far, however, we will be able to support our analysis and conclusions. Nonetheless, we will end this subsection by making at least a brief reference to some of these other systems, projects and organizations. Many others are certainly possible.

- *Rutgers Office of Advance Research Computing (OARC)* [46]. OARC is building a state-wide federation of organizations to support scientific research.
- *The SDSC Data Science Hub* [47]. The Data Science Hub at the San Diego Supercomputer Center is a community organization which, in part, is building collaborative science platforms.
- *NeCTAR and ADRC* [48]. The National eResearch Collaboration Tools and Resources (NeCTAR) project is part of the Australian Data Research Commons (ADRC) to support national and international scientific collaborations.
- *Massachusetts Open Cloud* [49]. The Massachusetts Open Cloud is building an OpenStack-based federated infrastructure cloud among state institutions with the ultimate goal of creating a public open cloud exchange.
- *Open Science Grid* [50]. OSG supports scientific collaboration by providing access to compute resources across many institutions using a Virtual Organization concept.
- *The Aristotle Cloud* [51]. The Aristotle cloud is a federation across three academic institutions that provides a common allocation and accounting system to manage the exchange and use of resources.
- *Earth System Grid Federation* [52]. The ESGF is a single federation of different nodes (sites) that can provide both compute and access to scientific data.
- *HealthIT.gov* [53]. Health IT is creating a *Health Information Exchange* with the goal of enabling patient data to be securely shared among primary care physicians, specialists, laboratories, pharmacies, and hospitals.
- *Helix Nebula* [54]. This is a public-private partnership of major European research institutions and commercial cloud providers to create a science cloud with an interoperable, secure data layer across organizations.
- *GeRDI* [55]. The Generic Research Data Infrastructure is a German project to support data sharing across scientific disciplines, and will be the German contribution to the European Open Science Cloud.

6.2. Relevant Standards

For a technical overview and description of these standards, the interested reader is referred to the OGC Testbed-14 Security Engineering Report [1]. In this document, we will present key observations concerning how these standards — while highly relevant — are fundamentally different from the kind of general federation capabilities that the NIST Federation Model provides.

6.2.1. OpenID, OAuth, OpenId Connect, and UMA

Existing standards like OpenID, OAuth, OpenID Connect and UMA all do not assume that there is any pre-existing relationship between the Relying Party/client making a request and the service/resource provider. The sequence diagrams of all these standards essentially start with the client making a request to a service. This assumes that the client has already discovered the service, by some means which is unspecified or out-of-scope for the standard. Hence, these standards do not address the issue of resource discovery policy, i.e., how a resource provider can control who can discover and access their resource/service.

As an example, OpenID Connect (OIDC) essentially combines OpenID authentication with OAuth authorization. When a User's Browser makes a request to a website (Relying Party, RP), the RP immediately makes a request to an OIDC Provider (OP). The OP provides an authorization endpoint to which the User's browser is redirected. The User is challenged to authenticate through the OP using an appropriate back-end IdP which returns a set of user attributes (on successful authentication). The User is again redirected to the RP which uses the authenticated user's attributes to request an access token from the OP. After the access token has been validated, the User reissues the original request to the RP through a third redirection, which can finally be honored.

It is noted that OAuth (and OpenID Connect) utilize a *token exchange* protocol [56] whereby a client can exchange some representation of resource authorization for an access token. This can also include token exchange to support delegation and impersonation. Since this interaction takes place between a *client* and a *Security Token Service*, this interaction can be called a *token exchange*. This can be contrasted with systems such as AARC and the INDIGO-DataCloud that use what can be more aptly called *token translation*. Since these systems proxy all communication, e.g., service invocation, the client never sees or uses the translated token. In all cases, though, a new token is derived from an existing token. The primary difference is who gets to use the new token once it is created.

While User-Managed Access (UMA) also does not assume any pre-existing relationship, it does enable (a) a Resource Owner to register a set of resource-specific policy conditions with an UMA *Authorization Server* ahead of time, and (b) a client to *pre-register* a set of *RegisteredScopes*. When doing an authorization assessment as part of a service request from a client, the UMA Auth Server must compare the client's *RegisteredScopes* with the *RequestedScopes*, along with the *TicketScopes()* associated with the resource. This is essentially testing if the client has the authorization to execute the requested operation on the given resource. After this, any applicable policies are evaluated, which is out-of-scope. The UMA standard is also silent on how such pre-registration is managed and by whom.

In reviewing the functional behavior of these standards, it is clear that a key capability that the NIST Federation Model provides is establishing and managing a set of relationships prior to any

federation member attempting to invoke a federation resource. That is to say, a federation is a set of pre-existing relationships. This pre-established context/relationship enables resource discovery, scope, and policy issues to be managed prior to use. It allows the definition of governance, i.e., who can manage what within any given federation. Such a set of pre-established relationships can be called *virtual domain* or a *virtual organization*, and can be managed by a *Federation Manager*. By explicitly managing the relationship, and doing so more completely, we can (a) address issues such as resource discovery, and (b) simplify resource invocation by avoiding numerous redirections to collect information about a user. Also, the resource owner still defines local access policy and makes final access decisions.

6.2.2. SAML

The Security Assertion Markup Language (SAML) is an XML-based standard for communicating user information concerning authentication, entitlement and attributes. As such, SAML has been used as the implementation vehicle for other tools and standards, such as Internet2 Shibboleth and OASIS WS-Security. SAML enables Single Sign-on (SSO) by enabling the communication of an authentication assertion from one site to another site. SAML can also use the same secure mechanism to communicate additional attribute information about a user. This also enables a user's attributes to be selectively released. That is to say, if a particular user attribute is not important for the Service Provider to make an authorization decision, it can be withheld.

SAML consists of *assertions, protocols, bindings* and *profiles*. Assertions can concern authentication, attributes or access decisions. SAML has several request/response protocols for communications with SPs and IdPs. This includes interactions such as authenticating a user, requesting specific attributes assertions, and terminating login sessions. When originally defined, SAML used a SOAP binding, but the HTTP Redirect binding is used far more commonly today. Several profiles are available for different usage scenarios. For example, the Web Browser SSO Profile manages communication between a SP and IdP to enable SSO from a browser. There is also a X.500/LDAP Profile for how X.500/LDAP attributes are carried within SAML assertions.

It is also useful to point out the distinctions between using SAML assertions and PKI Certificates for doing authentications. SAML can retrieve attributes on each invocation, while attributes become part of a signed PKI certificate. This can be an issue if a User's attributes can change with some frequency. It is possible to use PKI for authentication and then collect authorization information using SAML. SAML assertions can also be protected by signing them with a PKI key.

While SAML has been a very successful standard, it nonetheless does not address the issue of pre-established relationships, including resource discovery. It is assumed that the User already knows about the service that is being invoked. The primary function that SAML is used for is to provide a way for a Service Provider to redirect a User's request to a SAML Provider that can authenticate the User with an external IdP. While SAML was not intended to address authorization, it is noted that in the absence of any pre-established relationships, there is also no agreement on which attributes SAML should be used to retrieve to support access to a given service. SAML may very well have a role in supporting general federation, but it will have to be integrated with a system of one or more Federation Managers.

6.2.3. XACML and GeoXACML

The eXtensible Access Control Markup Language (XACML) standard actually defines far more than just a markup language. The authorization architecture of Policy Enforcement Points, Policy Decision Points, Policy Administration Points, and Policy Information Points provides a nice separation of concerns. This avoids the issues encountered when things like separate *Access Control Lists* are hard-coded into each application. XACML provides *Attribute-Based Access Control (ABAC)* whereby resource access control decisions are made based on the attributes associated with users. It is noted that *Role-Based Access Control (RBAC)* can be also implemented as a special case of ABAC. Clearly XACML can be used with the attributes carried in PKI certificates.

Not surprisingly, since both SAML and XACML have been developed by OASIS, there is a profile for integrating SAML2 with all versions of XACML. SAML can be used to carry XACML policies, policy queries, and policy query responses. SAML attribute assertions can be consumed by XACML PDPs, and authorization decisions can be returned.

Even though XACML was designed to be a general-purpose access control mechanism with a general-purpose Policy Language, it could not easily express geospatial policy constraints. Hence, OGC defined the GeoXACML extension to XACML. In essence, this extension defined a geometry model whereby geometric data types could be included in access policies, along with functions to test for topological relationships between geometries.

As an access control architecture, XACML (and GeoXACML) could very well be used in federated environments. If being used in a distributed deployment, the policy stores would have to be kept consistent. The policies themselves would also have to be managed in a way that consistently captures the desired governance of any given federation instance. Nonetheless, the general XACML design approach could possibly be integrated into the design and implementations of Federation Managers and how they are used by federation participants.

Chapter 7. Evaluation of the Testbed-14 Federated Cloud Tasks

Having established the fundamental functional requirements for general federations, based on the NIST Federated Cloud Reference Architecture, and having reviewed the current landscape of federation-related systems, tools, and standards, we are now in a position to evaluate the tasks in Testbed-14 concerning federation. These evaluations are directly based on the information presented in the Testbed-14 Security Engineering Report [1].

7.1. Evaluation of the Authorization Server

The essential function of the Authorization Server integrates an OpenID Connect Provider with an LDAP server. As a directory service, LDAP enables information to be cataloged. In the context of this Authorization Server (AS), End-Users, or more specifically their client applications, must be *registered* with the AS prior to the AS granting tokens to a client, or validating tokens on request from a Relying Party, in this case, a WFS Service.

It is noted that just the fact of *registering* an End-User is an important aspect of managing a federation. In a general federation, registering a user through a Federation Manager is granting a user membership in a specific federation. Granting membership in a specific federation is tantamount to *scoping* any possible authorizations. This AS, however, is not being used to managing different sets of users, i.e., federations.

The AS supports *dynamic client registration*. The use of dynamic client registration must be treated carefully in a federation. In typical usage, an authorized federation administrator will grant and revoke federation membership to different users. This identity should be logically consistent across a federated environment. If dynamic client registration must be used — say to create "disposable" clients for workflow services — care must be taken that appropriate federation membership requirements are still enforced.

It is noted that the AS is not providing any kind of resource/service discovery mechanism. It is assumed that the End-User knows the desired service to invoke through out-of-band methods. A Federation Manager would provide a discovery service whereby an End-User would be able to discover only those services they have some authorization to use.

The Relying Party (RP) — here the WFS Service — must be configured to know where the AS is and the authorization endpoints that can be used to validate tokens. This is actually consistent with the notion of a Relying Party consulting with a Federation Manager to validate credentials and make an access decision. At this point, it is possible for the RP to request information about the End-User. In the case of a well-defined federation, the necessary information to request should be well-known, and there should be no issue of attribute releasability. In the absence of a federation, there is the possibility of some semantic mismatch between which attributes a user possesses and which attributes an RP may need to make an access decision. It is noted that a federation essentially defines a *scope* for the releasability of attributes. In the absence of a well-defined federation, this scope would have to be defined by other means.

7.2. Evaluation of the Mediation Server

The Security ER considers several options for the design of the Mediation Service. The approach taken is that of a *passport* service when used in conjunction with the AS. With this approach, the "local" AS can go through the Mediation Service to access an Authorization Service in a different Security Environment. More precisely, when authenticating to the AS in Environment A, a user from Environment B can be redirected to the AS in Environment B, which returns a token to the Mediation Service in Environment A. On consent from the User, the Mediation Service retrieves desired user information from the external AS, and passes it and the token to the local AS. This token can be retrieved by the User's client and used to access services in Environment A.

It is noted that the Mediation Service functions somewhat like a centralized Federation Manager. However, it still has some aspects of a distributed implementation since there will be a Mediation Service in every Security Environment. Specifically, a Mediation Service will be replicating user attributes from one Security Environment to the local environment. There will be an issue of "who is the authoritative source", or how the consistency of the replicated attributes will be maintained.

At this point, every time a user authenticates from one environment to another, their attributes are replicated (added or updated) to the target environment. However, removing replicated attributes has not been tested, so inconsistency might be possible. This is important since this relates to the revocation (or removal) of an authorization attribute. Once removing replicated attributes has been addressed, eventual consistency should be achievable. Nonetheless, it is noted that in all distributed systems, any kind of replicated state may be temporarily inconsistent, including attributes in a federation. As more experience is gained with federated systems, the effects of such fleeting inconsistencies should be identified and addressed.

The issue of attribute release must also be clearly understood and managed. In the current implementation, the Mediation Service in Domain A will have Domain B client information that establishes the scope for that client's releasable attributes. End-users can also terminate an interaction with a Mediation Service when they are prompted with a list of attributes being shared. This raises an outstanding question: How does the use of scope in the current Mediation Service implementation relate to the notion of scope implied by the virtual organizations/federation concept? Could the scope implied by a virtual organization be used to define the scope managed by an OpenID Connect implementation? This could be a key element of subsequent investigations.

The Mediation Service is an example of something like a token translation service (in an exchange with a client). Rather than establishing a "universal token", or a "federation token", that is understood everywhere within a given federation, the Mediation Service essentially translates between any pair of Authorization Services. While this does provide a mechanism for establishing an authorization across two Security Environments, it still lacks a number of capabilities that would enable a general federation where consistent governance can be achieved.

First, there is apparently no control over which users that are being managed by an External Auth Server can authenticate to the local environment. A user can self-identify their "home" environment and which AS to use. There is no notion of the user being a member of a federation wherein their authorizations can be managed. In general, federations will need to control which users from which sites can become authorized to discover and access resources. Also, there is no common agreement concerning identity and authorization attributes or their semantics within a common context, i.e., with a federation or virtual organization.

It is noted, however, that the AS can generate specific internal user attributes when importing external user attributes. While this has not been tested in this Testbed, this might be a mechanism whereby the federation semantics noted above could be supported. Of course, these semantics should be consistent across all Authorization Servers for any given federation.

While the Mediation Service was not intended to address this issue, we must note again that there are no resource/service discovery or discovery policies involved. There is no understanding or control over which "local" services are to be made available in a larger federated environment.

7.3. Evaluation of Workflow Securitization

The Security ER makes important observations concerning workflow security. A workflow is when multiple service invocations are "chained" together. That is to say, the output of one service is the input for another service.

A key point is identifying which Security Environment a service belongs to and if the service expects a security token. This concern is driven by the fact that services do "live in" different Security Environments and could have different security constraints.

While not a trivial task, such concerns could be addressed by having a federated environment that provides a consistent, well-known security environment in which to run workflows. Rather than attempting to deal with different security environments and constraints *on-the-fly* at run-time with no established relationships, an essential part of a federation is establishing those relationships prior to need. This greatly simplifies the concerns of running a workflow. The access requirements of any service endpoint are well-defined within a federation, regardless of which site is providing the service. The workflow engine itself will be a service within the federation.

It is noted that authorization to run a particular workflow will have to be delegated from a federation member. This federation member should have authorization to invoke all services in the workflow. To properly manage accounting and auditing, it should always be possible to trace a workflow service invocation back to the originator’s identity.

While not strictly an issue for federated workflows, it is noted that the *output storage* concept proposed to address the issue of chaining secured and unsecured services actually addresses an important issue in general workflow management. A naive approach to service chaining is for a workflow manager to invoke a service, retrieve the results, and then invoke the next service. While simple to conceive and implement, this means that the workflow manager server acts as a proxy for all communication between services. All data must be communicated twice (Svc A → Workflow Manager → Svc B) rather than once (Svc A → Svc B). Depending on the size of data involved, passing references and the proper authorization addresses this overhead.

7.4. Evaluation of Federated Cloud Securitization

The OGC Testbed-14 Security ER describes the use of several existing, established standards to construct capabilities that approach some of those needed in general federations. It is possible to consider these capabilities as a "centralized" design and a "federated" design. The centralized design is essentially the Authorization Server integrated with the Mediation Service. This approach does enable external users to access resources in a specific security environment, albeit with the

limitations examined above. The "two-way" federated design essentially mirrors two centralized implementations. Each Mediation Service can contact the OpenID Connect endpoints in the other security environment.

It is noted that the "centralized" design is centralized in that it only involves one OpenID Connect and Mediation Service pair, thus enabling remote users to authenticate and use services in the target Security Environment. The "federated" case, however, illustrates two such environments peering to one another. The point is made that each of these environments has the ability to define the scope and corresponding attributes can be requested by clients from the opposite Security Environment. While very useful, it is not apparent whether these two environments can coordinate in their scope and attribute definitions. That is to say, it is not clear that a consistent federated environment is created such that similar attributes and policies are applied to similar users and services within that federated environment. This could conceivably be established manually using out-of-band information. A challenge for federations will be how to define, establish and maintain such consistent virtual environments, while minimizing reliance on manual, out-of-band methods.

Chapter 8. An Overall Evaluation

Having reviewed each of the capabilities reported on in the Security ER [1], it is clear that these two ERs are focused at different levels in the design process. The Security ER is very much focused on the concrete implementation of specific capabilities using existing standards. The Federated Cloud ER, on the other hand, is starting from a conceptual Reference Architecture and attempting to map those conceptual capabilities into more concrete ones, also using established tooling.

It is not surprising that differences in terminology would be used at these different levels. To harmonize that terminology, we offer the following definitions:

- *Security Environment:* A system that securely manages end-user information for the purpose of providing Identity Management and Access Control. These capabilities are usually achieved by using cryptographic methods, secure network design, and observing data protection regulations.
- *Administrative Domain:* A system that augments a Security Environment by defining a desired set of policies and governance for managing users and resources. This includes defining an authoritative source for policy concerning resource discovery and use.
- *Federated Environment:* The creation and management of a *Virtual Administrative Domain (VAD)* whereby the same kind of policies and governance can be used to manage users and resources within the VAD that are, in fact, coming from an arbitrary number of non-federated Administrative Domains. This depends on a *Federated Security Environment*.
- *Trust Federations:* In the NIST Reference Architecture, multiple federations among two or more Sites can be created and managed that "ride on" one or more Federation Managers. Any such set of Sites and Federation Managers can be called a *Trust Federation* since these Sites and FMs must have established trust relationships.

It is within the context of these definitions that we make the following observations.

8.1. Establishing Pre-Existing Relationships

The standards on which these capabilities are built do not assume any pre-existing relationship among the participants in an interaction. This includes OpenID Connect and OAuth. (This is also true for UMA.) Interactions (as shown in the sequence diagrams) begin with a client invoking a service. A series of redirections typically take place whereby the right IdP, Attribute Authority, Mediation Service, etc. are contacted, after which the authorization decision can be made and the service execution results returned. This may be workable up to a point for some use cases. However, for larger collaborative environments, it will be necessary to establish a consistent virtual environment across all participating sites.

This is a key concept and goal for federation. Establishing such relationships and common understanding prior to need enables consistent governance to be achieved across a federation. This involves a common understanding of the goals and "business model" of a federation by all the participants. Establishing this joint business model in a *defined environment* enables the management of which resources are to be shared, and what types of resources are to be shared, and what types of users should be authorized to access those resources. How this can be done is

spelled-out — at the conceptual level — in the NIST Federated Cloud Reference Architecture. This is accomplished through the use of one or more Federation Managers.

As discussed above (and at more length in [2]) a set of Federation Managers can be called a *Trust Federation*. FMs could be deployed and operated by different organizations, but in all cases, they must have established trust relationships. While general Trust Federations could exist (in the spirit of [31]), *domain specific* Trust Federations could also exist. As an example, there could be an *International Disaster Response Trust Federation*. Such an organization could define the business rules whereby different international organizations collaborate to respond to disasters anywhere in the world. This could involve defining roles such as medical personnel, structural engineers, first responders, etc. The necessary types of identity credentials and member on-boarding policies could also be defined. When a disaster happens, a VO could be instantiated on-demand. The members would include the appropriate agencies in the affected areas and all other responding organizations, such as various governmental agencies and NGOs, such as the International Red Cross. All of this would be enabled by having the necessary pre-existing relationships in place.

Establishing and maintaining the information that constitutes an operational federation does represent additional complexity. It is another set of "moving parts" that have to be installed and operated. However, the benefit is that consistent governance can be achieved. In practical terms, this does mean that organizations will have to define, agree upon, and enforce that common governance. Of course, there may be non-technical reasons that make this difficult, but this is fortunately outside the scope of this document.

8.2. Resource Discovery

A key aspect of these pre-existing relationships that deserves further attention is that of *resource discovery*. As noted above, the protocols for the relevant existing standards typically start with a client invoking a service. How did the client learn about the service? How does that client know they have authorization to use the service? The standards used in Testbed-14 assume that a client knows this information *a priori*.

In a federation, this information must be explicitly managed. This is a central part of the pre-existing relationships among the federation participants. In the open web, it is common for service providers to make their services widely known to any potential client. However, for a group of organizations that wish to collaborate for a specific purpose, they may wish to make only specific services available, and only to specific members of that collaboration. This includes not only making access decisions, but also simply discovering that a service exists and is available. Hence, resource cataloging, along with resource discovery and access policies, must be an integral part of managing federations.

To be fair, resource discovery was not an explicit requirement in the Testbed-14 demonstration scenarios, which centered around the Mediation Service. While the Authorization Service was based on an LDAP server, the LDAP server was not used for this purpose. This points to an important area of development and integration for future work.

The critical observation here is that a resource discovery capability must be an integral part of a federation management capability. Resource discovery has long been recognized as a critical part of distributed computing. Resource discovery is such a vast field that a survey is not possible in the scope of this engineering report. We will just review key examples.

First, we must clearly understand the scope of what is meant when a standard has a discovery service. While OpenID Connect has a discovery service, this is only intended to enable a Relying Party to discover an End-User's OpenID Provider. It is not intended to manage the discovery of arbitrary, application-level services.

Presumably LDAP could be used for federated resource discovery. While LDAP is commonly used for storing account names and passwords, as a general directory service, a directory structure could be defined whereby all necessary federation information could be maintained, including federated resources. However, for distributed deployments where multiple LDAPs are used, there would have to be some mechanism for keeping the directories consistent. It is noted that Active Directory is a Microsoft product built on top of LDAP primarily used as a centralized domain manager for Windows domain networks. Active Directory *Federation Services* provide *identity federation* to accomplish single sign-on, but currently does not support the kind of general, federated resource discovery and access control of the NIST Reference Architecture.

Presumably the OGC Catalogue Service 3.0 could also be used to manage federation information. Schemas could be defined for all necessary information. This could perhaps be standardized. The OGC Common Query Language could also be extended to cover federated resource discovery queries. OGC Catalogue Service 3.0 also has support for *distributed search* over a *catalog topology* that is recursively discovered. While this Catalog Service does seem promising, a more detailed examination should be done to verify applicability.

With regards to the security of such a catalog service for federation purposes, we can point to the *Use Case IV* of the OGC Web Service Security document (17-007). This Use Case covers a *protected service* with *private data* in a *protected catalogue* with *secure communication*. Since the goal of most federations will be to securely and privately manage collaborations among participants, this does seem like the most applicable Use Case.

8.3. Federated Identity

Another key aspect of consistent federation governance is that of *federated identity*. From a conceptual perspective, there must be some *logical* identity in the federated environment, i.e., in the virtual administrative domain, that is meaningful for authentication and making authorization decisions. Such logical identities can be functionally implemented in a number of ways.

One approach is to use a separate identity in the federated environment. In this case, a user's "home" identity is mapped into their federation identity. This is the approach taken by KeyVOMS since it is a simple, centralized, third-party VO management system. Such federation credentials must be understood and trusted by all federation participants. This could also require a service owner to modify their services to be able to validate the credential and extract the necessary information to make access decisions.

Another approach is to translate a user's "home" credential into whatever credential type the federated service understands. This is the approach taken by the AARC project which may become more of an "operational" capability in EU scientific projects. In one sense, this is what the Testbed-14 Mediation Service has done. However, rather than have a centralized universal token translation service, each Mediation Service in each Security Environment does an *in-bound* translation for the local environment. Presumably, ensuring a common understanding across different Security

Environments must be manually managed. The current standards manage "where" attributes are meaningful by using the notion of *scope*. In the NIST model, this scope for establishing the common understanding of attributes is essentially the federation itself, or virtual organization. An outstanding question is whether the notion of scope in the existing standards could be used to support scope in the context of a federation.

Chapter 9. Findings and Recommendations

NOTE	*Intent* Out of the analysis presented in the previous section, we make more specific recommendations for follow-on efforts in Testbed-15. The material presented here is summarized in Section 1.

As noted in Chapter 1, the term "federated cloud" is being used to denote collaboration of all sorts. Given the wide popularity of cloud computing, it is not uncommon for people to think "cloud federation" means the federation of anything running on a cloud. This underscores the fact that federation can be done at any level in the system stack, i.e., IaaS, PaaS, and SaaS. In fact, once in a service architecture, access to services can be managed by federation techniques, regardless of whether they are hosted on a cloud or bare metal.

The NIST Cloud Federation Reference Architecture provides a complete model for all aspects of managing federated environments. However, it is clearly recognized that many aspects of this architecture are not strictly necessary in all deployments. In fact, very simple, small-scale, manually managed federation deployments need only a core set of capabilities.

Based on the properties and functions from the NIST Reference Architecture reviewed in Section 5.3, we can identify the following as the simplest deployment models with the simplest set of governance functions. Two deployment models are the simplest but with an important distinction:

Figure 4. A Centralized, Third-Party Deployment.

1) Centralized Third-Party (Figure 4). In this deployment, there is exactly one Federation Manager (FM). This FM can manage all necessary state for multiple federations. Any number of member sites can use this FM to engage in different collaborative environments, i.e., federations. Within each federation, the member sites make different services available. Different users from each site can be granted membership in different federations and make use of the available services. It is noted that the FM could be operated by an entity that is distinct from the participating sites.

Figure 5. A Peer-to-Peer Deployment.

2) Pair-Wise P2P (Figure 5). In this deployment, there are exactly two FMs that peer to one another. Also, there are exactly two sites involved that each operate their own FM. These two sites can use their FMs to define any number of federations for collaboration purposes. Like the centralized, third-party deployment, the two sites can make different services available in different federations. Different users can be granted membership and make sure of the available services. Clearly, the ultimate notion is that any number of sites could deploy and operate an FM and peer with any number of other sites. The simplest P2P deployment, however, is just two sites.

These two deployment models are definite candidates for further work. As noted in the previous section, though, there are key capabilities that need to be addressed:

- Federated Identity
- Roles/Attributes and their Scope
- Resource/Service Discovery and Discovery Policies
- Resource/Service Access Policies

Ideally these need to be uniformly managed across any federation, i.e., a virtual administrative domain. Clearly, a centralized, third-party FM will make this easier since all state will be managed in one place. However, deploying a P2P FM may be a common method whereby an organization engages in federations. In this case, though, it may be useful to replicate some federation state among the peering FMs.

It is also noted that these fundamentally different deployment models involve different trust and governance issues. It is possible that the Third-Party FM may have a Federation Administrator for each federation. This Federation Administrator would have the authorization to grant/revoke federation membership and roles/attributes for individual users. In this case, each site must trust the FM and the Administrator to properly administer the federation.

While there could be one Federation Administrator in a P2P deployment, another governance model that may become common is that each site has its own Federation Administrator. Each site's FA would have the authorization to make local services available to a federation, and to grant/revoke federation membership and roles/attributes to local users. In this case, each site's FM and FA would have to trust each other to properly administer the federation jointly.

The key question at this point is this: *How can the tools built in Testbed-14 and reported in the Security ER [1] be leveraged to address either of these fundamental deployment models identified in the NIST Federated Cloud Reference Architecture?*

It is noted that the *two-way federation design* in the Security ER is essentially "back-to-back" deployments of the *modified OpenID Connect security environment*. Hence, the capabilities developed in Testbed-14, and as deployed in the two-way federation design, is actually closest to the pair-wise P2P deployment model.

Based on the key capabilities list itemized above, we can make the following technical observations:

- *Handling Federated Identity.* When a call is made through a Mediation Service on the remote side, the remote Auth Server is contacted, user information is retrieved and populated into the local Auth Server. This does create a usable credential in the remote security environment. However, the notion of an identity in a federated environment means that that identity only has meaning, and a specific meaning, within that federated environment. How to enforce that brings us to the next topic.
- *Use of OpenID Scope.* OpenID *scopes* can be associated with a set of attributes. In the NIST model, a federation or virtual organization is tantamount to a scope. It defines a virtual administrative domain within which the attributes have a known, common meaning. The challenge here would be to ensure that that meaning is the same in all security environments. Also, in the NIST model, federations are "first class" objects that can be created and terminated on-demand by a

federation owner. When a federation scope is created, along with a set of associated attributes, the meaning of that scope must somehow be known in other security environments. Likewise, the Federation Administrator in a given security environment must understand which scopes to associate with which users, i.e., how to grant federation membership. The requirements for these functional behaviors must be addressed in future work on general federations.

- *Managing resource registration, discovery and discovery policies.* Another key concept in the NIST model is that only specific resources/services are made discoverable and available in a federation, and only to specific federation members. This requires that resources, e.g., service endpoints, are associated (registered) with a specific scope (federation). The tools developed in Testbed-14 do not address these requirements. However, the Security ER does consider the use of the *User-Managed Access (UMA)* standard to rectify this. UMA enables a resource owner to register their services and define a policy (set of required user claims) necessary to access the service. It may be possible that such claims may include a scope, i.e., something that denotes membership in federation.
- *Minimizing Difficulty in Participating in Federations.* All of these potential solutions involve some degree of extra capabilities or services that must be stood-up and operated to realize the benefits of federation-based collaborations. From a practical perspective, the difficulty in making an existing, application service available through a federated environment should be minimized. Some implementation approaches could require that existing services be modified, such as integrating it with a *VO Policy Enforcement Point*. Rather than requiring a site to deploy and operate their own FM, FM services could be provided by an external *Federation Provider*. To possibly eliminate the need to modify existing services, it might be possible to have an FM *proxy* all service invocations. While this might prevent the need for service modifications, it does carry implications for trust and performance. When weighing different federation implementation approaches, such trade-offs should be clearly recognized and evaluated.
- *Use of Standards for What They Were Not Intended.* For completeness, we must consider the possibility that these existing standards were not designed for, nor intended for use in, general federated environments. The NIST model is based on the notion that a federation constitutes a set of pre-existing relationships, which these standards do not rely on. Attempting to use them in such a context may be attempting to "shoe-horn" them into an environment for which they are not actually suited. It may be possible to approximate well-governed federations, but *purpose-built* standards and tooling may ultimately provide better solutions.

With these observations, we can make the following recommendations:

1. *Clearly define and demonstrate how federated identity can be consistently managed and used.*

   ```
   Two fundamental options exist for handing identity in a federated environment: (1)
   issuing a federation-specific credential, and (2) doing credential translation (and
   possible client exchange) from a user's native identity credential to whatever
   credential type that a SP will understand.  The federation-specific credential must
   be trusted and understood by all participants, while the SP-specific credential is
   usually mapped to a group or project attributes.  Other variations exist.  Which
   approach will work best for geospatial applications and organizations needs to be
   determined with more experience.
   ```

2. *Clearly define and demonstrate how the scope of attributes and authorizations can be used to*

consistently manage federated environments.

```
Current standards manage the scope of meaning for attributes and authorizations,
but usefulness of this approach is constrained by the lack of any existing
relationships on which to rely.  Such attribute and authorization scopes are
explicitly and consistently managed in the NIST Reference Architecture as part of
each federation instance.  The ability to consistently manage attribute and
authorization scope through the use of Virtual Organizations or Virtual
Administrative Domains should be validated through experimental demonstration.
```

3. *Clearly define and demonstrate how resource discovery and access can be consistently managed across all participating administrative domains.*

```
A resource discovery capability needs to be an integral part of a federated
environment.  This will enable resource discovery policies to be defined and
enforced.  Discovery policies could also be related to access policies.  Catalog
services already exist.  Experience integrating such services with other federation
management services needs to be gained, along with consistently managing policies.
```

4. *Clearly define and demonstrate how federation administration is done.*

```
Since it is always necessary to "walk before you run", it is certainly reasonable
to start with developing federation demonstration scenarios that are statically
defined and managed.  However, for federations to be truly useful in operational
deployments, there must be an easy, effective way to administer those federations.
That is to say, there must be an admin function whereby federations can be created,
membership can be granted or revoked, authorization attributes can be granted or
revoked, access policies can be defined, etc.
```

5. *Strategize on the development and use of federation deployment models.*

```
The NIST Reference Architecture identifies a set of deployment models based on
having a Federation Manager (FM) that provides a set of fundamental federation
functions.  These deployment models range in size and complexity.  The two simplest
models are (a) a single, centralized third-party, and (b) a simple pair-wise
deployment.  The third-party deployment is simply one FM that enables federation
across two or more participants.  This only requires FM-to-Participant
interactions.  The pair-wise deployment is two FMs that enable federation between
two participants.  This requires FM-to-Participant interaction, and also a single
instance of FM-to-FM interaction.  It would be most feasible to start with these
two deployment models.  If done right, these could enable larger deployments to be
realized.
```

6. *Clearly identify and evaluate implementation trade-offs with regards to practical adoption issues, e.g., modifications to existing services.*

```
To be more specific, would it be possible to build a simple federation-aware,
service front-end (perhaps based on WSGI) that makes it simple to make a local
service federation-capable?  Would it be possible to do this using a Web Service
API Gateway, possibly using an open source version, such as WS02?  Would it be
possible to proxy all federated service access (perhaps as part of an FM) such that
services do not have to be modified or have a front-end added to make them
federation-capable?  In this case, the service owner would be delegating access
control to the proxy.  Could the service owner own the proxy, or at least the part
of the proxy that is controlling access to their services?
```

7. *Investigate and evaluate the benefits and necessary investment for developing purpose-built standards and tooling.*

```
The development and adoption of general federation tools will be largely an
economic and organizational culture issue, and less of a technical issue.
Organizational and application domain requirements need to be systematically
surveyed and understood.  The deployment and governance models with the widest
appeal, applicability and tractable deployments need to be understood.  Outreach
and engagement with stakeholders cannot be underdone.
```

8. *Develop awareness and understanding at the organizational level of the purpose and need for Trust Federations.*

```
While there are many smaller, technical goals for developing standardized
federation tooling, the large-scale use of these tools will depend on the
organizational understanding and cultural acceptance of this technology.  This will
require a wider understanding of what Trust Federations are and how they can
support business and mission goals.  This will be a long-term process of engagement
with different application domains and stakeholders.
```

In all of these recommendation areas, whatever specific tasks are defined for Testbed-15 would have to be properly scoped for the Testbed process where specific issues could be explored and experimentally validated. These recommendations, however, are actually relevant for all organizations working in the area of federation management. Hence, there are a number of organizations where OGC and Testbed-15 might wish to pursue external collaborations. These include:

- *The European Open Science Cloud.* A specific goal of EOSC is to deploy the EOSC-hub and enable data sharing across European science projects and organizations.
- *InCommon*. While InCommon has some inherently limitations (as noted in the survey), it is an operational federation system. This could provide a valuable environment in which to develop further capabilities for OGC standard tools.
- *OpenStack*. As an international, open source, cloud project, OpenStack has been building out support for federation. Extending the current tooling to provide a general federation capability is a definite possibility.

While collaboration with these organizations might be technical interesting and feasible, all organizational requirements and constraints for collaboration must be identified and satisfied first.

Appendix A: Relevant Systems, Projects, and Events

Listed here are a number of systems, projects and events that are relevant, in one way or another, to the Federated Cloud demonstration. Maximizing the impact of this task means engaging with potential stakeholders to get their buy-in or feedback on the future direction that the development of federation capabilities should take. In all cases, these engagements have observed the OGC IPR confidentiality policies as stated in: https://portal.opengeospatial.org/wiki/Testbed14/IPRPolicyConfidentiality

NIST Public Working Group on Federated Clouds
http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/FederatedCloudPWGFC
Request to be on NIST PWGFC Mailing List: fedcloud@nist.gov [mailto:fedcloud@nist.gov]

IEEE P2302
http://sites.ieee.org/sagroups-2302
Request to be on IEEE P2302 Intercloud Working Group List: STDS-P2302@ieee.org [mailto:STDS-P2302@ieee.org]

Open Research Cloud Alliance (ORCA)
http://www.openresearchcloud.org

OpenStack Summit Discussion Forum Topic
Supporting General Federation for Large-Scale Collaborations, May 22, 2018
Hosted by OpenStack Summit, Vancouver, BC, May 21-25, 2018
https://www.openstack.org/summit/vancouver-2018/summit-schedule/events/21786/supporting-general-federation-for-large-scale-collaborations

The Fourth International Open Research Cloud Congress
http://www.openresearchcloud.org/orc-events/vancouver-may-24-2018
May 24, 2018. Hosted by OpenStack Summit, Vancouver, BC, May 21-25
https://www.openstack.org/summit/vancouver-2018

International Advanced Research Workshop on High Performance Computing: From Clouds and Big Data to Exascale and Beyond
Workshop Session: Federation Management for Big Science — and Elsewhere
Cetraro, Italy, July 2-6, 2018
http://www.hpcc.unical.it/hpc2018

Appendix B: Revision History

Table 1. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
April 23, 2018	C. Lee	0.1	most	Initial skeleton of Fed Cloud ER derived from Example template
May 1, 2018	C. Lee	0.2	most	Additional Deimos Security Arch diagram added. Helper text commented out.
May 29, 2018	C. Lee	0.3	most	Initial ER draft
September 28, 2018	C. Lee	0.8	most	First initial Draft ER
October 29, 2018	C. Lee	0.9	most	All comments addressed. Some survey items to be completed.
November 21, 2018	C. Lee	1.0	most	Final survey items completed. Final complete review and minor revisions done.
November 29, 2018	C. Lee	1.1	Section 6.2.1	Added paragraph to make distinction between token exchange and token translation.

Appendix C: Bibliography

1. Doval, J.J., Rodríguez, H.: OGC Testbed-14: Security Engineering Report. OGC 18-026r1,Open Geospatial Consortium, https://docs.opengeospatial.org/per/18-026r1.html (2019).
2. Lee, C.A.: The NIST Cloud Federation Reference Architecture. (2018).
3. Liu, F., others: NIST Cloud Computing Reference Architecture, https://www.nist.gov/publications/nist-cloud-computing-reference-architecture, (2011).
4. Messina, J., Lee, C.A.: The NIST Public Working Group on Federated Cloud (PWGFC), https://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/FederatedCloudPWGFC.
5. IEEE: P2302 - Standard for Intercloud Interoperability and Federation (SIIF), https://standards.ieee.org/project/2302.html.
6. Lee, C.A.: Cloud Federation Management and Beyond: Requirements, Relevant Standards, and Gaps. IEEE Cloud Computing. 3, 42–49 (2016).
7. Lee, C.A., Z. Zhang, Y. Tu, A. Afanasyev, L. Zhang: Supporting Virtual Organizations Using Attribute-Based Encryption in Named Data Networking. In: Fourth IEEE International Conference on Collaboration and Internet Computing (CIC 2018), Invited paper October, 2018.
8. European Open Science Cloud: EOSC-hub, https://eosc-hub.eu.
9. EGI: European Grid Infrastructure, https://www.egi.eu.
10. EUDAT: EUDAT, https://www.eudat.eu/catalogue.
11. INDIGO-DataCloud: INDIGO-DataCloud, https://www.indigo-datacloud.eu.
12. Open Grid Forum: Open Cloud Computing Interface, http://occi-wg.org/about/specification.
13. SNIA: Cloud Data Management Interface, https://www.snia.org/cdmi.
14. iRODS: iRODS, https://www.irods.org.
15. Unity: Identity Relationship Management, http://www.unity-idm.eu.
16. OASIS: Topology and Orchestration Specification for Cloud Applications, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca.
17. The OpenStack Foundation: InCommon Federation, https://www.incommon.org.
18. Internet2: Internet2, https://www.internet2.edu.
19. eduGAIN: eudGAIN, https://www.edugain.org.
20. eduRoam: https://eduroam.org.
21. IETF: The eduroam Architecture for Network Roaming, https://tools.ietf.org/html/rfc7593.
22. Basney, J., Fleury, T., Gaynor, J.: CILogon: A Federated X.509 Certification Authority for CyberInfrastructure Logon. (2013).
23. GEANT: The Research and Education FEDerations Group.
24. Rocha, R., Filemon, C.: Multicloud Kubernetes Federation at CERN, (2018).
25. Rocha, R., Filemon, C.: CERN Experiences with Multicloud, Federated Kubernetes, (2018).

26. The Condor Project: High Throughput Condor, https://research.cs.wisc.edu/htcondor.
27. The OpenStack Foundation: OpenStack, https://www.openstack.org.
28. Lee, C.A., N. Desai, A. Brethorst: A Keystone-Based Virtual Organization Management System. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom). pp. 727–730 (2014).
29. Lee, C.A., Dimpfl, E., Cathers, S.: A Keystone-based General Federation Agent. Fifth IEEE International Workshop on Cloud Computing Interclouds, Multiclouds, Federations, and Interoperability (Intercloud 2016). 160–165 (2016).
30. I. Foster, C. Kesselman, S. Tuecke: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Int. J. High Perform. Comput. Appl. 15, 200–222 (2001).
31. IGTF: The Interoperable Global Trust Federation, https://www.igtf.net.
32. The GENI Project: GENI, https://www.geni.net.
33. The CloudLab Project: CloudLab, https://cloudlab.us.
34. The PlanetLab Project: PlanetLab, https://www.planet-lab.org.
35. The Chameleon Cloud Project: A configurable experimental environment for large-scale cloud research, https://www.chameleoncloud.org.
36. Mambretti, J.: GENI Federation with Chameleon: A Large-scale, Reconfigurable Experimental Environment for Cloud Research. GENI-FIRE Federation Workshop. (2015).
37. Indiana University Pervasive Technology Institute: Jetstream, https://jetstream-cloud.org.
38. The Globus Project: Globus Auth, https://www.globus.org/tags/globus-auth.
39. Tuecke, S., Ananthakrishnan, R., Chard, K., Lidman, M., McCollam, B., Rosen, S., Foster, I.: Globus auth: A research identity and access management platform. In: 2016 IEEE 12th International Conference on e-Science (e-Science). pp. 203–212 (2016).
40. Chard, K., Lidman, M., McCollam, B., Bryan, J., Ananthakrishnan, R., Tuecke, S., Foster, I.: Globus Nexus: A Platform-as-a-Service provider of research identity, profile, and group management. Future Generation Computer Systems. 56, 571–583 (2016).
41. The AARC Project: AARC: Authentication and Authorisation for Research and Collaboration, https://aarc-project.eu.
42. Fogbow: Federation, Opportunism and Greenness in private infrastructure-as-a-service clouds through the Barter of Wares, https://www.fogbowcloud.org.
43. FICAM: Federal Identity, Credential and Access Management Architecture, https://arch.idmanagement.gov.
44. Ping Identity: https://documentation.pingidentity.com/pingfederate.
45. Ping Identity: https://www.pingidentity.com.
46. Rutgers University: Office of Advanced Research Computing, http://oarc.rutgers.edu.
47. San Diego Supercomputer Center: Data Science Hub, https://datascience.sdsc.edu.
48. NeCTAR: NeCTAR, http://nectar.org.au.
49. Masschusetts Open Cloud: Mass Open Cloud, http://massopen.cloud.

50. Open Science Grid: Open Science Grid, http://opensciencegrid.org.
51. Aristotle Cloud Federation: Aristotle Cloud Federation, https://federatedcloud.org.
52. ESGF: Earth Systems Grid Federation, https://esgf.llnl.gov.
53. Health IT: Health IT, https://www.healthit.gov.
54. Helix Nebula: Helix Nebula: The Science Cloud, http://www.helix-nebula.eu/helix-nebula-vision.
55. The GeRDI Project: Generic Research Data Infrastructure, https://www.gerdi-project.eu.
56. IETF OAuth Working Group: Auth 2.0 Token Exchange, draft-ietf-oauth-token-exchange-15, https://tools.ietf.org/html/draft-ietf-oauth-token-exchange-15.