IEEE P2302™/D0.2
Draft Standard for Intercloud Interoperability and Federation (SIIF)

Sponsor

Standards Activities Board (C/SAB) Committee
of the
IEEE Computer Society

Approved <XX MONTH 20XX>

IEEE-SA Standards Board

Copyright © 201X by the Institute of Electrical and Electronics Engineers, Inc.
Three Park Avenue
New York, New York 10016-5997, USA

All rights reserved.

This document is an unapproved draft of a proposed IEEE Standard. As such, this document is subject to change. USE AT YOUR OWN RISK! Because this is an unapproved draft, this document must not be utilized for any conformance/compliance purposes. Permission is hereby granted for IEEE Standards Committee participants to reproduce this document for purposes of international standardization consideration. Prior to adoption of this document, in whole or in part, by another standards development organization, permission must first be obtained from the IEEE Standards Association Department (stds.ipr@ieee.org). Other entities seeking permission to reproduce this document, in whole or in part, must also obtain permission from the IEEE Standards Association Department.

IEEE Standards Association Department
445 Hoes Lane
Piscataway, NJ 08854, USA

Table 1 - History

Date	By	Comments
12/07/2011	Deepak K. Vij	Initial Draft Authoring.
02/08/2012	David Bernstein	Added substantial content.

Copyright © 2011 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Abstract: This standard creates an economy amongst cloud providers that is transparent to users and applications, which provides for a dynamic infrastructure that can support evolving business models.
Keywords: “Cloud Computing”, “Cloud Standards”, “Intercloud”, “Cloud Security”, “Root Cloud”, “Cloud Exchange”, “RDF for Cloud”, “Cloud Ontology”

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA
Copyright © 20XX by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published <XX MONTH 20XX>. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by the Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-0-XXXX-XXXX-X STDXXXXX
Print: ISBN 978-0-XXXX-XXXX-X STDPDXXXXX

IEEE prohibits discrimination, harassment and bullying. For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.
No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

In publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board

445 Hoes Lane

Piscataway, NJ 08854

USA

Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Copyright © 2011 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Introduction

This introduction is not part of IEEE P2302/D0.2, Draft Standard for Intercloud Interoperability and Federation (SIIF).

Cloud computing is a new design pattern for large, distributed datacenters. Cloud computing offers end consumers a "pay as you go" model - a powerful shift for computing, towards a utility model like the electricity system, the telephone system, or more recently the Internet. However, unlike those utilities, clouds cannot yet federate and interoperate. Such federation is called the "Intercloud". The concept of a cloud operated by one service provider or enterprise interoperating with a clouds operated by another is a powerful idea. So far that is limited to use cases where code running on one cloud explicitly references a service on another cloud.

Currently there are no implicit and transparent interoperability standards in place in order for disparate cloud computing environments to be able to seamlessly federate and interoperate amongst themselves. Proposed P2302 standards are a layered set of such protocols, called "Intercloud Protocols", to solve this interoperability challenges. The P2302 standards propose the overall design of decentralized, scalable, self-organizing federated "Intercloud" topology.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE standards should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE Standards Association web site at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

Copyright © 2011 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA web site at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/findstds/interps/index.html.

Patents

[If the IEEE has not received letters of assurance prior to the time of publication, the following notice shall appear:]

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

[The following notice shall appear when the IEEE receives assurance from a known patent holder or patent applicant prior to the time of publication that a license will be made available to all applicants either without compensation or under reasonable rates, terms, and conditions that are demonstrably free of any unfair discrimination.]

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. A patent holder or patent applicant has filed a statement of assurance that it will grant licenses under these rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses. Other Essential Patent Claims may exist for which a statement of assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Copyright © 2011 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Participants

At the time this draft standard was submitted to the IEEE-SA Standards Board for approval, the Intercloud P2302 (C/SAB/IWG/2302_WG) Working Group had the following membership:

David Bernstein, *Chair*
<Vice-chair Name>, *Vice Chair*

The following members of the balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

(to be supplied by IEEE)

Balloter1	Balloter4	Balloter7
Balloter2	Balloter5	Balloter8
Balloter3	Balloter6	Balloter9

When the IEEE-SA Standards Board approved this standard on <XX MONTH 20XX>, it had the following membership:

(to be supplied by IEEE)

<Name>, *Chair*
<Name>, *Vice Chair*
<Name>, *Past Chair*
<Name>, *Secretary*

SBMember1	SBMember4	SBMember7
SBMember2	SBMember5	SBMember8
SBMember3	SBMember6	SBMember9

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

<Name>, *NRC Representative*
<Name>, *DOE Representative*
<Name>, *NIST Representative*

<Name>
IEEE Standards Program Manager, Document Development

<Name>
IEEE Standards Program Manager, Technical Program Development

Copyright © 2011 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Contents

1. Overview 1
1.1 Scope 1
1.2 Purpose 1

2. Normative references 2

3. Definitions 2

4. Cloud to Cloud Interoperability and Federation - Intercloud 2
4.1 Introduction to Intercloud - Background and Concept 2
4.2 Introduction to Intercloud - Topology and Protocols 3

5. Intercloud Topology Elements 4
5.1 Intercloud Root 4
5.2 Intercloud Exchanges 5
5.3 Intercloud Capable Individual Clouds – Intercloud Gateways 6

6. Intercloud Protocols and Interoperability 6
6.1 Base Intercloud Protocols - XMPP 7
6.2 Intercloud Protocols Services Framework – XEP 0244 and XWS4J 8
6.3 Intercloud Protocols Encryption and Authentication – TLS and SASL and SAML 8
6.4 Intercloud Exchange Service Invocation – XMPP 10
6.5 Intercloud Protocol - Presence & Dialog with XMPP 11
6.6 Intercloud Trust Model 12
6.7 Intercloud Identity and Access Management – SAML, XACML 13
6.8 Intercloud PKI Certificates Deployment 15
6.9 Intercloud Exchange Service Discovery – XMPP Based RDF and SPARQL approach 16
6.10 Intercloud Resources Catalog Deployment 22

Annex A (informative) Bibliography 24

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Draft Standard for Intercloud Interoperability and Federation (SIIF)

IMPORTANT NOTICE: This standard is not intended to ensure safety, security, health, or environmental protection. Implementers of the standard are responsible for determining appropriate safety, security, environmental, and health practices or regulatory requirements.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading "Important Notice" or "Important Notices and Disclaimers Concerning IEEE Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

1. Overview

1.1 Scope

This standard defines topology, functions, and governance for cloud-to-cloud interoperability and federation. Topological elements include clouds, roots, exchanges (which mediate governance between clouds), and gateways (which mediate data exchange between clouds).

Functional elements include name spaces, presence, messaging, resource ontologies (including standardized units of measurement), and trust infrastructure. Governance elements include registration, geo-independence, trust anchor, and potentially compliance and audit.

The standard does not address intra-cloud (within cloud) operation, as this is cloud implementation-specific, nor does it address proprietary hybrid-cloud implementations.

1.2 Purpose

This standard creates an economy amongst cloud providers that is transparent to users and applications, which provides for a dynamic infrastructure that can support evolving business models.

In addition to the technical issues, appropriate infrastructure for economic audit and settlement must exist.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

3. Definitions

As of June 2009, please use the following introductory paragraph and there is no requirement to number definitions. Please refer to the 2009 Style Manual for updates (http://standards.ieee.org/guides/style/2009_Style_Manual.pdf).
To format terms and definitions in the IEEE-SA word template, you may NOW simply bold the "term:" and use regular body text (IEEEStds Paragraph style) for the definitions. DO NOT USE the IEEEStds Definitions or IEEEStds DefTerms+Numbers style. However, if the definitions have already been numbered with the template tool, STAFF will remove the numbering during the publication process. (NOTE: There are instances when a draft will need to number terms - please consult with an IEEE-SA editor). A new version of the template is being designed and tested to deal with the new definitions options. (2.2010).

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary: Glossary of Terms & Definitions* should be consulted for terms not defined in this clause.[1]

4. Cloud to Cloud Interoperability and Federation - Intercloud

4.1 Introduction to Intercloud - Background and Concept

Cloud computing is a new design pattern for large, distributed datacenters. Cloud computing offers end consumers a “pay as you go” model - a powerful shift for computing, towards a utility model like the electricity system, the telephone system, or more recently the Internet. However, unlike those utilities, clouds cannot yet federate and interoperate. Such federation is called the “Intercloud”. The concept of a cloud operated by one service provider or enterprise interoperating with a clouds operated by another is a powerful idea. So far that is limited to use cases where code running on one cloud explicitly references a service on another cloud.

Currently there are no implicit and transparent interoperability standards in place in order for disparate cloud computing environments to be able to seamlessly federate and interoperate amongst themselves. Proposed P2302 standards are a layered set of such protocols, called “Intercloud Protocols”, to solve the

[1] *The IEEE Standards Dictionary: Glossary of Terms & Definitions* is available at http://shop.ieee.org/.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

interoperability related challenges. The P2302 standards propose the overall design of decentralized, scalable, self-organizing federated "Intercloud" topology.

The goal of IEEE P2302 is to define the topology, protocols, functionality, and governance required to support cloud-to-cloud interoperability. The vision is an analogy with the Internet itself: in a world of TCP/IP and the WWW, data is ubiquitous and interoperable in a network of networks known as the "Internet"; in a world of Cloud Computing, content, storage and computing is ubiquitous and interoperable in a network of Clouds known as the "Intercloud".

The overall intent of P2303 is to take on a very narrow and focused slice of the overall cloud computing work and go deep as far as defining the overall topology, functions, and governance for cloud-to-cloud interoperability and federation. Topological elements include clouds, roots, exchanges (which mediate governance between clouds), and gateways (which mediate data exchange between clouds). Functional elements include name spaces, presence, messaging, resource Ontologies, and trust infrastructure.

4.2 Introduction to Intercloud - Topology and Protocols

Cloud instances must be able to dialog with each other. One cloud must be able to find one or more other clouds, which for a particular interoperability scenario is ready, willing, and able to accept an interoperability transaction with and furthermore, exchanging whatever subscription or usage related information which might have been needed as a pre-cursor to the transaction. Thus, an Intercloud Protocol for presence and messaging needs to exist which can support the 1-to-1, 1-to-many, and many-to-many use cases. The discussion between clouds needs to encompass a variety of content, storage and computing resources.

The vision and topology for the Intercloud we will refer to is an analogy with the Internet itself: in a world of TCP/IP and the WWW, data is ubiquitous and interoperable in a network of networks known as the "Internet"; in a world of Cloud Computing, content, storage and computing is ubiquitous and interoperable in a network of Clouds known as the "Intercloud"; this is illustrated in Figure 1.

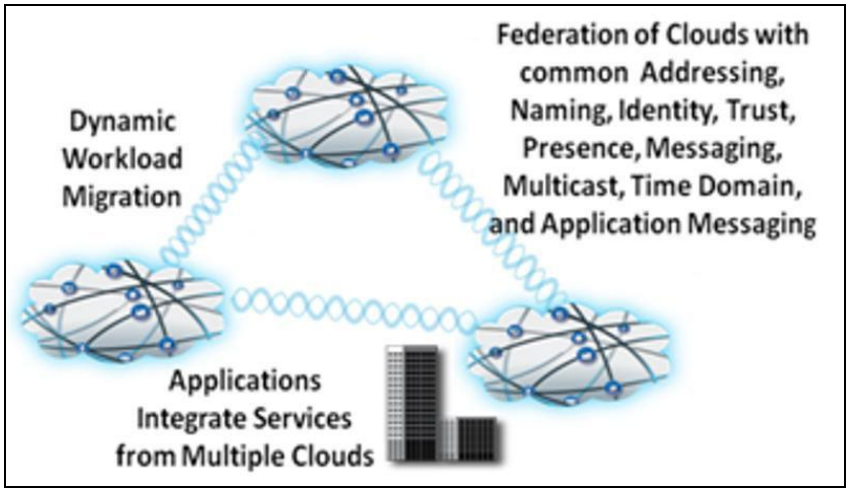

Figure 1. **The Intercloud Vision**

The reference topology for realizing this vision is modeled after the public Internet infrastructure. Again, using the generally accepted terminology, there are Public Clouds, which are analogous to ISP's. There are Private Clouds which is simply a Cloud which an organization builds to serve itself. There are Intercloud Exchanges (analogous to Internet Exchanges and Peering Points) where clouds can interoperate, and there

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

is an Intercloud Root, containing services such as Naming Authority, Trust Authority, Directory Services, and other "root" capabilities. It is envisioned that the Intercloud root is of course physically not a single entity, a global replicating and hierarchical system similar to DNS would be utilized.

All elements in the Intercloud topology contain some gateway capability analogous to an Internet Router, implementing Intercloud protocols in order to participate in Intercloud interoperability. We call these Intercloud Gateways. The entire topology is detailed in Figure 2.

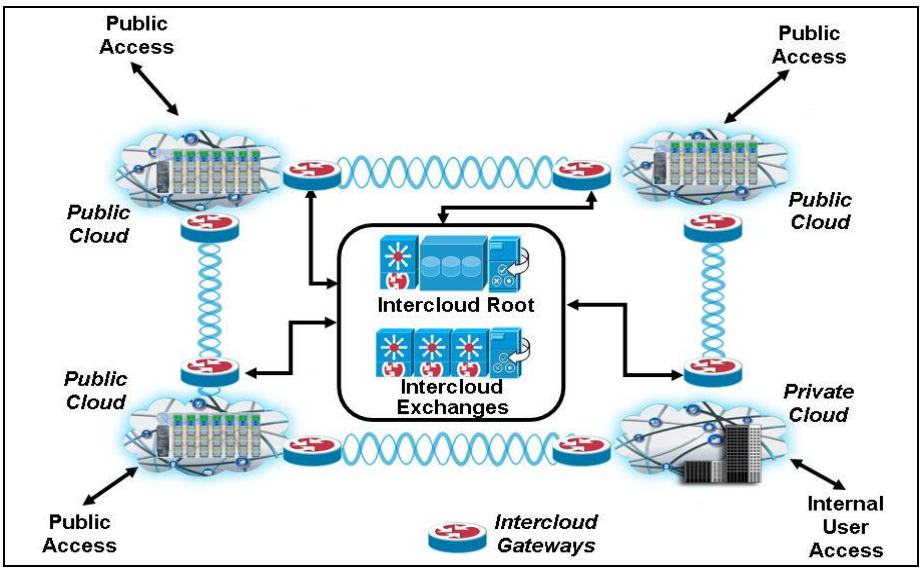

Figure 2. Reference Network Intercloud Topology and Elements

The Intercloud Gateways would provide mechanism for supporting the entire profile of Intercloud protocols and standards. The Intercloud Root and Intercloud Exchanges would facilitate and mediate the initial Intercloud negotiating process among Clouds.

Once the initial negotiating process is completed, each of these Cloud instance would collaborate directly with each other via a protocol and transport appropriate for the interoperability action at hand; for example, a reliable protocol might be needed for transaction integrity, or a high speed streaming protocol might be needed optimized for data movement over a particular link.

5. Intercloud Topology Elements

5.1 Intercloud Root

There will be a community governed set of Intercloud Root providers who will act as brokers and host the Cloud Computing Resource Catalogs for the Intercloud computing resources, similar to DNS would be utilized.

They would be governed in a similar way in which DNS and Top Level Domains by an organization such as ISOC or ICANN. There would also be a responsible for mediating the trust based federated security among disparate clouds by acting as Security Trust Service providers using standards such as SASL and SAML. This might be the IGTF.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

As part of the proposed topology, the Intercloud Root providers would be federated in nature. Each of these federated nodded in the overall Intercloud topology will independently manage the "root" capabilities such as Cloud Resources Directory Services, Trust Authority, Presence Information etc. Additionally, each Intercloud Root instance will be associated with its affiliated Exchanges by defining the affiliation relationship as part of the Intercloud "root" instance.

As part of the proposed topology, the Intercloud Root providers would be federated in nature. Each of these federated nodded in the overall Intercloud topology will independently manage the "root" capabilities such as Cloud Resources Directory Services, Trust Authority, Presence Information etc. Additionally, each Intercloud Root instance will be associated with its affiliated Exchanges by defining the affiliation relationship as part of the Intercloud "root" instance.

In order for the Intercloud capable Cloud instances to federate or otherwise interoperate resources, a Cloud Computing Resources Catalog system is necessary infrastructure. This catalog is the holistic and abstracted view of the computing resources across disparate cloud environments. Individual clouds will, in turn, will utilize this catalog in order to identify matching cloud resources by applying certain Preferences and Constraints to the resources in the computing resources catalog.

The technologies to use for this are based on the Semantic Web which provides for a way to add "meaning and relatedness" to objects on the Web. To accomplish this, one defines a system for normalizing meaning across terminology, or Properties. This normalization is called Ontology. Cloud Computing resources can be described, cataloged, and mediated using Semantic Web Ontologies, implemented using RDF techniques.

Due to the sheer size of global resources ontology information, a centralized approach for hosting the repository is not a viable solution due to the fact that one single entity can not be solely responsible and burdened with this humongous and globally dispersed task. Instead, Intercloud Roots will host the globally dispersed computing resources catalog in a federated manner.

One important difference for the cloud capabilities is that the root systems would be replicating and hierarchical, but would not replicate in a hierarchical fashion.

The roots replicate "sideways" and "upwards" using Peer to Peer technology in order to scale. The sideways replication would be "master node" replication, as is common in P2P topologies, whereas the upwards replication would be to multiply interconnected peer replication, also as is common in P2P topologies.

The Intercloud Root instances will work with Intercloud Exchanges to solve the n^2 problem by facilitating as mediators for enabling connectivity among disparate cloud environments. This is a much preferred alternative to each cloud vendor establishing connectivity and collaboration among themselves (point-to-point), which would not scale physically or in a business sense.

From Intercloud topology perspectives, Intercloud Roots will provide PKI CA root like functionality. According to the current PKI based trust model, once the CA authorizes the certificate for an entity, the entity is either trusted or non-trusted. However, in the cloud computing environment, especially in the Intercloud environment, this model needs to be extended to have "Trust Zone" to go along with the existing PKI based trust model. Intercloud exchanges will be responsible for the "Trust Zone" based trust model layered on top of the PKI certificate based trust model.

5.2 Intercloud Exchanges

Intercloud Exchange providers will facilitate the negotiation dialog and collaboration among disparate heterogeneous cloud environments, working in concert with Intercloud Root instances as described previously. Intercloud Root instances will host the root servers containing all presence information for

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Intercloud Root instances, Intercloud Exchange Instances, and Internet visible Intercloud capable Cloud instances. Intercloud Exchanges will host second-tier servers.

Intercloud Exchanges, in turn, will leverage the globally dispersed resources catalog information hosted by federated Intercloud Roots in order to match cloud resources by applying certain Preferences and Constraints to the resources. From overall topology perspectives, Intercloud Exchanges will provide processing nodes in a peer-to-peer manner on the lines of DHT overlay based approach in order to facilitate optimized resources match-making queries. Ontology information would be replicated to the Intercloud Exchanges (DHT overlay nodes) from their affiliated Intercloud Roots using a "Hash" function.

Nodes within the DHT overlay system are uniformly distributed across key space and maintain list of neighbors in the routing table. Each peer in the DHT overlay system is responsible for some part of the overall key space and maintains additional routing information to forward queries to neighboring peers. As the number of machines taking part in the network and the amount of shared information evolve, peers opportunistically organize their routing tables according to a dynamic and distributed binary search tree.

Exchanges are the custodians/brokers of "Domain based Trust" systems environment for their affiliated cloud providers. Cloud providers rely on the Intercloud exchanges to manage trust. As part of the identification process for matching desired cloud resources, individual consumer cloud provider will signify the required "Trust Zone" value such as "Local Intercloud Exchange" domain or "Foreign Intercloud Exchange". Depending on the desired "Trust Zone" value, for example, one Intercloud provider might trust another provider to use its storage resources but not to execute programs using these resources. Intercloud Exchanges, in turn, will utilize the desired "Trust Zone" value as part of the matching Preferences and Constraints in order to identify matching cloud resources.

5.3 Intercloud Capable Individual Clouds – Intercloud Gateways

Individual Intercloud capable Clouds will communicate with each other, as clients, via the server environment hosted by Intercloud Roots and Intercloud Exchanges. These clouds connect to the Intercloud via an Intercloud Gateway.

The gateway capability is analogous to an Internet Router, implementing Intercloud protocols in order to participate in Intercloud interoperability.

The Intercloud Gateways would provide mechanism for supporting the entire profile of Intercloud protocols and standards. The Intercloud Root and Intercloud Exchanges would facilitate and mediate the initial Intercloud negotiating process among Clouds.

Once the initial negotiating process is completed, each of these Cloud instance would collaborate directly with each other via a protocol and transport appropriate for the interoperability action at hand; for example, a reliable protocol might be needed for transaction integrity, or a high speed streaming protocol might be needed optimized for data movement over a particular link.

6. Intercloud Protocols and Interoperability

The vision is an analogy with the Internet itself: in a world of TCP/IP and the WWW, data is ubiquitous and interoperable in a network of networks known as the "Internet"; in a world of Cloud Computing, content, storage and computing is ubiquitous and interoperable in a network of Clouds.

As shown, it is modeled after the public Internet infrastructure. Again, using the generally accepted terminology,

- There are Public Clouds, which are analogous to ISP's.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

- There are Private Clouds which is simply a Cloud which an organization builds to serve itself.

- There are Intercloud Exchanges (analogous to Internet Exchanges and Peering Points – called Brokers in the NIST Reference Architecture) where clouds can interoperate,

- There is an Intercloud Root, containing services such as Naming Authority, Trust Authority, Directory Services, and other "root" capabilities. It is envisioned that the Intercloud root is of course physically not a single entity, a global replicating and hierarchical system similar to DNS would be utilized.

6.1 Base Intercloud Protocols - XMPP

The Intercloud Gateways would provide mechanism for supporting the entire profile of Intercloud protocols and standards utilizing a common transport such as XMPP. The Intercloud Root and Intercloud Exchanges would facilitate and mediate the initial Intercloud negotiating process among Clouds.

One cloud must be able to find one or more other clouds, which for a particular interoperability scenario is ready, willing, and able to accept an interoperability transaction with and furthermore, exchanging whatever subscription or usage related information which might have been needed as a pre-cursor to the transaction. Thus, an Intercloud Protocol for presence and messaging needs to exist which can support the 1-to-1, 1-to-many, and many-to-many Cloud to Cloud use cases.

Extensible Messaging and Presence Protocol (XMPP) will be used as the base protocol for Intercloud communications.

- See Extensible Messaging and Presence Protocol (XMPP): Core, and related other RFCs at http://xmpp.org/rfcs/rfc3920.html

- See XMPP Standards Foundation at http://xmpp.org/

XMPP is a set of open XML technologies for presence and real-time communication developed by the Jabber open-source community in 1999, formalized by the IETF in 2002-2004, continuously extended through the standards process of the XMPP Standards Foundation. XMPP supports presence, structured conversation, lightweight middleware, content syndication, and generalized routing of XML data.

For Intercloud protocols, XMPP is a viable control plane presence and dialog protocol. XMPP root services would be located in the Intercloud Root in the topology explained above.

XMPP defines protocols for communicating between groups of entities which register with an XMPP server. Registration is dynamic and provides the basis for Presence. In a large implementation, such as the global Intercloud envisioned herein, XMPP servers are connected together. This is identical to the way service providers connect XMPP servers together already supporting cross-domain Instant Messaging. In this way, XMPP facilitates both presence and many-to-many messaging across service provider domains. XMPP messages are extensible, and can be used to carry messages of different types. For example, an XMPP Message can carry Instant Messaging (IM) type traffic. We will be using a Cloud extension to XMPP.

XMPP servers support encrypted communication (SASL (Simple Authentication and Security Layer) and TLS (Transport Layer Security)) with the option to restrict XMPP servers to accept only encrypted client-to-server and server-to-server connections.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

6.2 Intercloud Protocols Services Framework – XEP 0244 and XWS4J

The Intercloud Protocols must include a Services Framework layer on top of XMPP, analogous to the HTTP-based Web service technologies, like the Simple Object Access Protocol (SOAP) and REpresentational State Transfer (REST) services. Today these are the most common technologies for interfaces on a services framework.

However, the intrinsically synchronous HTTP protocol is unsuitable for time-consuming operations, like computationally demanding database lookups or calculations, and server timeouts are common obstacles. A very common workaround is to implement a ticketing mechanism in the service, where the client receives a ticket that can be used to repetitively poll for results and is highly inefficient.

XMPP based services, on the other hand, are capable of asynchronous communication. This implies that clients do not have to poll repetitively for status, but the service sends the results back to the client upon completion. As an alternative to RESTful or SOAP service interfaces, XMPP based services are ideal for lightweight service scenarios.

To address this issue, the Intercloud protocols leverage a series of XMPP extensions (XEP series) defined by XMPP standards foundation. One of these extensions is XEP-0244.

- See XEP-0244: IO Data, at http://xmpp.org/extensions/xep-0244.html

Extension XEP-0244 provides a "services" framework on top of base XMPP, named IO Data, which was designed for sending messages from one computer to another, providing a transport for remote service invocation and attempting to overcome the problems with SOAP and REST. This is the services framework on for the Intercloud protocols.

The Intercloud services framework reference implementation for the IO Data XEP, XMPP Web Services uses Java (called xws4j).

- See XMPP Web Services for Java (XWS4J), at http://sourceforge.net/projects/xws4j/

6.3 Intercloud Protocols Encryption and Authentication – TLS and SASL and SAML

XMPP includes a method for securing the XML stream from tampering and eavesdropping. This channel encryption method makes use of the Transport Layer Security (TLS) protocol, along with a "STARTTLS" extension that is modeled after similar extensions for the IMAP and POP3 protocols.

- See The Transport Layer Security (TLS) Protocol, at http://tools.ietf.org/html/rfc5246
- See Internet Message Access Protocol (IMAP), at http://tools.ietf.org/search/rfc3501
- See Post Office Protocol (POP3), at http://tools.ietf.org/html/rfc1939

Intercloud enabled Clouds use TLS to secure the streams prior to attempting the completion of SASL based authentication negotiation. SASL is a method for authenticating a stream by means of an XMPP-specific profile of the protocol.

- See Simple Authentication and Security Layer (SASL), at http://tools.ietf.org/html/rfc4422

SASL provides a generalized method for adding authentication support to connection-based protocols. The Intercloud protocols will use the XMPP profile for SASL. Currently, the following authentications methods

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

are supported by XMPP-specific profile of SASL protocol: “DIGEST-MD5”, “CRAM-MD5”, “PLAIN”, and “ANONYMOUS”.

SAML provides authentication in a federated environment and will be used as such in the Intercloud protocols.

- See Security Assertion Markup Language (SAML), at http://saml.xml.org/saml-specifications

The support for SAML in XMPP-specific profile of SASL protocol specifies a SASL mechanism for SAML 2.0 that allows the integration of existing SAML Identity Providers with applications using SASL.

The following sample shows the data flow for a Cloud securing a stream to an Intercloud Root, using STARTTLS. It also shows SAML2.0 based authentication steps.

Step 1: Cloud starts stream to Intercloud Root:

```
<stream:stream
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'
    to='intercloudexchg.com'
    version='1.0'>
```

Step 2: Intercloud Root responds by sending a stream tag to client:

```
<stream:stream
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'
    id='cloud1_id1'
    from='intercloudexchg.com'
    version='1.0'>
```

Step 3: Intercloud Root sends the STARTTLS extension to Cloud:

```
<stream:features>
  <starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'>
    <required/>
  </starttls>
</stream:features>
```

Step 4: Cloud sends the STARTTLS command to Intercloud Root:

```
<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

Step 5: Intercloud Root informs Cloud that it is allowed to proceed:

```
<proceed xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
```

Step 5 (alt): Intercloud Root informs Cloud that TLS negotiation has failed and closes both stream and TCP connection:

```
<failure xmlns='urn:ietf:params:xml:ns:xmpp-tls'/>
</stream:stream>
```

Step 6: Cloud and Intercloud Root attempt to complete TLS negotiation over the existing TCP connection.

Step 7: If TLS negotiation is successful, Cloud initiates a new stream to Intercloud Root:

```
<stream:stream
    xmlns='jabber:client'
    xmlns:stream='http://etherx.jabber.org/streams'
    to='intercloudexchg.com'
    version='1.0'>
```

Step 7 (alt): If TLS negotiation is unsuccessful, Intercloud Root closes TCP connection.

Step 8: Intercloud Root responds by sending a stream header to Cloud along with any available stream features:

```
<stream:stream
    xmlns='jabber:client'
```


Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

```
    xmlns:stream='http://etherx.jabber.org/streams'
    from='intercloudexchg.com'
    id=' cloud1_id2'
    version='1.0'>
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    <mechanism>DIGEST-MD5</mechanism>
    <mechanism> CRAM-MD5</mechanism>
    <mechanism>PLAIN</mechanism>
    <mechanism>ANONYMOUS</mechanism>
    <mechanism>EXTERNAL</mechanism>
    <mechanism>SAML20</mechanism>
  </mechanisms>
</stream:features>
```

Step 9: Cloud continues with SASL based authentication negotiation.

Step 10: Cloud selects an authentication mechanism:

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl' mechanism='SAML20'/>
```

Step 11: Intercloud Root sends a BASE64 encoded challenge to Cloud in the form of an HTTP Redirect to the SAML assertion consumer service with the SAML Authentication Request as specified in the redirection URL.

- See The Base16, Base32, and Base64 Data Encodings, at http://www.ietf.org/rfc/rfc4648.txt

Step 12: Cloud sends a BASE64 encoded empty response to the challenge:

```
<response xmlns='urn:ietf:params:xml:ns:xmpp-sasl'> = </response>
```

Step 13: The Cloud now sends the URL to the local Intercloud Gateway for processing. The Intercloud Gateway engages, just like a browser would, in a normal SAML authentication flow (external to SASL), like redirection to the Identity Provider. Once authenticated, the Intercloud Gateways is passed back to the Cloud who sends the AuthN XMPP response to the Intercloud Root, containing the subject-identifier and the "jid" as an attribute.

Step 14: Intercloud Gateway informs Cloud of successful authentication:

```
<success xmlns='urn:ietf:params:xml:ns:xmpp-sasl'/>
```

Step 14 (alt): Intercloud Gateway informs Cloud of failed authentication:

```
<failure xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
<temporary-auth-failure/>
</failure>
</stream:stream>
```

6.4 Intercloud Exchange Service Invocation – XMPP

The following request invokes a SPARQL query over an XMPP connection to the Intercloud Root, to apply preferences and constraints to the resources in the computing semantics catalog for determining if the service description on another Cloud meets the constraints of the first Cloud's interest. We use IO Data XEP, XMPP Web Services for Java (xws4j):

```
<iq type='set'
    from='user@cloud1.org'
    to='service.intercloudexchg.com'
    id='cloud1_id1'>
    <command xmlns=
      'http://jabber.org/protocol/commands'
      node='constraint_catalog_resources'
      action='execute'>
      <iodata xmlns=
      'urn:xmpp:tmp:io-data' type='input'>
      <in>
      <constraints xmlns='http://www.csp/resOntology'>
        <constraint>
         <attribute>availabilityQuanity </attribute>
```


Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

```
          <value>99.999</value>
         </constraint>
         <constraint>
          <attribute>replicationFactor</attribute>
          <value>5</value>
         </constraint>
         <constraint>
          <attribute>tierCountries</attribute>
          <value>JAPAN</value>
         </constraint>
         <constraint>
          <attribute>StorageReplicationMethod
          </attribute>
          <value>AMQP</value>
         </constraint>
         <constraint>
          <attribute>InterCloudStorageAccess
          </attribute>
          <value>NFS</value>
         </constraint>
       </constraints>
       </in>
     </iodata>
   </command>
</iq>
```

The above service invocation request results into the following result set:

```
<iq type='result'
    from='service.intercloudexchg.com'
    to='user@cloud1.org'
    id='cloud1_id1'>
  <command xmlns=
    'http://jabber.org/protocol/commands'
    sessionid='RPC-SESSION-0000001'
    node='constraint_catalog_resources'
    status='completed'>
    <iodata xmlns=
    'urn:xmpp:tmp:io-data' type='output'>
      <out>
        <matchingClouds
         xmlns=' http://www.csp/resOntology'>
           <cloudName>cloud2</cloudName>
           <cloudName>cloud5</cloudName>
        </matchingClouds>
      </out>
    </iodata>
  </command>
</iq>
```

The example shows how the service invocation works inside of an XMPP conversation.

6.5 Intercloud Protocol - Presence & Dialog with XMPP

Next, assume that the requesting cloud has found a target cloud with which to interwork. It must now turn directly to the target cloud and dialog with it. This last section describes such a cloud-to-cloud presence and dialog scenario. The code sample is based on Google AppEngine XMPP JAVA API set.

- See *Google App Engine, The XMPP Java API*, at http://code.google.com/appengine/docs/java/xmpp/

The following code sample tests for a service availability then sends a message as part of the collaboration dialog:

```
// ...
        JID jid = new JID("user@cloud2.com");
        String msgBody = "Cloud 2, I would like to use your resources for storage replication using
AMQP over UDT protocol.";
        Message msg = new MessageBuilder()
            .withRecipientJids(jid)
            .withBody(msgBody)
            .build();

        boolean messageSent = false;
```


Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

```
        XMPPService xmpp = XMPPServiceFactory.getXMPPService();
        if (xmpp.getPresence(jid).isAvailable()) {
            SendResponse status = xmpp.sendMessage(msg);
            messageSent = (status.getStatusMap().get(jid) == SendResponse.Status.SUCCESS);
        }

        if (!messageSent) {
            // Send an email message instead...
        }
```

Step 2: The following code sample shows how the recipient Cloud responds back to the chat message as part of the collaboration dialog.

```
/* Handler class for all XMPP activity. */
public class XmppReceiverServlet extends HttpServlet {
  private static final XMPPService xmppService =
      XMPPServiceFactory.getXMPPService();

  public void doPost(HttpServletRequest request, HttpServletResponse response)
      throws IOException {
    Message message = xmppService.parseMessage(request);

    Message reply = new MessageBuilder()
        .withRecipientJids(message.getFromJid())
        .withMessageType(MessageType.NORMAL)
        .withBody("Cloud 1, please go ahead and use my resources for storage replication using
AMQP/UDT protocol.")
        .build();

    xmppService.sendMessage(reply);
  }
```

6.6 Intercloud Trust Model

At a basic level, proposed Intercloud topology subscribes to the PKI based trust model. In accordance to the PKI trust model, the Intercloud Root systems will serve as a Trust Authority

In the current trust architecture, a Certificate issued by a Certificate Authority (CA) [28], must be utilized in the process to establish a trust chain. The CAs which provides certificates must provide them in specific formats, undergo annual security audits by certain types of accountancy corporations, and conform to a host of best practices known as Public Key Infrastructure [29]. These requirements can vary by country.

- See Certificate Authority, http://en.wikipedia.org/wiki/Certificate_authority

- See Internet X.509 Public Key Infrastructure, Certificate Policy and Certification Practices Framework, http://tools.ietf.org/html/rfc3647

Certificates not only need to identify the clouds, but the resources the clouds offer, and the workloads that the cloud wishes federation with other clouds, to work upon. Where web sites are somewhat static, and a certificate can be generated to trust the identity of that web site, cloud objects such as resources and workloads are dynamic, and the certificates will have to be generated by a CA. As per the architecture of the CA, the Intercloud Exchange will need to be the intermediate CA, acting in a just-in-time fashion to provide limited lifetime trust to the transaction at hand.

From Intercloud topology perspectives, Intercloud Roots will provide static PKI CA root like functionality. On the other hand, Intercloud exchanges will be responsible for the dynamic “Trust Level” model layered on top of the PKI certificate based trust model. The overall trust model is more of a “Domain based Trust” model. It divides the cloud provider computing environment into several trust domains. Nodes in the same domain usually are much more familiar with each other, they have a higher degree of trust for each other.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

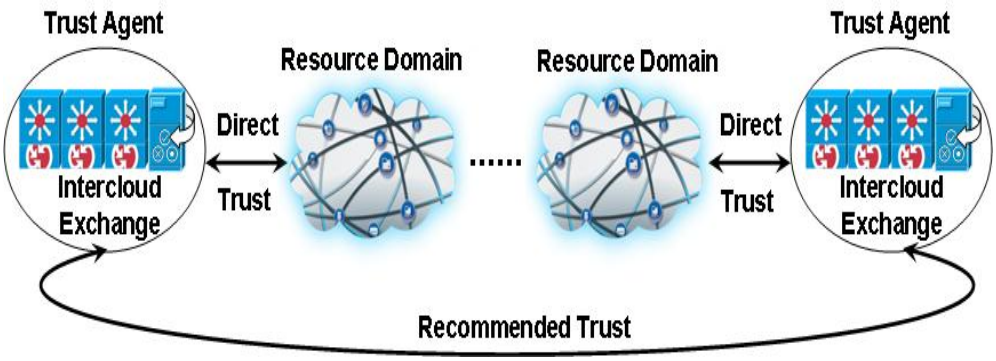

Figure 3: Intercloud Trust Management Model

Exchanges are the custodians/brokers of "Domain based Trust" systems environment for their affiliated cloud providers. Cloud providers rely on the Intercloud exchanges to manage trust. As Domain trust agents, Intercloud exchanges store other domains' trust information for inter-domain cooperation. Essentially, the trust information stored reflects trust value for a particular resource type (compute, storage etc.) for each domain. Exchanges also recommend other domains trust levels for the first time inter-domain interaction.

6.7 Intercloud Identity and Access Management – SAML, XACML

One of the key requirements to have success in effectively managing identities in the Intercloud environment is the presence and support for a robust standards based federated identity management capability using prevailing federation standards such as SAML, WS-Federation, and Liberty ID-FF.

- See Security Assertion Markup Language (SAML), http://saml.xml.org/saml-specifications
- See Web Services Federation (WS-Federation), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsfed
- See Liberty ID-FF, http://projectliberty.org/liberty

Comprehensive Identity Management systems typically provide services such as: User Provisioning and User Management, Authentication and Authorization, Role Engineering, and Identity Data Integration/Virtualization.

In a typical federated identity model, in order for a cloud provider to establish secure communication with another cloud provider, it asks the trust provider service for a trust token. The trust provider service sends two copies of secret keys, the encrypted proof token of the trust service along with the encrypted requested token.

On the other hand, if the recipient cloud is affiliated to another Intercloud Exchange, the XMPP server will send the message to the recipient's XMPP server hosted by the affiliated Intercloud Exchange. This is essentially termed as XMPP federation — the ability of two deployed XMPP servers to communicate over a dynamically-established link between the servers. In the Intercloud topology, a server accepts a connection from a peer only if the peer supports TLS and presents a digital certificate issued by a root certification authority (CA) that is trusted by the server — **Trusted Federation**.

In a typical federated identity model, in order for a cloud provider to establish secure communication with another cloud provider, it asks the trust provider service for a trust token. The trust provider service sends

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

two copies of secret keys, the encrypted proof token of the trust service along with the encrypted requested token.

For scenarios where collaboration between initiating cloud provider and recipient cloud provider is across Intercloud Root or Intercloud Exchange, Intercloud Root systems will serve as a Trust Authority and act as the identity providers to mediate trust relationship as part of the **Trusted Federation**. The detail flow for this scenario is illustrated in Figure 4:

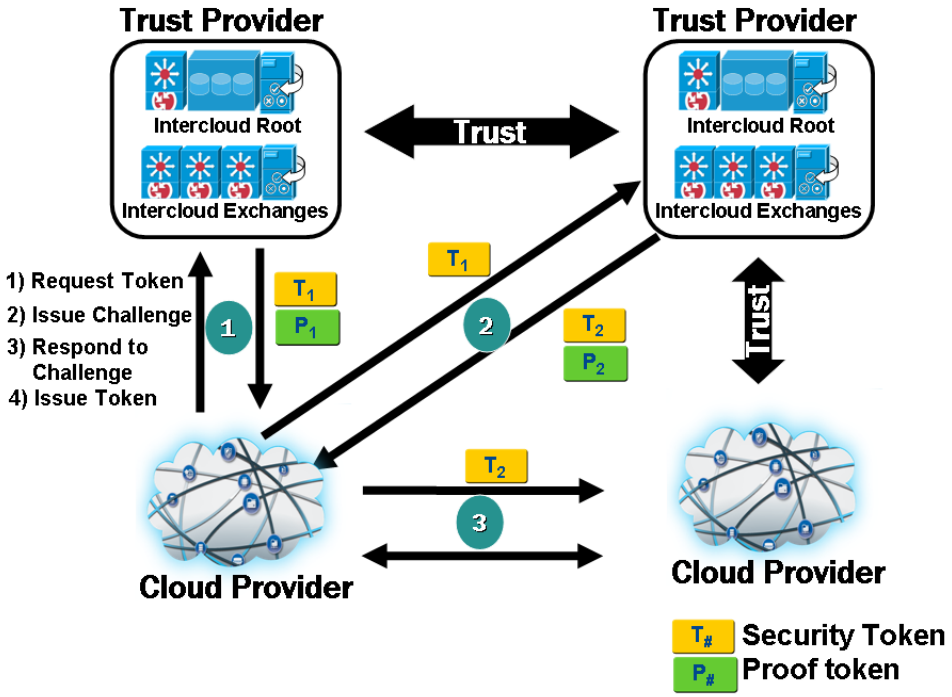

Figure 4: Inter "Intercloud Root" and Inter "Intercloud Exchange" Collaboration Scenario

For scenarios where collaboration between initiating cloud provider and recipient cloud provider is within the same Intercloud Exchange, Intercloud Exchanges will themselves will serve as a Trust Authority and act as the identity providers to mediate the trust relationship as part of the **Trusted Federation**. The detail flow for this scenario is illustrated in Figure 5.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

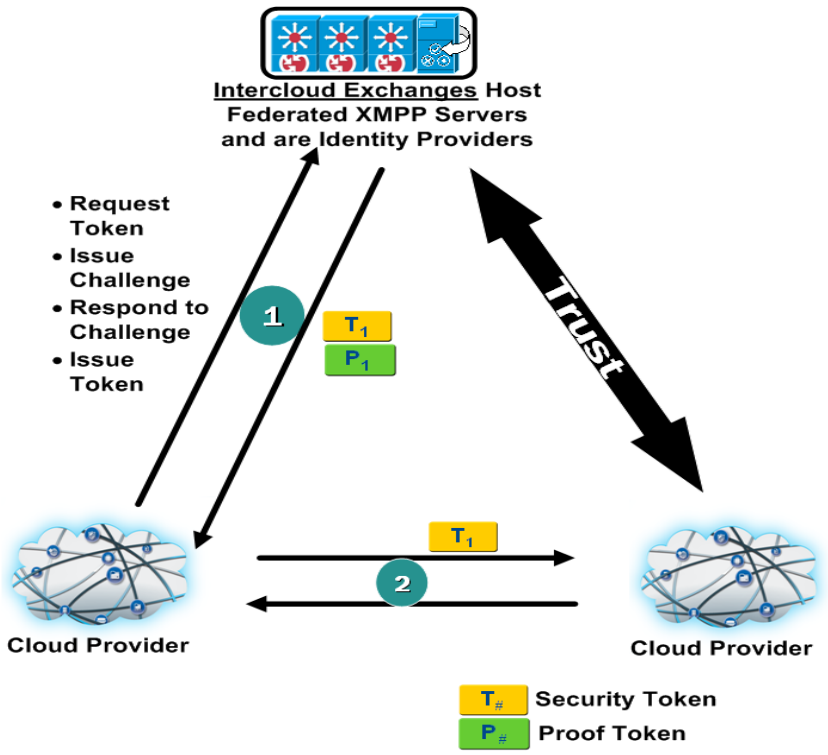

Figure 5. Intra "Intercloud Exchange" Collaboration Scenario

As regards to granular level authorization in the Intercloud environment, support of XACML-compliant entitlement management is highly desirable. XACML provides a standardized language and method of access control and policy enforcement.

- See OASIS xEtensible Access Control Markup Language (XACML), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

XACML (eXtensible Access Control Markup Language) is an XML-based language for access control that has been standardized in OASIS. XACML describes both an access control policy language and a request/response language. The policy language is used to express access control policies (who can do what when). The request/response language expresses queries about whether a particular access should be allowed (requests) and describes answers to those queries (responses).

6.8 Intercloud PKI Certificates Deployment

In an Intercloud cross-clouds federated environment, security concerns are even more important and complex. Intercloud paradigm or cloud computing paradigm, in general, will only be adopted by the users, if they are confident that their data and privacy are secured. Trust is one of the most fundamental means for improving security across heterogeneous independent cloud environments.

Currently, Public Key Infrastructure (PKI) based trust model is the most prevalent one. PKI trust model depends on a few leader nodes to secure the whole system. The leaders' validity certifications are signed by well established Certificate Authorities ("CA"s).

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

At a basic level, proposed Intercloud topology subscribes to the PKI based trust model. In accordance to the PKI trust model, the Intercloud Root systems will serve as the Root Certificate Authority (CA) and issue certificates to their affiliated Intercloud Exchange systems.

PKI Certificates not only need to identify the clouds, but the resources the clouds offer, and the workloads that the cloud wishes federation with other clouds, to work upon. Where web sites are somewhat static, and a certificate can be generated to trust the identity of that web site, cloud objects such as resources and workloads are dynamic, and the certificates will have to be generated by a CA. As per the proposed Intercloud topology, the Intercloud Exchange will serve as the intermediate "CA"s, issue temporary PKI certificates to their affiliated cloud providers acting in a just-in-time fashion to provide limited lifetime trust to the transaction at hand

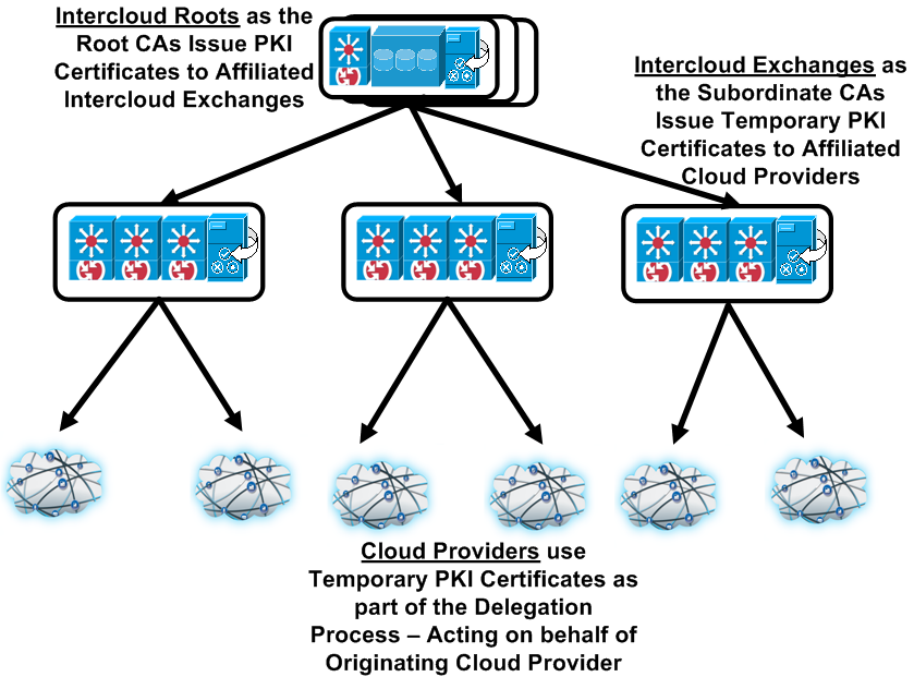

Figure 6. Intercloud PKI Certificates Topology

Cloud Providers, in turn, will use the temporary PKI certificates as part of the delegation process – acting on behalf of the originating cloud provider.

6.9 Intercloud Exchange Service Discovery – XMPP Based RDF and SPARQL approach

In order for the Intercloud capable Cloud instances to federate or otherwise interoperate resources, a Cloud Computing Resources Catalog system is necessary infrastructure. This catalog is the holistic and abstracted view of the computing resources across disparate cloud environments. Individual clouds will, in turn, will utilize this catalog in order to identify matching cloud resources by applying certain Preferences and Constraints to the resources in the computing resources catalog. The technologies to use for this are based on the Semantic Web which provides for a way to add "meaning and relatedness" to objects on the Web. To accomplish this, one defines a system for normalizing meaning across terminology, or Properties. This normalization is called an Ontology.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The way a Cloud would find the appropriate services is by leveraging a catalog of available resources published in a directory residing in the Intercloud Root. The Cloud's resource needs would be specified similarly, and a query would match the availability to the need.

The technologies to use for this are based in the Semantic Web which provides for a way to add "meaning and relatedness" to objects on the Web, by way of specifying Ontologies. For the Intercloud, this technique is used to specify resources such as storage, computing, and all the other possible services which Cloud both expose and consume. RDF is a way to specify such resources, and SPARQL is a query/matching system for RDF.

- See W3C Semantic Web Activity, at http://www.w3.org/2001/sw/
- See Resource Description Framework (RDF), at http://www.w3.org/RDF/
- See SPARQL Query Language for RDF, at http://www.w3.org/TR/rdf-sparql-query/

Once the Cloud has now secured a connection to the Intercloud root, it can look for a suitable other Cloud with which to interoperate. It will either interoperate through an Intercloud Exchange, or directly Cloud to Cloud, as the case may be.

As illustrated below, the following diagram shows the overall approach of how ontology based available cloud computing resources information will reside in a directory as part of the overall Intercloud topology:

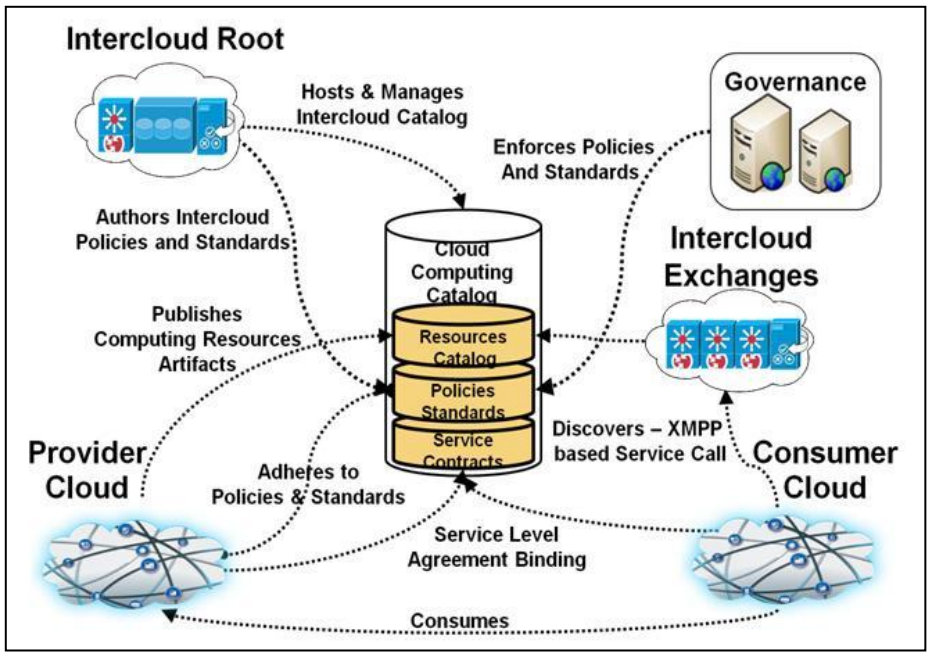

Figure 7. Intercloud Roots, Exchanges, and Catalog

An Ontology describes a domain completely. The essential mechanisms that ontology languages provide include their formal specification (which allows them to be queried) and their ability to define properties of classes. Through these properties, very accurate descriptions of services can be defined and services can be related to other services or resources.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

The Intercloud protocols use an RDF/OWL ontology framework. This catalog captures the computing resources across all clouds in terms of "Capabilities", "Structural Relationships" and Policies (Preferences and Constraints).

- See Web Ontology Language, at http://www.w3.org/TR/owl-features/

a similar ontology based semantic model that captures the features and capabilities available from a cloud provider's infrastructure. These capabilities are logically grouped together and exposed as standardized units of provisioning and configuration to be consumed by another cloud provider/s. These capabilities are then associated with policies and constraints for ensuring compliance and access to the computing resources.

The proposed ontology based model not only consists of physical attributes but quantitative & qualitative attributes such as "Service Level Agreements (SLAs)", "Disaster Recovery" policies, "Pricing" policies, "Security & Compliance" policies, and so on.

The following is a high level schematic of such ontology based semantic model.

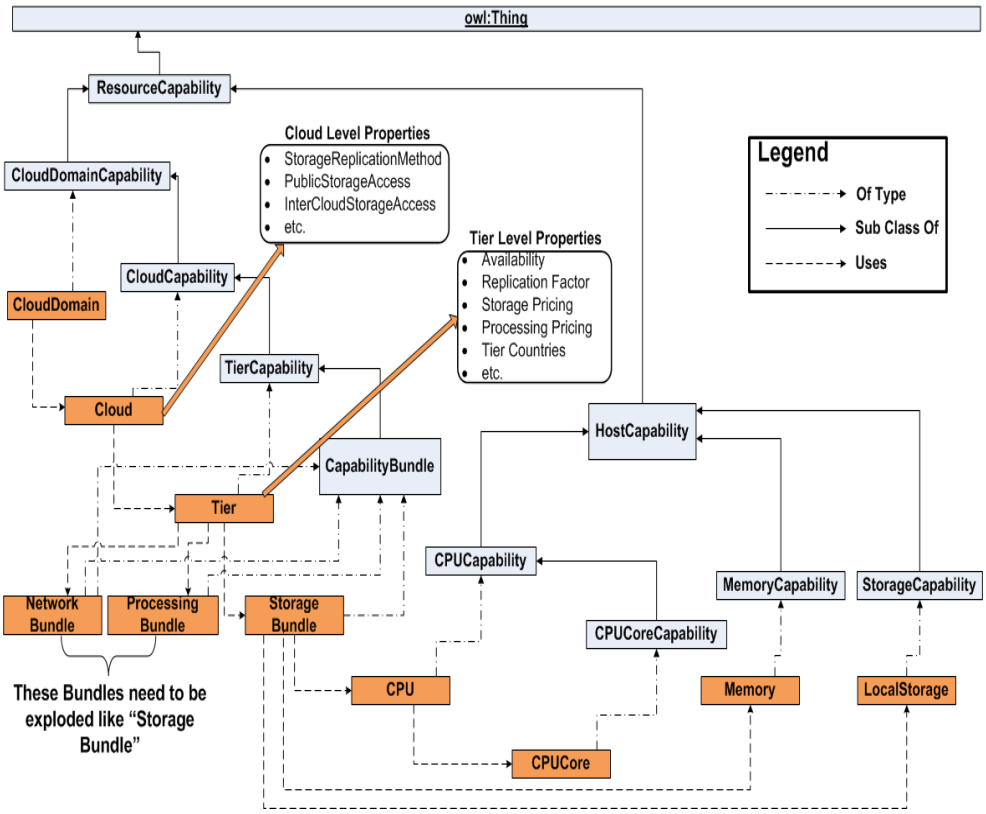

Figure 8. Cloud Computing Resources Ontology

At a very basic level, the RDF model is called a "triple" as it consists of three parts, Subject/Property/Object. It essentially contains one or more "descriptions" of resources. A "description" is a set of statements about a resource. It is structurally similar to entity/attribute/value. Essentially, a statement in RDF pulls resources, properties, and property values together. Statements are typically called triples because they include a subject (the resource), a predicate/verb (the property), and an object (the property value or another resource itself).

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

RDF allows you to define a group of things with common characteristics called "Classes". "Classes" are allowed to inherit characteristics and behaviors from a parent class. Each user-defined class is implicitly a subclass of super class called "`owl:Thing`".

The hierarchy of user-defined classes in our proposed ontology scheme are "ResourceCapability" → "CloudDomainCapability" → "CloudCapability" → "TierCapabil;ity" → "CapabilityBundle".

In order to demonstrate a working example, the following is a code snippet of N-Triples based ontology semantic model instead. N-Triples & Turtle are a human-friendlier alternative to RDF/XML. N-Triples or Turtle code, in turn, can be easily converted to RDF/XML format using a converter tool.

- See N-Triples, at http://www.w3.org/2001/sw/RDFCore/ntriples/
- See *SPARQL Query Language for RDF*, at http://www.w3.org/TR/rdf-sparql-query/

The following sample shows the flow for semantic model for cloud computing resources. Due to the large size of the proposed semantic model for cloud computing resources, we are unable to capture the sample RDF code snippet in this document. In order to demonstrate our working example N-Triples code snippet is shown instead.

Step 1: In our ontology example, "CloudDomain" is an instance of class "CloudDomainCapability". It consists of three resources "Cloud.1", "Cloud.2" & "Cloud.3":

```
<http://cloud/domain> <http://www.csp/resOntology#hasCapability> <http://cloud/domain/#cloud.1>.

<http://cloud/domain> <http://www.csp/resOntology#hasCapability> <http://cloud/domain/#cloud.2>.

<http://cloud/domain> <http://www.csp/resOntology#hasCapability> <http://cloud/domain/#cloud.3>.

<http://cloud/domain> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.csp/resOntology#ClouddomainCapability>.

<http://cloud/domain> <http://www.w3.org/2000/01/rdf-schema#label> "Cloud Computing
domain"^^<http://www.w3.org/2001/XMLSchema#string>.
```

Step 2: "Cloud.1", in turn, consists of tier instances "tier.1", "tier.2" & "tier.3":

```
<http://cloud/domain/#cloud.1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1#tier1>.

<http://cloud/domain/#cloud.1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1#tier2>.

<http://cloud/domain/#cloud.1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1#tier3>.
```

Step 3: Each of these cloud instances has associated properties such as "StorageReplicationMethod", "InterCloudStorageAccess" etc. etc. These properties are, in turn, used for determining if the computing resources of a cloud provider meet the preferences & constraints of the requesting cloud's interest and requirements:

```
<http://cloud/domain/#cloud.1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1#Storage-Replication-Method>.

<http://cloud/domain/#cloud.1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1#Inter-Cloud-Storage-Access>.

<http://cloud/domain/#cloud.1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1#Public-Storage-Access>.

<http://cloud/domain/#cloud.1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1#VPNGatewayAddress>.
```


Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

```
<http://cloud/domain/#cloud.1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.csp/resOntology#CloudCapability>.

<http://cloud/domain/#cloud.1> <http://www.w3.org/2000/01/rdf-schema#label> "Cloud
1"^^<http://www.w3.org/2001/XMLSchema#string>.
```

Step 4: Computing resources are logically grouped together as bundles and exposed as standardized units of provisioning and configuration to be consumed by another cloud provider/s. These bundles are "StorageBundle", "ProcessingBundle" & "NetworkBundle". Each "Tier", in turn, consists of instances of resource bundles such as "StorageBundle" etc. Each "Tier" also has its own associated properties depicting preferences and constraints:

```
<http://cloud/domain/cloud.1#tier1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/#storage1>.

<http://cloud/domain/cloud.1#tier1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/#processing1>.

<http://cloud/domain/cloud.1#tier1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/#network1>.

<http://cloud/domain/cloud.1#tier1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1.tier.1#replicationfactor>.
<http://cloud/domain/cloud.1#tier1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1.tier.1#availability>.

<http://cloud/domain/cloud.1#tier1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1.tier.1#storageprice>.

<http://cloud/domain/cloud.1#tier1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1.tier.1#processingprice>.

<http://cloud/domain/cloud.1#tier1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1.tier.1#countries>.

<http://cloud/domain/cloud.1#tier1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.csp/resOntology#TierCapability>.
```

Step 5: "StorageBundle", in turn, consists of resources such as "CPU", "CPU Cores", "Memory" & "LocalStorage":

```
<http://cloud/domain/cloud.1/bundle/#storage1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/storage1#CPU>.

<http://cloud/domain/cloud.1/bundle/#storage1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage0>.

<http://cloud/domain/cloud.1/bundle/#storage1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage1>.

<http://cloud/domain/cloud.1/bundle/#storage1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage2>.

<http://cloud/domain/cloud.1/bundle/#storage1> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/storage1#Memory>.

<http://cloud/domain/cloud.1/bundle/#storage1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.csp/resOntology#CapabilityBundle>.

<http://cloud/domain/cloud.1/bundle/#storage1> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.csp/resOntology#storageCapabilityBundle>.

<http://cloud/domain/cloud.1/bundle/#storage1> <http://www.w3.org/2000/01/rdf-schema#label> "EC2
Large"^^<http://www.w3.org/2001/XMLSchema#string>.

<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage1> <http://www.csp/resOntology#quantity>
"450971566080"^^<http://www.w3.org/2001/XMLSchema#long>.

<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage1> <http://www.csp/resOntology#unit>
<http://www.csp/resOntology#Byte>.

<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage1> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://www.csp/resOntology#StorageCapability>.
```


Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

```
<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage0> <http://www.csp/resOntology#quantity>
"450971566080"^^<http://www.w3.org/2001/XMLSchema#long>.

<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage0> <http://www.csp/resOntology#unit>
<http://www.csp/resOntology#Byte>.

<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage0> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://www.csp/resOntology#StorageCapability>.

<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage2> <http://www.csp/resOntology#quantity>
"10737418240"^^<http://www.w3.org/2001/XMLSchema#long>.

<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage2> <http://www.csp/resOntology#unit>
<http://www.csp/resOntology#Byte>.

<http://cloud/domain/cloud.1/bundle/storage1#LocalStorage2> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://www.csp/resOntology#StorageCapability>.

<http://cloud/domain/cloud.1/bundle/storage1#Memory> <http://www.csp/resOntology#quantity>
"8053063680"^^<http://www.w3.org/2001/XMLSchema#long>.

<http://cloud/domain/cloud.1/bundle/storage1#Memory> <http://www.csp/resOntology#unit>
<http://www.csp/resOntology#Byte>.

<http://cloud/domain/cloud.1/bundle/storage1#Memory> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.csp/resOntology#MemoryCapability>.

<http://cloud/domain/cloud.1/bundle/storage1#CPU> <http://www.csp/resOntology#hasCapability>
<http://cloud/domain/cloud.1/bundle/storage1#CPUCore>.

<http://cloud/domain/cloud.1/bundle/storage1#CPU> <http://www.csp/resOntology#hasCapability>
<http://www.csp/resOntology#X86-64Compatible>.

<http://cloud/domain/cloud.1/bundle/storage1#CPU> <http://www.csp/resOntology#quantity>
"2200000000"^^<http://www.w3.org/2001/XMLSchema#long>.

<http://cloud/domain/cloud.1/bundle/storage1#CPU> <http://www.csp/resOntology#unit>
<http://www.csp/resOntology#Hertz>.

<http://cloud/domain/cloud.1/bundle/storage1#CPU> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.csp/resOntology#CPUCapability>.

<http://cloud/domain/cloud.1/bundle/storage1#CPUCore> <http://www.csp/resOntology#quantity>
"2"^^<http://www.w3.org/2001/XMLSchema#int>.

<http://cloud/domain/cloud.1/bundle/storage1#CPUCore> <http://www.w3.org/1999/02/22-rdf-syntax-
ns#type> <http://www.csp/resOntology#CPUCoreCapability>.
```

SPARQL (SPARQL Protocol And RDF Query Language) is a very powerful SQL-like language for querying and making semantic information machine process-able. The structure and example of a SPARQL Query is illustrated in Figure 5.

Structure:

```
PREFIX: Prefix definition (optional)
SELECT: Result form
FROM: Data sources (optional)
WHERE: Graph pattern (=path expression)
    •   FILTER
    •   OPTIONAL
```

Example:

```
PREFIX geo: <http://www.geography.org/schema.rdf#>
SELECT ?X ?Y
FROM <http://www.geography.org>
WHERE {  ?X geo:hasCapital ?Y.
                    ?Y geo:areacode ?Z }
ORDER BY ?X
```

Figure 9. Structure & Example of SPARQL Query

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

SPARQL provides a very powerful language for executing very complex queries into the RDF data which are often necessary. In our case, the following example query applies certain Preferences and Constraints to the resources in the computing semantics catalog for determining if the service description on another cloud meets the constraints of the first cloud's interest:

```
PREFIX  xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT  ?cld1 ?cld2 ?cld3 ?cld4 ?cld5
WHERE   { ?cld1 <http://www.csp/resOntology#availabilityQuanity> ?availabilityQuanity .
  ?cld2 <http://www.csp/resOntology#replicationFactor> ?replicationFactor .
 ?cld3 <http://www.csp/resOntology#tierCountries> ?tierCountries .
  ?cld4 <http://www.csp/resOntology#StorageReplicationMethod> ?StorageReplicationMethod .
  ?cld5 <http://www.csp/resOntology# InterCloudStorageAccess > ?InterCloudStorageAccess .

FILTER ( ?availabilityQuanity = 99.999 )
FILTER ( ?replicationFactor = 5)
FILTER ( ?tierCountries = "Japan")
FILTER ( ?StorageReplicationMethod = "AMQP")
FILTER ( ?InterCloudStorageAccess = "NFS")

   }
```

6.10 Intercloud Resources Catalog Deployment

Intercloud Roots will host the globally dispersed computing resources catalog in a federated manner.

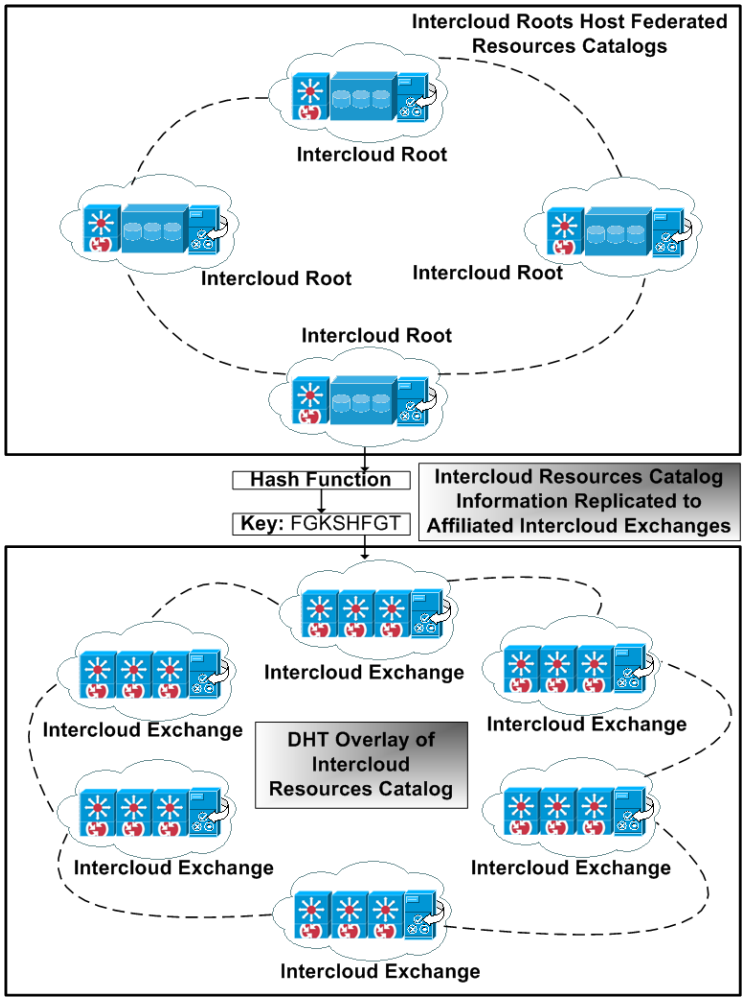

Figure 8. Intercloud Topology – Resources Directory Deployment

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Intercloud Exchanges, in turn, will leverage the globally dispersed resources catalog information hosted by federated Intercloud Roots in order to match cloud resources by applying certain Preferences and Constraints to the resources. From overall topology perspectives,

Intercloud Exchanges will provide processing nodes in a peer-to-peer manner on the lines of Distributed Hash Table (DHT) overlay based approach in order to facilitate optimized resources match-making queries. Ontology information would be replicated to the Intercloud Exchanges (DHT overlay nodes) from their affiliated Intercloud Roots using a “Hash” function.

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.

Annex A

(informative)

Bibliography

Copyright © 2010 IEEE. All rights reserved.
This is an unapproved IEEE Standards Draft, subject to change.