IEEE INTERCLOUD TESTBED

TECHNICAL OVERVIEW AND ENGINEERING PLAN

SPONSORED BY THE IEEE CLOUD COMPUTING INITIATIVE

MANAGED BY THE IEEE STANDARDS ASSOCIATION

INDUSTRY CONNECTIONS PROGRAM

Supervising Authority:
Kathy L. Grise
Future Directions Senior Program Director
IEEE Technical Activities
445 Hoes Lane, Piscataway, NJ 08854 USA
k.l.grise@ieee.org
Tel 732 981 2871
cloudcomputing.ieee.org

Prepared By:
David Bernstein
Managing Director
Cloud Strategy Partners, LLC, USA
david.bernstein@ieee.org
Tel 408 857 9872,
www.cloudstrategypartners.com

Contents

Background

The IEEE "Intercloud" exploratory research began in 2008 with a research group in Cisco who decoded to explore ideas around interoperability of clouds, in an analogous way that the interoperability of networks had been solved. As the term "Inter-Networking" and then "Internet" was coined to address the interoperability of networks, the term "Inter-Cloud Computing" or just "Intercloud" was coined to address the interoperability of Cloud Computing.

In May of 2009 at the International IEEE Workshop on Cloud Computing/International Symposium on Cluster Computing and the Grid in Shanghai China, a team paper was presented called "Cloud Computing Interoperability Protocols and Formats – Defining the Intercloud". Later that month at the 4th International IARIA/IEEE Conference on Internet and Web Application Services in Venice, Italy a more complete paper was presented: "Blueprint for the Intercloud – Protocols and Formats for Cloud Computing Interoperability". This paper won "best research" of the conference and is referenced by Wikipedia[1] as the first documented invention of the Intercloud concept. This compelling idea rapidly spread to interest several researchers worldwide in Cloud Computing.

The group of interested researchers expanded, and the now "Intercloud community" published several dozen academic papers further defining possible Intercloud protocols, exploring the architecture, and solving the security issues. There has been dozens of talks on this subject with research teams including NSF, University of Melbourne, Purdue University, labs from AT&T, Orange, NTT, and countless IEEE and other conferences and workshops including several workshops sponsored by NIST. A description of the proposed Intercloud architecture is included as a reference implementation for Cloud Broker in NIST Special Publication 500-293 U.S. Government Cloud Computing Technology Roadmap, Volume III.

In the fall of 2010 this group of researchers proposed to the IEEE through a PAR (Project Authorization Request) that a new standard working group be chartered to formalize these discussions. In early 2011 the IEEE assigned P2302 as the standards working group identifier. And in July of 2011 the first P2302 working group meeting was held. Good progress has been made and as of early 2013 the team is working on the 3rd Draft Standard.

With no reference implementation, or "test bed" as it was called to try out the proposed standard, the working group has decided it is running into a challenge. As everyone knows the Internet was constructed with the principles of "rough consensus and working code" and it was felt the Intercloud should acknowledge such philosophy; therefore, the IEEE Intercloud Testbed project was founded, to run in parallel with and be bi-directionally complementary to the IEEE P2302 working group. The IEEE Intercloud Testbed team would integrate participant clouds, start with the IEEE P2302 design, and develop a working implementation of the Intercloud architecture. Issues found would be fixed, the created code placed in an open source project, and the system further documented. As needed the IEEE Intercloud testbed engineering would deviate, improve, expand, or change the details as specified in IEEE P2302, feeding back those changes to the standards working group.

This document represents the initial engineering plan for the IEEE Intercloud Testbed.

[1] http://en.wikipedia.org/wiki/Intercloud

Project Motivation

Cloud computing is a new design pattern for large, distributed data centers. 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. The concept of a cloud operated by one service provider or enterprise interoperating with a cloud operated by another is powerful.

In other words, **there are no implicit and transparent interoperability mechanisms or standards in place in order for disparate cloud computing environments to be able to seamlessly federate and interoperate.** This is an issue now that Cloud Computing has become such an important technology to Government, Communications, Industry, and Entertainment.

Importance of the Project to Government, Academia, and Industry

Academia and Research are now at a cusp in the development of electronic infrastructure to support research. It seems clear that, with the exception of high-end capability computing (application of the largest and most powerful computers to the most challenging problems), variations on the cloud computing model will ultimately dominate computing of most kinds in the future. Enabling Academia and Research to coordinate and interoperate their compute platforms would place US capability at an advantage.

For Industry, Cloud Computing is proving to be a cost effective, elastic and scalable, and easy to use platform for web or mobile applications. If one follows some basic design blueprints, the "cloud side" of your web or mobile applications can handle the sudden explosion of subscribers that every application author hopes for. Cloud Computing platforms are proving, especially to the mobile developer, to offer more than just an easy to use, scalable platform. Application capabilities using advanced analytics, predictive modeling, and "big data" can't do without a cloud platform. Search, speech recognition, and location based services are all cloud-based now due to the data and processing requirements. And the newest communications platforms implementing rate adaptive codecs, multi-device transcoding, and multi-participant conferences and rooms are all using cloud techniques.

Furthermore, users have become accustomed to the "infinity" of the cloud - as much storage as they ever need, without the need to delete anything, and search working perfectly fine on a lifetime of data - even if it is multimedia. Wherever a user is, they expect their data to be fast in arriving to them, fast in streaming up to the cloud, and fast showing up in their blog, on their Facebook page, or in their mobile storage folder. They expect access to "their" cloud world, their data, their history, their preferences; and they expect the cloud to be smart and adapt to where they happen to be at the time. When communications devices don't look like phones much anymore, all Internet and Mobile apps will have a significant dependency on the cloud.

Of course, users will be subscribers of a carrier, and use directly, or over the top, many of these cloud capabilities. Carriers will capture more and more revenue per subscriber as they become more and more involved in providing the cloud infrastructure upon which all this runs. From this subscriber perspective, they will expect roaming of all these new services to "simply work." The bar has been set quite high by today's mobile industry, where enumeration, text, voice, and data roaming are all implemented - at a price - in today's global mobile network. It is easy to see, that "everything cloud" will also need to support roaming - in a "global mobile Intercloud".

Organization of the Testbed

The Testbed is an activity of the IEEE Cloud Computing Initiative (CCI), operating as an activity of the IEEE "Industry Connections" program.

The Testbed is governed by an Executive Committee, called the IEEE Intercloud Testbed Executive Committee (ITEC), which includes CCI representatives and selected activity participants.

The ITAC and the Testbed activity are overseen by the IEEE Cloud Computing Standards Committee (CCSC) and the IEEE-SA Board of Governors (BOG).

The ITEC provides the strategic direction for the activity, manages the growth of participation and directs the high level development of all deliverables.

Envisioned Participants

Industry and University labs are invited to participate as such, not only for the clouds themselves but the for the Intercloud Root and Exchange systems as well.

The map details organizations who have expressed interest in participating in the Testbed:

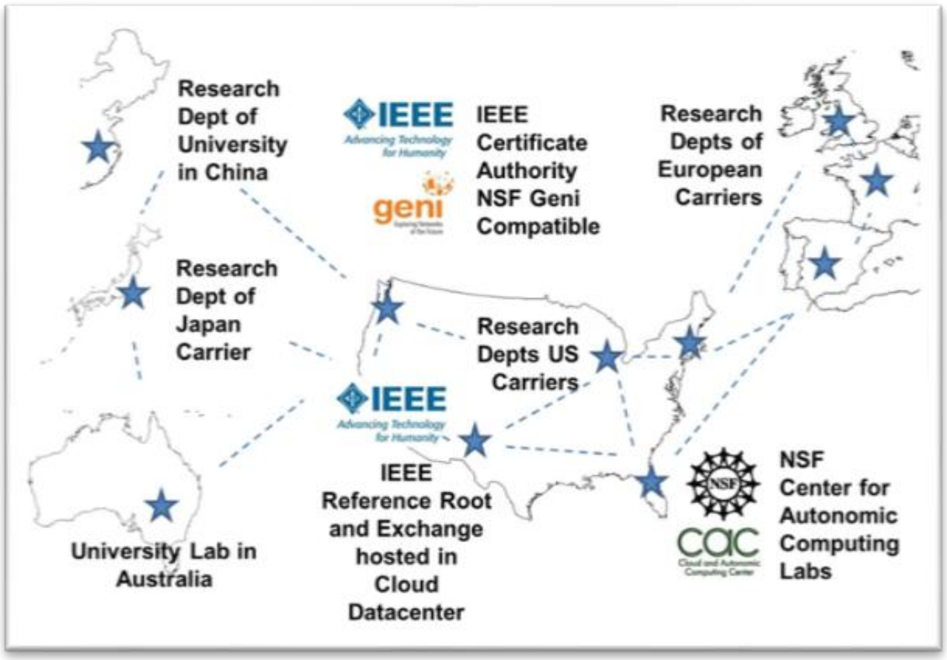
Research Dept of University in China
IEEE Certificate Authority NSF Geni Compatible
Research Depts of European Carriers
Research Dept of Japan Carrier
Research Depts US Carriers
IEEE Reference Root and Exchange hosted in Cloud Datacenter
University Lab in Australia
NSF Center for Autonomic Computing Labs

Figure 1. Map showing interested parties in the IEEE Intercloud Testbed and their Location

What Participants will Do

Participants will:

1. Volunteer to re-use existing cloud implementations, or construct a new cloud, of their choice in a well-connected data center in a geography;
2. Join the engineering project to code, test, re-engineer, and contribute to an open source implementation of the Intercloud protocol suite;
3. Adapt protocols to the various cloud platforms and resource types in use in the Testbed;
4. Connect to the reference Intercloud Root and Exchange IEEE which are running;
5. Explore the overall interoperability and applicability of the NSF GENI Project, in particular the root trust and governance mechanisms (certificate authority) of the GENI-ABAC project.
6. Experiment with cloud federation through, further develop protocols, ontologies, explore topology issues for scalability;
7. Feed results to the IEEE P2302 Standard project;
8. Publish Papers on their research and implementation experience to constituencies;
9. Create Reference Implementations of:
    a. An Intercloud root cloud including messaging, trust, and semantic directory
    b. An Intercloud exchange cloud
    c. An Operational multi-cloud Intercloud protocol suite
    d. Open Source projects of Reference Implementation

A Note on the Open Sourcing of Testbed Project Code

All source code will be made open as a GitHub repository under the Apache 2.0 license. For example a location such as https://github.com/intercloud will be set up.

Technical Description

There has been good initial work on this problem, collectively a set of mechanisms and standards which are a layered set of such protocols, called "Intercloud Protocols", to solve this interoperability challenges. The architecture proposed leads to an overall design of decentralized, scalable, self-organizing federated "Intercloud" topology.

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.

Topology

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.

The elements and topology for the Intercloud has been proposed: where a reference Intercloud network topology and elements has been developed.

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

- Several Intercloud Gateways: analogous to the Internet Router which connects an Intranet to the Internet.
- Several Intercloud Exchanges: analogous to Internet Exchanges and Peering Points – called Brokers in the NIST Reference Architecture where clouds can interoperate.
- Intercloud Roots: containing services such as Naming Authority, Trust Authority, Messaging, Semantic Directory Services, and other "root" capabilities. The Intercloud root is not a single entity – it's a globally replicating and hierarchical system.

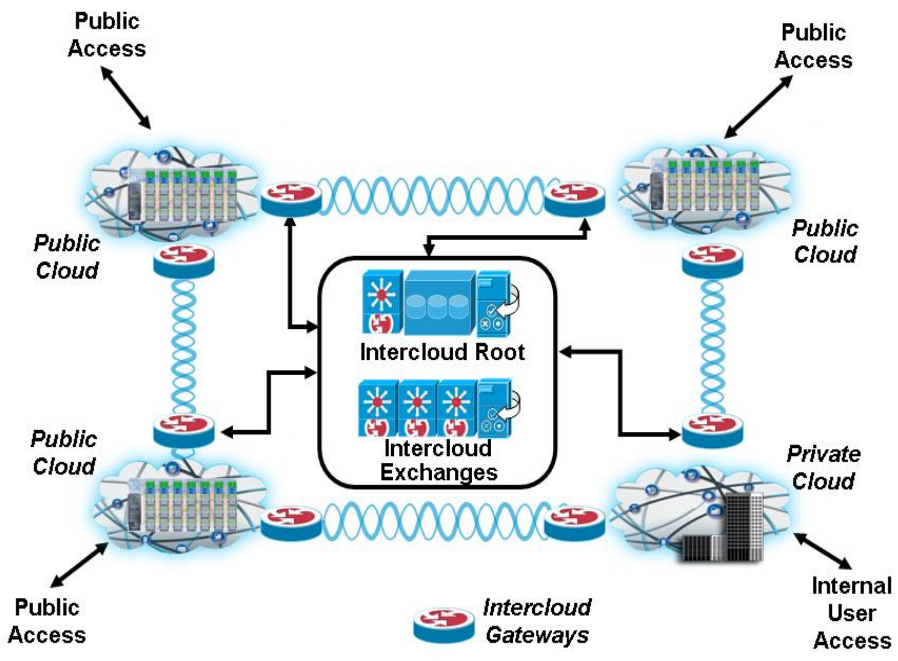

Figure 2. Reference Intercloud network topology and elements

Intercloud Gateway

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.

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.

Intercloud Roots

As described earlier that various providers will emerge in the enablement of the Intercloud. One can first envision 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. 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. A specific mechanism to look closely at is the NSF GENI project's ABAC model.

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.

Naming

How to name clouds is an open issue. Clouds are in the end IP addresses on the Internet and so the temptation to use DNS with a naming scheme (URN-based) appropriate to the communications substrate (see below) is tempting. DNS, especially with DNS SEC, contains mechanism to return trusted addresses as a result of name resolution.

XMPP, the envisioned communications substrate, indeed uses URNs for identifying a resource which one could "chat" with and establish identity. XMPP naming is very flexible, depending on the services at the target end to figure out what exactly is being asked for; as a result XMPP supports names in inbox form such as `foo@example.com`, but also `<foo>@example.com`, `::foo::@example.com`, `foo@example.com/service`, or `service@foo@example.com`.

On the one hand then if the communications substrate is going with XMPP it does makes sense to use the XMPP naming scheme, this implies the inclusion of a DNS controlled domain reference.

There are some new opportunities here to create a different naming scheme for a couple of reasons. First of all, we are specifying system architecture more like internet routing protocols than like computer endpoints. To that extent, while XMPP might be the right communications substrate, a naming system more like Autonomous Systems (AS) might be more appropriate. The rationale is, XMPP names, like IP addresses or domain names, represent specific endpoints. They are names for connectivity. Here, we are talking about system to system federation. This represents one cloud operator's willingness to federate with a specific other cloud operator. Each cloud operator may have several (using AWS terminology) availability zones or regions in their “cloud”. How they manage their own internal cloud, and which parts of it they decide to actually make available for Intercloud federation is up to them. It’s very analogous to the BGP policies they have in place for network transit.

Let’s call as a working term the cloud equivalent of autonomous system to be CS for Cloud System and continue the exploration of the considerations. The point of cloud federation, is for a workload is using a “home” cloud, to transparently obtain as much resource of the kind and in the quantity that it needs irrespective of whether that home cloud has to federate to get it or not.

The workload will not know anything about a CS, however the home cloud will know (at least the intercloud gateways which interface to an Intercloud exchange will know) what CS number they are. What they will not know is the equivalent of the AS Routing Tables that are in an internet exchange Route Server (or router); just as in the Internet Exchange this is the job of the exchange routers or the route server, in our Intercloud architecture this is the job of the Intercloud Exchange (somewhere).

We have designed that in the exchange there is some kind of “solver” which has on the “supply side” a map of which CS has what type of resource to offer and on the “consumer side” a map of which CS is asking for what type of resource. All the CS names are held in the exchange just as AS numbers are in the routers or the route servers. Just as there is a numbering authority for AS numbers in the Internet (IANA, and the local Regional Internet Registries (RIRs)), there will have to be some kind of numbering authority for the CS names.

For now the IEEE can be the numbering authority for the CS names.

Communications Substrate

In the initial designs, the end clouds will each have Intercloud Gateway code affixed to them. They will support the Conversational Protocol (XMPP) as well as the Transport Protocol (Web Sockets). The Intercloud Root is supporting the Conversational (XMPP) server system.

We will build-out the XMPP part of the portable gateway code to complete at least the XMPP-Core (RFC 6120) and XMPP-IM (RFC 6121) Profiles, as far as a Client goes. We will leverage a series of XMPP extensions (XEP series) defined by XMPP standards foundation. One of these extensions is XEP-0244. 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. A reference implementation for the IO Data XEP, XMPP Web Services for Java (xws4j), will then be completed.

Later stages will build out XMPP-ADDR (RFC 6122), and XMPP-E2E (RFC 3923). The roles and exact strategy for XMPP-JRN (RFC 4854) and/or XMPP-ENUM (RFC 4979) and/or XMPP-JRI (RFC 5122) need to await the output of the namespace design component. In other words we need to merge the CS Names proposal with XMPP (JID) Naming.

On top of this there needs to be a services framework. This is not as well thought out yet but is imagined to be WebSockets. WebSockets are described in RFC 6455. The protocol consists of an opening handshake followed by basic message framing, layered over TCP. The goal of this technology is to provide a mechanism for cloud to cloud two-way payload communication that does not rely on opening multiple HTTP connections.

Trust Infrastructure

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.

Audit Trail

The Root servers will support XMPP audit trails. These implementations will likely use XMPP S2S, but have not been designed yet. Raw audit traffic will need to be folded and reduced such that conversations relating to decisions of fulfilling federation requests can be reproduced and proven to have matched the request in the most optimal way. In this way arbitrage will be enabled and trusted.

Semantic Resource Directory

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. The EU FP7 MOSAIC project is an excellent implementation of exactly this element.

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 cannot 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.

Intercloud Exchanges

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.

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. Due to very large size of "Cloud Ontology" set in the intercloud environment, we are expecting a very large RDF dataset. SPARQL queries against such a large RDF dataset would be highly inefficient and slow.

We believe that such a large RDF dataset should be stored on a Distributed File System such as HDFS (Hadoop Distributed File System). By storing RDF dataset in HDFS and querying through Hadoop "Map-Reduce" programming would make SPARQL queries highly efficient and faster.

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.

Sequence Diagram

In summary the sequence diagram in the Intercloud federation would look something like this:

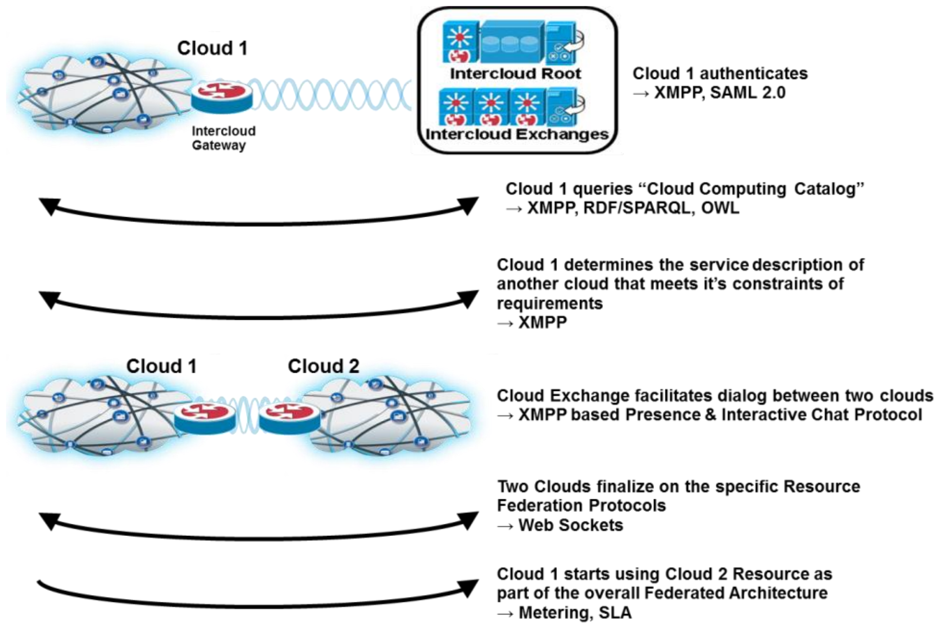
Cloud 1
Intercloud Gateway
Intercloud Root
Intercloud Exchanges
Cloud 1 authenticates
→ XMPP, SAML 2.0
Cloud 1 queries "Cloud Computing Catalog"
→ XMPP, RDF/SPARQL, OWL
Cloud 1 determines the service description of another cloud that meets it's constraints of requirements
→ XMPP
Cloud 1
Cloud 2
Cloud Exchange facilitates dialog between two clouds
→ XMPP based Presence & Interactive Chat Protocol
Two Clouds finalize on the specific Resource Federation Protocols
→ Web Sockets
Cloud 1 starts using Cloud 2 Resource as part of the overall Federated Architecture
→ Metering, SLA

Figure 3. Example Intercloud conversational sequence diagram

Protocols Summary

Elements of the Intercloud topology speak with each other over a variety of protocols, suited for the task at hand:

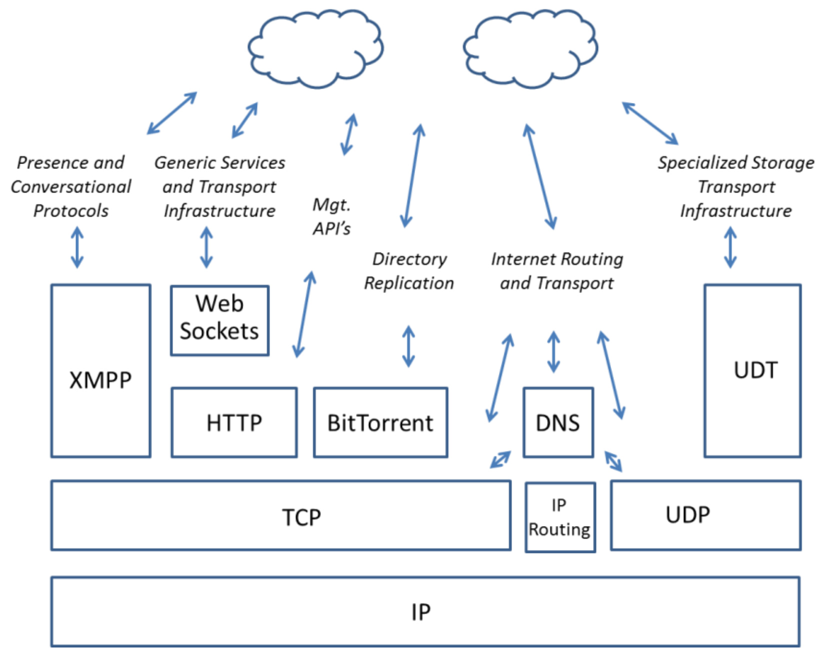
Presence and Conversational Protocols
Generic Services and Transport Infrastructure
Mgt. API's
Directory Replication
Internet Routing and Transport
Specialized Storage Transport Infrastructure
XMPP
Web Sockets
HTTP
BitTorrent
DNS
UDT
TCP
IP Routing
UDP
IP

Figure 3. Intercloud network protocols

Implementation Use Cases for Consuming Clouds

Once the clouds have negotiated federation, then they can actually begin to federate. The actual model used to federate is different depending on the resource being federated. Consider all the types of resources which clouds can seek to access from other clouds, through federation:

- Virtual Machines for computing
- Application containers (Java, Groovy, PHP)
- Storage (file level, replication level, etc), (Block, Object, etc), (ephemeral, persistent)
- Memory (Working, Cache, Disk Cache, etc)
- Transcoding
- Stream Processing
- XML processing
- Etc

A Consuming Cloud Making Federated Computing look like Its Own

In this case, we are federating some kind of comouting resource, consider for example Virtual Machines (VMs). A Requesting Cloud needs more VM's to satisfy user demand and so inquires to the Intercloud Exchange to which it is connected to try to federate VM's. The Exchange goes through all of the mechanism described above and delivers to the Requesting Cloud appropriate instructions as to how to access these resources (this has yet to be finally designed, but the example is still illustrative).

For computing resources, The Requesting Cloud wants resources which are transparent and equivalent to it's own computing resource. The user request which caused the federation did not ask to have a "foreign" VM, he wants a VM which is equivalent to a "native" VM. More precisely, the Requesting Cloud will make a federation request specifying a computing resource which meets the SLA which has been promised to the user.

- The Requesting Cloud **does not care** where the resource runs, but still cares that it meets the SLA which has been promised to the user.
- The Requesting Cloud **does care** where the resource runs, and also still cares that it meets the SLA which has been promised to the user.

On inspection, these two use cases have the same end-solution – that is **in both cases the resource is physically not running on the requestor cloud** (hence the need for Federation) **but it must "appear" to be** (resources which are transparent and equivalent to its own computing resource). In the first case a preferred location is simply not specified in the "Resource Properties", in the second case, it is. The issues of specifying the SLA and properties (like location) required, and "solving" for an available fulfilling resource, are the job of the semantic resource description framework and the exchange. Let's assume this has been handled. Next, the resources must be made available to the Requesting Cloud, which means they have to be provisioned in the Fulfilling Cloud in such a way as they appear to be "local" in the Requesting Cloud. One will recognize this as the "Virtual Private Cloud" scenario, except that the requirement of dynamic, on-demand provisioning exists (today's VPC solutions rely on static configuration of IPSEC VPN tunnels in Customer Premise edge routers, as well as in Public Cloud networking infrastructure). Perhaps SDN or MPLS VPN would be useful for this scenario. What results are VMs which appear local to the Requesting Cloud, as illustrated below:

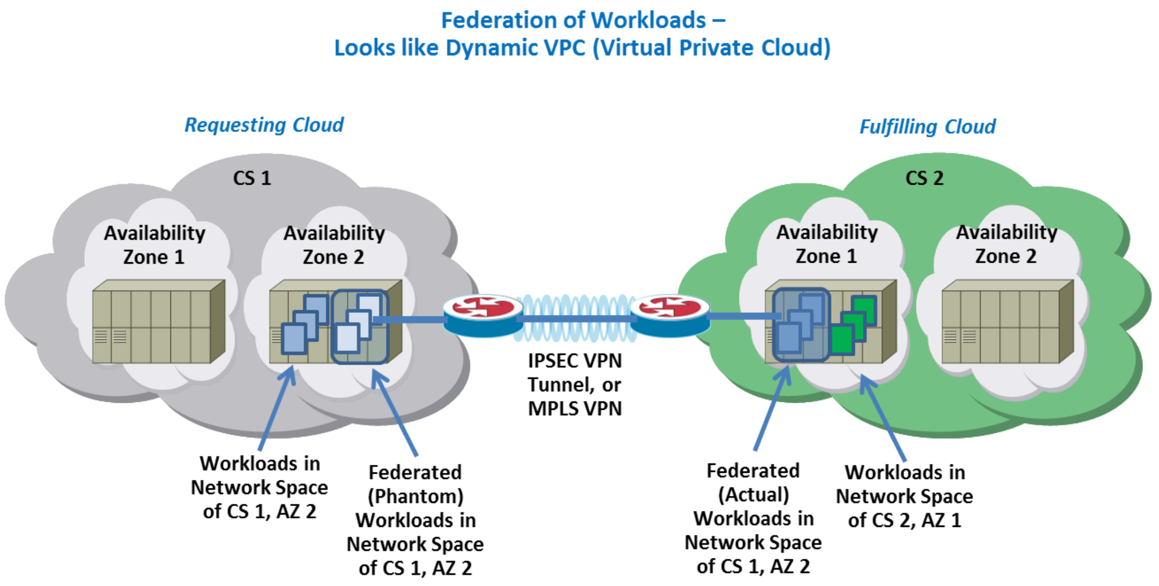
Federation of Workloads –
Looks like Dynamic VPC (Virtual Private Cloud)
Requesting Cloud
Fulfilling Cloud
CS 1
CS 2
Availability Zone 1
Availability Zone 2
Availability Zone 1
Availability Zone 2
IPSEC VPN Tunnel, or MPLS VPN
Workloads in Network Space of CS 1, AZ 2
Federated (Phantom) Workloads in Network Space of CS 1, AZ 2
Federated (Actual) Workloads in Network Space of CS 1, AZ 2
Workloads in Network Space of CS 2, AZ 1

Figure 4. Intercloud federation of workloads example

There are many interesting uses for the capabilities described above, and they follow the two cases described above:

- In the case where the Requesting Cloud **does not care where** the resource runs, but still cares that it meets the SLA which has been promised to the user, this is a classic "cloud overflow" [some people call this cloud bursting but that is not really accurate] capability. **It essentially makes the Requesting Cloud of infinite compute capability.**

- In the case where the Requesting **Cloud does care where** the resource runs, and also still cares that it meets the SLA which has been promised to the user, this is like a global MPLS VPN situation, where network can be provided by a carrier in a geography where they don't operate network, but can wholesale it and interwork with it via MPLS Inter-Carrier Interconnect (MPLS-ICI) standardization. **It essentially makes to Requesting Cloud of infinite coverage.**

This is quite substantial, to have these outcomes at once – even a small cloud, for computing, can be limitlessly large and everywhere through federation! Of course, the little browser, can access any public web server, no matter where it sits, and this is exactly how the internet works.

A Consuming Cloud Making Federated Storage look like Its Own

Storage is a somewhat different model. The address space where the server lives is less important for example. And here, we are fitting in to the model where the storage already replicates within an availability zone, and therefore the natural expansion level must tie into replication. Given that, here is an illustration for how that would occur:

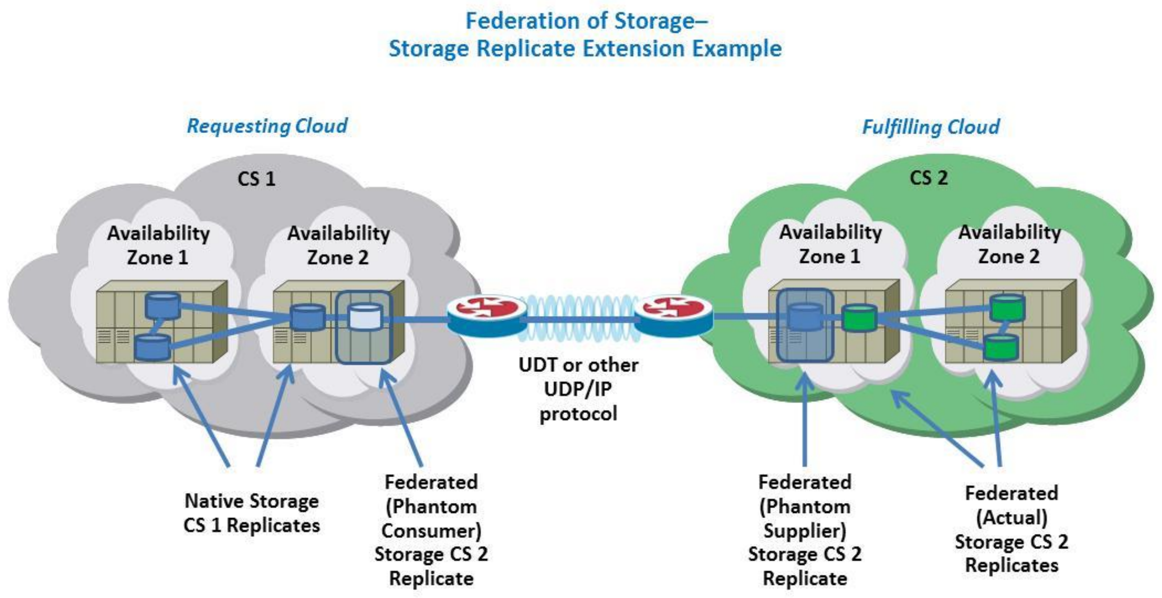
Federation of Storage–
Storage Replicate Extension Example
Requesting Cloud
Fulfilling Cloud
CS 1
CS 2
Availability Zone 1
Availability Zone 2
Availability Zone 1
Availability Zone 2
UDT or other UDP/IP protocol
Native Storage CS 1 Replicates
Federated (Phantom Consumer) Storage CS 2 Replicate
Federated (Phantom Supplier) Storage CS 2 Replicate
Federated (Actual) Storage CS 2 Replicates

Figure 5. Intercloud federation of storage example

You can see both of the above cases at work. One the one hand, the Phantom Consumer can behave in a mode such that it believes all of the storage is on CS1 but in fact, it is transparently on CS2.

It can also behave in a mode such that it believes all of the storage is on C2, which is transparently where it is. These concepts of "placement" may have to be added to storage API's to easily get this visibility.

We show a non-IP protocol for high capacity, high latency interconnect between the storage nodes. This illustrates that agreements between edge elements.

So again we end up with the dualistic advantage from storage federation:

- In the case where the Requesting Cloud **does not care where** the storage is, but still cares that it meets the SLA which has been promised to the user, this is a classic "storage overflow" capability. **It essentially makes the Requesting Cloud of infinite storage capacity**.
- In the case where the Requesting **Cloud does care where** storage is, still cares that it meets the SLA which has been promised to the user, and explicitly wants the storage to be in specific geographies, then, this federation accomplishes that. **It essentially makes to Requesting Cloud of infinite storage *and* geographical coverage.**

Again, this is quite substantial, to have these outcomes at once – even a small cloud, for storage, can be limitlessly large and everywhere through federation!

Engineering Project Workpackages

This project has an objective to develop a working implementation of this technology including the solving of the remaining issues, the creation of the code in an open source project, and the documenting of the system as a standard and via papers in the community. Here are the work packages in the project:

Workpackage: Completion of Master Technical Design Work

This workpackage will complete the overall design of the system, essentially expanding the work of the published research papers to end on specific initial decisions of formats, protocols, state diagrams, and so on.

Workpackage: Small Scale Experimental Implementation/Redesign Cycle

This workpackage will begin to set up an experimental testbed, large enough to try the modules and see if the overall system functions.

Two "end clouds" will be set up. They will be of different implementation. For example, one will be an Openstack cloud, and one will be an Open Nebula cloud.

The end clouds will each have Intercloud Gateway code affixed to them. They will support the Conversational Protocol (XMPP) as well as the Transport Protocol (Web Sockets).

There will be a small cloud acting as an Intercloud Exchange. This system will initially be configured as a MapReduce (Hadoop) SPARQL solver (in it's most salient description). It will also have some auditing enabled.

There will be a small cloud acting as an Intercloud Root. The three main functions of the Intercloud Root: that is supporting the Conversational (XMPP) server system, the Trust/security (CA, etc) capability, and the Semantic Resource Directory.

The entire system will be worked and iterated with the design, until a basic coherency of operation is obtained.

Workpackage: Portable Gateway (Conversational Part) Development

This workpackage will build-out the XMPP part of the portable gateway code to complete at least the XMPP-Core (RFC 6120) and XMPP-IM (RFC 6121) Profiles, as far as a Client goes.

We will leverage a series of XMPP extensions (XEP series) defined by XMPP standards foundation. One of these extensions is XEP-0244. 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. A reference implementation for the IO Data XEP, XMPP Web Services for Java (xws4j), will then be completed.

Later stages of this workpackage will build out XMPP-ADDR (RFC 6122), and XMPP-E2E (RFC 3923).

The roles and exact strategy for XMPP-JRN (RFC 4854) and/or XMPP-ENUM (RFC 4979) and/or XMPP-JRI (RFC 5122) – eg merge the CS Names proposal with XMPP (JID) Naming.

Workpackage: Portable Gateway (Transport Part) Development

This workpackage will build out the transport protocol which is WebSockets. WebSocketds are decribed in RFC 6455. The protocol consists of an opening handshake followed by basic message framing, layered over TCP. The goal of this technology is to provide a mechanism for cloud to cloud two-way payload communication that does not rely on opening multiple HTTP connections.

Workpackage: Portable Gateway (Trust/Security Part) Development

This workpackage will build out the XMPP method for securing the XML stream from tampering and eavesdropping.

This channel encryption method makes use of the Transport Layer Security (TLS) protocol, Clouds use TLS to secure the streams prior to attempting the completion of SASL based authentication negotiation. SASL has a method for authenticating a stream by means of an XMPP-specific profile of the protocol. SASL provides a generalized method for adding authentication support to connection-based protocols.

Currently, the following authentications methods are supported by XMPP-specific profile of SASL protocol: "DIGEST-MD5", "CRAM-MD5", "PLAIN", and "ANONYMOUS". SAML provides authentication in a federated environment. Currently, there is no support for SAML in XMPP-specific profile of SASL protocol. However, there is a draft proposal published that specifies a SASL mechanism for SAML 2.0 that allows the integration of existing SAML Identity Providers with applications using SASL.

We will follow this down and implement appropriately.

Workpackage: Open Source Contribution

The Gateway code will be packaged such that it can be submitted into an open source project. As mentioned, initially there will be two ports or versions of the gateway distribution. This is to ensure a development in a portable mode.

Workpackage: Reference Root (Conversational Part) Development

The workpackage for the reference root will first build out the Transport component in a manner compatible with the client side of XMPP as explained above.

Workpackage: Reference Root (Transport Part) Development

The workpackage for the reference root will first build out the Transport component in a manner compatible with the client side of WebSockets as explained above.

Workpackage: Reference Root (Trust/Security Part) Development

This part of the project will build out the root "trust" capability of the protocols. Currently, Public Key Infrastructure (PKI) based trust model is the target. 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).

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 currently proposed trust architecture, a Certificate issued by a Certificate Authority (CA), must be utilized in the process to establish a trust chain.

This will be implemented in the Root.

Workpackage: Reference Root (Semantic Directory Part) Development

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. 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. We are proposing a new and improved service directory on the lines of UDDI but based on RDF/OWL ontology framework instead of current tModel based taxonomy framework. This catalog captures the computing resources across all clouds in terms of "Capabilities", "Structural Relationships" and Policies (Preferences and Constraints).

This semantic directory will be implemented.

Workpackage: Reference Root (Replication Part) Development

This workpackage addresses 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.

The basic idea of DHT overlay system is to map a key space to a set of peers such that each peer is responsible for a given region of this space and storing data whose hash keys pertain to the peer’s region. The advantage of such systems is their deterministic behavior and the fair balancing of load among the peers (assuming an appropriate hash function).

Furthermore, DHT overlay system provides location transparency: queries can be issued at any peer without knowing the actual placement of the data. Essentially, the DHT peer-to-peer overlay is a self-organizing, distributed access structure, which associates logical peers representing the machines in the network with keys from a key space representing the underlying data structure.

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.

These will be implemented.

Workpackage: Reference Exchange (Conversational Part) Development

This workpackage refers to the same XMPP technologies which are mentioned above, except, 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. This code will be implemented.

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 XMPP servers containing all presence information for Intercloud Root instances, Intercloud Exchange Instances, and Internet visible Intercloud capable Cloud instances. Intercloud Exchanges will host second-tier XMPP servers. Individual Intercloud capable Clouds will communicate with each other, as XMPP clients, via XMPP server environment hosted by Intercloud Roots and Intercloud Exchanges.

Workpackage: Reference Exchange (Transport Part) Development

For each of the matched resources the underlying protocol should be negotiated to be the most natural protocol for that resource. Initially, all payloads will traverse over WebSockets. This will be implemented according to the RFC.

Workpackage: Reference Exchange (Trust/Security Part) Development

Instead of each cloud provider establishing connectivity with another cloud provider in a Point-to-Point manner resulting into n2 complexity problem, as part of the Intercloud topology we propose that Intercloud Exchanges will help facilitate as mediators for enabling connectivity and collaboration among disparate cloud providers. As stated earlier that Intercloud Exchanges will leverage XMPP as control plane operations protocol for such collaboration and host the XMPP servers in a Trusted Federated manner to facilitate the end-to-end collaboration.

In order to establish collaboration with another cloud, an Intercloud enabled cloud will simply send a XMPP message to its affiliated Intercloud Exchange which hosts the XMPP server. If the recipient cloud is affiliated to the same Intercloud Exchange, the XMPP server will send the message directly to the recipient cloud.

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

This will be implemented.

Workpackage: Reference Exchange (Solver/Arbitrage Part) Development

In order to ensure that the requirements of an intercloud enabled cloud provider are correctly matched to the infrastructure capabilities in an automated fashion, there is a need for declarative semantic model that can capture both the requirements and constraints of computing resources.

We are proposing 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. Due to very large size of "Cloud Ontology" set in the intercloud environment, we are expecting a very large RDF dataset. SPARQL queries against such a large RDF dataset would be highly inefficient and slow. We believe that such a large RDF dataset should be stored on a Distributed File System such as HDFS (Hadoop Distributed File System). By storing RDF dataset in HDFS and querying through Hadoop "Map-Reduce" programming would make SPARQL queries highly efficient and faster.

We propose that the Intercloud Exchanges will leverage Hadoop based distributed processing for serving SPARQL request across federated resource catalogs hosted by Intercloud Root providers

Workpackage: Reference Exchange (Replication Part) Development

This part will duplicate the replication work done for the root, but for the exchanges.

Workpackage: Reference Exchange (Audit Part) Development

This workpackage will implement an audit subsystem for the resolution of a resource exchange. In other words, when requests for a particular resource is described, the exchange (through the solver algorithms) will select a particular federation target for the request. The decision tree which was walked to deliver the result and the alternatives available at the time will be kept in an audit trail which will be reduced by XML Schema to a machine readable form.

Workpackage: SSRP Implementation Attempt

This workpackage will follow the technical details set out in the SSRP research paper and attempt to make an implementation of that protocol.

Workpackage: IEEE 2302 Standard Contribution

The IEEE P2302 Standard for Intercloud Interoperability and Federation (SIIF) – will define topology, functions, and governance for cloud-to-cloud interoperability and federation.

This working group is actively creating a standard which this project would be the main lockstep contributor for.

Next Steps

Gantt Chart with dependencies and resources.

Appendix 1: Basis of Technology Architecture/Published Research

This overall design has been detailed in papers published by staff which have been reviewed and refereed for publication to the highest of standards:

1. D. Bernstein, S. Diamond, K, Shankar "Cloud Computing Interoperability Protocols and Formats – Defining the Intercloud",*1st International Workshop on Cloud Computing (Cloud 2009), in conjunction with the 9th International Symposium on Cluster Computing and the Grid - IEEE CCgrid, , Shanghai China;* May 18-21, 2009
2. D. Bernstein, S. Diamond, K, Shankar, E. Ludvigson, M. Morrow, "Blueprint for the Intercloud – Protocols and Formats for Cloud Computing Interoperability", *4th International Conference on Internet and Web Application Services – IARIA/IEEE ICIW09***,** *Venice, Italy;* May 2009
3. Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros, "InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services", *Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing* (ICA3PP 2010, LNCS 6081, Springer, Germany), 13-31pp, Busan, South Korea, May 21-23, 2010.
4. D. Bernstein, D Vij, "Using Semantic Web Ontology in Intercloud Directories and Exchanges", *ICOMP'10 - The 2010 International Conference on Internet Computing, Las Vegas, NV, USA,* July 2010
5. A. Celesti, F. Tusa, M. Villari, A. Puliafito, "Security And Cloud Computing: Intercloud Identity Management Infrastructure", *Proceedings of The 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2010) - ETNGRID, Tei of Larissa, Greece* June 2010
6. D. Bernstein, D. Vij, "Intercloud Directory and Exchange Protocol Detail using XMPP and RDF", *1st Workshop on Cloud-based Services and Applications (CBSA/IEEE 2010), Miami Florida, USA,* July 2010
7. D. Bernstein, D. Vij, "Simple Storage Replication Protocol (SSRP) for Intercloud", *The Second International Conference on Emerging Network Intelligence – IEEE/IARIA EMERGING 2010, Florence, Italy,* October 2010
8. D. Bernstein, D. Vij, "Intercloud Security Considerations", *The Second International Conference on Cloud Computing Technology and Science – IEEE CloudCom 2010, Indianapolis, Indiana,* December 2010
9. Rocco Aversa, Beniamino Di Martino, Francesco Moscato, Dana Petcu, Massimiliano Rak and Salvatore Venticinque, "An Ontology for the Cloud in mOSAIC", *Cloud Computing 2010: Methodology, System, and Applications,* CRC Press, ISBN 978-1-4398-5641-3
10. D. Bernstein, D. Vij, S. Diamond, "An Intercloud Cloud Computing Economy - Technology, Governance, and Market Blueprints", *IEEE Service Research and Innovation Global Conference – SRII 2011, San Jose, California,* March 2011
11. Dana Petcu, Ciprian Craciun, Massimiliano Rak. "Towards a cross-platform cloud API. Components for Cloud Federation", *Procs. 1st Intern. Conference on Cloud Computing & Services Science, CLOSER 2011 The Netherlands,* May 2011
12. D. Bernstein, D. Vij, "Intercloud Exchanges and Roots Topology and Trust Blueprint", *ICOMP'11 - The 2011 International Conference on Internet Computing, as part of*

WORLDCOMP'11 - The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing, Las Vegas, NV, July 2011

13. Demchenko, Y., C.Ngo, M.Makkes, R.Strijkers, C. de Laat, "Defining Inter-Cloud Architecture for Interoperability and Integration". *The Third International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING 2012*), July 22-27, 2012, Nice, France
14. Ngo, C., Y.Demchenko, C. de Laat, "Toward a Dynamic Trust Establishment Approach for Multi-provider Intercloud Environment", *The 4th IEEE Conf. on Cloud Computing Technologies and Science (CloudCom2012),* 3 - 6 December 2012, Taipei, Taiwan.
15. Demchenko, Y., Canh Ngo, Cees de Laat, Joan Antoni Garcia-Espin, Sergi Figuerola, Juan Rodriguez, Luis Contreras, Giada Landi, Nicola Ciulli, "Intercloud Architecture Framework for Heterogeneous Cloud based Infrastructure Services Provisioning On-Demand". *Workshop The Second International Workshop on inter-Clouds and Collective Intelligence (iCCI-2013). The 27th IEEE International Conference on Advanced Information Networking and Applications (AINA2013)*. 25-28 March 2013