IEEE Intercloud Testbed Engineering Bootstrap

Project Notes
Last Edit 7/30/2014

Contents

Introduction to Intercloud Testbed Bootstrap Strategy

This document follows the originally published top level categories, but is updated to reflect the started project (August 1, 2014)

1) Completion of Master Technical Design Work
2) Collaboration, Source Code, Specs, Internal and Public Site(s)
3) Overall (Naming Part) Development and Policy/Procedures and Governance
4) Overall (Conversational Part) Development
5) Overall (Transport Part) Development
6) Overall (Trust/Identity Part) Development
7) Overall (Protocol/API Part) Development
8) Overall Root (Semantic Directory Part) Development
9) Overall Root (Audit Part) Development
10) Overall Root and Exchange (Deployment/Replication Part) Development
11) Reference Root (Integration of above services) Integration/Development
12) Overall Exchange (Solver/Arbitrage Part) Development
13) Reference Exchange (Integration of above services) Integration/Development
14) Reference Root(s) Infrastructure, Physical and Networks
15) Reference Exchange(s) Infrastructure, Physical and Networks
16) Portable Gateway (per Cloud OS flavor) Development
17) Use Case of IaaS Federation and Base Ontology, Implementation
18) Use Case of PaaS Federation and/or Specific Engine (ex, transcoding), Implementation
19) SSRP Implementation Attempt
20) MPLS VPC based Federation Attempt
21) Integration to LTE IPX network with LTE Signaling Exchange integration Attempt

NOTE: While the above list is a progression through the project it is not accurate as to precise order of Development work tasks. See the per work-item descriptions for more accurate sub-task ordering.

1. Continue to Recruit and Identify Resources to lead Work-areas.

- Identify leaders for “main work areas” for this whole project, at least through “IaaS Federation/Base Ontology” which is where we can really declare an end to end Intercloud first working.

2. Identify Local Labs.

- Create directory of labs where we will have software running. Existing labs may have a VMware, OpenStack, Citrix, OpenNebula or other flavor of Cloud running. We want to have a portable Gateway in as many implementations as possible.

- Could that cloud be modified/reconfigured and software on it changed for this project, or is the use of this resource as a user level "well behaved".
- What networking and storage resources are available in Member labs.
- National Research and. Education Networks (NRENs), hopefully one of our Labs can provide a bridge if they have in-lab access to an NREN. NRENs around the world, like Internet2, provide critical high-bandwidth connectivity to Universities and Research Labs.

3. Do initial work per Work-area.

Naming Part Development

- Current CPN (Cloud Provider Naming) proposal Finalization
- CPN proposal has some open issues including XMPP name binding and decisions on name space allotments, etc.
- CPN proposal finished and written up as paper/suitable for P2302
- Put up of find a DNS server we can use which allows use the record types specified in the CPN spec.
- Create test client to write/retrieve CPNs
- Write working/simple policies guide for name allocation [Best Practices/SDK]
- Make server accessible to entire team and leave running

Trust/Identity Part Development

- Put up OpenCA or DogTag as a CA Server
- Configure CA: Real trust anchor, Authentication profiles, Certificate issuance, revocation, and retrieval, Certificate Revocation List (CRL) generation and publishing, Certificate profiles, Simple Certificate Enrollment Protocol (SCEP), Local Registration Authority (LRA) for organizational authentication and policies, Encryption key archival and recovery
- Establish a working process for generating Certificates (X.509 PKI).
- Decide on which XMPP/SASL to use: Currently, the following authentications methods are supported by XMPP-specific profile of SASL protocol: "DIGEST-MD5", "CRAM-MD5", "PLAIN", and "ANONYMOUS".
- Sort out in XMPP-specific profile of SASL protocol. Chase down, 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.
- Add X.509 based SAML/SASL framework
- Create SAML Framework tests.
- Best Practices/SDK

- Make server accessible to entire team and leave running

Conversational Part Development

- Put up Jabber or other open XMPP Server
- Put up corresponding Client XMPP implementations
- Establish core XMPP profile. Suggest using tat least the XMPP-Core (RFC 6120) and XMPP-IM (RFC 6121) Profiles. Needs review.
- Evaluate plans for XMPP-ADDR (RFC 6122), and XMPP-E2E (RFC 3923), are they needed in our core profile. Needs review.
- Evaluate plans for XMPP-JRN (RFC 4854) and/or XMPP-ENUM (RFC 4979) and/or XMPP-JRI (RFC 5122) in the context of the output of the namespace design component. In other words we need to merge the CPN Names proposal with XMPP (JID) Naming. Needs design and implementation.
- Build 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.
- Interface to CA Server
- Interface to DNS Server
- Create XMPP Framework tests using the server and clients
- Best Practices/SDK
- Formalization with Write up Security and Naming w.r.t. a Security Assessment
- Make server accessible to entire team and leave running

Service Framework/Transport Part Development

- Workflow and Message foundational protocols design/development
- Service Catalog development (API Design)
- Service Catalog to servers (Interface)
- Investigation of WebSockets RFC 6455 for the Services Transport
- Services Implementation with API
- Machines talking WebSockets
- Machines talking WebSockets with Services Stubs
- Websockets and XMPP interoperability/mutual handoff
- Integration with XMP server
- Best Practices/SDK for service flows

Overall Semantic Directory/Solver Part Development

- Reference Ontology for Resources
- Reference Schema for Resources
- Reference Ontology for Bearer Network
- Reference Schema for Bearer Network
- Resource Design Prototypes: Compute, BLOB, Replicate, Transcoder, GPU
- Semantic Directory Catalog Repository (Database)
- Put up server running Directory
- Core Solver Design and Implementation – SPARQL on HDFS?
- Put up server running Solver
- Make server and solver accessible to entire team and leave running
- Directory Best Practices/SDK
- Solver Best Practices/SDK

Gateway bundled Protocol/API Part Development

- Bundle modules together for a Client side participant (Gateway Core)
- Management API/Best Practices/SDK
- Put up server running Gateway Core as client
- Make server accessible to entire team and leave running

4. Stand up Protocol Test Jigs. Integrate/Stand a First Root

- The Intercloud Root has quite a few services in it. It contains
    - Naming Part
    - Conversational Part
    - Transport Part
    - Trust/Identity Part
    - Protocol/API Part
    - Semantic Directory Part
    - Audit Part
    - Deployment/Replication Part
- These parts need to be integrated and then deployed using the app blueprint of a replicated, distributed Cloud app.

- Have some test jig on another system where one develops the workflow and the use of the parts into the Intercloud protocol. Eg, implements the state diagrams of the total cycle, going to whichever protocol is needed for that sequenced function.
- This where the interface (protocols and API's) gets proven as workable and least for some operations
- Operationalize Management API
- Get Root to work!
- Make server accessible to entire team and leave running

5. Stand up a First Exchange and have it talk to First Root.

- The Intercloud Exchange has quite a few services in it. It contains
    - Naming Part
    - Conversational Part
    - Transport Part
    - Trust/Identity Part
    - Protocol/API Part
    - Semantic Directory Part
    - Audit Part
    - Deployment/Replication Part

  .. just as the Root does, but adds ..
    - Exchange Broker/Solver/Arbitrage Part

- The Exchange has a different name space from the Root (CS Number). In practice we will have the Root and Exchange be in the same AS number.
- Standup the Exchange functionality, in a networked configuration where the Reference Root has instrument-able and open connectivity between the two. In a rack deployment likely they would be configured as adjacent in Layer 3, eg on the same IP subnet.
- Script API/protocol from one side or other and get communications working.
- Operationalize Management API
- Get Exchange to work!
- Make server accessible to entire team and leave running

6. Stand up Cloud to Exchange Functionality (write Gateway code).

- You are setting out to attach a cloud (of some distribution or architecture) to the Intercloud.
- So first gather up all the core modules (those common to Root and Exchange) and think about how to integrate these to your actual cloud.

- Substrates and communications protocols will be quite portable.
- Creating the "factory" for the federated resource to be created and run, is work specific to each cloud implementation. For example, a "Storage Replication" request has to hook into that clouds replication with a federated replication protocol. Design IaaS "ECU" (Elastic Compute Unit) federation to use VPC-like mechanism? (these are areas which need additional design work). Please refer to the Engineering Plan for more information on this.
- Script API/protocol from one side or other and get communications working.
- Hooking up the "Factory" for that Cloud to the protocols
- Making the Intercloud Gateway work!

7. Spec and Provision the Neutral Reference Root and Exchange locations.

- Would like to strive for Reference Root and Exchange in two different locations. This way we have a basis to server multiple geographies, and also to implement replication and distribution of the Root.
- **Need Donated Gear!!!** Does anyone have an ability to donate 8 to 16 servers to be installed in the US facility, and another 8 to 16 servers to be later installed in a Europe facility?
- The locations should have extensive multi-carrier access capabilities and in number one tier Hosting locations.

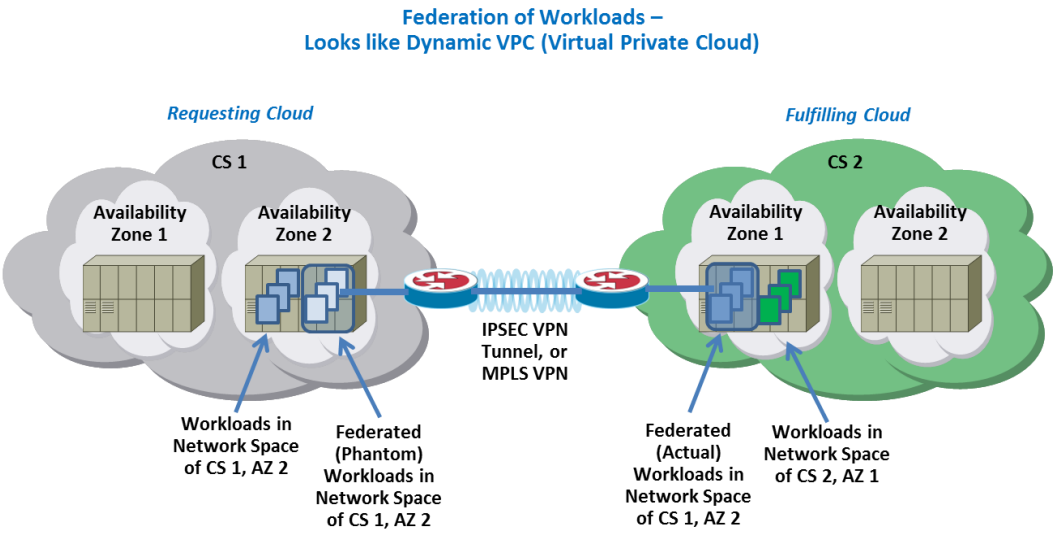
Telx Silicon Valley

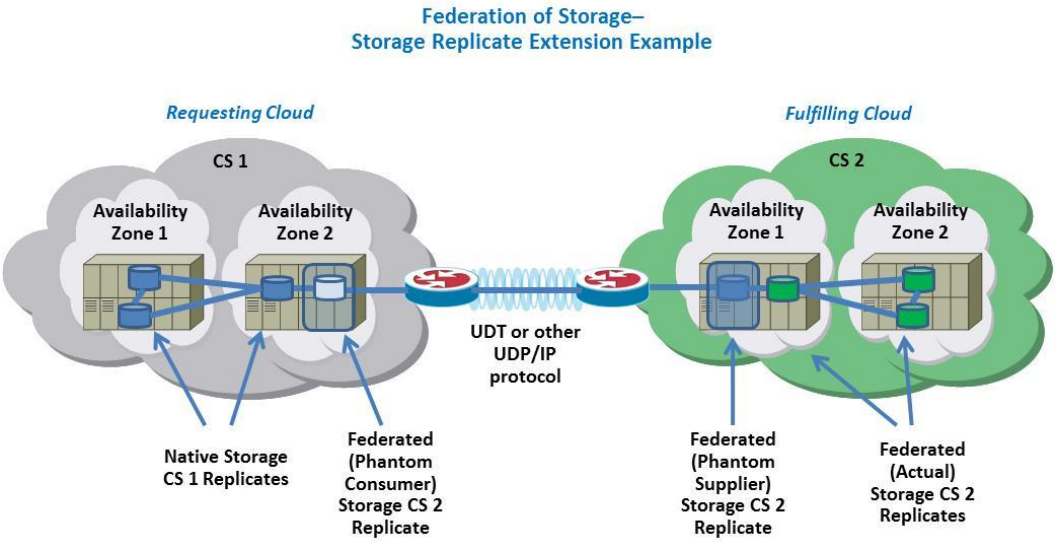
JT in Europe

- Both Root and Exchange will be small clouds themselves, of course. There will be a Root/Exchange pair in each main datacenter. They will likely need to be at least 4 servers, preferably 8. We need to procure servers and top of rack switch, and rack. Installing and sysadm for example needs a resource. Reference software will be open source, OpenStack for example.
- Eventually, need partner with Asia location.
- System will be best installed as an independent AS number and we need to conform to IPv6, to keep most options open for future.
- Naming (so-called CPN) overall architecture, as naming is utilized in many of the sub-systems.
- As all issues of naming/numbering arise, from possible MAC address conflicts to numberspace management for CPN numbers, IEEE will provide assistance on numbering authority.
- Physical server installs. Connectivity, and Cloud OS provisioning.

8. Use Cases

- Begin to write modules and understand how to Federate clouds, at the implementation level, depending on each cloud appropriate model





- Use Cases:

Wireline Service Federation / Intercloud MPLS Integraton	**Telecommunications Carrier**	**Allows the inclusion of Wholesale Virtual Storage and Compute integrated with Wholesale MPLS Virtual Network Offering. Remote resources are made available in the requester's address space through the Virtual Private Cloud Mechanism (VPC). This mechanism is currently anticipated to use SDN and the most logical SDN to use in this case in OpenContrail which uses an MPLS based SDN control protocol.**
Wireline / NREN Service Federation / Intercloud Regional Storage Replication	**Telecommunications Carrier / Cloud Service Provider**	**Allows the replication of storage between providers with high bandwidth, high latency (long distance high speed fiber for example) by transiting VPC based VPN base Storage federation over for example an IPSEC tunnel but instead of using TCP with it's known poor high bandwidth, high latency performance using a UDP based protocol such as UDT. Applies well to NREN based interconnects.**
Open Cloud Exchange style (L0-L2) based Intercloud Federation	**Research Clouds, Grid/Cloud Convergence**	**Allows the use of LAN in-Datacenter or Metro-E connected clouds to federate directly (with or without IPSEC tunneling, depending on address space and security needs) using shared Ethernet transport (so-called Open Cloud Exchange style). Attractive way to stretch Grid/HPC applications adapted to Clouds proper, across more than one cloud without using Multi-cloud techniques.**

Infiniband based Intercloud Federation	**Research Clouds, Grid/Cloud Convergence**	**Speculative Experimentation on using Infiniband to connect trusted clouds for shared memory pooling as a type of Intercloud Federation. Requires investigation of shard memory type as semantic resource and also requires HPC/Grid type Clouds to interconnect intimately via Infiniband. As Infiniband extends over Ethernet may lead to a novel way to dynamically construct shared memory spaces across Intercloud.**
Distributed Switch Fabric based Intercloud Federation	**Storage Service Provider feature of Virtual Portable Data**	**Experimentation with so-called "Powered by Peak Extended Switch Fabric" style network where Storage Service Provider co-locate swith multiple cloud providers in a datacenter and connects storage to clouds via shared distributes switch fabric. Data in Storage Service Provider can virtually appear to move/be available to multiple Cloud Platforms with consistency ensured through Intercloud Federation.**
LAN based Intercloud Federation (requester address space aka VPC)	**Telecommunications Carrier / Cloud Service Provider**	**Canonical case of Intecloud Federation of Storage and Compute using overlay IPSEC Virtual Network. Remote resources are made available in the requester's address space through VPC. This mechanism is currently anticipated to use SDN. OpenContrail (which uses an MPLS based SDN control protocol) is the only open source SDN may not be the only choice given MPLS is not used, might want to use OpenFlow.**

LAN based Intercloud Federation (provider address space aka Roaming)	**Telecommunications Carrier / Cloud Service Provider**	**Variation on canonical Intercloud Federation using overlay IPSEC Transport (not Tunneling) mode. Same as above case but Remote resource are made available in the providers address space. This is a roaming use case where the requester knows they are running on foreign resources (nevertheless automatically federated to them), there is application specific code to address this. SDN is utilized but inverted.**
LAN based Intercloud Federation for Big Data	**Hybrid and/or public cloud applicable**	**Intercloud Federation as in the Roaming use case with application specific code (in this case, distributed Map Reduce/Hadoop) to control and sweep back the multiple Big Data nodes forming a cloud-spanning Intercloud platform for limitless Big Data**
VOIP or SIP / Distributed Transcoder based Intercloud Federation for Voice & Video; Fixed & Mobile variations	**OTT Internet Phone apps, or LTE Network Substrate**	**Intercloud Federation as in the Roaming use case except using additional layer of VOIP or SIP protocols to set up multiple cooperating endpoints for voice or video calls. Use case requires development of transcoder as a federated resource. Applications specific code from call set up to adaptive transcoder provisioning. Enablement of Location Based information into Intercloud Requester data structures if access to the back of a Carrier LTE network can be had. So called specific code can be placed in user space or an investigation of the Intercloud XMPP substrate with the LTE SIP substrate will be in order to more firmly mobile enable the very design core of Intercloud Federation**

Intercloud Federation for Internet of Things, Connected Vehicle Example	**Automobile with Carrier Partner**	**Intercloud Federation as in the Roaming use case, leveraging the mobile integration, experimenting with applications in Connected Vehicle. Self driving via Local Cloud and Carrier control is a must. Extension to safety features including multiple auto dynamics (traveling in packs, collision avoidance) as well as augmented reality (Intercloud Federated Deep Web) for concierge, shopping, and advertising.**
Intercloud Federation for Internet of Things, Smart Grid/Sensors Example	**Utilities companies, smart homes, Carrier Partner**	**Intercloud Federation as in the Roaming use case, but where the smart elements (meters/sensors) are static location. Because broadband networks are asymmetric (much slower upload), in commercial deployments of smart grids/meters, carriers provide lower cost off hour network access (Wired, SMS, 3G/4G data) for upload. This causes small time windows for uploads and processing. It is well known that in many sensor applications burst windows are major processing challenges. Most sensor processing uses event processing models (Storm/Kafka, or in-memory sale-out DBs like VoltDB) to process sensor inputs. This experiment examines burst window dynamic Intercloud federation for supporting burst window capacities of event processing models.**
Core Intercloud Standards Framework	**Financial Commoditization of Cloud**	**In order to commoditize Cloud Resources suitable for automated exchange based commerce, both object and method standard must be put into place, eg, semantic resource ontologies and core federation transports and mechanisms. This effort formalizes these in the context of automated commodity exchange mechanism.**

Geographic Governance Models	**Policies of Countries w.r.t. Intercloud Governance; Cloud as National Asset**	**There are various Trust Hierarchy/Proxy models for Intercloud, starting with "local" and "foreign" exchange (think local and long distance telephone call model) all the way to geographical or economic/political. This experiment models the effects and impacts of Governance/Policy Experimentation for Societal and Sustainability outcome analysis**
Commodities Trading Enablement	**Financial Commoditization of Cloud**	**Simple "Peering" as exchange (original Internet model) is surely not how cloud resources will be federated. This experiment involves enablement of the core Intercloud Platform with ability for multiple extension and algorithmic experimentation to enable automated commodities of cloud resources through Intercloud Federation. The result will be creation of new Industries/Business Models of Clouds.**
Intercloud Advanced Econometrics	**Financial Commoditization of Cloud**	**This experiment builds on the Commodities Trading Enablement experiments to model several so called exotic Economic Model Experiments including Trading, Arbitrage, Derivatives, Hedging, and Volatility**