

# IEEE Intercloud Testbed Engineering Bootstrap

---

Project Notes  
Last Edit 8/30/2013

## Contents

Introduction to Intercloud Testbed Bootstrap Strategy .....	2
1. Set up the Engineering Collaboration Site (Central Desktop Wiki). .....	2
2. Finish the Master Design and Chunk the Work-areas. ....	2
3. Recruit and Identify Resources to lead Work-areas.....	5
4. Spec and Provision the Neutral Reference Root and Exchange locations. ....	5
5. Identify Local Labs. ....	6
6. Do initial work per Work-area. ....	6
7. Stand up a First Root and some Protocol Test Jigs. ....	6
8. Stand up a First Exchange and have it talk to First Root. ....	7
9. Stand up Cloud to Exchange Functionality (write Gateway code). ....	7
10. Member Cloud Join and Federate for use case .....	8



## Introduction to Intercloud Testbed Bootstrap Strategy

In order to begin the Testbed, the following activities will need to run somewhat in parallel:

1. **Set up the Engineering Collaboration Site** (Central Desktop Wiki). Here we can put all of our working documents, coordinate activities, etc. Set up Code repository.
2. **Finish the Master Design and Chunk the Work-areas.** Assemble the core initial engineering team, and design review the proposed master design. There are several interesting and unfinished design areas, more will be discovered. Create engineering specifications for each topological area, and for each sub-system.
3. **Recruit and Identify Resources to lead Work-areas.** Understand what we have in terms of actual engineers.
4. **Spec and Provision the Neutral Reference Root and Exchange locations.** Procure systems and networking for them, begin the wheels rolling to put racks in each place. Need Donated Gear!
5. **Identify Local Labs** where early stage work can occur. Some Local Labs will also reference a lead for a flavor of a participating Cloud System.
6. **Do initial work per Work-area.** Collect base code for Open Source parts. Identify completely new coding efforts required.
7. **Stand up a First Root and some Protocol Test Jigs.** Poke the Root and be able to test jig through each major sub-system.
8. **Stand up a First Exchange and have it talk to First Root.** Roots only talk to Exchanges. Initially connectivity will be on one Root to one Exchange basis (replication of Roots; m:n Root/Exchange ratios later)
9. **Stand up Cloud to Exchange Functionality.** First use case/logic module to perform a workflow of cloud interoperability put on Exchange side. Includes from API's to solver pipeline and computation. Initially, simplify.
10. **Member Cloud Join and Federate for use case.** First test jig of Member cloud, with conversational models to work out logic flow of brokering, etc..

### 1. Set up the Engineering Collaboration Site (Central Desktop Wiki).

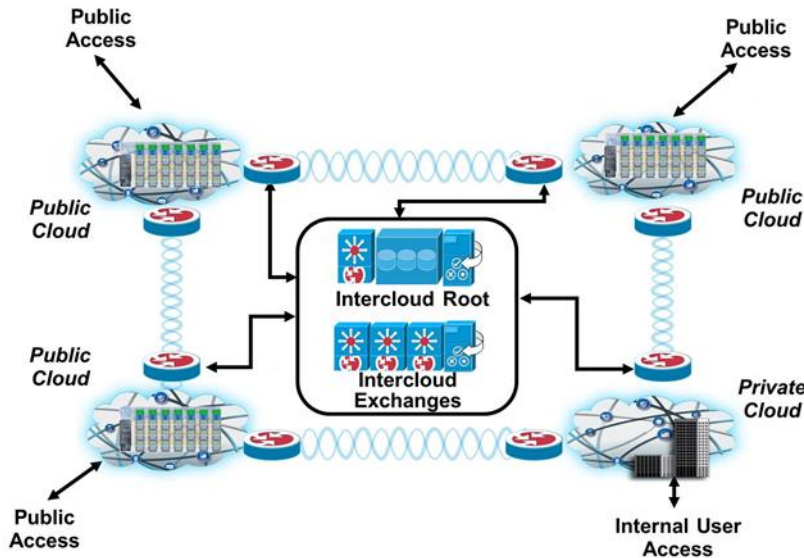
- IEEE has provided a collaboration site through a provider called Central Desktop, who is a “value added Wiki provider”. Every Member, individual Subject Matter Expert, and otherwise participant will have access to the Wiki.
- The Wiki will be password protected from the public.
- All documents posted to the Wiki are subject to the IPR policy of the Testbed project, which is that all documents are “open source” and licensed under the BSD Copyright license.

### 2. Finish the Master Design and Chunk the Work-areas.

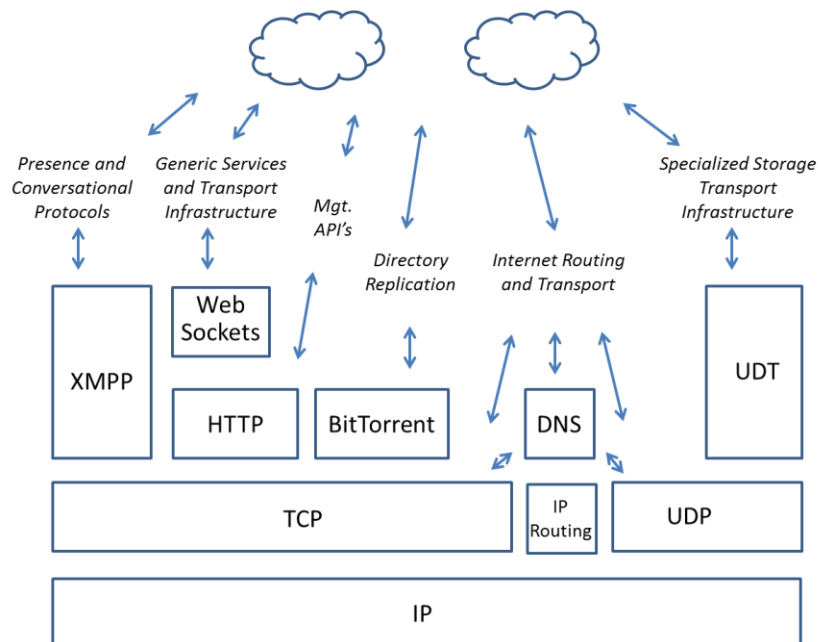
- An end to end MeetUp/design review session on the proposed architecture should be gone through.



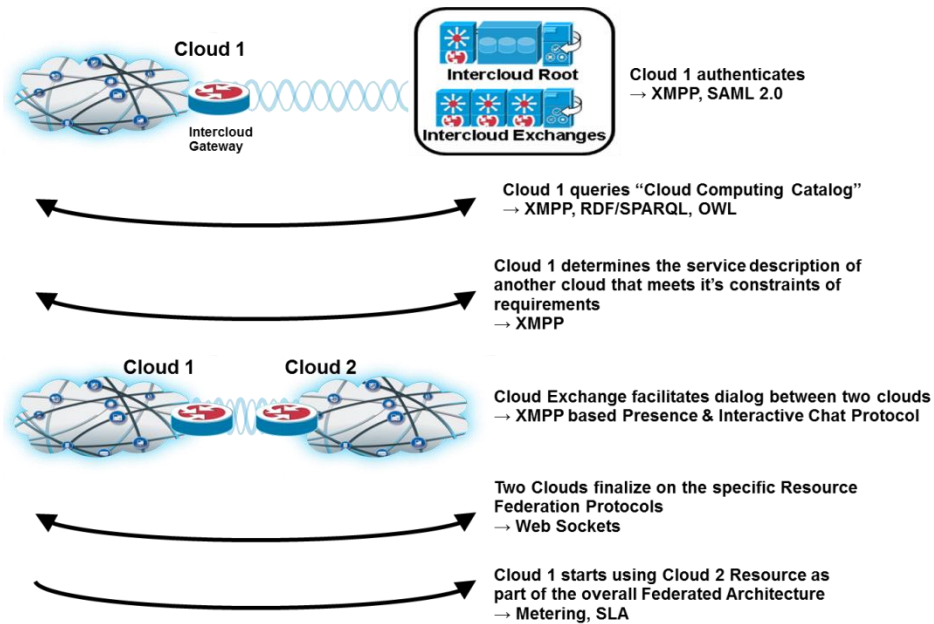
- There is a whole-system design philosophy and scheme.



- There are also working assumptions for implementation choices for many of the areas. All of these should be open to analysis as to the now best approach available.



- While some areas are quite well thought out, with module implementations from the open source work.



- Here is a tentative list of main design areas across which we will have these design reviews, specs written, etc:

- 1) Completion of Master Technical Design Work
- 2) Collaboration, Source Code, Specs, Internal and Public Site(s)
- 3) Reference Root(s) Infrastructure, Physical and Networks
- 4) Reference Exchange(s) Infrastructure, Physical and Networks
- 5) Overall (Naming Part) Development and Policy/Procedures and Governance
- 6) Overall (Conversational Part) Development
- 7) Overall (Transport Part) Development
- 8) Overall (Trust/Identity Part) Development
- 9) Overall (Protocol/API Part) Development
- 10) Overall Root (Semantic Directory Part) Development
- 11) Overall Root (Audit Part) Development
- 12) Overall Root and Exchange (Deployment/Replication Part) Development
- 13) Reference Root (Integration of above services) Integration/Development
- 14) Overall Exchange (Solver/Arbitrage Part) Development
- 15) Reference Exchange (Integration of above services) Integration/Development
- 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

### 3. 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.

### 4. 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.



- 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 CS) overall architecture needs to be completed, 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 CS numbers, IEEE will provide assistance on numbering authority.
- Physical server installs. Connectivity, and Cloud OS provisioning.

## 5. 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.
- Identify upcoming projects and fundings for synergies, or for Students or Projects “looking for a purpose”.

## 6. Do initial work per Work-area.

- Work Area leads are stepping up get that work area prototyped and working, even if they don't have the resources needed.
- It's not a commitment, it's a leader, you do the best with what you can
- Some areas, maybe many, might start off a little slow as we recruit people. Please be patient as we move the ball forward.
- Many areas apply to all of the modules (the core protocols). Work on those first,
- Regular effort-wide distributed meetings.
- Collaboration area use.

## 7. Stand up a First Root and some Protocol Test Jigs.

- 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
- Get Root to work!

## 8. 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.

## 9. 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)



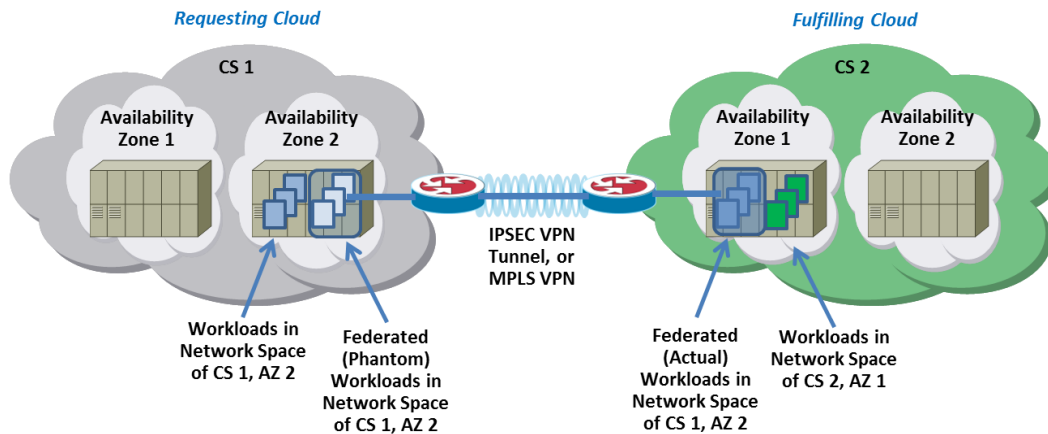
federation to use VPC-like mechanism? (these are areas which need additional design work). Please refer to the Engineering Plan for more information on this.

- 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!

## 10. Member Cloud Join and Federate for use case

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

**Federation of Workloads –  
Looks like Dynamic VPC (Virtual Private Cloud)**



**Federation of Storage –  
Storage Replicate Extension Example**

