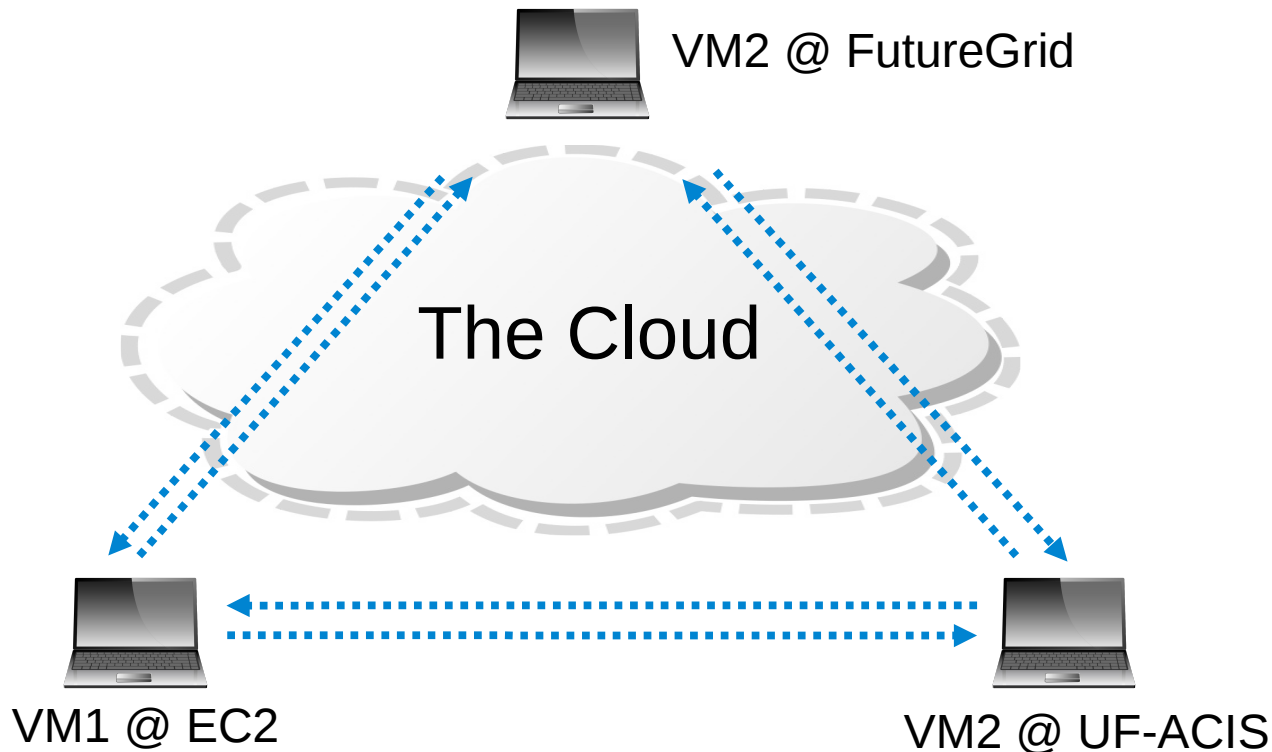


IPOP and SocialVPN Demonstrations

Pierre St Juste, Renato Figueiredo
Advanced Computing and Information Systems Lab
University of Florida
March 22, 2013

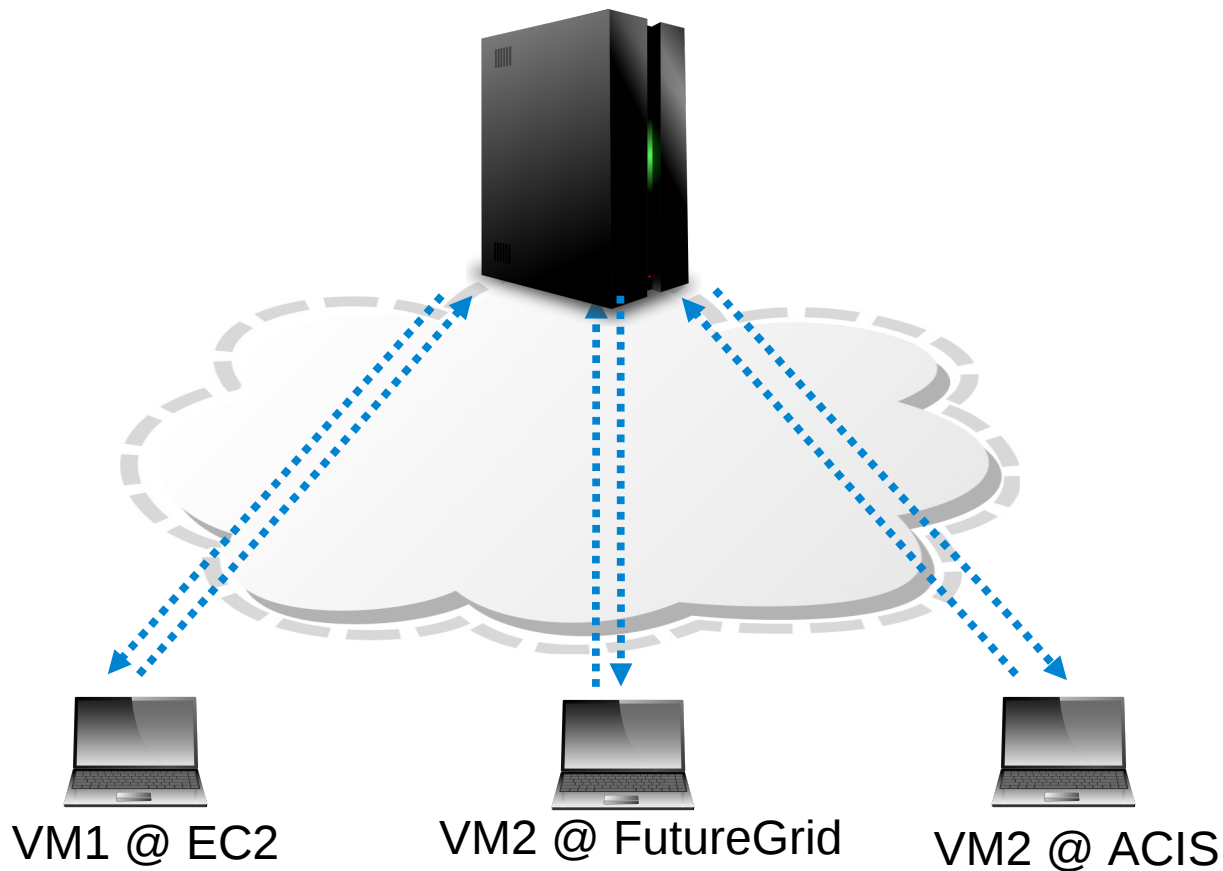
Motivation

Virtual machines **want to communicate as if they are part of the same LAN**, this can be difficult when they are **located at different cloud providers** or administrative domains



Motivation

Naive Centralized VPN Gateway (OpenVPN)



1. Performance

All IP packets are routed through the gateway causing extra latency and bandwidth bottlenecks

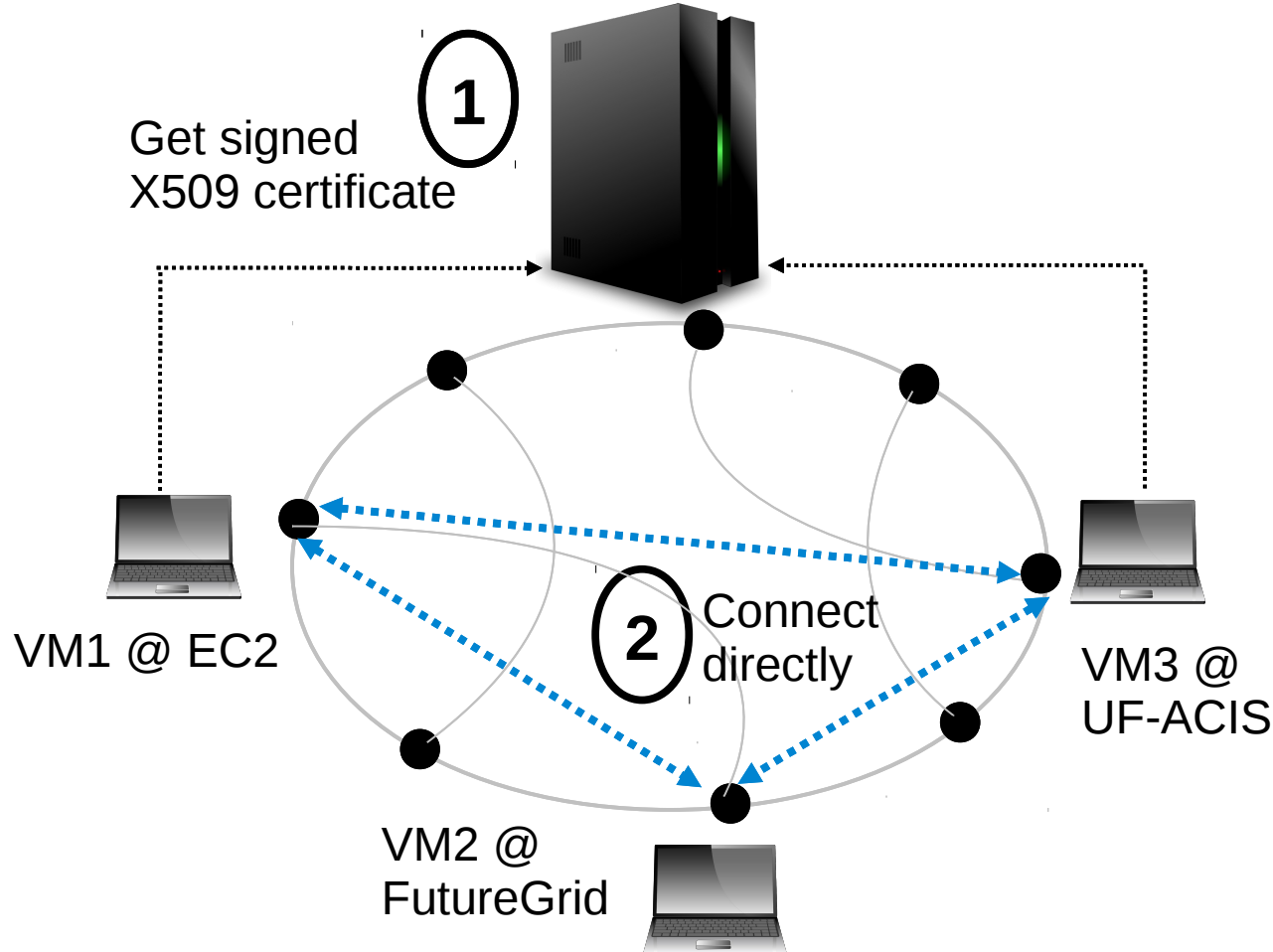
2. Lack of trust

All IP packets are encrypted and decrypted at the gateway possibly leading to undetectable packet spoofing

3. Single Point

Gateway maintains all routing state, along with assignment of IP address thus creating a single point of failure

The Peer-to-Peer Approach



1. Performance

IP packets are sent over direct P2P tunnels thus removing the extra latency and bandwidth limitations of the gateway approach

2. Security

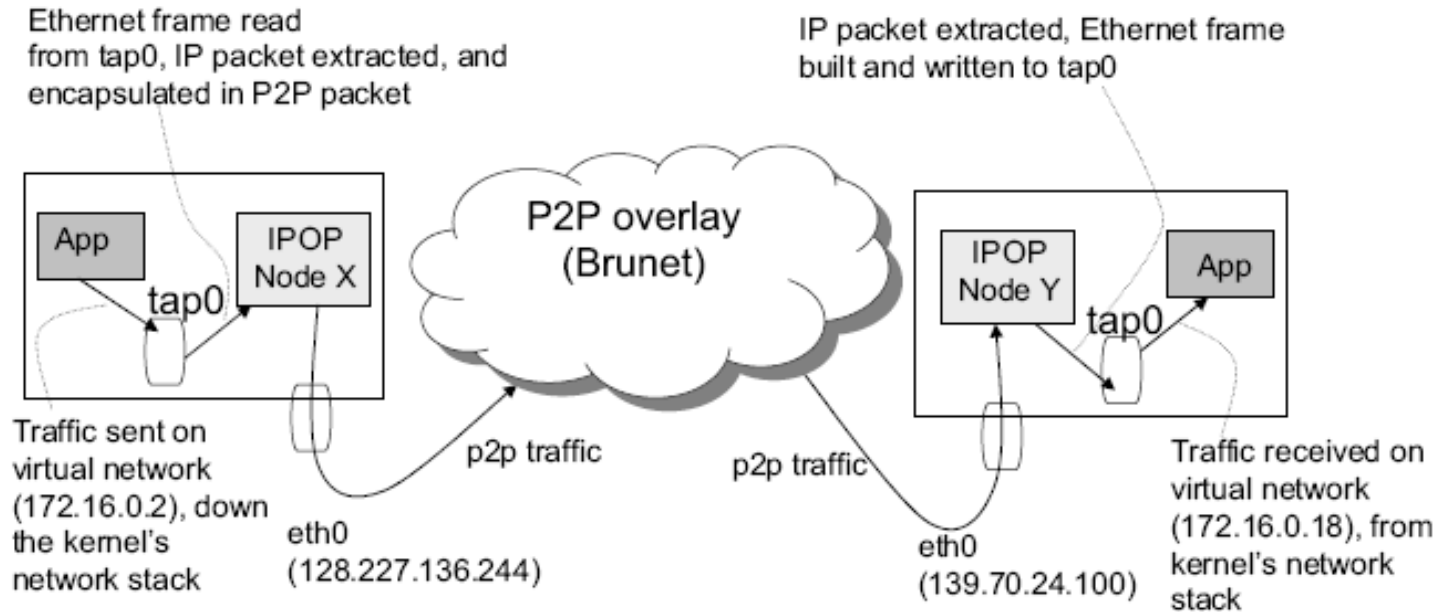
End-to-end encryption (e.g. IPSec) ensures no middleman can modify a packet unnoticed

3. Low maintenance

P2P technology provides self-managing/self-adapting network with no single-point of failure

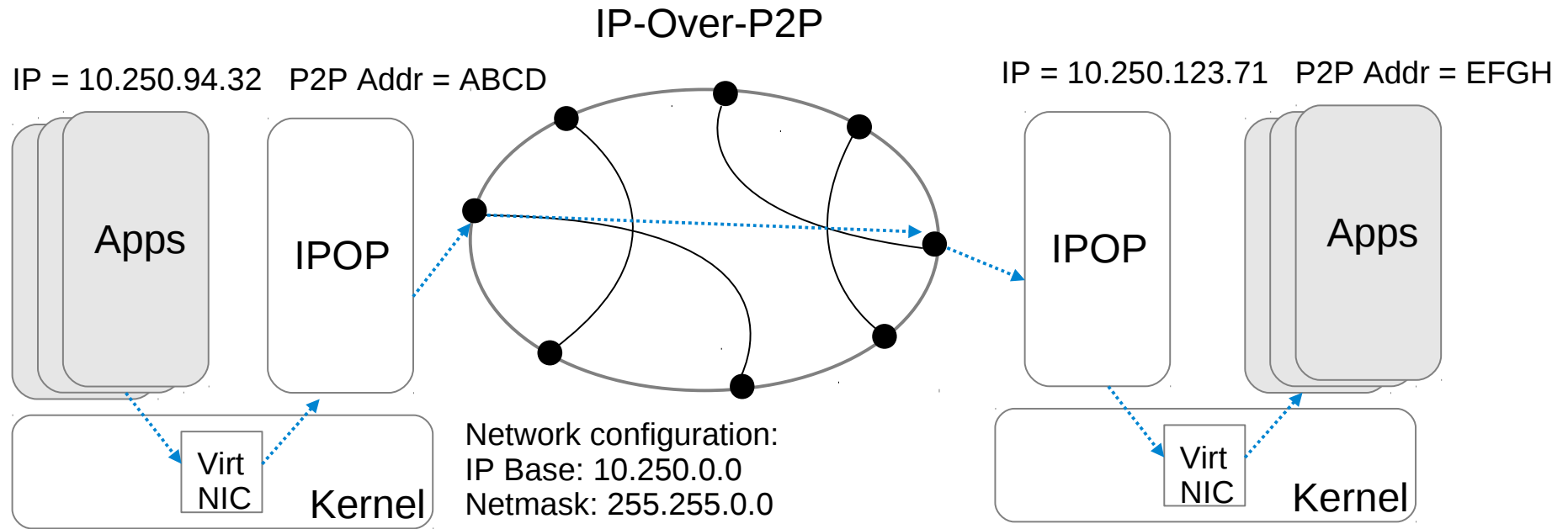
Use **P2P technology** to remove the inefficiencies of the centralized approach, the gateway simply functions as a **certificate authority**

IPOP Design (1)



- Virtual LAN
 - Self-configuring, decentralized virtual network
 - P2P VPN routes IP packets over P2P overlay
 - Builtin security stack (or IPSec)
- P2P Overlay
 - Structured P2P overlay
 - NAT/firewall traversal
 - Robust, scalable, and self-organizing system

IPOP Design (2)



IP Allocation

IP addresses are assigned to nodes through the use of a DHT by performing a lookup to see if the key already exists, if so a different IP is looked up until an unallocated IP is found

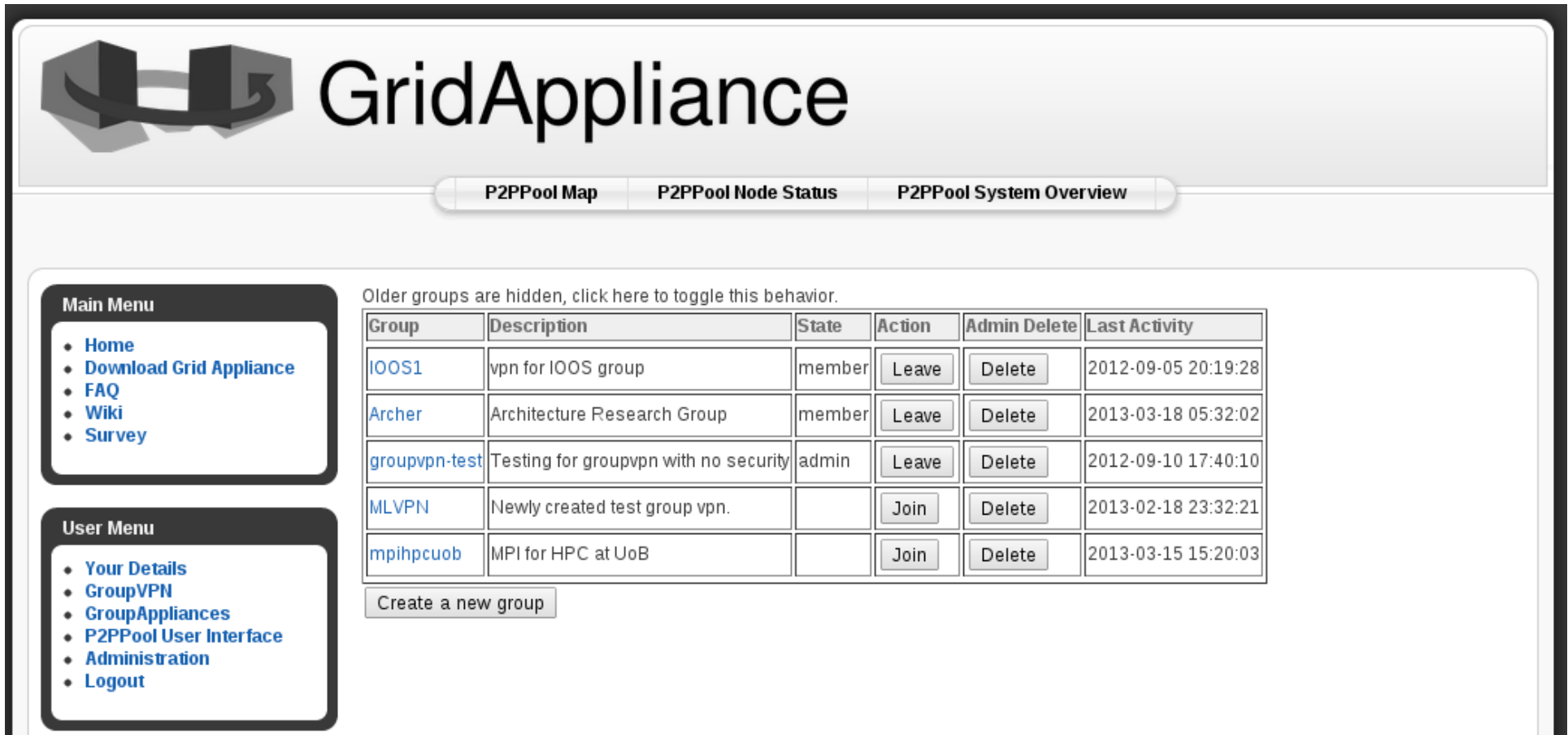
IP Resolution

IPOP uses the DHT to lookup the P2P address belonging to the destination virtual IP address

IP Tunneling

IP packets addressed to the virtual IP range are captured by virtual NIC, given to IPOP and tunneled through Brunet

GroupVPN (1)



The screenshot shows the GridAppliance web interface. At the top, there is a logo and the title "GridAppliance". Below the title, there are three tabs: "P2PPool Map", "P2PPool Node Status", and "P2PPool System Overview". On the left side, there are two menu boxes: "Main Menu" and "User Menu". The "Main Menu" contains links to Home, Download Grid Appliance, FAQ, Wiki, and Survey. The "User Menu" contains links to Your Details, GroupVPN, GroupAppliances, P2PPool User Interface, Administration, and Logout. The main content area displays a table of VPN groups. Above the table, there is a note: "Older groups are hidden, click here to toggle this behavior." The table has columns for Group, Description, State, Action, Admin Delete, and Last Activity. The groups listed are IOOS1, Archer, groupvpn-test, MLVPN, and mpihpcuob. Each group has a "Leave" or "Join" button and a "Delete" button. Below the table, there is a button labeled "Create a new group".

GridAppliance

P2PPool Map P2PPool Node Status P2PPool System Overview

Main Menu

- Home
- Download Grid Appliance
- FAQ
- Wiki
- Survey

User Menu

- Your Details
- GroupVPN
- GroupAppliances
- P2PPool User Interface
- Administration
- Logout

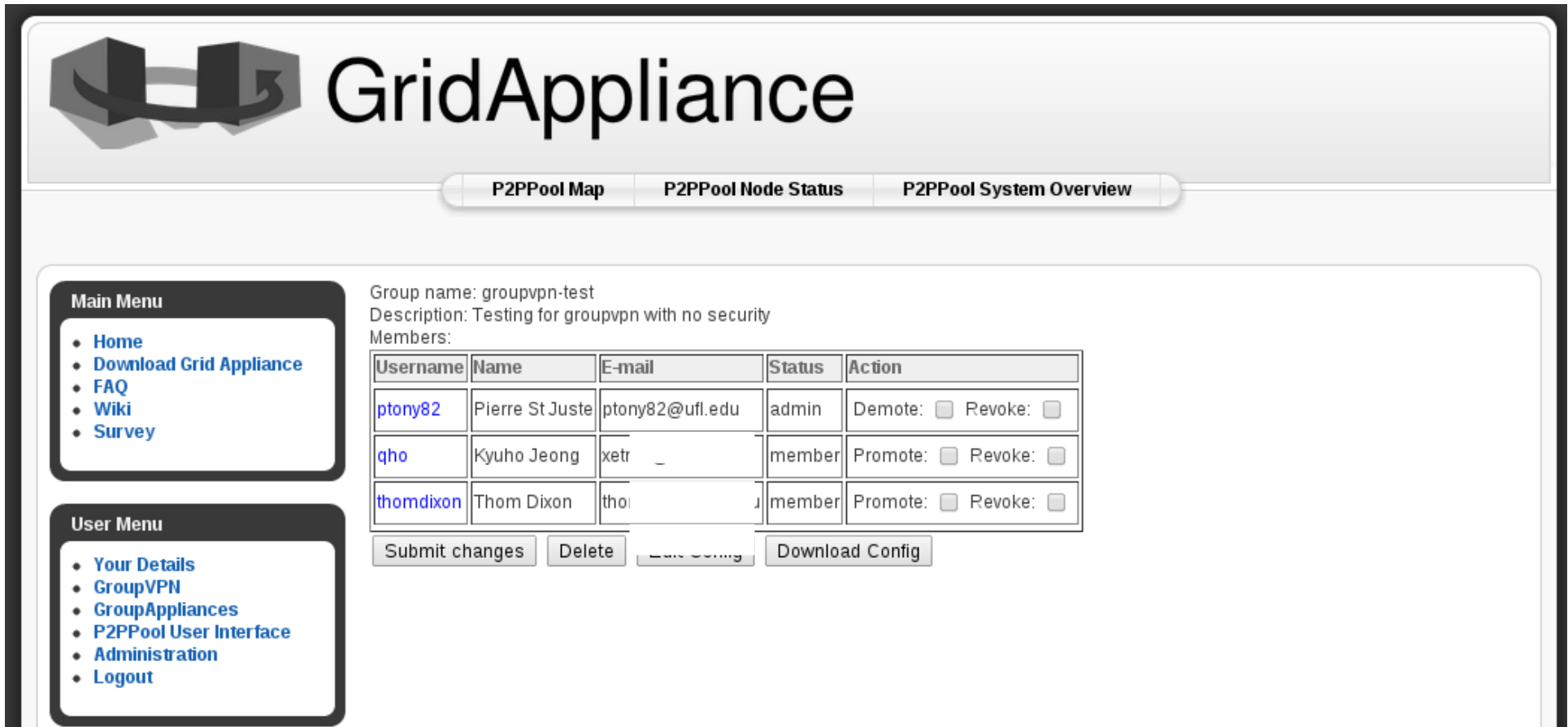
Older groups are hidden, click here to toggle this behavior.

Group	Description	State	Action	Admin Delete	Last Activity
IOOS1	vpn for IOOS group	member	Leave	Delete	2012-09-05 20:19:28
Archer	Architecture Research Group	member	Leave	Delete	2013-03-18 05:32:02
groupvpn-test	Testing for groupvpn with no security	admin	Leave	Delete	2012-09-10 17:40:10
MLVPN	Newly created test group vpn.		Join	Delete	2013-02-18 23:32:21
mpihpcuob	MPI for HPC at UoB		Join	Delete	2013-03-15 15:20:03

[Create a new group](#)

Membership to the network is managed through the grid-appliance.org website or users can run their own web service (the code is freely available on github)

GroupVPN (2)



The screenshot shows the GridAppliance web interface. At the top, there is a logo and the title "GridAppliance". Below the title, there are three tabs: "P2PPool Map", "P2PPool Node Status", and "P2PPool System Overview". The "P2PPool Node Status" tab is selected.

On the left side, there are two menu boxes:

- Main Menu**
 - Home
 - Download Grid Appliance
 - FAQ
 - Wiki
 - Survey
- User Menu**
 - Your Details
 - GroupVPN
 - GroupAppliances
 - P2PPool User Interface
 - Administration
 - Logout

The main content area displays the following information:

Group name: groupvpn-test
Description: Testing for groupvpn with no security
Members:

Username	Name	E-mail	Status	Action
ptony82	Pierre St Juste	ptony82@ufl.edu	admin	Demote: <input type="checkbox"/> Revoke: <input type="checkbox"/>
qho	Kyuho Jeong	xetr	member	Promote: <input type="checkbox"/> Revoke: <input type="checkbox"/>
thomdixon	Thom Dixon	tho	member	Promote: <input type="checkbox"/> Revoke: <input type="checkbox"/>

Below the table, there are buttons: "Submit changes", "Delete", "Download Config", and a partially visible "Cancel" button.

Each user has to go to the management site to download their configuration which includes a signed X.509 certificate, a list of bootstrap nodes for the P2P overlay, and the namespace for the

Condor over
Wide Area

Packaged as
virtual machine

Supports
Vmware, Xen,
Virtualbox, KVM,
EC2, FutureGrid

VMs join the
same VPN over
wide area

```
Grid Appliance XMonitor
Welcome to the Grid Appliance
Your user name is ptony82
The default password is password

System Information
The appliance can be access via: False
Networking is currently running
Grid middleware is currently running

ptony82@localhost: ~
slot9@C113202001.i LINUX X86_64 Unclaimed Idle 0.000 4032201+12:18:42
slot1@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 52+09:36:11
slot2@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 52+09:36:17
slot3@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:15
slot4@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:16
slot5@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:17
slot6@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:18
slot7@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:19
slot8@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:12
slot9@C179250183.i LINUX X86_64 Owner Idle 0.000 16081303+08:05:24
slot10@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 0+00:35:05
slot11@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032102+14:58:03
slot12@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032102+14:58:04
slot1@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:46:57
slot2@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:02
slot3@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:03
slot4@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:04
slot5@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:03
slot6@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:04
slot7@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:05
slot8@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:04
slot9@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:06
slot10@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:42
slot11@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032 0+07:45:09
slot12@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+18:37:10
slot1@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:41
slot2@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+18:37:08
slot3@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:43
slot4@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:44
slot5@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032 66+23:47:18
slot6@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+10:36:03
slot7@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+10:31:26
slot8@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:35
slot9@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+10:36:20

Total Owner Claimed Unclaimed Matched Preempting Backfill
INTEL/LINUX 1 1 0 0 0 0 0
X86_64/LINUX 78 2 0 76 0 0 0
Total 79 3 0 76 0 0 0
ptony82@localhost:~$
```



1) Download GridAppliance VM with Condor pre-installed

2) Download customized floppy from GridAppliance website

3) Start your VM and run condor_status

The screenshot shows two windows from a Grid Appliance. The top window, titled 'Grid Appliance XMonitor', displays a welcome message and system information. The bottom window shows the output of the 'condor_status' command, listing various slots and their states.

Grid Appliance XMonitor

```
Welcome to the Grid Appliance
Your user name is ptony82
The default password is password

System Information
The appliance can be access via: False
Networking is currently running
Grid middleware is currently running
```

condor_status

```
ptony82@localhost: ~
slot9@C113202001.i LINUX X86_64 Unclaimed Idle 0.000 4032201+12:18:42
slot10@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 52+09:36:11
slot20@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 52+09:36:17
slot30@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:15
slot40@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:16
slot50@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:17
slot60@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:18
slot70@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:19
slot80@C179250183.i LINUX X86_64 Unclaimed Idle 0.000 2010 84+07:28:12
slot90@C179250183.i LINUX X86_64 Owner Idle 0.000 16081303+08:05:24
slot100@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 0+00:35:05
slot110@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032102+14:58:03
slot120@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032102+14:58:04
slot130@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:46:57
slot140@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:02
slot150@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:03
slot160@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:04
slot170@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:03
slot180@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:04
slot190@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:05
slot200@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:04
slot210@C230146205.i LINUX X86_64 Unclaimed Idle 0.000 4032 52+09:47:06
slot220@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:42
slot230@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032 0+07:45:09
slot240@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+18:37:10
slot250@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:41
slot260@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+18:37:08
slot270@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:43
slot280@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:44
slot290@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032 66+23:47:18
slot300@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+10:36:03
slot310@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+10:31:26
slot320@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+14:36:35
slot330@C231037236.i LINUX X86_64 Unclaimed Idle 0.000 4032201+10:36:20

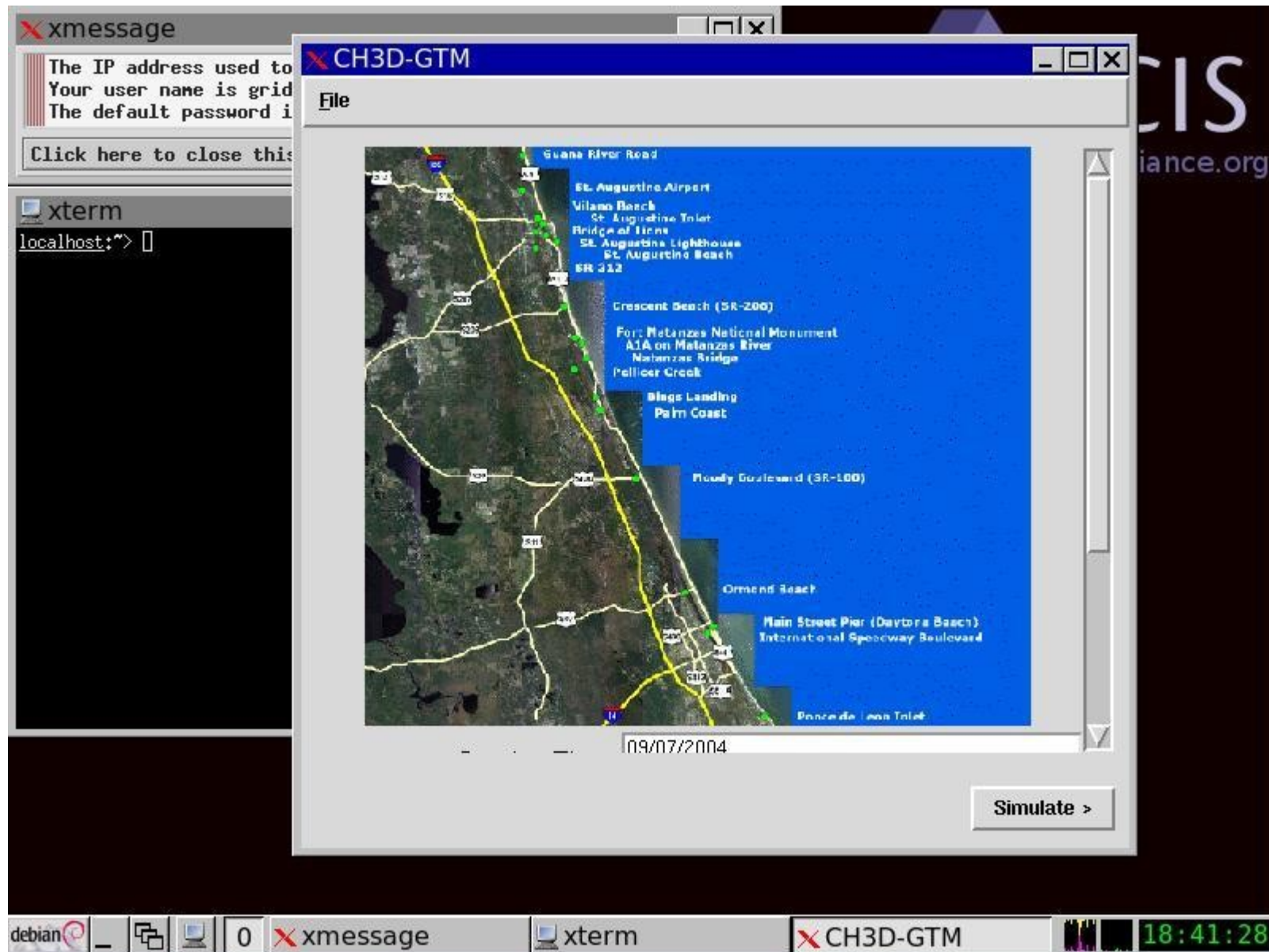
Total Owner Claimed Unclaimed Matched Preempting Backfill
INTEL/LINUX 1 1 0 0 0 0 0
X86_64/LINUX 78 2 0 76 0 0 0
Total 79 3 0 76 0 0 0
```



Easy with very little management



Coastal Ocean Observing and Prediction with Grid-Appliance



Coastal Ocean Observing and Prediction with Grid-Appliance

CH3D-GTM Interface - Windows Internet Explorer

http://192.168.125.128/CI-TEAM/v2/

File Edit View Favorites Tools Help

CH3D-GTM Interface

Start Date : 31

End Date : 31


Release location :

Number of vertical layers : 4 Layers

Include Wind : Yes : ☒ No : ☐

Include River Discharge : Yes : ☒ No : ☐

Select model parameters and click "Run Simulation" button to start simulation



The map displays the St. Augustine area, including St. Augustine Inlet Entrance, St. Augustine Beach, Crescent Beach, AIA on Matanzas River, Bings Landing, Moody Boulevard, Daytona Beach, Ponce de Leon Inlet Entrance, Ormond Beach, Palm Coast, and Pellicer Creek. A scale bar indicates 50 km and 20 mi. The coordinates 0.00000, 0.00000 are shown at the bottom right of the map area.

IPOP/Grid-appliance Talking Points

- 1) Grid-appliance image running on EC2 and FutureGrid**
- 2) Networking without a gateway using P2P technology**
- 3) Self-managing for years**

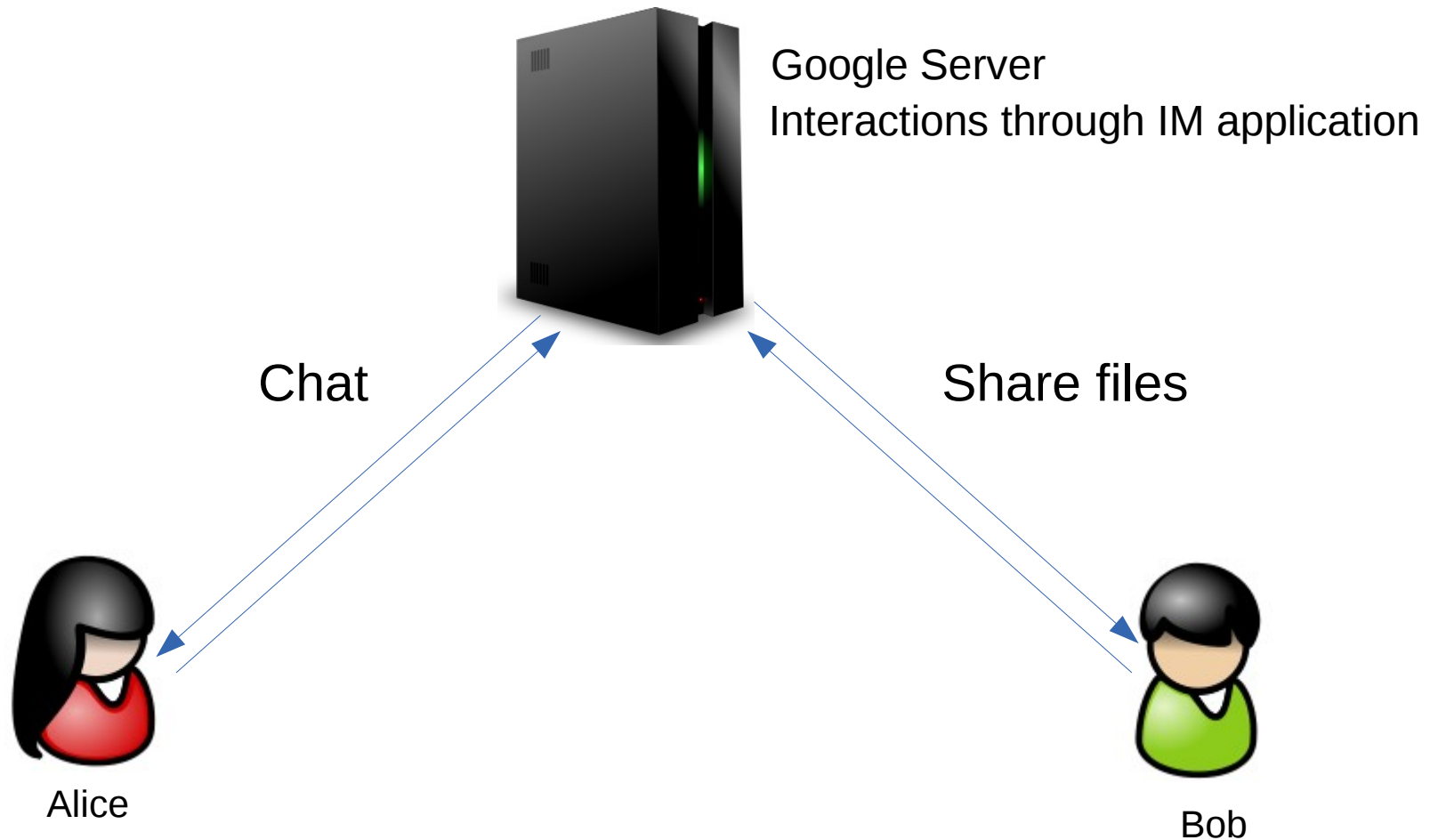
IPOP Demo Details

1. Create/Join virtual network group on grid-appliance.org
2. Download configuration from grid-appliance.org
3. Start Ubuntu 12.04 LTS instance on EC2
4. Add Debian repository to Ubuntu package manager
5. Install ipop using apt-get
6. Upload configuration to EC2
7. Start IPOP and connect over VPN

What is SocialVPN?

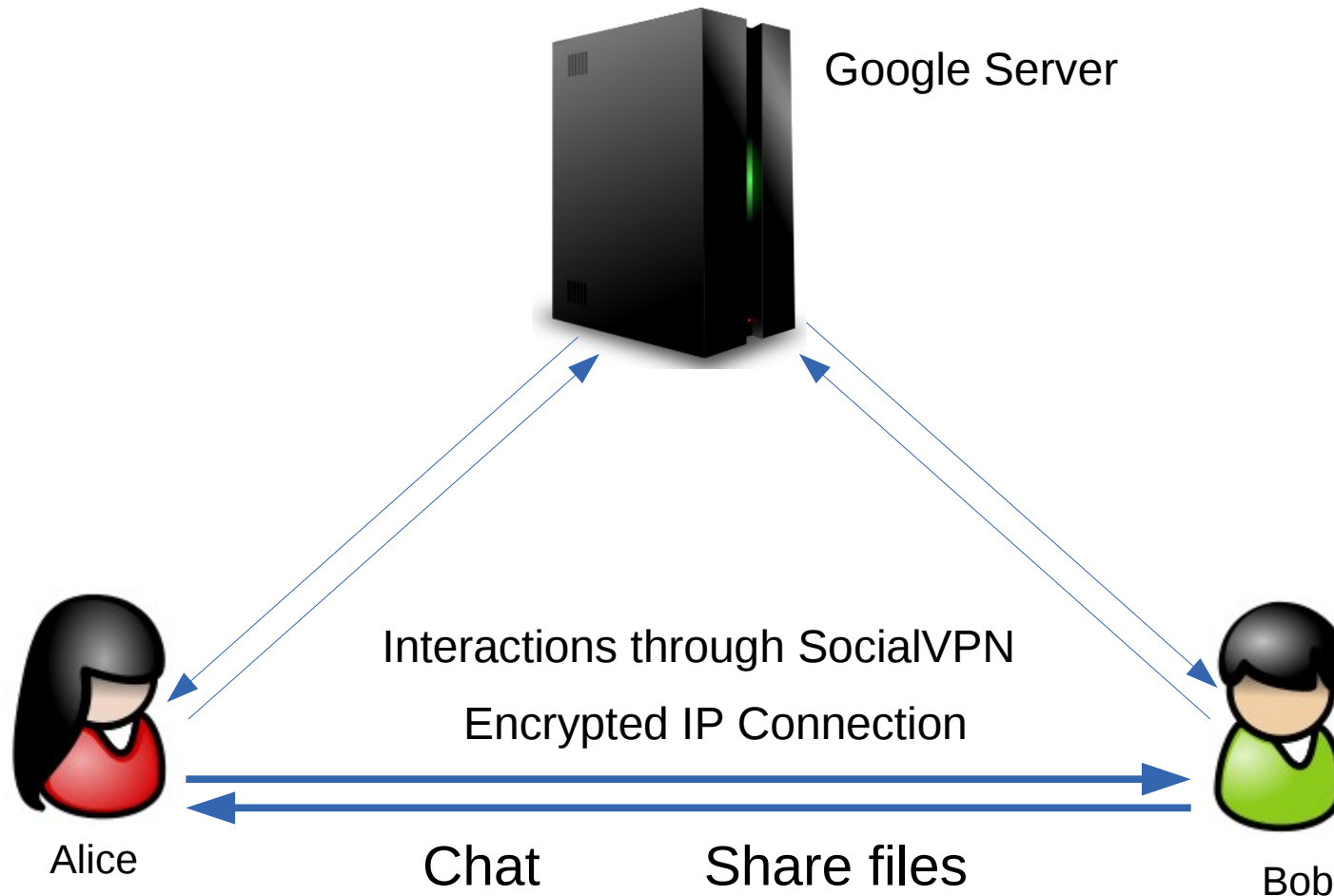
SocialVPN is a free and open-source **P2P Social Virtual Private Network** that seamlessly networks your computer with your friends' computers

Alice and Bob are Google Chat buddies



Other desktop applications cannot take advantage of this communication link between Alice and Bob

Alice and Bob are Google Chat buddies



SocialVPN extends this social link between Alice and Bob into an encrypted network level connection so that **other applications such as iTunes can also communicate through this link**

What you can do with SocialVPN:

Access a folder on your friend's PC directly from your PC without installing any new software

Share your iTunes playlist with your friends

Play multi-player LAN games with your friends

Access your home PC from anywhere

Run a website on your computer that only your friends can access

And much more...

What makes SocialVPN different?

Distributed NAT Traversal

- P2P overlay is used as a distributed STUN server

Centralized backend is not required

- Peers can exchange certificate fingerprints over the phone

Open-source, and mature code base

- Has been running on Planetlab for over 5 years
- Anyone can contribute

SocialVPN Design (1)

Carol's Mappings

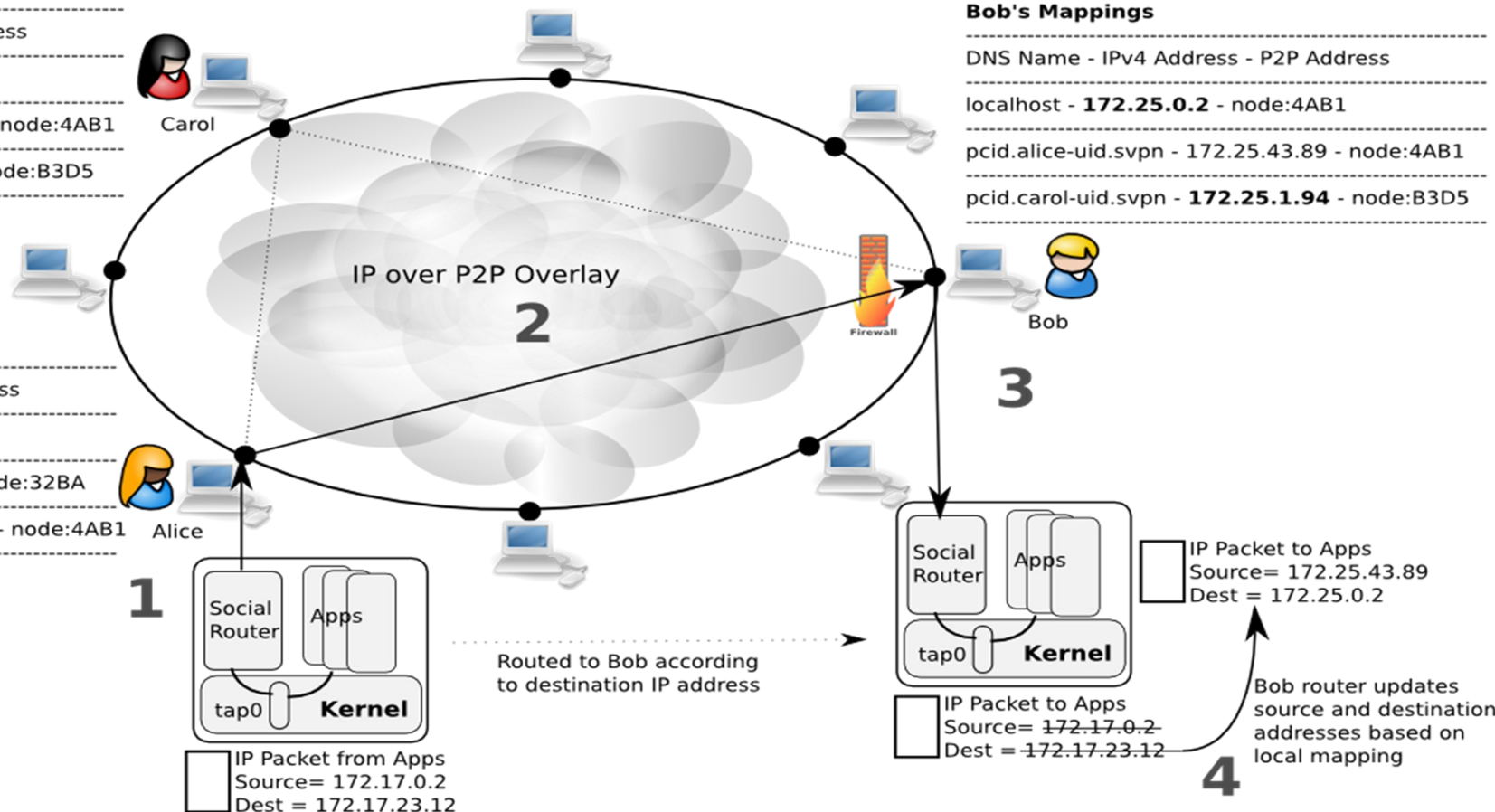
DNS Name	IPv4 Address	P2P Address
localhost	172.31.0.2	node:32BA
pcid.alice-uid.svpn	172.31.143.12	node:4AB1
pcid.bob-uid.svpn	172.31.21.31	node:B3D5

Alice's Mappings

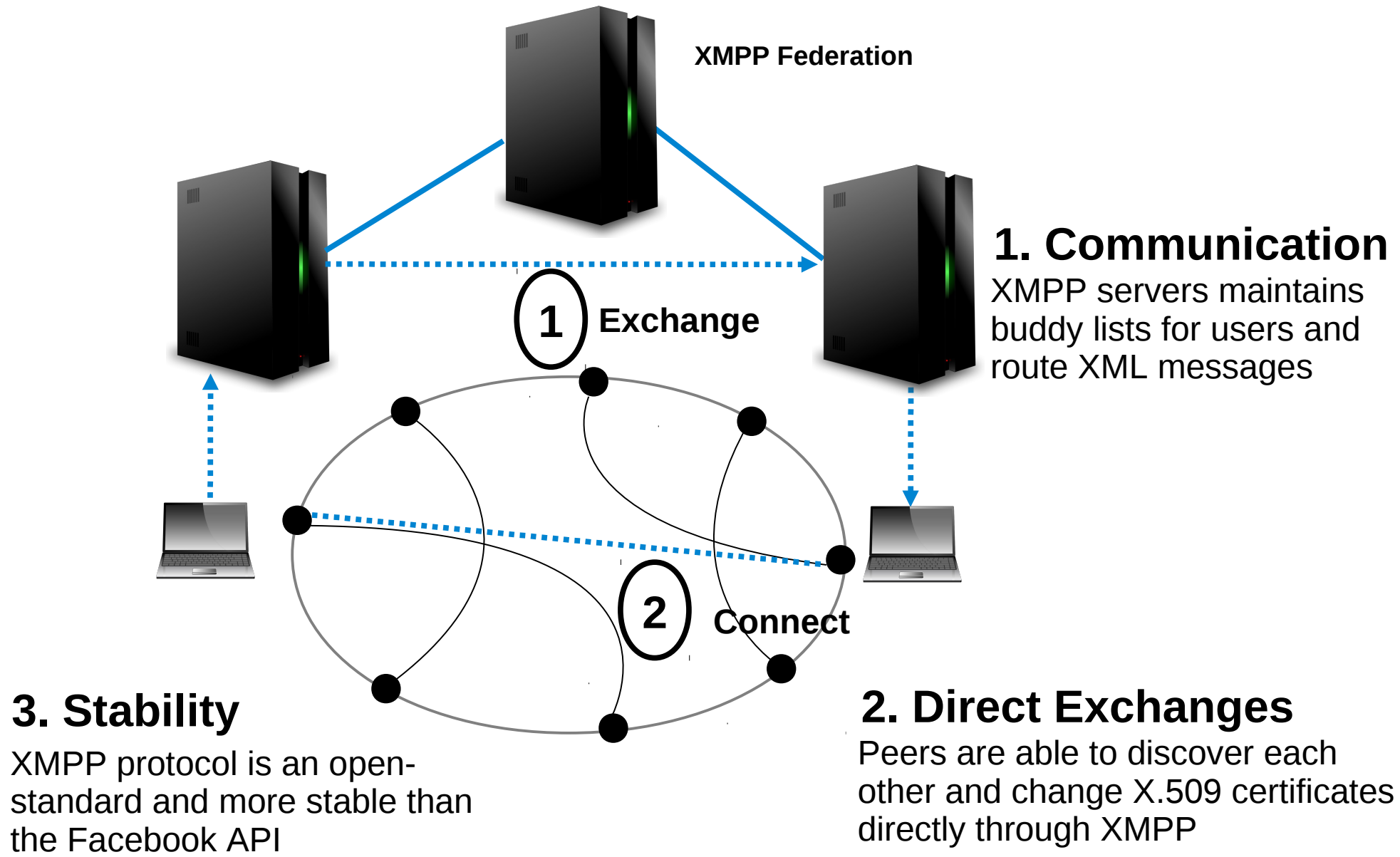
DNS Name	IPv4 Address	P2P Address
localhost	172.17.0.2	node:B3D5
pcid.bob-uid.svpn	172.17.23.12	node:32BA
pcid.carol-uid.svpn	172.17.34.231	node:4AB1

Bob's Mappings

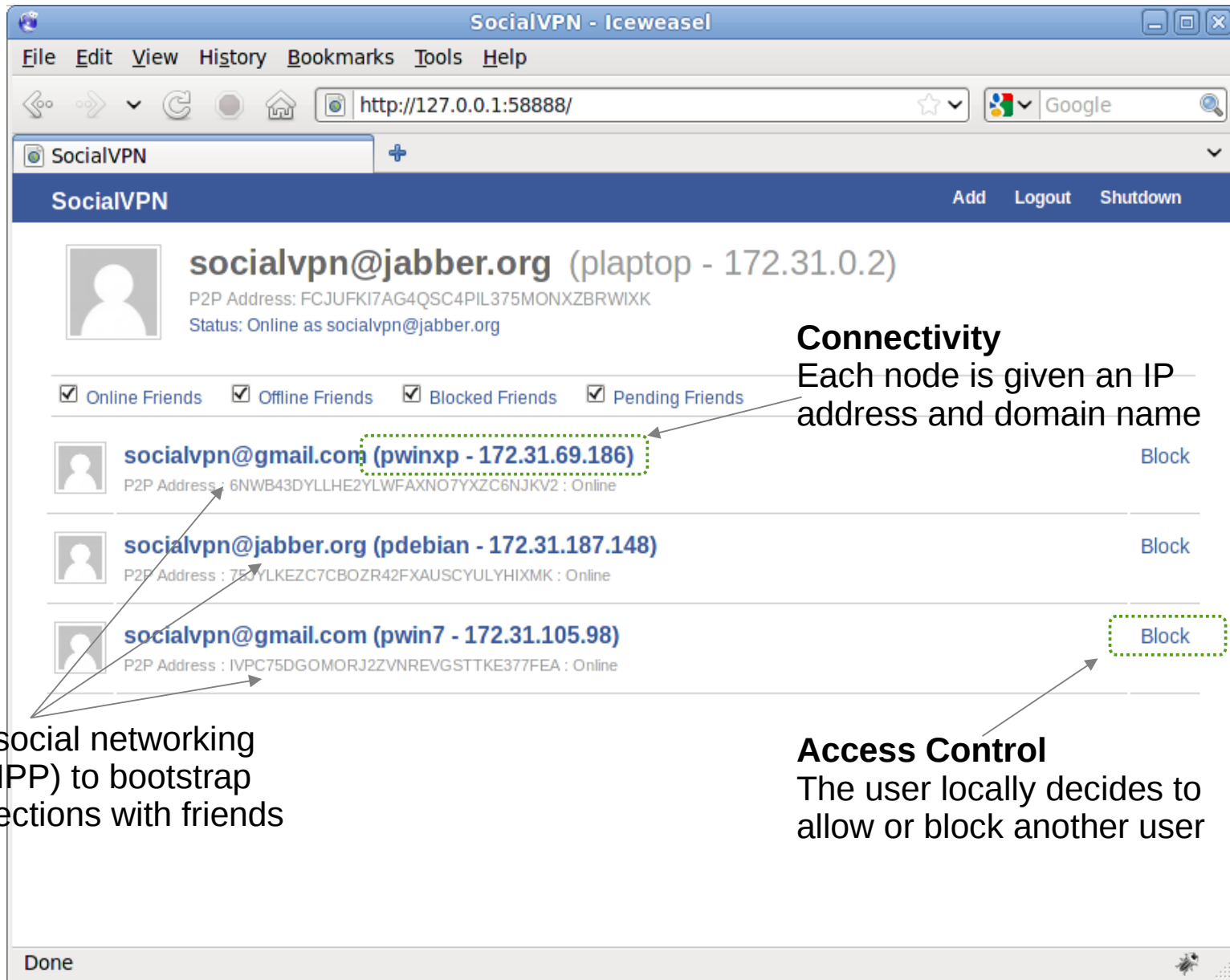
DNS Name	IPv4 Address	P2P Address
localhost	172.25.0.2	node:4AB1
pcid.alice-uid.svpn	172.25.43.89	node:4AB1
pcid.carol-uid.svpn	172.25.1.94	node:B3D5



SocialVPN Design (3)



SocialVPN Design (2)



SocialVPN Design (2)



The code is **open-source** (socialvpn.org) and runs on Windows and Linux; it has been **downloaded over 10,000 times** since September 2009, there are between **80 – 120 users at any given time**

Trust

Use current social networking systems (XMPP) to bootstrap secure connections with friends

Access Control

The user locally decides to allow or block another user

SocialVPN Demo Details

1. Download SocialVPN from socialvpn.org
2. Extract and run SocialVPN
3. Login using Google credentials
4. Ping machine