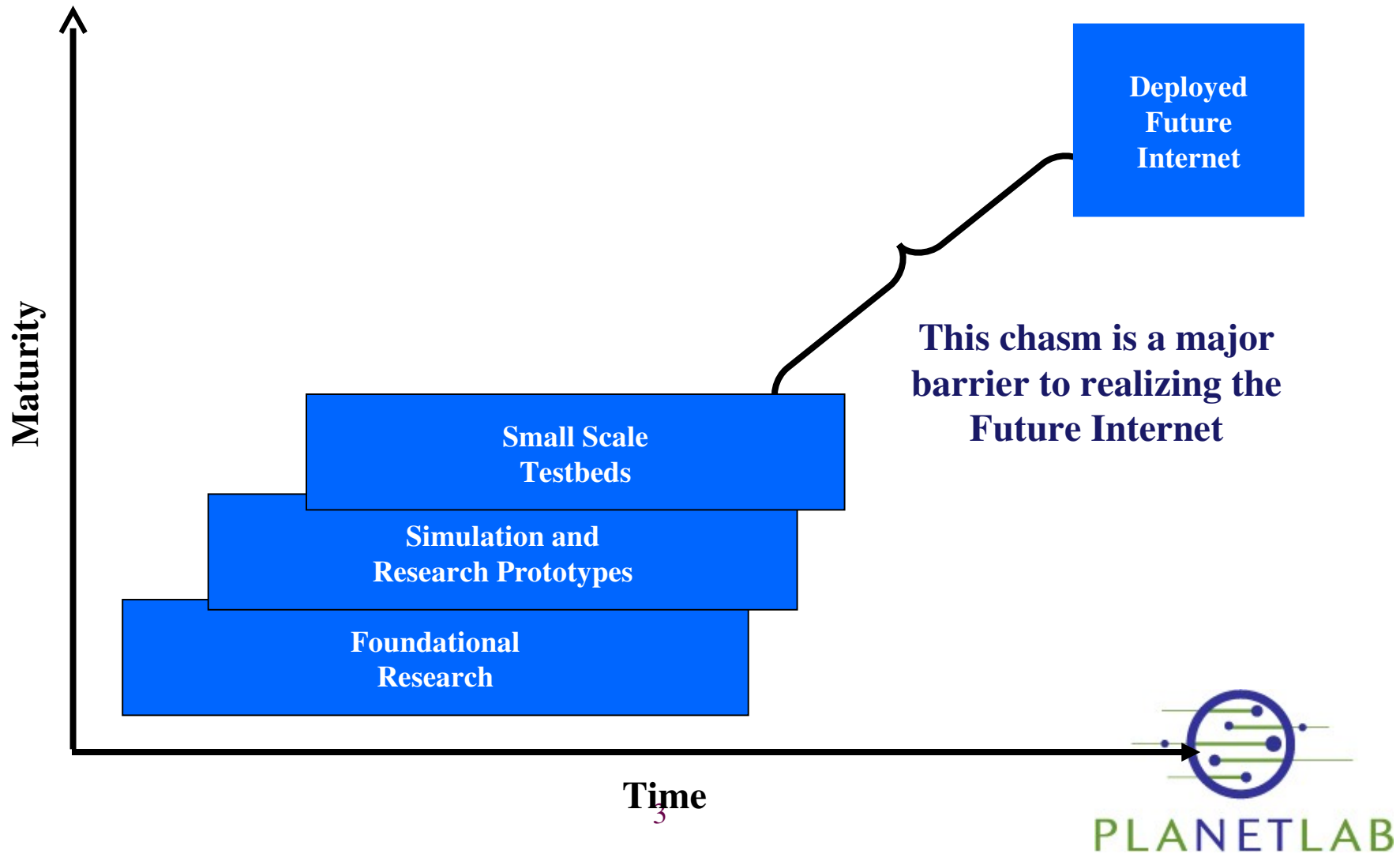

PlanetLab

An open, shared platform for developing, deploying, and accessing planetary-scale applications

In this Lecture

- Introduction to PlanetLab
- PlanetLab Architecture
- PlanetLab at UCY
- PlanetLab Tutorial: A step by step guide

Case for PlanetLab



Overview

- Global distributed system infrastructure
 - platform for long-running services
 - testbed for network experiments
- Launched in March 2002



The value proposition

- Institutions join, provide nodes
 - IA32 architecture servers
 - Hosted outside the firewall
 - Provide power & bandwidth
- In exchange, researchers get to use a small “slice” of many machines worldwide

PlanetLab History

- **March 2002**: Larry Peterson (Princeton) and David Culler (UC Berkeley and Intel Research) organize an "underground" meeting of researchers interested in planetary-scale network services, and propose PlanetLab as a community testbed
- **September 2003**: PlanetLab passes the 200 node mark
- **December 2003**: PlanetLab passes the 300 node mark
- **July 2004**: PlanetLab passes the 400 node mark.
- **December 2004**: PlanetLab passes the 500 node mark
- **September 2005**: PlanetLab passes the 600 node mark
- **September 2006**: PlanetLab passes the 700 node mark
- **June 2007**: PlanetLab passes the 800 node mark



About PlanetLab

- 815 nodes around the world
 - 35 countries
 - 405 sites (universities, research labs)
- A collection of machines distributed over the globe
 - Most of the machines are hosted by research institutions
 - All of the machines are connected to the Internet
- Software Package
 - All PlanetLab machines run a common software package, called MyPLC
 - The software is based on Fedora Core 4 (or Fedora Core 2)

The PlanetLab Consortium (1)

- Overseeing the long-term growth of PlanetLab's hardware infrastructure; designing and evolving its software architecture; providing day-to-day operational support; and defining policies that govern appropriate use.
- Managed by
 - U. Washington, U.C. Berkeley, U. Cambridge, Princeton U.
 - Based in Princeton, NJ, USA.
- Funded by Industry and Govts.
 - NSF, EU, Cernet, etc.
 - Intel, HP, Google, AT&T, FranceTelecom, etc.

The PlanetLab Consortium (2)

- Node resources provided by member institutions
- Small “support” team NOC in Princeton
 - Additional NOCs planned in Europe (Paris), China (Tsinghua)
- Steering Committee
 - University representatives
 - Top-level industrial sponsors

The PlanetLab Consortium (3)

Membership

- **Charter (\$300k annual dues)**
 - Permanent seat on Steering Committee.
 - Unlimited number of slices.
 - Access to PlanetLab events, research papers, and working groups.
- **Full (\$75k annual dues)**
 - Rotating seat on Steering Committee.
 - 10 slices.
 - Access to PlanetLab events, research papers, and working groups.
- **Associate (\$25k annual dues)**
 - 2 slices.
 - Access to PlanetLab events, research papers, and working groups.
- **Sponsor (\$10k annual dues)**
 - Access to PlanetLab events and research papers.
- **Academic (no annual dues)**
 - Seat on Steering Committee by invitation.
 - 10 slices.
 - Access to PlanetLab events, research papers, and working groups.



What is it used for?

- PlanetLab addresses the related problems of:
 - Deploying widely-distributed services
 - Evaluating competing approaches in a realistic setting
 - Evolving the network architecture to better support such services
- So far, PlanetLab is highly successful at doing this.

PlanetLab is not...

- A distributed supercomputer
- A simulation platform
- An Internet emulator
- An arena for repeatable experiments
- Completely representative of the current Internet
- Grid

PlanetLab is not the Grid

- The Grid aims at location-transparency for large computations
 - “I don’t care where this protein-folding job runs as long as it’s done by Monday”
- PlanetLab is all about small, long-running services in specific locations
 - “I need to run a new secure file cache for the next 6 months in Seoul, Sydney, and Vancouver”

PlanetLab is not the Grid

- The Grid is about standardizing one particular paradigm for large scale utility computing.
- PlanetLab provides a low-level platform over which many distributed computing paradigms can be tried.
 - You could build the Grid over PlanetLab's abstractions if you really wanted

PlanetLab is not the Grid

- The Grid starts from scratch in putting together an execution environment for remote computation (e.g. OGSA)
- PlanetLab starts with a well-known, simple interface and encourages the community to evolve multiple, competing execution environments

PlanetLab is...

- An opportunity to qualitatively validate distributed systems in a real deployment environment
- An opportunity to gain valuable experience about what works and what doesn't in the wide area at scale

Terminology

- **Site** A site is a physical location where PlanetLab nodes are located (e.g. Princeton University or HP Labs).
- **Node** A node is a dedicated server that runs components of PlanetLab services.
- **Slice** A slice is a set of allocated resources distributed across PlanetLab. To most users, a slice means UNIX shell access to a number of PlanetLab nodes. Slices have a finite lifetime and must be periodically renewed to remain valid.
- **Principal Investigator (PI)** The PIs at each site are responsible for managing slices and users at each site. PIs are legally responsible for the behavior of the slices that they create. Most sites have only one PI (typically a faculty member at an educational institution or a project manager at a commercial institution).
- **User** A user is anyone who develops and deploys applications on PlanetLab. PIs may also be users.

PI's Roles and Responsibilities

- **Oversight.** PIs are responsible for overseeing all slices that they create on behalf of the users at their site.
- **Account management.** PIs can:
 - Enable, disable, and delete user accounts.
 - Create slices.
 - Delete slices.
 - Assign users to slices.
 - Allocate resources to slices.
- **Node management.** PIs are responsible for the physical maintenance of the nodes at their site



PlanetLab Architecture

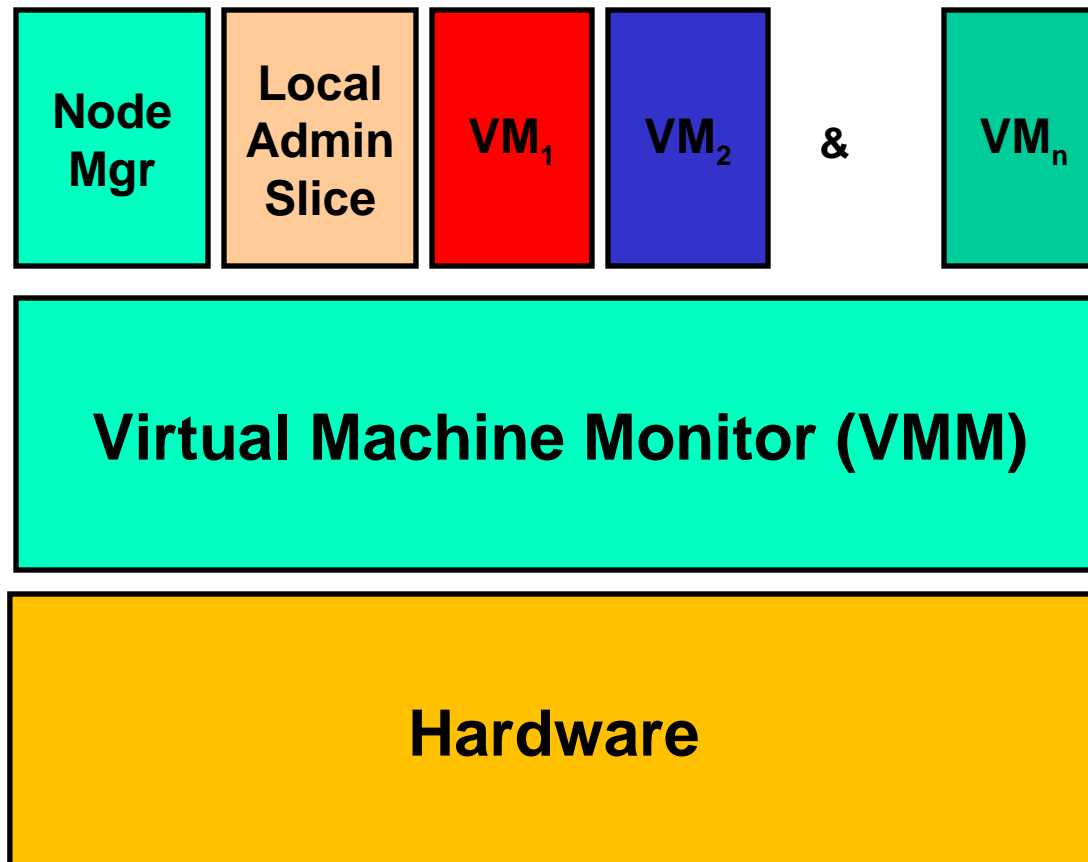
Short-term Requirements (March 2002)

- To support current research work in distributed & P2P systems and networking:
 - Shared by many simultaneous users
 - Isolation and protection
 - Use familiar API (Linux)
 - Networking flexibility
 - Manageable
 - Must be fully operational in 3 months!

Long-term Requirements

- To change the world by incubating the next Internet:
 - Must evolve over time
 - Allow parallel approaches to coexist
 - Produce a viable replacement for the existing network and services

Node Architecture

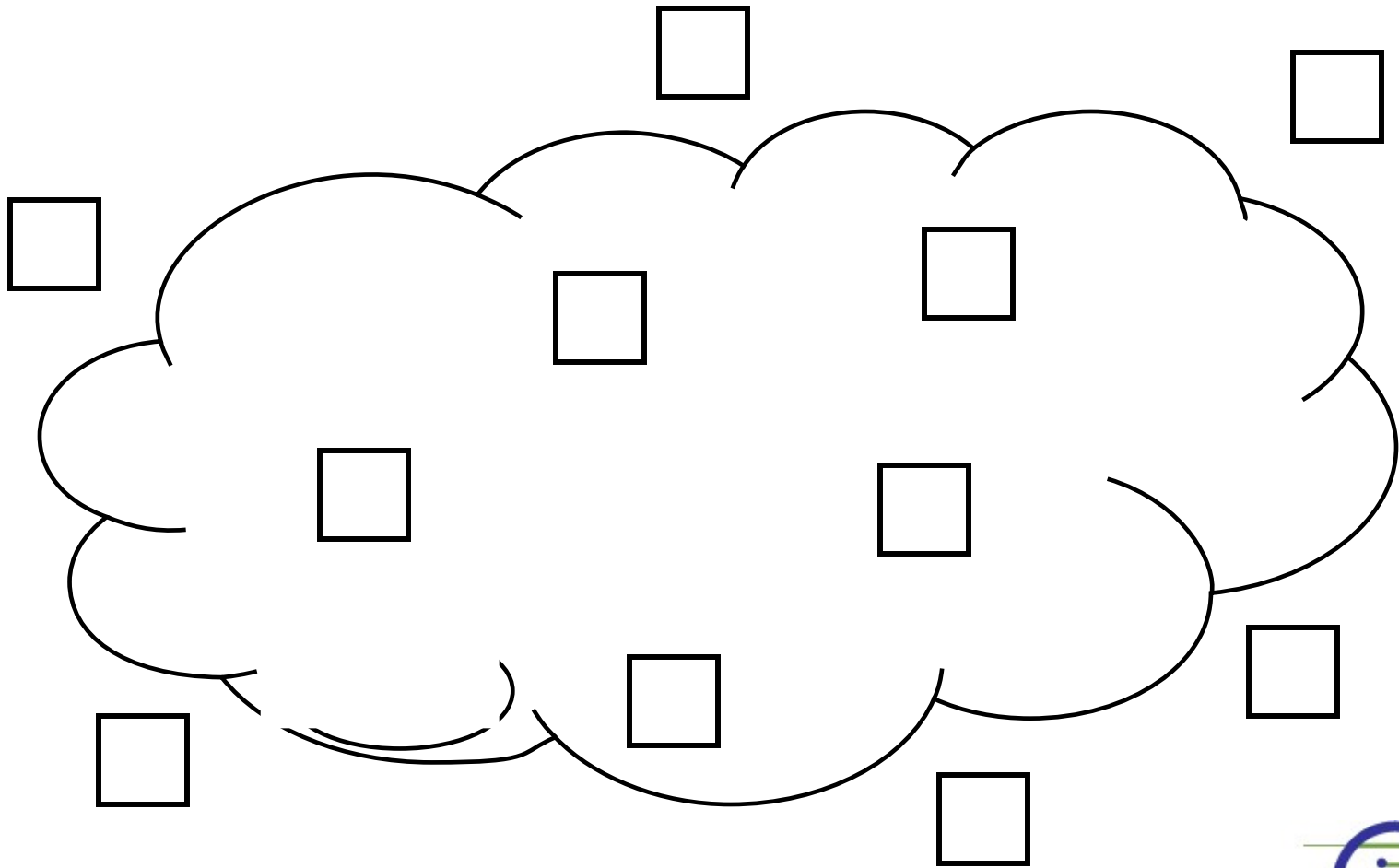


Distributed Virtualization

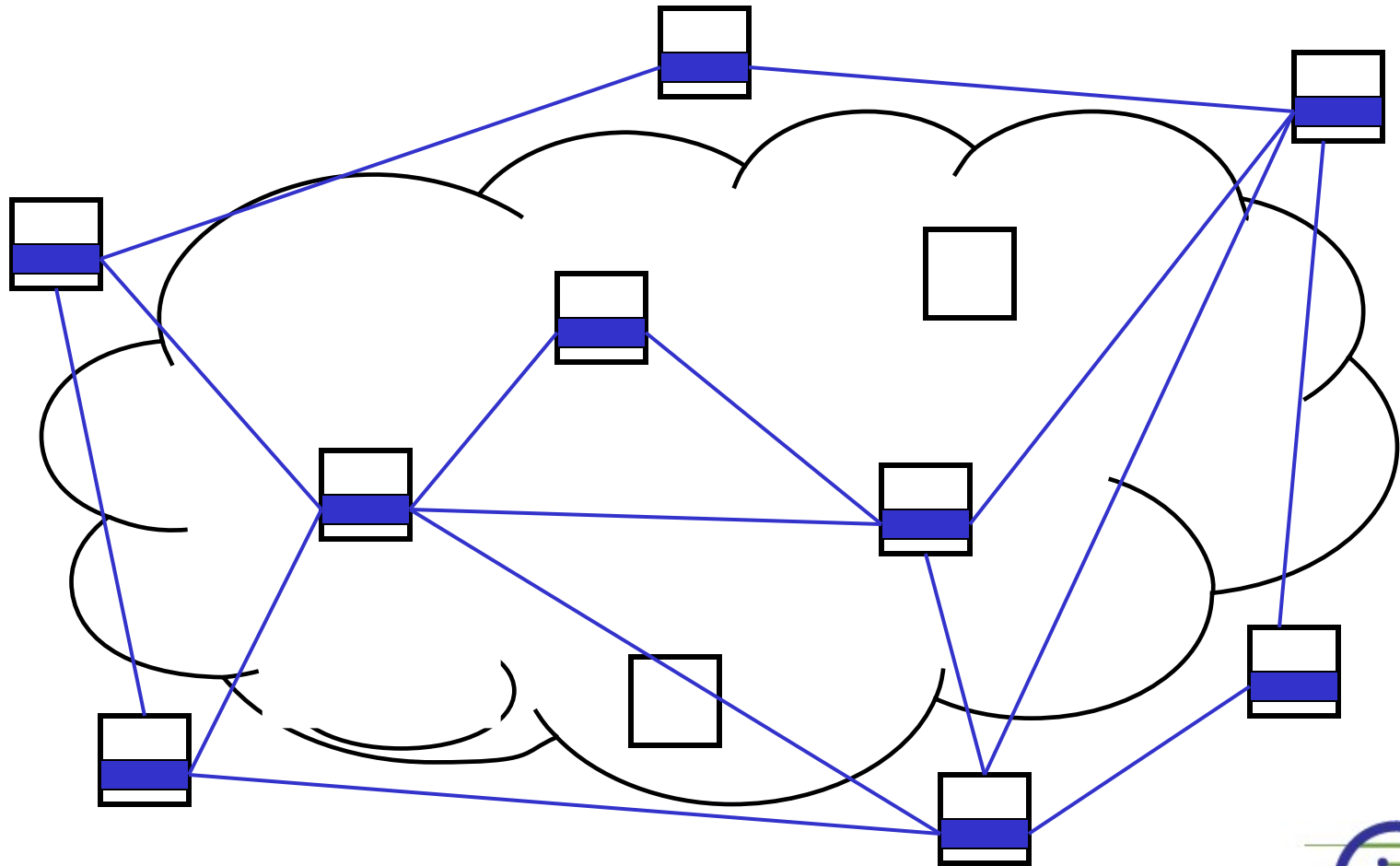
“Allocating a widely distributed set of virtual machines to a user or application”

- Slices
 - Basic unit of isolation and sharing
 - Distributed set of virtual machines (slivers)
 - Services & applications run in slices
- Nodes
 - Physical machines, grouped into Sites.
 - One node hosts many VMs
- Infrastructure Services
 - Provide functionality to developers or other services rather than users

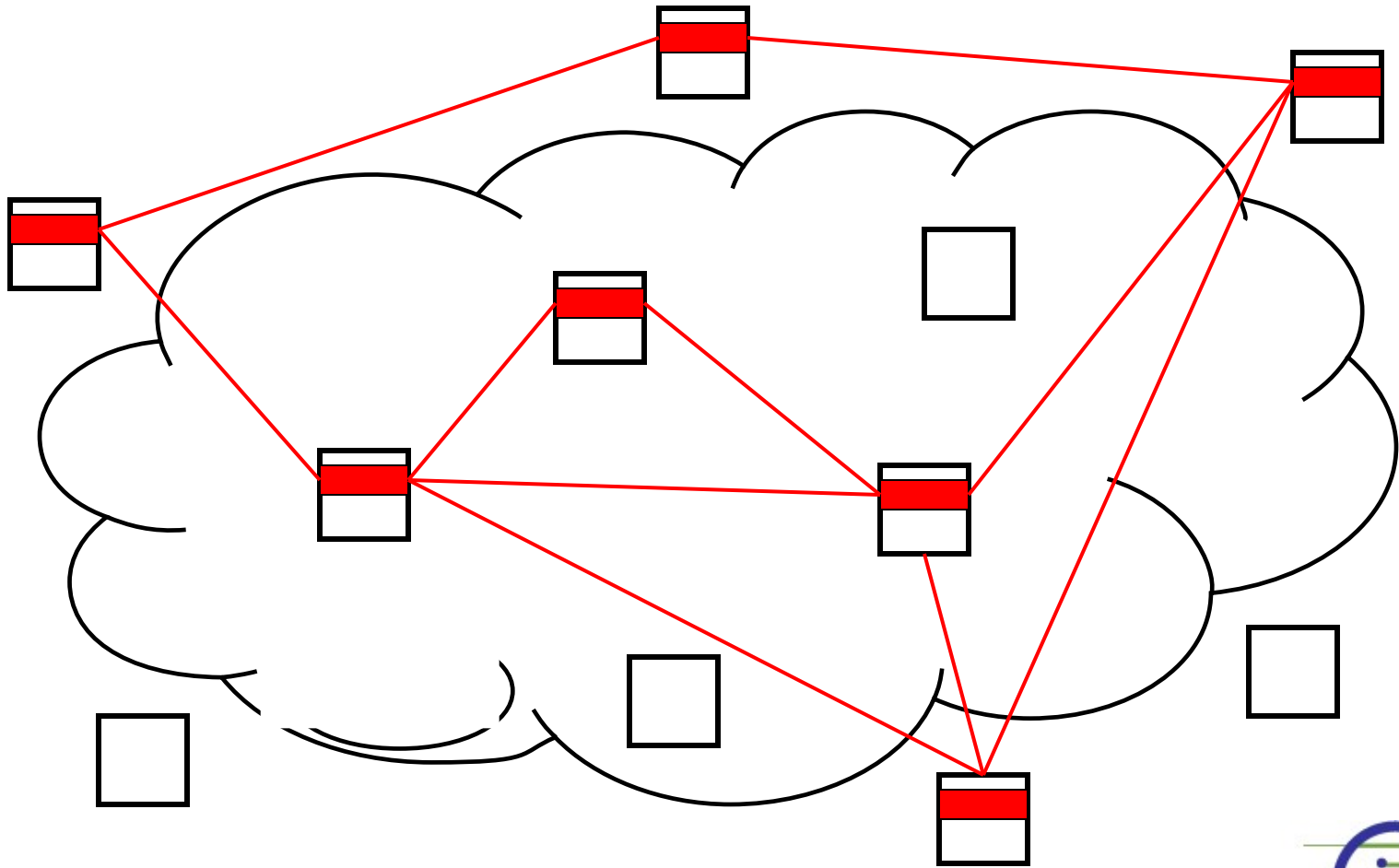
Slices



Slices



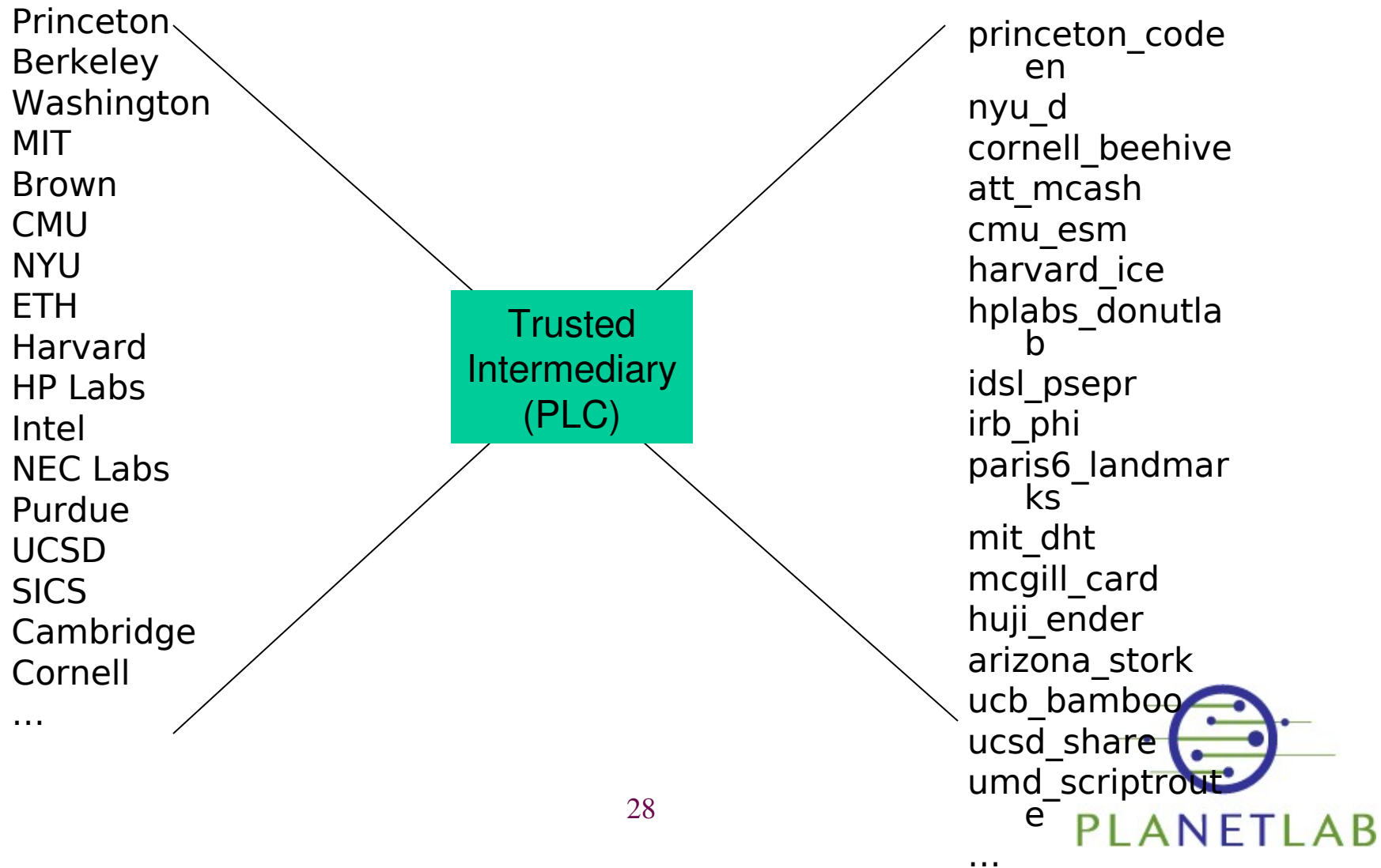
Slices



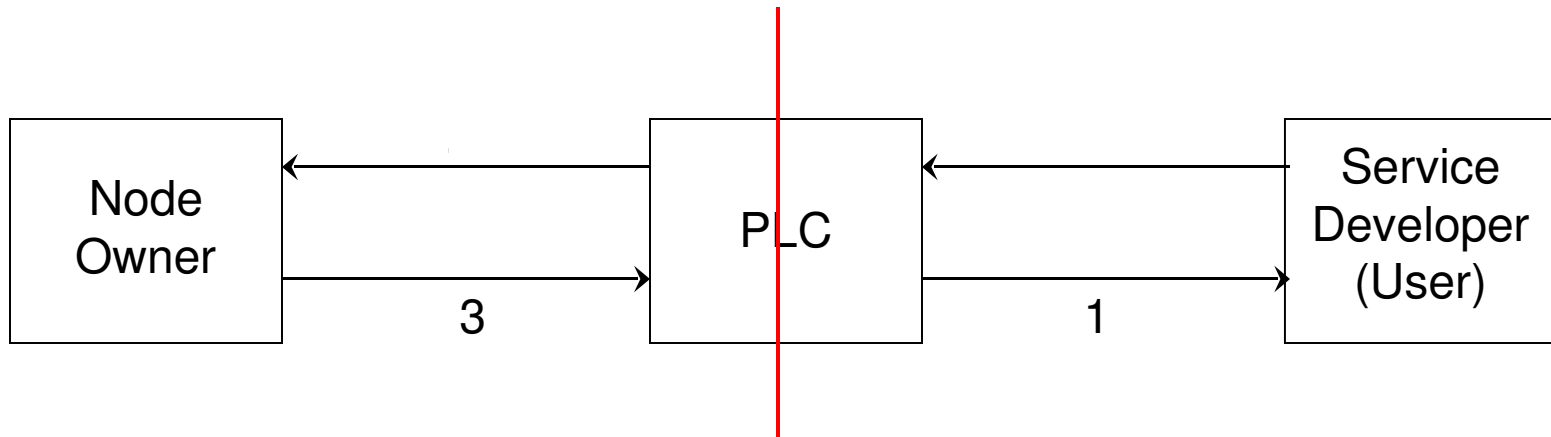
Architecture

- Node Operating System
 - isolate slices
 - audit behavior
- PlanetLab Central (PLC)
 - remotely manage nodes
 - bootstrap service to instantiate and control slices
- Third-party Infrastructure Services
 - monitor slice/node health
 - discover available resources
 - create and configure a slice
 - resource allocation

Trust Relationships (1)

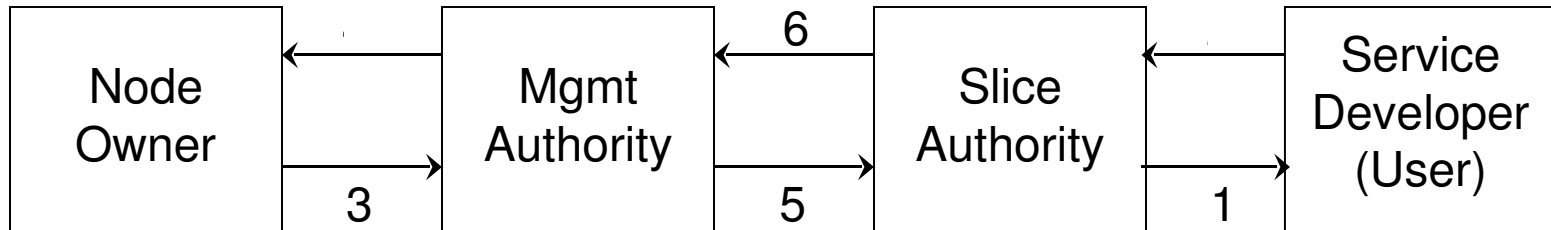


Trust Relationships (2)



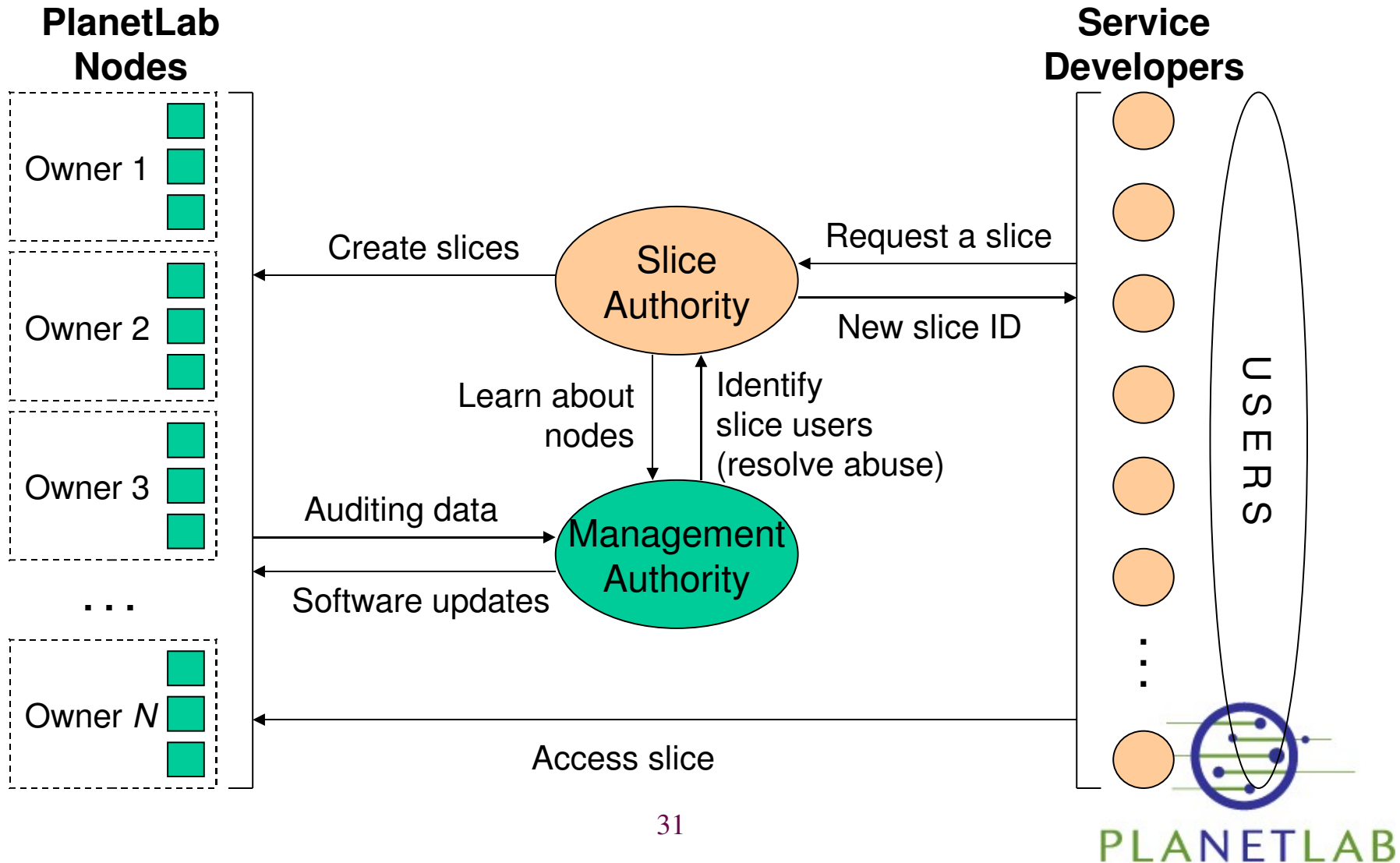
- 1) PLC expresses trust in a user by issuing it credentials to access a slice
- 2) Users trust to create slices on their behalf and inspect credentials
- 3) Owner trusts PLC to set users and map network activity to right user
- 4) PLC trusts owner to keep nodes physically secure

Trust Relationships (3)

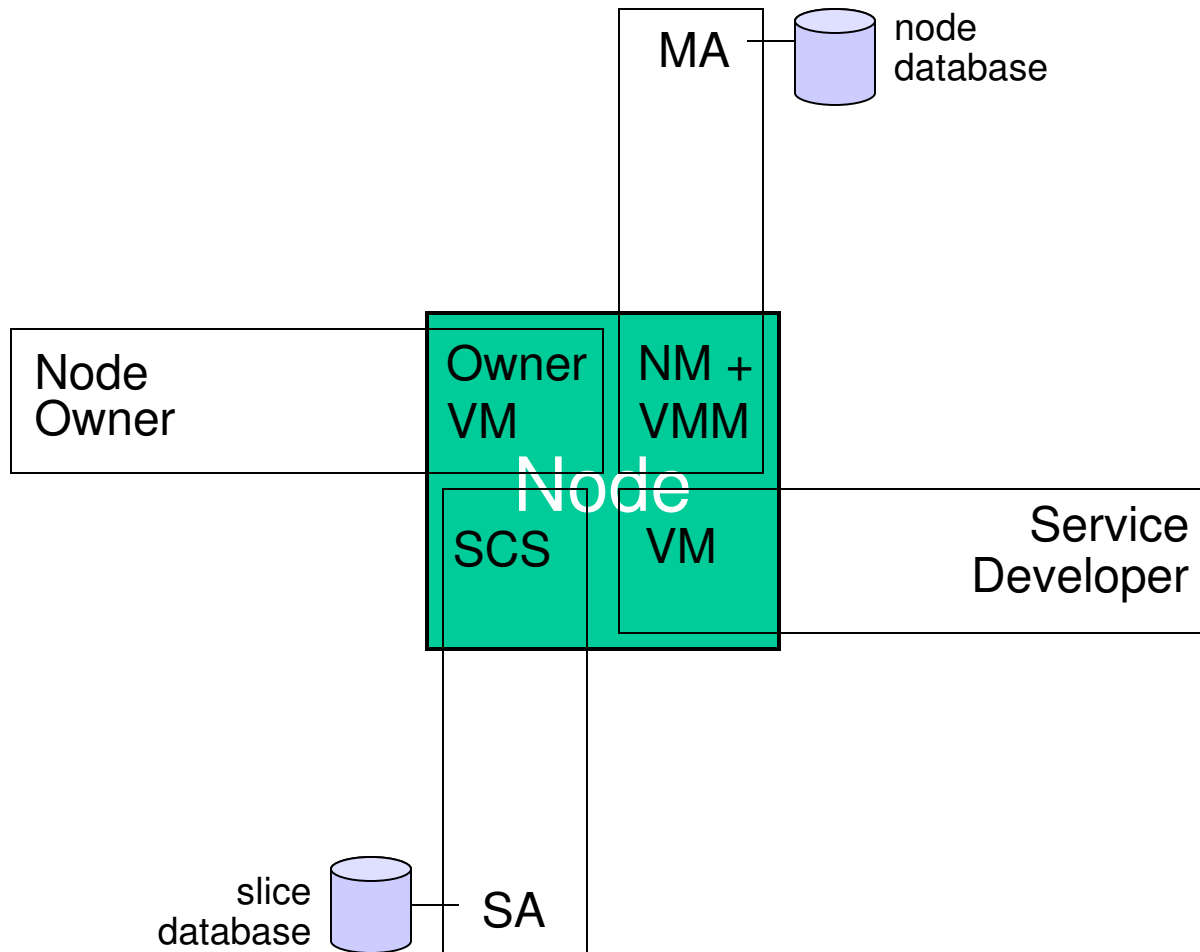


- 1) PLC expresses trust in a user by issuing credentials to access a slice
- 2) Users trust to create slices on their behalf and inspect credentials
- 3) Owner trusts PLC to set users and map network activity to right user
- 4) PLC trusts owner to keep nodes physically secure
- 5) MA trusts SA to reliably map slices to users
- 6) SA trusts MA to provide working VMs

Trust Relationships (4)



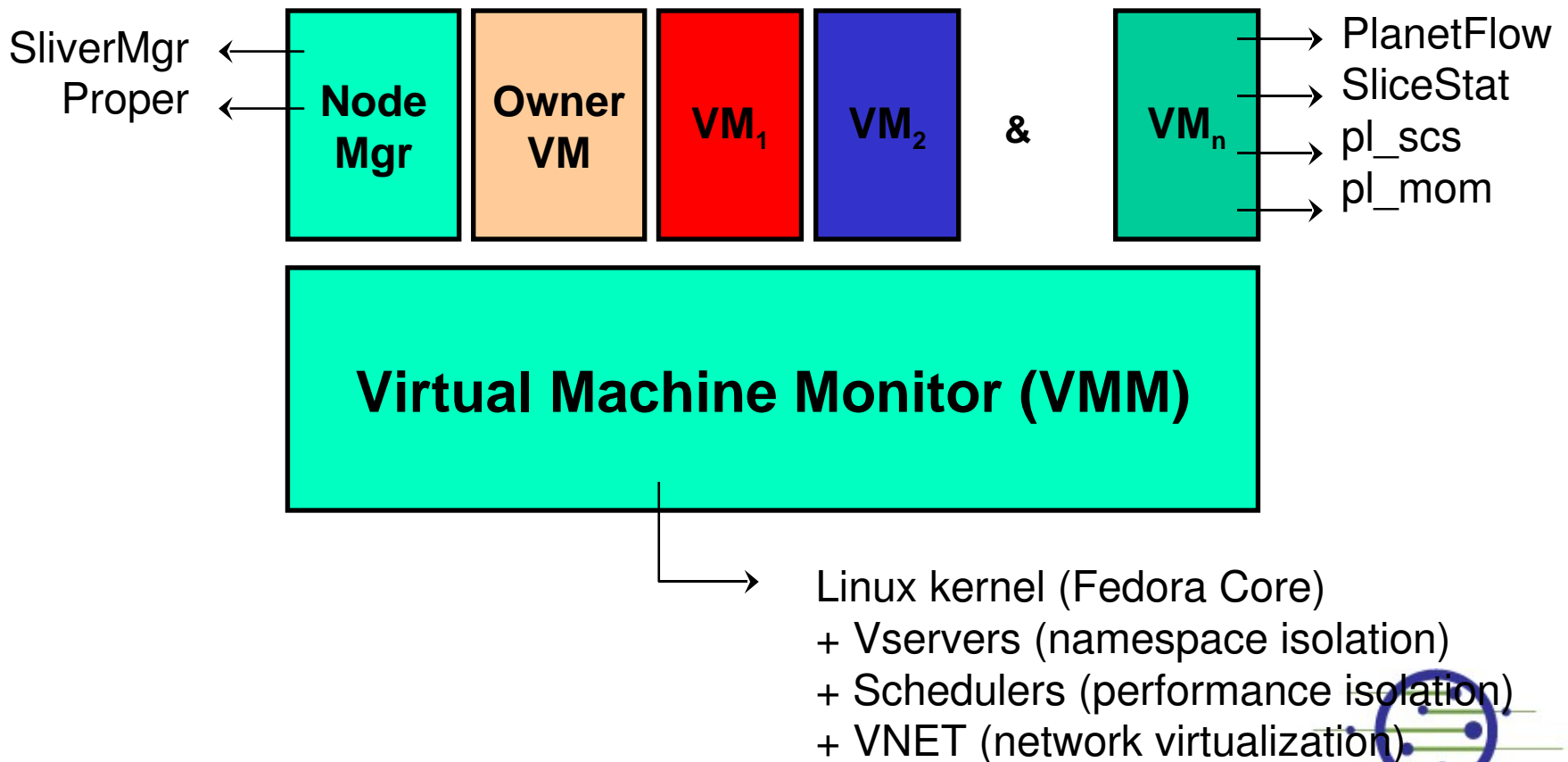
Architecture (3)



Principals

- Node Owners
 - host one or more nodes (retain ultimate control)
 - selects an MA and approves of one or more SAs
- Service Providers (Developers)
 - implements and deploys network services
 - responsible for the service's behavior
- Management Authority (MA)
 - installs and maintains software on nodes
 - creates VMs and monitors their behavior
- Slice Authority (SA)
 - registers service providers
 - creates slices and binds them to responsible provider

Per-Node Mechanisms



VMM (1)

- Linux
 - significant mind-share
- Vserver
 - scales to hundreds of VMs per node (12MB each)
- Scheduling
 - CPU
 - fair share per slice (guarantees possible)
 - link bandwidth
 - fair share per slice
 - average rate limit: 1.5Mbps (24-hour bucket size)
 - peak rate limit: set by each site (100Mbps default)
 - disk
 - 5GB quota per slice (limit run-away log files)
 - memory
 - no limit
 - pl_mom resets biggest user at 90% utilization

VMM (2)

- VNET
 - relies on Linux's Netfilter system
 - slices should be able to send only...
 - well-formed IP packets
 - to non-blacklisted hosts
 - slices should be able to receive only...
 - packets related to connections that they initiated (e.g., replies)
 - packets destined for bound ports (e.g., server requests)
 - Supports the following protocols:
 - TCP
 - UDP
 - ICMP
 - GRE and PPTP
 - also supports *virtual devices*
 - standard PF_PACKET behavior
 - used to connect to a “virtual ISP”

Node Manager

- SliverMgr
 - creates VM and sets resource allocations
 - interacts with...
 - bootstrap slice creation service (pl_scs)
 - third-party slice creation & brokerage services (using tickets)
- Proper: PRivileged OPERations
 - grants unprivileged slices access to privileged info
 - effectively “pokes holes” in the namespace isolation
 - examples
 - files: open, get/set flags
 - directories: mount/unmount
 - sockets: create/bind
 - processes: fork/wait/kill

Auditing & Monitoring

- PlanetFlow
 - logs every outbound IP flow on every node
 - accesses ulogd via Proper
 - retrieves packet headers, timestamps, context ids (batched)
 - used to audit traffic
 - aggregated and archived at PLC
- SliceStat
 - has access to kernel-level / system-wide information
 - accesses /proc via Proper
 - used by global monitoring services
 - used to performance debug services

Long-Running Services (1)

- Content Distribution
 - CoDeeN: Princeton
 - Coral: NYU
 - Cobweb: Cornell
- Internet Measurement
 - ScriptRoute: Washington, Maryland
- Anomaly Detection & Fault Diagnosis
 - PIER: Berkeley, Intel
 - PlanetSeer: Princeton
- DHT
 - Bamboo (OpenDHT): Berkeley, Intel
 - Chord (DHash): MIT

Long-Running Services (2)

- Routing
 - i3: Berkeley
 - Virtual ISP: Princeton
- DNS
 - CoDNS: Princeton
 - CoDoNs: Cornell
- Storage & Large File Transfer
 - LOCI: Tennessee
 - CoBlitz: Princeton
 - Shark: NYU
- Multicast
 - End System Multicast: CMU
 - Tmesh: Michigan

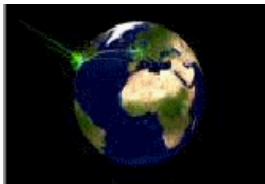
Infrastructure Services

- Brokerage Services
 - Sirius: Georgia
 - Bellagio: UCSD, Harvard, Intel
 - Tycoon: HP
- Environment Services
 - Stork: Arizona
 - AppMgr: MIT
- Monitoring/Discovery Services
 - CoMon: Princeton
 - PsEPR: Intel
 - SWORD: Berkeley
 - IrisLog: Intel

Example: CoDeeN (Princeton)

- Content Distribution Network
 - ~330 (open) caching proxy servers
 - Open to all users (see URL)
- Highly available (after lots of work!)
- Spawned many subprojects / services:
 - CoBlitz, scalable distribution of large files.
 - CoDeploy, efficient synchronization for slices.
 - CoDNS, fast and reliable name lookup.
 - CoMon, node monitoring for PlanetLab
 - CoTest, login debugging tool for nodes
 - PlanetSeer, distributed network anomaly tracing
- Illustrates how deployment of a real service spurs research
 - <http://codeen.cs.princeton.edu/>

PlanetLab AT UCY



PlanetLab AT UCY

October 3, 2007

- ▶ home
- ▶ about us
- ▶ infrastructure
- ▶ documentation
- ▶ contact
- ▶ HPCL



PlanetLab is ...

A Model for Evolving the Next Generation Internet.

PlanetLab is a network of computers strategically located at sites around the world, forming a testbed or platform for creating and deploying planetaryscale services massive applications that span a significant part of the globe. PlanetLab enables users primarily researchers to test and validate new planetaryscale services in an environment that is intended to replicate the environment of the Internet but does not disrupt the Internet's performance. It currently consists of 639 nodes at 305 sites around the world.

An Open Testbed for Developing, Deploying, and Accessing planetary-scale Services.

PlanetLab creates a unique environment in which to conduct experiments at "Internet Scale." The most obvious is that network services deployed on PlanetLab experience all of the behaviors of the real Internet where the only thing predictable is unpredictability (latency, bandwidth, paths taken). A second advantage is that PlanetLab provides a diverse perspective on the Internet in terms of connection properties, network

Infrastructure

- Two SuperMicro Servers (SuperServer 6022P-I) support the global PlanetLab infrastructure. Each SuperMicro has:
 - 2.4 GHz Pentium4
 - 2GByte Maximum RAM
 - 80+160 GByte total disk space
 - Fast Ethernet interface

PlanetLab Tutorial

A step-by-step guide

First steps to using PlanetLab

1. Register as a user
2. Create an ssh key
3. Create a slice
4. Add nodes to the slice
5. Describe the slice
6. Log into a sliver

Step 1: Register

Go to <http://www.planet-lab.org/>:

PlanetLab

- About
 - Consortium
 - Federation
 - History
 - Sites
 - Projects
- Status
- Support
 - Site Assistant
- Documentation
 - API
 - AUP
 - Bibliography
 - FAQ
 - Tutorial
 - PDNs
 - Guides
- Community
 - Courseware
 - Mailing Lists
 - Meetings
 - Presentations
- Software
 - CVS
 - Roadmap
 - Services
 - User Tools

Search

Search

Home

Account Registration

Your E-mail address must be able to receive e-mail and will be used as your PlanetLab username.

Do not select the **Principal Investigator** or **Technical Contact** roles unless you have spoken with the current PI of your site, and you intend to assume either or both of these roles.

First name: *

Last name: *

Title:

E-mail: *

Password: *

Telephone:

Site: *
University of Cyprus

Additional Roles:
 Technical Contact

Register

PlanetLab login

E-mail: *

Password: *

Log in

Forgot your password?

Create an account

Announcements

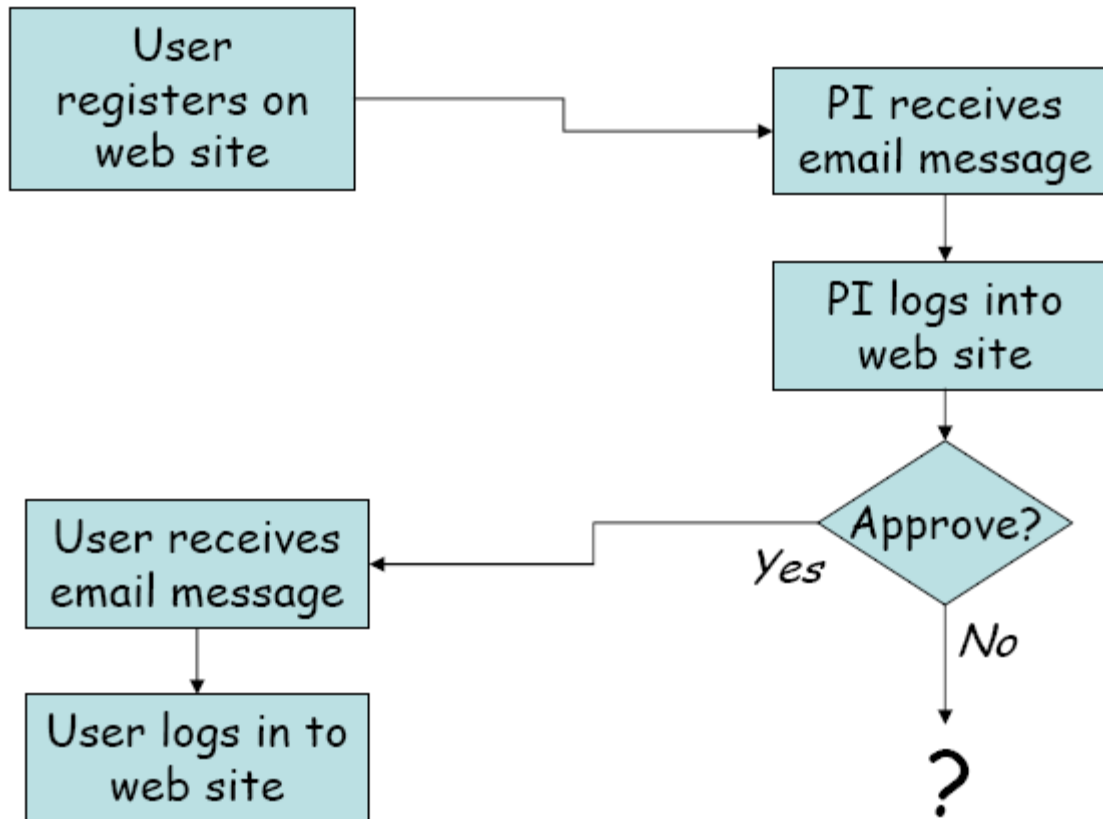
- Site Assistant
PlanetLab Operations is developing a new tool, called
- ROADS Workshops
A series of international workshops on Real Overlays and Distributed Systems (
- PlanetLab Upgrade
We have recently upgraded PlanetLab to Version 4.0. Part of the upgrade involved revamping this webs...
- PlanetLab Services
Many user services run continuously on PlanetLab, generating the majority of our traffic and facilit...
- Sirius Upgrade
The Sirius calendar service has been upgraded to allow users to acquire guaranteed resources across ...

Copyright © 2007 The Trustees of Princeton University

Filling out the Form

- Fill in contact details
- Select your site
 - E.g. University of Cyprus
- Are you a PI?
 - PIs are responsible for PlanetLab site: participation, approve users, create slices
 - Most users are not PIs
- Are you a tech contact?
 - Probably not!
- Submit the form

User Registration



Account approval (as PI)

Accounts at Intel Research Berkeley		
Buonadonna, Philip	buonadonna@acm.org	user
Chang, James	jameschn@uclink.berkeley.edu	user
Chun (AdminAPI Account), Brent	devnull@intel-research.net	user
Chun (Non-admin Account), Brent	bnc@theether.org	user
Chun, Brent	bnc@intel-research.net	admin user tech
Fall, Kevin	kfall@cs.berkeley.edu	user
Fulton, Bryan	bfulton@eecs.berkeley.edu	user

Account approval (as PI)

PlanetLab: Enable Account - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://www.planet-lab.org/db/accounts/setenable.php?

 **PLANETLAB**
An open platform for developing, deploying, and accessing planetary-scale services

Home User **PI** Admin About Status Contact Us

Sites

- » View All Sites
- » View My Site
- » Update My Site
- » Switch Site

Nodes

- » View All Nodes
- » View Site Nodes
- » Other Node Groups
- » Control Site Nodes
- » Bandwidth Limits

Enable Account

Are you sure you want to enable this account? The user will be able to login and use any slices assigned to them.

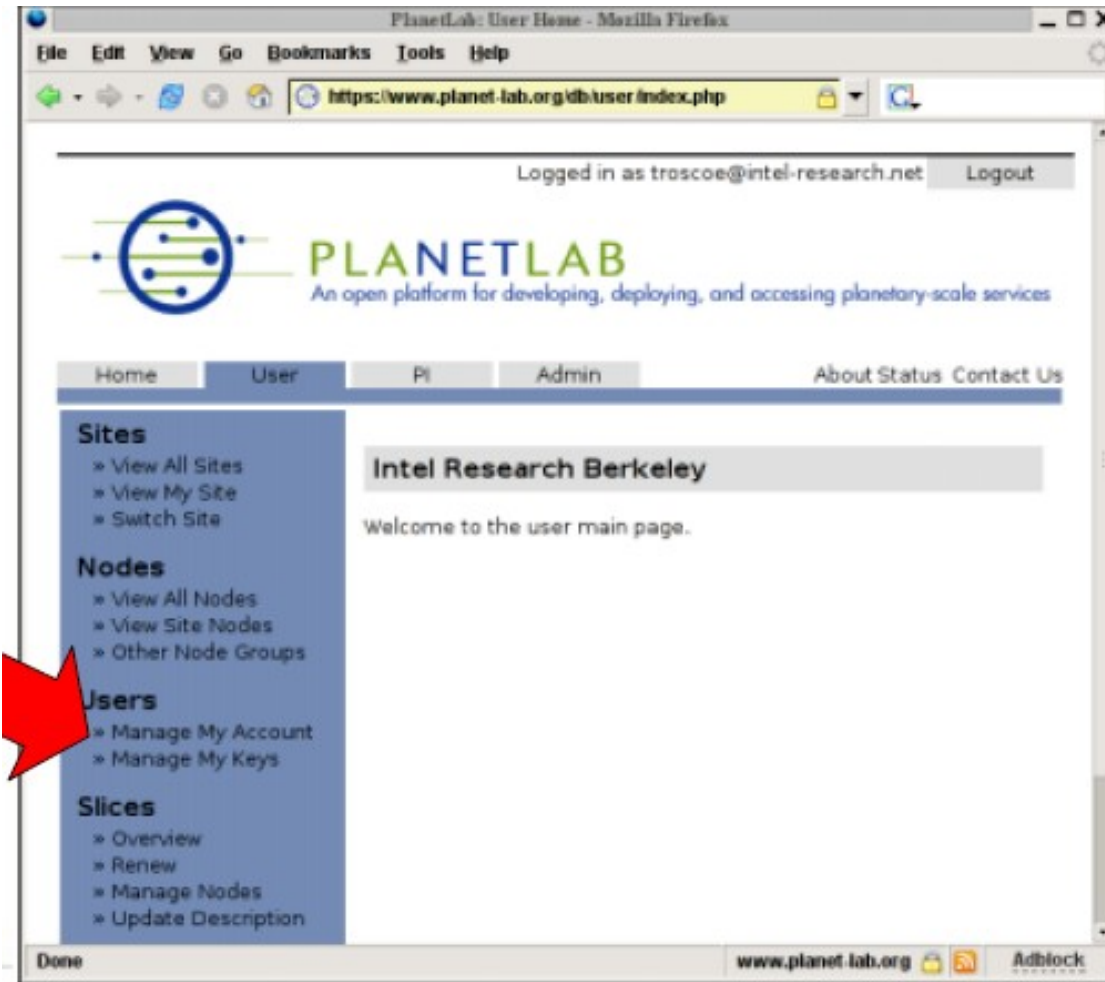
Name: Tristan Koo

Email: tkoo@intel-research.net

Enable Account

Done www.planet-lab.org Adblock

You are now a user!



Step 2: Create an ssh key

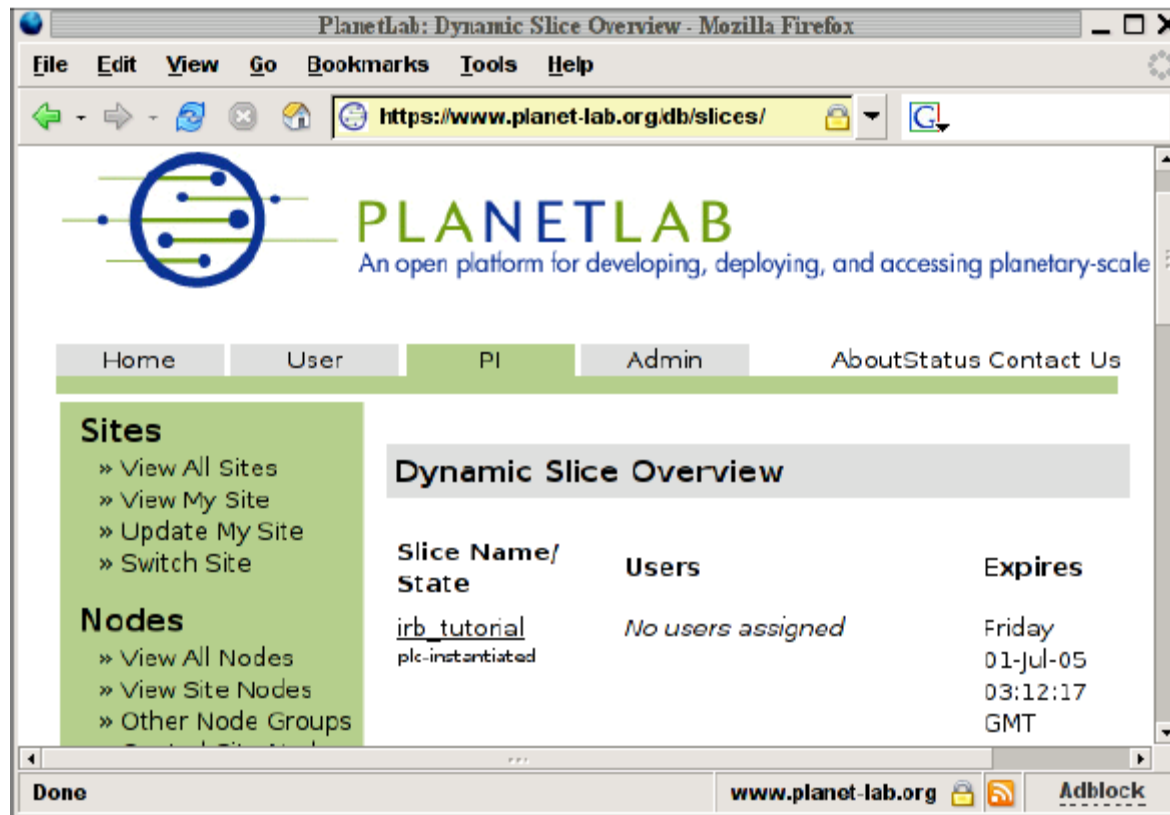
- PlanetLab uses 1024-bit RSA keys for authentication on nodes
- Upload your key to the website using OpenSSH:
 - `ssh-keygen -t rsa -f ~/.ssh/id_planetlab`
- Do use a secure passphrase

Step 3: Creating a slice

Actually, ask your PI to create a slice...

Slice names are of form:
<site>_<local name>


Assign users to a slice (as PI)



PlanetLab: Dynamic Slice Overview - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://www.planet-lab.org/db/slices/

 **PLANETLAB**
An open platform for developing, deploying, and accessing planetary-scale

Home User **PI** Admin AboutStatus Contact Us

Sites

- » View All Sites
- » View My Site
- » Update My Site
- » Switch Site

Nodes

- » View All Nodes
- » View Site Nodes
- » Other Node Groups

Dynamic Slice Overview

Slice Name/ State	Users	Expires
irb_tutorial plc-instantiated	<i>No users assigned</i>	Friday 01-Jul-05 03:12:17 GMT

Done www.planet-lab.org Adblock

By now...

- You have a user account
- Your PI has approved it
- Your PI has created a slice
- Your PI has assigned you as a user of the slice
- Next step: add nodes to the slice

Step 4: Adding nodes



Adding nodes

PlanetLab: Manage Node Assignments - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://www.planet-lab.org/tib/slices/assign_nodes

 **PLANETLAB**
An open platform for developing, deploying, and accessing planetary-scale services

Home **User** PI Admin AboutStatus Contact Us

Sites

- » View All Sites
- » View My Site
- » Switch Site

Nodes

- » View All Nodes
- » View Site Nodes
- » Other Node Groups

Users

- » Manage My Account
- » Manage My Keys

Slices

- » Overview
- » Renew
- » Manage Nodes
- » Update Description

[Back](#) to slice details.

Manage Node Assignments

Show:

Either all nodes from this site are already assigned, or no assignable nodes exist.

All nodes currently assigned to slice irb_phi:

- 140.136.206.232
- 200-102-209-151.paemt7001t.brasitelecom.net.br
- 200-102-209-152.paemt7001t.brasitelecom.net.br
- alice.cs.princeton.edu
- arari.snu.ac.kr
- blast.uwaterloo.ca

Done www.planet-lab.org Adblock

Adding nodes

- Page for adding/removing nodes
 - Select site to see nodes to add
 - Shown are all nodes currently in slice
 - Nodes can be removed
- For your first time, it is recommended:
 - Add a couple of nodes, one nearby
- Note: Changes will take ~5 minutes to propagate!

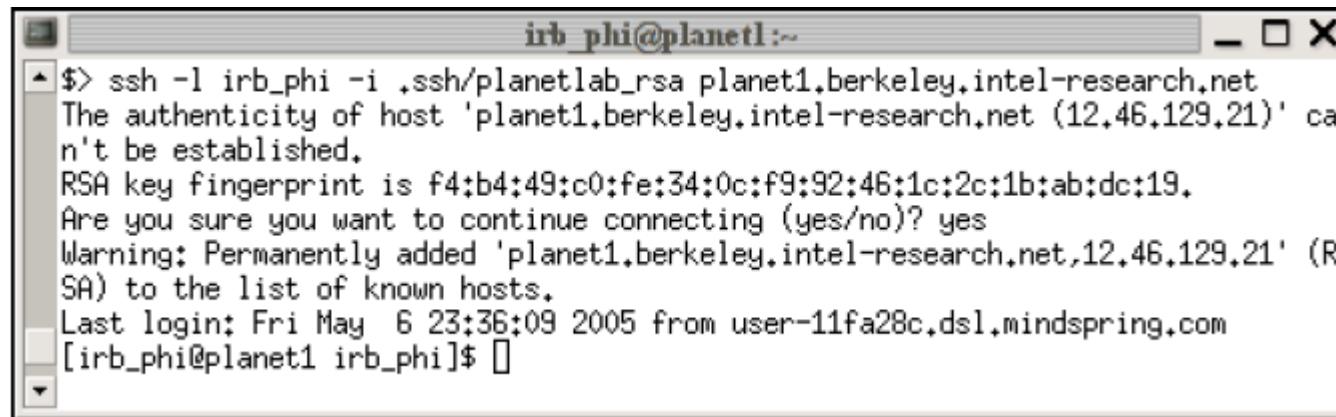
Step 5: Describe your slice



Step 6: Log in to a node

- Your user name is your slice name
- E.g., with OpenSSH:

```
- ssh -l irb_phi \  
  -i ~/.ssh/id_planetlab \  
  planet1.berkeley.intel-research.net
```

A terminal window titled 'irb_phi@planet1:~' showing the execution of an SSH command. The command is 'ssh -l irb_phi -i .ssh/planetlab_rsa planet1.berkeley.intel-research.net'. The output shows a warning about the host's authenticity, the RSA key fingerprint, a confirmation prompt 'Are you sure you want to continue connecting (yes/no)? yes', a warning to permanently add the host to the list of known hosts, and the last login information: 'Last login: Fri May 6 23:36:09 2005 from user-11fa28c.dsl.mindspring.com'. The prompt is now '[irb_phi@planet1 irb_phi]\$' with a cursor.

```
irb_phi@planet1:~  
$> ssh -l irb_phi -i .ssh/planetlab_rsa planet1.berkeley.intel-research.net  
The authenticity of host 'planet1.berkeley.intel-research.net (12.46.129.21)' ca  
n't be established.  
RSA key fingerprint is f4:b4:49:c0:fe:34:0c:f9:92:46:1c:2c:1b:ab:dc:19.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'planet1.berkeley.intel-research.net,12.46.129.21' (R  
SA) to the list of known hosts.  
Last login: Fri May 6 23:36:09 2005 from user-11fa28c.dsl.mindspring.com  
[irb_phi@planet1 irb_phi]$
```

Deploying Software to Your Nodes

- Copy binaries to the node to run
 - scp or rsync works for small numbers of nodes
 - Ensure library dependencies are satisfied

References

- PlanetLab official Web site: <http://www.planet-lab.org/>
- L. Peterson, S. Muir, Timothy Roscoe, and Aaron Klingaman PlanetLab Architecture: An Overview. Technical Report, PlanetLab, May 2006
- L. Peterson and T. Roscoe. The Design Principles of PlanetLab. *Operating Systems Review (OSR)*, 40(1):11.16, Jan. 2006.