Implementing the Emulab-PlanetLab Portal: Experiences and Lessons Learned

Kirk Webb Mike Hibler Robert Ricci
Austin Clements Jay Lepreau

University of Utah

December 5, 2004

1st WORLDS

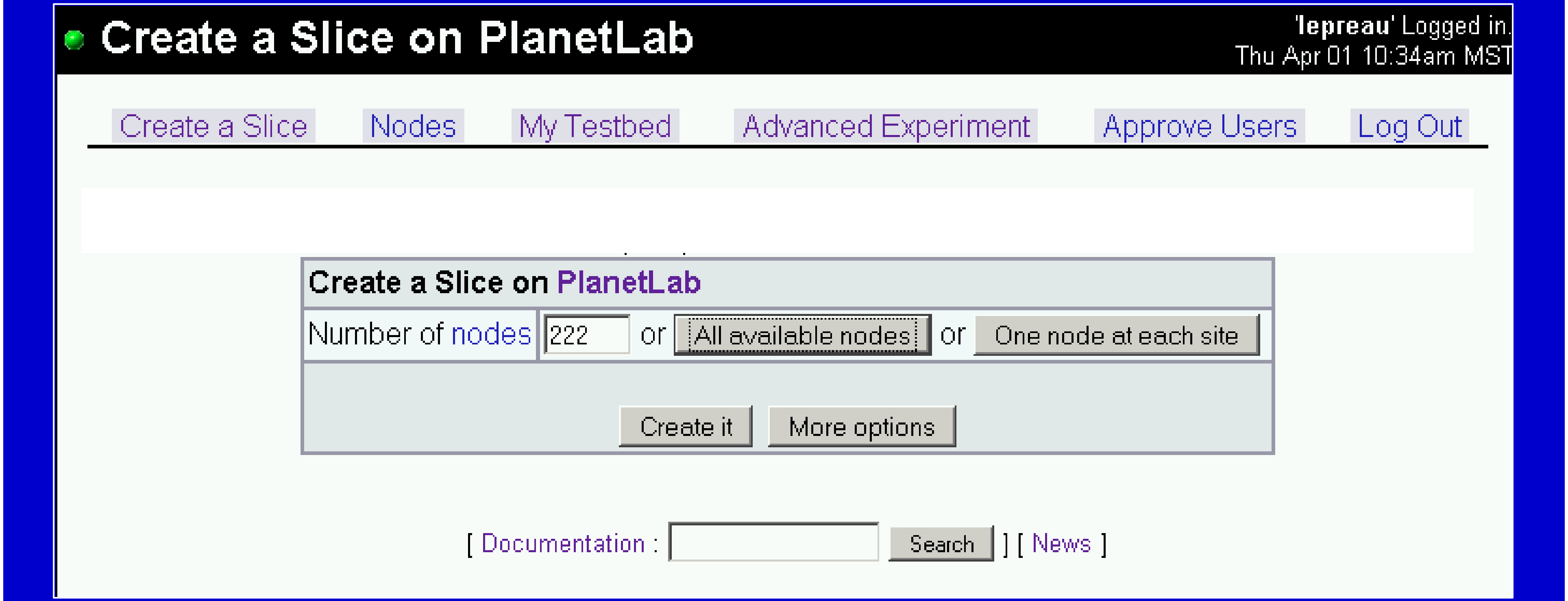
What is Emulab?

- Software to control network testbeds
  - Instantiates user-requested topologies on available resources
  - Most popular UI is fancy Web interface; XML-RPC
- Emulab "Classic"
  - ~200 PCs in a densely connected cluster
  - Dozens of experiments "swap" in and out each day
- Extended to wide area in late 2002
  - RON testbed and Emulab's own wide-area nodes
- Now: a testbed with diverse resources
  - Physical, virtual and simulated nodes and links

Why Create "The Portal"?

- Diversify Emulab with new resources
- Explore challenges of integrating with other testbed environments
- Provide PlanetLab users with a powerful but easy-to-use interface

K.I.S.S.



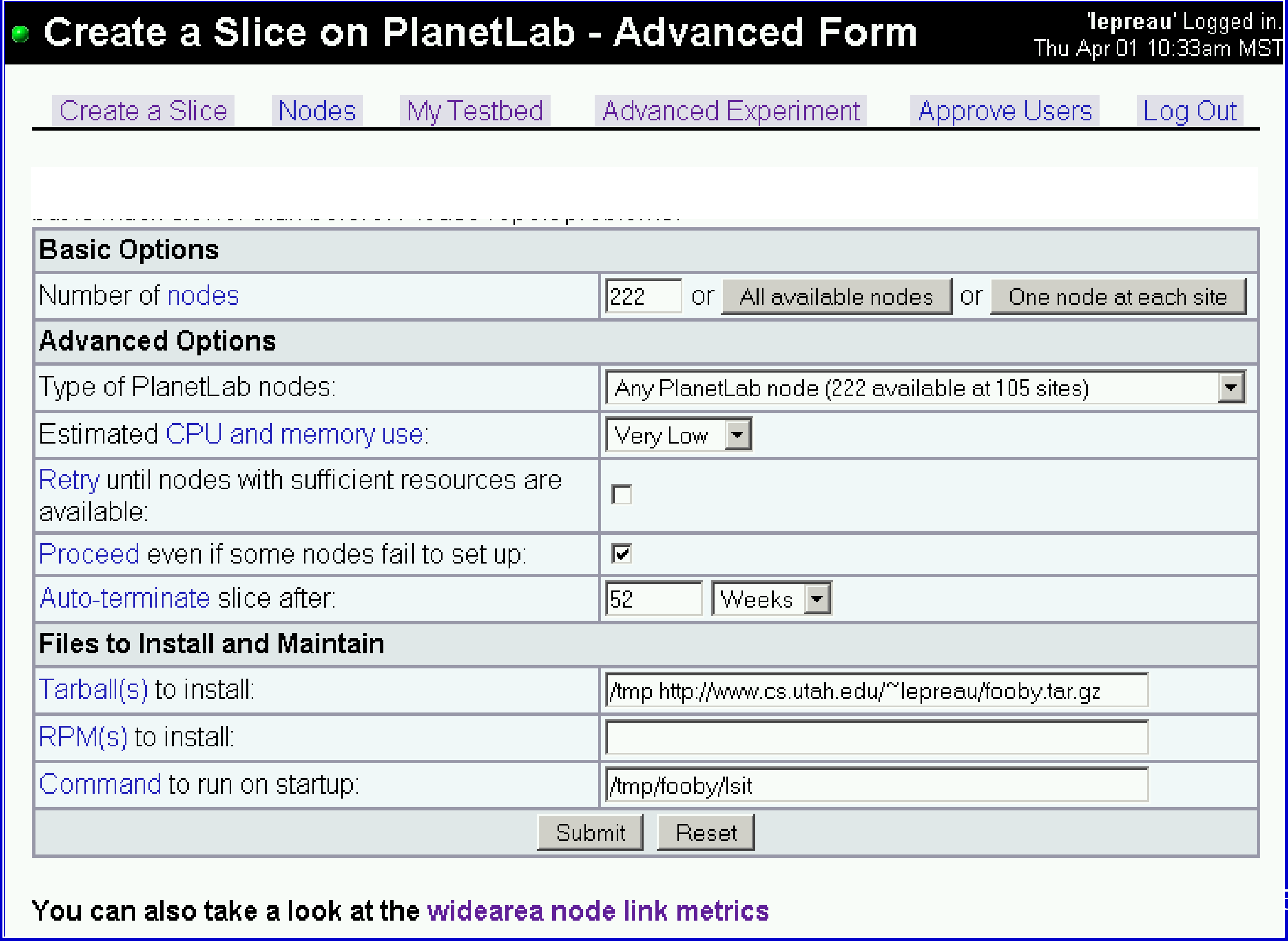
Create a Slice on PlanetLab

'lepreau' Logged in.
Thu Apr 01 10:34am MST

Create a Slice on PlanetLab

Number of nodes: 222

Decisions, Decisions



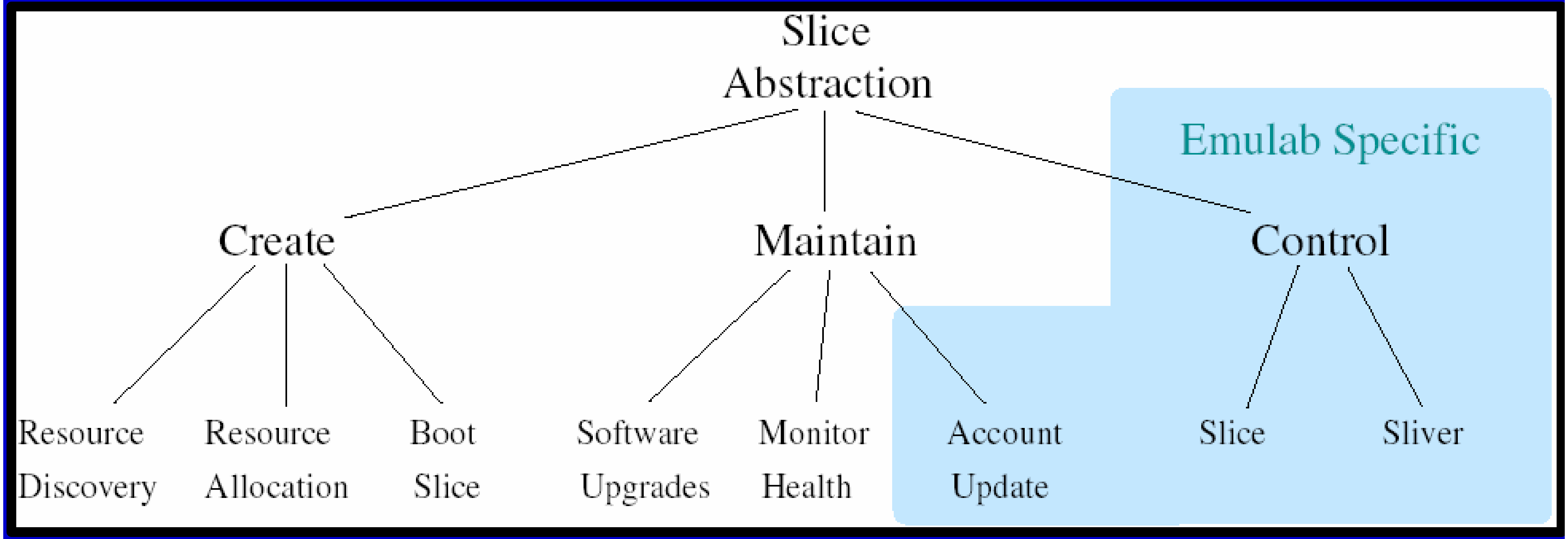
Create a Slice on PlanetLab - Advanced Form

'lepreau' Logged in.
Thu Apr 01 10:33am MST

Basic Options	
Number of nodes	222 or All available nodes or One node at each site
Advanced Options	
Type of PlanetLab nodes:	Any PlanetLab node (222 available at 105 sites)
Estimated CPU and memory use:	Very Low
Retry until nodes with sufficient resources are available:	□
Proceed even if some nodes fail to set up:	☑
Auto-terminate slice after:	52 Weeks
Files to Install and Maintain	
Tarball(s) to install:	/tmp http://www.cs.utah.edu/~lepreau/fooby.tar.gz
RPM(s) to install:	
Command to run on startup:	/tmp/fooby/lsit

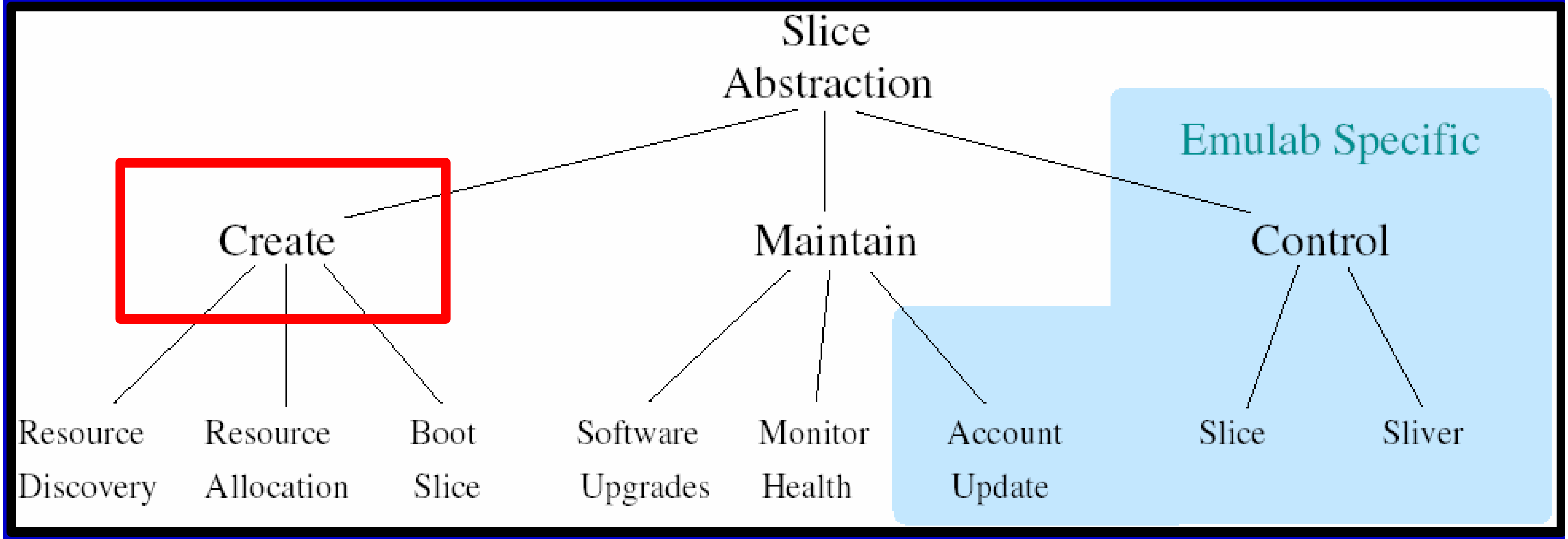
You can also take a look at the widearea node link metrics

Emulab-PlanetLab Portal Features



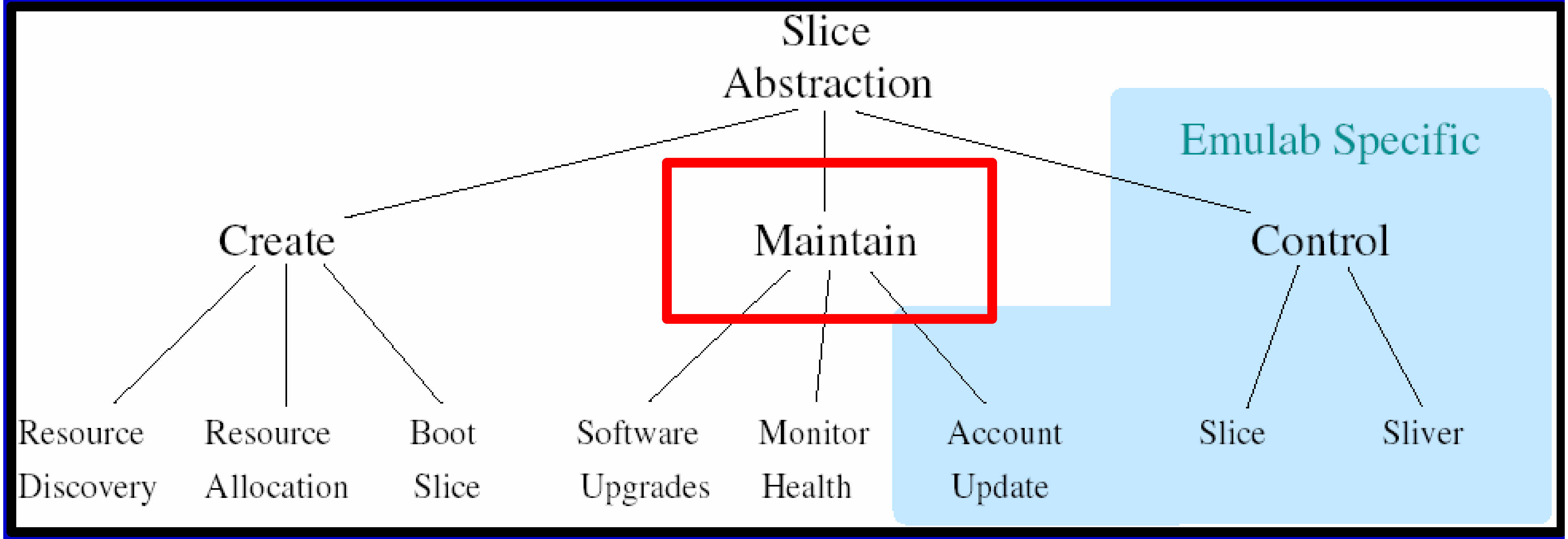
- Emulab provides all elements of the PlanetLab infrastructure service taxonomy, plus more

Portal Features (cont'd)



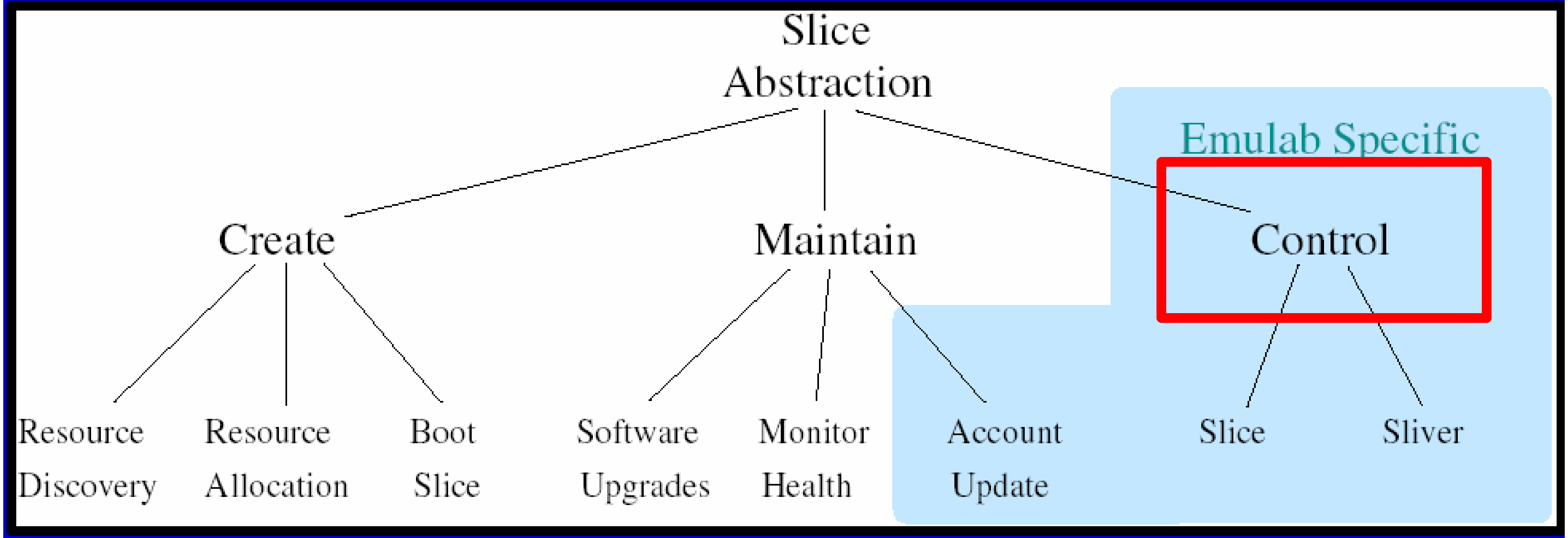
- Monitors sensors to ascertain node characteristics
- Three selection methods: manual, link-, and node-centric

Portal Features (cont'd)



- Watchdog process per virtual node
- Software upgrades and account updates

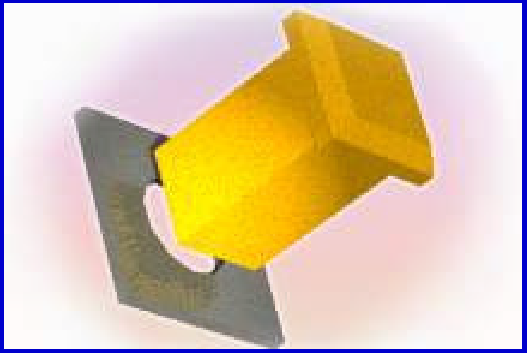
Portal Features (cont'd)



- “Reboot” a single virtual node, or all of them
- Soon: wide-area event system for control

Challenges and Lessons

- Different use models
- State management
- Interface evolution
- Failure



Different Use Models

- Emulab: rapid cycle experiments (mostly)
  PlanetLab: long-running services (mostly)
- Average Emulab experiment duration: five hours
- Building fast/synchronous on delayed/async?

- Delayed, asynchronous interfaces force fast synchronous clients to waste resources
- Exposing lower-level API primitives allows a wider range of service models

State Management

- Locations of data are spread out
- Data coupling issues
  - Identity crisis!
  - Balance between coherency and overhead (age-old problem)

- Persistent & reliable node identifiers are a must
- Should not assume long-term state synchronization

Interface Evolution

- Research infrastructures evolve rapidly
- Tension between PlanetLab goals:

  "Evolving Architecture" → change

  vs.

  "Unbundled Management" → many services,
  many players

- Internally, use the same API that you export
- Embrace the inevitable: changing APIs.
  Make that code modular

Failure

- All node "liveness" metrics are unreliable
  - Trumpet, Ganglia, Emulab Watchdog …
- Anything can fail
  - Disk space, fds, PLC, …

- Only execution of the application itself indicates node "liveness"!

Conclusions

- Hard to keep it working
- Will people build large systems on other parties' constantly-changing research systems?