# P2P technologies, PlanetLab, and their relevance to Grid work

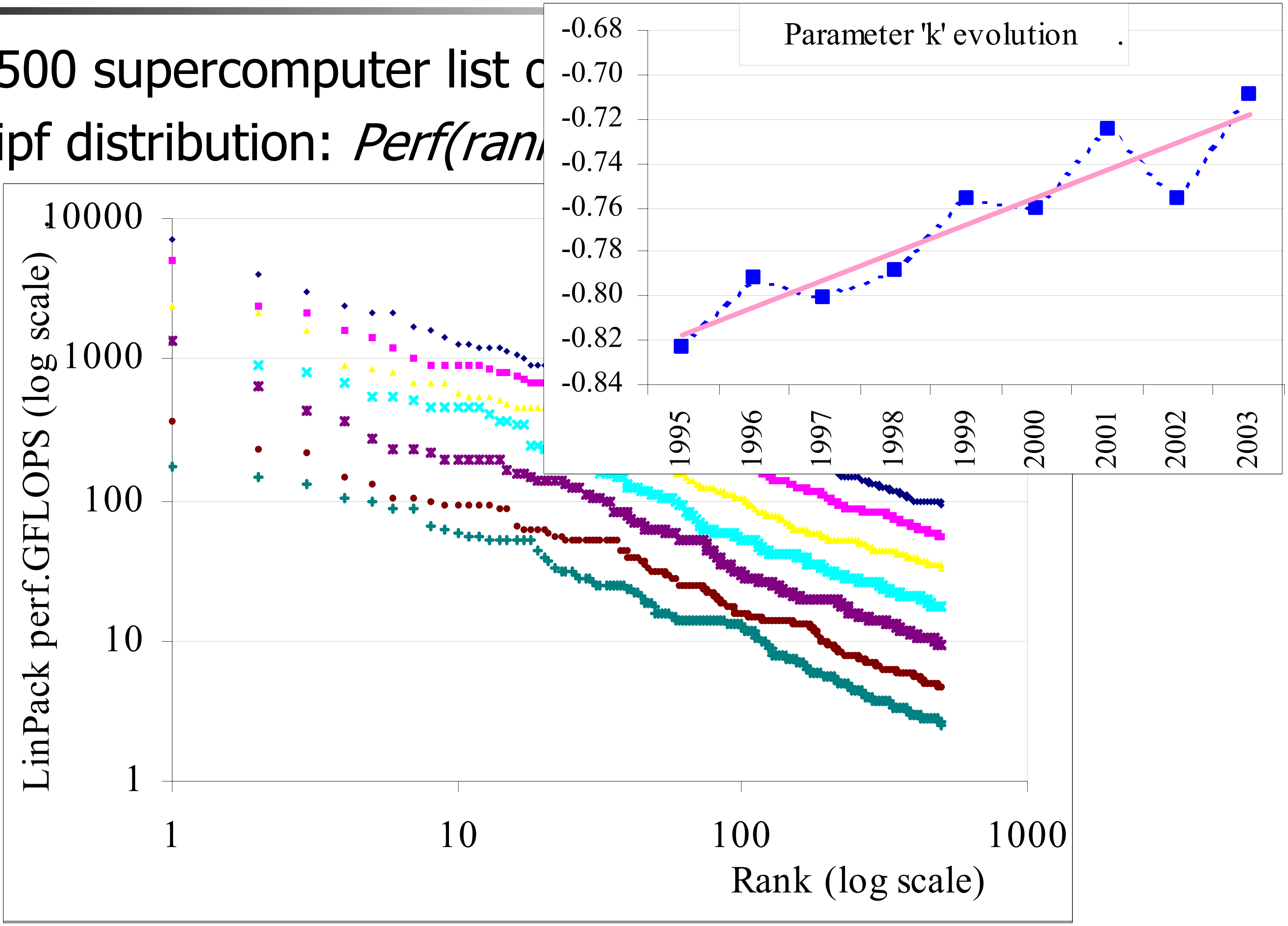
*Matei Ripeanu*

*The University of Chicago*

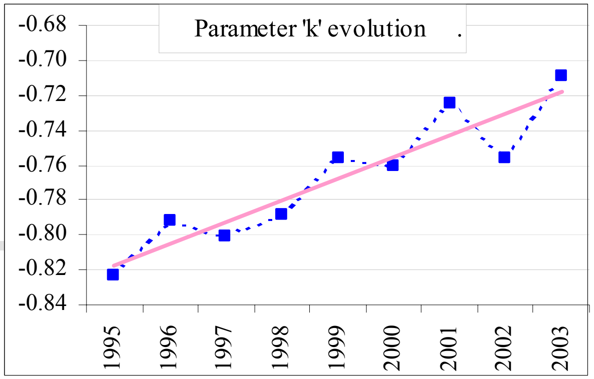
# Why don’t we build a huge supercomputer?

Top500 supercomputer list c

- Zipf distribution: *Perf(ran*



# Trend



- Increasingly interesting to aggregate capabilities of the machines in the tail of this distribution.
  - A virtual machine that aggregates the last 10 in Top500 would rank 32nd in ’95 but 14th in ‘03
- Grid and P2P computing are, in part, results of this trend:
  - Grids focus: *infrastructure enabling controlled, secure resource sharing (*for a relatively small number of resources)
  - P2P focus: *scale, deployability* using integrated stacks.

Challenge: design services that offer the best of both worlds

- complex, secure services, that deliver controlled QoS, are scalable and can be easily deployed.

# Roadmap

P2P

- Impact
- Applications
- Mechanisms

PlanetLab

- Why is this interesting for you?
- Compare resource management in PlanetLab and Globus

Wrap-Up

# P2P Definition(s)

- *Def 1:* “A class of applications that take advantage of resources (e.g., storage, cycles, content) available at the edge of the Internet.” (‘00)
  - Edges often turned off, without permanent IP addresses, etc.
- *Def 2:* “A class of decentralized, self-organizing distributed systems, in which all or most communication is symmetric.” (IPTPS’02)
- Lots of other definitions that fit in between

# P2P impact today (1)

- Widespread adoption
  - KaZaA – 360 million downloads (1.3M/week) one of the most popular applications ever!
- leading to (almost) zero-cost content distribution:

… is forcing companies to change their business models

… might impact copyright laws

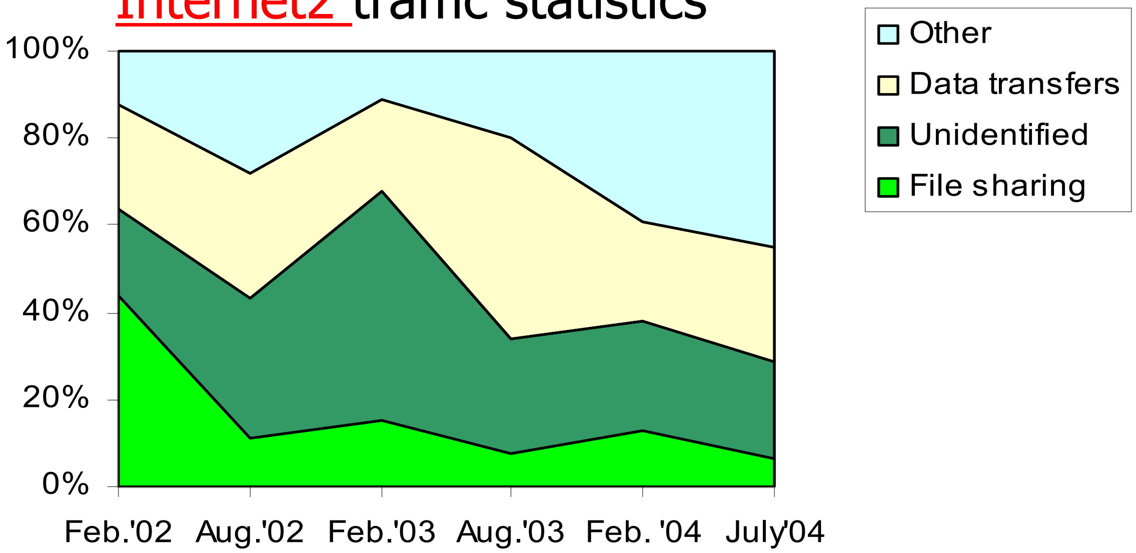
| | |
|---|---|
| FastTrack | 2,460,120 |
| eDonkey | 1,987,097 |
| Overnet | 1,261,568 |
| iMesh | 803,420 |
| Warez | 440,289 |
| Gnutella | 389,678 |
| MP2P | 267,251 |

Sources: www.slyck.com, www.kazaa.com, July '04

# P2P impact today (2)

- P2P – file-sharing - generated traffic may be the single largest contributor to Internet traffic today
- Driving adoption of consumer broadband

Internet2 traffic statistics



Source: www.internet2.edu, July ‘04

# P2P impact today (3)

- A huge pool of underutilized resources lays around,
- users are willing to donate these resources
  - Seti@Home $1.5M / year in additional power consumed
- which can be put to work efficiently (at least for some types of applications)

| | Total | Last 24 hours |
|---|---:|---:|
| Users | 4,236,090 | 23,365 |
| Results received | 764M | 1.13M |
| Total CPU Time | 1.3M years | 1.3K years |
| Floating point operations | | **51.4 TFLOPS** |

Source: Seti@Home website, Oct. 2003

# Roadmap

P2P

- Impact
- Applications
- Mechanisms

PlanetLab

- Why is this interesting for you?
- Compare resource management in PlanetLab and Globus

Wrap-Up

# Applications: Number crunching

- Examples: Seti@Home, UnitedDevices, DistributedScience many others
- Approach suitable for a particular class of problems.
  - Massive parallelism
  - Low bandwidth/computation ratio
  - Error tolerance, independence from solving a particular task
- Problems:
  - Centralized.
  - How to extend the model to problems that are not massively parallel?

Relevant to Grid space:

- Ability to operate in an environment with limited trust and dynamic resources

# Applications: File sharing

- The ‘killer’ application to date
- Too many to list them all: Napster, FastTrack (KaZaA, iMesh), Gnutella (LimeWire, BearShare), Overnet

Relevant to Grid space:

- Decentralized control
- Building a (relatively) reliable, data-delivery service using a large, heterogeneous set of unreliable components.
- Chunking, erasure codes

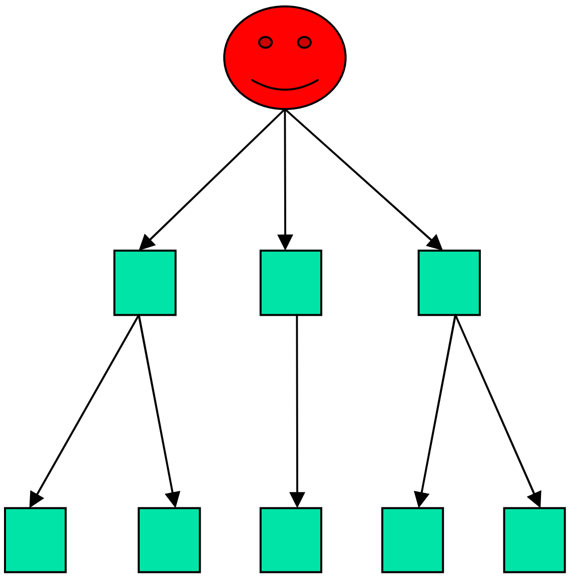
FastTrack (Kazaa) load at a small ISP

| | |
|---|---|
| Bytes transferred | 0.8 TB/day on average |
| Number of download sessions | 230,000/day |
| Number of local users | ≥ 10,000 |
| Number of unique files | ~300,000 |

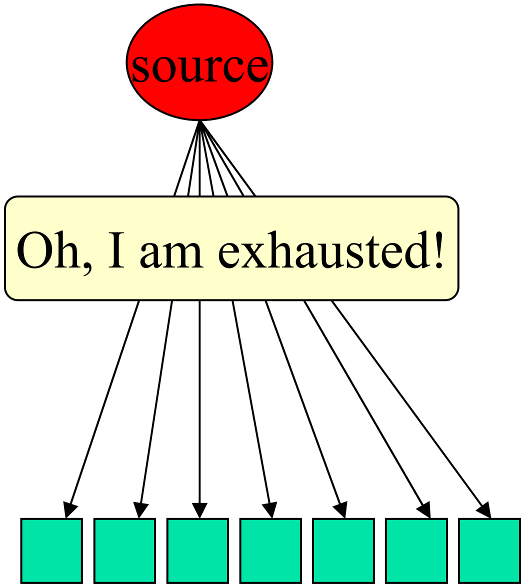
Source: Israeli ISP, data collection 1/15-2/13/2003

# Applications: Content Streaming

- Streaming: the user ‘plays’ the data as as it arrives
- Possible solution:
  - The first few users get the stream from the server
  - New users get the stream from the server or from users who are already receiving the stream



P2P approach



Client/server approach

Relevant to Grid space: offload part of the server load to consumers to improve scalability

# Applications: Performance benchmarking

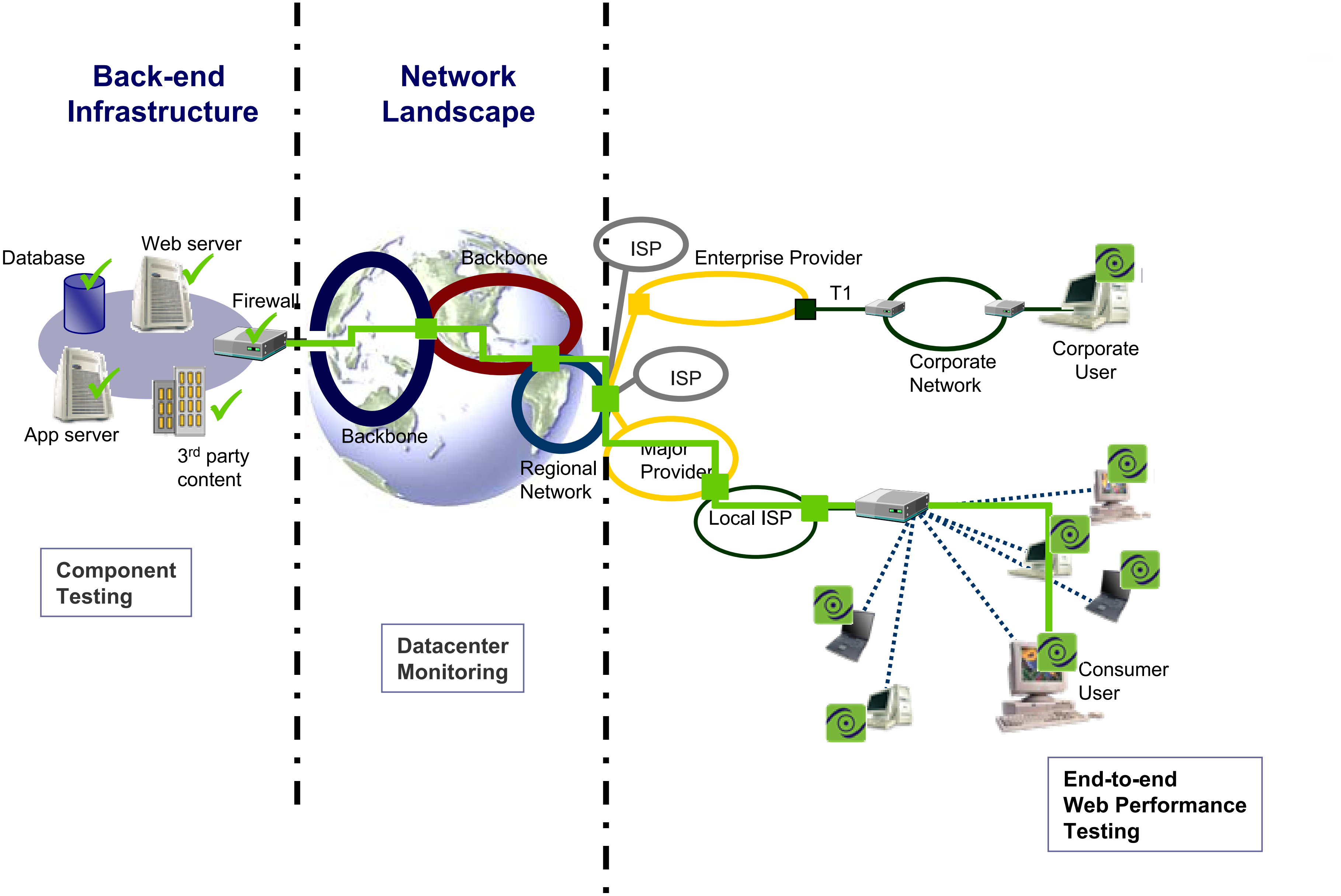
Problem:

- Evaluate the performance of your service (Grid service, HTTP server) form *end-user* perspective
  - Multiple views on your site performance

Relevance to Grid space:

- Grid clients are heterogeneous, geographically dispersed
- Benchmark services for this set of consumers

# End-to-end Performance Benchmarking



**Back-end Infrastructure**

**Network Landscape**

Database

Web server

Firewall

App server

3rd party content

Backbone

Backbone

Regional Network

ISP

Enterprise Provider

T1

ISP

Corporate Network

Corporate User

Major Provider

Local ISP

Consumer User

**Component Testing**

**Datacenter Monitoring**

**End-to-end Web Performance Testing**

Slide source: www.porivo.com

# Many other P2P applications …

- Backup storage (HiveNet, OceanStore)
- Collaborative environments (Groove Networks)
- Web serving communities (uServ)
- Instant messaging (Yahoo, AOL)
- Anonymous email
- Censorship-resistant publishing systems (Ethernity, Freenet)
- Spam filtering

# Mechanisms

To obtain a resilient system:

- integrate multiple components with uncorrelated failures, and
- use data and service replication.

To improve delivered QoS:

- move service delivery closer to consumer,
- integrate multiple providers with uncorrelated demand curves (reduces over-provisioning for peak loads)

To generate meaningful statistics, to detect anomalies:

- provide views from multiple vantage points

To improve scalability:

- Use decentralized (local) control, unmediated interactions

# Roadmap

P2P

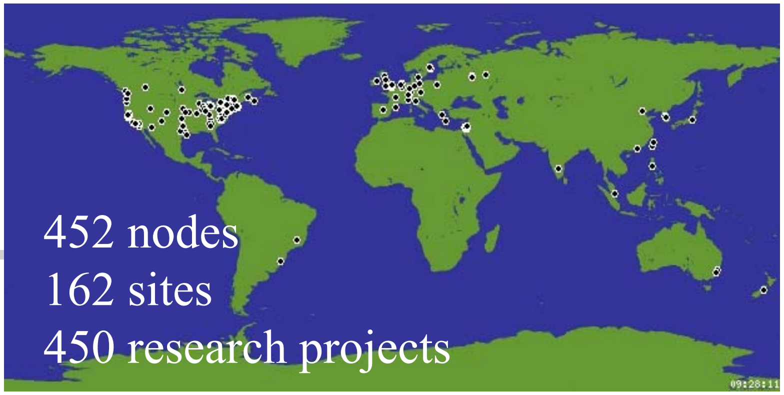
- Impact
- Applications
- Mechanisms

PlanetLab

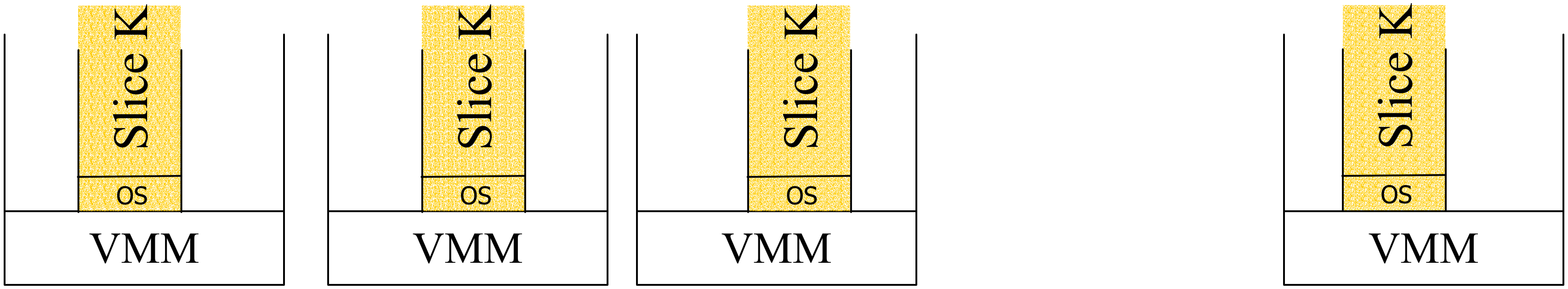
- Why is this interesting for you?
- Compare assumptions and resource management mechanisms in PlanetLab and Globus

Wrap-Up

# PlanetLab

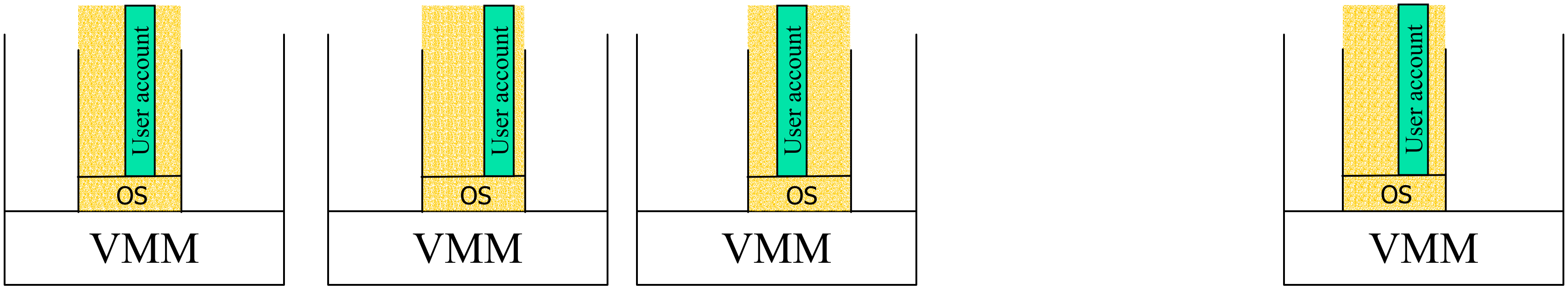


- Testbed to experiment with your networked applications.
  - >400 nodes, >150 sites,
  - PlanteLab consortium: 80+ universities, Intel, HP
- View presented to users: a distributed set of VMs
  - Allocation unit: a slice = a set of virtual machines (VM), one VM at each node.



# PlanetLab usage examples

- Stress-test your Grid services (Globus RLS)
- GSLab: a playground to experiment with grid-services
- ‘Better-than-Internet’ services:
  - Resilient Overlays
  - Multipath TCP (mTCP)
  - Multicast Overlays



# Why should you find PlanetLab interesting?

1. Open, large-scale testbed for your P2P applications or Grid services
2. Solves a similar problem to Grids/Globus: building virtual organizations (or resource federations)

   *Grids*: testbeds (deployments of hardware and software) to solve computational problems.

   *PlanetLab*: testbed to play with CS applications

*Main problem for both*: enable resource sharing across multiple administrative domains

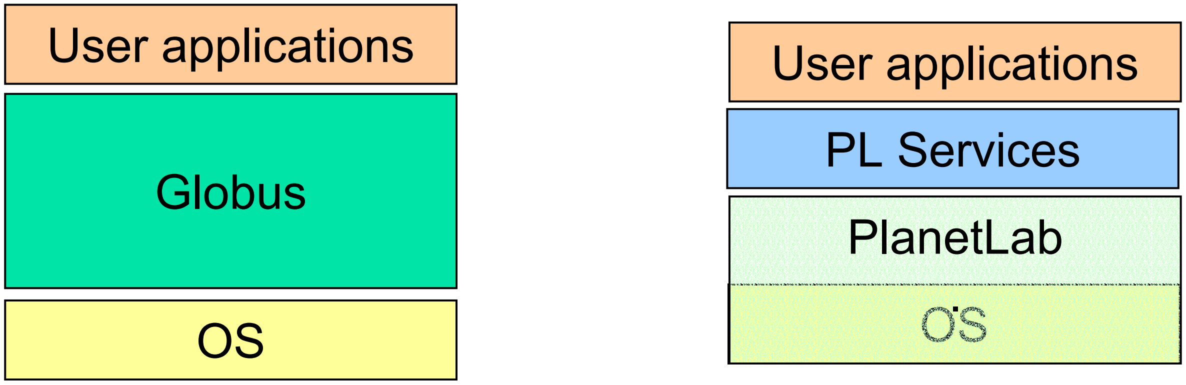
# Roadmap

- Summarize starting assumptions on: user communities, applications, resources and attempt to explain differences
- Look at mechanisms to build VO.

# Assumptions: User communities

- PlanetLab: users are *CS scientists* that experiment with and deploy *infrastructure* services.
- Globus: users from a more *diverse pool of scientists* that are interested to run efficiently their (*end-user*) applications.

Implication: functionality offered



# Assumptions: Application characteristics

Different view on geographical resource distribution:

- PlanetLab services: «distribution is a goal»
  - leverage multiple vantage points for network measurements, or to exploit uncorrelated failures in large sets of components
- Grid applications: «distribution is a fact of life»
  - resource distribution: a result of how the VO was assembled (due to administrative constraints).

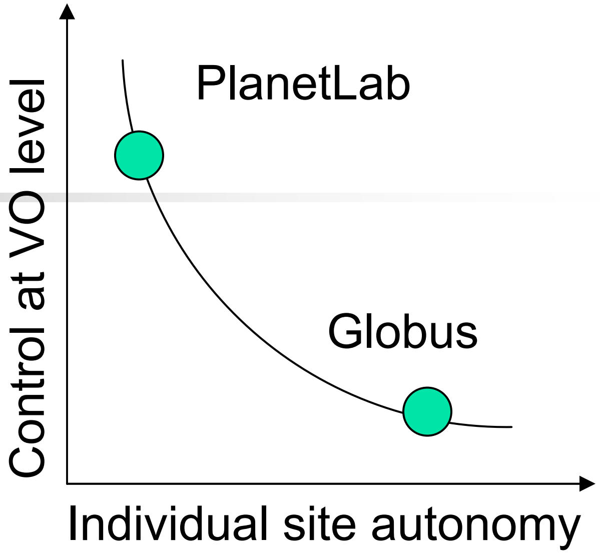
Implication: mechanism design for resource allocation

# Assumptions: Resources

- PlanetLab mission as testbed for a *new* class of networked services allows for little HW/SW heterogeneity.
- Globus supports a large set architectures; sites with multiple security requirements

Implications: complexity, development speed

# Assumptions: Resource ownership



Goal: individual sites retain control over their resources

- PlanetLab limits the autonomy of individual sites in a number of ways:
  - VO admins: Root access, Remote power button
  - Sites: Limited choice of OS, security infrastructure
- Globus imposes fewer limits on site autonomy
  - Requires fewer privileges (also can run in user space, )

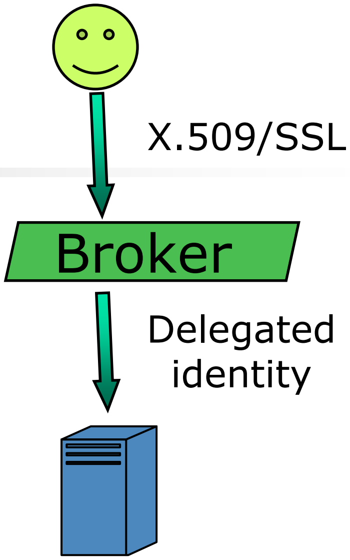
PlanetLab emphasizes global coordination over local autonomy to a greater degree than Globus

Implications: ease to manage and evolve the testbed

# Building Virtual Organizations

- Individual node/site functionality
- Mechanisms at the aggregate level
  - Security infrastructure
    - Delegation mechanisms
  - Resource allocation and scheduling
  - Resource discovery, monitoring, and selection.

# Delegation mechanisms: Identity delegation



Broker/scheduler usage scenario:

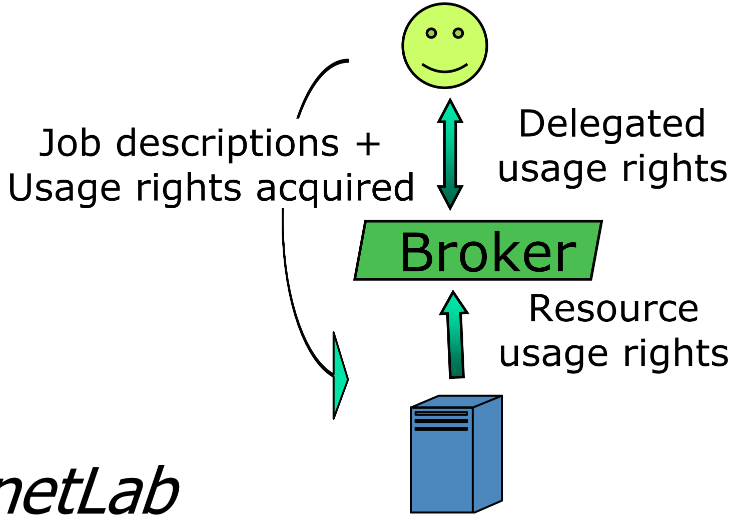
- User A sends a job to a broker service which, in turn, submits it to a resource. The resource manager makes authorization decisions based on the identity that originated the job (A).

| *Globus* | *PlanetLab* |
|---|---|
| Implementation based on delegated X.509 proxies | None |

# Delegation mechanisms: Delegating rights to use resources

Broker/scheduler usage scenario:

- User A acquires capabilities from various brokers then submits the job.



Job descriptions + Usage rights acquired

Delegated usage rights

Broker

Resource usage rights

*GGF/Globus*

WS-Agreements protocols:

- To represent 'contracts' between providers and consumers.
- Local enforcement mechanism is not specified

The two efforts are complementary!

*PlanetLab*

- Individual nodes managers hand out *capabilities:* akin to time-limited reservations
- Capabilities can be traded
- Extra layer to: provide secure transfer, prevent double spending, offer external representation

# Global resource allocation and scheduling

Users

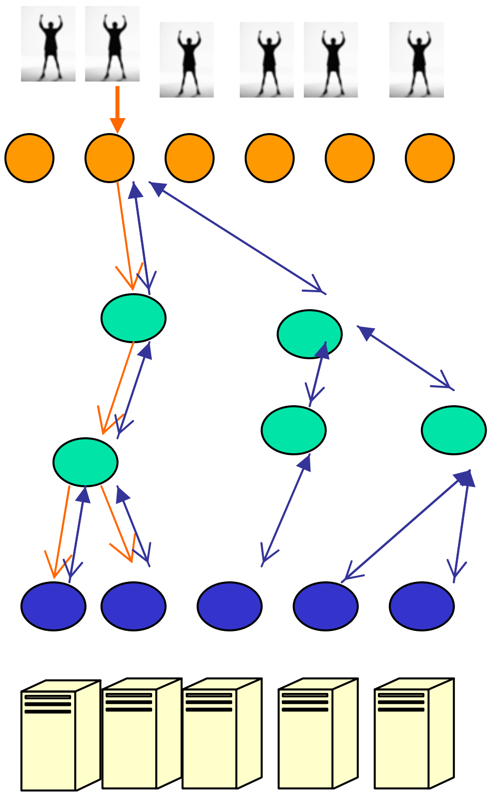
Application Managers

Brokers / Agents

Node Managers

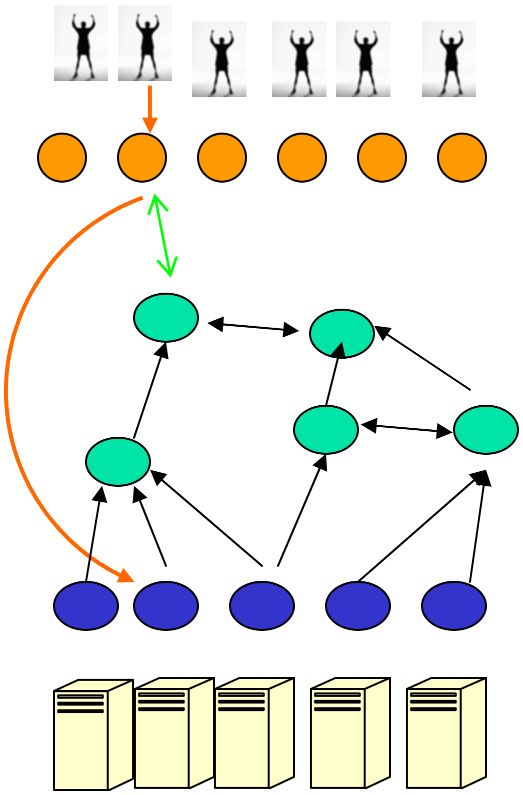
Nodes (Resources)

*Globus*



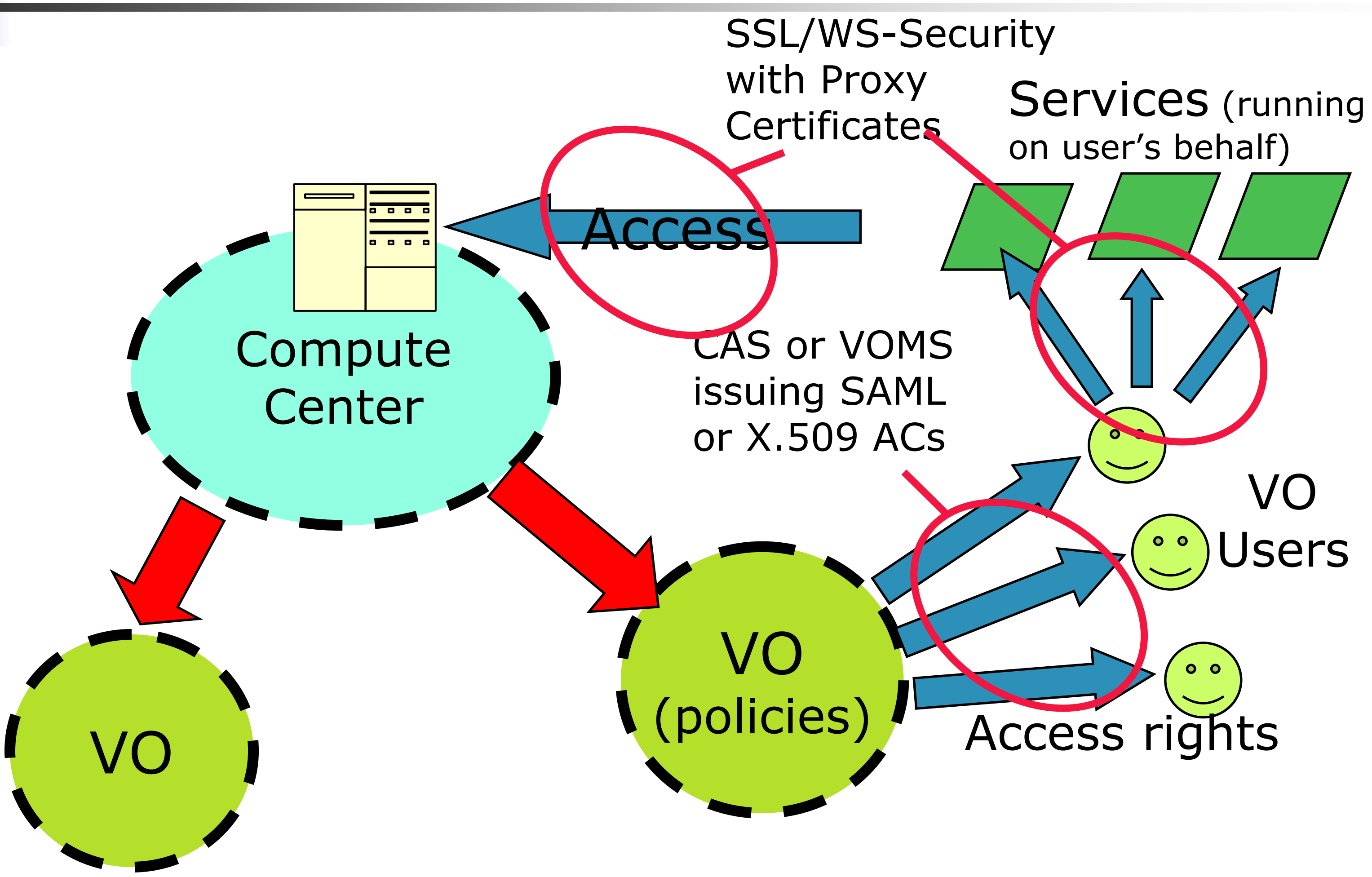
- Identity delegation
- Sends job descriptions

*PlanetLab*



- Resource usage delegation
- Sends capabilities (leases)

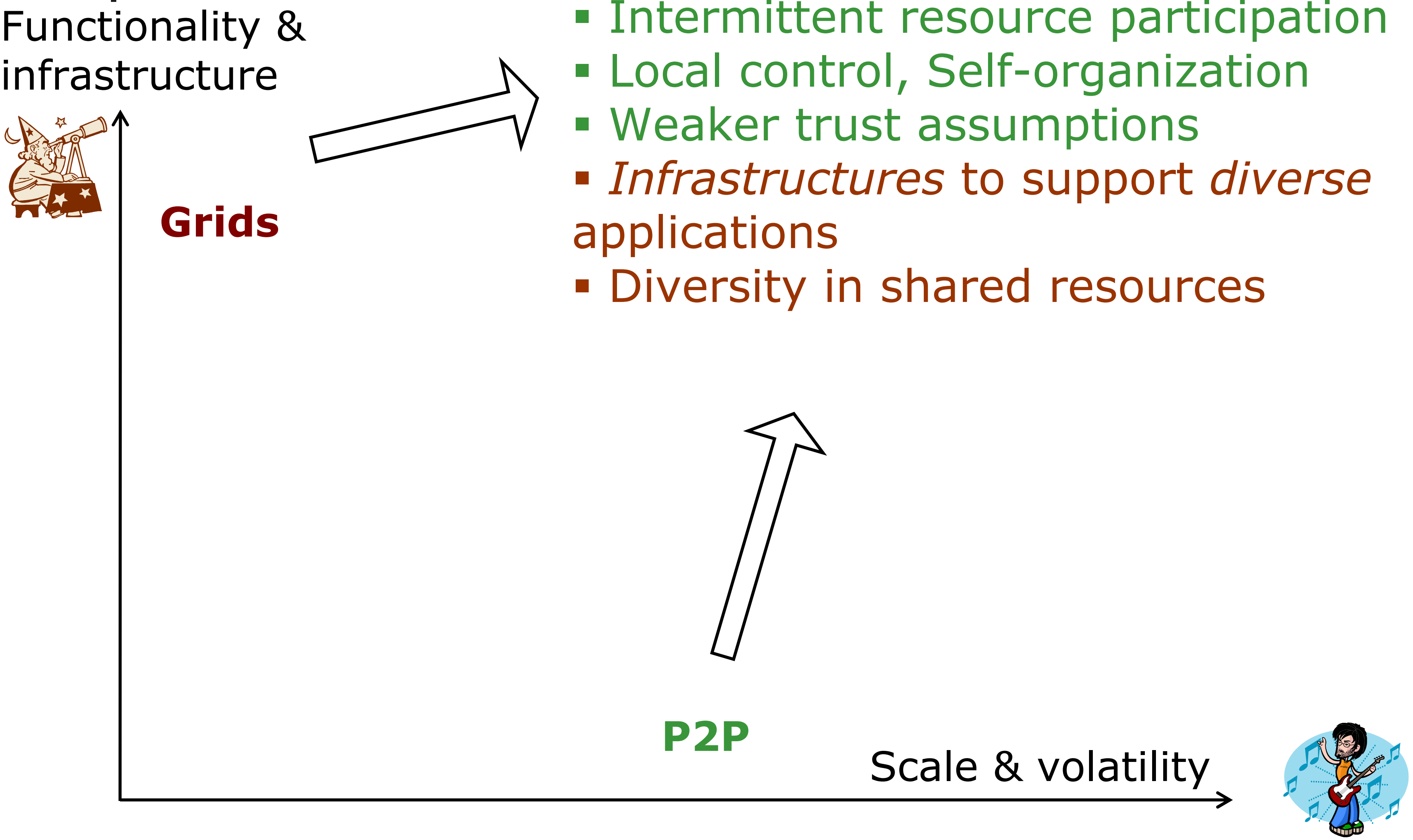
# Multiple VOs



# Wrap-up

Convergence:
Large, Dynamic, Self-Configuring Grids

- Large scale
- Intermittent resource participation
- Local control, Self-organization
- Weaker trust assumptions
- *Infrastructures* to support *diverse* applications
- Diversity in shared resources



# Fin

Links to papers, tech-reports, slides:

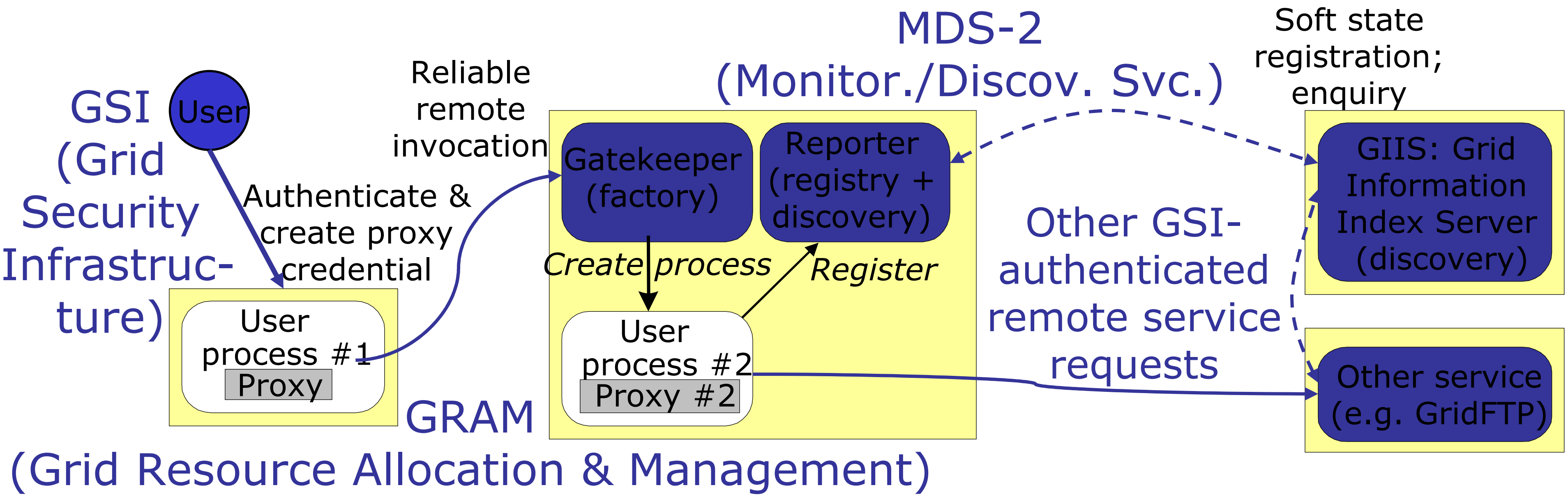
Distributed Systems Group @ UChicago

- http://dsl.cs.uchicago.edu

Thank you.

# GT2 in One Slide

- Grid protocols (GSI, GRAM, …) enable resource sharing within virtual orgs; toolkit provides reference implementation ( ■ = Globus Toolkit services)



MDS-2
(Monitor./Discov. Svc.)

Soft state
registration;
enquiry

GSI
(Grid
Security
Infrastructure)

User

Reliable
remote
invocation

Authenticate &
create proxy
credential

User
process #1
Proxy

Gatekeeper
(factory)

Reporter
(registry +
discovery)

*Create process*

*Register*

User
process #2
Proxy #2

Other GSI-
authenticated
remote service
requests

GIIS: Grid
Information
Index Server
(discovery)

Other service
(e.g. GridFTP)

GRAM
(Grid Resource Allocation & Management)

- Protocols (and APIs) enable other tools and services for membership, discovery, data mgmt, workflow, …