

Virtual Machines in Distributed Environments

Maurício Tsugawa on behalf of
José A. B. Fortes
Advanced Computing and
Information System Laboratory
University of Florida, Gainesville

ACIS Lab history + statistics

- Founded 8/10/01
- 20+ people
- 5 “associated” faculty
 - ECE, CISE, MSE
 - Also 10+ “associated” faculty from Purdue, CMU, U. Colorado, Northwestern U., U. Mass., NCSU ...
- 7 M dollars in funding, 5 M dollars in equipment
 - Approximately 1.5 M of subcontracts
 - NSF, NASA, ARO, IBM, Intel, SRC, Cisco, Cyberguard ...
- Computer infrastructure/lab
 - 300+ CPUs, .5 Tflops, 9 TBytes, Gigabit connections
 - Access to CPUs at Purdue, Northwestern, Stevens I.

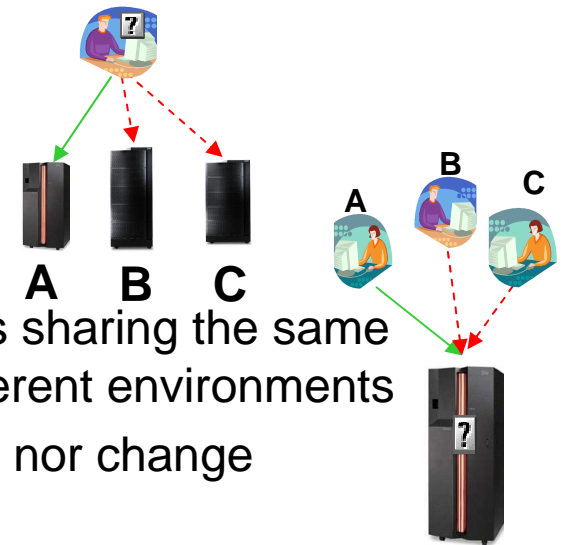


Outline

- What's in a talk title
 - Environment as a container for app execution
 - Distributed “a la” power grid
 - Virtualization for creation and coexistence of different environments in physical resources
- A Grid-building recipe
- Words are easy, let's build it: In-VIGO
 - Architecture, Deployments and Futures
 - Virtual Machines, Data, Networks and Applications
 - Turning Applications into Grid Services
- Conclusions

Resource sharing

- Traditional computing/data center solutions:
 - Multitask/multiuser operating systems, user accounts, file systems ...
 - Always available but static configurations
 - Sharing possible if apps run on similar execution environments
 - Centralized administration
 - Tight control on security, availability, users, updates, etc
- Distributed Grid/datacenter requirements
 - Multiple administrative domains
 - Different policies and practices at each domain
 - Many environments possible
 - Dynamic availability
 - Must run all kinds of applications
 - Application user will neither trust unknown users sharing the same resource nor redevelop application to run in different environments
 - Resource owner will neither trust arbitrary users nor change environment for others' applications



“Classic” Virtual Machine

- Copy of a real machine
 - “Any program run under the VM has an effect identical with that demonstrated if the program had been run in the original machine directly” ¹
- Isolated from other virtual machines
 - “...transforms the single machine interface into the illusion of many” ²
- Efficient
 - “A statistically dominant subset of the virtual processor’s instructions is executed directly by the real processor” ²
- Also known as a “system VM”

¹ “Formal Requirements for Virtualizable Third-Generation Architectures”, G. Popek and R. Goldberg, *Communications of the ACM*, 17(7), July 1974

² “Survey of Virtual Machine Research”, R. Goldberg, *IEEE Computer*, June 1974

Process vs. System VMs

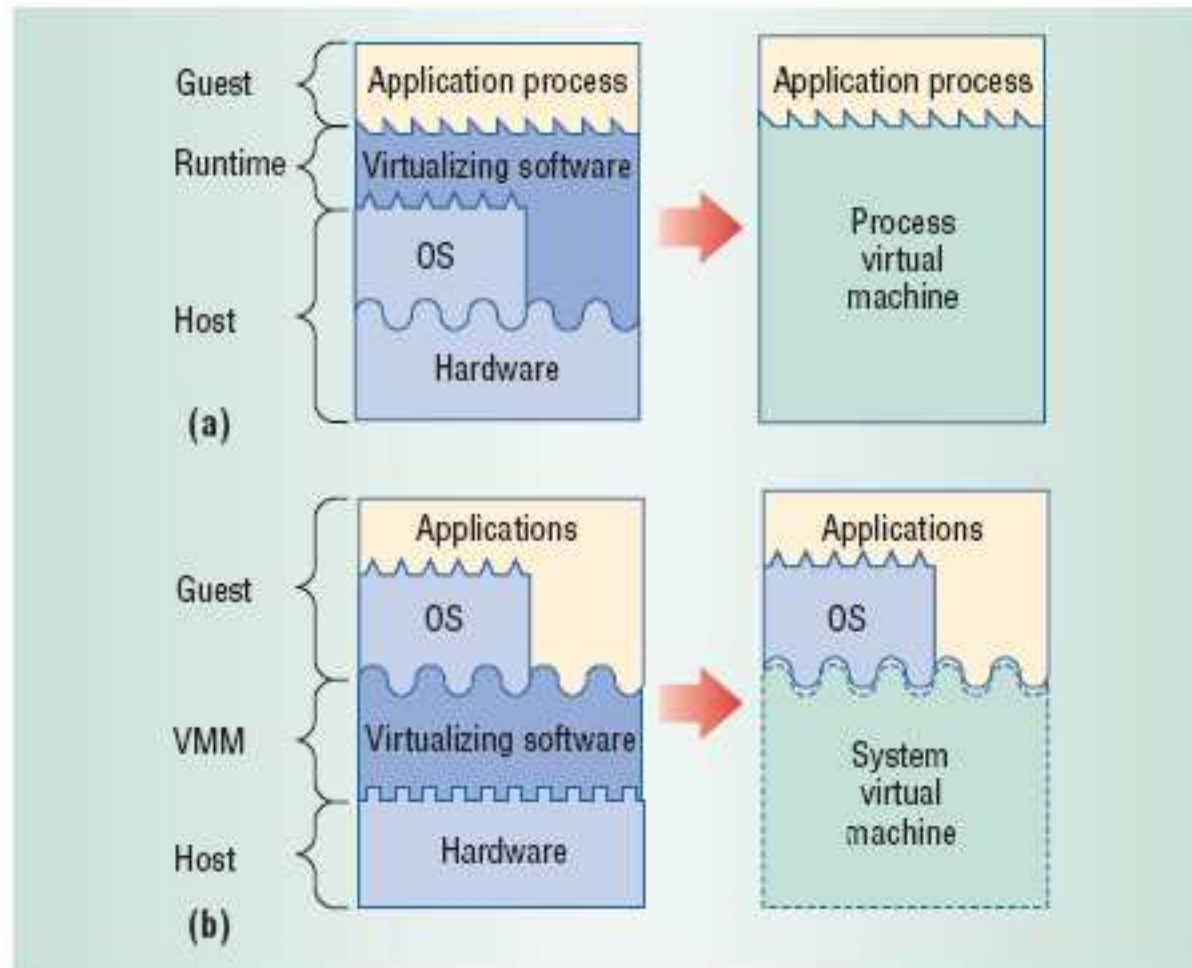
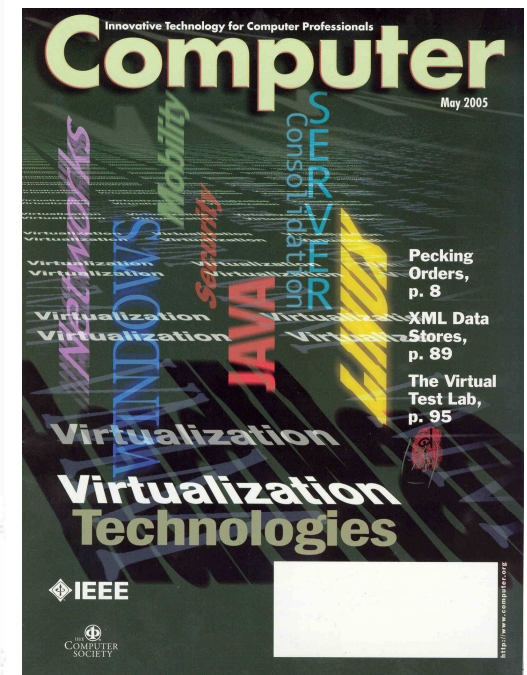


Figure 3. Process and system VMs. (a) In a process VM, virtualizing software translates a set of OS and user-level instructions composing one platform to those of another. (b) In a system VM, virtualizing software translates the ISA used by one hardware platform to that of another.

- In Smith and Nair's "The architecture of Virtual machines", Computer, May 2005

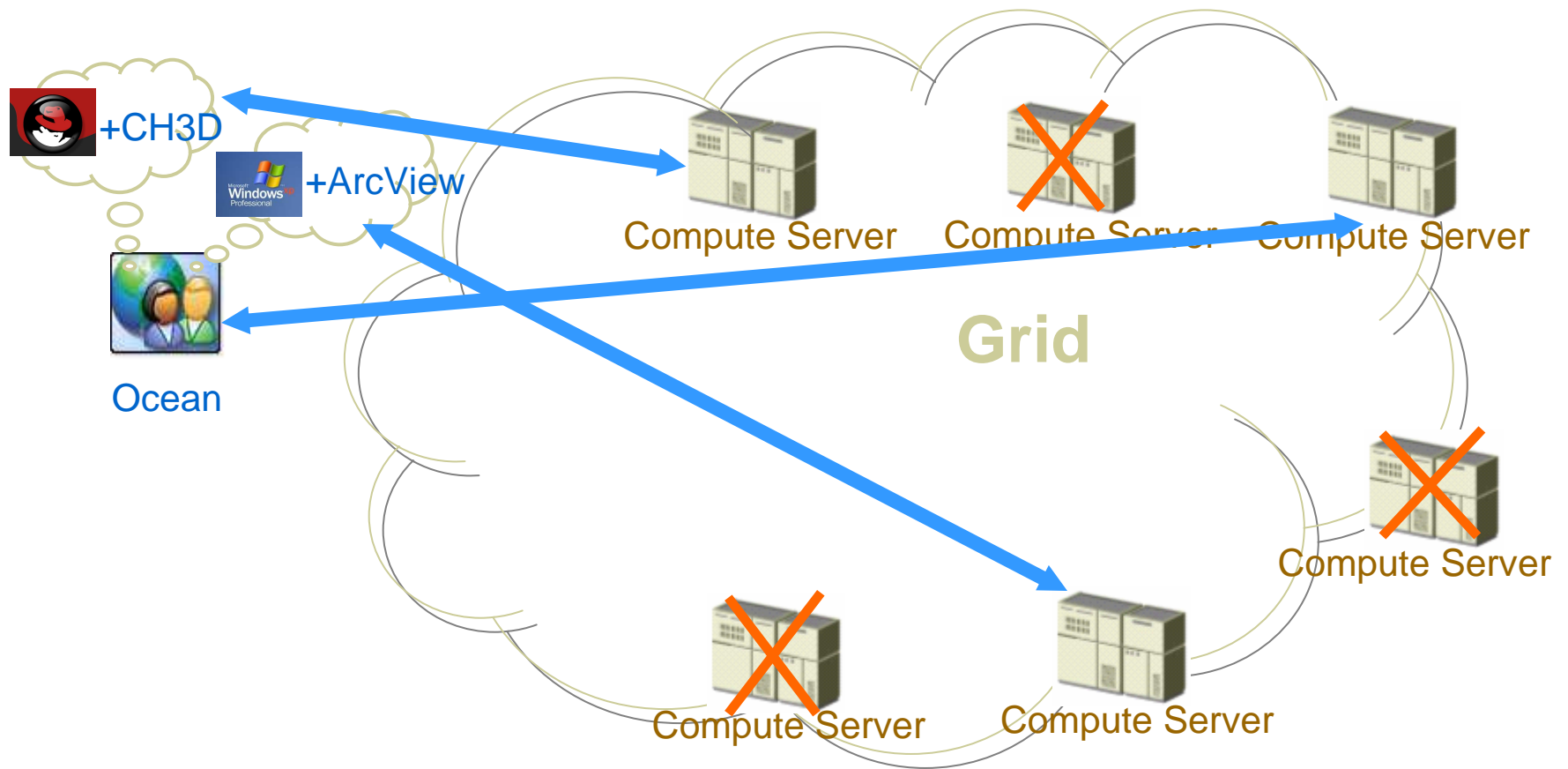


Classic Virtual Machines

- Virtualization of instruction sets (ISAs)
 - Language-independent, binary-compatible (*not* JVM)
- 70's (IBM 360/370..) – 00's (VMware, Microsoft Virtual Server/PC, z/VM, Xen, Power Hypervisor, Intel Vanderpool, AMD Pacifica ...)
- ISA+ OS + libraries + software = execution environment

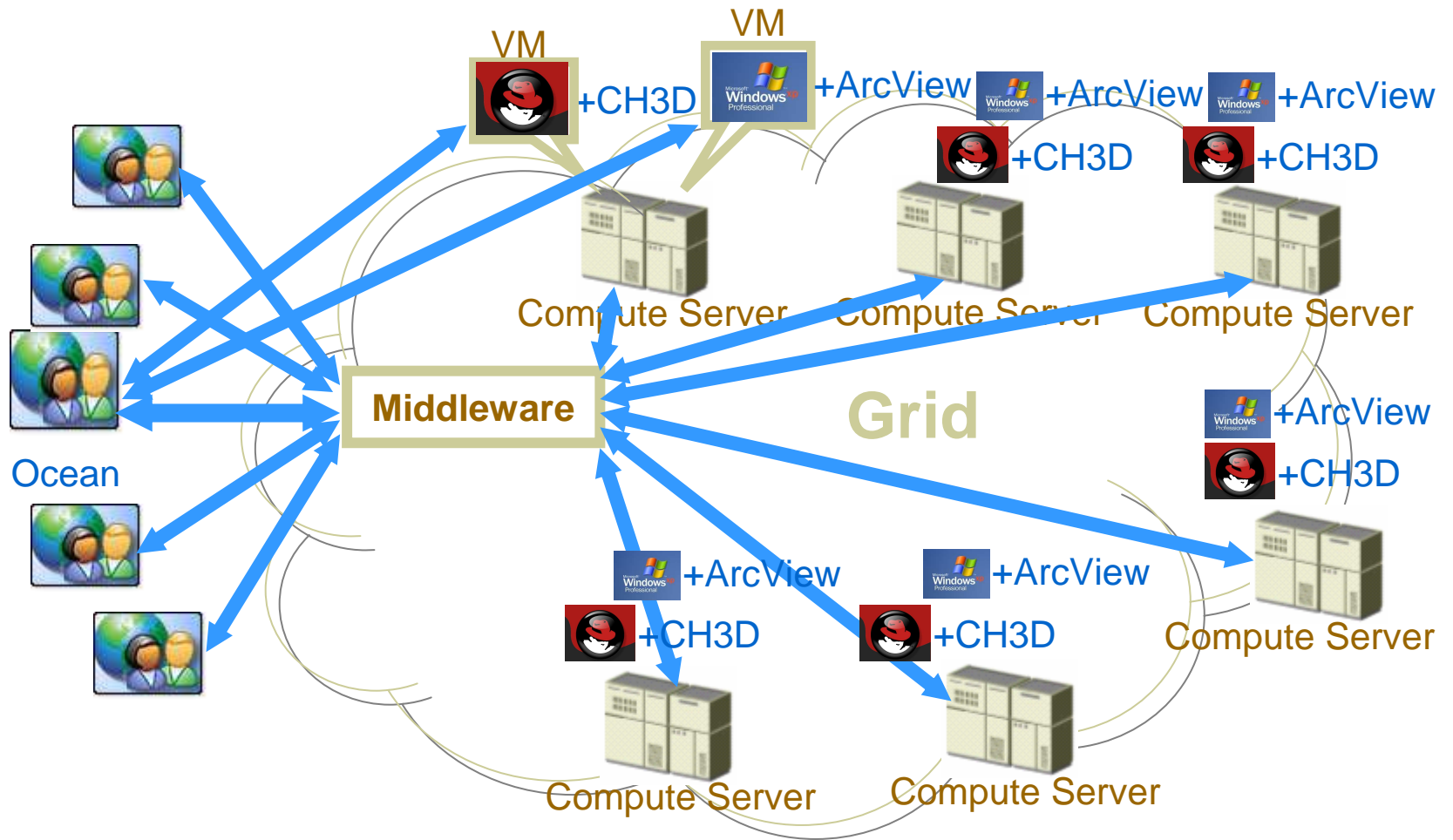


1 user, 1 app, several environments



Slide provided by M. Zhao

Many users, 1 app, many environments



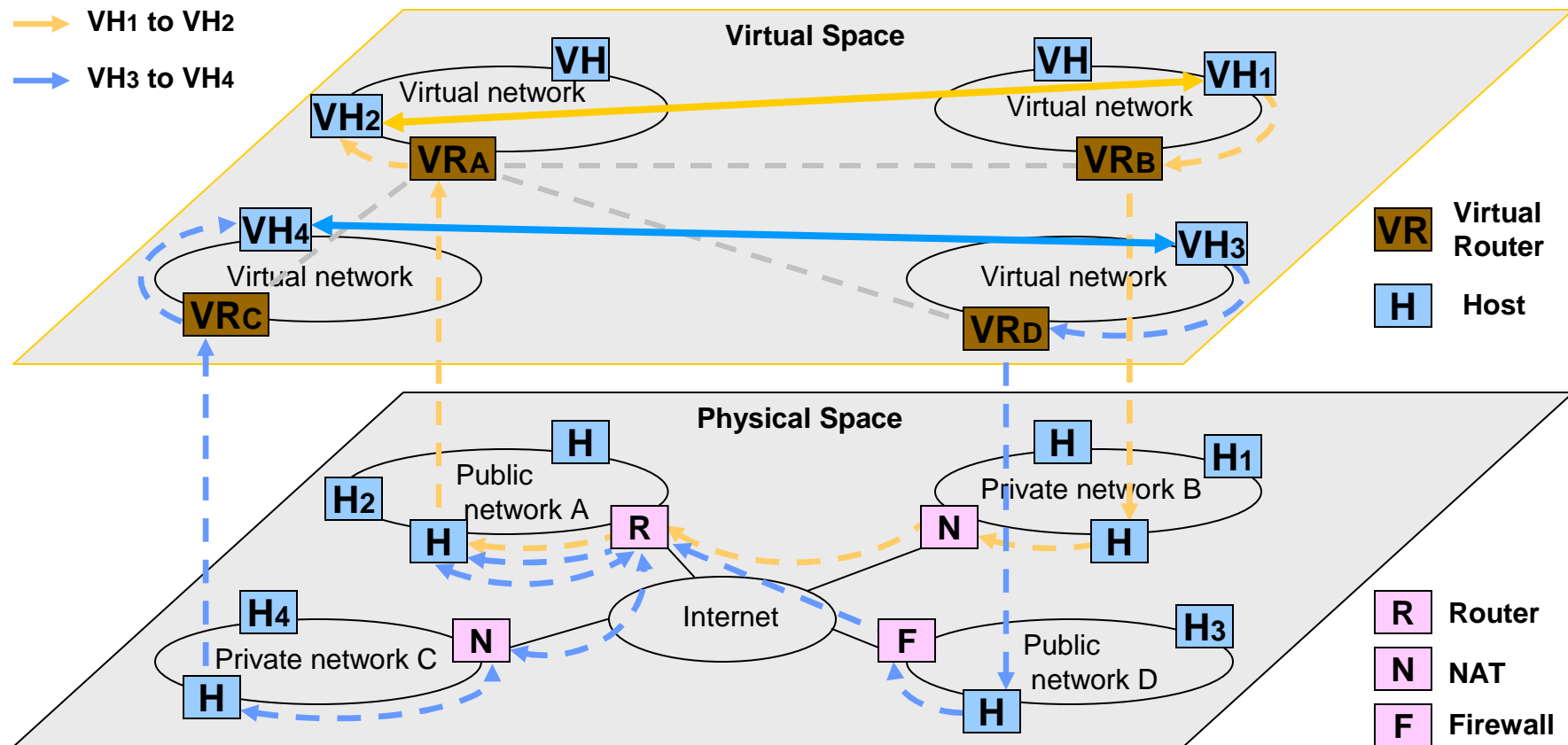
Slide provided by M. Zhao

Virtualization technology for grids

- Resource virtualization technology
 - Enables a resource to simultaneously appear as multiple resources with possibly different functionalities
 - Polymorphism, manifolding and multiplexing
- Virtual networks, data, applications, interfaces, peripherals, instruments ...
 - Emergent technologies

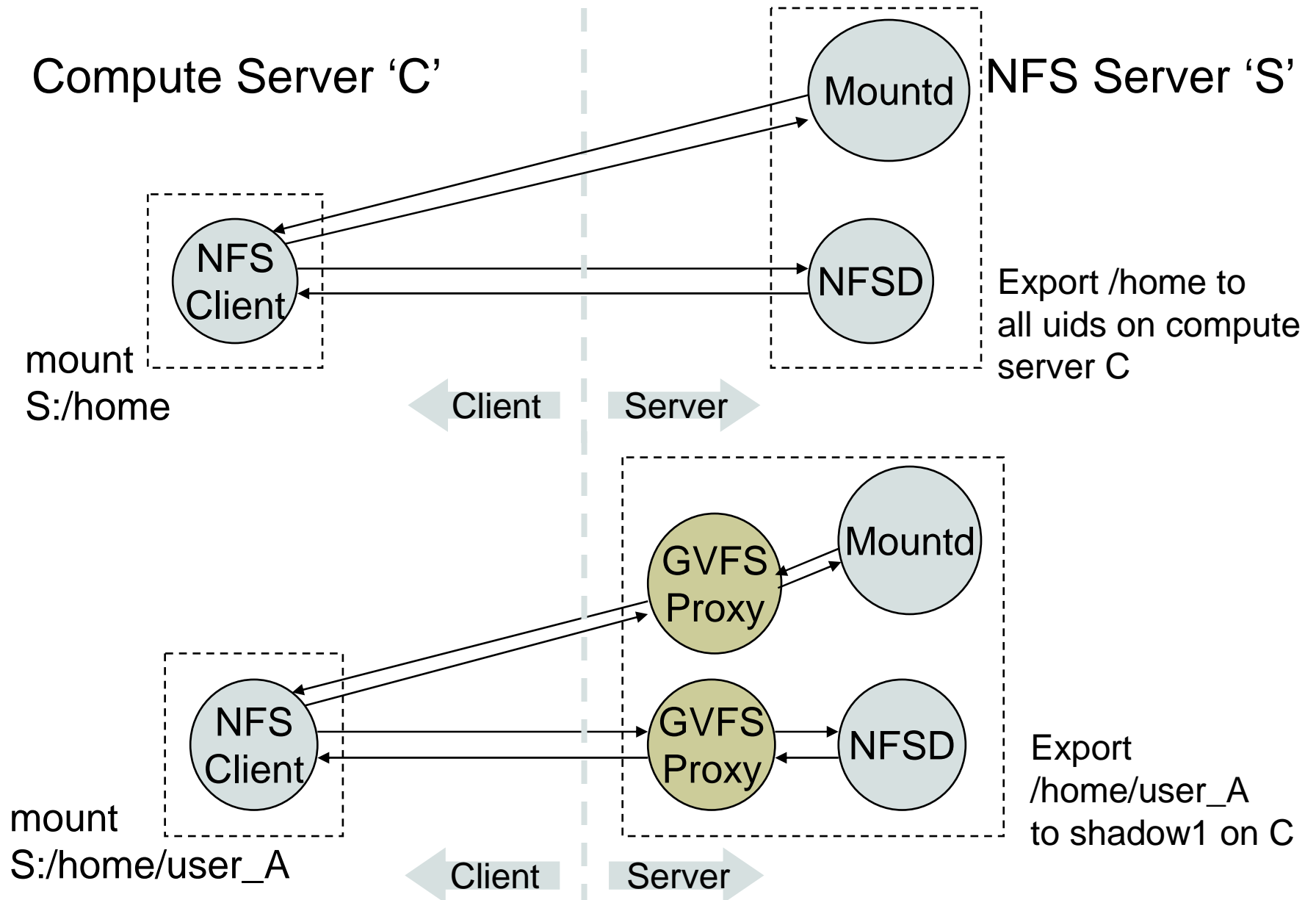
Virtual networks

- logical links:
 - multiple physical links, routing via native Internet routing
 - tunneling, virtual routers, switches, ...
 - partial to total isolation



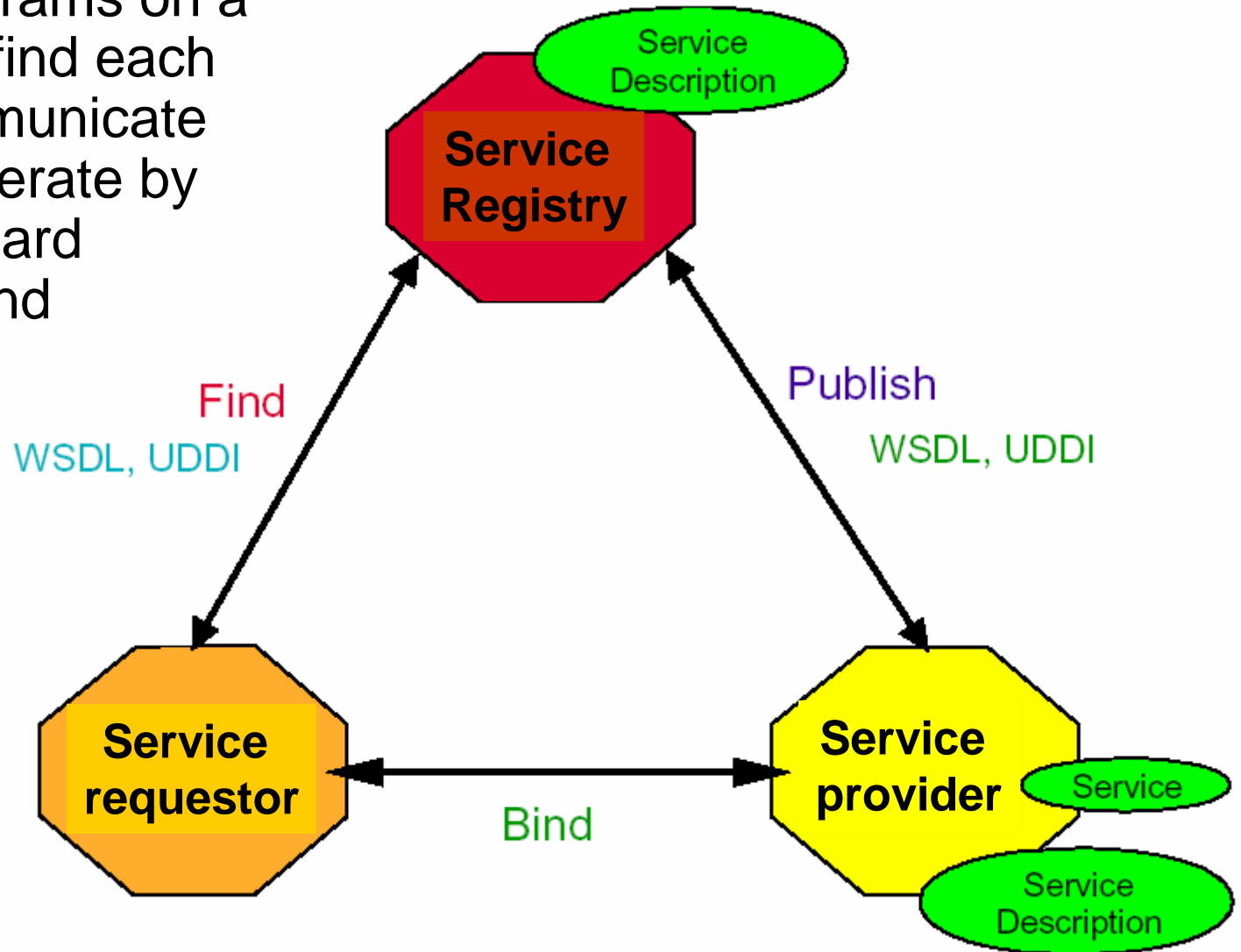
Slide provided by M. Tsugawa

Data/file virtualization



Web services framework

- allows programs on a network to find each other, communicate and interoperate by using standard protocols and languages



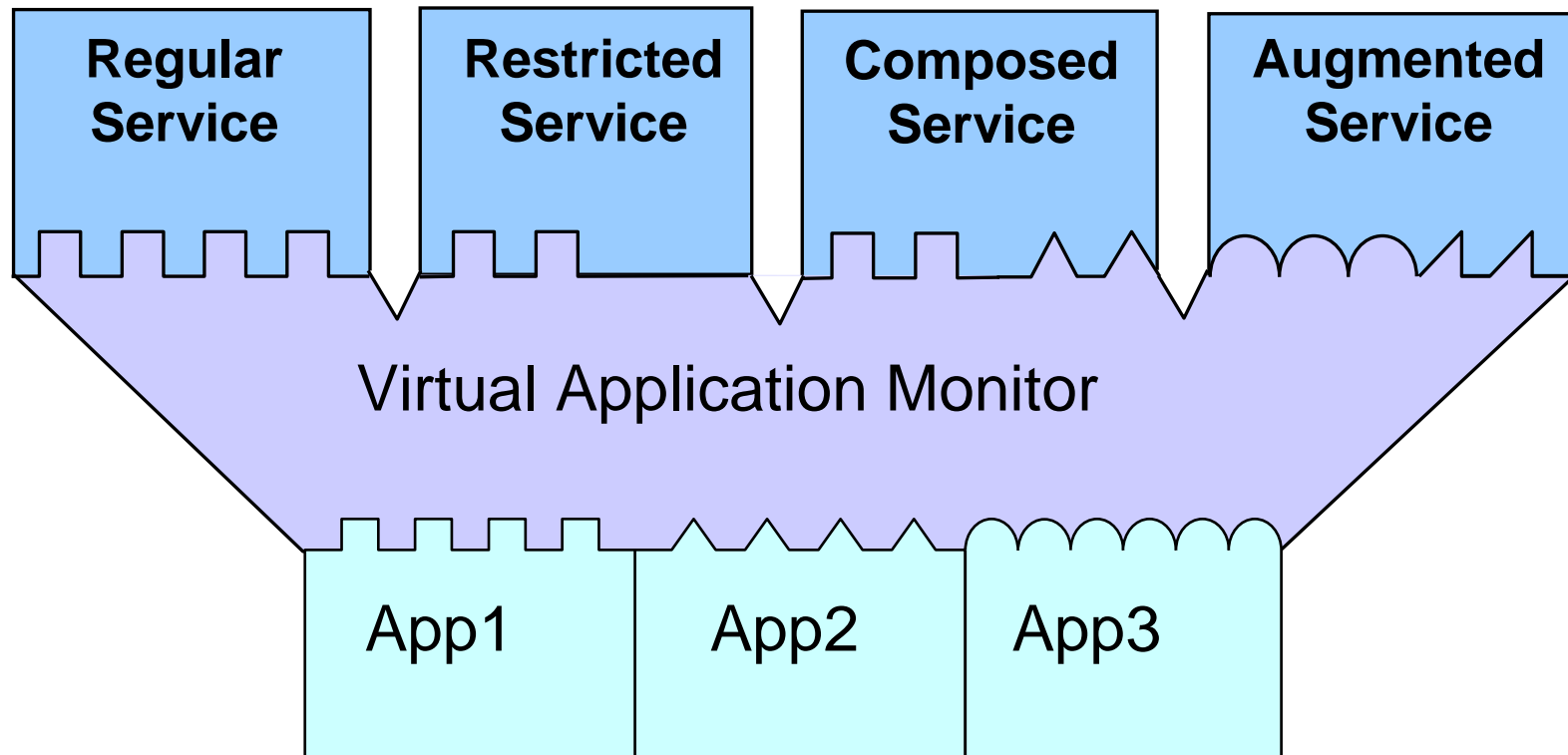
Basic service description: interface definition

- abstract or reusable service definition that can be instantiated and referenced by multiple service implementation definitions



- different implementations using the same application can be defined to reference different service definitions – a form of virtualization

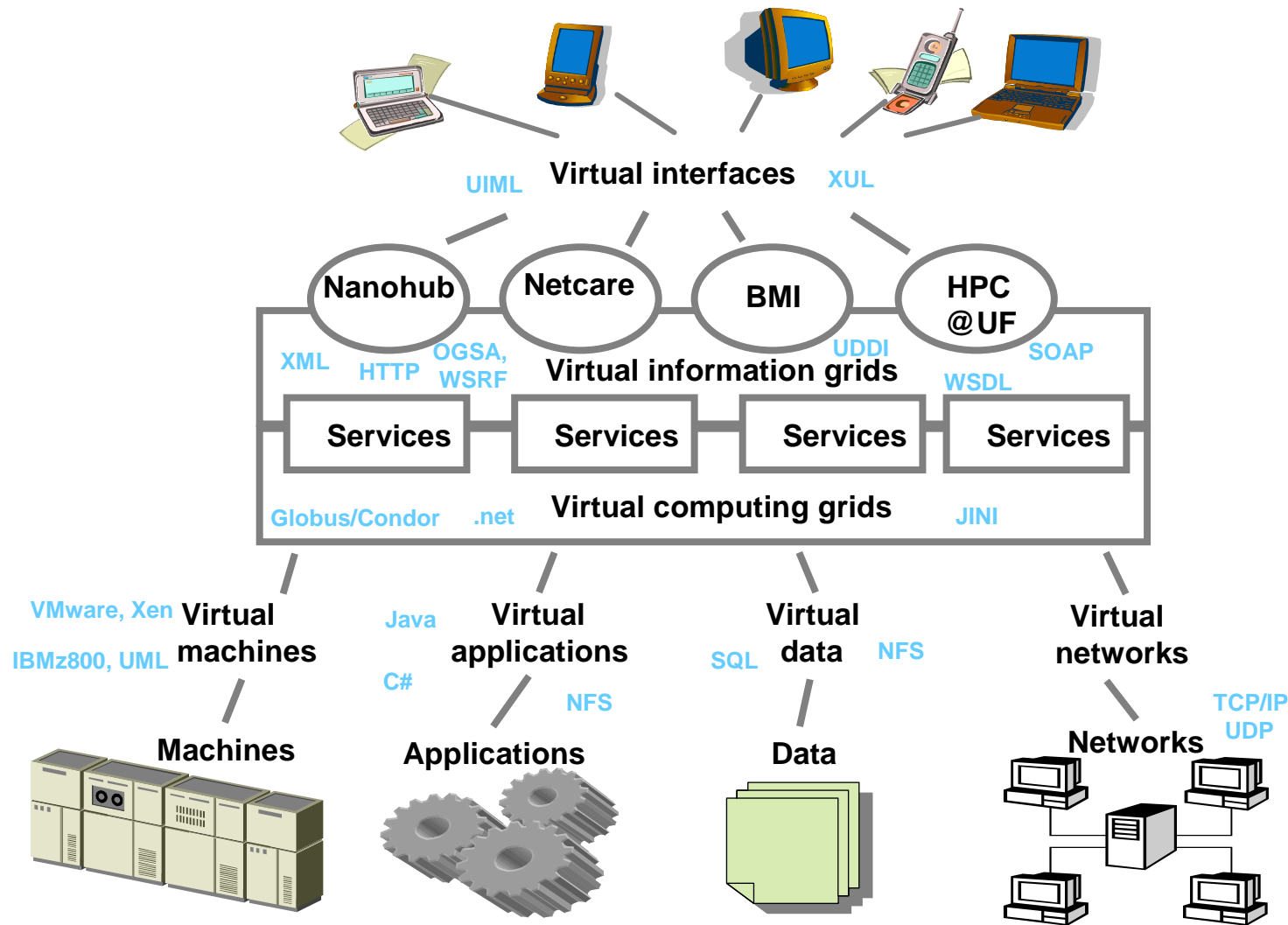
Application virtualization



A Grid-building recipe

- ❶ Virtualize to fit needed environments
 - ❷ Use services to generate “virtuals”
 - ❸ Aggregate and manage “virtuals”
 - ❹ Repeat ❶ ❷ ❸ as needed
- Net result:
 - users interact with virtual entities provided by services
 - middleware interacts with physical resources
 - In-VIGO is a working proof-of-concept!

The In-VIGO approach

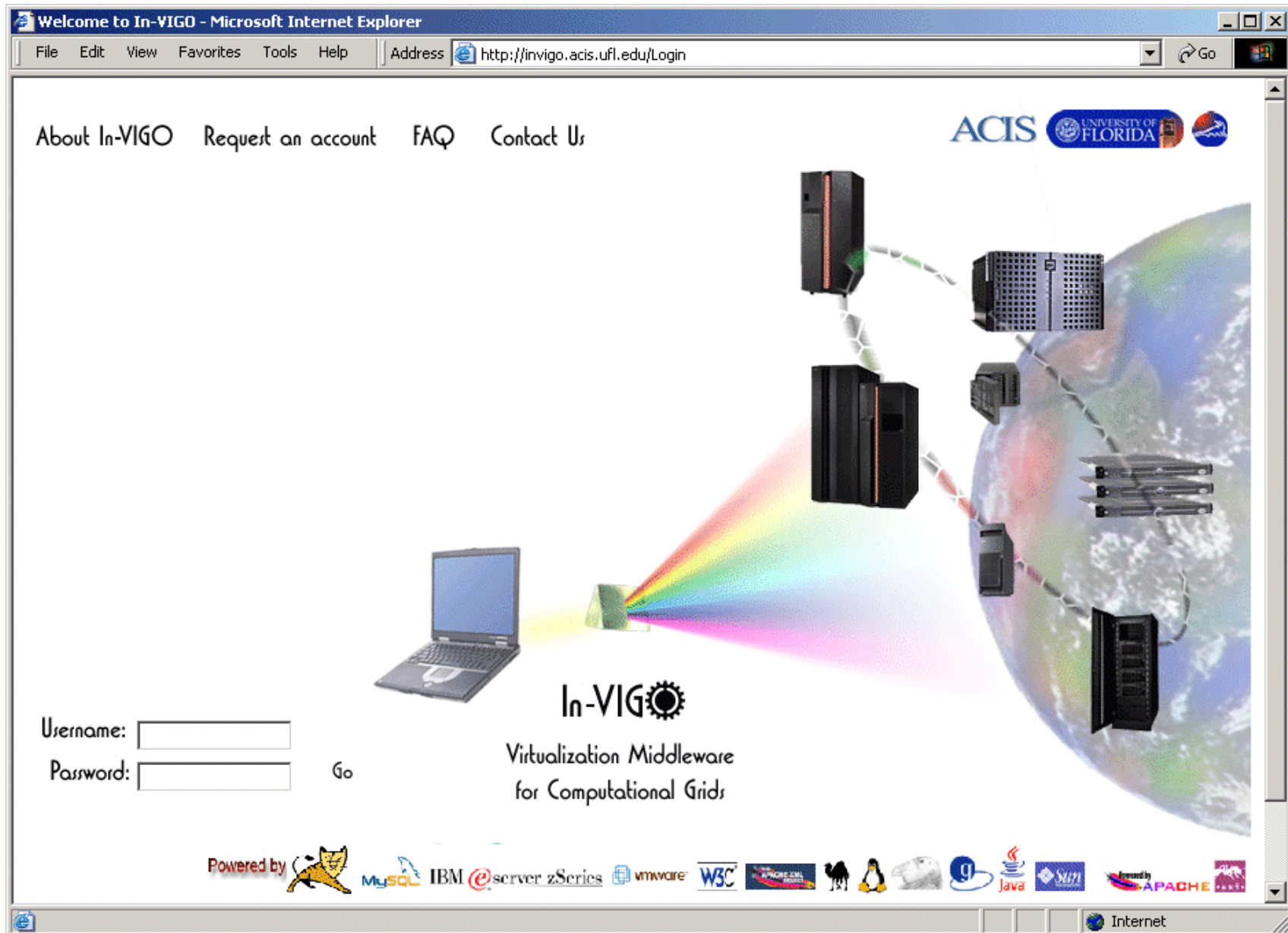


- ✓ local control, decentralized management
- ✓ open general-purpose standards
- ✓ non-trivial QoS

In-VIGO: a user's view

- Enables computational engineering and science [In-Virtual Information Grid Organizations](#)
- Motivations:
 - Hide complexity of dealing with cross-domain issues
 - From application developers
 - From end users
 - Provide secure execution environments
- Goals:
 - Application-centric: support unmodified applications
 - Sequential, parallel
 - Batch, interactive
 - Open-source, commercial
 - User-centric: support Grid-unaware users

<http://invigo.acis.ufl.edu>



The In-VIGO portal

In-VIGO Project - ACIS - UFL - Microsoft Internet Explorer

File Edit View Favorites Tools Help Address <https://invigo.acis.ufl.edu/Select?bid=249e4d0d6a8e2b48638ceec52e31a003.2&tab=app> Go

In-VIGO In-Virtual Information Grid Organizations

Virtual-Workspace File Manager FAQ Logout

User: fortes

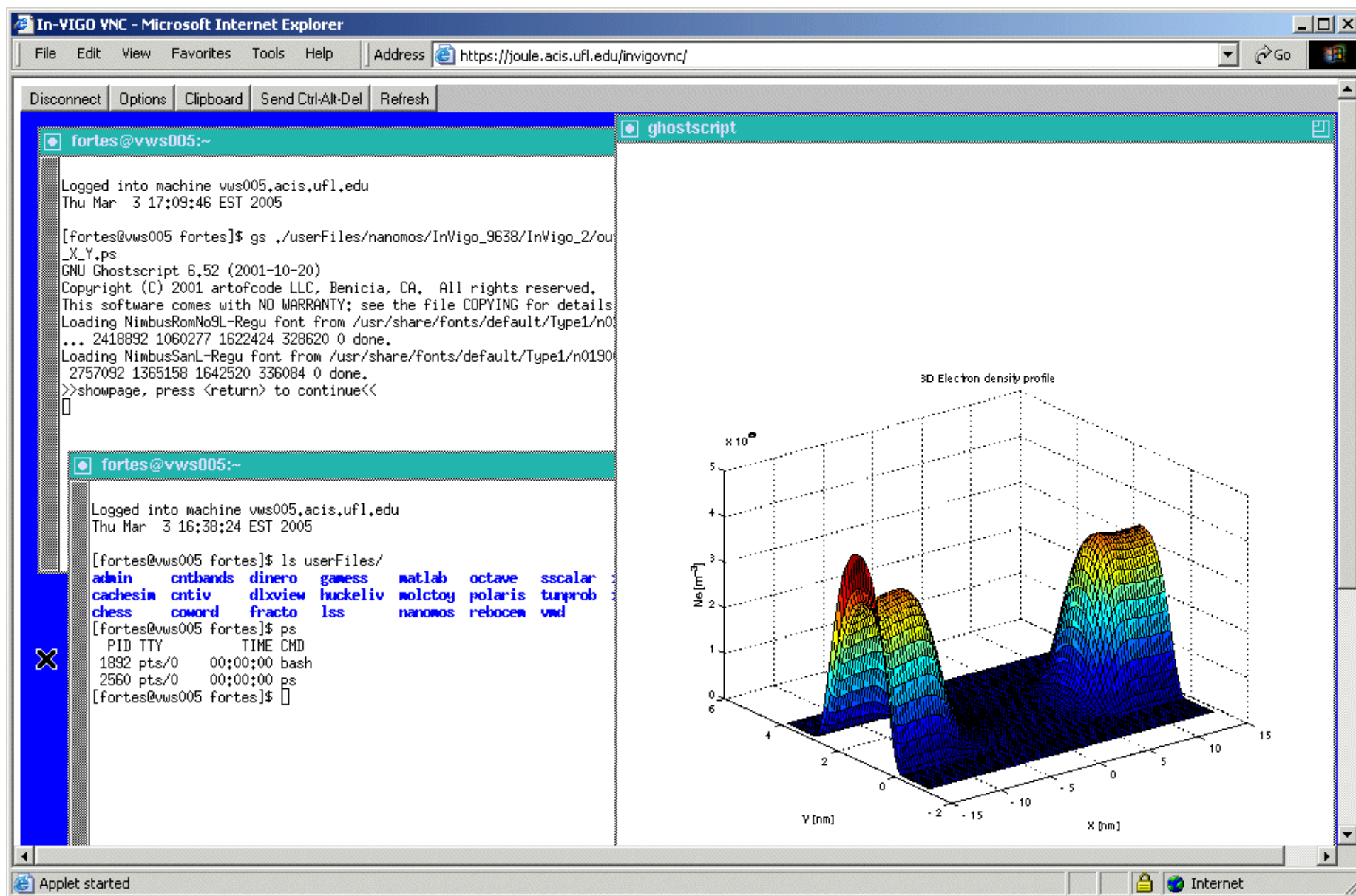
Applications My Sessions My In-VIGO

Scientific Applications

Abinit	<input checked="" type="checkbox"/> regular user	Admintool	<input checked="" type="checkbox"/> admin
BugXSpim	<input checked="" type="checkbox"/> regular user	CacheSim5	<input checked="" type="checkbox"/> regular user
CACTI	<input checked="" type="checkbox"/> regular user	ChaosSim	<input checked="" type="checkbox"/> regular user
Cntbands	<input checked="" type="checkbox"/> regular user	Dinero IV	<input checked="" type="checkbox"/> regular user
DLX-View	<input checked="" type="checkbox"/> regular user	FETToy	<input checked="" type="checkbox"/> regular user
FTMS	<input checked="" type="checkbox"/> regular user	Gamess	<input checked="" type="checkbox"/> regular user
Huckel-IV	<input checked="" type="checkbox"/> regular user	LSS	<input checked="" type="checkbox"/> regular user
Matlab	<input checked="" type="checkbox"/> regular user	MolCToy	<input checked="" type="checkbox"/> regular user
NanoMOS	<input checked="" type="checkbox"/> regular user	Octave	<input checked="" type="checkbox"/> regular user
Polaris	<input checked="" type="checkbox"/> regular user	Rasmol	<input checked="" type="checkbox"/> regular user
REBO-CEM_Dev	<input checked="" type="checkbox"/> reboteam	Schred	<input checked="" type="checkbox"/> regular user

Internet

Virtual workspace



The In-VIGO portal

In-VIGO Project - ACIS - UFL - Microsoft Internet Explorer

File Edit View Favorites Tools Help Address <https://invigo.acis.ufl.edu/Select?bid=249e4d0d6a8e2b48638ceec52e31a003.2&tab=app> Go

In-VIGO In-Virtual Information Grid Organizations

[Virtual-Workspace](#) [File Manager](#) [FAQ](#) [Logout](#)

User: fortes

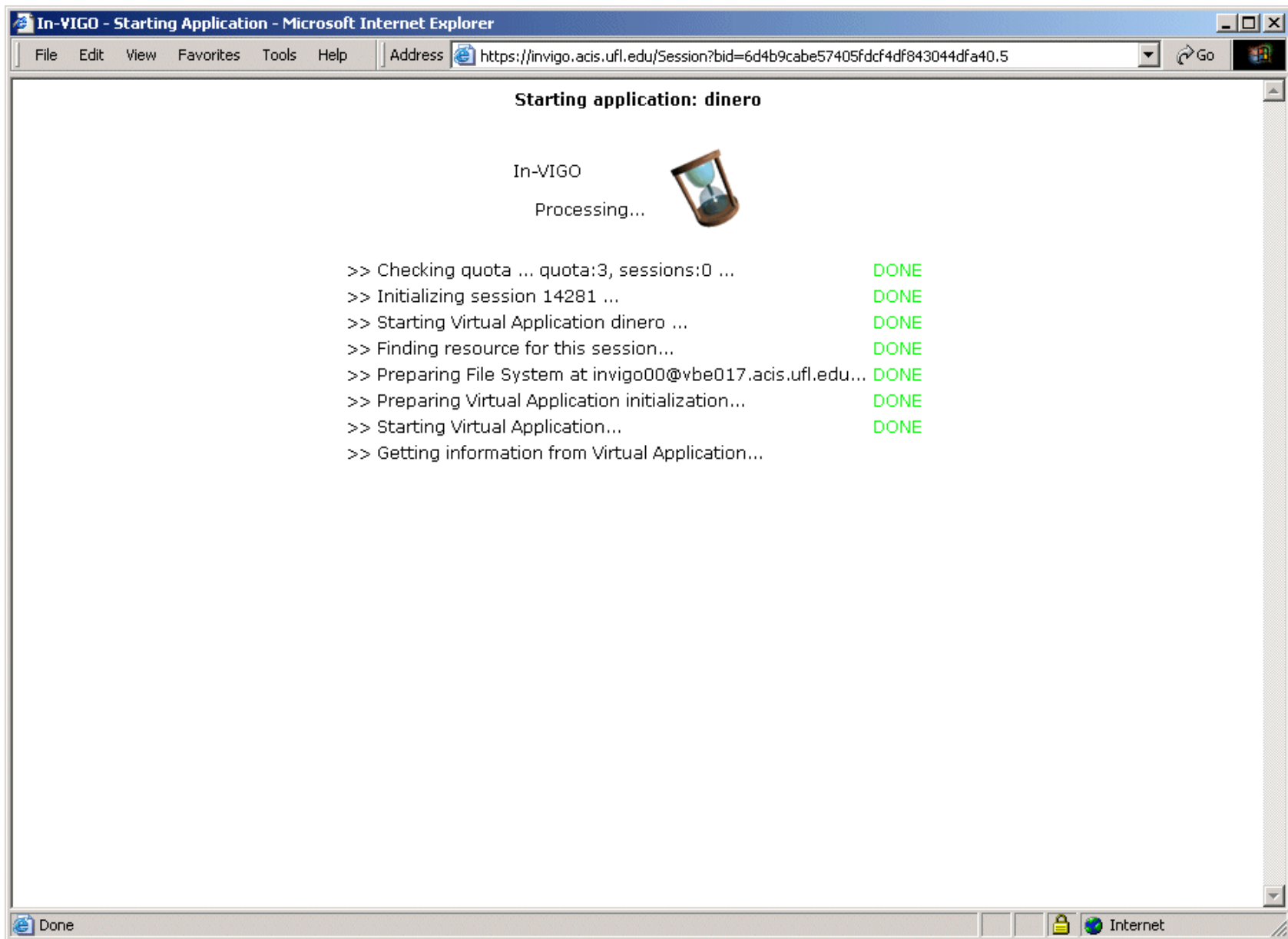
[Applications](#) [My Sessions](#) [My In-VIGO](#)

Scientific Applications

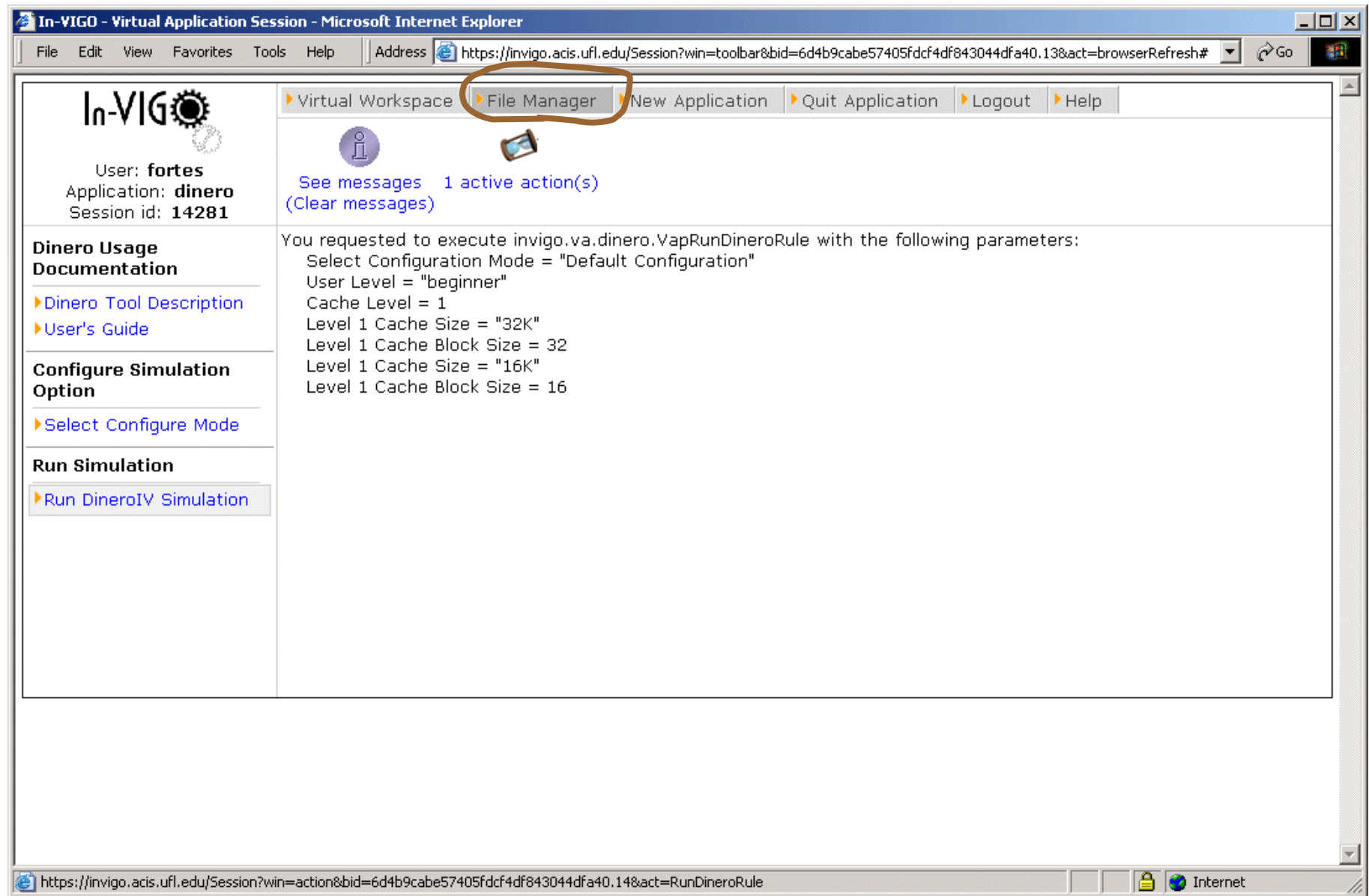
Abit <input checked="" type="checkbox"/> regular user	Admintool <input checked="" type="checkbox"/> admin
BugXSpim <input checked="" type="checkbox"/> regular user	CacheSim5 <input checked="" type="checkbox"/> regular user
CACTI <input checked="" type="checkbox"/> regular user	ChaosSim <input checked="" type="checkbox"/> regular user
Cntbands <input checked="" type="checkbox"/> regular user	Dinero IV <input checked="" type="checkbox"/> regular user
DLX-View <input checked="" type="checkbox"/> regular user	FETToy <input checked="" type="checkbox"/> regular user
FTMS <input checked="" type="checkbox"/> regular user	Gamess <input checked="" type="checkbox"/> regular user
Huckel-IV <input checked="" type="checkbox"/> regular user	LSS <input checked="" type="checkbox"/> regular user
Matlab <input checked="" type="checkbox"/> regular user	MolCToy <input checked="" type="checkbox"/> regular user
NanoMOS <input checked="" type="checkbox"/> regular user	Octave <input checked="" type="checkbox"/> regular user
Polaris <input checked="" type="checkbox"/> regular user	Rasmol <input checked="" type="checkbox"/> regular user
REBO-CEM_Dev <input checked="" type="checkbox"/> reboteam	Schred <input checked="" type="checkbox"/> regular user

Internet

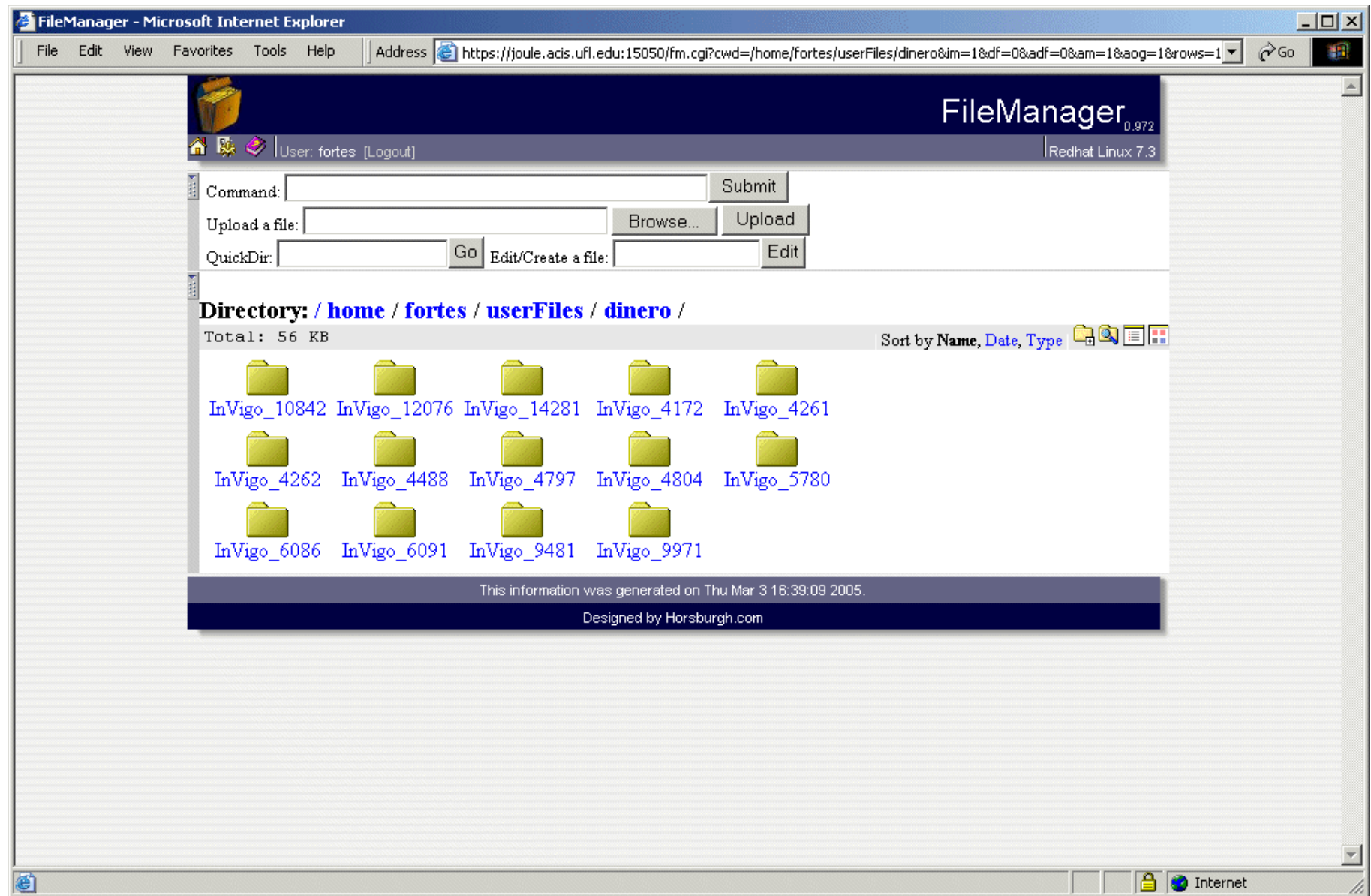
Setting up ...



Interface and workflow



File manager (1)



File manager (2)

FileManager - Microsoft Internet Explorer

File Edit View Favorites Tools Help Address https://joule.acis.ufl.edu:15050/fm.cgi?cwd=/home/fortes/userFiles/dinero/InVigo_14281/InVigo_1&cmd=_view%20%2 Go

FileManager 0.972

User: fortes [Logout] Redhat Linux 7.3

Command Result

---Simulation complete.

l1-icache

Metrics	Total	Instrn	Data	Read
Demand Fetches	757341	757341	0	0
Fraction of total	1.0000	1.0000	0.0000	0.0000
Demand Misses	12731	12731	0	0
Demand miss rate	0.0168	0.0168	0.0000	0.0000
Multi-block refs	0			
Bytes From Memory	407392			
(/ Demand Fetches)	0.5379			
Bytes To Memory	0			


Command: Submit


Upload a file: Browse... Upload


QuickDir: Go Edit/Create a file: Edit


Directory: / home / fortes / userFiles / dinero / InVigo_14281 / InVigo_1 /

Total: 2.52 KB Sort by Name, Date, Type


In-VIGO.README


runCommand


runResult.out


runSTD.err

This information was generated on Thu Mar 3 16:43:24 2005.

Designed by Horsburgh.com

Internet

The In-VIGO portal

The screenshot shows the In-VIGO portal interface within a Microsoft Internet Explorer browser window. The browser's address bar displays the URL: `https://invigo.acis.ufl.edu/Select?bid=249e4d0d6a8e2b48638ceec52e31a003.2&tab=app`. The page title is "In-VIGO Project - ACIS - UFL - Microsoft Internet Explorer".

The portal header includes the In-VIGO logo and the text "In-Virtual Information Grid Organizations". Navigation links for "Virtual-Workspace", "File Manager", "FAQ", and "Logout" are located in the top right corner. The user is logged in as "User: fortes".

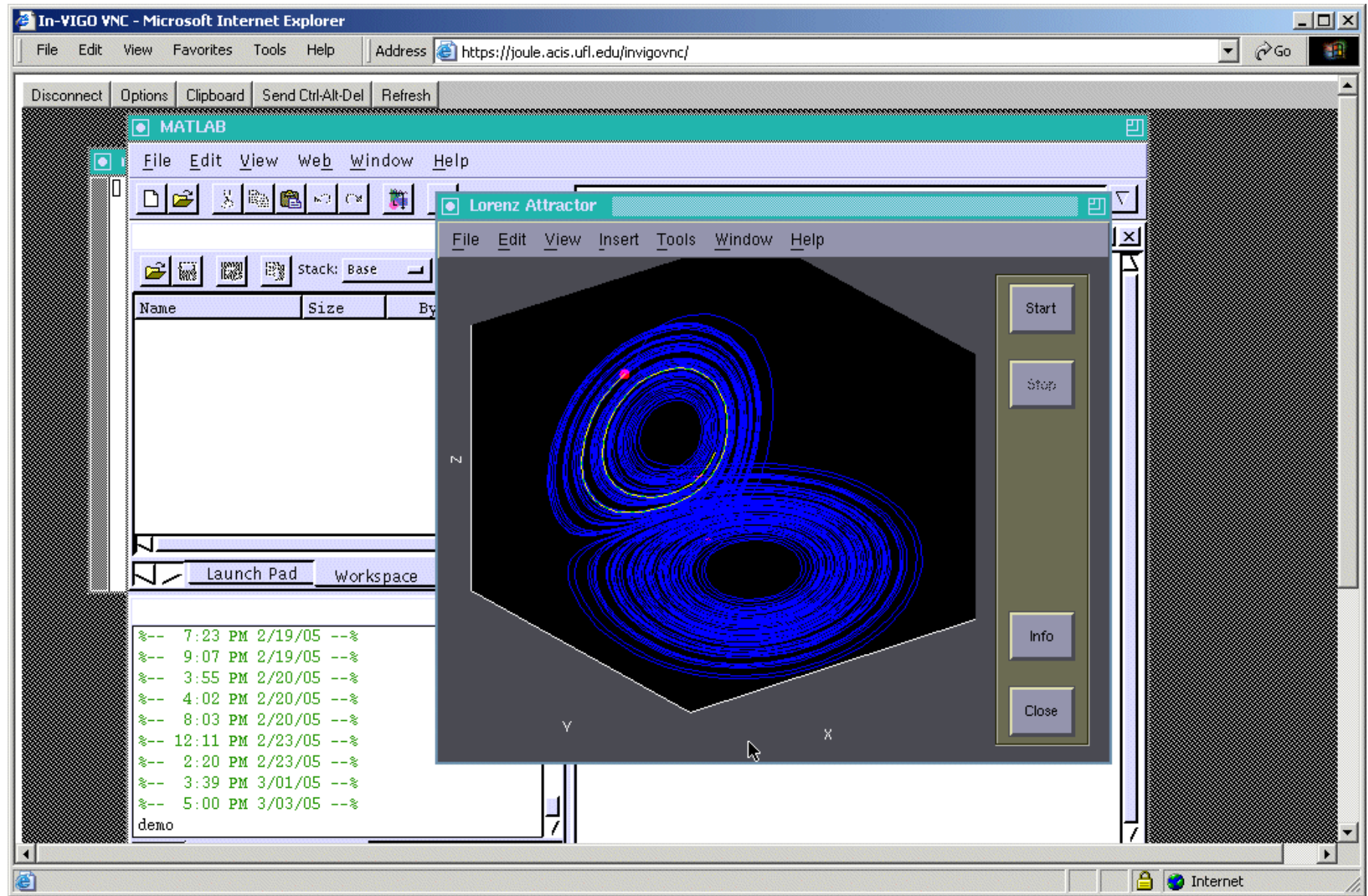
Below the user information, there are three tabs: "Applications", "My Sessions", and "My In-VIGO". The "Applications" tab is currently selected.

The "Scientific Applications" section displays a list of applications, each with a button and a checkbox indicating the user's status:

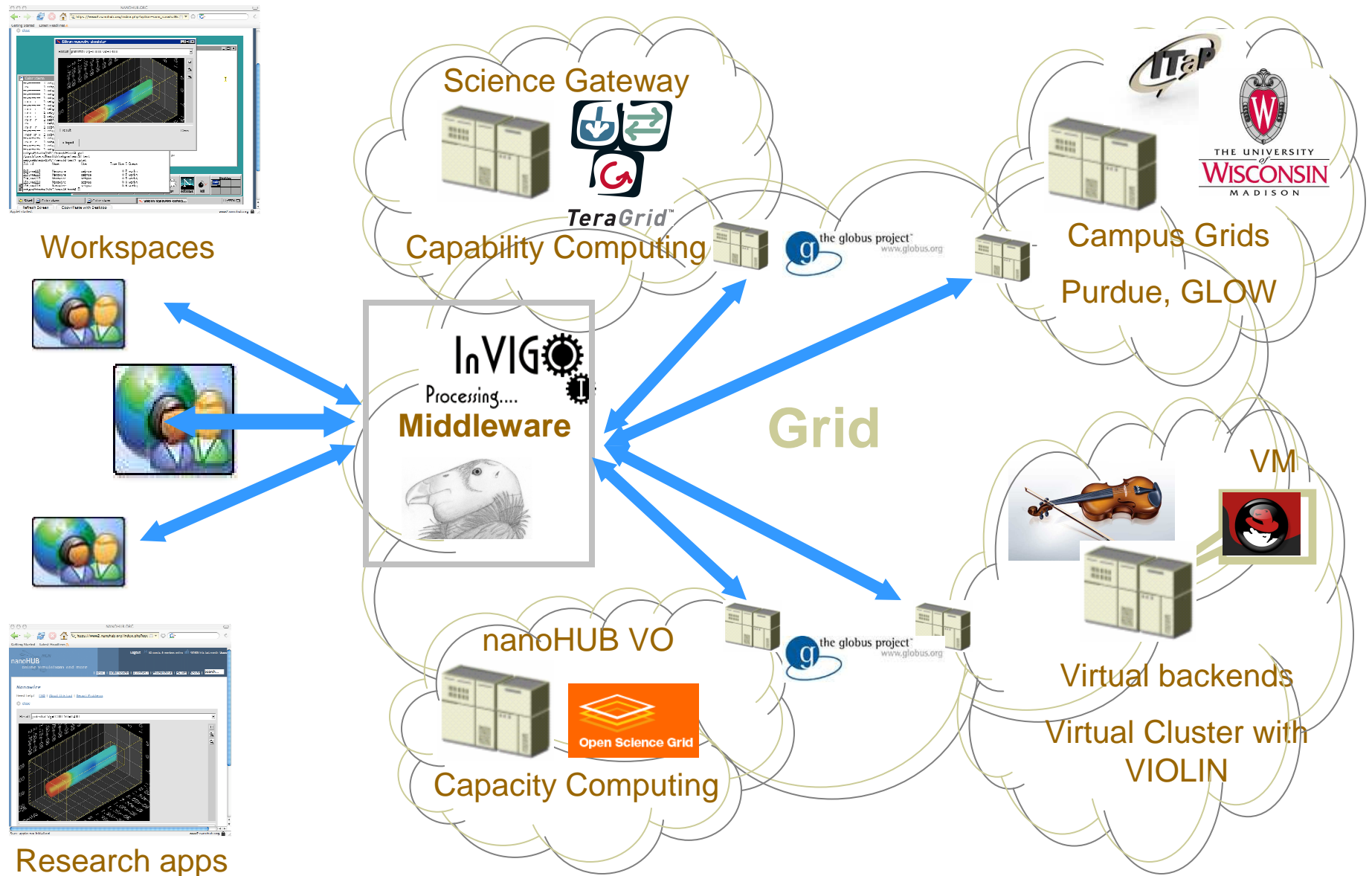
Application	User Status
Abinit	<input checked="" type="checkbox"/> regular user
BugXSpim	<input checked="" type="checkbox"/> regular user
CACTI	<input checked="" type="checkbox"/> regular user
Cntbands	<input checked="" type="checkbox"/> regular user
DLX-View	<input checked="" type="checkbox"/> regular user
FTMS	<input checked="" type="checkbox"/> regular user
Huckel-IV	<input checked="" type="checkbox"/> regular user
Matlab	<input checked="" type="checkbox"/> regular user
NanoMOS	<input checked="" type="checkbox"/> regular user
Polaris	<input checked="" type="checkbox"/> regular user
REBO-CEM_Dev	<input checked="" type="checkbox"/> reboteam
Admintool	<input checked="" type="checkbox"/> admin
CacheSim5	<input checked="" type="checkbox"/> regular user
ChaosSim	<input checked="" type="checkbox"/> regular user
Dinero IV	<input checked="" type="checkbox"/> regular user
FETToy	<input checked="" type="checkbox"/> regular user
Gamess	<input checked="" type="checkbox"/> regular user
LSS	<input checked="" type="checkbox"/> regular user
MolCToy	<input checked="" type="checkbox"/> regular user
Octave	<input checked="" type="checkbox"/> regular user
Rasmol	<input checked="" type="checkbox"/> regular user
Schred	<input checked="" type="checkbox"/> regular user

The "Matlab" application button is highlighted with a red oval.

Native interactive interface



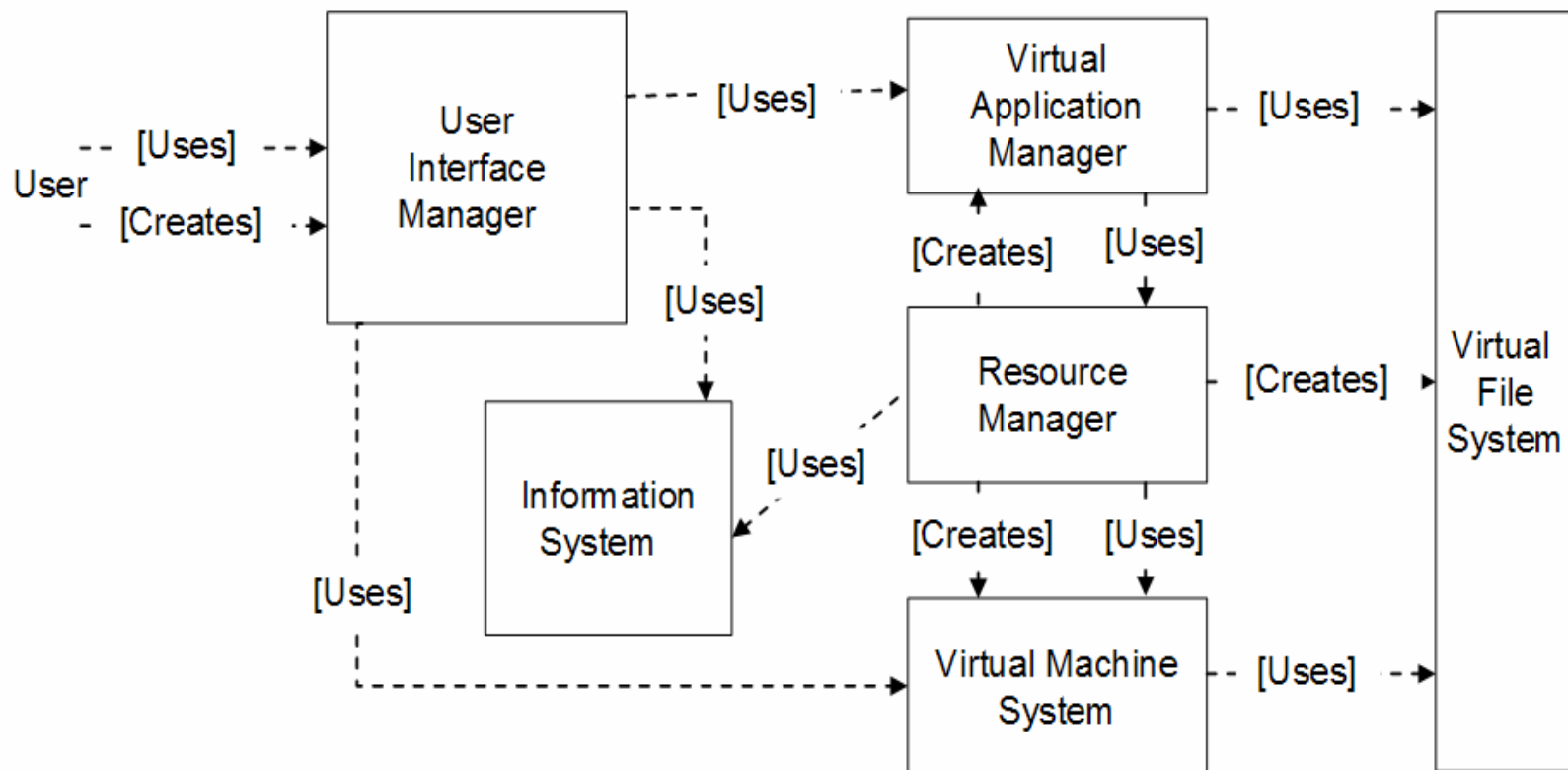
nanoHUB (current middleware infrastructure)



Slide provided by Sebastien Goasguen

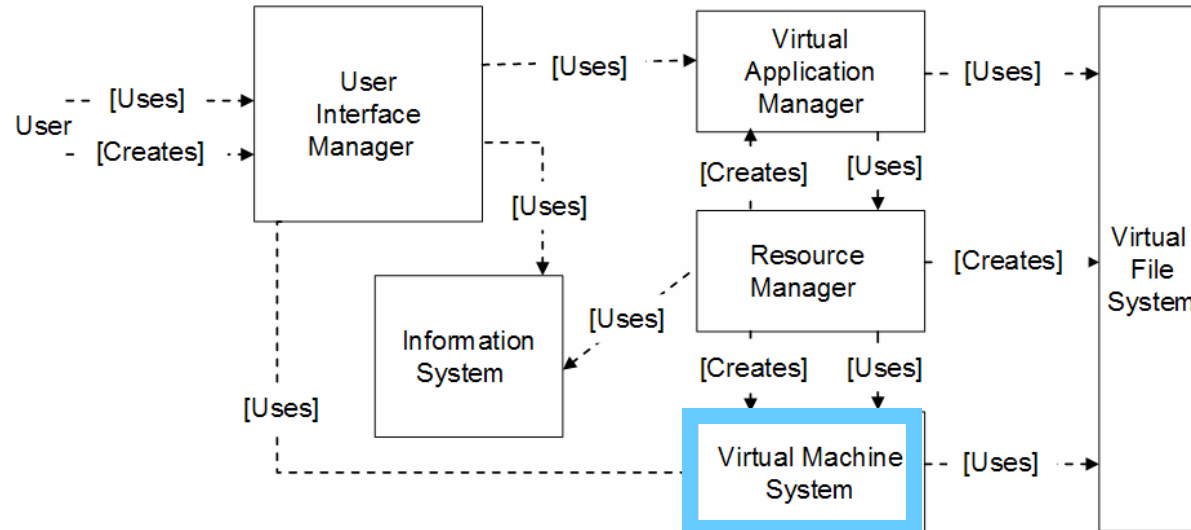
In-VIGO 1.0 architecture diagram

- Deployed at UF/ACIS since Fall 2003
- nanoHUB: Summer 2005
- On-going deployments: SURA/SCOOP, UF/HPC



Virtual Machine System

- *Enables on-demand instantiation of whole-O/S VMs for virtual workspaces*
- Has access to physical resources (host)
- Create, configure, query, destroy VMs
- In-VIGO users have access to virtual resource (guest)

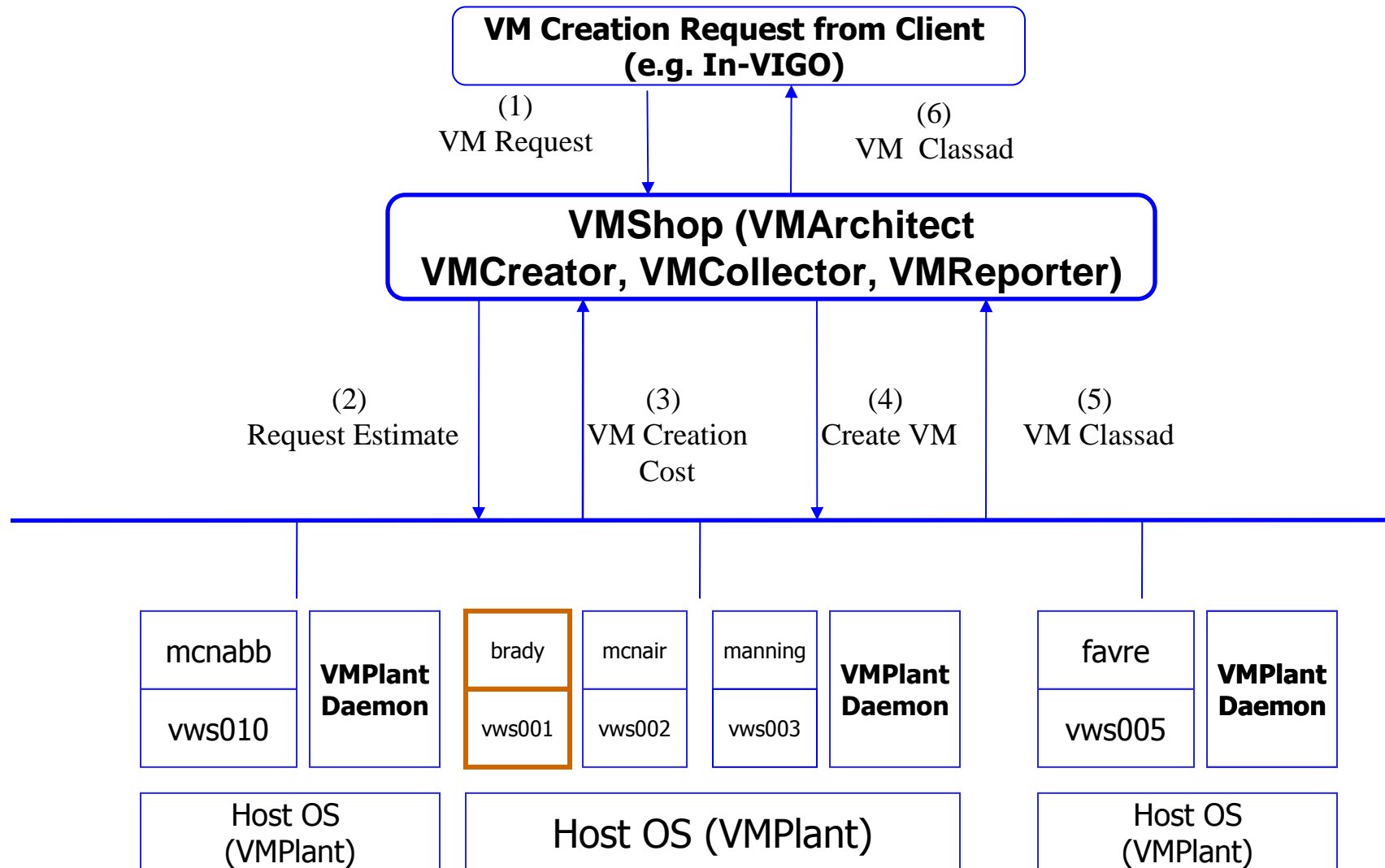


VM services

- Provide means to efficiently create/configure/destroy VMs,
 - generic across VM technologies
- Directed Acyclic Graph (DAG) model for defining application-centric VMs
- Cost-bidding model for choosing compute servers for VM instantiation

(SC 2004)

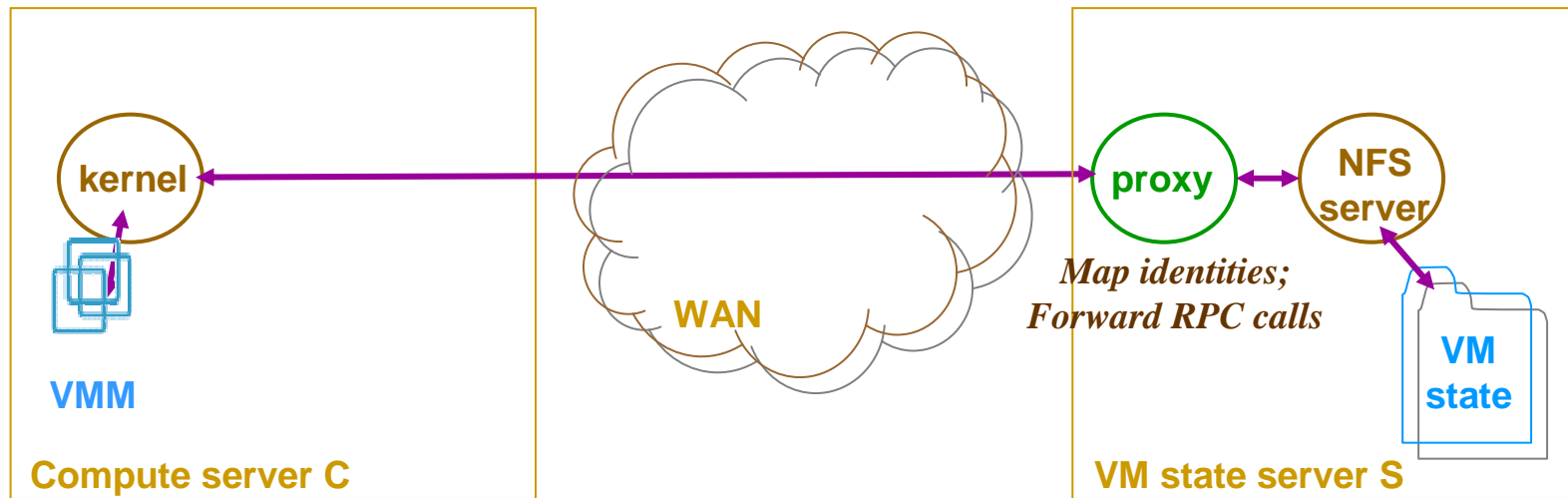
Architectural Components of VM Service



Data access virtualization

- Grid virtual file systems (GVFS)
 - On-demand setup, configuration and tear-down of distributed file systems
 - Unmodified applications access file-based data in the same manner they would in a local environment
 - Use and extend Network File Systems (NFS)
 - *Multiple, independent* file system sessions *share* one or more accounts in file servers
 - File system data is transferred on-demand, on a per-block basis

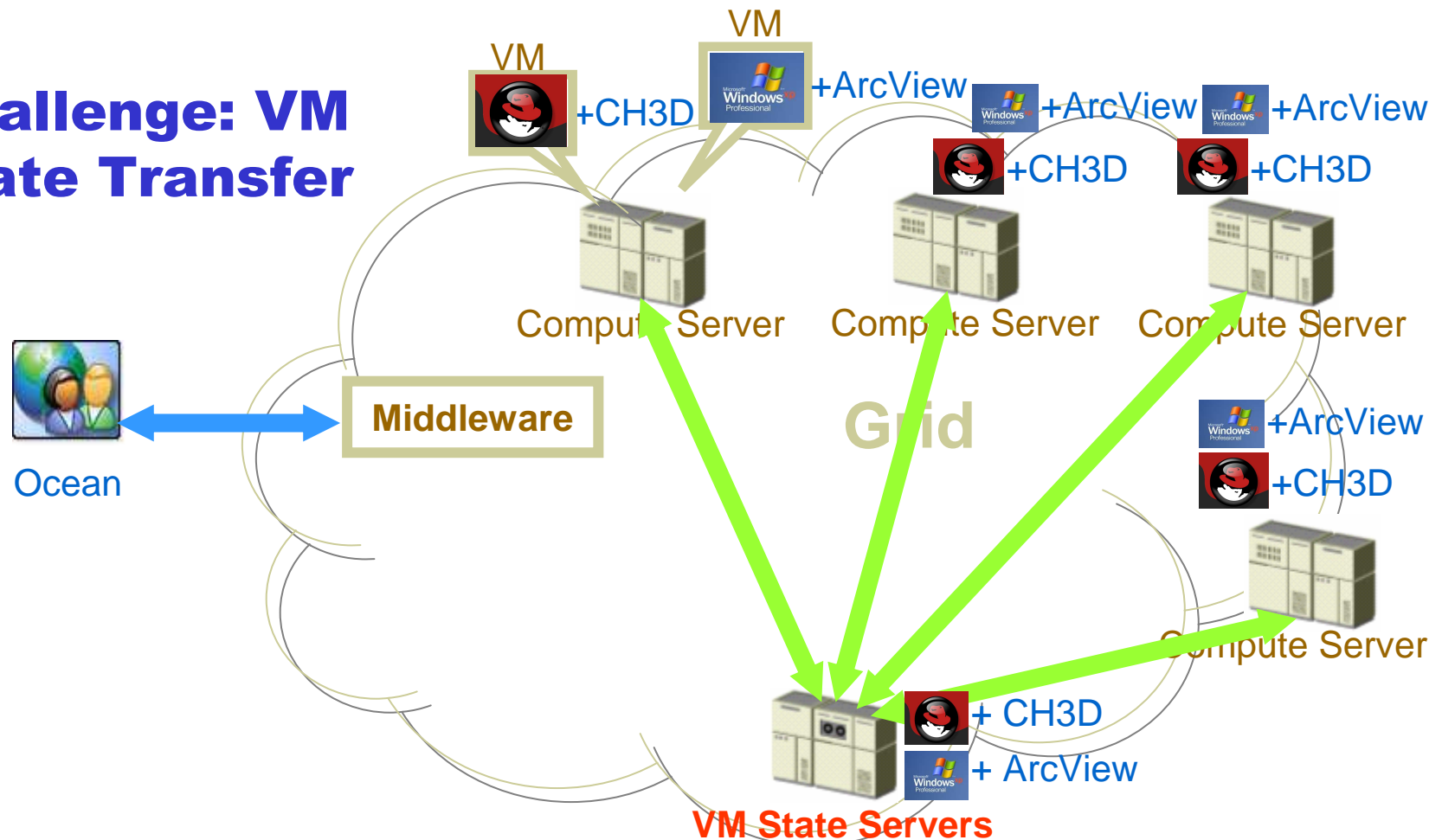
Grid Virtual File System (GVFS)



- Logical user accounts [HCW'01] and virtual file system [HPDC'01]
 - Shadow account + file account, managed by middleware
 - NFS call forwarding via middle tier **user-level** proxy
 - User identities mapped by proxy
- Provides access to user data, VM images

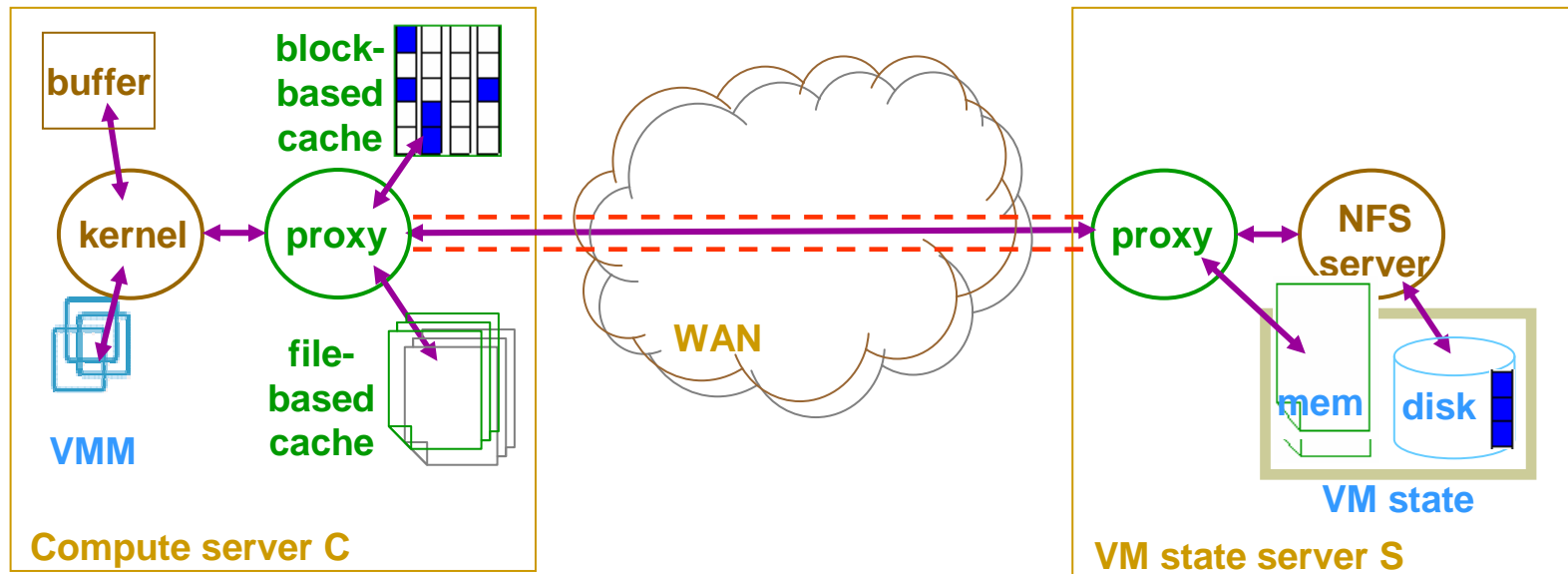
Many users, apps and environments

Challenge: VM State Transfer



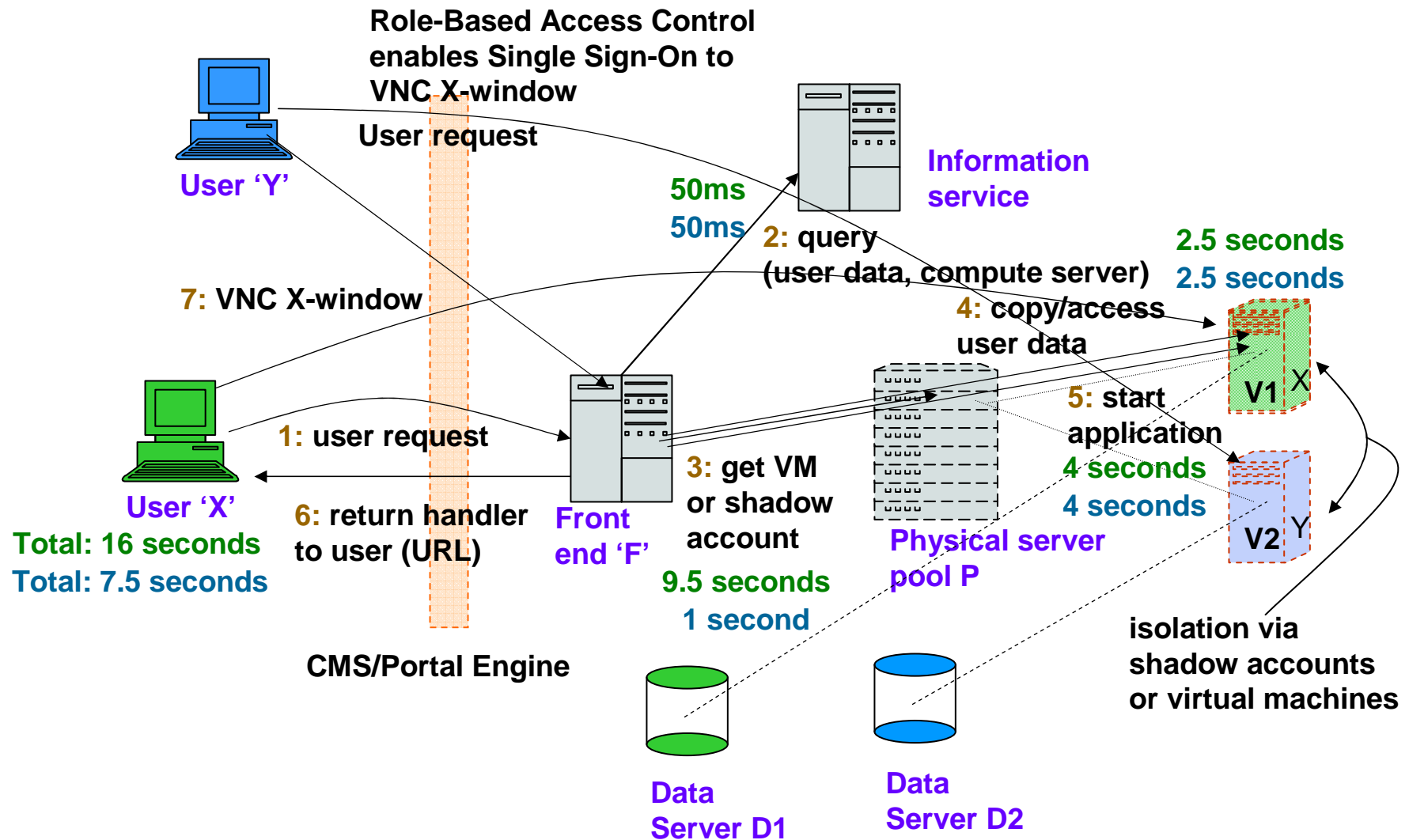
Dynamic, efficient transfer of large VM state is important

User-level Extensions



- Client-side proxy disk caching
- Application-specific meta-data handling
- Encrypted file system channels and cross-domain authentication
- *[Zhao, Zhang, Figueiredo, HPDC'04]*

Putting it all together: GUI Application example



Virtual network services

- VMShop allocates a remote VM
 - Great; now how to access it?
 - Need to isolate traffic from host site
 - For most flexibility, need full TCP/IP connectivity
- ViNe, IPOP being developed at UF ACIS
 - Related work: Virtuoso/VNET (NWU), Violin (Purdue)

In-VIGO Virtual Networks - ViNe

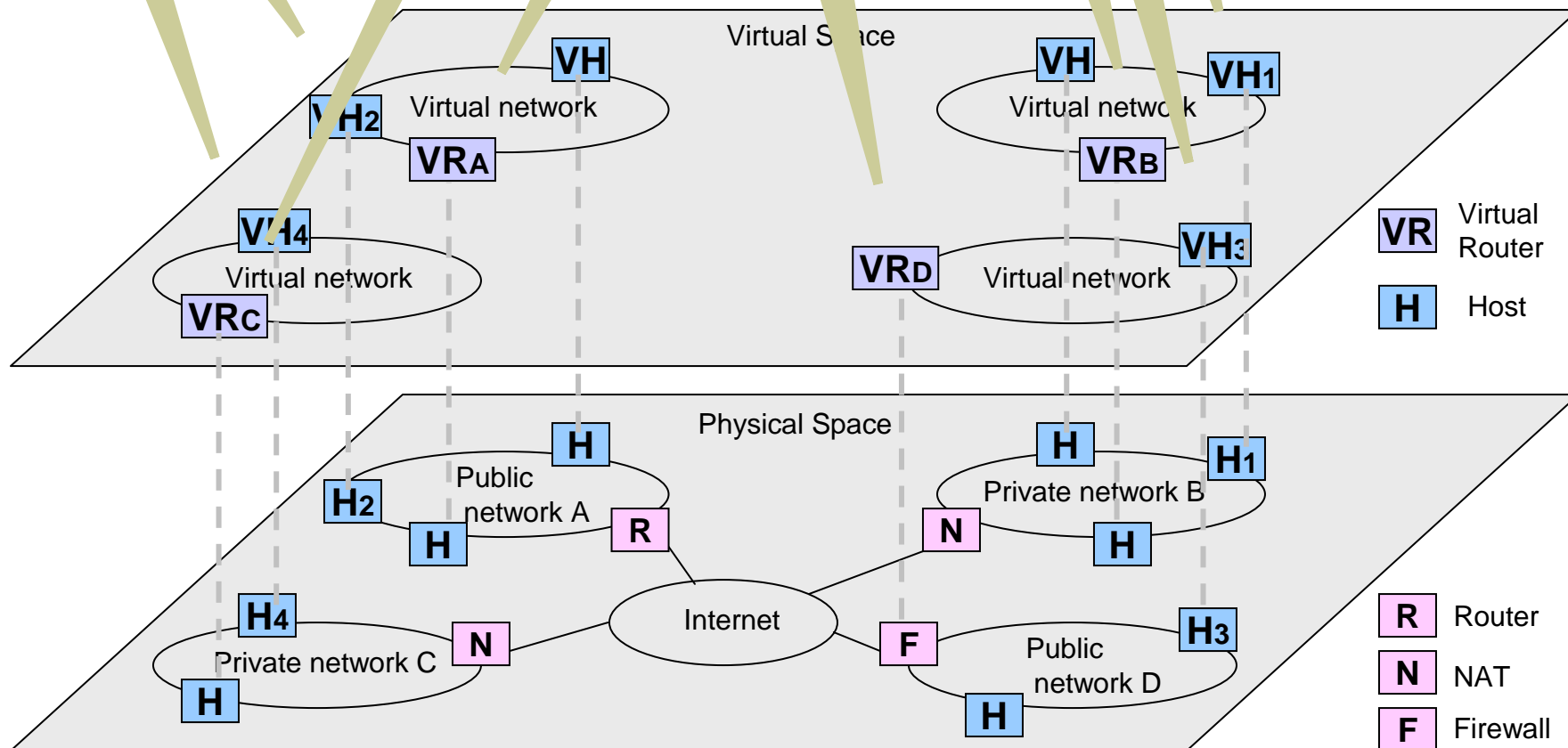
- IP overlay on top of the Internet
- Operation similar to site-to-site VPN
- Designed to address issues that VPN does not solve:
 - High administrative overhead for many sites
 - VPN firewalls need a static public IP address

In-VIGO Virtual Networks - ViNe

- Each participating host is configured with an additional IP address in ViNe space (IP aliasing)
- Packets with destination in ViNe space are directed to VRs for routing in ViNe space.

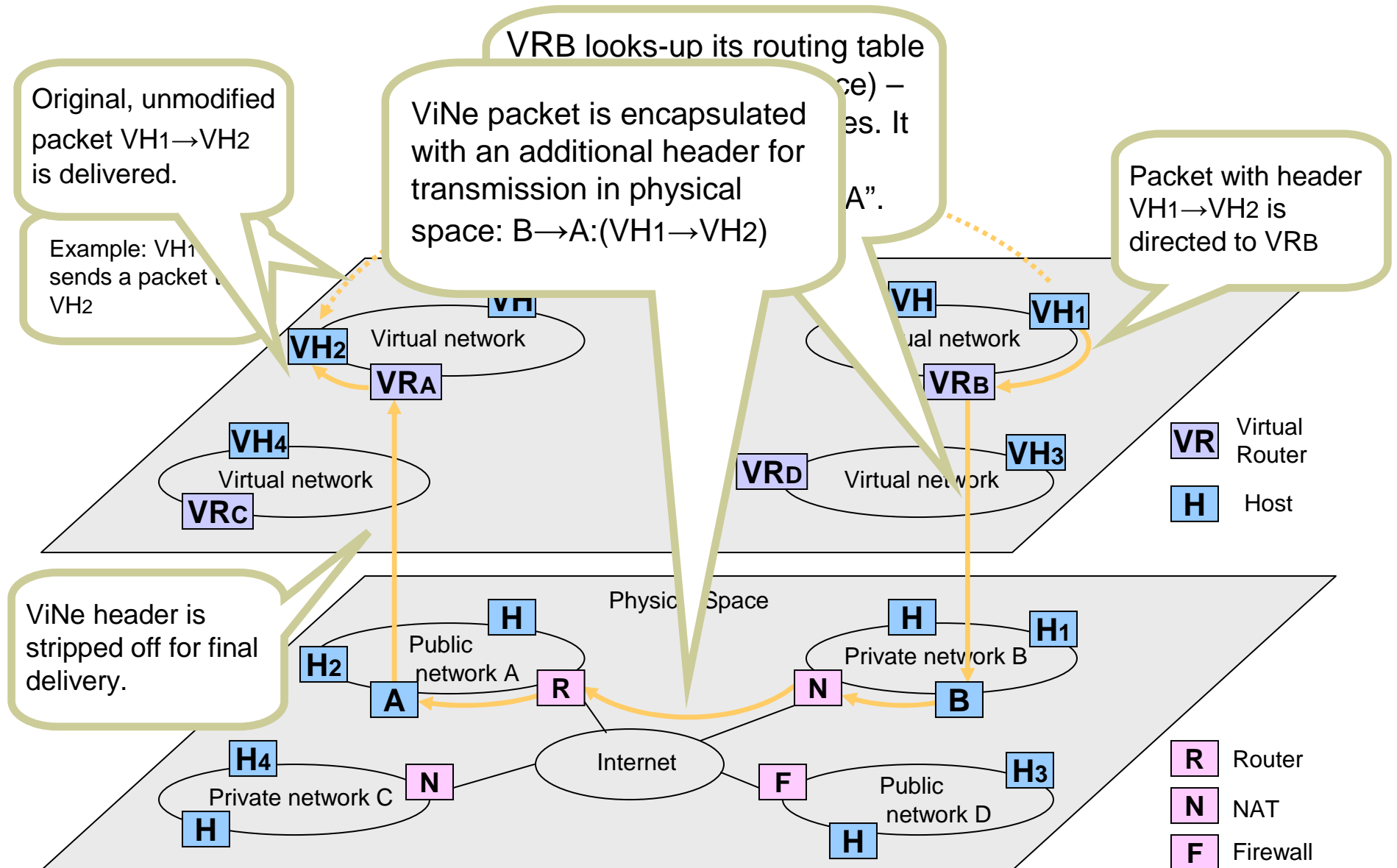
to route (tunnel)

configured with
IP private space)



Slide provided by M. Tsugawa

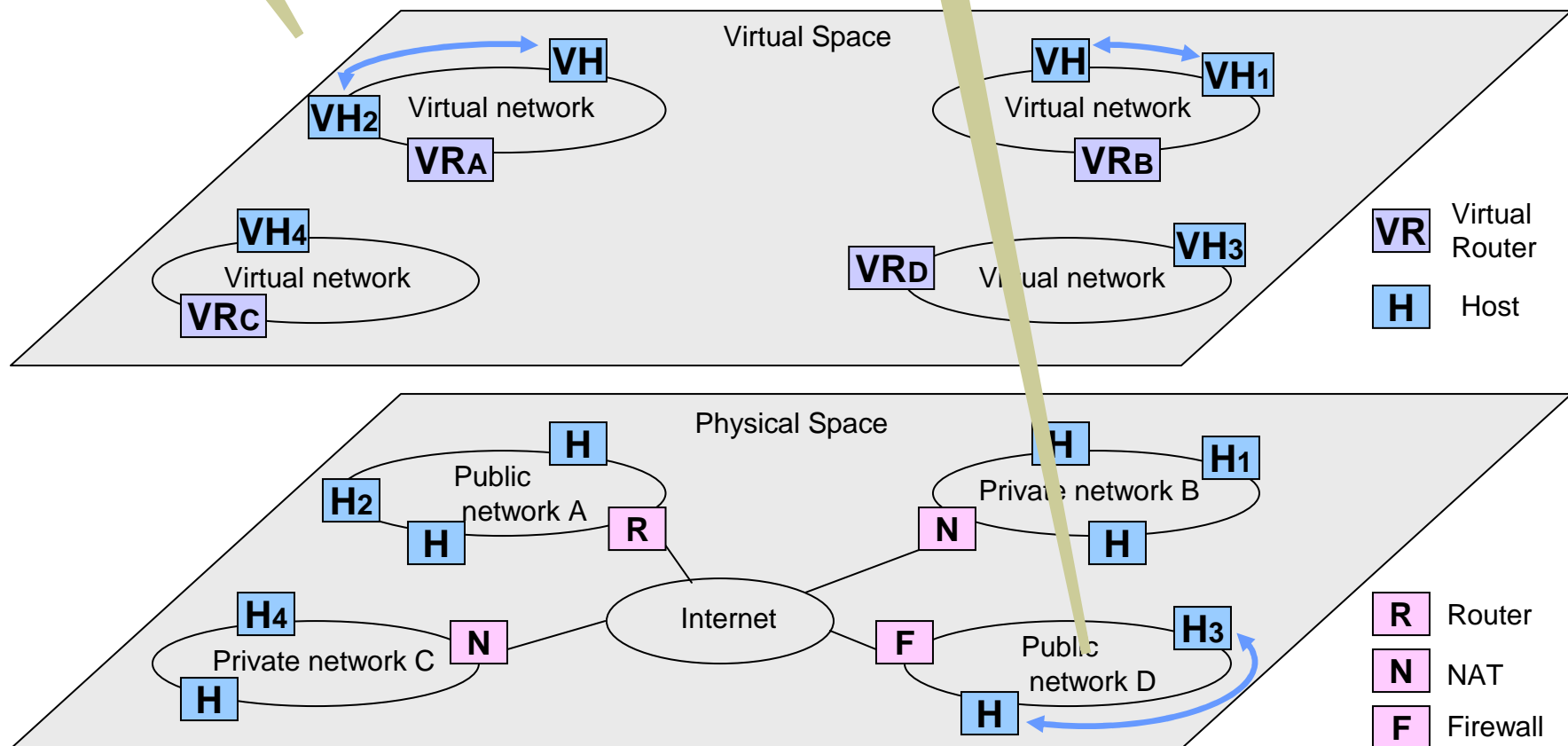
ViNe – Communication in virtual space



Slide provided by M. Tsugawa

ViNe – Local communication

- Local communication is kept local in both Physical and Virtual space.
- ViNe does not interfere with physical communication.
- Virtual space can be used only when needed.



Slide provided by M. Tsugawa

ViNe Firewall/NAT traversal

- VRs connected to the public network proxy (queue) packets to VRs with limited connectivity. The latter open connection to the queue VR to retrieve packets.
- VRs with limited connectivity are not used when composing routing tables. Routing tables are made to direct packets to queue VRs.
- The approach supports multi-level NAT.
- The approach also works under DHCP since the changing IP is not considered for routing.

ViNe organization

- Routing tables are created/destroyed as needed (e.g., join/leave of sites, creation of a new ViNe, etc).
- VRs exchange routing information with each other
- Communication of sensitive information (e.g., routing tables, VRs host certificates) is encrypted.
- Administrator of a participating site is involved only during the setup/configuration phase. No intervention is needed when machines join/leave network.

Slide provided by M. Tsugawa

ViNe Overhead

- When firewall/NAT traversal is not required
 - depends on performance of VRs and available physical network
 - Overhead = 0 – 5% of available bandwidth
 - up to 150 Mbps for VR on 3 GHz Xeon
- When firewall/NAT traversal is required
 - also depends on the allocation of VRs to proxy/queue traffic
 - 10 – 50% in initial experiments. Optimizations under investigation.

Slide provided by M. Tsugawa

ViNe Security

- Site-related
 - security policies are not changed by enabling ViNe
 - minimal change may be needed to allow ViNe traffic in private IP space
 - ViNe traffic consists of IP packets that are visible in LANs (tunneling is only used across domains)
 - Network policies can be applied to ViNe traffic
 - Firewalls can inspect ViNe traffic
 - Intrusion detection systems and monitoring works unmodified
- ViNe-related
 - ViNe routers do not route packets to/from the Internet
 - All communication between VRs are authenticated
 - Sensitive VR messages are encrypted
 - VRs are not accessible in ViNe space
- ViNe connects hosts without links in physical IP infrastructure
 - But it does so only where we want to have it

Slide provided by M. Tsugawa

ViNe: On-going work

- Management of Virtual Networks
 - Automated and secure management (definition, deployment, tear-down, merge, split and join/leave of hosts) of virtual networks is under development in the context of ViNe project
 - The idea is to dynamically and securely and reconfigure ViNe routers in response to client (privileged users, local site administrators, grid administrators, grid middleware) requests
 - In collaboration with ANL

Slide provided by M. Tsugawa

ViNe: Auditability

- ViNe does not modify packets generated by participating hosts
- Regular network traffic inspection can be performed in each participating site
- In addition, ViNe Routers can log all routed traffic (performance implications are under investigation)
 - Side-process can combine traffic logs for global network traffic analysis

Slide provided by M. Tsugawa

IPOP virtual network

- Motivations:
 - Enable self-configuring virtual networks – focus on making it simple for individual nodes to join and leave
 - Decentralized traversal of NATs and firewalls
- Approach: IP-over-P2P
 - Overhead of adding a new node is constant and independent of size of the network
 - Peer to peer routing
 - Self-organizing routing tables
 - Ring topology with shortcuts
 - N nodes, k edges per node; $O(1/k \log^2(M))$ routing hops
 - Adaptive, 1-hop shortcuts based on traffic inspection
 - Mobility: same IP even if VM migrates across domains

A. Ganguly, A. Agrawal, P. O. Boykin, R. Figueiredo – IPDPS 2006, HPDC 2006

Slide provided by R. Figueiredo

Applications

- Distributed computing VM “appliances”:
 - Define once, instantiate many
 - Homogeneous software configuration and private network address spaces
 - Facilitates a model where resources are pooled by the various users of a community (e.g. nanoHUB)
 - Homogeneous configuration facilitates deployment of security infrastructures (e.g. X.509-based IPsec host authentication)

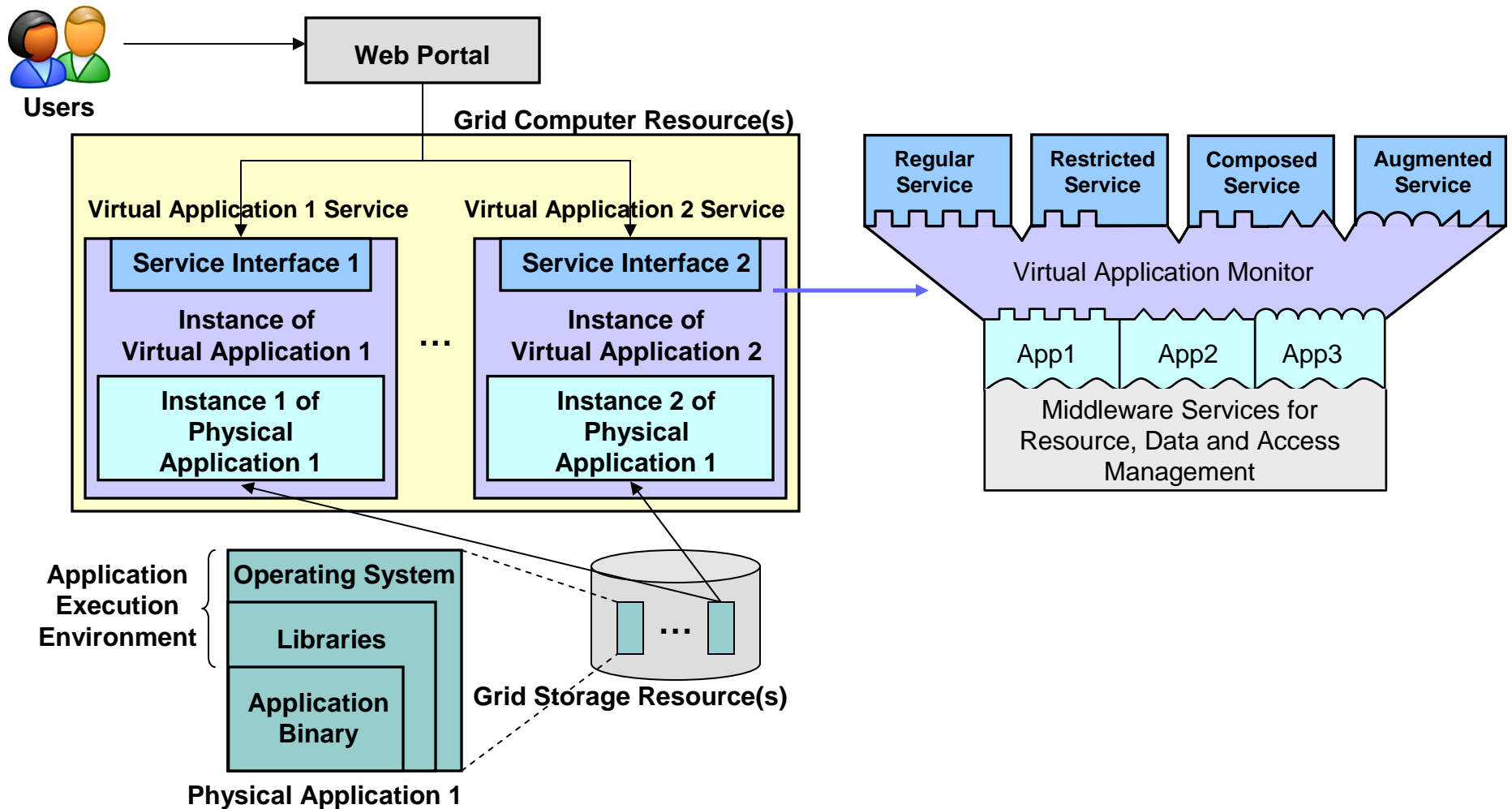
Slide provided by R. Figueiredo

Usage examples

- “Grid appliance”
 - Condor node for job submission/execution
 - Automatically obtains a virtual address from virtualized DHCP server and joins a pool
 - Can submit and flock jobs within virtual network
 - Download VMware player and VM image:
http://www.acis.ufl.edu/~ipop/grid_appliance
- On-going: domain-specific customizations
 - nanoHUB: WebDAV client, Rappture GUI toolkit
 - SCOOP (coastal ocean modeling): clients to access data catalog and archive
 - Archer (computer architecture): support for large, read-only checkpoints and input files

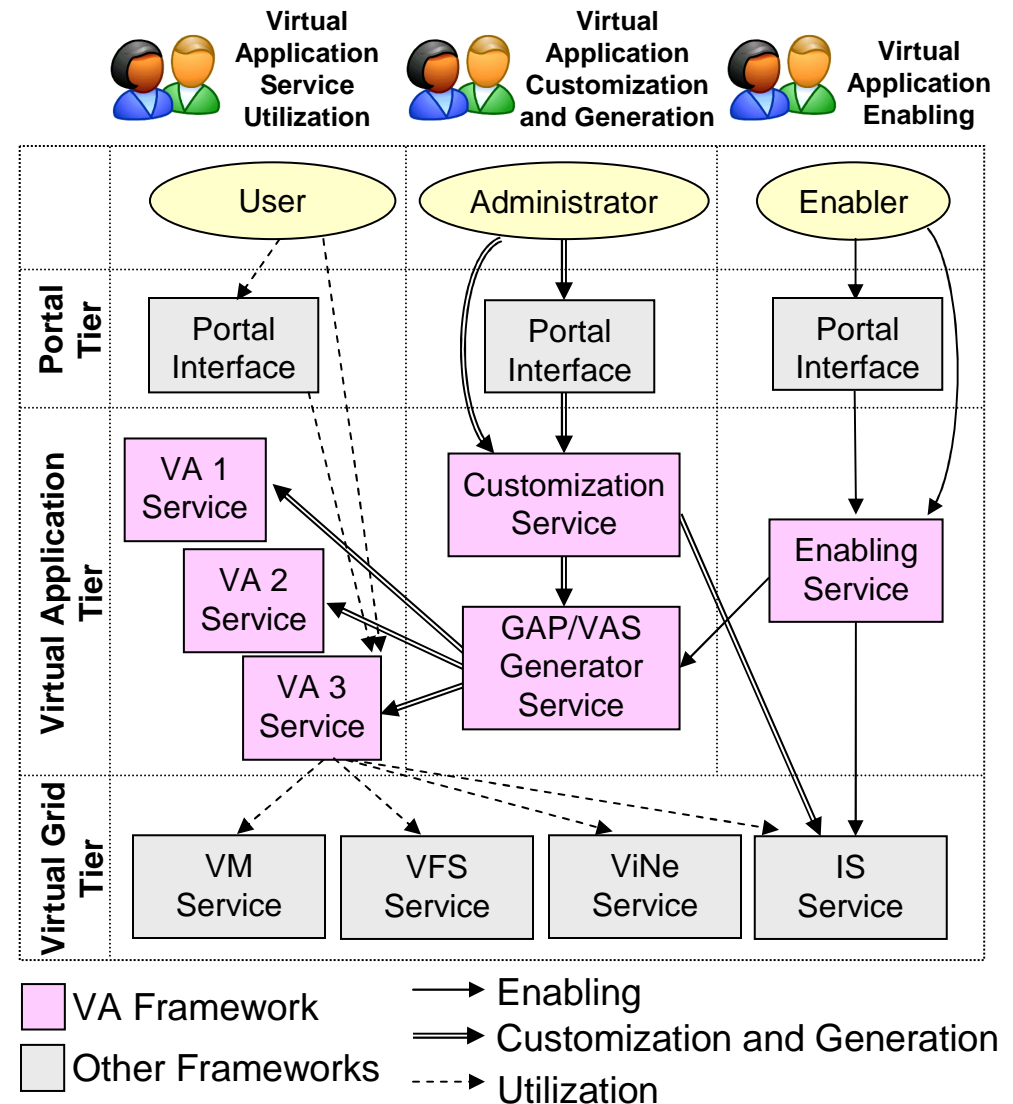
Slide provided by R. Figueiredo

Application virtualization



Grid-enabling unmodified applications

- Enabler provides
 - Command-line syntax
 - Application-related labels
 - Parameter(s), type-set values, entire applications
 - Resource and execution environment metadata
 - Architecture, OS libraries, environment variables
- Grid-services created, deployed and possibly customized using
 - Generic Application Service (GAP)
 - Virtual Application Service (VAS)
- Grid-user interacts with the virtual application through a Web-portal to execute applications on virtualized resources



Summary and conclusions

- Virtualization technology decouples physical resource constraints from user and application requirements
 - Big win, novel rethinking
 - Virtual resources are to grid computing what processes are to operating systems
 - Developers can concentrate on applications, not end resources
- Web-services provide interoperability and a framework for composition and aggregation of applications
 - Includes delivering virtuals and virtualizing applications
 - Wide adoption creates large reusable toolboxes, e.g. for automatic interface generation
 - Users need only know of service interfaces
- *In-VIGO middleware effectively integrates virtualization and Web-services technologies to easily enable and deliver applications as Grid-services*

Acknowledgments

- **Collaborators**

- In-VIGO team at UFL
 - <http://www.acis.ufl.edu/invigo>
- Rob Carpenter and Mazin Yousif at Intel
- Peter Dinda and Virtuoso team at NWU
 - <http://virtuoso.cs.northwestern.edu>
- NCN/NanoHub team at Purdue University
 - <http://www.nanohub.org>
- Kate Keahey, ANL

- **Funding**

- NSF
 - Middleware Initiative and Research Resources Program
 - DDDAS Program
- Army Research Office
- IBM Shared University Research
- Intel
- VMWare
- Northrop-Grumman