

# *Science gateways made easy: the In-VIGO approach*

A. Matsunaga, M. Tsugawa, S. Adabala, R. Figueiredo, H. Lam  
and J. Fortes

**Advanced Computing and Information Systems  
Laboratory**

**University of Florida**

June 28, 2005 – GGF 14 – Chicago, IL, USA

Workshop: Science Gateways: Common Community Interfaces to Grid Resources

# What is In-VIGO?

- Enables computational engineering and science
- In-Virtual Information Grid Organizations

The screenshots illustrate the In-VIGO system's components and user interface:

- Welcome to In-VIGO:** A login page with fields for Username (guest) and Password (\*\*\*\*\*).
- In-VIGO Project - ACIS - UFL:** A page listing various applications available to the user 'guest'. The application 'MolCToy' is highlighted with a red circle.
- In-VIGO - Virtual Application Session:** A window showing the 'MolCToy -- Molecular Conduction: Toy Models' interface. It includes a 'File Manager' button, a 'New Application' button, and a 'Simulate' button. The simulation parameters are set to: Model: 1L-D (One level, discrete), Ambient temperature: 300K, Single-electron charging energy: 1eV, Coupling factor (left contact): 0.1, Coupling factor (right contact): 0.1, Voltage Sweep +/-: 4V.
- In-VIGO VNC:** A window showing a molecular model of a molecule connected to two contacts.
- Index of /molctoy/InVigo\_1545/InVigo\_1/:** A directory listing showing files and folders, including 'Parent directory', 'Vnc', 'Xauthority', and several files with timestamps.

# Motivations and goals for science gateways

- Motivations:

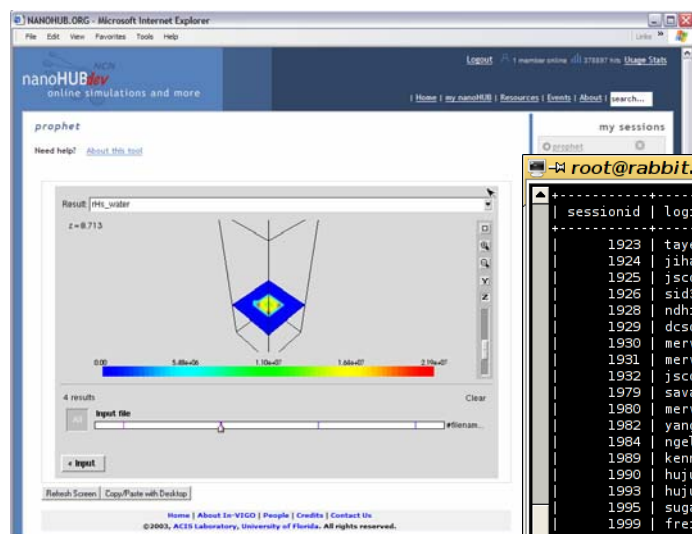
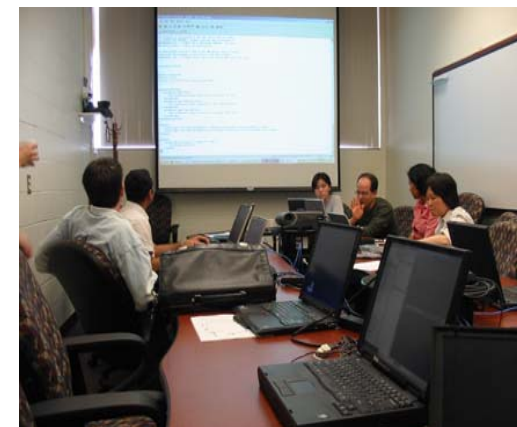
- Need middleware to hide complexity of dealing with cross-domain issues
  - From application developers
  - From end users
- While preserving security and privacy of data, codes and other users' information

- Goals:

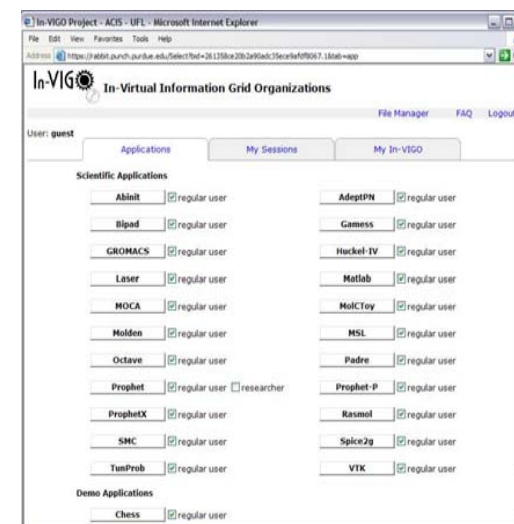
- Application-centric:
  - Support unmodified applications
    - Sequential, parallel
    - Batch, interactive
    - Open-source, commercial
  - Support automated Grid-enabling of applications
- User-centric: support Grid-unaware users

# What does In-VIGO bring to science gateways?

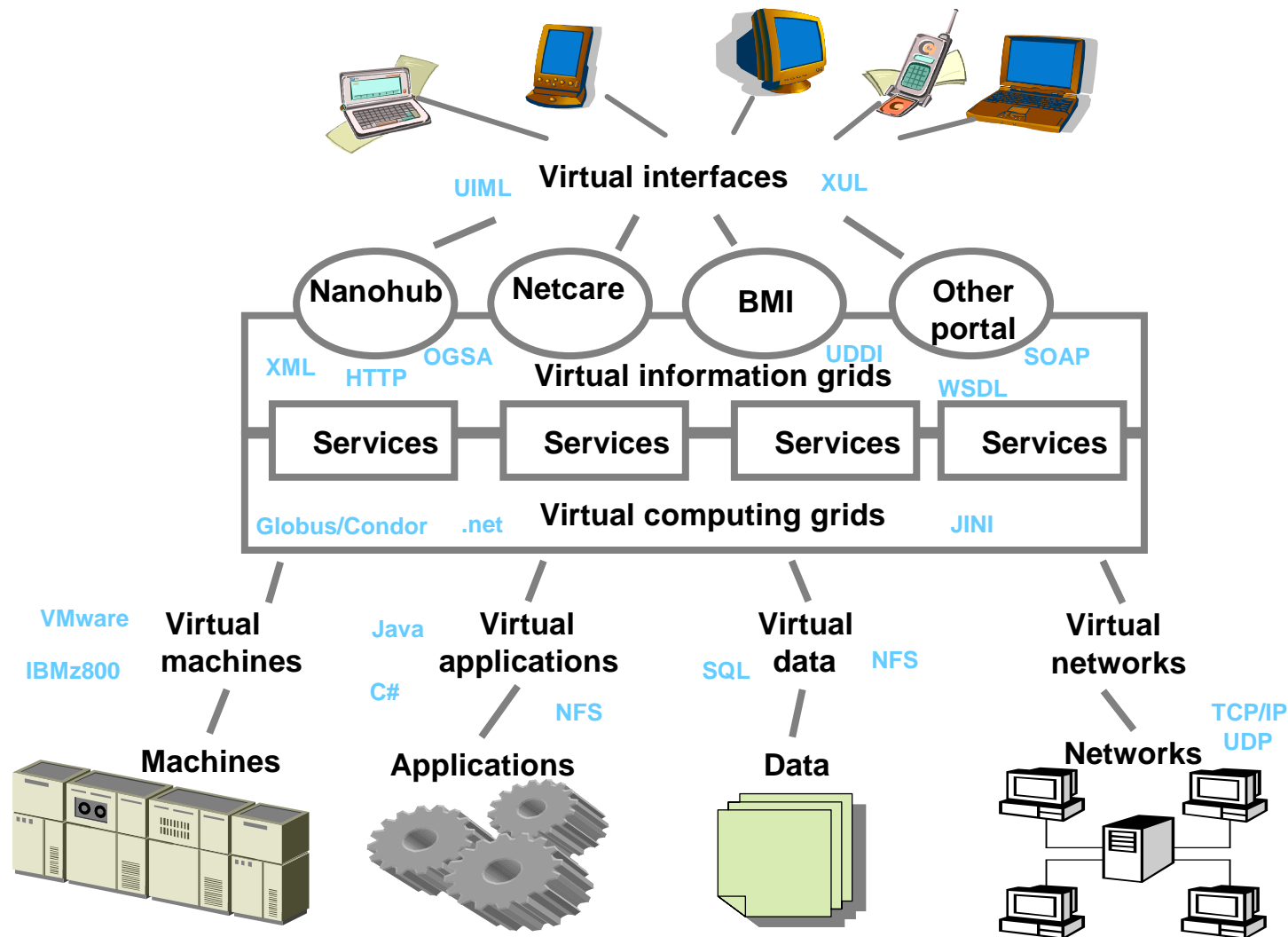
- Enables **simple**, **flexible** and **sustainable** addition of computational applications and computing power
  - nanoHUB: ~20 tools in 3 months
  - GUI and batched applications



sessionid	login	appname	clientname	shadowacc
1923	tayebi	prophet	round6.punch.purdue.edu	job000
1924	jihankim	prophet	round8.punch.purdue.edu	job000
1925	jscovel	prophet	round7.punch.purdue.edu	job000
1926	sid30985	prophet	round5.punch.purdue.edu	job000
1928	ndhillon	prophet	round4.punch.purdue.edu	job000
1929	dcantos	prophet	round2.punch.purdue.edu	job000
1930	merve	prophet	round3.punch.purdue.edu	job000
1931	merve	prophet	round1.punch.purdue.edu	job000
1932	jscovel	prophet	round2.punch.purdue.edu	job001
1979	savas	prophet	round8.punch.purdue.edu	job001
1980	merve	prophet	round7.punch.purdue.edu	job001
1982	yangliu1	prophet	round6.punch.purdue.edu	job001
1984	ngel1914	prophet	round3.punch.purdue.edu	job001
1989	kennell	prophet	round1.punch.purdue.edu	job001
1990	hujun	prophet	round1.punch.purdue.edu	job002
1993	hujun	prophet	round4.punch.purdue.edu	job002
1995	sugatani	prophet	round5.punch.purdue.edu	job001
1999	freibeuter	prophet	round7.punch.purdue.edu	job002
2000	zhuxg	prophet	round4.punch.purdue.edu	job001
2001	freibeuter	prophet	round3.punch.purdue.edu	job002



# In-VIGO Approach





# In-VIGO Approach

- Extensive use of virtualization
  - Security
  - Flexibility/customization
- Decouples grid users from resources
  - Users do not have to manage several credentials
  - In-VIGO proxies on behalf of user(s) simplify resource administration for providers
- Provides execution environments transparently
- Turns tools into Grid-services accessible via a user-friendly Web-interface efficiently

The image displays the In-VIGO Virtual Application Session interface, which is accessible via a web browser. The interface is divided into several sections:

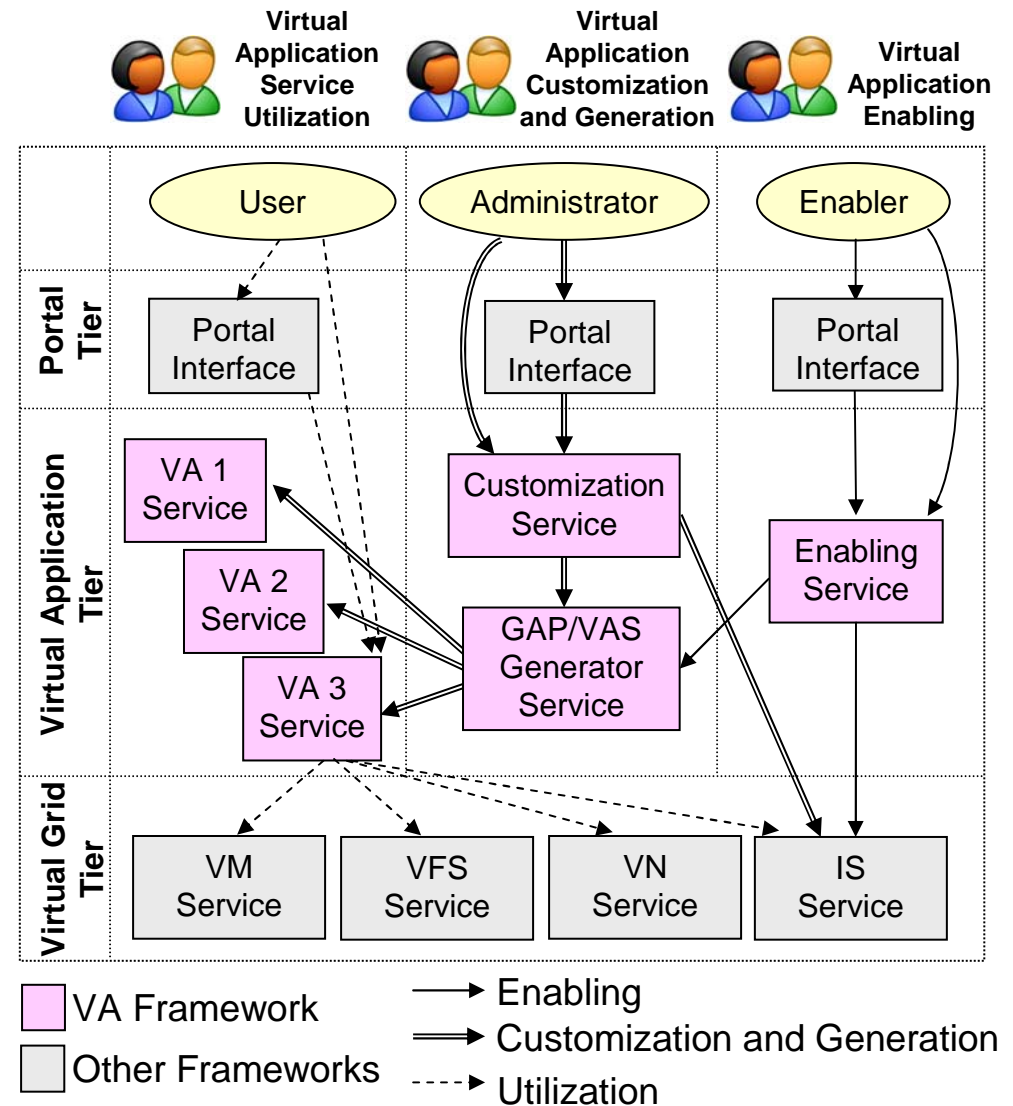
- User Information:** Shows the user as 'lipping', application as 'install', and session id as '16818'.
- Upload File:** Includes options to upload grammar files and application information.
- Set Application Basic Information:** Allows setting application info.
- Execute smartInstaller:** Includes options to check CFG file syntax and enable the application.
- Messages:** A section for viewing messages and active actions. It shows a message from the application install, stating that the virtual application is ready to process the request and that the smolctoy.cfg file has been uploaded successfully.
- Terminal Output:** A window showing the execution of the application. It displays the command 'In-VIGO codes for application smolctoy have been generated' and 'The application smolctoy has been deployed!'. The terminal also shows the execution of the 'smolctoy' command, which outputs the following information:
 

```

[fortes@vws005 fortes]$ smolctoy
GNU Ghostscript 6.52 (2001-10-20)
Copyright (C) 2001 artofcode LLC, Benicia, CA. All rights reserved.
This software comes with NO WARRANTY; see the file COPYING for details.
Loading NimbusSanL-Regu font from /usr/share/fonts/default/Type1/n0190...
... 2418932 1060277 1622424 328620 0 done,
Loading NimbusSanL-Regu font from /usr/share/fonts/default/Type1/n0190...
2757092 1365158 1642520 336084 0 done,
>>showpage, press <return> to continue<<
[fortes@vws005 fortes]$
            
```
- Simulation Results:** A 3D plot showing the results of the simulation. The plot is titled 'MolCToy - Molecular Conduction Toy Model' and shows the energy profile of a molecule. The x-axis is labeled 'X (nm)' and the y-axis is labeled 'Y (nm)'. The z-axis is labeled 'Energy (eV)'. The plot shows a series of peaks and valleys, representing the energy landscape of the molecule.

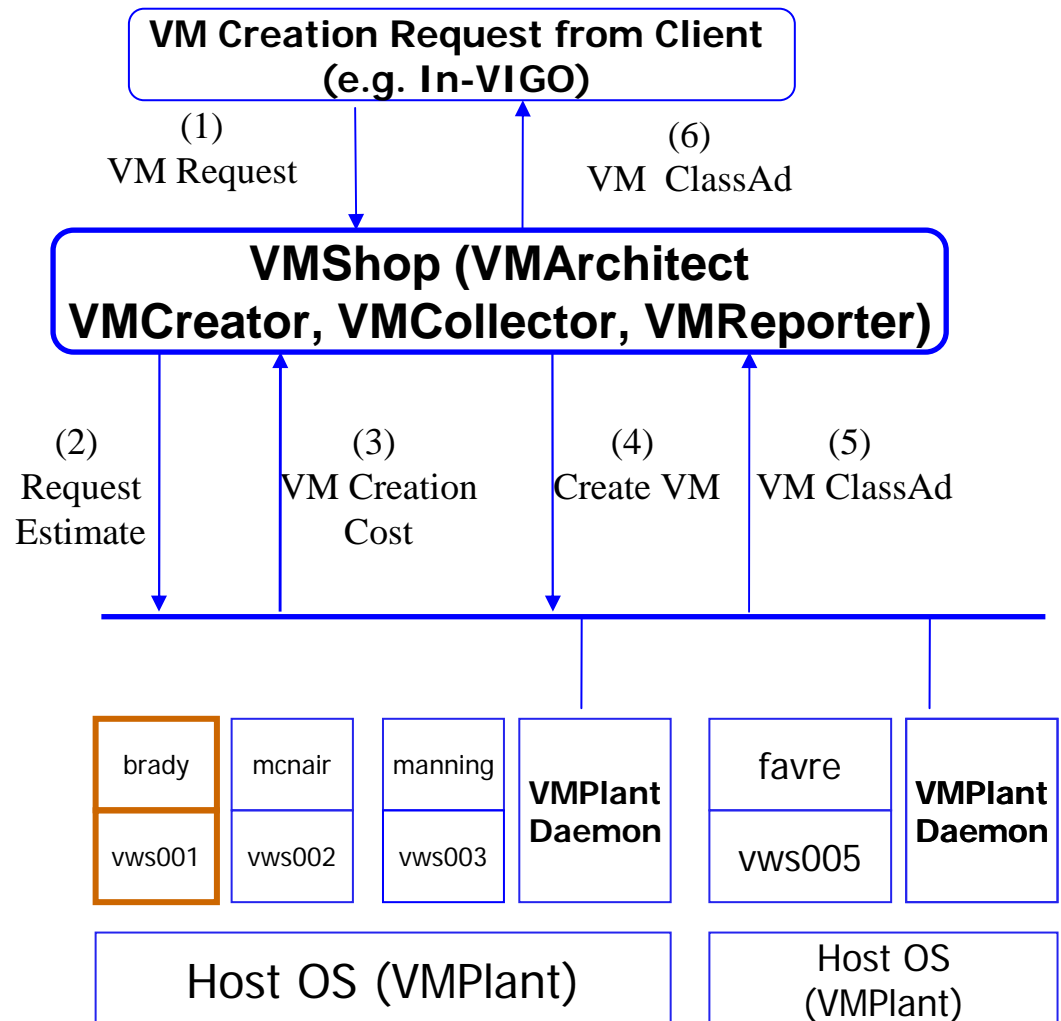
# Grid-enabling unmodified applications

- 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 are 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



# Virtual Machine System

- Provides means to efficiently create/configure/destroy VMs, that is generic across VM technologies [SC 2004]
- Directed Acyclic Graph (DAG) model for defining application-centric VMs
- Cost-bidding model for choosing compute servers for VM instantiation

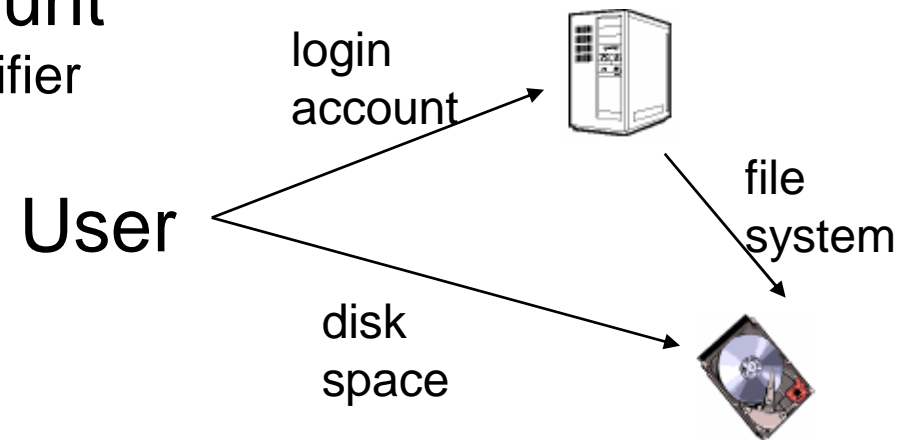




# Logical User Accounts

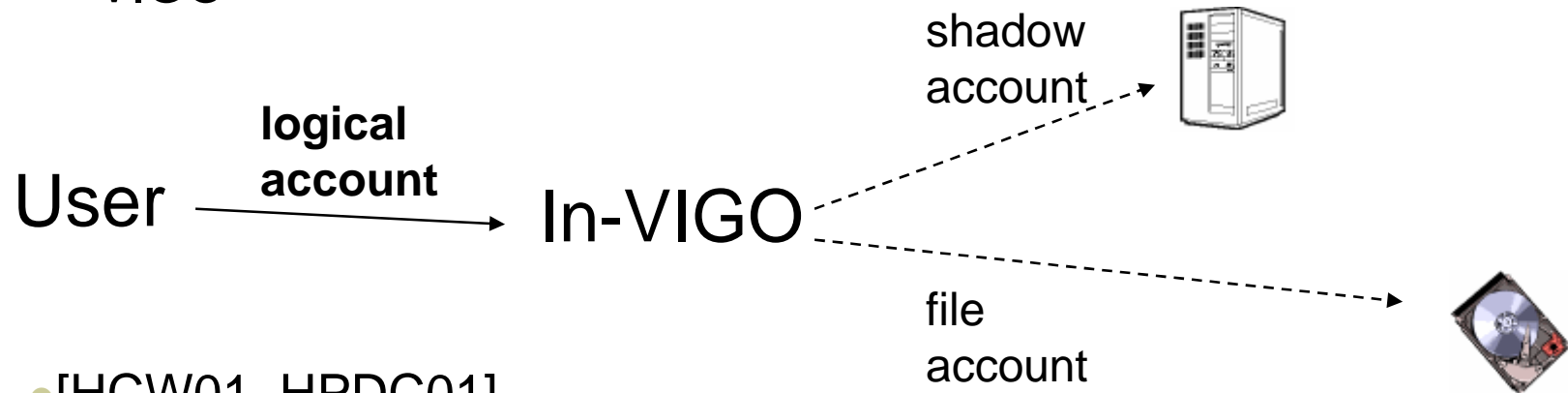
- Traditional user account

- User is assigned fixed identifier (e.g. Unix UID)



- Logical user account

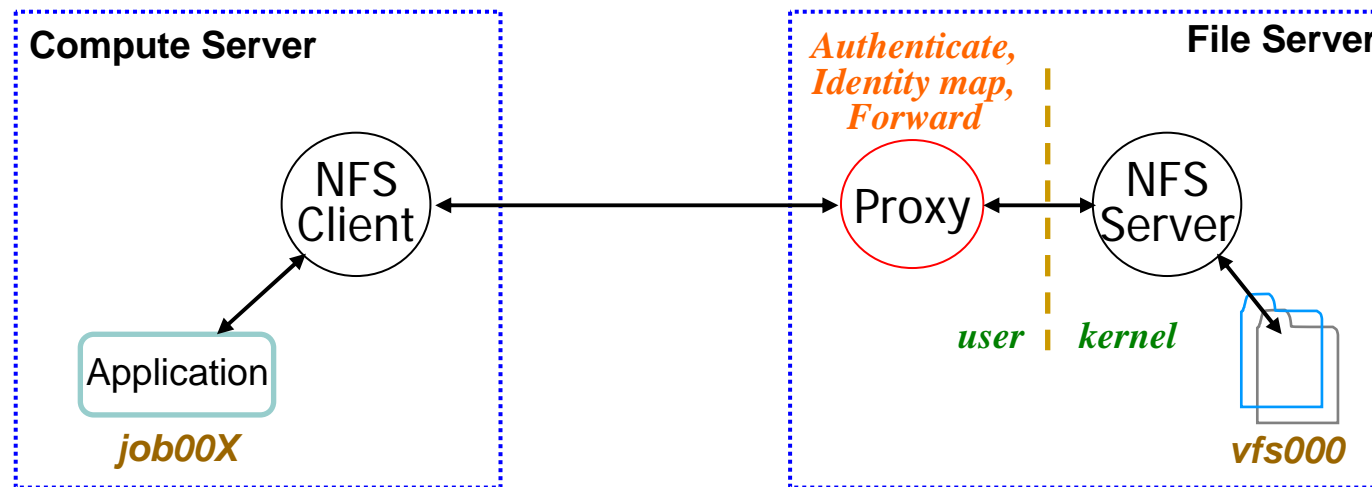
- Physical (shadow) account temporarily assigned to a user by resource management in In-VIGO



- [HCW01, HPDC01]

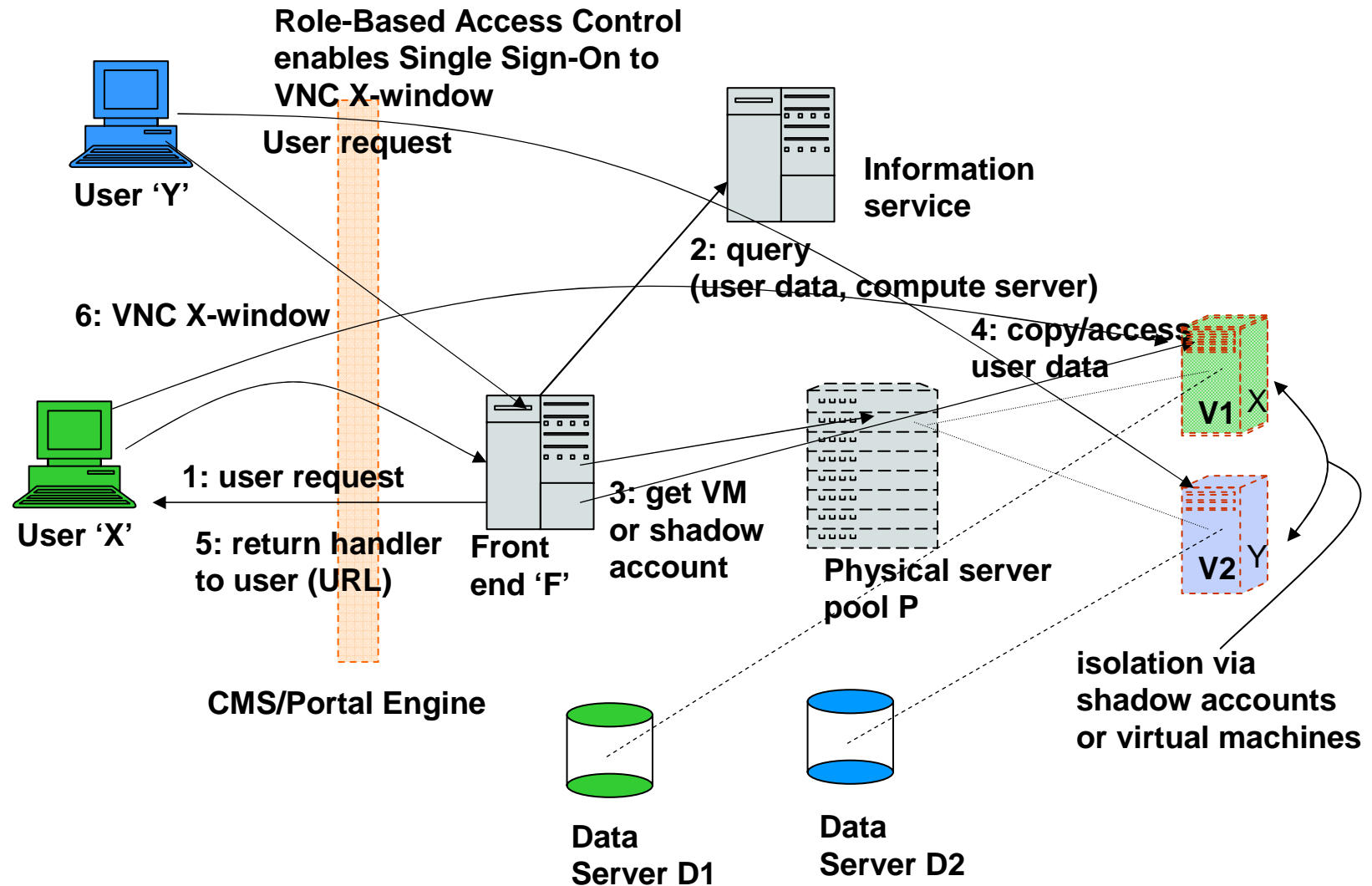
# Grid Virtual File System (GVFS)

- Distributed virtual file system [HPDC 2001, 2004, 2005]
  - Virtualization on Network File System (NFS)
  - User-level proxy based implementation



Support for { on-demand, cross-domain data access  
unmodified binary applications  
unmodified NFS clients/servers

# Putting it all together: GUI Application example



# In-VIGO Status

- Portals
  - SCOOP, HPC, Netcare, nanoHUB
- Publications
  - IEEE Proc., FGCS, DCS, HPDC, SC, IPDPS, ICAC, Europar, GGF, ...
- Classes
  - Distributed Computing
  - Virtual Computers
- Team: 10 Ph.D. students, 4 Faculty

