



# Retele de calculatoare

## Introducere in *Grid Computing*

**Sabin-Corneliu Buraga**

<http://www.infoiasi.ro/~busaco>



*“A successful tool is one that was used to do something undreamt of by its author.”*

**Stephen C. Johnson**

# Cuprins

- Punerea problemei
- Istoric
- Arhitectura \*
- Tipuri de *Grid*
- Implementari
- Initiative & aplicatii

\* Multumiri **Lenutei Alboaie**

# Introducere

- Termenul *Grid* a aparut in anii '90
  - analogie cu rețelele electrice (*power grids*)
- Propunere pentru o **infrastructura de calcul** distribuit destinata initial proiectelor stiintifice si industriale
- Oferă suport pentru cautarea si regasirea informatiilor, indiferent de localizarea lor fizica
- Permite executarea de *task*-uri pe mai multe masini, privite ca un calculator unic

## Scopuri

- Partajarea flexibilă, sigură & coordonată a resurselor între colecții dinamice de indivizi, instituții și resurse
- Abilitatea de a forma organizații colaborative virtuale (*virtual organizations*) – eventual, constituite dinamic – partajind aplicații & date într-un mediu deschis și eterogen pentru a rezolva mutual diverse probleme

# Scopuri

- Abilitatea de a agrega un volum considerabil de resurse computationale dispersate geografic în vederea solutionarii unor probleme complexe
- O infrastructura hardware & software care ofera acces permanent, ieftin, de oriunde, în maniera consistenta, la resurse de calcul
- O modalitate de a procesa în maniera distribuita informatiile disponibile în Internet



# Caracterizare

- Tehnologii inrudite:
  - *Peer-to-peer (P2P)*
  - *Cluster computing*
  - *Internet computing*
  - *Web computing*
  - *Cloud computing*

# Caracterizare

- **Resursa:**

- Entitate partajabila, existenta in cadrul unui *Grid*
- Tipuri de resurse:
  - De calcul: PDA, PC, statie de lucru, server, cluster
  - De stocare: *hard disk*, RAID, SAN,...
  - De tip I/O: senzori, retele, imprimante etc.
  - Logice: utilizatori, contoare de timp,...

# Istoric

- Generatia 1 – proiectul **Globus** (Goble & Foster)
  - Aplicatii necesitind putere mare de calcul
  - Include protocoale (X.509, LDAP, FTP) si instrumente de dezvoltare eterogene
  - Suport pentru accesul la & transferul de fisiere
  - Foloseste tehnologiile Internet, dar ignora Web-ul
  - Dezvoltare in medii academice
  - Partajarea resurselor se realizeaza via GridFTP
  - Urmat de **Legion, Condor, Unicore,...**

# Istoric

- Generatia 2 – **OGSA**  
(*Open Grid Services Architecture*)
  - Aplicatii necesitind prelucrarea unor cantitati mari de date (*data intensive*  $\Rightarrow$  *knowledge intensive*)
  - Locul central este luat de serviciile Web
  - **GGF** (*Grid Global Forum*)
  - Dezvoltare in medii preponderent industriale
  - Numeroase implementari, inclusiv **Globus Toolkit**



## Istoric

- Generatia 3 – prezent & viitor
  - Oferirea de servicii specifice pentru aplicatii bazate pe arhitectura Grid
  - Integrare de paradigme si tehnologii privitoare la calcul distribuit si paralel:
    - Agenti software
    - Servicii Web
    - *Peer-to-peer*
    - *Wireless*
    - *Cloud computing*



# Istoric

- Generatia 3 – prezent & viitor
  - Domenii de interes:
    - Infrastructura
    - Arhitectura
    - *Middleware*
    - Programare
    - Aplicatii
    - Proiecte



# Arhitectura

- Arhitecturile *Grid* utilizeaza simultan un numar mare de resurse (hardware, software, logice)

# Arhitectura

- Aspecte privind partajarea resurselor *Grid*:
  - Identitate & autentificare
  - Autorizare & asocieri de politici de acces (*policy*)
  - Descoperirea resurselor
  - Caracterizarea resurselor
  - Alocarea resurselor
  - Managementul resurselor
  - Contorizare, plata (*billing*), contracte privitoare la serviciile oferite – *Service Level Agreement (SLA)*
  - Securitate



# Arhitectura

- Aspecte de interes:
  - Interoperabilitatea
  - Oferirea unor servicii de baza
  - punerea la dispozitie de servicii de nivel inalt

# Arhitectura

- **Interoperabilitatea**

- Recurgerea la protocoale generice
- Arhitectura Grid  $\equiv$  arhitectura protocolului
- Arhitectura Grid este o arhitectura a serviciilor utilizate, independenta de API
  - punct de vedere impartasit de SOA  
(*Service Oriented Architecture*)
- In prezent, se apeleaza la servicii Web

# Arhitectura

- **Ce servicii de baza trebuie oferite?**
  - Numire (*Naming*)
  - Autorizare (*Authorization*)
  - Securitate (*Security*)
  - Confidentialitate (*Privacy*)
  - ...

# Arhitectura

- **Ce servicii de nivel inalt trebuie oferite?**
  - Controlul activitatilor  
fluxuri de lucru (*Workflows*)
  - Tranzactii (*Transactions*)
  - *Data/relation mining*
  - Descoperirea cunostintelor  
(*Knowledge discovery*)
  - ...

# Arhitectura

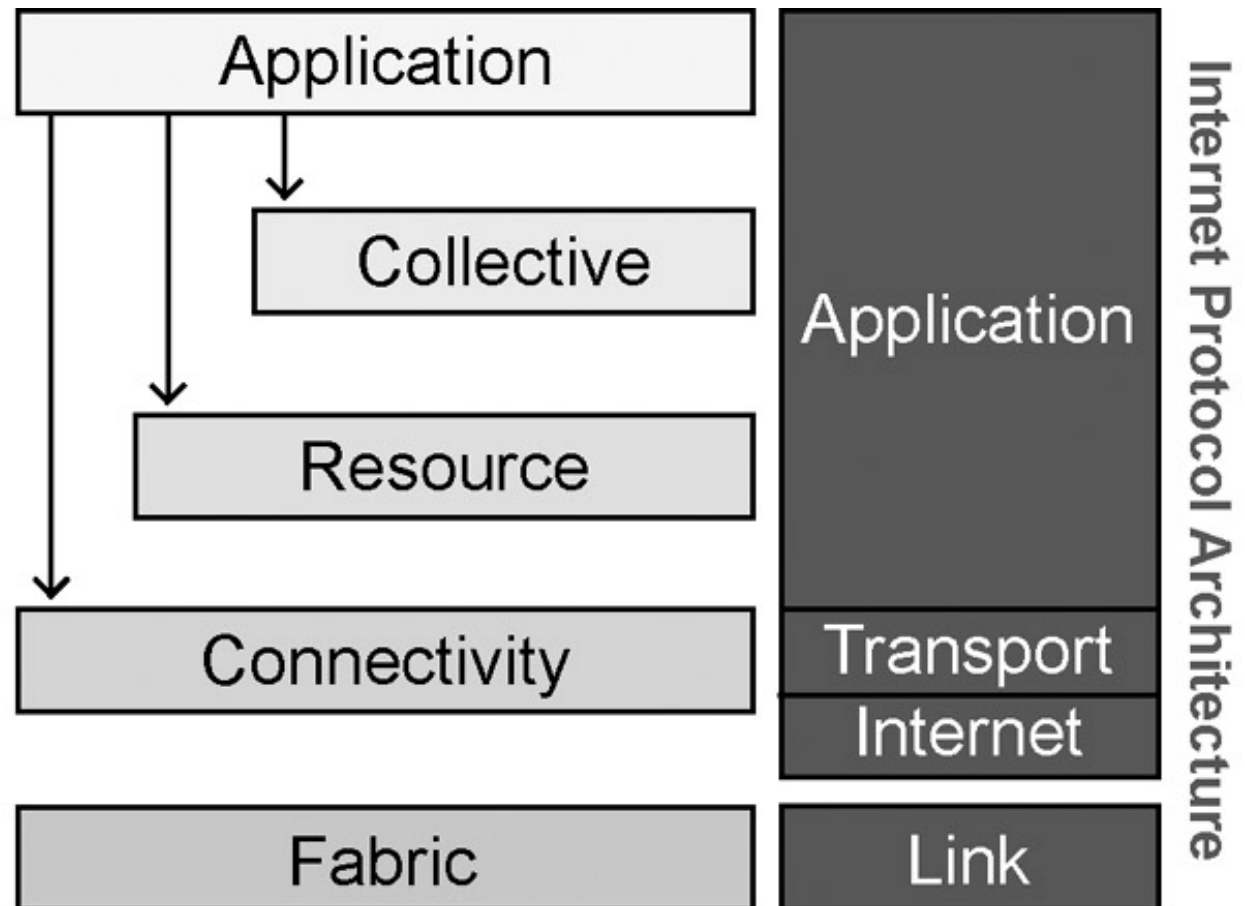
Necesitatea unei arhitecturi stratificate:

"Coordinating multiple resources": ubiquitous infrastructure services, app-specific distributed services

"Sharing single resources": negotiating access, controlling use

"Talking to things": communication (Internet protocols) & security

"Controlling things locally": Access to, & control of, resources



# Arhitectura

- *Fabric*
  - Furnizeaza resurse (de calcul, de stocare, cataloage, retea,...) pentru care accesul partajat este mediat de protocoalele *Grid*-ului
  - Oferă componente care implementează operațiile locale, specifice fiecărui tip de resursă

# Arhitectura

- *Connectivity*
  - Nucleu al protocoalelor de comunicare si de autentificare pentru tranzactiile de retea
  - Servicii minimale pentru comunicare:  
*transport, dirijare si numire*
  - Moduri de autentificare:  
*inscriere singulara (user subscribe), delegare, integrarea cu solutii de securitate locala, relatii bazate pe incredere (trust)*

# Arhitectura

- *Resource*

- Scop: definirea de protocoale (+API/SDK) pentru negocieri sigure, monitorizare, control, contabilitate si plata a operatiilor de partajare a resurselor individuale
- Protocoalele se concentreaza asupra resurselor individuale, iar problemele legate de starea globala sunt ignorate la acest nivel

# Arhitectura

- *Resource* – continuare
  - Utilizarea de **protocoale de informare**
    - Obținerea informațiilor despre structura și starea unei resurse:
      - configuratie,
      - incarcare,
      - politica de utilizare
      - ...

# Arhitectura

- *Resource* – continuare
  - **Protocoloale de administrare**
    - Negocierea accesului la resursele partajate:
      - **solicitari**  
(*e.g.*, calitatea serviciilor – *QoS*, rezervare)
      - **operatii**  
(*e.g.*, creare, accesare de date,...)
    - Instanțierea relațiilor de partajare
    - Monitorizarea/controlul stării unei operații

# Arhitectura

- *Collective*
  - Oferă protocoale și servicii care privesc global resursele Grid-ului
    - *e.g.*, facilitează interacțiunile dintre seturi de resurse
  - Poate implementa servicii de partajare a resurselor:
    - Catalogare (*directory*)
    - Co-alocare, planificare și intermediere
    - Monitorizare și diagnosticare
    - Replicare și descoperire
    - Colaborare
    - ...

# Arhitectura

- *Application*
  - Cuprinde aplicatiile-utilizator care opereaza pe baza arhitecturii *Grid*
    - Medii de programare + biblioteci de nivel inalt
    - Exploatarea sinergiei dintre *e-business* si servicii Grid

# Protocoale & API-uri

- Nivelul *fabric* include protocoalele & API-urile oferind acces la resursele partajate
  - Viziune logica asupra resurselor si nu una fizica
- *Connectivity layer* defineste protocoale de baza pentru tranzactii de retea specifice *Grid*-ului
  - Protocoalele Internet standard
  - Protocoale vizind securitatea:
    - *Grid Security Infrastructure (GSI)*
      - autentificare, autorizare & protejare

# Protocoale & API-uri

- *Resource layer* – protocoale pentru initierea & controlul partajarii resurselor *locale*
  - Managementul alocarii resurselor *Grid*-ului: *Grid Resource Allocation Management (GRAM)*
    - alocarea, rezervarea, monitorizarea & controlul de la distanta ale resurselor
  - **GridFTP** – acces & transport eficient al datelor
  - Serviciul de informare privind resursele *Grid*: *Grid Resource Information Service (GRIS)*
    - acces la structura si starea unui nod al *Grid*-ului

# Protocoale & API-uri

- **Nivelul *collective*** specifica protocoale oferind servicii la nivel de sistem pentru exploatarea pe scara larga a unui sistem *Grid*
  - Servicii de indexare si meta-catalogare pentru structurarea si accesarea resurselor
  - *Broker*-i de resurse pentru descoperirea si alocarea resurselor conform cerintelor aplicatiilor



## **Grid applications**

Science, engineering, commercial applications, Web portals

## **Grid programming environments and tools**

Languages, interfaces, libraries, compilers, parallelization tools

## **User-level middleware—resource aggregators**

Resource management and scheduling services

## **Core grid middleware**

Job submission, storage access, info services, trading accounting

## **Security infrastructure**

Single sign-on, authentication, secure communication

## **Grid fabric**

PCs, workstations, clusters, networks, software, databases, devices

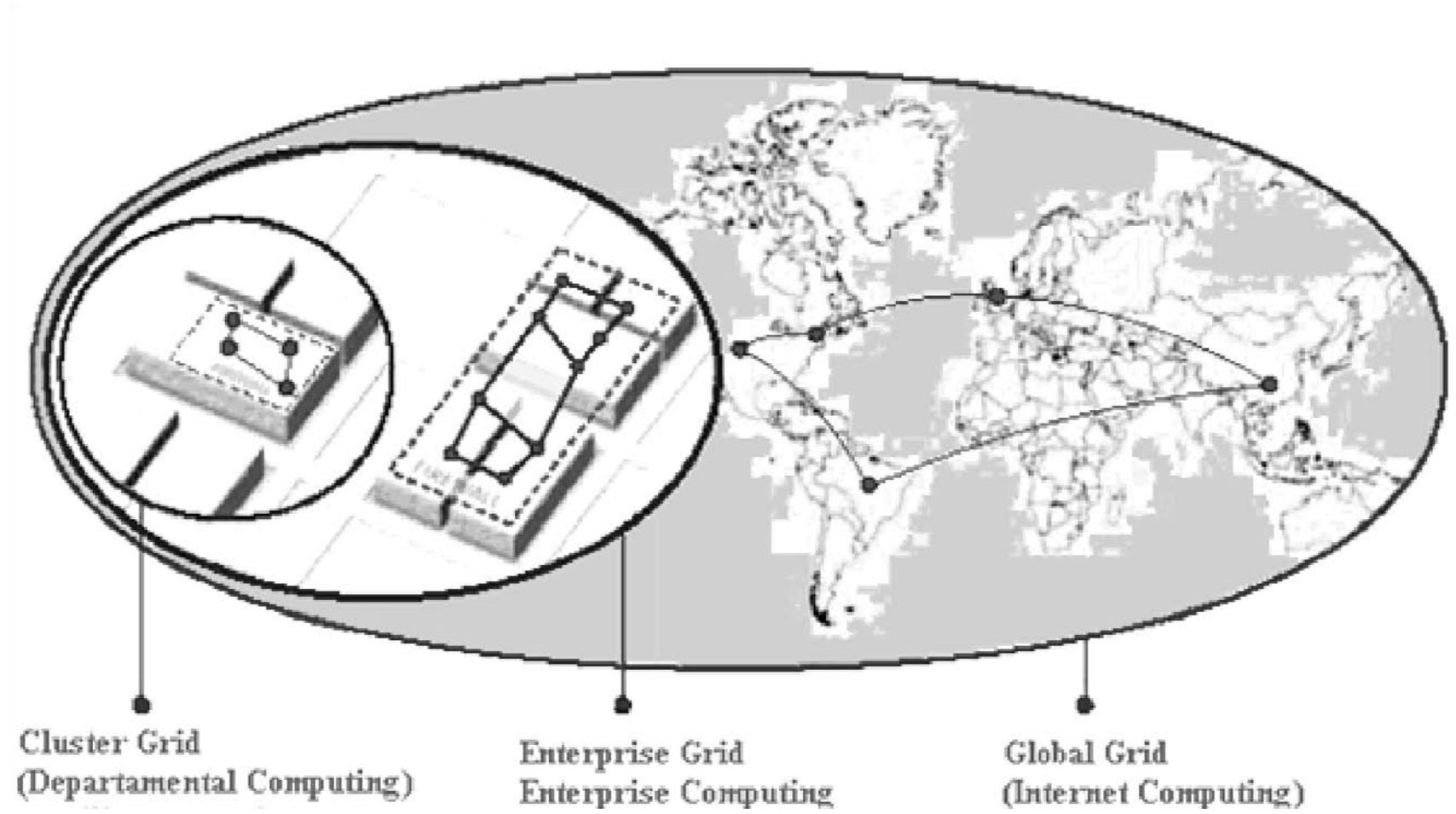


# Tipuri

- Tipuri de *Grid* – la nivelul localizarii
  - *Global Grid*
  - *Enterprise Grid*
  - *Cluster Grid*



## Tipuri



# Tipuri

- *Global Grid*
  - Reprezinta o colectie de *Grid*-uri de tip *enterprise* si *cluster*, plus alte resurse distribuite geografic pe o arie larga – continent sau planeta
    - Politica de utilizare globala
    - Protocoale generale de partajare a resurselor
  - Numit si *inter grid* (IBM)

# Tipuri

- *Enterprise Grid*
  - Faciliteaza partajarea de resurse intre mai multe departamente si colective din cadrul unei organizatii (virtuale)
    - Politici de management a resurselor
  - Numit si *intra grid* ori *campus grid*
  - In contextul extranet: *Extraprise Grid*

# Tipuri

- *Cluster Grid*
  - Reprezinta o colectie de calculatoare interconectate folosite (si vazute) ca o resursa unica la nivel de departament/grup
    - *departmental grid* (Sun) / *infra grid* (IBM)
  - Face posibila utilizarea la maxim a resurselor colectiei de calculatoare  
(*mainframe*-uri, PC-uri, *laptop*-uri, *smartphone*-uri,...)
  - **Cluster**  $\equiv$  ansamblu de calculatoare – dintr-un LAN – care formeaza o resursa unica de calcul

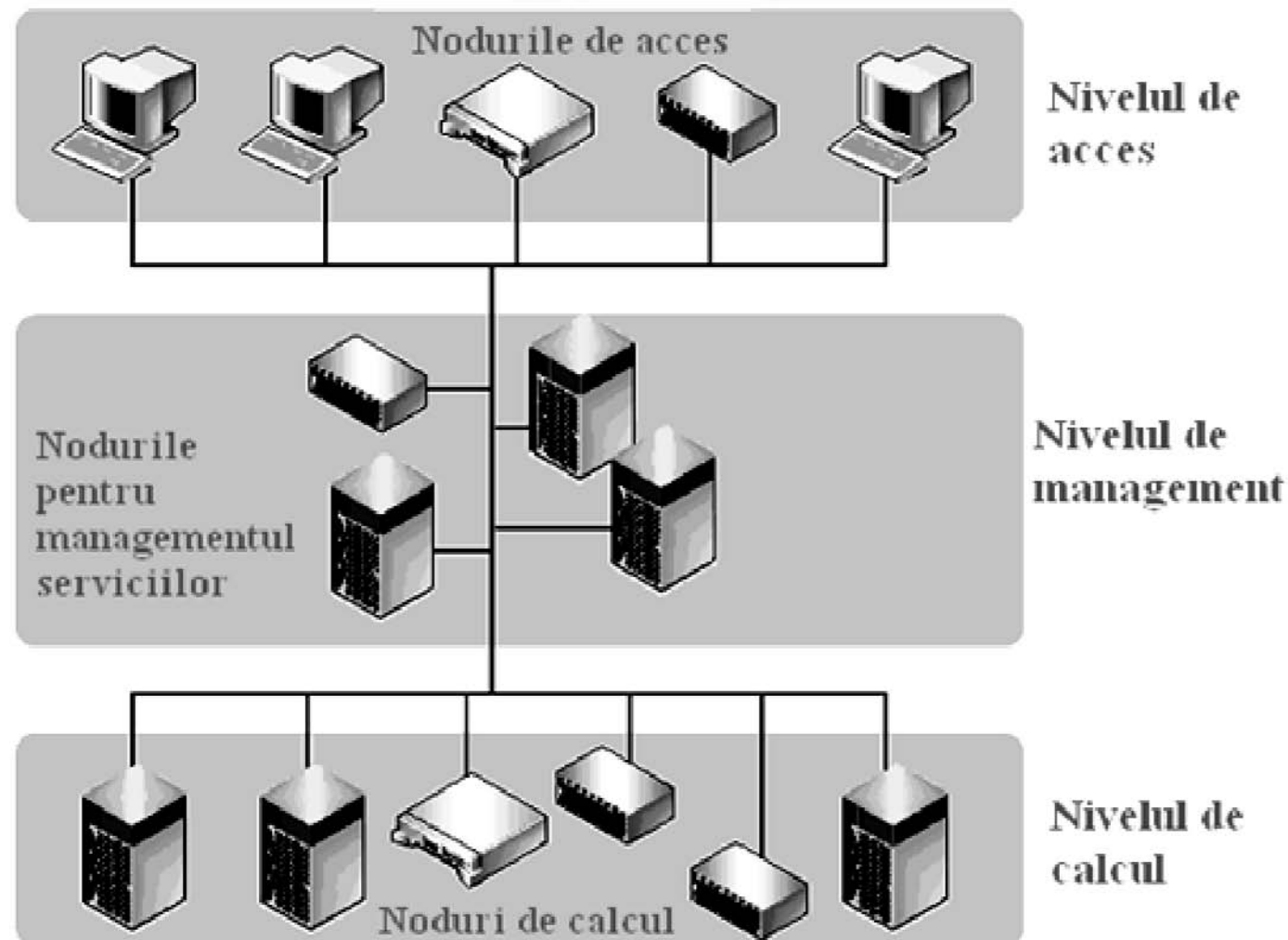


# Tipuri

- *Cluster Grid*
  - Alternative:
    - NOW (*Networks of Workstations*),
    - COW (*Clusters of Networks Workstations*)



## Tipuri



# Tipuri

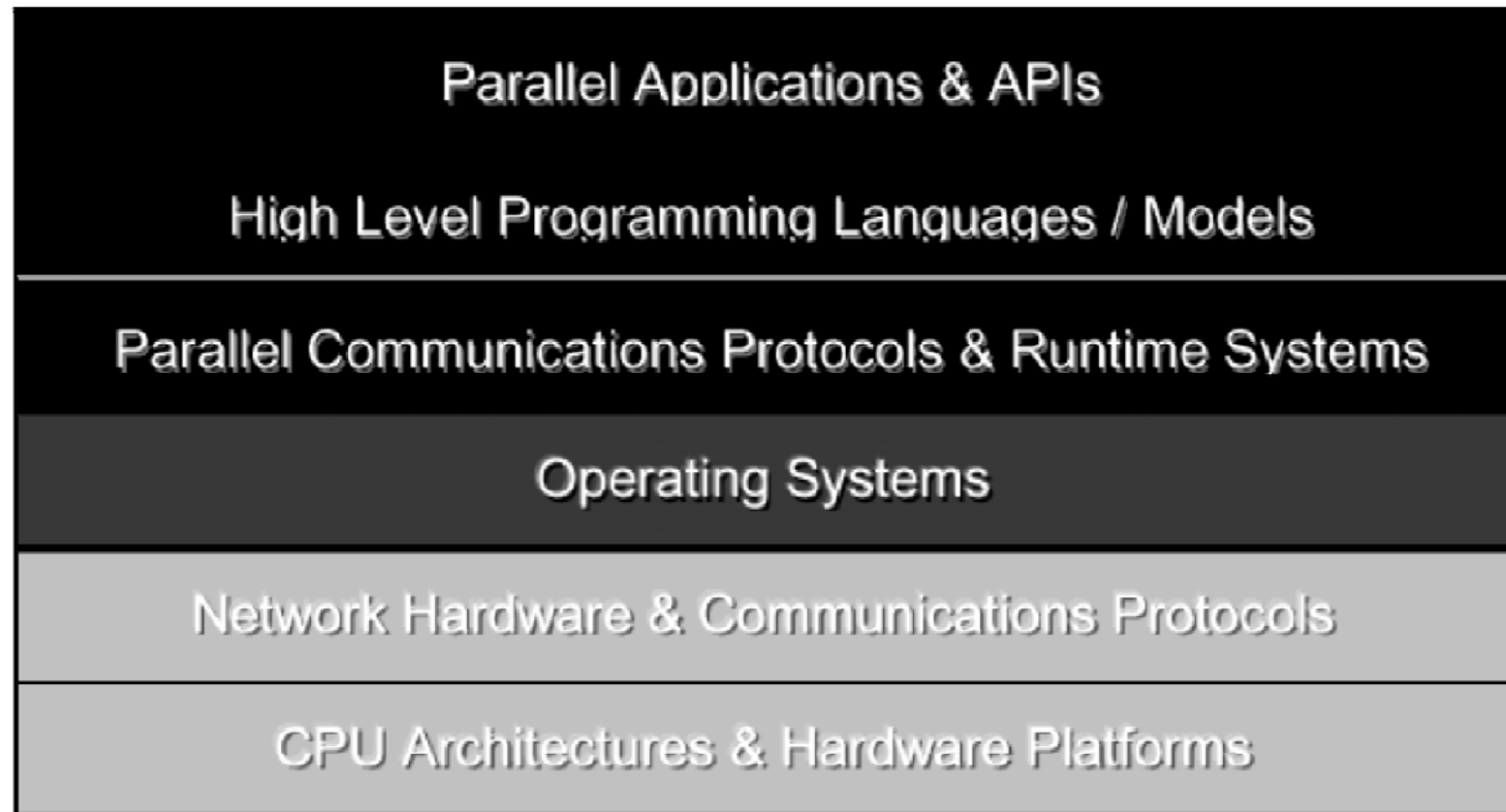
- Categoriile de cluster:
- **Cluster dedicate**
  - Constituite din noduri omogene (procesor, OS,...) – *rack* de placi de baza
- **Cluster de intreprindere**
  - Eterogene, distribuite geografic, control limitat
  - Activitatile locale au prioritate mai mare decat cele ale intreprinderii

# Tipuri

- Categoriile de cluster (continuare)
  - *Cluster cycle-sharing*
    - Partajeaza ciclurile procesoarelor
  - *Cluster ad-hoc*
    - Create in mod dinamic pentru rezolvarea unor probleme tranzitorii
  - *Cluster mobile*
    - Pur mobile (dispozitive mobile: telefon, *smartphone*, *laptop*, *palmtop*, consola de jocuri,...) sau mixte (mobile + fixe)

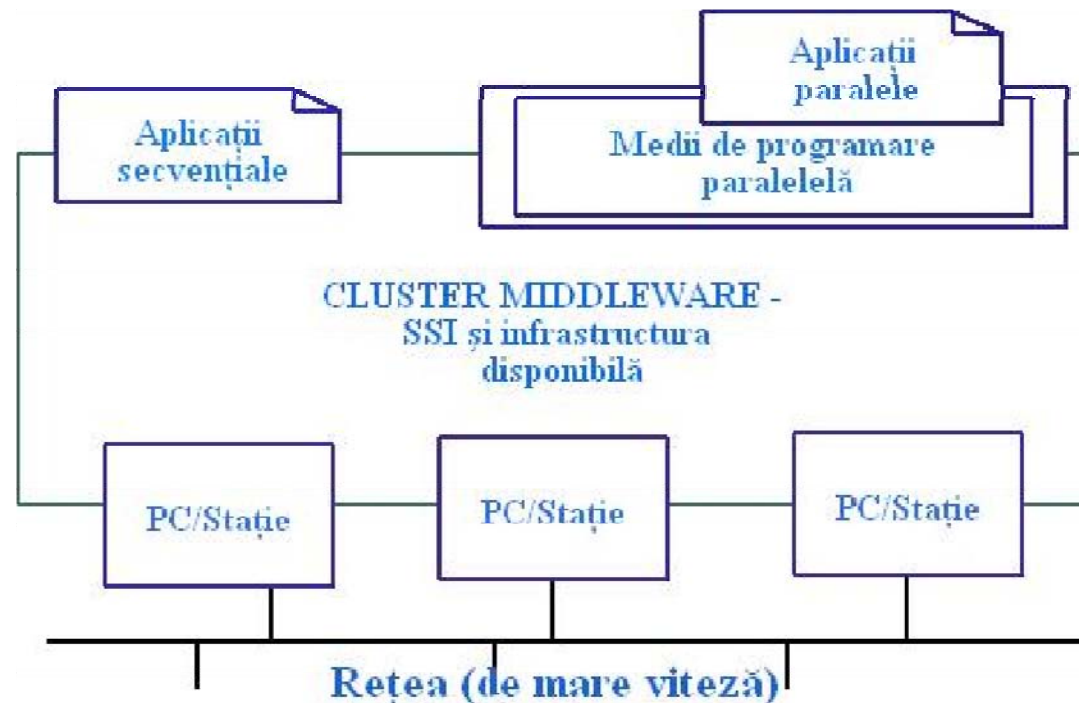


# Tipuri



# Tipuri

## Arhitectura unui cluster:



SSI = *Single System Image*

# Tipuri

- Aplicații specifice *cluster computing*:
  - Procesari intensive
  - (**HPC – High Performance Computing**):
    - calcul numeric
    - grafică computațională 2D/3D  
(*rendering – e.g., raytracing, shading,...*),
    - simulări (*biocomputing, domeniul militar,...*)
  - *cluster raytracing*
    - lucrare de licență: Silviu Homoceanu, 2006



## Tipuri

- Aplicații specifice *cluster computing*:
  - Cautarea distribuită de resurse
  - Aplicații critice rulând în timp-real
  - Stocare distribuită a unui volum mare de date + depozite de date (*warehouses*)
  - Divertisment – de exemplu: jocuri *online*

# Tipuri

- Implementare: **Beowulf** (aparut in 2003)
  - Suport pentru constituirea de clase de *cluster*-e (*low-end/high-end*)
  - Calculatoarele pot fi adaugate dinamic
  - Comunicare via MPI (*Message Passing Interface*)
  - Model de programare independent de infrastructura, de tehnologiile de retea sau de componente
  - Noduri de tip *master* (coordonatori) si *slave/worker* (procesatori)



# Tipuri

- Tipuri de Grid – la nivelul logic
  - *Computational Grid*
  - *Information Grid*
  - *Knowledge Grid*

# Tipuri

- *Computational Grid*
  - Nivel de baza, axat pe oferirea de servicii de calcul si date la scara larga
    - Necesitatea unei infrastructuri pentru monitorizarea si controlul resurselor vazute ca un ansamblu
  - Rafinari: *desktop grid*, *server grid* & *high-performance/cluster grid*
  - Exemple: procesari grafice intensive

# Tipuri

- *Information Grid*
  - Permite accesul uniform la surse eterogene de informatii si ofera servicii publice de baza rulind pe resurse computationale distribuite
    - Resursele computationale pot varia de la simple invocari de metode la distanta pana la aplicatii sofisticate
    - Exemple: *grid*-uri in domeniul bioinformaticii sau stiintelor pamintului



## Tipuri

- *Knowledge Grid*
  - Nivel superior oferind (meta-)servicii specializate pentru descoperirea de resurse stocate in depozite distribuite si pentru managementul serviciilor informationale
    - Meta-serviciile pot fi compuse din alte servicii

# Tipuri

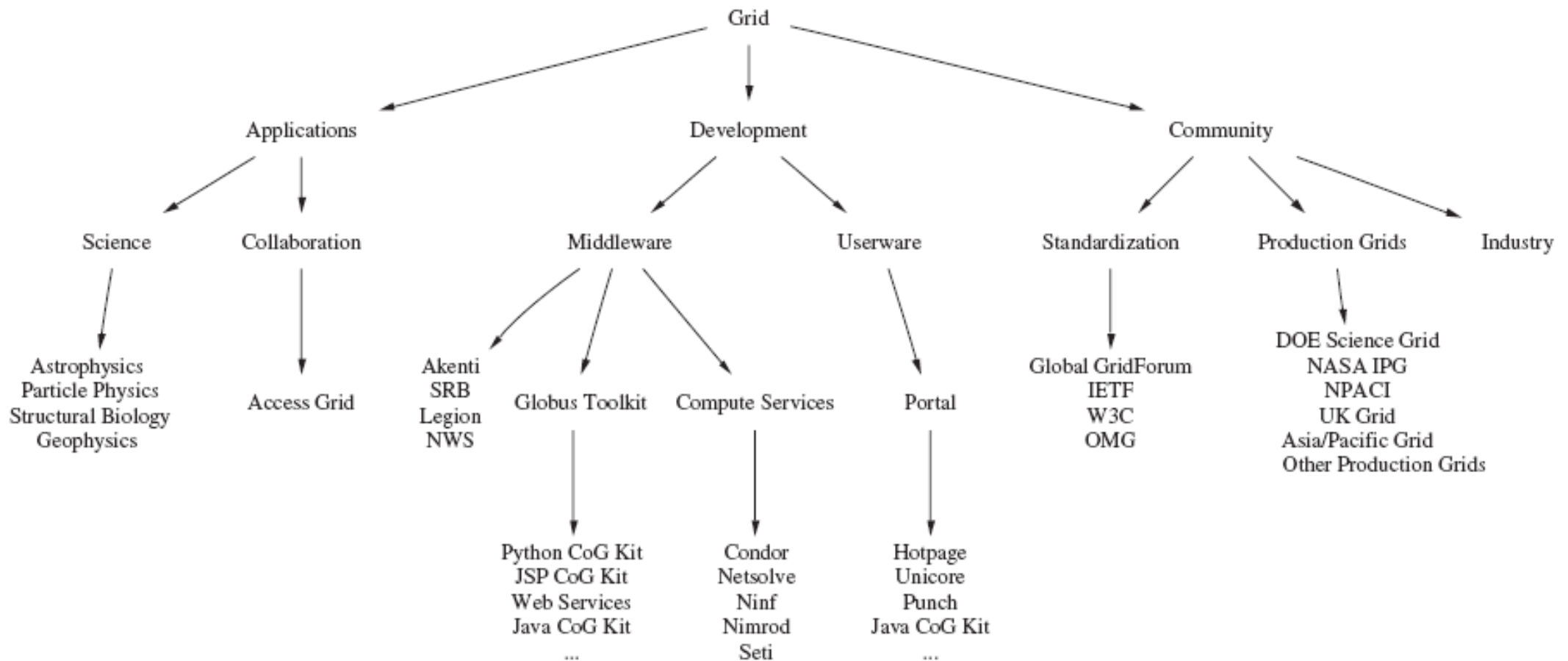
- Conform specificului aplicatiilor disponibile intr-un *Grid*:
  - Arhitecturi multi-strat (*N-tiered*)



- Aplicatii *role-based*
- Aplicatii bazate pe servicii



## Tipuri



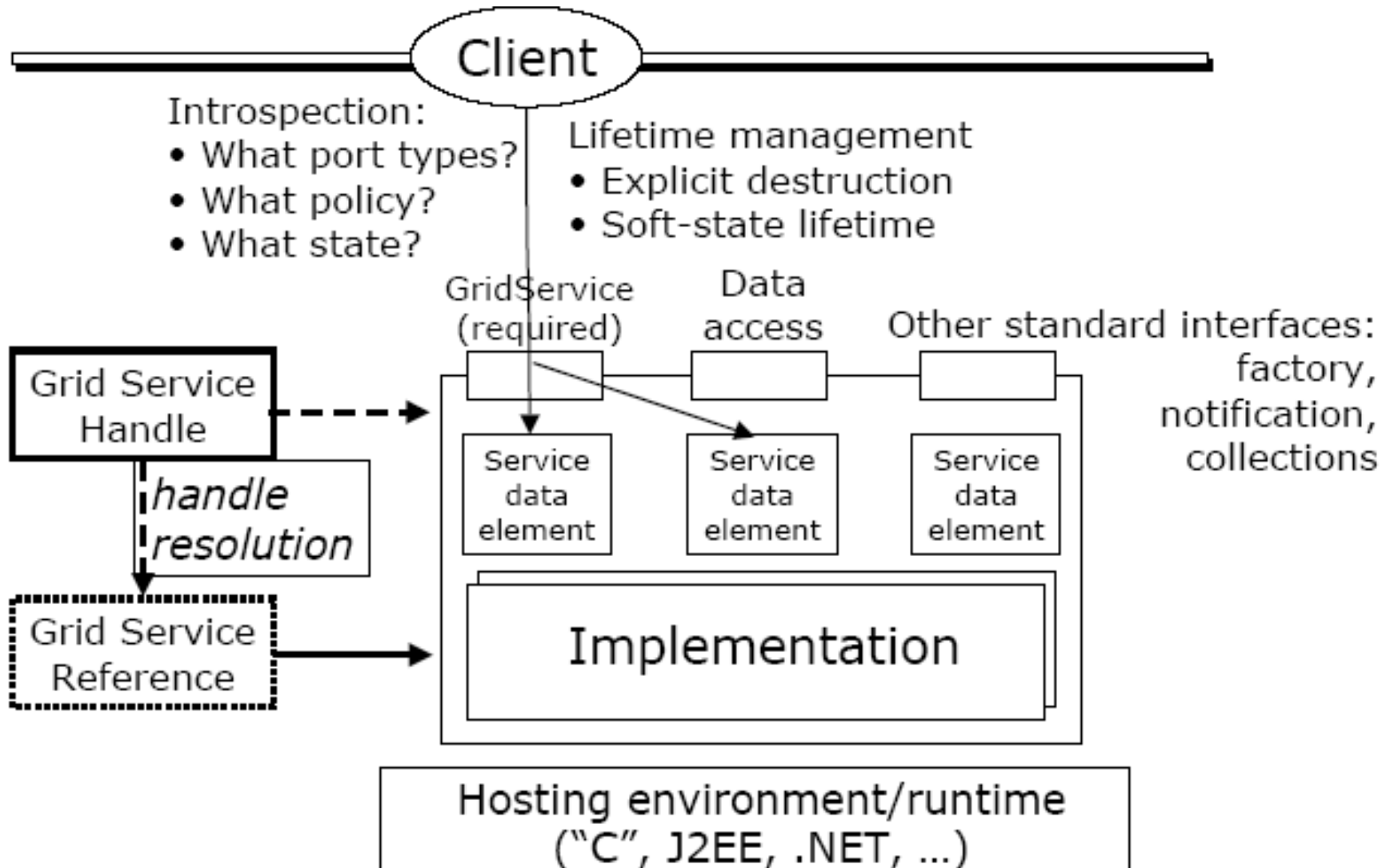
# Implementari

- *Open Grid Service Architecture (OGSA)* extinde serviciile Web
  - Controlul instantelor serviciilor Web tranzitorii
  - Serviciile Web sunt dependente de stare
  - Managementul timpului de viata a serviciilor (*lifetime management*)
  - Existenta unei infrastructuri deschise: *OGSI (Open Grid Service Infrastructure)*

# Implementari

- Serviciile Grid constituite trebuie sa fie:
  - **Dinamice si volatile** – multimi de servicii compuse create, invocate si eliminate “din zbor”
  - **Ad-hoc** – nu exista locatie centrala ori control central; nu exista relatii de incredere
  - **Pe scara larga** – orchestrarea unui numar mare de servicii (>100) trebuie realizata oricind
  - **Disponibile, potential, pe termen lung** (*e.g.*, o simulare poate dura saptamini)

# Implementari



# Implementari

- Specificatii tehnice:
  - *Open Grid Services Infrastructure (OGSI)*
  - Securitate, acces la date, Java *binding* etc.
  - Standardizate de *Global Grid Forum*

# Implementari

- Globus Toolkit 4
  - implementare de referinta bazata pe OGSA
  - [www.globus.org](http://www.globus.org)
  - varianta minimala: *Glite*
  - *Java Grid Commodity Kit (CoG)* – dezvoltatorul poate evita interactiunea directa cu serviciile Globus (executia *job*-urilor prin GRAM, transfer via GridFTP, descoperirea caracteristicilor resurselor etc.)
  - *pyGlobus* – implementare Python

# Implementari

- Instrumente oferite de IBM, Microsoft, NEC, Oracle si altii pentru UNIX/Linux sau Windows
- Alte proiecte:
  - Apple XGrid
  - Sun Grid Engine

# Initiative

- **GridPP** (*UK Computing Grid for Particle Physics*)
  - Parte integranta a celui mai mare *Grid* din lume:  
**LCG** (*LHG Computing Grid*),  
unde LHG = *Large Hadron Collider* (CERN, din 2007)
  - LCG este compus din peste 5200 CPU-uri,  
4000 TB memorie, rulind >5000 de *task*-uri simultan
  - Face parte din cadrul proiectului EuroGrid
  - **[www.gridpp.ac.uk](http://www.gridpp.ac.uk)**



# Initiative

- **Fraunhofer Grid Alliance**
  - Scop: oferirea unui *Grid* computational pentru acces facil la resursele *Grid*-ului via un portal Web
  - Bazat pe Globus Toolkit
  - Functioneaza in mediul academic & industrial
  - **[www.fhrg.fhg.de](http://www.fhrg.fhg.de)**

# Initiative

- **JGrid**
  - *Framework* pentru *Grid*-uri compuse din componente hardware/software vazute ca servicii
  - Se bazeaza pe tehnologia **Jini** – infrastructura & model programatic pentru crearea de sisteme distribuite dinamice in Java
  - Aplicatiile JGrid pot fi dezvoltate via **P-Grade** (mediu de dezvoltare grafic)
  - **<http://jgrid.jini.org/>**

# Initiative

- **Alchemi**

- Sistem *Grid* bazat pe .NET Framework
- asigura interoperabilitatea cu alte sisteme *Grid* via *Gridbus Grid Service Broker*
- utilizat in cadrul proiectului **DisMy** – sistem *Grid* semantic bazat pe modelul de coordonare Linda:  
Andrei Iacob – lucrare de licenta, 2008

# Initiative

- Exemple de aplicatii:
  - Vizualizarea 3D fotorealistica
    - Renderizare prin **POV-Ray**, date distribuite via **Grix**
  - Chirurgie vasculara virtuala
    - **CrossGrid** – componenta-cheie: **Migrating Desktop**
    - <http://www.crossgrid.org/>

# Initiative

- Exemple de aplicatii:
  - Rezolvarea problemelor de optimizare
    - Proiectul TRACER (foloseste arhitecturi ca Globus, Condor, Legion, Sun Grid Engine)
    - <http://neo.lcc.uma.es/>
  - Monitorizarea resurselor dintr-un *Grid* – GriW
    - Bazata pe paradigma SOA
    - Implementare multi-platforma (Java, .NET, C#, JavaScript, SVG)
    - Lucrare de licenta a Alinei Sirbu (2008)



## In loc de concluzii

- *Grid*-ul reprezinta o aplicatie reala, implicind un set larg de resurse eterogene, cu scopul de a realiza calcul de inalta performanta sau de a oferi acces transparent la un volum mare de cunostinte indiferent de localizarea lor in cadrul retelelor/Internetului
- Tehnologiile & aplicatiile pentru *Grid* implica rezolvarea unor probleme de scalabilitate, flexibilitate, performanta si securitate



# Rezumat

- Punerea problemei
- Istoric
- Arhitectura
- Tipuri de *Grid*
- Implementari
- Initiative & aplicatii



# Intrebari?