

Grades Computacionais

uso de recursos geograficamente dispersos



o que são grades

- analogia com rede elétrica:
 - capacidade de ligar-se na rede e obter energia que não se sabe de onde vem
 - capacidade de ligar-se numa grade e obter processamento e memória idem...
 - » ainda um pouco distante
- uso de pool de recursos de diversas instituições



aplicações

- supercomputação distribuída
 - paralelismo
- alto throughput
 - tarefas independentes
- uso de recursos específicos por tempo limitado
- computação data-intensiva
- computação colaborativa
 - interação coletiva com simuladores ou mundos virtuais



diferenças?

- ainda se trata de computação distribuída
 - fator badalação...
- heterogeneidade maior
 - sistemas operacionais
 - bibliotecas, etc
 - **políticas administrativas e segurança**
- transparência pode ser requisito mais complicado



sistemas e ferramentas

- ProActive
- Legion
- DEISA
 - distributed european infrastructure for supercomputing applications
- Condor-G
- ...
- Globus



Globus

- projeto multi-institucional
- modelo de bolsa de serviços
 - conjunto de bibliotecas e serviços que podem ser usados de forma independente
 - » vantagens e desvantagens
- serviços principais:
 - gerência de recursos
 - segurança
 - informação
 - transferência de arquivos
- atualmente versão 4 (web services)
 - profusão de versões e alternativas!!!



segurança

- GSI - Globus Security Infrastructure
- uso de uma operação de *sign-on* que gera um certificado temporário
 - comando `grid-proxy-init`
 - certificado proxy em arquivo temporário
- autenticação e delegação baseada em openssl
- autorização baseada em mapeamento de usuários

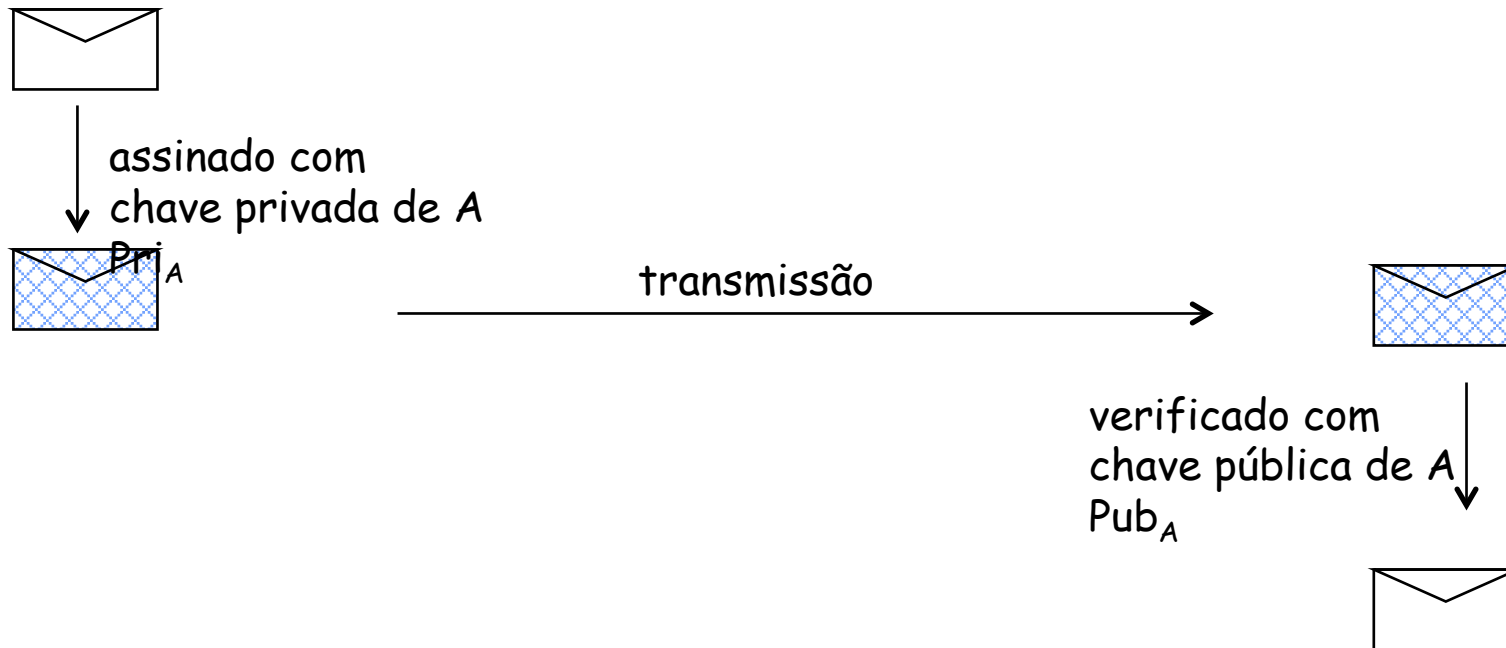


certificados

- autenticação
 - senhas
 - chaves simétricas e assimétricas
- chaves assimétricas
 - autenticação, privacidade e integridade
 - quem garante a validade de um par de chaves?
- protocolos de autenticação



autenticação



- quem garante que essa realmente é a chave pública de A?
 - uso de certificados X509 assinados por autoridades certificadoras



certificados proxy

- cada interação que precisa ser autenticada precisa da chave privada do usuário
 - desgaste da chave
 - acesso a chave privada tipicamente protegido por senha ou frase no sistema de arquivos local
- Globus: geração de certificado proxy assinado com a chave privada do usuário



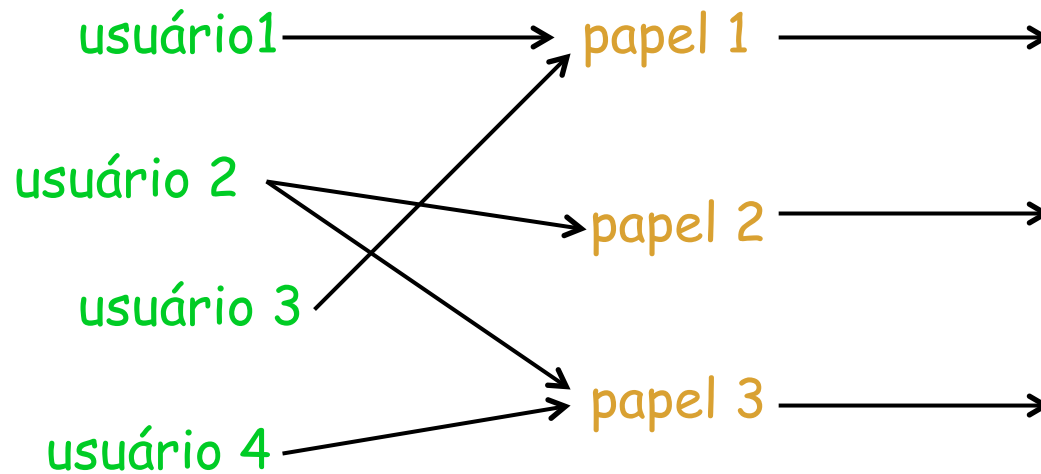
autorizações

- uma vez assegurada a identidade de um usuário, a que recursos ele deve ter acesso?
 - matriz recursos X usuários
 - listas de autorizações por usuário ou por recurso
- com sistemas multi-multidomínio
 - registro de todos os usuários possíveis
 - » sobrecarga para cada administrador
 - registro de um "usuário" por instituição do grupo
 - » dificuldade de fazer auditoria
- no Globus: mapeamento de usuários remotos a locais
 - tipicamente um usuário ppor grupo remoto



autorizações

- crescimento da idéia do uso de papéis
 - atributos



autorizações

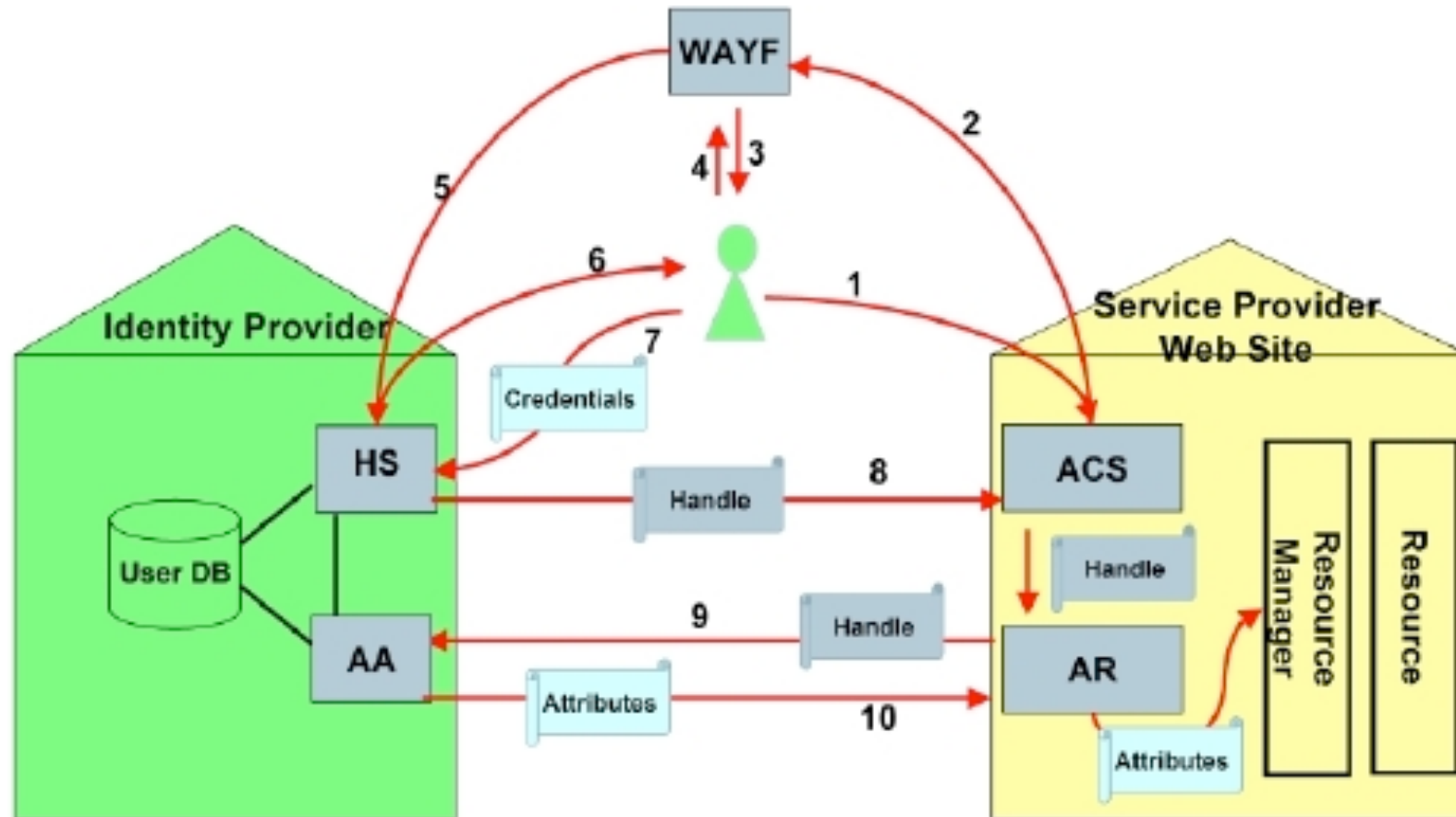


serviços de atributos

- autorizações baseadas em atributos:
 - professor, aluno, sócio IEEE, ...
- quem vai fornecer esses atributos?
- importância do anonimato
 - não sobrecarregar cada serviço acessado
 - anonimato tradicional: não divulgar mais do que o necessário
- serviço Shibboleth: iniciativa Internet2
 - integrável apenas com serviços acessados via http
 - projeto de integração com Globus



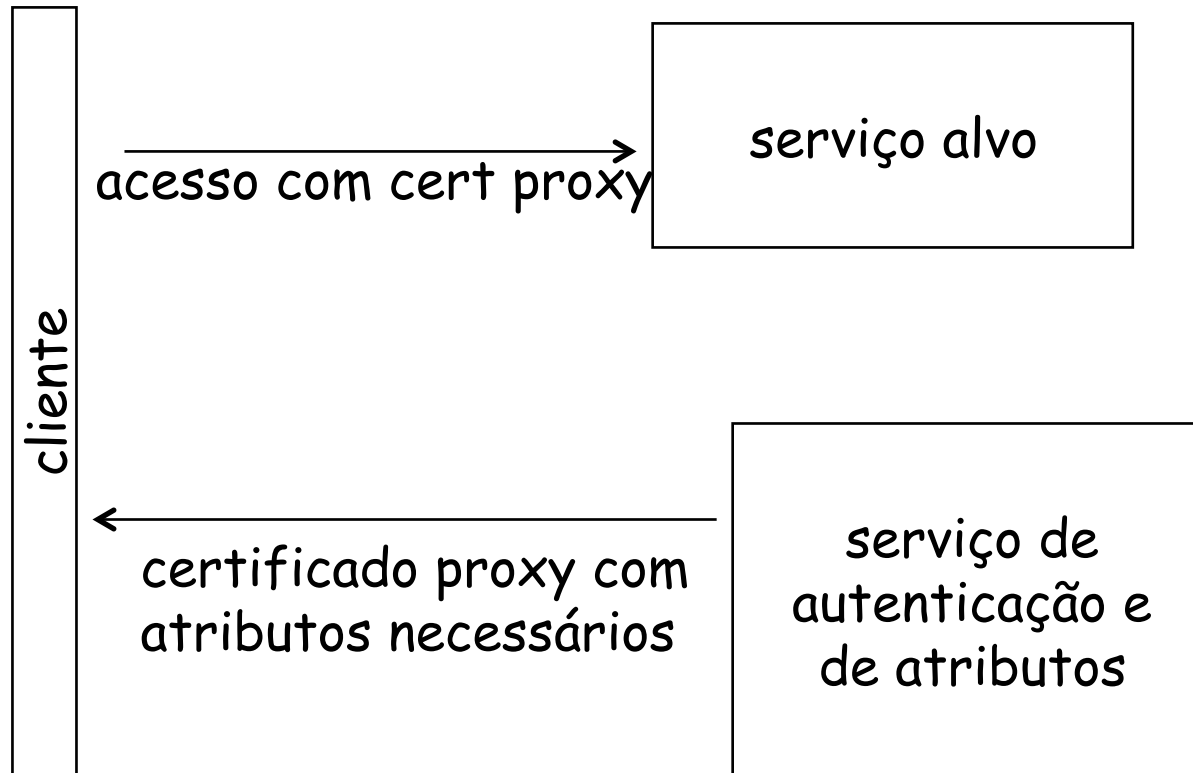
Shibboleth



© SWITCH



vantagens do modelo shibboleth



- serviço alvo pode receber apenas atributos relevantes
- serviço de atributos pode fornecer dados para auditoria

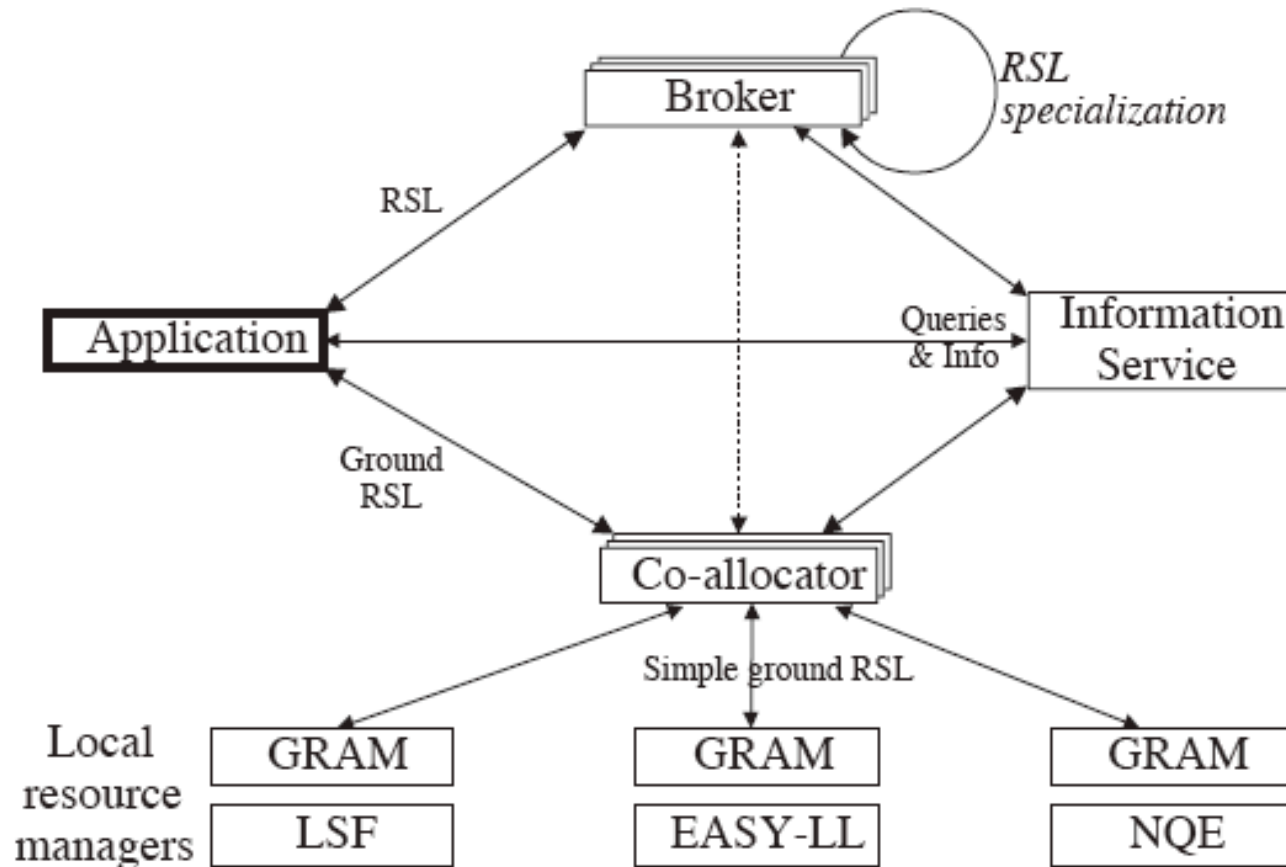


voltando ao Globus

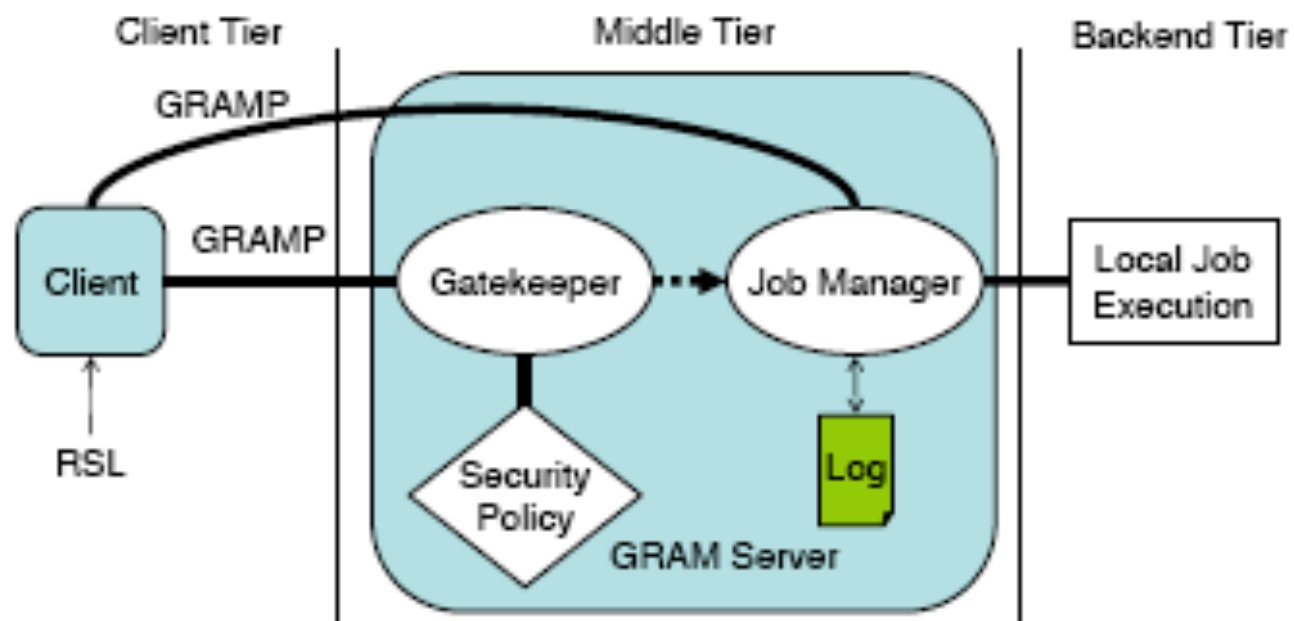
- outra questão de segurança está relacionada à proteção do ambiente de execução
- uso de *sandboxes*
- monitores de máquinas virtuais



Gerência de Recursos - 1



Gerência de Recursos - 2



GRAM - gerência de recursos

- comando *globusrun* e APIs
- Resource Specification Language (RSL)
- daemon gatekeeper
- gerente de tarefas



uso de recursos em grades

- escalonamento
 - escolha da melhor máquina
- reserva de recursos
 - negociação com sistema que gerencia máquina
- oportunismo (ou *scavenging*)
 - uso de máquinas que não estão sendo requisitadas localmente



resource specification language

- linguagem usada para descrever tarefas
- string RSL passada para globusrun tem info sobre:
 - arquivo executável
 - argumentos
 - stdin, stderr, stdout

```
& (rsl_substitution = (EXECDIR "/bin"))  
(executable = $(EXECDIR)/ps )  
(arguments=ef)(directory=/tmp)  
(stdout=/tmp/temp)(count = 1)
```

```
% globusrun -r evelyn.nas.nasa.gov\  
'&(executable=/bin/echo) (arguments="Hello, Globus world.")'
```



gatekeeper

- “porta de entrada” de um domínio
 - ou um por máquina
- autenticação de usuário
 - uso da infraestrutura GSI
- disparo de um gerente de job
 - delegação de autoridade para se comunicar com cliente
 - possível criação de ambiente de execução protegido
 - » sandbox
 - escalonadores específicos em cada domínio

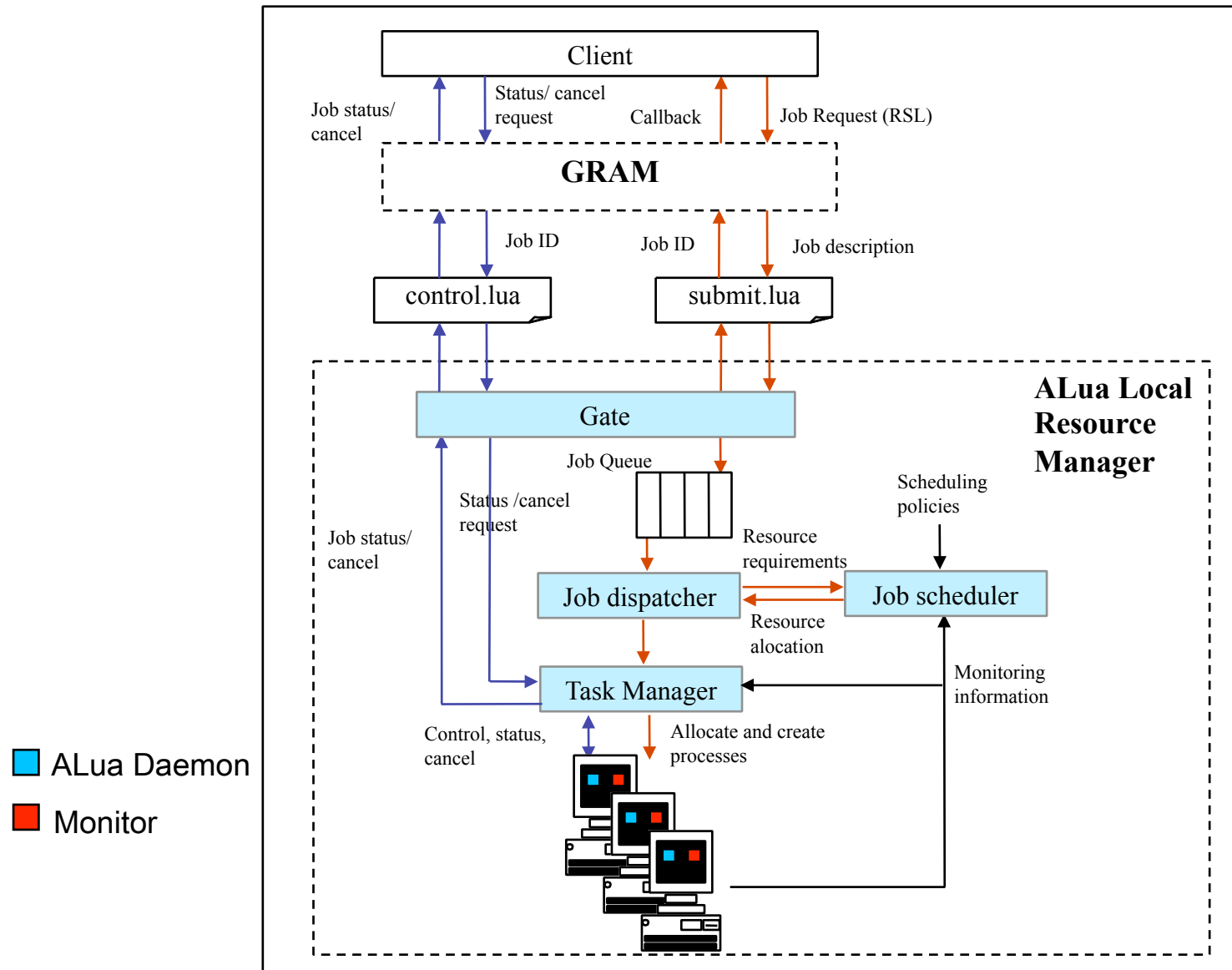


job manager

- interpreta a string RSL passada pelo cliente
- aloca requisições de jobs a gerentes locais
 - interação com escalonadores
- envia callbacks a clientes se necessário
- recebe pedidos de status e de cancelamento de clientes
- envia resultados para clientes
 - uso de GASS se necessário



ex. gerente: Local Resource Manager

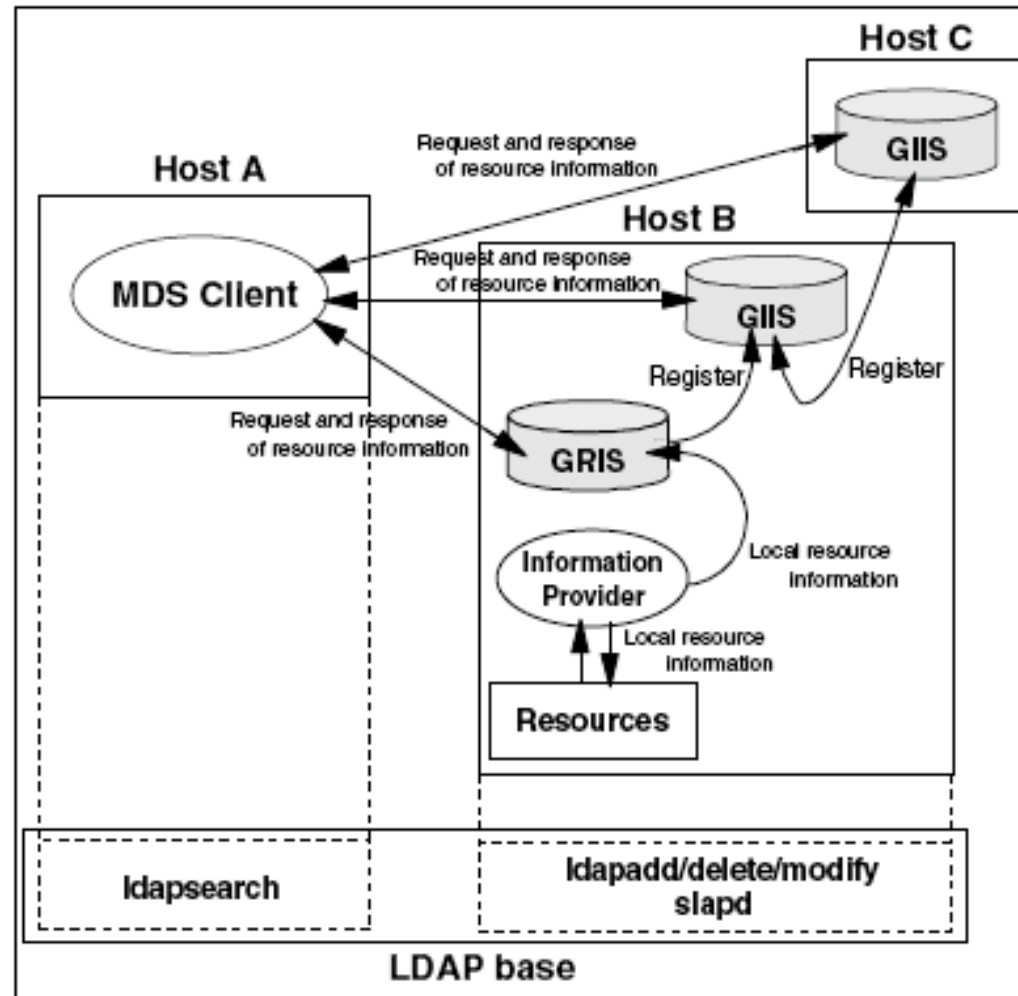


informação sobre recursos

- MDS (monitoring and discovery service)
- informação estática
 - sistema operacional
 - cache. número de processadores, total de memória física e virtual
 - dispositivos, serviços
- informação dinâmica
 - média de carga (fila de prontos)
 - total de espaço em disco, total de espaço livre
 - memória física disponível, memória virtual disponível,



MDS



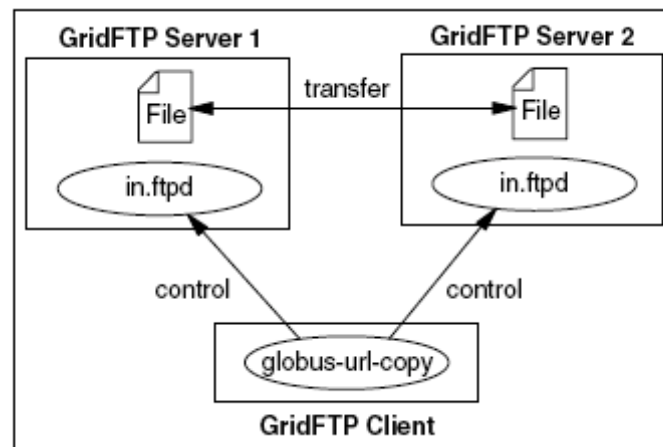
MDS

- provedores de informação
 - específicos de cada sistema
- GRIS
 - repositório local a cada máquina
- GIIS
 - repositório com infos sobre várias máquinas
- repositórios eram servidores LDAP, agora WS



transferência de dados

- diversos serviços propostos ao longo das versões
- mais estável: GridFTP
 - uso de GSI
 - otimizações de transferência
 - » múltiplos streams TCP
 - » striped data transfer
 - transferências iniciadas por terceiros



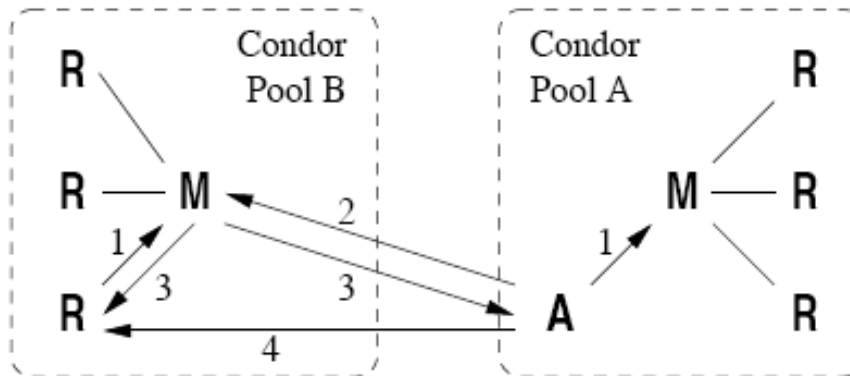
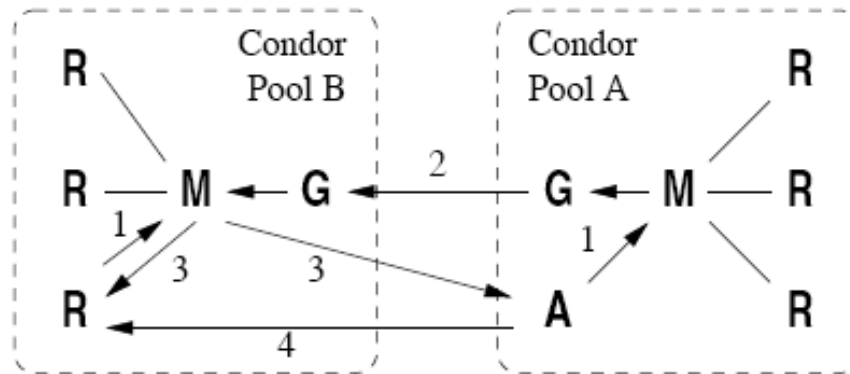
ferramentas de grade

- Globus: mecanismos bastante baixo nível que podem ser usados modularmente
 - complexidade
 - mudanças entre versões
- outras ferramentas
 - “stand-alone” - ferramentas completas
 - uso do Globus como base
 - » facilidades de padronização entre ambientes
 - sistemas ligados a grades específicas (testbeds)



voltando ao Condor

- tínhamos falado em gateway flocking e direct flocking



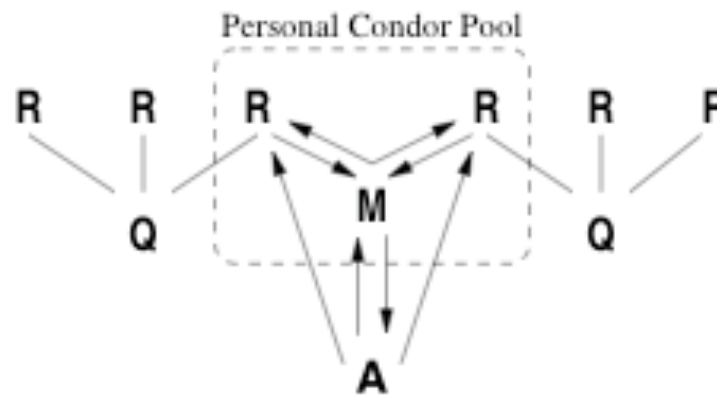
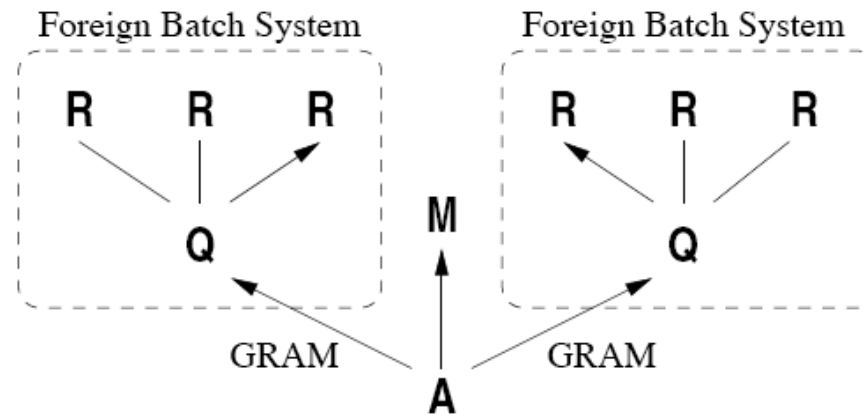
Condor e grades

- integração com o Globus para fazer os acessos interinstitucionais
 - protocolos de segurança
 - acesso padronizado a sistemas remotos



Condor e grades

- uso do Condor "por cima" do Globus: *gliding in*



voltando ao ProActive

- suporte para descritores que retiram a info sobre máquinas físicas de dentro do programa

Component Name: C3D-Dispatcher-Renderer

Dependencies:

Provides: class C3DDispatcher

Needs: <none>

Use: class C3DUser

VirtualNodes:

Dispatcher RegisteredIn RMIregistry, Globus

Renderer1

Renderer2

Renderer3

Renderer4

Mapping:

Dispatcher --> DispatcherJVM

Renderer1 --> JVM1

Renderer2 --> JVM1

Renderer3 --> JVM2

Renderer4 --> JVM2

JVMs:

DispatcherJVM = Current // (the current JVM, running the main),

JVM1 = //lo.inria.fr/ Protocol rsh

JVM2 = //ClusterSophia.inria.fr/ Protocol

LSF <1> VIA galere1JVM

galere1JVM = //galere1.inria.fr Protocol SSH

- protocolos de acesso podem ser integrados a mecanismos de segurança Globus



ProActive

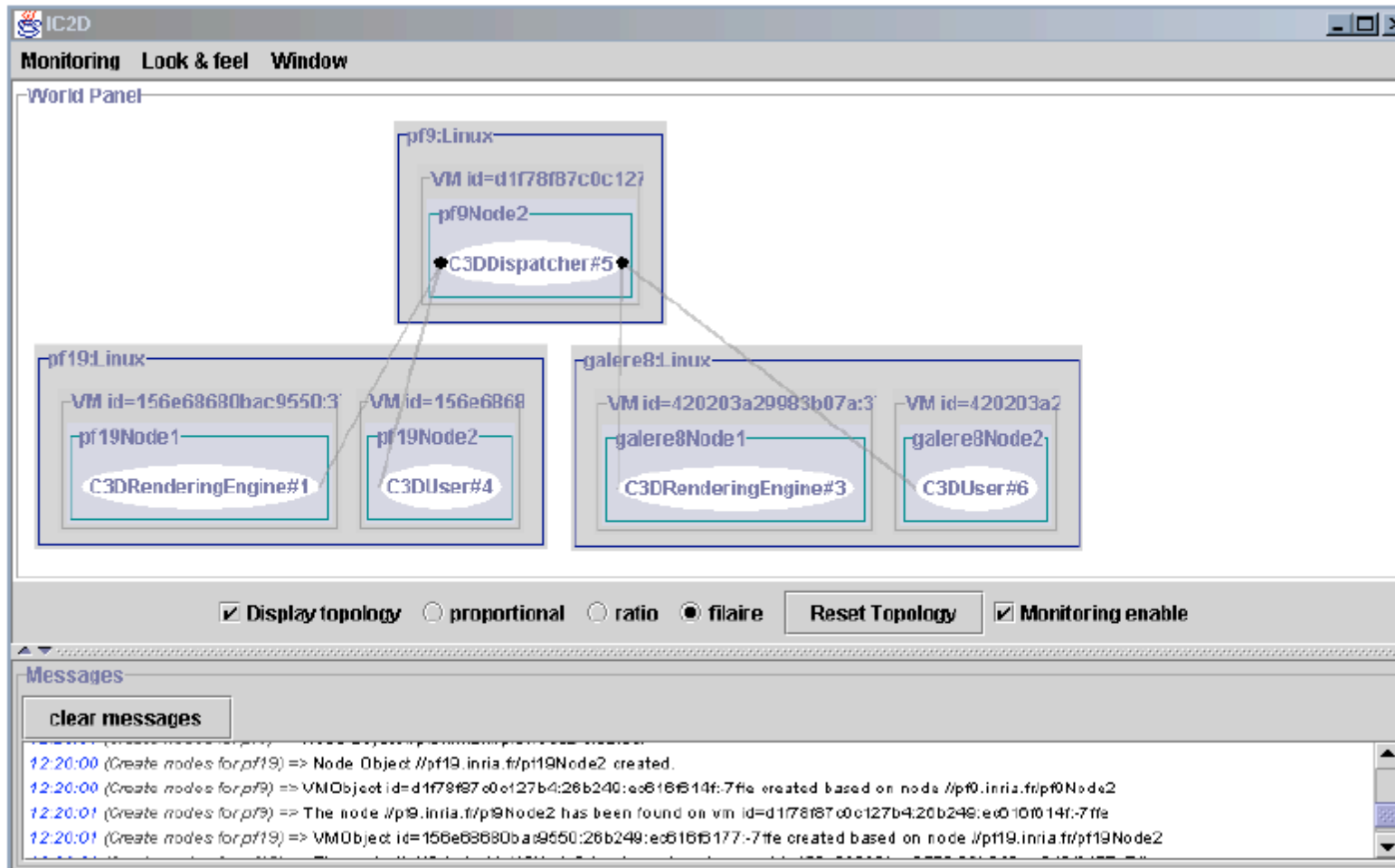
- no programa:

```
ProActiveDescriptor pad = ProActive.  
    getProActiveDescriptor("file:.ProActiveDescriptor");  
VirtualNode vn = pad.activateMapping("Renderer1");  
    // "Renderer1" is the virtual node name described  
    // in the XML-descriptor. It triggers the  
    // JVM on which "Renderer1" is mapped to.  
Node node = vn.getNode();
```

- projeto originalmente baseado na plataforma distribuída Java
- fora do programa: configuração por XML
 - » urgh!



ProActive: interação

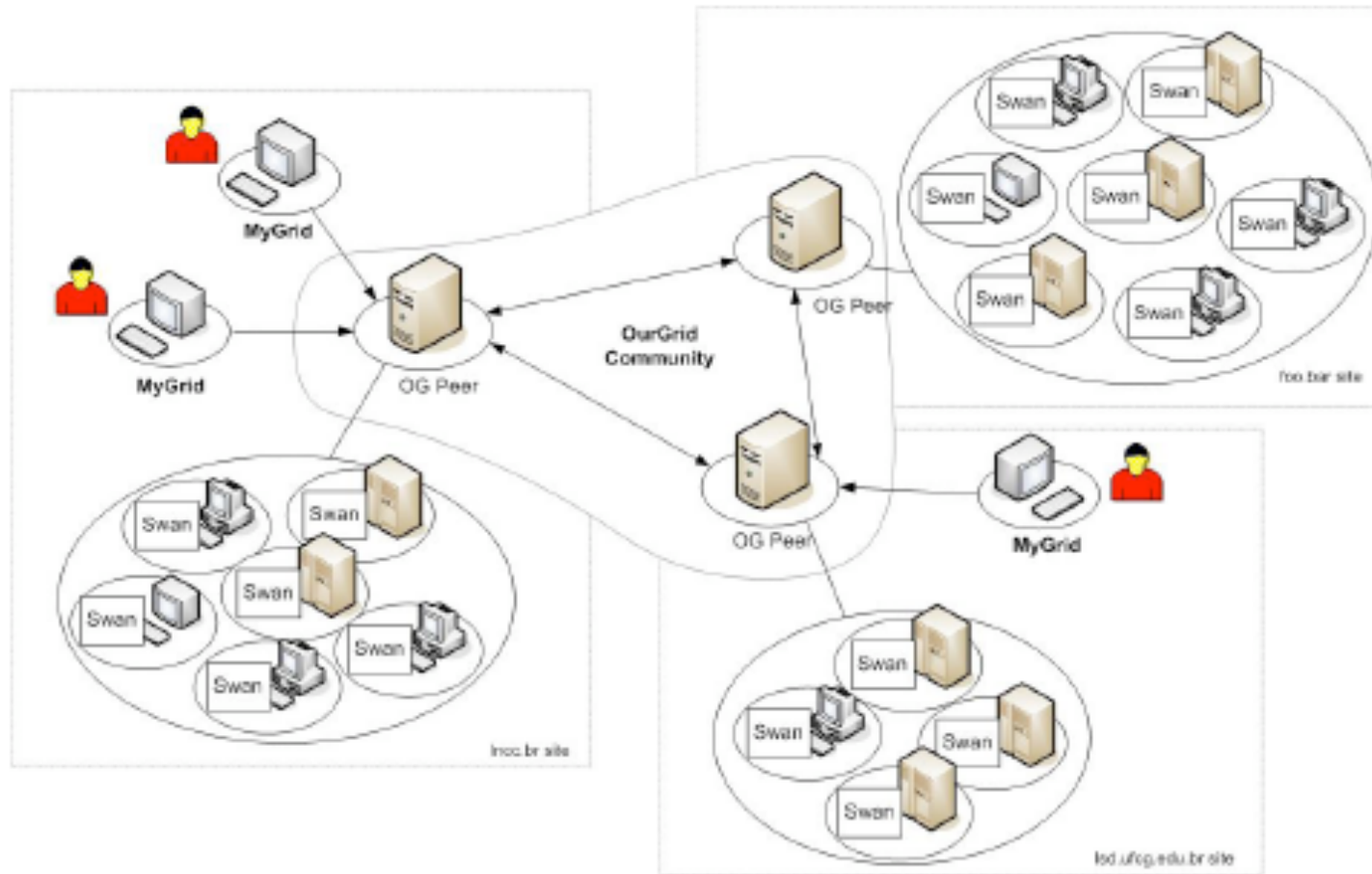


OurGrid

- <http://www.ourgrid.org/>
- Universidade Federal de Campina Grande
- suporte a execução de programas estilo bolsa de tarefas



arquitetura ourgrid



arquitetura OurGrid

- mygrid: local à máquina de cada usuário
 - lê descrição de job
 - contacta ourGrid e submete job
- ourgrid: comunidade de *peers*
 - cada um responsável por um conjunto de máquinas
 - recebem pedidos de usuários e alocam às suas máquinas ou submetem a outros peers
 - » sandboxing



tarefas OurGrid

- cada tarefa é dividida em
 - initial
 - Grid
 - final
- programador deve codificar a organização inicial do ambiente remoto e a coleta de resultados
 - variáveis de ambiente \$PROC, \$JOB, \$STORAGE, ...



máquinas no OurGrid

- têm que saber responder às operações
 - ping() - verifica disponibilidade da máquina
 - run() - inicia processo
 - putFile() - armazena dados
 - getFile() - recupera dados
 - kill() - finaliza processo



preocupação com *free-riding*

- sistemas de reputação
- uso de redes de favores
- sistema onde cada peer toma decisões baseando-se em informações locais



segurança - uso do xen

- tarefa submetida a máquina remota executa em sandbox baseado no xen
- não permite comunicação com outras tarefas

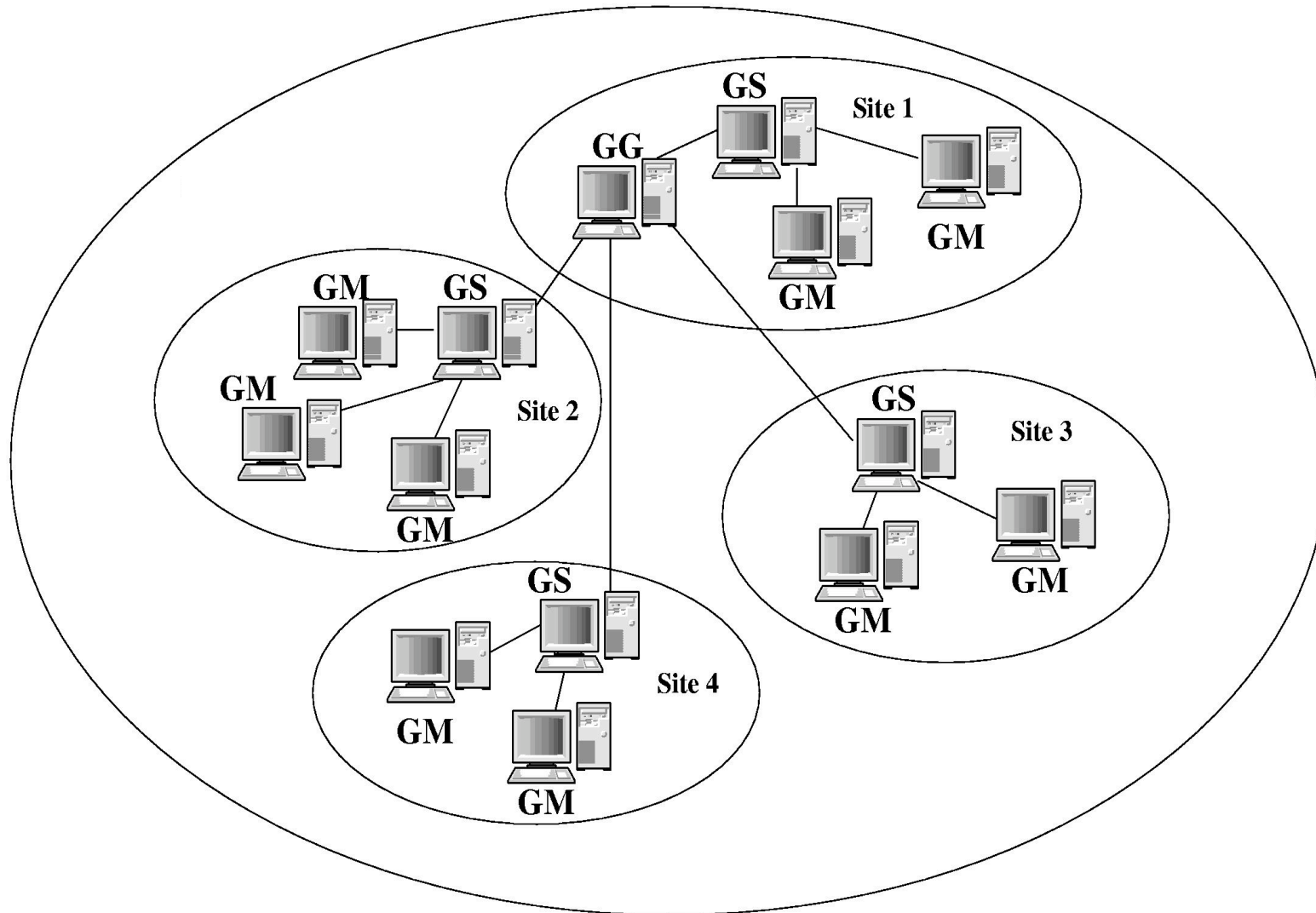


EasyGrid

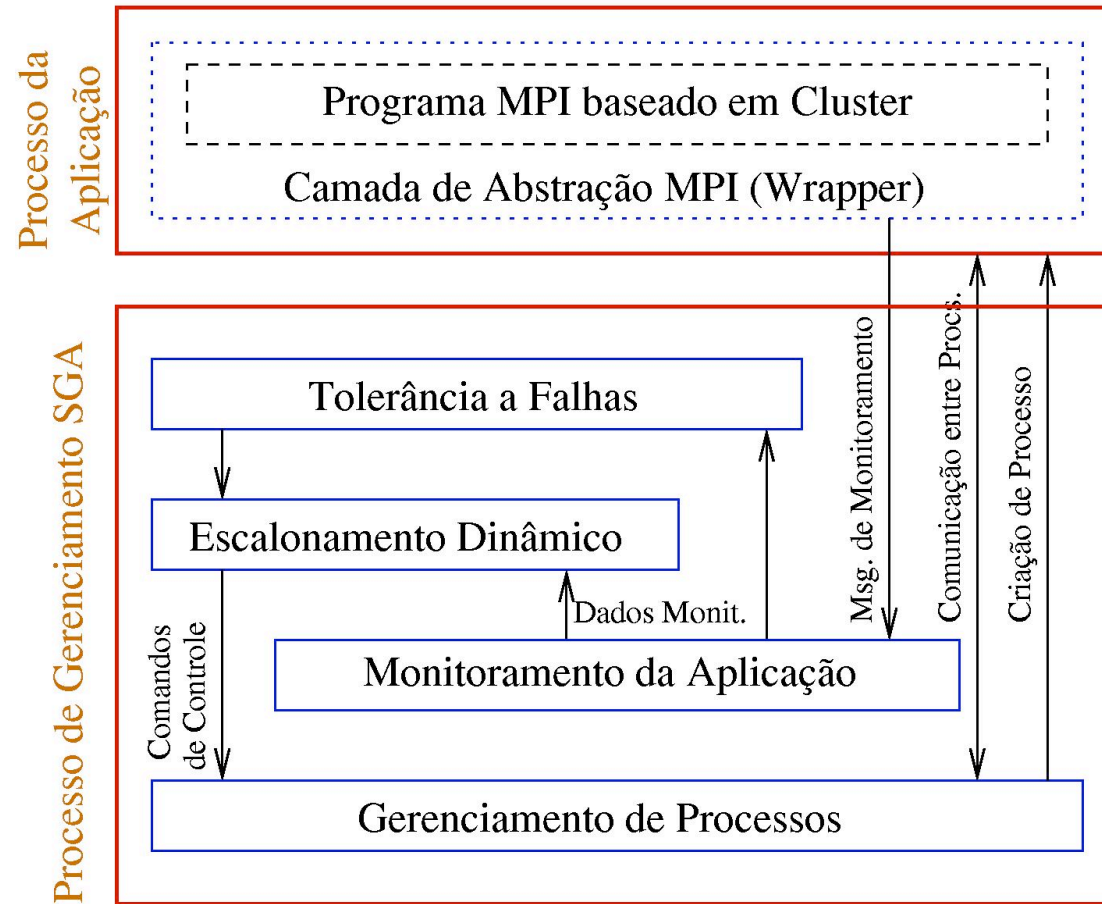
- <http://easygrid.ic.uff.br/>
- Universidade Federal Fluminense
- baseado no Globus
- escalonamento híbrido
 - informações sobre estado de máquina e tarefas adicionadas à saída de algoritmo de escalonamento estático
- preocupações com:
 - Escalonamento de Tarefas;
 - Tolerância a Falhas;
 - Monitoramento;



Hierarquia EasyGrid



Estrutura de um Processo Gerenciador



Modelo EasyGrid

- inicialmente é aplicado um algoritmo de escalonamento estático
- uso de GADs
 - grafos de dependências
- filas de processos alocadas a cada máquina
 - pendentes e prontos
 - em execução
- processos ainda não em execução transferidos entre máquinas



e mais outros muitos...

