

# Análise de frameworks para construção de portais para ambientes de computação em grade e sua aplicabilidade no AppMan

Michel David da Costa, Patrícia Kayser Vargas Mangan

Curso de Ciência da Computação – Centro Universitário La Salle (Unilasalle)  
Av. Victor Barreto, 2288 – 92010-000 – Canoas/RS – Brasil  
michel.david@gmail.com, kayser@unilasalle.edu.br

**Abstract.** *This paper investigates the grid resources access problem that occurs when performing direct access to the AppMan's grid environment. Using the resources managed by AppMan requires specific knowledge about the prototype and user authentication in a grid node. Besides, there is a lack of integrated monitoring information view. To deal with these problems, we use the strategy of build up a grid portal for the investigation purpose. It is also specified the way in which the solution will be validated and the expected results.*

**Resumo.** *Este artigo investiga o problema de dificuldade de acesso a recursos de grade que ocorre ao tentar acesso ao ambiente de grade AppMan de forma direta. O uso de um ambiente gerenciado pelo AppMan exige conhecimentos específicos sobre o protótipo e autenticação em um nó da grade. Também não há uma visão integrada das informações de monitoramento de aplicações e recursos. Para tratar estes problemas é utilizada a estratégia de construção de um portal para acesso à grade como proposta de investigação. Também é apresentada a forma como a solução proposta será validada.*

## 1. Introdução

De acordo com Foster (1998), *Grids* (grades) são ambientes que permitem o compartilhamento de recursos computacionais de forma consistente entre uma comunidade pervasiva com poucos interesses em comum, dessa forma oferecendo uma solução barata para capacidades computacionais avançadas. Grades se enquadram em duas categorias: *Computational Grids*, focadas em processamento massivo, ou *Data Grids*, focadas em armazenar grandes quantidades de informações. Para fazer o uso destes recursos compartilhados é necessária um infra-estrutura de hardware e software. Inicialmente, foi definida uma arquitetura chamada de *Grid Architecture* (FOSTER, 2001) para implementar esta infra-estrutura.

Nesta arquitetura é possível o compartilhamento de diversos componentes, que permitem a colaboração entre o desenvolvimento e aperfeiçoamento dos mesmos, incluindo a interação entre estes. Com isto foi criado um portfólio cada vez maior de aplicações que fazem parte do nível de usuário da grade, onde são, na maior parte das vezes, usadas via comandos de *script*, a partir de um nó gerenciador da grade, o que vem a criar uma restrição de acesso. A partir desta limitação de uso foi notada a necessidade

da criação de portais para grades, os quais produzem uma interface amigável para o usuário e possuem certa integração com o *middleware* de usuário que, por sua vez, permite a comunicação com a camada de baixo nível da grade.

Segundo Novotny (2000) portais para grade são aplicações web que permitem, através do uso de um navegador de Internet, fornecer uma interface amigável para a interação do usuário com o *middleware* da grade. Esta interface deve disponibilizar uma forma de login na segurança local e para os serviços da grade, além de uma visualização do software e dos recursos de hardware da grade (HAILI *et al*, 2005).

Um portal de grade deve ser capaz de se adaptar a qualquer uma das categorias de grade, seja computacional ou de dados. Com o crescente interesse e desenvolvimento de portais que integrem ambientes de grade também se tornaram necessárias a padronização e interoperabilidade destes portais e serviços da grade (GGF, 2008). Desde as primeiras propostas de portais de grade, tanto a tecnologia de *middleware* quanto a de portais evoluíram. De modo geral, pode-se dizer que ambos evoluíram no sentido de implementar as funcionalidades através das tecnologias de componentes e serviços. Também surgiram trabalhos com objetivo de facilitar o controle para as aplicações que rodam sob o *middleware* de grade, sendo referidos com Gerenciadores de Aplicação. Além disso, diversos trabalhos foram propostos a fim de disponibilizar *frameworks* que facilitassem a construção de portais.

Embora existam vários *frameworks* (ASADZADEH *et al*, 2005) para a construção de portais de grade, pouco foi feito em termos de estudos sistemáticos sobre estes *frameworks* de modo a facilitar que os desenvolvedores de portais escolham o *framework* mais adequado para as suas necessidades. Por exemplo, no manual técnico de Shankar (2006) foram realizadas comparações entre portais com fins de determinar o *framework* ideal para uma aplicação, onde apresenta comparações entre o portal OGCE, o *framework* GridPort Toolkit e Gridsphere. Zhang (2007) também realiza comparações entre o portal OGCE e o *framework* Gridsphere, buscando suas limitações e motivações para a evolução das soluções de portal para grids.

Dado o contexto apresentado, este estudo busca sanar as necessidades de comparação entre os *frameworks* para a construção de portais. Uma comparação sistemática facilita a escolha do desenvolvedor sobre qual *framework* permite solucionar de forma mais adequada o problema de dificuldade de acesso detectado em *middlewares* de grade. Além disso, este trabalho busca como estudo de caso analisar o gerenciador de aplicação AppMan verificando suas necessidades com relação a interação com os usuários e os administradores via portal. Cabe lembrar que o AppMan na versão atual usa de scripts Linux para prover suas funcionalidades e é um gerenciador de aplicação implementado a partir do modelo GRAND. A motivação para a realização deste trabalho é permitir o acesso de usuários sem grandes conhecimentos técnicos ao ambiente de grade.

Na próxima etapa deste artigo será apresentada a anatomia do ambiente de grade, explicitando sua arquitetura inicial e a evolução desta arquitetura, em um *middleware* bastante utilizado em nível mundial, o Globus Toolkit. Como um segundo exemplo, será apresentado o modelo GRAND e o protótipo AppMan, que será o estudo de caso deste trabalho. Finalmente, apresenta-se as formas que estes *middlewares* possuem para comunicação com portais. Na Seção 3 serão apresentados *frameworks* que possibilitam

a construção de portais para grades, como o GPDK, GridPort e Gridsphere. A partir destes será realizada uma comparação entre os recursos que cada um oferece e o que será necessário para a integração com o AppMan, seguido da proposta de uma solução para o problema. Comprometido com tais objetivos, este artigo classifica-se como uma pesquisa de natureza aplicada, com objetivos exploratórios, adotando assim o procedimento técnico de estudo de caso do AppMan. A Seção 6 encerra este artigo, apresentando a conclusões e trabalhos futuros.

## 2. Infra-estrutura de software para grades

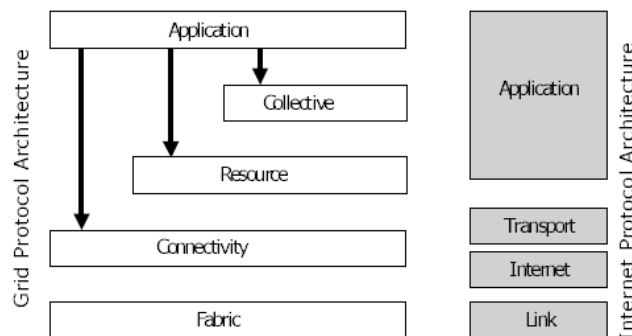
Grades contam com uma infra-estrutura para realizar as tarefas pertinentes à sua gerência. Nesta infra-estrutura, *middlewares* são partes importantes na unificação do controle da grade, onde são desenvolvidas as principais integrações entre os serviços, fornecendo assim acesso a estes para os usuários da grade.

O serviço de gerenciamento de aplicação visualiza a aplicação como um todo, e é encarregado de enviar tarefas específicas ao serviço de submissão de tarefas. Outro serviço são os *Resource Management Systems* (RMS), que são responsáveis pela comunicação direta com os recursos, contribuindo com a submissão de tarefas e escalonamento da execução das tarefas.

Para compreender melhor o que é um *middleware* de grade, apresenta-se a seguir a visão de duas versões do Globus Toolkit que resume as duas principais tecnologias empregadas na construção das atuais infra-estruturas em uso corrente. Também será apresentado o protótipo do modelo GRAND, o AppMan, onde serão vistos os principais componentes e formas de interação. Por último nesta seção será realizada uma comparação entre as formas de comunicação para interface de usuário oferecida em *middlewares* de grade.

### 2.1 Arquitetura de grade

De acordo com Foster (2001), a arquitetura não define os protocolos a serem usados ou serviços, mas identifica os requisitos para classes de componentes. A arquitetura proposta organiza os componentes em camadas, cujo comparativo desta com as camadas dos protocolos de Internet pode ser visualizado na Figura 1.



**Figura 1. As camadas da arquitetura de grade e sua relação com a arquitetura de protocolo da internet (FOSTER, 2001).**

A camada *fabric* provê recursos como ciclos computacionais, sistemas de armazenamento, catálogos, recursos de rede e sensores de forma compartilhada à grade.

Em *connectivity* são definidos os protocolos essenciais para comunicação e autenticação necessários em transações específicas para a rede de grade. A camada *resource* se baseia nos protocolos de comunicação e autenticação para definir protocolos, APIs e SDKs para oferecer formas seguras de negociar, inicializar, monitorar e controlar as operações compartilhadas em cada recurso. As implementações dos protocolos da camada *resource* executam funções da camada *fabric* para acessar e controlar os recursos locais. Na camada *collective* são focadas as interações com um único recurso, contendo protocolos e serviços que buscam trabalhar os recursos de forma global. Este tipo de visão norteou a implementação das primeiras infra-estruturas de grade, e em particular a versão 2 do Globus Toolkit.

Com a mudança de foco para a colaboração global distribuída, a arquitetura de grades evoluiu para uma abordagem orientada a serviços (SOA) e camadas de informação (ROURE *et al*, 2003). Seguindo este conceito a arquitetura do projeto Globus Toolkit 4 aborda esta arquitetura de serviços, chamada *grid services*, como será apresentado na próxima seção.

## 2.2. Globus Toolkit 4

Dentre os projetos de *middleware* para ambientes de grade, o projeto Globus Toolkit (GT) é o mais usado e conhecido no meio acadêmico. Como foi apresentado anteriormente, este *middleware* usa o modelo de arquitetura mais atual, onde parte para uma abordagem SOA. Este projeto não é auto-contido, ou seja, ele é composto pela junção de diversos componentes que, ao se integrarem ao GT, permitem a gerência de escalonamento e distribuição das tarefas na grade (VARGAS, 2007).

A base de serviços do GT é provida através de interações similares às interações de Web Services. Esta técnica usa padrões abertos, XML e protocolos abertos para permitir esta comunicação (SUN, 2006). Por padrão um *Web Service* é *stateless*, ou seja, não guarda estado, tornando-o mais flexível. De acordo com a padronização do *Web Service Resource Framework* (GLOBUS, 2004), Web Services são adaptados para disponibilizar *Statefull Web services*, ou seja, um *Web Service* que mantém seu estado entre as chamadas. O WSRF define a infra-estrutura de um *Web Service* associado a um recurso, que pode guardar informações de estado de certo recurso computacional, apresentando também um identificador único de recurso. Esta infra-estrutura foi adotada pelo *Open Grid Services Architecture* (GGF, 2002) para definir um *Grid Service*, parte essencial da arquitetura usada pelo Globus.

O projeto Globus define cinco categorias ou áreas de domínio para seus componentes (SOTOMAYOR e CHILDERS, 2006). Estas categorias servem para classificar os componentes, permitindo assim a escolha de quais serão usados no ambiente, pois, como o Globus não é auto-contido, é necessário o uso de componentes diferenciados, dependendo da necessidade e das características do ambiente.

Caracterizando a forma de acesso aos *middlewares* de grade é possível identificar as dificuldades do processo para um usuário leigo. A partir desta premissa se deu a justificativa para a construção de portais de grade. No capítulo seguinte serão apresentados trabalhos relacionados à criação de portais para ambientes de grade, os quais buscam facilitar a construção de tais portais e, assim, serão úteis na resolução do problema em estudo.

### 2.3. Estudo de Caso: AppMan

O ambiente de gerenciamento de aplicações de grade AppMan hoje conta com um modo de submissão de tarefas hierárquico, criado a partir do modelo *GRAND* (Grid Robust Application Deployment) (VARGAS, 2006).

O AppMan possui características bastante peculiares, como o uso de um *Application Manager* que envia a tarefa a um ou mais *Submission Managers* que, por sua vez, encaminham para os *Task Managers*, que executarão as tarefas. Dado o modelo de arquitetura *GRAND* usado pelo AppMan é possível notar que não temos hoje um framework para construção de portais web que possua qualquer forma de interação com este modelo.

A seguir são apresentadas as formas com que este *middleware* efetua a submissão de arquivos, monitoração de aplicações e a monitoração de recursos no ambiente da grade.

#### 2.3.1. Submissão de aplicações

Atualmente a submissão de aplicações se dá através da geração de um Grafo Acíclico Direto (*DAG*) da aplicação a ser executada. A linguagem usada para descrever a aplicação é a *GRID-ADL*, a qual possibilita a criação deste *DAG* em memória e o devido particionamento conforme suas dependências. Este é construído baseando suas dependências nos arquivos de entrada e saída de cada parte da aplicação com o auxílio de um script de *shell*.

#### 2.3.2. Monitoramento de aplicações

O modelo *GRAND* oferece um serviço de monitoramento através do serviço de diretório chamado *GRID-AMDS* (Grid Application Monitoring Directory Service), o qual segue a especificação *GMA* e oferece quatro funções: adicionar, atualizar, remover e pesquisar (VARGAS, 2006).

A função *add* (adicionar) adiciona uma entrada no serviço de diretório. Esta entrada é composta pelo identificador único da aplicação, nome do usuário que iniciou a aplicação, arquivo de descrição da aplicação em *GRID-ADL*, total de tarefas e uma lista de tarefas, contendo identificador da tarefa, identificador das tarefas predecessoras, estado e o nó em que a tarefa está vinculada.

A função *update* (atualizar) pode atualizar submissões de tarefas para indicar mudanças de estado ou atualização de informações do nó. A função *remove* (remover) remove o registro da aplicação, para os casos onde não é mais interessante monitorar a mesma. E a função *search* (pesquisar) busca informações sobre o estado da aplicação.

Na implementação do modelo *GRAND*, através do protótipo AppMan, este monitoramento é feito de uma forma simplificada. Estas operações de busca, ao invés de serem obtidas através de um serviço, são obtidas pela consulta a um arquivo de comum acesso, o qual contém os dados. Isso garante independência entre a submissão e o monitoramento, e abre espaço para a implementação futura de um serviço mais sofisticado de monitoramento.

### 2.3.3. Monitoramento de recursos

O modelo GRAND se aproveita dos projetos existentes como Ganglia (MASSIE *et al*, 2004) e MonALisa (NEWMAN *et al*, 2003) para monitoração de recursos, a fim de evitar a criação de novas regras de firewall e modificações administrativas. Entretanto, o AppMan oferece o serviço de monitoração da execução da aplicação como um serviço de monitoramento próprio de recursos.

As informações sobre a disponibilidade dos recursos serão buscadas através de pesquisas no serviço de diretório, providenciado pelos administradores locais do nó da grade. O modelo GRAND se baseia em eventos fornecidos pelos serviços de monitoração atuais (GUNTER, MAGOWAN, 2004) para suas decisões a respeito da alocação de tarefas. Os principais tipos de eventos fornecidos são *CPU load*, correspondente ao tempo livre de processamento e *available memory*, correspondente à quantidade de memória disponível.

Para monitorar os recursos foi implementado um sistema de monitoramento denominado MoonGrid (LEMOS *et al*, 2007), que é baseado em um modelo hierárquico para monitoração de hardware e software. Este implementa a coleta de dados através de um componente sensor através de um serviço. A comunicação é realizada através documentos XML, disponibilizados por webservices ao gerenciador de tarefas. Esta solução foi desenvolvida para trabalhar em nós rodando na plataforma Microsoft, através do *Microsoft .NET Framework*. Para utilizar este sistema no portal do AppMan deverá ser implementado um adaptador, sendo que a forma de comunicação é interoperável com a plataforma java do AppMan.

### 2.4. Interfaces de comunicação

Conforme explicitado, *middlewares* de grade estão evoluindo para atender serviços globais. No contexto do projeto Globus, é proposta como forma de comunicação a plataforma WSRF. Esta forma de comunicação já é empregada desde a versão 3 do Globus Toolkit, embora alguns serviços ainda não são providos via WSRF por questões estratégicas, como por exemplo, a transmissão de arquivos.

Já o protótipo AppMan utiliza-se de técnicas de script de *shell* como plataforma de comunicação com o usuário. Esta técnica foi empregada no Globus Toolkit até a versão 2, e existem portais que utilizam-se destas técnicas para prover uma interface de acesso web ao usuário final.

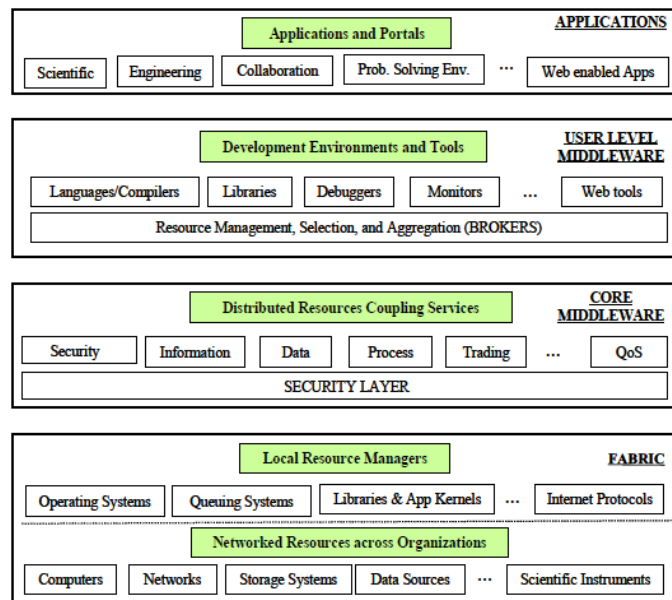
Para permitir facilidade de acesso a usuários na web são usados portais de grade, e estes são atualmente construídos a partir de frameworks para a construção de portais, onde estes facilitam o desenvolvimento dos portais, fornecendo meios para a interligação com os *middlewares* de grade. Trabalhos estes, como o GPDK, GridPort e Gridsphere, que serão apresentados na seção seguinte.

## 3. Frameworks para Construção de Portais de Grade

Inicialmente grades foram usadas como solução para pesquisas científicas e grandes empresas para o armazenamento de um volume de dados cada vez maior e o processamento dos mesmos. Dada a solução, primordialmente o acesso ao *middleware* da grade era realizado através de terminais e comandos *shell*. Com uma comunidade

crescente demandando os serviços das grades, foram detectados problemas com relação ao monitoramento e à facilidade de acesso da comunidade a estes serviços.

Com o intuito de facilitar o acesso aos recursos da grade, foram criados portais para o acesso a diversos *middlewares* de grade. A organização dos ambientes de grade permitiu a integração deste tipo de *framework*, a fim de permitir a modularização e divisão das tarefas a serem empregadas. Na Figura 2 (ASADZADEH *et al*, 2005) abaixo podemos visualizar a divisão destas camadas de componentes.



**Figura 2: Arquitetura em camadas de grades e seus componentes**

De acordo com as camadas das arquiteturas de grade, os *middlewares* que devem suprir funcionalidade aos portais, entre outras formas de interfaces, se encontram na camada de usuário da grade, os quais também são responsáveis pela comunicação com as camadas de mais baixo nível da grade.

Com o grande crescimento de grades e diferentes tipos de componentes para submissão de tarefas, escalonamento e gerência de recursos, os *middlewares* de grade estão preparados para adaptações, conectando-se assim a componentes que possuem a mesma funcionalidade, o que é abstraído pelo *middleware* ao fornecer o serviço ao portal.

Dentre os inúmeros frameworks para construção de portais de grade, será dada maior atenção aos frameworks mais conhecidos no meio acadêmico, que são os seguintes:

- Grid Portal Development Kit (GPDK);
- Grid Portal Toolkit (GridPort);
- Gridsphere;

### 3.1. GPDK (*Grid Portal Development Kit*)

O GPDK (NOVOTNY, 2001) procura criar uma interface web fácil de usar e que permita o uso de componentes comuns entre portais a fim de permitir a autenticação de usuários em recursos remotos, além de ajudar na tomada de decisões ao agendar aplicações. A interface permite ao usuário visualizar informações pertinentes dos recursos, as quais foram obtidas e estão guardadas em bases de dados remotas. Este framework também permite a criação de perfis, que agrupam e monitoram aplicações enviadas e o resultado das mesmas. Também são oferecidas estruturas básicas, definidas pelo CoG (*Java Commodity Grid*), para interação com os *middlewares* de cliente da grade, como submissão de aplicações, transferências de arquivos e busca de informações. Com o uso destas estruturas básicas se garante uma grande compatibilidade com ambientes de grade, e possibilita a interoperabilidade entre os componentes desenvolvidos, permitindo a estes a manipulação de dados vitais ao funcionamento da grade.

É possível usar o componente *MyProxy* em qualquer momento, internamente no portal GPDK. Este componente permite o uso de credenciais remotas ao realizar qualquer tarefa do portal, assim evitando o tráfego desnecessário e vulnerável de chaves ou certificados de segurança.

Este framework adota *Java Server Pages* e *Java Beans* como padrão para a construção de páginas web, sendo possível realizar a disponibilização do portal no *Tomcat*, um servidor de aplicação bastante simplificado e código aberto, da *Apache Software Foundation*. Também são fornecidas APIs, compatíveis com o CoG, que permitem acesso a diversos recursos do *Globus Toolkit*, como upload de dados via *GSIFTP* (*Grid Security Infrastructure File Transfer Protocol*) e pesquisa em servidores *LDAP* (*Lightweight Directory Access Protocol*).

### 3.2. GridPort Toolkit

O projeto GridPort Toolkit (THOMAS *et al*, 2001) foi construído a fim de fornecer um meio único de comunicação, além de permitir a integração de componentes desenvolvidos em outros projetos. Inicialmente o GridPort foi usado como *engine* de portal para o NPACI (*National Partnership for Advanced Computational Infrastructure*) em 1999 e desde então passou por diversas mudanças para permitir maior integração entre os componentes de diversos ambientes de grade.

Este framework, para atingir seus objetivos de acesso universal e possuir uma infra-estrutura escalável, permite que múltiplas aplicações de portais possam acessar a mesma instância da biblioteca do GridPort. O login único foi desenvolvido contanto que todos os portais sejam providos pelo mesmo domínio da internet. A implementação do framework por parte do servidor é feita usando a linguagem de programação *Perl/CGI*, pois é facilmente instalável e é suportada em todos sistemas HPC (*High Performance Computing*).

Através da camada de serviços, é possível oferecer serviços comuns aos componentes que estão executando em outros ambientes que, no caso, foi testado no NPACI. O GridPort é um projeto de framework para portais que trabalha de forma a permitir a agregação de outras aplicações menores, chamadas *portlets*, as quais podem acessar os *middlewares* de cliente da grade a fim de gerar dados úteis para o usuário.



Através da arquitetura adotada pelo GridPort torna-se fácil criar plugins para a interação com serviços pré-existentes, como o serviço de submissão de tarefas Condor e GRAM.

O framework do GridPort é parte integrante do portal *TeraGrid User Portal* (DAHAN *et al*, 2006), o qual usa Web Services como forma de comunicação com o *middleware* do TeraGrid.

### 3.3. Gridsphere

O projeto Gridsphere (RUSSELL *et al*, 2006) faz parte dos projetos da GridLab, fundado pela *European Commission* (EU). Este projeto possui compatibilidade total com a especificação JSR168 (JAVA, 2003), é código aberto e livre para uso acadêmico e profissional e contribui com a comunidade de grade através de Gridportlets.

Para a construção de serviços de portlets é provido um framework de serviços que adota interfaces de serviço do WebSphere, da IBM (NOVOTNY *et al*, 2004) para permitir futura integração de portlets. As portlets de núcleo disponíveis por padrão são as seguintes (NOVOTNY *et al*, 2004):

- *Login/logout*: permite login baseado em usuário e senha;
- *Account Request*: requisição de uma conta e escolha de associação a grupos;
- *Account Management*: fornece a administradores e superusuários a habilidade de adicionar ou remover papéis aos usuários;
- *User Management*: fornece a superusuários a habilidade de aprovar ou negar requisições de contas e a administradores a habilidade de aprovar ou negar a requisição de acesso a um grupo;
- *Portlet Subscription*: adiciona e remove portlets da área de trabalho do usuário;

O container de portlets do Gridsphere é implementado como uma aplicação web simplificada, a qual necessita de um servidor web e algumas bibliotecas adicionais para tornar-se acessível, sendo assim facilmente inicializado.

O portal ePhysics (BEESON *et al*, 2005) é um exemplo de uso do portal Gridsphere, o qual foi integrado ao *middleware* Gridbus com RMI para comunicação e utiliza o Globus para submissão de tarefas.

## 4. Funcionalidades

Os *middlewares* de grade estão em constante evolução sempre que novas tecnologias são desenvolvidas e mais pessoas e empresas se envolvem no desenvolvimento da grade. Nesta evolução, o suporte às funcionalidades dos *middlewares* de grade se modifica rapidamente. Para permitir estas mudanças os frameworks para construção de portais também devem se adaptar, por isso serão estudadas as funcionalidades oferecidas e as funcionalidades buscadas para o uso com o *middleware* AppMan.

### 4.1. Funcionalidades oferecidas

Dentre as funcionalidades oferecidas é possível destacar a presença de componentes de núcleo pré-desenvolvidos e, de forma cada vez mais comum, o uso de portlets, de

acordo com a especificação JSR168, além da dependência de outros projetos e a compatibilidade entre portlets e para outros portais.

Dentre as características básicas dos portais, estão a forma de segurança aplicada na execução das tarefas, gerência dos recursos da grade, forma de submissão de tarefas, serviços de monitoramento providos, capacidades de transferência de dados e integração com *portlets*. Dentre os componentes pré-disponíveis para uso do portal estão a gerência dos usuários do portal, customização do portal, gerenciamento de tarefas, replicação de dados e componente de notificações.

Para esclarecer a diferença das funcionalidades oferecidas pelos *frameworks* GPDK (NOVOTNY, 2001), GridPort (THOMAS *et al*, 2001) e Gridsphere (NOVOTNY, 2004), estes foram comparados e apresentam-se na Tabela 1.

Funcionalidade	GPDK	GridPort	Gridsphere
Segurança	MyProxy	SDSC, MyProxy	SSO, MyProxy, GAMA
Recursos	GRAM	GRAM	GRAM, MDS, iGrid
Submissão de tarefas	GRAM, GSI	GRAM	Globus, GridLab GRMS
Monitoramento	GSI, LDAP	GSI, LDAP	LDAP, GRIS
Transferência de dados	GSIFTP, GridFTP, GSISCP	GridFTP, CFT	GridFTP
Forma de comunicação	Shell	SOA	SOA
Portlets	Não	Sim	Sim
Gerência de usuários	Não	Sim	Sim
Customização de leiaute	Não	Sim	Sim
Gerência de tarefas	Sim	Sim	Sim
Réplica de dados	Sim	Sim	Sim
Notificações	Não	Não	Sim

**Tabela 1. Comparativo de serviços e componentes entre frameworks para construção de portais**

## 4.2. Funcionalidades buscadas

Visando o escopo de frameworks adotado para construção de portais, será preciso determinar o que é desejável de se obter como serviço básico para a construção de um portal para o AppMan.

Funcionalidade	Versão Atual	Desejável
Segurança	Credencial Local	MyProxy
Recursos	Arquivo Compartilhado	LDAP, MoonGrid
Submissão de tarefas	ISAM/EXEHDA	ISAM/EXEHDA
Submissão de aplicação	GRID-ADL para Submission Managers	GRID-ADL para Submission Managers
Monitoramento	Arquivo Compartilhado	LDAP
Transferência de dados	NFS	NFS, GridFTP <sup>1</sup>

1. Em desenvolvimento pelo acadêmico Rodrigo Reis - UNILASALLE, 2008

**Tabela 2. Componentes interessantes para o portal do AppMan**

Para isso, estão identificados na Tabela 2 as funcionalidades atuais e desejáveis para o AppMan para controle de credenciais, monitoramento de recursos, submissão de tarefas e aplicações, monitoramento de informações e transferência de dados (VARGAS, 2006).

Através deste trabalho de identificação é possível determinar o modelo, o qual está descrito na próxima seção.

## 5. Modelo proposto

O AppMan realiza as chamadas aos serviços da grade através de *shell scripts*. Esta característica limita a forma de interação com os frameworks para construção de portais bem como com o usuário. Por exemplo, o uso de scripts não é compatível com o GridSphere nem com o GridPort. Dado este problema, foi proposto inicialmente a **construção de um portal** para suprir as necessidades de acesso ao *middleware* do AppMan do ponto de vista do usuário.

Para a construção deste portal, algumas carências devem ser supridas, como o plano de serviços sugeridos para o AppMan na Seção 4.2, onde observa-se a necessidade do controle de credenciais, que hoje é realizado através do acesso, físico ou via ssh, a um nó da grade. Também serão necessárias modificações para suportar monitoramento integrado via logs da aplicação, de dados do LDAP e o monitoramento de recursos do MoonGrid.

Além disso, foi observado que a solução a ser empregada deve ser compatível com o rumo que os portais de grade estão tomando em nível global, para não impossibilitar a evolução do portal. Por este motivo, a solução adotada é a criação de uma camada de abstração para o *middleware* do AppMan, permitindo assim sua compatibilidade com frameworks de portais atuais.

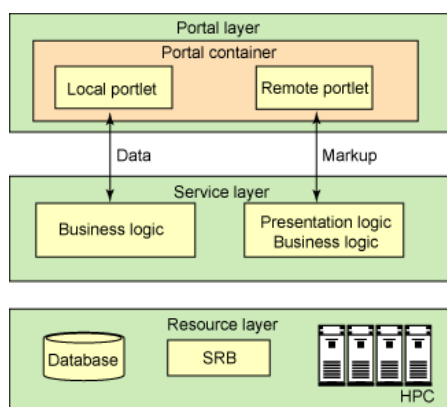


Figura 3. Arquitetura SOA para portais de grade (YANG *et al*, 2007)

Esta camada de abstração trará as funcionalidades do AppMan em compatibilidade com a geração atual de portais, que aborda os serviços a partir da arquitetura SOA (YANG *et al*, 2007), como pode ser observado na Figura 3. Embora esta camada de abstração gere um *overhead* ao *middleware*, um *overhead* também seria necessário, entretanto possivelmente em menor escala, no caso da criação de um portal específico para o AppMan. Contudo, a adaptação nas chamadas do *middleware* em questão trará muitas vantagens, como por exemplo, a compatibilidade deste com

diversos portais de grade, aumentando assim a confiança no portal, possibilitando atualizações de versão e a escolha do portal que mais agrada aos usuários.

A **validação** da solução apresentada no item anterior inicialmente ocorrerá através da execução do *middleware* do AppMan no portal escolhido como mais vantajoso, o Gridsphere.

Devido ao formato da solução, será esperada a validação da execução do AppMan em mais de um portal, sem a necessidade de modificações no código de adaptação proposto, apenas na configuração apropriada deste segundo portal.

## 6. Conclusão

Uma das motivações deste trabalho é facilitar o acesso por usuários comuns ao serviço de grade oferecido pelo gerenciador de aplicações AppMan. Acredita-se que a evolução contínua do protótipo deve ser facilitada a medida que mais usuários finais tenham acesso fácil aos recursos da grade através do sistema. Deste modo, além de uma contribuição científica ao propor soluções para integração de soluções via frameworks distintos, e de contribuir para o estudo da viabilidade de implantação de um mecanismo de segurança mais robusto para o modelo GRAND, tem-se um benefício direto de facilidade de uso aos usuários do sistema.

Considera-se de grande valia a identificação dos requisitos para a transformação do AppMan em um portal pois, uma vez realizada, é peça crucial para a viabilização do portal. Nestes requisitos se enquadra a questão de segurança, onde será dado um salto em direção da modernização dos serviços prestados pelo *middleware*. Também é importante ressaltar que, ao adotar o uso de um portal compatível com portlets não será necessário seu desenvolvimento por completo, diminuindo drasticamente o tempo necessário para a conclusão do portal e possibilitando a continuação do trabalho no prazo inicialmente estipulado.

Com o uso facilitado do AppMan através do portal, é possível obter melhor controle de execução e visão das tarefas executadas pelo *middleware* e sua integração com outros serviços. Além disso, possibilitará a visualização da integração entre entidades e colaboração com a grade mundial. Esta integração de informações pode ser também um primeiro passo para melhorar a qualidade das informações utilizadas no escalonamento das tarefas em possíveis trabalhos futuros.

## Referências

- Ian Foster, Carl Kesselman, The Globus toolkit, The grid: blueprint for a new computing infrastructure, p.37-60, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1998.
- Ian Foster, Carl Kesselman, Steven Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of High Performance Computing Applications, v.15 n.3, p.200-222, agosto de 2001.
- Xiao Haili, Wu Hong, Chi Xuebin, Deng Sungen, Zhang Honghai, An Implementation of Interactive Jobs Submission for Grid Computing Portals. Conferences in Research and Practice in Information Technology Series; Vol. 108, Proceedings of the 2005

- Australasian workshop on Grid computing and e-research - Volume 44, p.67-70, Australian Computer Society, Newcastle, New South Wales, Australia. 2005.
- Global Grid Forum, Grid Computing Environments, <http://www.computingportals.org/>, acessado em outubro de 2008.
- Patrícia Kayser Vargas, GRAND: Um Modelo de Gerenciamento Hierárquico de Aplicações em Ambiente de Computação em Grade, Tese de Doutorado (COPPE – Engenharia de Sistemas e Computação/UFRJ – Universidade Federal do Rio de Janeiro), 2006.
- David De Roure, Mark A. Baker, Nicholas R. Jennings, e Nigel R. Shadbolt. The evolution of the grid. In Grid Computing: Making the Global Infrastructure a Reality, Berman F, Fox GC, Hey AJG (eds). Wiley Series in Communications, Networking and Distributed Systems. Wiley: Chichester, 2003. p.65-100.
- Patrícia Kayser Vargas, Marluce Rodrigues Pereira, Claudio F. Resin Geyer. Computação em Grade e o Globus Toolkit 4. Em: Minicurso do ERAD 2007 – Escola Regional de Alto Desempenho – 16 a 19 de janeiro de 2007. Porto Alegre, RS, 2007. p.11-31.
- Sun Microsystems. Java Web Services At a Glance, <http://java.sun.com/webservices/>, acessado em outubro de 2008.
- Globus Alliance, IBM e HP, The WS-Resource Framework, 20 de janeiro de 2004. Disponível em: <http://www.globus.org/wsrf/>.
- Global Grid Forum e OASIS, The Open Grid Services Architecture, 29 de janeiro de 2002. Disponível em: <http://www.globus.org/ogsa/>.
- Borja Sotomayor, Lisa Childers. Globus Toolkit 4: Programming Java Services. San Francisco, California. Ed. Morgan Kaufmann, 2006.
- Matthew L. Massie, Brent N. Chun and David E. Culler, The ganglia distributed monitoring system: design, implementation, and experience, Parallel Computing, v. 30, n. 7, pp. 817–840, Julho de 2004.
- H.B. Newman, I.C.Legrand, P. Galvez, R. Voicu, C. Cirstoiu, MonALisa: A distributed monitoring service architecture, In: Computing in High Energy and Nuclear Physics (CHEP03), (La Jolla, California, USA), 24 a 28 de março de 2003.
- Dan Gunter, James Magowan, An analysis of “Top N” Event Descriptions, GGF Discovery and Monitoring Event Descriptions Working Group, janeiro de 2004.
- Daniel da Trindade Lemos, Everton Stadolni, Patrícia Kayser Vargas, MoonGrid: Monitoramento de Recursos em Ambientes de Grade, Em: Workshop de Processamento Paralelo e Distribuído (WSPPD 2007), 10 de agosto de 2007.
- Parvin Asadzadeh, Rajkumar Buyya, Chun Ling Kei, Deepa Nayar, e Srikumar Venugopal, Global Grids and Software Toolkits: A Study of Four Grid Middleware Technologies, High Performance Computing: Paradigm and Infrastructure, Laurence Yang and Minyi Guo (eds), pp.431-458 (Chapter 22), ISBN: 0-471-65471-X, Wiley Press, New Jersey, USA, Junho de 2005.

- Chongjie Zhang, Ian Kelley e Gabrielle Allen, Grid portal solutions: a comparison of GridPortlets and OGCE: Research Articles. Workshop on Grid Computing Portals (GCE 2005), Concurrency and Computation: Practice and Experience, Volume 19, Issue 12 (Agosto de 2007), p.1739-1748.
- Mary Thomas, Steve Mock, Maytal Dahan, Kurt Mueller, Don Sutton, John R. Boisseau, The GridPort Toolkit: A System for Building Grid Portals, Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01), p.216, 07 a 09 de agosto de 2001.
- Maytal Dahan, Eric Roberts e Jay Boisseau, TeraGrid User Portal v1.0: Architecture, Design, and Technologies, Second International Workshop on Grid Computing Environments GCE06 at SC06, Tampa, FL. 12 a 13 de novembro de 2006.
- Michael Russell, Jason Novotny e Oliver Wehrens, GridSphere's Grid Portlets, Computational Methods in Science and Technology (CMST) Vol. 12, Scientific Publishers OWN, p. 89-97, 2006.
- Java Technology Specification, JSR (*Java Specification Request*) 168: Portlet Specification, 2003. Disponível em: <http://jcp.org/en/jsr/detail?id=168>.
- Jason Novotny, Michael Russell, Oliver Wehrens, GridSphere: a portal framework for building collaborations: Research Articles, Concurrency and Computation: Practice & Experience, v.16 n.5, p.503-513, abril de 2004.
- Brett Beeson, Steve Melnikoff, Srikumar Venugopal, David G. Barnes, A portal for grid-enabled physics, Proceedings of the 2005 Australasian workshop on Grid computing and e-research, p.13-20, 01 de janeiro de 2005, Newcastle, New South Wales, Australia.
- Jason Novotny, Grid Portal Developers Kit, Building a Portal Using GSDK: A Developers Tutorial, Grid Portal Developers Workshop (GPDW) 2001, 04 a 05 de junho de 2001, Lawrence Berkeley Laboratory, California.
- Xiaobo Yang, Xiao Dong Wang, Robert Allan, Development of standards-based grid portals, Part 1: Review of grid portals, IBM developerWorks article, <http://www-128.ibm.com/developerworks/grid/library/gr-stdportal3/index.html>, 20 de março de 2007.
- Anurag Shankar, A General Introduction to Portals/Gateways, TeraGrid Gateways Team, Dept. of Astronomy & Univ. IT Services, Indiana University, novembro de 2006.