

MidGov: Middleware para Governo Eletrônico baseado em Grades Computacionais

Geraldo M. Silva, Ivo José G. dos Santos e Edmundo R. M. Madeira

Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6.176 – 13.083-970 – Campinas – SP – Brasil

geraldoms@lrc.ic.unicamp.br, {ivo, edmundo}@ic.unicamp.br

Abstract. *The demand for integrated and efficient public services has increased in recent years, mainly due to the proliferation of information and communication technologies. In order to develop these new applications, some challenges must be faced, including greater interoperability, scalability and security issues. Grid computing can be considered an interesting solution for Middleware applications of e-government, thanks to its high storage and processing capacity, in addition to its recent service orientation. This paper presents MidGov, a middleware for e-government applications in the Grid with support provided by services of the Globus Toolkit 4 and features of the Semantic Web.*

Resumo. *A demanda por serviços públicos integrados e eficientes vem aumentando ao longo dos últimos anos, devido principalmente à proliferação das tecnologias de informação e comunicação. A fim de desenvolver essas novas aplicações, alguns desafios precisam ser enfrentados, incluindo maior interoperabilidade, escalabilidade e questões de segurança. Computação em Grade pode ser considerada uma solução interessante para Middleware em aplicações de Governo Eletrônico, graças a sua alta capacidade de armazenamento e processamento, além de sua recente orientação a serviços. Este trabalho apresenta o MidGov, um Middleware para aplicações de Governo Eletrônico em Grade com suporte fornecido por serviços do Globus Toolkit 4 que utiliza recursos da Web Semântica.*

1. Introdução

A proliferação das Tecnologias de Informação e Comunicação (TICs) vem proporcionando uma demanda cada vez maior por serviços integrados e eficientes. A utilização dessas tecnologias na área de Governo Eletrônico (e-Government), Participação Eletrônica (e-Participation) e Democracia Eletrônica (e-Democracy) se tornaram foco em diversos órgãos governamentais ao redor do mundo, onde, diversas pesquisas ([Medjahed and Bouguettaya 2005, Alfonso et al. 2007, Patra and Das 2007]) são realizadas com ênfase em aplicações visando um governo mais transparente, participativo e colaborativo.

Aplicações para esse tipo de cenário introduzem uma série de desafios, como a interoperabilidade entre sistemas, privacidade dos dados, transparência, escalabilidade, entre outros. Nesse sentido, o paradigma de Computação Orientada a Serviços (*Service-Oriented Computing* – SOC) aparece como forte proposta para mitigar a heterogeneidade dos variados serviços prestados pelas entidades envolvidas, onde uma aplicação é construída a partir da interação entre serviços [Curbera et al. 2003].

Atualmente, uma das grandes limitações da *Internet* é o baixo suporte a atribuição de significado ao conteúdo publicado, impossibilitando que um processamento mais preciso possa ser realizado. Para sobrepor esse obstáculo a chamada *Web Semântica* propõe um cenário onde dados começam a ter significado. Nesse contexto, surgiram os chamados Serviços na *Web Semântica* (*Semantic Web Services – SWS*), que fornecem suporte semântico aos *Web Services*, possibilitando a criação de mecanismos dinâmicos em sua composição e maior gama de possibilidades em sua busca [Tangmunarunkit et al. 2003].

Grades Computacionais fornecem alto desempenho em ambientes heterogêneos. Sua construção pode ser realizada através do *Globus Toolkit* desenvolvido pela Globus Alliance, a qual inclui um conjunto de serviços para este fim. Em sua versão 4, o *Globus Toolkit* implementa OGSA (*Open Grid Services Architecture*) [Foster et al. 2002] com WSRF (*Web Services Resource Framework*) [OASIS 2006] promovendo uma convergência entre Grades e Arquitetura Orientada a Serviços, a transformando em uma poderosa ferramenta para aplicações que envolvem serviços de diferentes entidades [Sotomayor and Childers 2006].

Dessa forma, este artigo tem como objetivo propor uma plataforma para aplicações de Governo Eletrônico sobre o ambiente fornecido pelas Grades Computacionais utilizando como suporte serviços fornecidos pelo *Globus Toolkit 4* aliado a recursos da *Web Semântica*. Suas principais contribuições se concentram na modelagem para utilização de Grades Computacionais no âmbito de aplicações de e-Governo e na utilização de serviços do *Globus Toolkit* para suprir necessidades impostas por essas aplicações.

O restante do artigo é organizando da seguinte forma: a Seção 2 descreve alguns conceitos referentes ao trabalho; relatos de outros trabalhos relacionados são apresentados na Seção 3; na Seção 4 é descrito o modelo proposto; na Seção 5 são expostos aspectos da implementação do modelo; e na Seção 6 é apresentada a conclusão.

2. Conceitos Básicos

Inicialmente iremos introduzir alguns conceitos gerais referentes a Governo Eletrônico, Computação Orientada a Serviços, *Web Semântica* e Grades Computacionais. Trata-se de fundamentos básicos para o entendimento do artigo.

2.1. Governo Eletrônico

Governo Eletrônico (*e-Government*) ou e-Governo utiliza-se de tecnologias da informação e computação para prover maior eficiência e serviços públicos amigáveis ao cidadão [Barnickel et al. 2006]. Trata-se de uma nova estrutura descentralizada que permite ao governo realizar uma espécie de rede horizontal entre os cidadãos, reforçando a cidadania e revigorando a administração pública.

Requisitos para aplicações dessa natureza são bem semelhantes. Entretanto, aplicações de Governo Eletrônico necessitam de determinados requisitos (recursos distribuídos geograficamente, transparência e a adesão de padrões abertos e softwares livres) e são mais sensíveis a outros (escalabilidade, interoperabilidade, privacidade e segurança) [Alfonso et al. 2007, e-PING 2010]. Dentre as principais estratégias propostas para aumentar o grau de interoperabilidade dos sistemas estão as baseadas em Computação Orientada a Serviços (*Service-Oriented Computing – SOC*).

2.2. Computação Orientada a Serviços

SOC é um paradigma de programação na qual as funcionalidades dos sistemas são construídos a partir de combinações de serviços. Esse paradigma emergiu em resposta à forma como as empresas conduziam seus negócios, na qual, empresas totalmente integradas tornaram-se redes empresariais. Nesse contexto, cada participante fornece serviços específicos formando as chamadas Organizações Virtuais (*Virtual Organizations – VO*).

Para apoiar o paradigma SOC, *Web Services* fornecem interoperabilidade universal, são baseados em padrões abertos da internet e fornecem técnicas para descrever, acessar, localizar e identificar os componentes de *software*. O seu conjunto básico de especificações XML (*eXtensible Markup Language*) são: WSDL, SOAP e UDDI. O paradigma SOC aliado às tecnologias dos *Web Services* visa combinar dois ou mais serviços, proporcionando assim a construção de *Workflows* [Russell et al. 2005]. A composição de serviços pode ser realizada através da Orquestração dos mesmos [Curbera et al. 2003].

2.3. A Web Semântica

A *Web* semântica consiste em estabelecer anotações para permitir que máquinas possam automaticamente compreender e processar dados. Nesse contexto, as ontologias realizam papel importante, agregando a descrição formal de determinado domínio e descrevendo as funcionalidades e os atributos de cada conceito [Tangmunarunkit et al. 2003].

A possibilidade de utilizar estratégias similares para a descrição de serviços motivou diferentes abordagens para a definição dos chamados Serviços na Web Semântica. Uma proposta nesse sentido é a linguagem OWL-S (*Semantic Markup for Web Service*) [Martin et al. 2004], que combina um conjunto de ontologias OWL [W3C 2004] inter-relacionadas para definir termos em aplicações orientadas a serviços.

O Midgov implementa uma estratégia de composição de serviços estática, tornando a utilização da *Web Semântica* um artefato não essencial, porém, devido à quantidade de serviços que o Governo disponibiliza e levando em consideração a utilização de aplicações manipuladas por cidadãos, esta tecnologia se transforma em um importante requisito não funcional.

2.4. Grades Computacionais

Grades de Computadores são ambientes de alto desempenho que possibilitam o compartilhamento coordenado de recursos em ambientes heterogêneos. Em [Foster 2002], uma descrição mais detalhada é fornecida, onde, Grades Computacionais são compostas por:

- Recursos coordenados que não estão sujeitos a controles centralizados;
- Utiliza-se de padrões abertos de interfaces e protocolos de propósito geral; e
- Oferece qualidades de serviço não triviais.

Para que a Grade se torne funcional, é necessário um conjunto de serviços para permitir submissão, controle de aplicações, descoberta de recursos, movimentação de dados, e mecanismos de segurança, entre outros. O GT, em sua versão 4, implementa a OGSA (*Open Grid Services Architecture*) com WSRF (*Web Services Resource Framework*) e oferece esse conjunto de serviços [Sotomayor and Childers 2006].

3. Trabalhos Relacionados

Governo Eletrônico se transformou em uma tendência global. Dessa forma, diversas pesquisas estão sendo realizadas com intuito de fornecer padrões e plataformas para aplicações nessa área. Um *Middleware* (*Citizen-oriented e-Government Platform* – Cog-Plat) para aplicações de e-Governo que fornece um conjunto de facilidades para desenvolver e operar serviços centrados no cidadão é proposto por [Santos 2008]. A abordagem inclui um mecanismo para a seleção dinâmica de serviços (utilizando serviços na *Web Semântica*) e introduz um conjunto de políticas para regular a colaboração entre entidades (diferentes níveis de autonomia, privacidade, rastreabilidade e gestão de identidade). Em [Medjahed and Bouguettaya 2005], é apresentado o projeto *WebSenior*, um *Middleware* para aplicações voltadas ao contexto da seguridade social para cidadãos da terceira idade. A abordagem propõe a geração de serviços customizados automaticamente às necessidade do cidadão, levando em conta seu perfil e parâmetros de qualidade dos *Web Services* capturados a partir do mesmo. A plataforma introduz a descrição de serviços e operações do e-Governo através de uma ontologia e propõe três níveis de customização: do cidadão, do serviço e de interface do usuário. Em [Drigas and Koukianakis 2009], uma infra-estrutura para a difusão de informação e para colaboração de serviços da administração pública é apresentado. O autor introduz a aplicação de serviços de comunicação (como fórum de discussão, *chat* e caixa de mensagem) e serviços de informação (como anúncios) para a difusão de informação e distingue níveis de usuários com determinados serviços relacionados a cada nível.

Diversas pesquisas relacionadas a Grades Computacionais têm proporcionado sua fácil adesão e a transformado em uma excelente ferramenta para vários tipos de aplicações. Uma infra-estrutura para *Workflow* chamada GPO (*Grid Process Orchestration*), controlável pelo usuário, que faz a orquestração de serviços e processos em grades computacionais é proposta por [Senna 2007]. A abordagem inclui uma linguagem (*Grid Process Orchestration Language* – GPOL) para descrição dos *Workflows*. Em [Goldchleger et al. 2004], é introduzido o *InteGrade*, um *Middleware* orientado a objetos, implementado em CORBA para Grades Computacionais com foco na utilização do poder computacional ocioso (Grades Oportunistas) em computadores pessoais. O Projeto é multi-institucional e sua estrutura é composta por *clusters*, que podem incluir: *Cluster Manager*, *User Node*, *Resource Provider Node* e *Dedicated Node*. Um conjunto de serviços (*open source*) para a construção de uma Grade Computacional e para a criação de aplicações para a mesma de uma forma incremental é proposto pela Globus Alliance em [Globus Alliance]. Em sua versão 4, o *Globus Toolkit* está totalmente orientado a serviços e fornece os principais serviços para a funcionalidade da Grade.

A adoção de padrões abertos, orientação a serviços, além de suas características naturais está provocando uma demanda por pesquisas e plataformas com foco na utilização de Grades Computacionais em aplicações de Governo Eletrônico. Em [Santos and Madeira 2007], os autores apresentam uma investigação sobre o estado da arte e discutem alguns dos principais desafios a serem vencidos para que as grades possam atender aos principais requisitos das aplicações de Governo Eletrônico. Já em [Li et al. 2005], os autores apresentam as principais realizações do Governo de Shanghai no desenvolvimento de aplicações de Governo Eletrônico, desafios enfrentados em sua utilização sobre Grades e propõe um modelo conceitual para e-Governo baseado em Grades Computacionais utilizando o *Globus Toolkit 3*. Um *Middleware* que propõe o uso

de Grades Computacionais baseado no *Globus Toolkit 4* para a integração de informação nos processos da administração pública é proposto em [Alfonso et al. 2007]. A abordagem utiliza diversos serviços do Globus Toolkit e adiciona novos componentes baseados em requisitos do e-Governo. Em [Senger et al. 2006], uma plataforma em Grades computacionais para Democracia Eletrônica que utiliza o *Glogus Toolkit 4* e alguns de seus serviços para o seu suporte é proposta. Um cenário de aplicação da infra-estrutura de Grade voltada a formulação de novas políticas é apresentado.

4. Modelo Proposto

O objetivo do *Middleware* é propor um ambiente sobre Grades Computacionais ao cidadão que permita a execução de serviços (*Workflows*) compostos por uma ou mais funções dos órgãos administrativos do governo e possibilite a interação entre cidadão e seus políticos. A Figura 1 apresenta uma visão geral do projeto, onde, a infra-estrutura do MidGov reside em uma das máquinas da Grade construída através do *Globus Toolkit*.

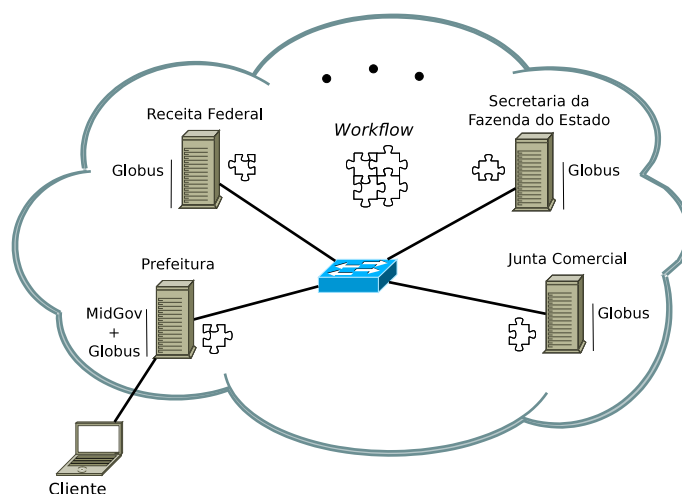


Figura 1. Visão do Ambiente.

A infra-estrutura interna proposta (*Middleware for Government* – MidGov) é baseada na infra-estrutura do *CogPlat*, apresentado em [Santos 2008], com adicional utilização de serviços fornecidos pelo *Globus Toolkit 4* (GT4) [Globus Alliance] para suporte e sua adequação sob a Grade. A plataforma inclui um barramento de serviços, quatro módulos compondo seu núcleo e um conjunto de serviços de suporte. A seguir iremos descrever melhor a infra-estrutura proposta.

4.1. Barramento de Serviços

Com intuito de facilitar a utilização, o barramento de serviços (*Service Bus*) fornece uma espécie de interface entre aplicações e a infra-estrutura, onde todos os seus serviços são acessíveis (expostos como *Web Services*).

4.2. Núcleo da Plataforma

A Infra-Estrutura interna do *Middleware* é composta por quatro módulos: *ServicesManagerCenter*, *MetamodelManagementCenter*, *E-GovernmentE-DemocracyCenter* e *TraceabilityAuditingCenter*. A Figura 2 apresenta o diagrama de classe para cada núcleo da plataforma.

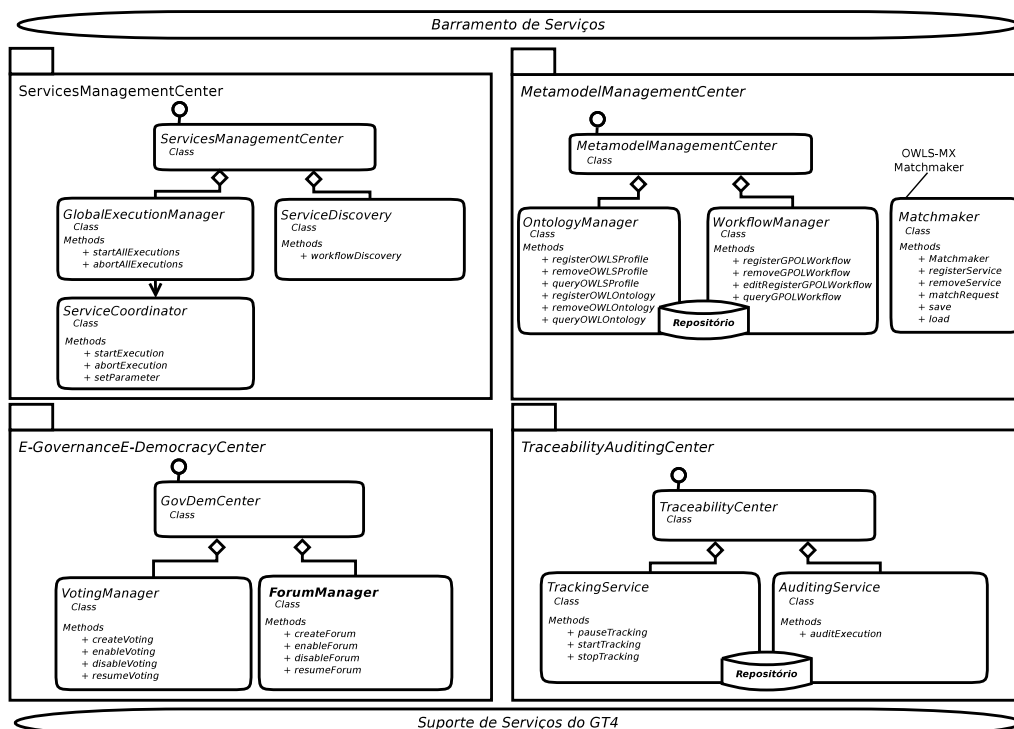


Figura 2. Infra-estrutura geral estendida.

O **Services Management Center** inclui serviços para a descoberta de *Workflows* e a gestão da execução dos mesmos sob a Grade Computacional. Sua estrutura interna possui os seguintes componentes:

- **Service Coordinator:** fornece métodos para controle de *Workflows* individuais sob a Grade Computacional e ajuste de seus valores para sua execução;
- **Global Execution Manager:** contém métodos para controle de todos os *Workflows* sob a Grade e utiliza funcionalidades prescritas na classe *Service Coordinator*;
- **Service Discovery:** realiza a descoberta de *Workflows* utilizando o *Owls-MX Matchmaker* (incluído no *MetamodelManagementCenter*). Este realiza uma espécie de casamento entre uma requisição e serviços registrados em sua base de conhecimento (ambos contendo um perfil em OWL-S) [Klusch et al. 2006].

O **Metamodel Management Center** oferece serviços para a gestão de Ontologias, perfis OWL-S e *Workflows* em GPOL (*Grid Process Orchestration Language*), uma linguagem para descrição de *Workflows* baseada em XML [Senna 2007]. Os componentes desse módulo são os seguintes:

- **OntologyManager:** possui serviços que acessam métodos do *Owls-MX Matchmaker* para a gestão de perfis OWL-S (que são armazenados em sua base de conhecimento) e serviços para a gestão de Ontologias;
- **WorkflowManager:** inclui facilidades para a gestão de *Workflows*, que são incluídos em um repositório;
- **Owls-MX Matchmaker:** oferece uma consulta semântica baseada no mapeamento de entrada/saída de perfis escritos em OWL-S juntamente com facilidades para a gestão dos perfis em sua base de conhecimento [Klusch et al. 2006].

O *E-Government and E-Democracy Center* tem como objetivo ampliar a participação do cidadão na administração pública, no sentido de aumentar a interação entre cidadãos e seus políticos. Sua infra-estrutura inclui os seguintes componentes:

- **VotingManager**: fornece funcionalidades para a gestão de uma votação eletrônica;
- **ForumManager**: o cidadão poderá introduzir sua opinião ou mesmo realizar discussões sobre determinado assunto.

O *Traceability and Auditing Center* fornece facilidades para rastrear os estágios de determinada execução e sua futura auditoria (após o processo ser completado). Sua infra-estrutura é composta pelos elementos a seguir:

- **TrackingService**: possui facilidades para realizar o rastreamento das execuções. Os dados monitorados são exibidos em tempo real e armazenados para sua possível auditoria;
- **AuditingService**: realiza a auditoria de serviços concluídos (acessando dados monitorados).

4.3. Suporte de Serviços do GT4

O MidGov utiliza o GPO (*Grid Process Orchestration*) para submissão de *Workflows* (em GPOL) na Grade Computacional e inclui a camada de segurança fornecida pela GSI (*Grid Security Infrastructure*) através do Java WS Core, exposta pelo MidGov através das políticas de interação (detalhes na Seção 4.4), que regulam a interação entre entidades.

O **GPO** é um middleware baseado nos conceitos da OGSA e permite a criação e gerência de fluxos de serviços que se interagem em uma Grade Computacional. Sua infra-estrutura é apresentada na Figura 3. O *GPO Run* recebe arquivos em GPOL e usa o *GPOMaestro Factory Service* (fábrica de serviços) para criar uma nova instância do *GPOMaestro Service Instance* que fica responsável pela execução do *Workflow* na Grade. Após o processo, o *GPO Run* termina a instância entregando o resultado à aplicação [Senna 2007].

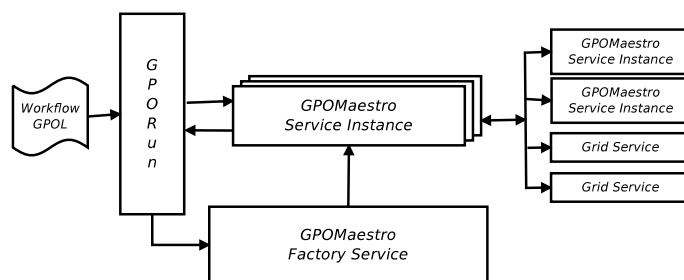


Figura 3. Arquitetura do GPO [Senna 2007].

Para possibilitar que a plataforma possa realizar execuções na Grade, assim como o *GPO Run*, o MidGov se comporta como um cliente do *GPOMaestro Factory Service*, acessando diretamente suas facilidades e importando as funcionalidades do *GPO Run* (análise sintática do arquivo em GPOL, gestão do ciclo de vida do WS-Resource e adicionais camadas de segurança a partir das especificações da requisição). Dessa forma, o *GPO Run* é descartado e o acoplamento entre as duas plataformas é realizado.

Java WS Core consiste em um conjunto de APIs e bibliotecas para desenvolvimento de serviços da Grade e a utilização dos mesmos. Possui ferramentas que implementam os padrões WSRF e *WS-Notification* em linguagem *Java*.

A **GSI** fornece uma camada de segurança a serviços da Grade. Os componentes da GSI utilizados pelo MidGov incluem:

- **Authorization Framework**: determina quais requisições são permitidas, possibilitando restringir o uso a uma requisição (identidade específica) ou a um grupo;
- **Message-level Security**: utiliza criptografia para fornecer segurança no envio de mensagens via SOAP. Apenas o conteúdo da mensagem é de fato criptografado; e
- **Authentication**: assegura que as partes envolvidas na colaboração são quem elas dizem que são (inclui mecanismos para a autenticação anônima).

A GSI inclui componentes (classes java) para facilitar a agregação de segurança em serviços da Grade. A segurança do lado do servidor é centralizada no descritor de segurança (*security descriptor*). Dessa forma, é preciso especificar qual descritor de segurança utilizar, adicionando-o no arquivo WSDD (*Web Service Deployment Descriptor*), como ilustram as Figuras 4 e 5:

```
<securityConfig xmlns="http://www.globus.org">
  <!-- ... -->
  <authz value="none"/>
  <method name="CadEstadual">
    <auth-method>
      <GSISecureConversation>
        <protection-level>
          <anonymous>
        </protection-level>
      </GSISecureConversation>
    </auth-method>
  </method>
  <!-- ... -->
</securityConfig>
```

Figura 4. Descritor de Segurança.

```
<service name="cadastro/sec/CadEstadual"
  provider="Handler" use="literal" style="document">
  <!-- ... -->
  <parameter name="securityDescriptor"
    value="etc/org_globus_cadastro_services_sec/
    security-config.xml"/>
  <!-- ... -->
</service>
```

Figura 5. Descritor de Implementação.

O código da Figura 4 indica que nenhuma autorização deve ser realizada ao acessar determinado serviço e que o método *CadEstadual* pode ser invocado utilizando apenas o esquema *GSISecureConversation* com anonimato. A Figura 5 ilustra a importação do arquivo de configuração de segurança no descritor de implementação (WSDD). Do lado do cliente, pode-se agregar segurança através do descritor de segurança ou mesmo em nível de implementação no cliente, possibilitando que o cliente possa obter a camada de segurança desejada.

4.4. Políticas de Interação

Com intuito de construir um ambiente mais adequado às aplicações do e-Governo, o MidGov implementa um conjunto de políticas de interação, baseadas em [Santos 2008], e agora inseridas no contexto das Grades. As políticas não interferem na busca por *Workflows* uma vez que são suportadas por serviços do GT4. Estas políticas englobam:

- **Políticas de Privacidade dos Dados** (*Entidade x Entidade*): determina o nível de confiança entre entidades da Organização Virtual (VO) incluindo autorização para o acesso a serviços e suporte a segurança dos dados (criptografia e autorização). A GSI fornece serviços para que cada entidade possa especificar grupos ou usuários individuais que possam utilizar determinado(s) serviço(s) (*Authorization Framework*) e para a proteção dos dados (*Message-level Security*);

- **Políticas de Gestão de Identidade** (*Cidadão x MidGov*): oferece serviços para a autenticação das entidades envolvidas, possibilitando seu anonimato. As facilidades oferecidas pelo componente *Authentication* presente no GSI são endereçadas para suprir as necessidades de autenticação e anonimato das entidades;
- **Políticas de Rastreabilidade de Serviços** (*MidGov x Serviço*): permite estabelecer a rastreabilidade de eventos durante a execução do *Workflow* na Grade. A opção *stat* oferecida pelo GPO (que fornece o estado da execução) e o conceito de *WS-Notification* são empregados. *WS-Notification* é um padrão de projeto que permite que o cliente seja notificado quando eventos de seu interesse aconteçam nos recursos do serviço [Sotomayor and Childers 2006].

4.5. Arquitetura do Projeto

O MidGov é um cliente de serviços da Grade, seus módulos internos utilizam as facilidades do GPO (*GPOMaestroFactoryService*) e a GSI (ambos previamente publicados na Grade) em suas tarefas. O *Services Management Center* contém serviços básicos, responsáveis pela descoberta (através do matchmaker) e coordenação de Workflows também acessíveis por outros módulos, além de permitir a recuperação do status de sua execução pelo *Traceability Auditing Center*.

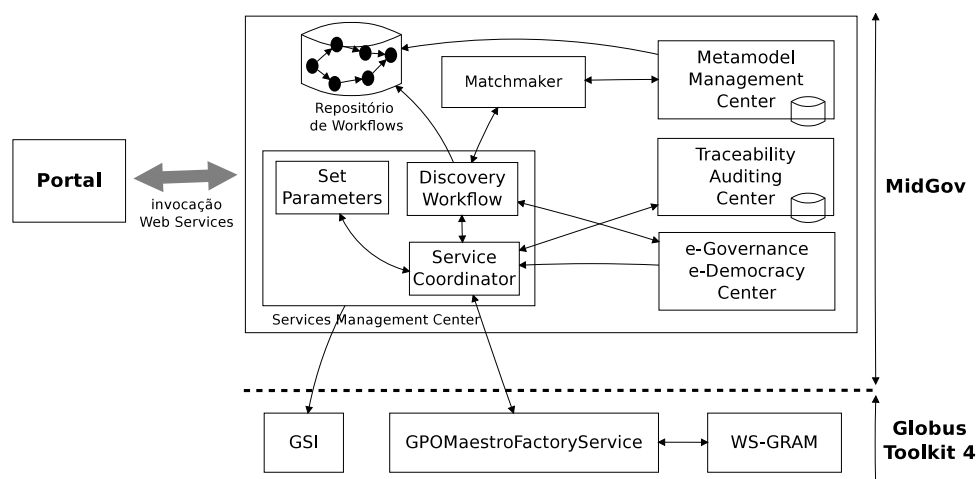


Figura 6. Arquitetura do MidGov.

Todos os serviços da plataforma são expostos como *Web Services* através do *Service Bus*, que permite que aplicações (como exemplo, um Portal Web) possam acessar as funcionalidades do *Middleware*. A Figura 6 apresenta o relacionamento entre o MidGov e os serviços do GT4, seus relacionamentos internos (entre módulos e serviços) e seus serviços mais básicos.

5. Aspectos de Implementação

Para possibilitar a integração junto ao *Java WS Core*, a linguagem Java foi escolhida para a implementação do *Middleware* proposto. O barramento de serviços exporta *Web Services* (WSDL + SOAP/HTTP) utilizando o servidor Tomcat versão 6.0.20 para executar o *framework* de *Web Services* Axis versão 1.4 (*Service Bus*). Estes são acessíveis através de um portal construído utilizando o JSP (*JavaServer Pages*) e HTML (*HyperText Markup Language*). Durante a descoberta de *Workflows*, o *Discovery Workflow* recebe uma

requisição (contendo um perfil OWL-S) do portal e utiliza as facilidades do matchmaker (implementado em Java/Jena) proposto em [Klusch et al. 2006].

Os *Workflows* são descritos em GPOL (*Grid Process Orchestration Language*) e sua análise sintática é realizada através da API DOM (*Document Object Model*) integrada ao JDK da *Sun*. *Workflows* e Ontologias (implementadas em OWL) são armazenados em sub-diretórios da plataforma e os demais repositórios são implementados com o banco de dados MySQL.

5.1. Descoberta e Execução de Workflows

O processo de inserção de um *Workflow* inicialmente requer sua descrição em GPOL e de seu perfil em OWL-S. Durante a inserção, um mapeamento é realizado (em uma base de dados) entre perfil e *Workflow* incluindo também todos os seus parâmetros de execução. Estes últimos são necessários para que a plataforma possa receber (a partir do portal) estes valores e reajustá-los no *Workflow* para sua execução (a GPOL não permite a passagem de parâmetros ao *Workflow*).

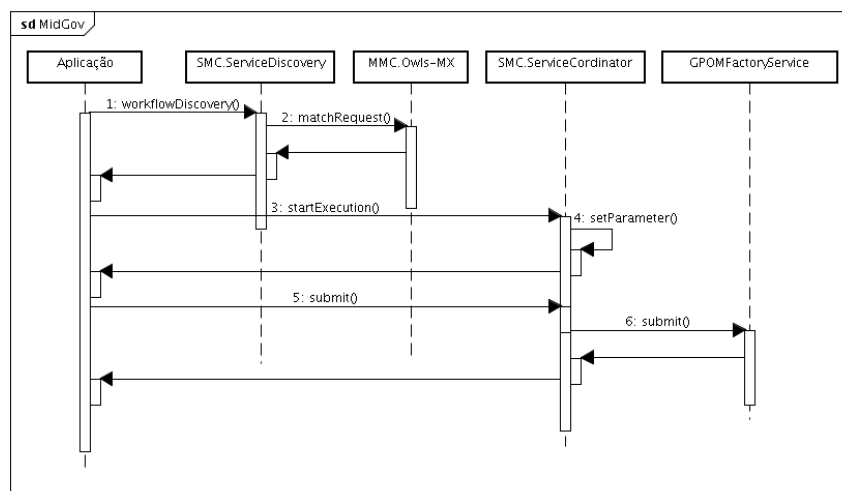


Figura 7. Diagrama de Sequência: Descoberta e execução de *Workflow*.

O processo de descoberta e execução de *Workflow* (submissão simples) segue os estágios ilustrados na Figura 7. Inicialmente, (1) a aplicação envia uma requisição (contendo o perfil semântico) para o *Services Management Center* solicitando uma busca por *Workflows* (*SMC.ServiceDiscovery.workflowDiscovery*); (2) facilidades do *Owls-MX Matchmaker* são utilizadas para realizar uma busca semântica (*MMC.Owls-Mx.matchRequest*) por perfis em sua base de conhecimento. O resultado é retornado para a aplicação que poderá (ou não) realizar sua execução. No caso de sua execução, uma requisição (3) é enviada ao *Services Management Center*, que por sua vez, solicita os dados (4) a partir da aplicação. Após a submissão dos dados (5), as variáveis do *Workflows* são ajustadas (reeditando o arquivo GPOL) e sua submissão (6:) junto ao *GPOMaestro Factory Service* é realizada retornando o resultado da execução à aplicação.

5.2. Cenário de Aplicação

Com o objetivo de ilustrar a utilização da plataforma, apresentamos um exemplo (em fase de teste) de aplicação da plataforma referente ao projeto Cadastro Sincronizado Nacio-

nal (CadSinc) [Receita Federal 2003]. O CadSinc tem como objetivo a simplificação do processo de gestão do cadastro de pessoa jurídica nas três esferas do Governo.

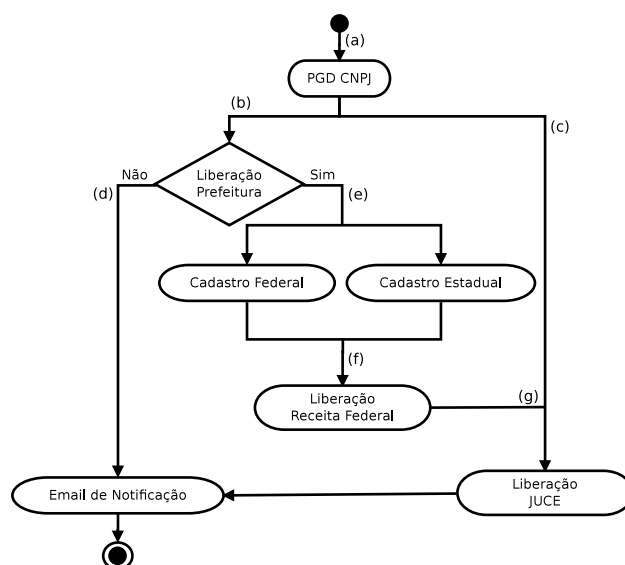


Figura 8. Diagrama de Atividade: Cadastro Sincronizado.

Após o cidadão ter escolhido o nome empresarial e a atividade econômica, o fluxo do cadastro jurídico segue as seguintes etapas (veja Figura 8): **(a)** O cidadão utiliza o PGD (Programa Gerador de Documentos) CNPJ para criar o DBE (Documento Básico de Entrada), que é enviado ao JUCE (Junta Comercial) e a Prefeitura Municipal. Em seguida, **(b)** a prefeitura realiza uma consulta prévia quanto a possibilidade de exercer a atividade no local pretendido com base em regras de zoneamento. O DBE é enviado ao JUCE **(c)**, possivelmente com demais documentos. Caso não seja possível o exercício da atividade **(d)**, o usuário é notificado. Em caso de aprovação pela Prefeitura **(e)**, é realizado o cadastro federal e estadual. Continuando o processo, uma consulta é realizada na Receita Federal **(f)** e caso não existam restrições, uma autorização é enviada a JUCE **(g)**. De acordo com a consulta na receita federal e a documentação (DBE e demais documentos), a JUCE poderá ou não autorizar o registro da empresa, sendo que nos dois casos uma notificação ao usuário é realizada.

Inicialmente, um especialista no domínio é responsável pela descrição do *Workflow* em GPOL, de seu perfil semântico e a publicação dos serviços utilizados pelo *Workflow* na Grade. A execução segue o fluxo demonstrado na Figura 7 e o *Workflow* (simplificado) em GPOL para o CadSinc é apresentado na Figura 9. Em GPOL define-se as variáveis para manter estado em *<gpo:variables>*, serviços usados na execução em *<gpo:definitions>* e o fluxo do processo em *<gpo:process>*. No fluxo do processo em GPOL, cada método do serviço é invocado através do elemento *<invoke>*, possibilitando a passagem de argumentos e o retorno de sua execução. Os serviços, como no exemplo, são oferecidos por diferentes entidades dentro da Organização Virtual.

Neste cenário, inicialmente o *Workflow* executa o método ObterDBE no serviço PGDCNPJ, que retorna o caminho para um arquivo que contém o resultado de sua execução. Em seguida, o serviço fornecido pela Prefeitura processa o resultado do serviço anterior (ObterDBE), retornando um inteiro indicando sua autorização ou não. Nesse mo-

mento, caso não seja autorizado pelo serviço da prefeitura, o serviço de notificação é utilizado para contactar o usuário. Caso contrário, é realizado o cadastro estadual e federal em paralelo, utilizando o elemento `<flow>`. Continuando o processo, o serviço da Receita Federal verificará possíveis restrições ao registro, retornando um inteiro (LibFederal) contendo o valor que será utilizado pelo serviço da Junta Comercial para autorizar ou não o registro jurídico. A resposta é enviada ao usuário através do serviço de notificação.

```
<gpo:job name="CadastroSincronizado">
  <gpo:definitions name="work_Definitions">
    <gpo:variables>
      <gpo:variable name="FileForm" type="string" />
      <gpo:variable name="PDGFile" type="string" />
      <!-- ... -->
    </gpo:variables>

    <gpo:gservices>
      <gpo:gsh name="PGDCNPJ" uri="host:port/.../PGDCNPJ" type="Factory"
        <return_type="string"/>
      <!-- ... -->
    </gpo:gservices>
  </gpo:definitions>

  <gpo:process name="work_Process">
    <gpo:invoke name="PGDCNPJ" method="ObterDBE">
      <gpo:argument value="FileForm" type="string"/>
      <gpo:return variable="PDGFile" type="string"/>
    </gpo:invoke>

    <gpo:invoke name="LibPrefeitura" method="LiberacaoJuridica">
      <gpo:argument variable="PDGFile" type="string"/>
      <gpo:argument variable="PrefPathoutFile" type="string"/>
      <gpo:return variable="LibP" type="int"/>
    </gpo:invoke>

    <!-- Condicional -->
    <gpo:if variable="LibP" opr="==" value="1">
      <!-- Fluxo paralelo -->
      <gpo:flow>
        <gpo:invoke name="CadFederal" method="CadastroJuridico">
          <gpo:argument variable="PDGFile" type="string"/>
          <gpo:return variable="FileFederal" type="string"/>
        </gpo:invoke>

        <gpo:invoke name="CadEstadual" method="CadastroJuridico">
          <gpo:argument value="PDGFile" type="string"/>
          <gpo:return variable="FileEstadual" type="string"/>
        </gpo:invoke>
      </gpo:flow>

      <gpo:invoke name="LibFederal" method="LibJuridica">
        <gpo:argument value="PDGFile" type="string"/>
        <gpo:return variable="FileFederal" type="string"/>
        <gpo:return variable="FileEstadual" type="string"/>
        <gpo:return variable="LibFederal" type="int"/>
      </gpo:invoke>

      <gpo:invoke name="FUCE" method="LibFUCE">
        <gpo:argument value="PDGFile" type="string"/>
        <gpo:argument value="LibFederal" type="int"/>
        <gpo:return variable="FileofFUCE" type="String"/>
      </gpo:invoke>

      <gpo:invoke name="notificacao" method="notificaemail">
        <gpo:argument value="FileofFUCE" type="string"/>
      </gpo:invoke>
    </gpo:if>

    <gpo:else>
      <gpo:invoke name="notificacao" method="notificaemail">
        <gpo:argument value="PrefPathoutFile" type="string"/>
      </gpo:invoke>
    </gpo:else>
  </gpo:process>
</gpo:job>
```

Figura 9. Workflow em GPOL: Cadastro Sincronizado.

6. Conclusão

Com a sua orientação a serviços, Grades Computacionais podem ser utilizadas para superar vários desafios impostos para a integração de aplicações heterogêneas entre domínios, como as apresentadas pelo Governo Eletrônico. Neste artigo propomos uma plataforma (MidGov) que utiliza Grades Computacionais baseadas no GT4 para aplicações de Governo Eletrônico com adicional políticas de interação. Apresentamos os principais conceitos relativos ao trabalho e demonstramos um cenário de utilização da plataforma. A

principal contribuição do artigo se concentra no modelo de utilização da Grade Computacional por aplicações de Governo Eletrônico e sua implementação através dos serviços do GT4. Nosso trabalho futuro envolve a implementação de seleção dinâmica de serviços publicados na Grade para composição do Workflow.

Agradecimentos

Os autores agradecem ao CNPq (processo 134416/2008-2) pelo suporte financeiro.

Referências

- Alfonso, C. D., Caballer, M., Carrión, J. V., Hernández, V., Alfonso, C. D., Caballer, M., and Carrión, J. V. (2007). “gCitizen: a grid middleware for a transparent management of the information about Citizens in the public administration”. *Journal of Theoretical and Applied Electronic Commerce Research*, 2(1):18–32.
- Barnickel, N., Fluegge, M., and Schmidt, K.-U. (2006). “Interoperability in eGovernment through Cross-Ontology Semantic Web Service Composition”. 3rd European Semantic Web Conference 2006, Budva, Montenegro.
- Curbera, F., Khalaf, R., Mukhi, N., Tai, S., and Weerawarana, S. (2003). “The next step in Web services”. *Commun. ACM*, 46(10):29–34.
- Drigas, A. and Koukianakis, L. (2009). “Government Online: An E-Government Platform to Improve Public Administration Operations and Services Delivery to the Citizen.”. In *Visioning and Engineering the Knowledge Society. A Web Science Perspective*, pages 523–532. Springer.
- e-PING (2010). “e-PING: Padrões de Interoperabilidade de Governo Eletrônico”. <http://www.eping.e.gov.br/>. Acessado em 27 Fev. 2010.
- Foster, I. (2002). “What is the Grid? A Three Point Checklist”. *Argonne National Laboratory & University of Chicago*.
- Foster, I., Kesselman, C., Nick, J. M., and Tuecke, S. (2002). “The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration”.
- Globus Alliance. “Globus Toolkit Version 4 – GT4”. <http://www.globus.org/toolkit>. Acessado em 18 Fev. 2010.
- Goldchleger, A., Kon, F., Goldman, A., Finger, M., and Bezerra, G. C. (2004). “InteGrade: Object-Oriented Grid Middleware Leveraging Idle Computing Power of Desktop Machines.”. *Concurrency and Computation: Practice and Experience*, 16(5):449–459.
- Klusck, M., Fries, B., and Sycara, K. (2006). “Automated semantic web service discovery with OWLS-MX”. In *Autonomous agents and multiagent systems*, pages 915–922. ACM.
- Li, Y., Li, M., and Chen, Y. (2005). “Towards Building E-Government on the Grid.”. In Böhlen, M. H., Gamper, J., Polasek, W., and Wimmer, M., editors, *E-Government: Towards Electronic Democracy*, volume 3416 of *Lecture Notes in Computer Science*, pages 205–212. Springer.

- Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004). “*OWL-S: Semantic Markup for Web Services*”. Internet [http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/].
- Medjahed, B. and Bouguettaya, A. (2005). “*Customized Delivery of E-Government Web Services*”. *IEEE Intelligent Systems*, 20:77–84.
- OASIS (2006). “*Web Services Resource 1.2 (WS-Resource)*”. <http://www.oasis-open.org/specs/index.php#wsrf1.2>. Acessado em 18 Fev. 2010.
- Patra, M. R. and Das, R. K. (2007). “*SORIG: a service-oriented framework for rural information grid – an implementation viewpoint.*”. In *ICEGOV '07: Proceedings of the 1st international conference on Theory and practice of electronic governance*, pages 49–52, New York, NY, USA. ACM.
- Receita Federal (2003). “*Cadastro Sincronizado Nacional*”. www16.receita.fazenda.gov.br. Acessado em 18 Março 2010.
- Russell, N., ter Hofstede, A. H. M., Edmond, D., and van der Aalst, W. M. P. (2005). “*Workflow Data Patterns: Identification, Representation and Tool Support*”. In *Conceptual Modeling –Entity Relationship*, pages 353–368. Springer.
- Santos, I. J. G. (2008). “*CoGPlat: Um Middleware para serviços de Governo Eletrônico Centrados no Cidadão*”. PhD thesis, Tese de Doutorado, Instituto de Computação - Universidade Estadual de Campinas.
- Santos, I. J. G. and Madeira, E. R. M. (2007). “*E-Government and Grid Computing - Potentials and Challenges Towards Citizen-Centric Services.*”. In *Proceedings of the Ninth International Conference on Enterprise Information Systems, Volume SAIC, Funchal, Madeira - Portugal*, pages 144–148.
- Senger, H., Silva, F. A. B., de J. Mendes, M., Rondini, R., and de Farias, C. R. G. (2006). “*Grid Platforms for e-Democracy Applications.*”. *Computers and Communications, IEEE Symposium on*, 0:334–339.
- Senna, C. R. S. (2007). “*GPO - Um Middleware para Orquestração de Serviços em Grades Computacionais*”. *Dissertação de Mestrado, Instituto de Computação - Universidade Estadual de Campinas*.
- Sotomayor, B. and Childers, L. (2006). “*Globus toolkit 4: programming Java services*”. The Elsevier series in Grid computing. Morgan and Kaufmann, San Francisco, CA.
- Tangmunarunkit, H., Decker, S., and Kesselman, C. (2003). “*Ontology-Based Resource Matching in the Grid - The Grid Meets the Semantic Web*”. In *International Semantic Web Conference*, volume 2870, pages 706–721. Springer.
- W3C (2004). “*OWL –Web Ontology Language*”. <http://www.w3.org/TR/owl-features>. Acessado em 19 Fev. 2010.