Pós-Graduação em Ciência da Computação

"SSC4Cloud Editor – Uma Ferramenta para Modelagem de Processos de Negócio com Anotações de Segurança para a *Cloud*"

Por

Robson Wagner Albuquerque de Medeiros

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, ABRIL/2011

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ROBSON WAGNER ALBUQUERQUE DE MEDEIROS

"SSC4Cloud Editor – Uma Ferramenta para Modelagem de Processos de Negócio para a *Cloud*"

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR(A): Nelson Souto Rosa, PhD

RECIFE, ABRIL/2001

Catalogação na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Medeiros, Robson Wagner Albuquerque de
SSC4Cloud editor – Uma ferramenta para modelagem de processos de negócio com anotações de segurança para a cloud / Robson Wagner Albuquerque de Medeiros - Recife: O Autor, 2011.
xx, 96 p. : il., fig., tab.

Orientador: Nelson Souto Rosa.
Dissertação (mestrado) Universidade Federal de Pernambuco. CIn. Ciência da Computação, 2011.

Inclui bibliografia e anexo.

1. Redes de computadores. 2. Sistemas distribuídos. 3. Composição de serviços. I. Rosa, Nelson Souto (orientador). II. Título.

004.6 CDD (22. ed.) MEI2011 – 051

Dissertação de Mestrado apresentada por **Robson Wagner Albuquerque de Medeiros** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**SSC4CLOUD EDITOR – Uma Ferramenta para Modelagem de Processos de Negócio com Anotações de Segurança para a CLOUD**", orientada pelo **Prof. Nelson Souto Rosa** e aprovada pela Banca Examinadora formada pelos professores:

Profa. Carina Frota Alves
Centro de Informática / UFPE

Prof. Americo Tadeu Falcone Sampaio
Universidade de Fortaleza

Prof. Nelson Souto Rosa
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 29 de março de 2011.

Prof. Nelson Souto Rosa
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Gostaria de dedicar este trabalho em primeiro lugar a Deus e a toda minha família, em especial a minha mãe, Vera, meu pai, Fernando e minha esposa, Adriana.

AGRADECIMENTOS

- A Deus, por ter me concedido sabedoria, força, saúde e colocado em meu caminho todas as pessoas que, diretamente ou indiretamente, ajudaram a me tornar quem eu sou;
- A toda minha família, em especial a minha esposa, Adriana Medeiros, minha mãe, Vera Lúcia, meu pai, Antônio Fernando e meu irmão, Fernando Júnior, que sempre estiveram ao meu lado em todos os momentos da minha vida me dando apoio, carinho e muito amor;
- Ao meu orientador e amigo Nelson, por sua disposição em me ajudar e por suas idéias, sugestões, críticas e conselhos pertinentes;
- Aos meus amigos do mestrado, em especial Fernando Aires, Bruno Silva, André Souza, Júlio Damasceno, David Aragão, Erica Sousa e Karina Rodrigues que me ajudaram na realização deste trabalho com valiosas contribuições.

“Se você quer ser bem sucedido, precisa ter dedicação total, buscar seu último limite e dar o melhor de si mesmo.”

Ayrton Senna da Silva

RESUMO

Computação Orientada a Serviço e Computação em Nuvem são áreas de grande interesse no cenário atual da computação. Em particular, a possibilidade de composição de serviços em ambientes de nuvem tem um papel central na união destas duas áreas. A composição de serviços permite que serviços sejam criados a partir da utilização de serviços existentes e disponibilizados por terceiros. Neste contexto, o tratamento das questões de segurança (e.g., confidencialidade, autorização, não repúdio) é essencial para que os usuários dos serviços sintam-se seguros no momento de projetar e executar uma composição.

Projetar, implantar e executar composição de serviços e os seus modelos de segurança associados são atividades naturalmente complexas. Isso ocorre porque estas atividades normalmente envolvem múltiplas partes interessadas, incluindo especialistas no domínio de negócio, especialistas em segurança, desenvolvedores de serviços e equipe de TI. Além disto, enquanto ambientes de modelagem de processos de negócio são comuns, não há nenhum ambiente de desenvolvimento que permita que estas partes trabalhem de forma colaborativa na definição de um processo de negócio que inclua aspectos de segurança.

Este trabalho propõe uma ferramenta de modelagem de composição de serviços em BPMN (*Business Process Modeling Notation)* com anotações de segurança. A principal contribuição deste trabalho é o desenvolvimento de um ambiente, chamado SSC4Cloud *Editor*, onde processos de negócios com anotações de segurança e *profiles* de segurança podem ser criados, refinados e compartilhados com diferentes pessoas de diferentes organizações, permitindo a reutilização de conhecimento entre as várias partes envolvidas. O SSC4Cloud *Editor* é também responsável por gerar um artefato com todas as anotações (serviços e segurança) utilizado como base para geração de código executável da composição que é implantado em um ambiente de nuvem computacional.

Palavras-chave: Composição de Serviço, Processo de Negócio, Computação em Nuvem, Segurança, Arquitetura Orientada a Serviços, Web Service

ABSTRACT

Service-Oriented Computing and Cloud Computing are areas of great interest in the current scenario of computing. In particular, the possibility of service composition in cloud environments has a central role in the union of these two areas. The service composition allows services to be created from the use of existing services and made available by third parties. In this context, the handling of security issues (e.g., confidentiality, authorization, non-repudiation) are essential to service users feel safe when designing and running a composition.

Design, deployment and execution of services compositions and their associated security models are complex activities. This is because these activities usually involve multiple stakeholders that include business domain experts, security experts, service developers and IT operation teams. Moreover, while configuration environments of business processes are common, there is no development environment that allows these participants to work collaboratively on defining of a business process that includes security aspects.

This work proposes a modeling tool of service composition in BPMN (Business Process Modeling Notation) with security annotations. The main contribution of this work is the development of an environment, called SSC4Cloud Editor, where business processes with security annotations and security profiles can be created, refined and shared with different people from different organizations, allowing the reuse of knowledge among various stakeholders. The SSC4Cloud Editor is also responsible for generating an artifact with all the notes (security and services) used as a basis for generating executable code of the composition that is deployed in a cloud computing environment.

Keywords: Service Composition, Business Process, Cloud Computing, Security, Service-Oriented Architecture, Web Service

LISTA DE FIGURAS

LISTA DE TABELAS

LISTAGENS

SUMÁRIO

1 INTRODUÇÃO

Este capítulo tem como objetivo introduzir o trabalho no contexto da área de sistemas distribuídos, mais especificamente em composição de serviços. Inicialmente, o contexto e a motivação deste trabalho são apresentados mostrando a importância da composição de serviços e a necessidade da especificação de requisitos de segurança. Em seguida, o problema tratado na dissertação é caracterizado. Os trabalhos relacionados são brevemente introduzidos, juntos com as deficiências para tratar o problema apresentado. Logo após, os objetivos e as principais contribuições deste trabalho são apresentadas. Por fim é detalhada a estrutura da dissertação.

1.1 Contexto e Motivação

Criar serviços através da composição de serviços já existentes possibilita a criação de serviços mais complexos e facilita a integração de negócios entre empresas. A construção desses serviços pode iniciar com a modelagem do processo de negócio através de notações gráficas, e.g. BPMN (*Business Process Modeling Notation*) (OMG, 2009), e ser realizado através de uma composição de serviços utilizando ambientes de execução, denominados de *engine* de orquestração.

Criar, implantar e executar processos de negócios (BPMs) com requisitos de segurança associados são tarefas inerentemente complexas, pois esse processo envolve múltiplos *stakeholders:* especialistas no domínio de negócios e segurança, os desenvolvedores e, finalmente, a equipe operacional de TI para implantar os artefatos da composição (que incluem o processo de negócio executável, imagens do sistema operacional e arquivos de configuração para as *engines* do processo de execução e os módulos de segurança).

Atualmente não há nenhum ambiente de desenvolvimento que permita aos *stakeholders* trabalharem de forma colaborativa para a definição e execução de processos de negócio seguros.

1.2 Caracterização do Problema

Alguns problemas aparecem no contexto exposto na seção anterior, em especial a dificuldade em fazer o trabalho de integração de serviços em um ambiente com um grande número de *stakeholders*. Além disto, pode haver a necessidade de aquisição de recursos computacionais (por exemplo, servidores, armazenamento, serviços) para executar o processo de negócio modelado e muitas empresas podem não prover de recursos financeiros para essas aquisições. Estes fatos requerem que os ambientes onde as composições são executadas, as *engines* de orquestração, estejam localizados em uma arquitetura elástica, ou seja, que possam crescer ou diminuir conforme sua demanda.

Além dos problemas mencionados, a segurança aparece como uma das principais preocupações na área da composição de serviço. Baseado no fato de que os dados são enviados através da Internet, podendo passar por vários serviços, os usuários podem exigir sua proteção. Os requisitos de segurança controlam como os dados devem ser expostos aos participantes da composição. Esses requisitos podem incluir comunicação de mensagens segura e o gerenciamento do armazenamento desses dados. Clientes com os dados sensíveis, com necessidade de algum tipo de segurança, não vão optar por usar serviços que não fornecem suporte adequado para as suas necessidades de segurança.

Criar composições de serviço segura envolve vários *stakeholders* que precisam trabalhar integrados. Várias empresas, normalmente do mesmo ramo de atividade, possuem processos de negócio similares, assim como podem utilizar os mesmos serviços com os memos requisitos de segurança nas suas composições. Esses *stakeholders* e empresas precisam de um ambiente integrado e colaborativo para criar seus processos de negócios seguros e compartilhar seus conhecimentos para diminuir o tempo na criação desses processos e reutilizar conhecimentos de outros usuários.

1.3 Estado da Arte

Atualmente existem várias ferramentas desenvolvidas para composição de *Web services* através de meta-modelos que abstraem a complexidade do desenvolvimento e, conseqüentemente, propiciam uma maior facilidade no seu uso por usuários com conhecimentos limitados em SOA e *Web services*. Algumas destas ferramentas utilizam

linguagens proprietárias para modelar e até executar sua composição. No entanto, existe uma forte tendência na adoção e uso de padrões como BPMN (*Business Process Model and Notation)* (OMG, 2009) e WS-BPEL (*Business Process Execution Language*) (OASIS, 2007), os padrões mais utilizados para modelagem e execução de processo de negócio, respectivamente.

O BPMN é um padrão amplamente adotado para descrever processos de negócios. Porém, o padrão BPMN não prevê a possibilidade de especificação de requisitos não-funcionais. Para resolver este problema, extensões de BPMN vêm sendo desenvolvidas para permitir a especificação de requisitos de segurança (RODRÍGUEZ, FERNÁNDEZ-MEDINA e PIATTINI, 2007) (MENZEL, THOMAS e MEINEL, 2009).

Rodríguez (RODRÍGUEZ, FERNÁNDEZ-MEDINA e PIATTINI, 2007) propõe uma abordagem para descrever requisitos de segurança no nível de modelagem de processo de negócio sem preocupação na execução do processo modelado. Menzel (MENZEL, THOMAS e MEINEL, 2009), por sua vez, também propõe uma abordagem para descrever requisitos de segurança durante a modelagem do processo de negócio. Adicionalmente, Menzel se preocupa com a posterior tradução desse modelo em configurações concretas de segurança. Sua abordagem utiliza padrões de segurança para transformar as informações do processo de negócio da camada de modelagem para um modelo de segurança independe de plataforma.

1.4 Objetivos

Este trabalho apresenta um ambiente de modelagem de processos de negócios que permite a especificação de requisitos de segurança destes processos, ao mesmo tempo que permite o compartilhamento e a implantação destes processos em um ambiente de computação em nuvem. Na prática, os processos de negócios e as anotações de segurança podem ser definidos em conjunto por vários especialistas de uma ou mais organizações e compartilhados com parceiros comerciais em diferentes organizações. Além disso, os processos de negócio são traduzidos em composições de serviços que podem ser implantados e executados automaticamente em um ambiente *multi-tenant* baseado na nuvem.

Ambientes de Computação em Nuvem (WANG, LASZEWSK, *et al.*, 2010) são flexíveis em termos de alocação e liberação de recursos de computação por demanda, onde

um serviço flexível pode ser desenvolvido e implantado rapidamente e facilmente atualizado de acordo com as necessidades do negócio. Além disso, estes ambientes incluem benefícios interessantes, tais como baixa sobrecarga de gerenciamento, acesso fácil a uma grande variedade de aplicações e serviços, e compartilhamento de recursos entre um grande grupo de usuários (*multi-tenant*).

Desta forma, os principais objetivos deste trabalho são:

- Desenvolvimento de composição de serviço com anotações de segurança de forma colaborativa;
- Compartilhamento de conhecimento de negócio e segurança;
- Reuso de composição de serviços e configurações de segurança;
- Executar processos de negócio de forma segura na nuvem.

1.5 Estrutura do Trabalho

Além deste capítulo de introdução, o presente trabalho inclui ainda mais cinco capítulos. O Capítulo 2 introduz os conceitos básicos necessários ao entendimento deste trabalho: processos de negócio (*business processes*), arquitetura orientada a serviços (SOA), *web services*, composição de serviços, segurança de serviços e computação em nuvem.

No Capítulo 3 é apresentado o ambiente proposto nessa dissertação para modelagem de composição de serviço em BPMN com anotações de segurança para ser executada em ambientes de nuvem.

O estudo de caso deste trabalho é apresentado no Capítulo 4. A aplicação escolhida é o VTA (*Virtual Travel Agency*) (STOLLBERG, LAUSEN, *et al.*, 2004), uma agência que disponibiliza serviços de turismo através de um portal da Internet.

O Capítulo 5 apresenta os trabalhos relacionados e uma análise comparativa com o que está sendo proposto nesta dissertação.

Por fim, o Capítulo 6 apresenta as conclusões da dissertação, ressaltando as contribuições, as limitações e os trabalhos futuros a serem realizados.

2 CONCEITOS BÁSICOS

Neste capítulo serão introduzidos os principais conceitos e tecnologias relacionados com o tema desta dissertação. Inicialmente é apresentada uma visão geral sobre a modelagem de processos com BPMN. Em seguida, são introduzidos os principais conceitos de *SOA* (*Service-Oriented Architecture*), *Web Services* e segurança de serviços. Por fim, são apresentados conceitos básicos de computação na nuvem.

2.1 BPMN

BPMN (*Business Process Modeling Notation*) (OMG, 2009) é um padrão que define uma notação gráfica para criação de processos de negócios. Para permitir aos envolvidos na definição do processo de negócio um aprendizado e um entendimento fácil e rápido de modelagem com BPMN, sua especificação foi criada com apenas quatro conjuntos de elementos básicos: objetos de seqüência, objetos de conexão, *swimlanes* e artefatos.

Os principais elementos do BPMN estão contidos nos objetos de seqüência (Tabela 2.1), que tem como finalidade representar o comportamento do processo de negócio.

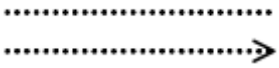
Tabela 2.1 - Objetos de Seqüência do Core BPMN

Elemento	**Descrição**	**Notação gráfica**
Evento	Todo processo BPMN deve iniciar e terminar com eventos. Porém, também podem haver eventos intermediários, que ficam localizados entre o evento de início e fim.	Início Intermediário Fim 
Atividade	Este elemento, como o próprio nome sugere, representa uma atividade do processo. Atividade pode ser do tipo *Tarefa*, *Processo* e *Sub-processo*.	

Gateway	Todo o fluxo do processo é controlado pelo *Gateway*. Com ele o fluxo do processo pode ser convergido ou divergido dependendo do tipo do *Gateway*.	

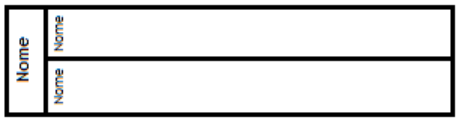
Os Objetos de Seqüência podem ser conectados entre si por meio de Objetos de Conexão. Além de conectar os objetos de sequência, esses elementos também provêem informação sobre o fluxo do processo de negócio (Tabela 2.2).

Tabela 2.2 - Objetos de Conexão do *Core* BPMN

Elemento	**Descrição**	**Notação gráfica**
Fluxo de Seqüência	Conecta as atividades para indicar a sua seqüência no processo.	
Fluxo de Mensagem	Mais de um participante pode ser representado no BPMN e este elemento modela o fluxo de mensagem trocado por eles.	
Associação	É utilizado para associar dados, texto e outros artefatos com os objetos de fluxo. Esse elemento, como exemplo, é utilizado para indicar se um objeto de dados está sendo lido ou criado por um objeto de seqüência, dependendo do sentido da seta.	

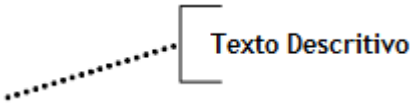
Um processo em BPMN pode conter mais de um participante na sua representação e um participante pode ser dividido em várias subdivisões. Os *Swimlanes* organizam as responsabilidades no processo por participantes ou subdivisão, conforme Tabela 2.3.

Tabela 2.3 - *Swimlanes* do *Core* BPMN

Elemento	Descrição	Notação gráfica
Pool	Um participante é representado no BPMN por um *Pool*. O *Pool* contém os processos de cada participante. Além disto, cada processo de negócio deve ter pelo menos um *Pool*.	
Lane	Uma *pool* poderá ter subdivisão para melhor representar e organizar as atividades de um processo	

A especificação BPMN define que "*artefatos são usados para prover informações adicionais sobre o Processo*" (OMG, 2009) e que modeladores e ferramentas estão livres para criar novos elementos deste conjunto, caso haja a necessidade. Em BPMN existem três artefatos (Objetos de Dados, Grupo e Anotação), como apresentados na Tabela 2.4.

Tabela 2.4 - Artefatos do *Core* BPMN

Elemento	Descrição	Notação gráfica
Objeto de dados	Objeto de dados representa um artefato usado no processo. Ele mostra como os dados são requeridos ou produzidos por atividades.	
Grupo	Um grupo pode ser usado para finalidades de documentação ou de análise.	
Anotação	Uma anotação é um mecanismo para fornecer informações adicionais para o leitor de um diagrama BPMN.	

A Figura 2.1 apresenta um exemplo de processo de negócio para compra de passagem aérea. Este processo de negócio tem como finalidade fazer uma reserva de vôo em uma companhia aérea, processar o pagamento e enviar uma confirmação para o cliente via SMS. A primeira tarefa BPMN (*Receber Dados do Cliente*) recebe os dados do cliente que serão processados por três outras tarefas (*Checar Disponibilidade de Vôo*, *Processar Pagamento* e *Enviar SMS de Confirmação*) antes do seu término.

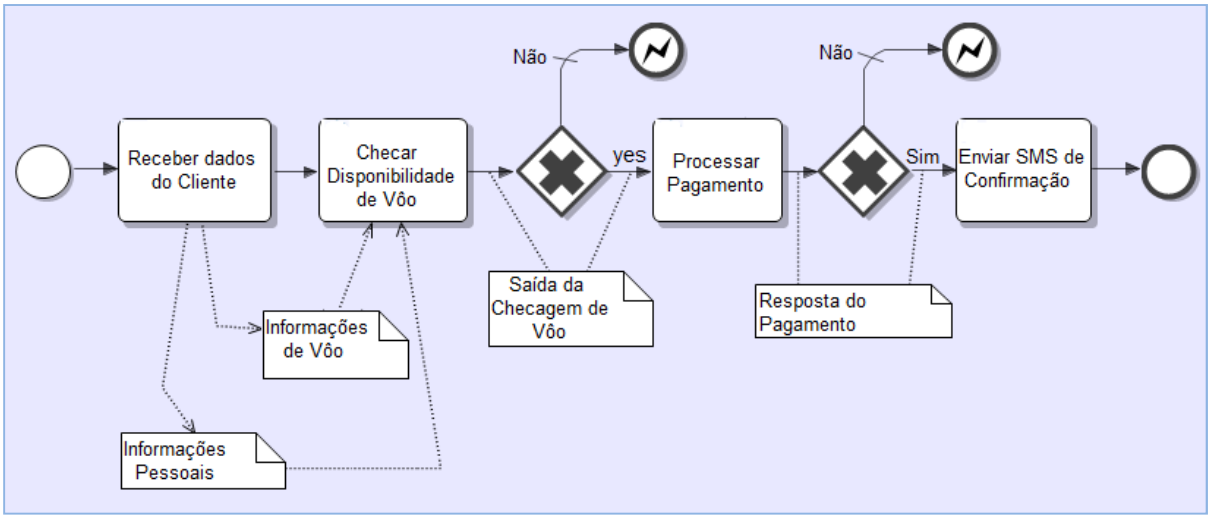


Figura 2.1 - Processo de Negócio em BPMN

Por ser uma notação de fácil entendimento, com um grande poder de expressividade e amplamente utilizado por analistas de negócio, a especificação BPMN define um mapeamento de seus elementos gráficos para WS-BPEL (OMG, 2009), uma linguagem de execução de processo de negócio, detalhado na Seção 2.2.2.

2.2 SOA

Atualmente as organizações precisam responder de forma rápida as oportunidades do mercado e a área de Tecnologia da Informação (TI) tem um papel de fundamental importância para que isso aconteça. Em geral, uma organização utiliza diferentes aplicações, atuais e legadas, espalhadas em diversos departamentos (ou áreas) ou até fora do seu domínio para realizar suas atividades. Essas aplicações precisam se comunicar de forma integrada com o objetivo de atingir agilidade e simplificar processos de negócio, tornando-os mais produtivos, frente à crescente e intensa competitividade do mercado (FURTADO, PEREIRA, *et al.*, 2009).

SOA (*Service-Oriented Architecture*) (PAPAZOGLOU e HEUVEL, 2007) é um estilo de arquitetura de software baseada nos princípios da computação distribuída onde as funcionalidades implementadas pelas aplicações devem ser disponibilizadas na forma de serviços. Esta arquitetura permite aos desenvolvedores superar muitos desafios da computação distribuída nas organizações, incluindo integração de aplicativos, gerenciamento de transações e políticas de segurança.

Em SOA a unidade de computação básica é o serviço e, desta forma, todos os componentes de software existente em uma empresa devem ser transformados em serviços de software que poderão ser disponibilizados internamente, entre as diversas aplicações existentes na organização, ou externamente, provendo serviços para fora da sua área administrativa. Neste segundo caso, a chance de ocorrer problemas de interoperabilidade é muito grande, pois cada empresa possui sua infra-estrutura com diferentes arquiteturas de hardware e principalmente de software. Para que isso não aconteça, serviços criados com base em SOA devem ser desenvolvidos baseados em padrões abertos (ERL, 2005).

A interoperabilidade dos sistemas em SOA acontece por meio de conjuntos de interfaces de serviços fracamente acopladas, onde essas interfaces provêem um contrato entre provedores e consumidores de serviços. Para que este contrato possa ser entendido por ambas as partes, é muito importante a utilização de uma IDL (*Interface description language*) (OMG, 2011), uma linguagem de descrição de interface, que ambos os lados consigam ler e interpretar.

Para um serviço poder ser descoberto e conseqüentemente utilizado por um consumidor de serviços, SOA provê o repositório de serviços. Se um provedor de serviço deseja disponibilizar seus serviços para serem utilizados pelos consumidores, ele deverá registrá-los no repositório onde posteriormente os consumidores poderão procurar pelo serviço desejado e usá-lo conforme sua necessidade. A Figura 2.2 ilustra como acontece o relacionamento entre o repositório, o provedor e o consumidor de serviço.

Figura 2.2 – Interação Entre Repositório, Provedor e Consumidor de Serviços

O repositório, embora seja importante, é um papel opcional na SOA, pois em muitos casos os serviços são disponibilizados de forma direta, entre o provedor e o consumidor, ou através de buscas na Internet.

SOA provê uma comunicação independente de protocolo e plataforma entre as aplicações distribuídas, onde os serviços não necessitam de detalhes técnicos dos outros serviços para a troca de informações. Isso viabiliza a integração entre muitos sistemas legados ainda existentes em muitas empresas, permitindo-o continuar operando e se comunicando com novos sistemas. A Figura 2.3 mostra a aplicação de uma camada de serviço em um sistema legado.

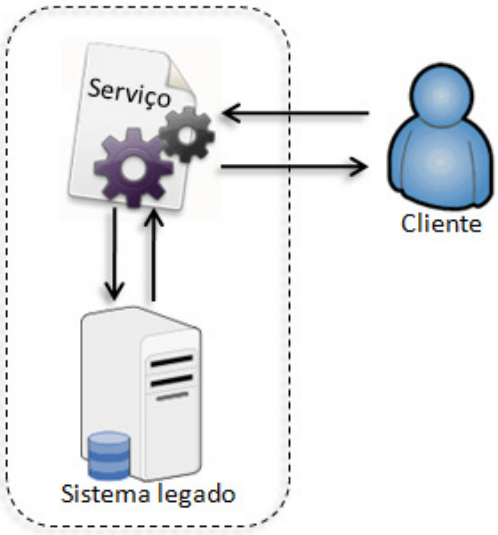


Figura 2.3 - Sistema Legado Encapsulado por Serviços

Uma das realizações de SOA mais usada atualmente é o *Web Services*. Seus detalhes e as tecnologias utilizadas na implementação de SOA serão apresentados na próxima seção.

2.2.1. Padrões de Web Services

Muitos sistemas de softwares necessitam interagir com outros sistemas através de uma rede corporativa ou da Internet. Tecnologias mais tradicionais como RMI (*Remote Method Invocation*) (ORACLE, 2011) e CORBA (*Common Object Request Broker Architecture*, (OMG, 2011) fornecem meios de aplicações interagirem, porém o grande número de diferentes linguagens de programação e os inúmeros mecanismos de segurança dificultam seus usos pelas organizações.

O uso *de Web Services* permite o desenvolvimento de aplicações complexas (COULOURIS, DOLLIMORE e KINDBERG, 2007) onde sistemas heterogêneos, desenvolvidos em diferentes plataformas, como exemplo, Microsoft .NET (MICROSOFT, 2009) e J2EE (ORACLE, 2011), são capazes de interagir através de protocolos padrões da Internet, possibilitando com isso a interação de aplicações novas e legadas, em redes locais ou através da Internet. A Figura 2.4 mostra as principais características dos *Web Services*.

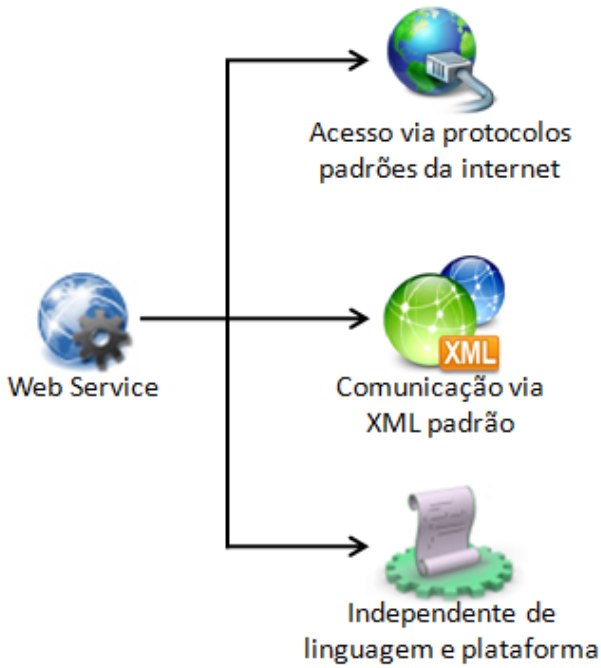


Figura 2.4 - Principais Características dos *Web Services*

Para que os *Web Services* consigam a transparência e a interoperabilidade desejadas, é fundamental o uso de tecnologias padronizadas e utilizáveis na Internet. A apresentação de dados e o empacotamento de mensagens trocadas entre clientes e *Web Services* são feitos em XML (*Extensible Markup Language*) (W3C, 2008).

O protocolo SOAP (*Simple Object Access Protocol*) (W3C, 2007), que é baseado em XML, especifica as regras de uso de XML para empacotar as mensagens dos serviços. Após o empacotamento, o SOAP transmite a mensagem através de protocolos como HTTP (*Hypertext Transfer Protocol*), TCP (*Transmission Control Protocol*) ou o SMTP (*Simple Mail Transfer Protocol*)(COULOURIS, DOLLIMORE e KINDBERG, 2007).

SOAP permite a interação síncrona e assíncrona pela Internet, especificada como um conjunto de informações em XML (W3C, 2007). Como mostrado na Listagem 2.1, as mensagens SOAP são compostas por um cabeçalho (Linhas 6-11) que contem informações pertinentes ao protocolo de comunicação, não visíveis para a aplicação, e seu corpo (Linhas 12-17) contendo os dados de entrada, de saída e de falhas (LOURIDAS, 2006).

Listagem 2.1 - Exemplo de Mensagem de Solicitação SOAP

```
1  <soapenv:Envelope
2            xmlns:q0="http://DefaultNamespace"
3  mlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
4            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
5            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
6      <soapenv:Header>
7        <n:alertcontrol xmlns:n="http://example.org/alertcontrol">
8            <n:priority>1</n:priority>
9            <n:expires>2011-01-22T14:00:00-05:00</n:expires>
10       </n:alertcontrol>
11     </soapenv:Header>
12     <soapenv:Body>
13        <q0:multiplicacao>
14           <q0:num1>2</q0:num1>
15           <q0:num2>5</q0:num2>
16        </q0:multiplicacao>
17     </soapenv:Body>
18 </soapenv:Envelope>
```

A Listagem 2.1 refere-se a uma mensagem de solicitação (*request*), com uma operação `multiplicacao` com dois argumentos de entrada (Linhas 13-16), de um *Web*

Service. Após a operação multiplicacao ser concluída, o serviço retornará uma mensagem SOAP de resposta (*response*) (Listagem 2.2) contendo o resultado da operação.

Listagem 2.2 – Exemplo de Mensagem de Resposta SOAP

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <soapenv:Envelope
3   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" ...
4        <soapenv:Body>
5              <multiplicacaoResponse
                xmlns="http://DefaultNamespace">
6                 <multiplicacaoReturn>10.0</multiplicacaoReturn>

7              </multiplicacaoResponse>
8        </soapenv:Body>
9  </soapenv:Envelope>
```

A definição da interface de um *Web Service* é feita usando WSDL (*Web Services Description Language*) (W3C, 2001), uma linguagem baseada em XML que, além de descrever os serviços, especifica como eles devem ser acessados e quais as operações e métodos disponíveis (FURTADO, PEREIRA, *et al.*, 2009). WSDL descreve uma interface do serviço que serve como um contrato entre o provedor e o consumidor do serviço.

A Listagem 2.3 mostra um documento WSDL de um serviço que possui a operação de multiplicação (Linhas 29-34) com dois parâmetros de entrada do tipo *float* (Linhas 9-10) e que retorna para o requisitante (consumidor) a multiplicação desses números (Listagem 2.3Linha 17).

Listagem 2.3 – Exemplo de Documento WSDL

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <wsdl:definitions targetNamespace="..."
3    ...
4    <wsdl:types>
5     <schema elementFormDefault="qualified"...>
6      <element name="multiplicacao">
7       <complexType>
8        <sequence>
9         <element name="num1" type="xsd:float"/>
10        <element name="num2" type="xsd:float"/>
11       </sequence>
12      </complexType>
13     </element>
14     <element name="multiplicacaoResponse">
15      <complexType>
```

```
16       <sequence>
17        <element name="multiplicacaoReturn" type="xsd:float"/>
18       </sequence>
19      </complexType>
20     </element>
21    </schema>
22   </wsdl:types>
23     ...
24     <wsdl:message name="multiplicacaoRequest">
25        <wsdl:part element="impl:multiplicacao"
    name="parameters">
26        </wsdl:part>
27     </wsdl:message>
28     <wsdl:portType name="Calculadora">
29        <wsdl:operation name="multiplicacao">
30           <wsdl:input message="impl:multiplicacaoRequest"
                    name="multiplicacaoRequest">
31         </wsdl:input>
32           <wsdl:output message="impl:multiplicacaoResponse"
                   name="multiplicacaoResponse">
33         </wsdl:output>
34        </wsdl:operation>
35     </wsdl:portType>
36   ...
37  </wsdl:definitions>
```

Antes que um serviço possa ser utilizado, sua interface em WSDL deve estar disponível para o consumidor. Muitos provedores de *Web Services* normalmente disponibilizam o WSDL na Internet ou fornece-o diretamente aos consumidores interessados no serviço. Porém, estas informações podem ser disponibilizadas em um UDDI (*Universal Description Discovery & Integration*) (OASIS, 2004). O UDDI define um conjunto de serviços de apoio à descrição e descoberta de serviços por empresas e organizações provedoras de *Web Services*.

O provedor do serviço com o objetivo de disponibilizar os seus *Web Services* deve definir a interface deste serviço em WSDL e publicá-la no repositório UDDI (ver Figura 2.5). Uma vez publicado, o cliente (consumidor do serviço) poderá fazer uma busca no repositório, obtendo com isso a interface publicada, criar sua aplicação com base nessa interface e comunicar-se com o respectivo serviço.

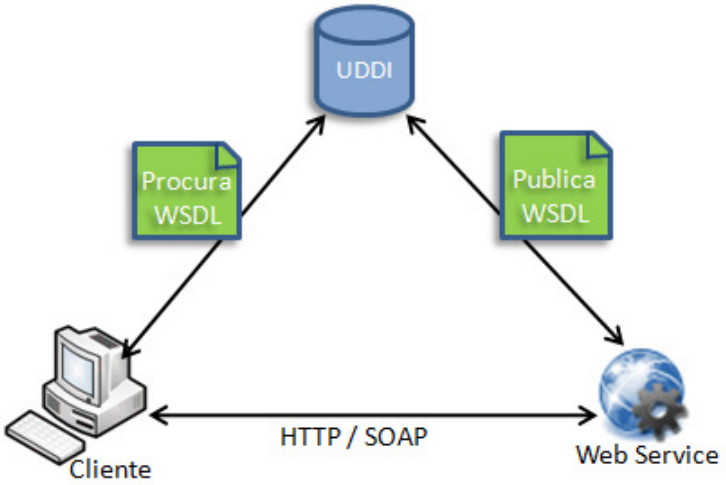


Figura 2.5 - Visão Geral do UDDI

2.2.2. Composição de Serviços

A composição de serviços tem surgido como uma importante estratégia para permitir a colaboração de aplicações entre empresas (*Business-to-Business*) (ZENG, BENATALLAH, *et al.*, 2004). É possível um provedor de serviço criar um serviço combinando várias operações de diferentes serviços, de diferentes provedores, e disponibilizá-lo para seus clientes sem que eles tenham nenhum conhecimento que o serviço oferecido por este provedor é, na verdade, uma combinação de serviços existentes. Para isso, é necessária uma linguagem para definição da composição de serviços. Idealmente, esta linguagem deve ser padronizada e, no caso de *Web Service*, precisa ser baseada em XML.

WS-BPEL (*Web Services Business Process Execution Language*) (OASIS, 2007) é uma linguagem para construção de composição de *Web Services*. Ela possui algumas construções de uma linguagem imperativa, como execução seqüencial, variáveis, comandos de atribuição e repetição. A diferença essencial está no fato de que ela descreve as interações entre os processos e seus parceiros. Nela são descritas as regras de negócio da composição, como os serviços serão compostos, qual o passo a passo de execução desses serviços e como suas entradas e saídas se relacionam e são tratadas.

Uma composição em WS-BPEL é vista externamente como um simples *Web Service* e, portanto, precisa fornecer uma interface em WSDL para que seus clientes possam invocá-la.

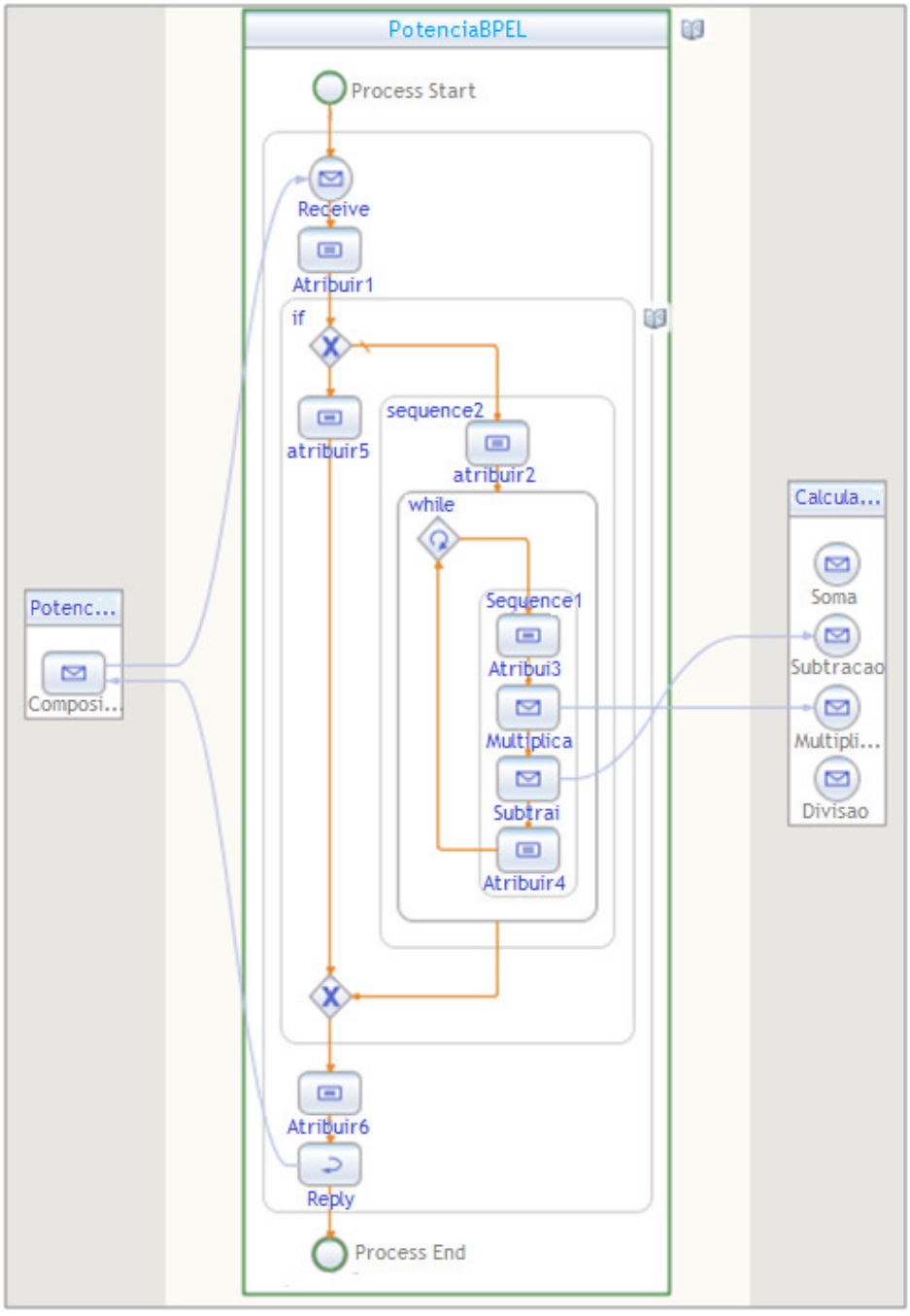


Figura 2.6 - WS-BPEL Para Calcular a Potência um Número Elevado a Outro Utilizando Multiplicação.

A Figura 2.6 apresenta um simples processo de negócio descrito em WS-BPEL para calcular a potência de um número (base) elevado a um expoente utilizando multiplicações. Após o cliente requisitar o cálculo, informando a base e o expoente, o processo de negócio

verifica se o expoente é igual a 0 (zero) e, caso o seja verdade, a potência recebe o valor 1 (um) como resposta e retorna para o cliente. Caso contrário, a potência receberá o valor da base e o processo entrará em um *loop* onde será invocado o serviço de calculadora para multiplicar a base com a potência até a quantidade de vezes igual ao especificado no expoente. Dentro do *loop* também será invocada a operação de subtração do serviço de calculadora para auxiliar na sua condição de parada. Uma parte do código WS-BPEL em XML deste processo pode ser visto na Listagem 2.4.

Listagem 2.4 - Parte do XML do WS-BPEL Para Calcular a Potência de um Número Elevado a Outro Utilizando Multiplicação.

```
1  <process name="PotenciaBPEL"
2     ...
3     <partnerLinks>
4        <partnerLink name="CalculadoraPartnerLink" .../>
5        <partnerLink name="PotenciaBPEL" .../>
6     </partnerLinks>
7     <variables>
8        <variable name="potencia" type="xsd:int"/>
9        <variable name="condicao" type="xsd:int"/>
10       ...
11    </variables>
12    <sequence>
13       <receive name="Receive" partnerLink="PotenciaBPEL"
   operation="ComposicaoCalculadoraOperation" . . . />
14       <assign name="Atribuir1">
15         ...
16          <copy>
17             <from>0</from>
18             <to variable="potencia"/>
19          </copy>
20       </assign>
21       <if name="If">
22          <condition>$ComposicaoCalculadoraOperationIn.expoente
   = 0</condition>
23         ...
24          <else>
25             <sequence name="Sequence2">
26                . . .
27                <while name="While">
28                   <condition>$condicao > 1</condition>
29                      <sequence name="Sequence1">
30                         <assign name="Atribuir3">
31                            ...
32                         </assign>
33                         <invoke name="multiplica"
   partnerLink="CalculadoraPartnerLink" operation="multiplicacao"
   ... />
34                         <invoke name="subtrai"
   partnerLink="CalculadoraPartnerLink" operation="subtracao" ...
   />
35                         ...
36                   </sequence>
```

```
37                </while>
38             </sequence>
39          </else>
40       </if>
41       ...
42       <reply name="Reply" partnerLink="PotenciaBPEL"... />
43    </sequence>
44 </process>
```

Os principais elementos de uma composição em WS-BPEL podem ser vistos nesta listagem:

- `process`: indica o processo propriamente dito (Linhas 1-44);
- `partnerLinks`: contém a lista dos servidores (partnerLink) onde estão disponíveis os *Web Services* (Linhas 3-6);
- `variables`: contém a lista de variáveis (variable) utilizadas no processo (Linhas 7-11);
- `sequence`: define um conjunto de operações invocadas em seqüência. Um WS-BPEL pode incluir mais de uma seqüência (Linhas 12-43);
- `receive`: aguarda uma mensagem de invocação do cliente chegar para dar início ao processo (Linha 13);
- `assign`: utilizado para manipular as variáveis. Com o *assign* pode-se atribuir valores em variáveis através de outras variáveis ou constantes (Linhas 14-20);
- `if`: define uma seqüência de operações que só serão executadas se a condição definida em condition (Linha 22) for verdadeira. Caso a condição não seja atendida o fluxo é transferido para a seqüência do seu elemento else (Linhas 24-39);
- `while`: representa um *loop* que possui uma seqüência de operações (Linhas 29-38) que serão executadas enquanto a condição em condition (Linha 28) for verdadeira;
- `reply`: envia para o cliente a resposta de uma operação invocada sincronamente. (Linha 42)

Para uma WS-BPEL ser executada é necessário um ambiente de execução específico para ela. As *Engines* de Orquestração são esses ambientes responsáveis por executar a lógica de negócio especificada em WS-BPEL.

Engines de orquestração são responsáveis por executar composições de serviços, fazendo invocações aos *Web Services*, controlando entradas e saídas da composição e realizando suas regras de negócio de acordo com o que está especificado na composição.

Linguagens de composição de serviço não são muito amigáveis para os usuários, pois praticamente todas são baseadas em XML e possuem muitos parâmetros para gerenciar. Para facilitar a composição existem algumas ferramentas que provêem suporte à modelagem e execução de composição de *Web Services* de forma gráfica. A Tabela 2.5 mostra algumas dessas ferramentas e quais as linguagens de execução e *engine* utilizadas por elas. Pode-se observar na Tabela 2.5 que a maioria das ferramentas dá suporte a WS-BPEL.

Tabela 2.5 - Ferramentas para Composição de Serviços

Ferramenta	Linguagem de Execução	Engine de Orquestração
Eclipse BPEL	WS-BPEL	Apache ODE
Netbeens 6.1	WS-BPEL	Glass Fish
JOpera	JVCL	Engine Proprietária
ActiveVOS Designer	WS-BPEL	ActiveVOS Server
WebRatio	WebML	Engine Proprietária
Oracle SOA Suite/JDeveloper	WS-BPEL	Oracle BPEL Process Manager

2.3 Segurança em *Web Services*

Segundo Coulouris (COULOURIS, DOLLIMORE e KINDBERG, 2007), "*há uma necessidade generalizada de medidas para garantir a privacidade, a integridade e a disponibilidade dos recursos em sistemas distribuídos*" e com *Web Services* não é diferente. As mensagens SOAP trocadas entre os *Web* Services podem passar por intermediários não confiáveis ou desconhecidos antes de chegarem ao seu destino (Figura 2.7) e, portanto, não é suficiente aplicar a segurança apenas na camada de transporte, ela deve ser aplicada também na própria mensagem (ONYSZKO, 2002).

A Figura 2.7 mostra um cliente invocando um *Web Service* (C*omposição 1*). Este *Web Service* (C*omposição 1*) por sua vez invoca outros dois *Web Services* (*Web Service A* e *Composição 2*) e o *Web Service Composição 2* também invoca outro *Web Service* (*Web*

Service B) e isso tudo sem que o cliente tenha conhecimento do que está acontecendo realmente. Nesta troca de mensagens, dados confidenciais podem ser acessados por indivíduos não autorizados, comprometendo a confidencialidade, ou mesmo sofrerem ataques maliciosos, podendo comprometer a integridade e disponibilidade desses dados.

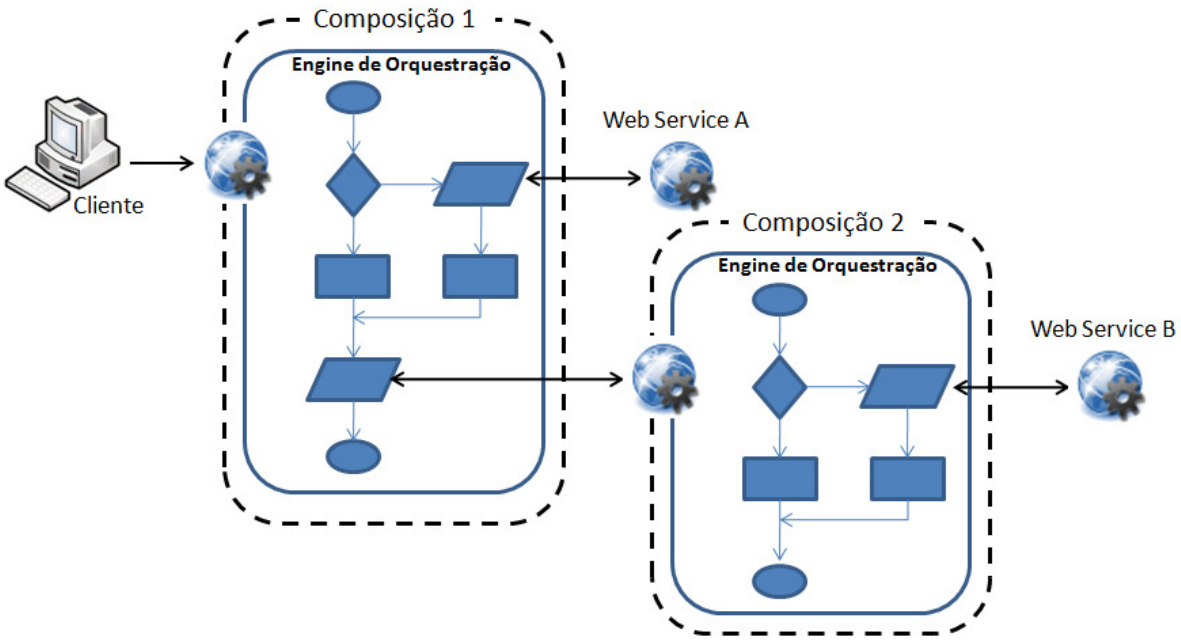


Figura 2.7 - Composição de *Web Services* Invocando Outra Composição.

Com a exposição dos serviços na Web e com a troca de informações provenientes da comunicação com esses serviços, os *Web Services* podem sofrer vários tipos de ameaças. Indivíduos não autorizados podem querer obter dados ou acessar *Web services* indevidamente, assim como tentar se passar por outros para enviar ou receber mensagem, podendo até modificar o conteúdo nela existente. Mecanismos como criptografia e autenticação são muito importantes para garantir a integridade e a confidencialidade, tanto dos dados como do próprio *Web Service*.

Mensagens SOAP utilizadas por *Web Services* podem ser protegidas através da especificação *WS-Security* (ATKINSON, DELLA-LIBERA, *et al.*, 2002). O *WS-Security* estende o padrão SOAP para incluir mecanismos de segurança que podem ser usados na construção de *Web Services* seguros e manter a integridade e confidencialidade das mensagens (OASIS, 2006). Esta especificação foi projetada para ser usada com uma grande variedade de modelos de segurança, incluindo PKI (KUMAR, PRAJAPATI, *et al.*, 2010), Kerberos (COULOURIS, DOLLIMORE e KINDBERG, 2007) e SSL (OPPLIGER, 2009), tecnologias de criptografia, *tokens* de segurança (LAVARACK e COETZEE, 2010) e

timestamps dentro de um elemento localizado no cabeçalho da mensagem SOAP chamado `<wsse:Security>` (Linhas 4-8 da Listagem 2.5).

Os elementos presentes no `<wsse:Security>` , são:

- *UsernameToken*: Utilizado para carregar informações de autenticação do usuário, como nome do usuário e senha;
- *BinarySecurityToken*: Contém informações que não se encaixam no formato XML e necessitam de um tipo especial de codificação (certificados X.509);
- *SecurityTokenReference*: Utilizado como *token* de segurança quando as informações sobre certificados estão localizadas em outro local;
- *Signature*: Contém informações sobre uma assinatura digital;
- *KeyInfo*: Especifica a chave usada para criptografar a mensagem;
- *ReferenceList*: Utilizado para identificar quais partes da mensagem estão criptografadas;
- *EncryptedData*: Expressa as partes da mensagem que estão criptografadas.

A Listagem 2.5 mostra um exemplo do uso de *token* para validação de usuário em uma mensagem SOAP. O *token* utilizado no exemplo foi o *UsernameToken* (Linhas 5-7). O atributo *Username* (Linha 6) informa o nome do usuário que pretende ser autenticado no *Web Service*.

Listagem 2.5 - Autenticação de Usuário com *WS-Security*

```
1  <S11:Envelope xmlns:S11="..." xmlns:wsse="...">
2      <S11:Header>
3              ...
4          <wsse:Security>
5              <wsse:UsernameToken>
6                  <wsse:Username>Pedro</wsse:Username>
7              </wsse:UsernameToken>
8          </wsse:Security>
9              ...
10     </S11:Header>
11     ...
12 </S11:Envelope>
```

Esta especificação permite múltiplas assinaturas e formatos de assinaturas em diferentes partes de uma mensagem e tanto o corpo quanto o cabeçalho da mensagem podem ser criptografados. Isso permite que uma mensagem SOAP possa percorrer vários caminhos,

passando por diferentes usuários onde cada um não terá acesso a informações indevidas, proporcionando assim um maior grau de segurança.

2.4 Computação em Nuvem

Computação em Nuvem tem como característica prover serviços de computação por demanda, onde o cliente pode adquirir aquilo que for necessário por um período desejado e pagar apenas o que foi utilizado (*Pay-per-Use*), com alta confiabilidade, escalabilidade e disponibilidade, utilizando a Internet como meio.

Os recursos da nuvem podem ser adquiridos na medida em que forem necessários. Com isso, empresas podem não ter que fazer grandes investimentos na infra-estrutura de TI, podendo contratar uma quantidade de recurso a priori e, a medida de sua necessidade, adquirir mais ou diminuir caso não seja mais necessário.

Empresas como Amazon, que já tinha uma grande quantidade de recursos de computação para seu uso, mas em alguns momentos ociosos, perceberam que podiam criar novos negócios vendendo esses recursos como serviço por meio da nuvem. A Amazon oferece a terceirização dos seus recursos através dos serviços EC2 (*Amazon Elastic Compute Cloud*) (AMAZON, a) e do S3 (*Simple Storage Service*) (AMAZON, b). A Google é outro exemplo de provedor de serviços na nuvem. Ela disponibiliza tanto softwares na nuvem, como é o caso do Google *Docs*, como também uma plataforma para desenvolvimento de aplicações de nuvem, o Google *App* Engine.

Em geral, os serviços da Computação em Nuvem são divididos em três categorias: *Infrastructure-as-a-Service* (IaaS), *Platform-as-a-Service* (PaaS) e *Software-as-a-Service* (SaaS) (GONG, LIU, *et al.*, 2010; WANG, LASZEWSK, *et al.*, 2010; PENG, ZHANG, *et al.*, 2009; ARMBRUST, FOX, *et al.*, 2009; ZHANG, CHENG e BOUTABA, 2010) (Figura 2.8).

Infrastructure-as-a-Service provê recursos de computação como processamento, armazenamento e rede. A Amazon é um exemplo de provedor de IaaS por meio dos seus serviços EC2 e S3. Plataforma como serviço fornecendo aos desenvolvedores de aplicação interfaces para abstrair a infra-estrutura da Nuvem. Um bom exemplo é a *Google App Engine* (GOOGLE) que fornece aos desenvolvedores de aplicação uma API (*Application Programming Interface*) para desenvolvimento na Nuvem. Software como serviço são as

aplicações providas para os clientes por meio da Internet, normalmente através de *Web Services*. Esse modelo tira do cliente a responsabilidade da manutenção do software e reduz o custo de aquisição de software por proporcionar a compra por demanda (WANG, LASZEWSK, *et al.*, 2010).

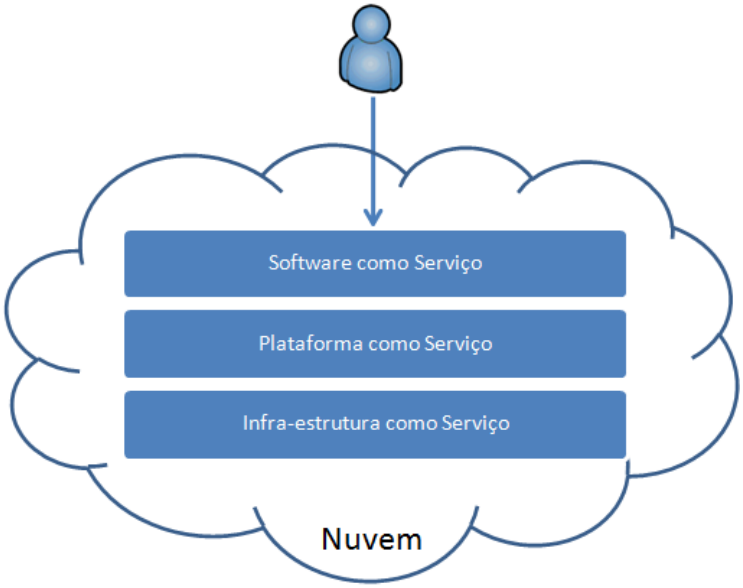


Figura 2.8 - Camadas da Nuvem

A Nuvem pode ser do tipo pública, privada ou híbrida (PENG, ZHANG, *et al.*, 2009). Nuvem privada é proprietária, onde apenas uma organização tem acesso a sua infra-estrutura. Nuvems privadas podem ser criadas utilizando softwares open sources como Eucalyptus (NURMI, WOLSKI, *et al.*, 2009), uma versão *Open Source* do Amazon EC2 e o OpenNebula (OPENNEBULA, 2011). Nuvem pública são vistas por mais de uma organização, possuindo mais de um cliente e administrada por aquela que a comercializa. A Amazon EC2 é um exemplo deste tipo de nuvem. Ela provê recursos de computação para vários clientes. A híbrida consiste de uma infra-estrutura formada tanto por nuvem pública como privada.

Uma nuvem pública normalmente está compartilhando recursos (hardware e software) com vários usuários (*tenants*) e é de fundamental importância que problemas e informações de um usuário não sejam perceptíveis para os demais. Para isso acontecer, provedores de serviços da nuvem devem assegurar que todos os usuários na nuvem estejam logicamente separados (TAURION, 2009; CHONG, CARRARO e WOLTER, 2006). Conforme Taurion (TAURION, 2009) essa separação pode ser feita de várias formas:

- Isolando todos os *tenants* por hardware, onde cada um ficará hospedado em um hardware diferente;
- Isolando os *tenants* por virtualização (SINGH, 2004), onde os *tenants* ficariam nas mesmas máquinas físicas, mas em máquinas virtuais diferentes; e
- Separados por instâncias de banco de dados ou compartilhando o mesmo banco de dados, porém com esquemas separados, onde cada cliente acessa apenas os objetos de banco de dados (e.g. tabelas, visões e funções) que ele tem permissão.

2.5 Considerações finais

Este capítulo introduziu os conceitos básicos usados na dissertação. Inicialmente, noções de modelagem de processo de negócio em BPMN. Noções sobre arquitetura orientada a serviços (SOA), *Web Services*, composição e segurança de serviços também foram apresentadas. Por fim, foram apresentados características da computação em nuvem, tipos de nuvem e estratégias existentes para isolamento dos clientes na nuvem.

3 SSC4CLOUD EDITOR – UM AMBIENTE DE MODELAGEM DE PROCESSOS DE NEGÓCIO COM ANOTAÇÕES DE SEGURANÇA

Este capítulo detalha o SSC4Cloud Editor, um ambiente de modelagem de processos de negócio com anotações de segurança que executa as composições em uma nuvem computacional. Inicialmente serão mostrados os princípios e fundamentos básicos sobre os quais a proposta foi concebida e em seguida as fases de desenvolvimento do SSC4Cloud Editor.

3.1 Visão Geral

Como já mencionado no Capítulo 1, este trabalho propõe uma ferramenta de modelagem de composição de serviços em BPMN com anotações de segurança para um ambiente de execução na nuvem como parte de uma solução chamada SSC4Cloud (*Secure Service Composition for Cloud*). O cenário de uso desta solução pode ser observado na Figura 3.1, onde diferentes *stakeholders*, trabalhando em diferentes locais, podem criar processos de negócio com anotações de segurança de forma colaborativa e executá-los em um ambiente localizado na nuvem.

O SSC4Cloud Editor é uma ferramenta executada localmente, no computador do usuário, para criar e compartilhar modelos de negócio e interagir remotamente com os repositórios de modelos e um ambiente de execução baseado em máquinas virtuais na nuvem (*Orchestration Engine* e *VM Manager*). Cada usuário é associado a um *workspace* na nuvem para armazenar artefatos de modelagem e de execução produzidos pelo editor. O SSC4Cloud Editor produz modelos, o *Translator* produz configurações de segurança e de execução baseados nos modelos produzidos pelo editor e a *Auxiliary Engine* executa e monitora os resultados da execução da composição. No ambiente *multi-tenant*, um *workspace* contém os arquivos de execução do usuário, como as configurações do processo de negócio do WS-BPEL e os arquivos de configuração do módulo de segurança.

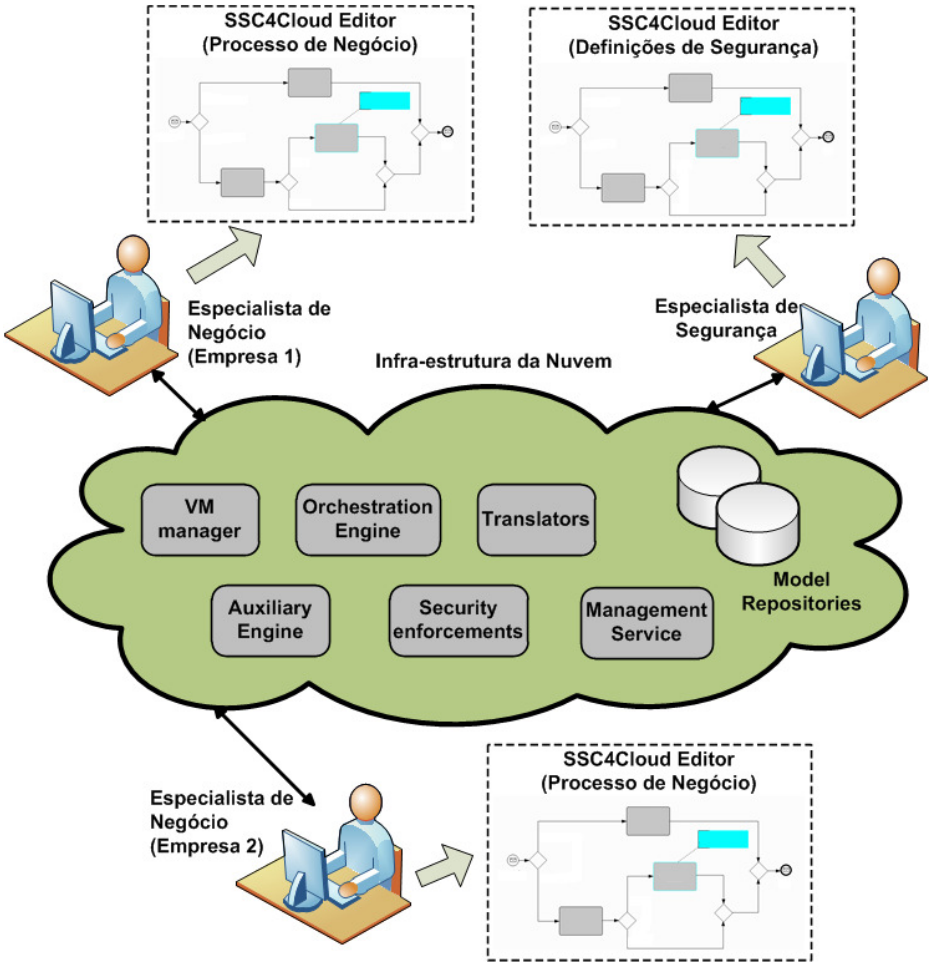


Figura 3.1 - Modelando, Compartilhando, Instalando e Executando um Processo de Negócio com Requisitos de Segurança na Nuvem

3.2 Definição de Requisitos

Observando o cenário apresentado na Figura 3.1, é possível observar algumas características desejáveis da SS4Cloud Editor. Primeiro, tendo em vista a necessidade de executar composições de serviços de forma segura, a ferramenta precisa fornecer recursos para analistas modelarem composições de serviços e exprimirem os requisitos de segurança da composição. Segundo, diferentes composições podem necessitar da mesma configuração de segurança em suas invocações. Desta forma, diferentes *profiles* de segurança poderão ser criados e reutilizados em diferentes composições de serviços. Terceiro, empresas diferentes, normalmente do mesmo ramo de atividade, podem possuir processos de negócios semelhantes. Essas empresas podem ter o desejo de compartilhar suas experiências de processos de negócios entre si para reutilização ou refinamento. Quarto, SSC4Cloud Editor poderá ser utilizado por especialistas em segurança que queiram criar *profiles* de segurança e

compartilhá-los com outras empresas para serem utilizadas em seus processos de negócio. Quinto, o ambiente de execução dos processos de negócios da solução SSC4Cloud estará localizado na nuvem e sendo utilizado por diferentes usuários. Portanto, os usuários do editor não deverão ter acesso a recursos de propriedades de outro usuário sem seu devido consentimento. Além disto, como o ambiente de execução dos processos está localizado na nuvem (distribuído), em um ambiente aberto de computação em nuvem denominado Open Cirrus (OPEN CIRRUS, 2010), todos os artefatos provenientes dela deverão permanecer atualizados na nuvem. Por fim, a ferramenta deverá ter a capacidade de instalar, executar e parar um processo de negócio em seu ambiente de execução na nuvem.

Além das características mencionadas, as anotações de segurança da ferramenta precisam ser apresentadas de forma estruturada. Para isto, será adotado o conjunto de abstrações de segurança definidos por Rosa (ROSA, 2001) e posteriormente estendidos pelo Sec-MoSC (SOUZA, SILVA, *et al.*, 2009):

- *NF-Attribute*: representa um requisito não funcional que pode ser primitivo, como por exemplo, integridade e confidencialidade, como também composto, como é o caso de segurança que tem, por exemplo, integridade e confidencialidade como requisitos primitivos.
- *NF-Action*: abstração usadas para expressar os mecanismos de implementação dos requisitos não funcionais (*NF-Attribute*). Elas são compostas por um conjunto de propriedades, representada como uma *tupla de <nome, valor>*. Como exemplo de *NF-Actions* tem-se *UseCryptography*, que refere-se ao uso da criptografia para assegurar a segurança na composição e é composta das seguintes propriedades: Tipo de criptografia (Simétrica ou Assimétrica); algoritmo; parte da mensagem criptografada; e o tamanho da chave de criptografia.
- *NF-Statement*: é uma restrição de alto nível, muitas vezes descrita em termos de níveis (alto, médio, baixo, personalizado), definida sobre *NF-Attribute*. Na prática, a *NF-Statement* implica um conjunto de *NF-Actions* que devem ser tomadas para alcançar um objetivo particular. Por exemplo, o conjunto de *NF-Actions* usado para implementar uma alta confidencialidade provavelmente será diferente do utilizado para implementar uma baixa confidencialidade.

A partir das características gerais apresentadas anteriormente, foram identificados os seguintes requisitos funcionais para a ferramenta:

- **[RF01]** A ferramenta deverá ser um *plugin* do Eclipse para reutilizar e estender todas as funcionalidades já implementadas no Sec-MoSC (SOUZA, SILVA, *et al.*, 2009);
- **[RF02]** A ferramenta deve possuir uma visão para modelagem de processos de negócios em BPMN com anotações de segurança para possibilitar que analistas de negócio modelem composições de serviços seguras mesmo sem conhecimentos técnicos necessários para implementá-la em um ambiente de execução;
- **[RF03]** Deve possuir uma palheta lateral com elementos gráficos de segurança para serem inseridos nos processos de negócio de forma rápida e fácil para não comprometer a usabilidade da ferramenta que já possui o mesmo componente para os elementos de BPMN;
- **[RF04]** Uma visão para configuração dos requisitos de segurança conforme anotação no diagrama e de acordo com as especificações do Sec-MoSC para auxiliar o especialista de segurança a configurar os aspectos de segurança sem que ele tenha a necessidade de conhecer os detalhes do processo e dos requisitos funcionais da composição;
- **[RF05]** Uma visão para configuração da composição dos serviços por especialistas em *Web Services* para que, assim como a visão de negócio e de segurança, esses profissionais tenham uma área específica para auxiliá-los na criação de uma composição executável;
- **[RF06]** Possibilitar criação e configuração de *profiles* de segurança por especialistas para que requisitos de segurança sejam personalizados com possibilidade de serem reutilizados em diferentes processos de negócio;
- **[RF07]** Permitir que usuários salvem e recuperem projetos e processos de negócios da nuvem, pois esses recursos devem estar disponíveis para os usuários sempre que for necessário, independente da instância do editor utilizada por ele;
- **[RF08]** Permitir o compartilhamento de projetos e processos de negócios entre usuários para que eles possam compartilhar conhecimento entre si e

possam prover o reuso de processos de negócios entre organizações que possuam atividades semelhantes;

- **[RF09]** Permitir que *profiles* de segurança sejam compartilhados entre usuários, possibilitando o compartilhamento de conhecimento de segurança e a reutilização destes *profiles* por outras organizações conforme o desejo do especialista de segurança que o criou;
- **[RF10]** Ter uma tela para o usuário fornecer suas informações para serem utilizadas na sua autenticação e para utilização dos seus recursos da nuvem;
- **[RF11]** O usuário deverá ser autenticado na nuvem antes de efetuar qualquer chamada de serviço na SSC4Cloud para evitar que os recursos deste ambiente sejam utilizados por usuários não autorizados ou que um usuário acesse indevidamente recursos oriundo de outros usuários;
- **[RF12]** Provê aos usuários a possibilidade de instalar seus processos de negócios na *engine* de orquestração da SSC4Cloud de forma simples para posterior execução na nuvem;
- **[RF13]** Provê aos usuários a possibilidade de iniciar seus processos de negócios na nuvem para que esses processos não fiquem apenas na fase de modelagem, mas também sejam executados em uma engine de orquestração que suporte os requisitos de segurança anotados e configurados;
- **[RF14]** Após seus processos de negócios estarem em execução na nuvem, o usuário deverá ser capaz de pará-los quando necessário, pois problemas de modelagem e configuração podem ser identificados após a composição iniciar sua execução e o usuário deve estar apto para interromper o processo, corrigi-lo e depois poder executá-lo novamente.

3.3 Arquitetura

Antes de descrevermos a arquitetura do SS4Cloud Editor, é preciso apresentar a arquitetura da solução SS4Cloud, pois o editor é um dos elementos principais desta arquitetura.

3.3.1. Arquitetura SSC4Cloud

A arquitetura da solução SS4Cloud foi inicialmente projetada para suportar um ambiente de execução privado, onde cada empresa era responsável pelo gerenciamento do seu domínio (SOUZA, SILVA, *et al.*, 2009). Posteriormente, esta arquitetura foi modificada para permitir a execução dos processos de negócio na nuvem. A principal modificação na arquitetura para migrá-la para o ambiente de nuvem atender ao requisito de multiusuário (*multi-tenant*) proveniente de uma nuvem pública. Nesta condição, foram criados mecanismos para gerenciamento de *tenant* (usuário) e para prover o compartilhamento de conhecimento de segurança e processo de negócio entre os *tenants* participantes da nuvem.

A Figura 3.2 apresenta a arquitetura SS4Cloud que é composto por dois elementos principais: ambiente de modelagem e o ambiente de execução. O SSC4Cloud Editor é parte do ambiente de modelagem e é utilizada para definição e compartilhamento de processo de negócio e modelos de segurança. Ele é executado localmente nos computadores dos usuários e conectado a um ambiente de execução (*Model Execution Environment*) que executa processos de negócios na nuvem em uma *workspace* privada. O editor é responsável por permitir aos analistas especificarem processos de negócios utilizando o padrão BPMN, fornecer aos especialistas de serviços a capacidade de associar serviços às tarefas dos processos de negócios e aos especialistas em segurança modelar aspectos de segurança no BPMN. Além da modelagem, o editor fornece aos usuários uma interface para acessar a infra-estrutura da nuvem por meio dos serviços disponibilizados por ela para gerenciamento de usuários, artefatos e execução de composição de serviços.

O segundo elemento principal da arquitetura SS4Cloud, o ambiente de execução, fornece o suporte para execução dos processos de negócio definidos no ambiente de modelagem. O *Authentication Service* e *User Account Management Service* são serviços utilizados pelo SSC4Cloud Editor para efetuar autenticação e autorização de usuários, respectivamente, no ambiente de nuvem. Além desses serviços, o ambiente da nuvem também possui cinco repositórios: artefatos, serviços, segurança, *tenant e workspace*.

O *Model Execution Environment* da arquitetura SSC4Cloud fornece todos os serviços referentes à execução do processo de negócio, tais como *Deploy*, *Start* e *Stop*. Com esses serviços os usuários poderão, além de criar suas composições, instalar, executar e parar as composições geradas pelo BPMN na nuvem.

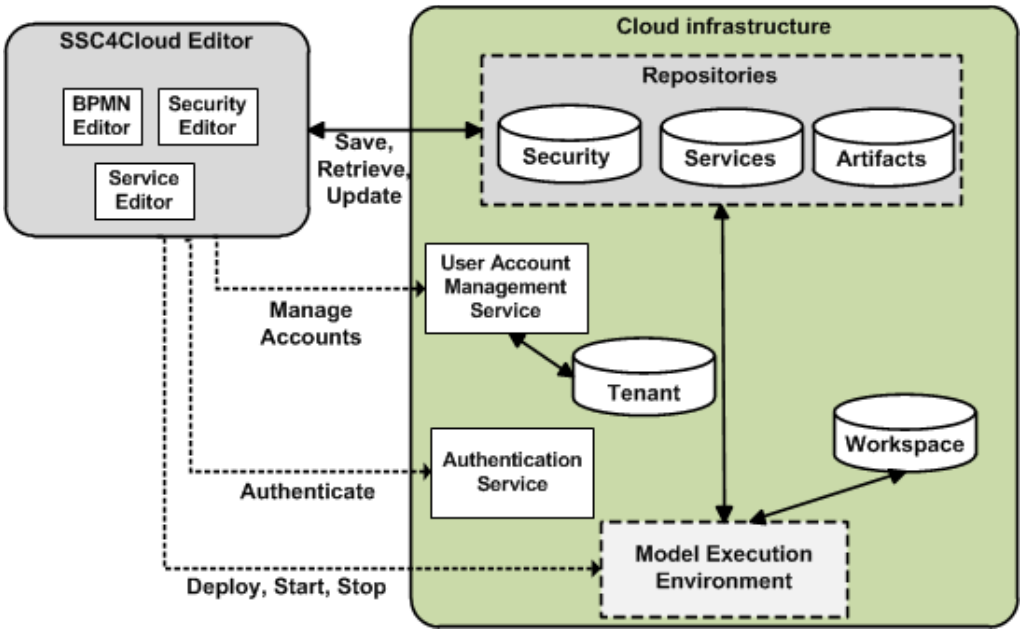


Figura 3.2 - Arquitetura SSC4Cloud

O repositório de artefatos tem como finalidade gravar na nuvem os modelos de BPMN gerados na fase de modelagem e os *profiles* de segurança criados pelos especialistas, que posteriormente poderão ser compartilhados entre os usuários da SSC4Cloud. O repositório de segurança mantém na nuvem as informações de segurança utilizadas na fase de modelagem e na criação dos *profiles* de segurança pelos especialistas. Nele é possível encontrar os *NF-Attributes*, *NF-Statements*, *NF-Actions* e *NF-Properties* suportados pelo ambiente de execução. Todas as informações necessárias para acessar os serviços utilizados pela *engine* de orquestrações, serviços esses disponibilizados pelos parceiros, estão localizadas no repositório de serviço. Além das WSDL dos serviços, este repositório também contém descrições textuais dos serviços, tipo do negócio, organizações provedoras dos serviços e outras informações que podem ser utilizadas para facilitar sua busca e utilização pelos usuários.

Por ter vários usuários acessando um mesmo ambiente, com recursos privados e com a possibilidade de compartilhamento, é necessário que as informações dos *tenants* sejam gravadas em um repositório, o repositório de *tenant*. Este repositório possui todas as informações dos usuários, tais como credenciais de acesso e informações pessoais e as políticas de acesso que dizem as quais recursos cada usuário pode acessar.

O *workspace* é um repositório onde são armazenados todos os artefatos gerados para a execução do processo de negócio (e.g., WSDL e WS-BPEL da composição) e os artefatos gerados na execução da composição (e.g., arquivos de *log* e informações geradas após a finalização e parada da composição). Quando uma composição em execução é pausada, seu estado deverá ser gravado no *workspace* para que a composição possa ser reiniciada posteriormente pela *engine* de orquestração exatamente de onde ela tinha parado.

3.3.2. Arquitetura SSC4Cloud Editor

Nesta seção será apresentada a arquitetura proposta para o SSC4Cloud Editor. A Figura 3.3 mostra esta arquitetura, conceitualmente dividida em seis módulos:

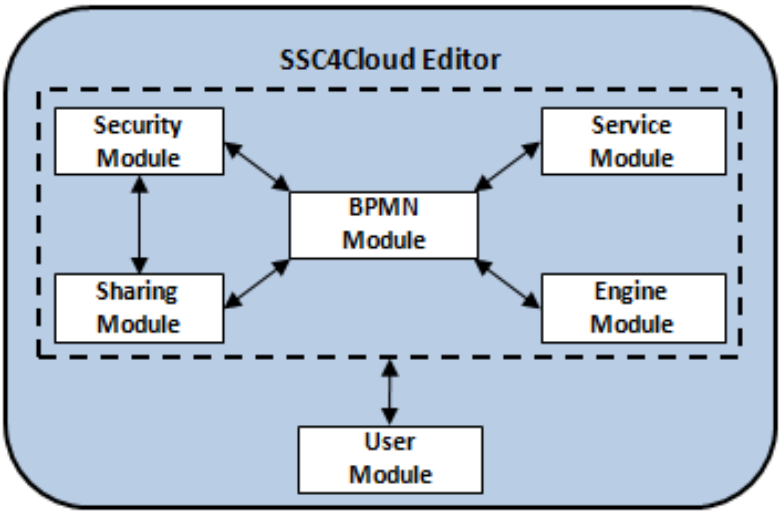


Figura 3.3 – Arquitetura SSC4Cloud Editor

- *BPMN Module*: Responsável por prover uma visão gráfica para o especialista de negócio modelar os seus processos de negócio em BPMN e adicionar anotações gráficas de forma rápida para representar os requisitos de segurança exigidos na composição **[RF02][RF03]**. Este módulo também é responsável por criar, recuperar e salvar projetos e processos de negócios na nuvem **[RF07]**;
- *Security Module*: Provê uma visão para o especialista de segurança anotar no BPMN os requisitos de segurança suportados pela Engine de execução com as devidas configurações **[RF04]**, como exemplo, algoritmo e chave de criptografia e possibilitar criação de *profiles* de segurança a fim de reutilizá-los em diferentes processos de negócios **[RF06]**;

- *Service Module*: Responsável em fornecer para o especialista de serviços e composição de serviço uma visão para configuração do processo de negócio de forma que ele possa ser executado na SSC4Cloud Engine **[RF05];**
- *Sharing Module*: Criado para fornecer aos usuários da ferramenta a possibilidade de compartilhar conhecimento através de compartilhamento de processos de negócios com anotações de segurança **[RF08]** e *profiles* de segurança desenvolvidos por especialistas de segurança **[RF09]**;
- *Engine Module*: Ele fornece a possibilidade dos usuários gerenciarem os processos de negócios na *engine* de execução **[RF13] [RF14] [RF15]**;
- *User Module*: Criado para gerenciar **[RF10]** e autenticar **[RF11]** os usuários na nuvem. Na SSC4Cloud vários usuários estarão compartilhando o mesmo ambiente e desta forma será necessário restringir o acesso dos usuários na nuvem, que provê uma separação lógica dos artefatos por usuários.

O *BPMN Module* usa o *User Module* para autenticar os usuários na SSC4Cloud e criar, recuperar e salvar projetos e processos de negócios na nuvem. Com o processo criado, o *Service Module* utiliza o *BPMN Module* para obter todos os elementos que serão configurados pelo analista na visão de serviço. Além disso, toda modificação feita na visão de serviço será também anotada no BPMN. O *Security Module* também utiliza o *BPMN Module* para obter os elementos que serão utilizados para anotar a segurança da composição e, assim como o *Service Module*, todas as modificações feitas em sua visão também serão refletidas no diagrama BPMN.

Como já foi mencionado, é possível baixar da nuvem diagramas criados pelo usuário através do *BPMN Module*. Porém, também é possível baixar e utilizar BPMN's criados por outros usuários e compartilhados através do *Sharing Module*. Para compartilhar artefatos, o *Sharing Module* utiliza o *BPMN Module* e o *Security Module* para obter os processos de negócio e os *profiles* de seguranças criados pelos respectivos módulos. Após um processo de negócio ser configurado adequadamente, ele poderá ser instalado e executado na nuvem através do *Engine Module* que utilizar o *BPMN Module* para obter os artefatos referentes ao processo de negócio antes de efetuar a devida operação. Todos os módulos da arquitetura SSC4Cloud Editor utilizam o *User Module* para obter o *token* do usuário autenticado no SSC4Cloud para posteriormente efetuar as devidas interações com os serviços da nuvem.

3.4 Projeto

Esta seção apresenta os detalhes do projeto da arquitetura proposta para o SSC4Cloud Editor. Esta ferramenta foi projetada como uma extensão do BPMN *Modeler* (ECLIPSE, 2009), um *plugin* do Eclipse desenvolvido para prover modelagem de processo de negócio em BPMN **[RF01]**.

Visto que a *engine* de orquestração juntamente com a base de dados da SSC4Cloud encontram-se na nuvem, toda a comunicação, tanto com o repositório quanto com a *engine*, deverá acontecer por meio de requisições de serviços.

Com base na arquitetura proposta, o projeto deste trabalho foi desenvolvido em pacotes cujos relacionamentos podem ser vistos na Figura 3.4. As seções a seguir detalharão todos os pacotes do projeto SSC4Cloud Editor.

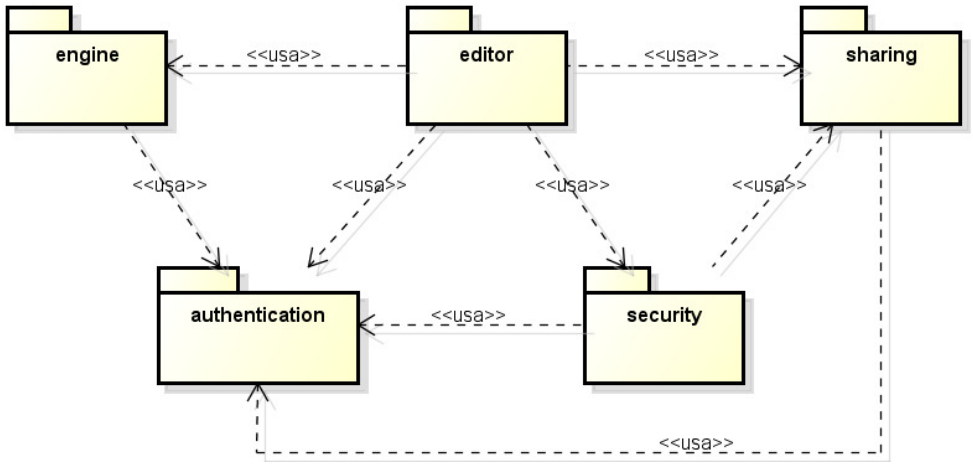


Figura 3.4 - Relação Entre os Pacotes do Projeto SSC4Cloud Editor

3.4.1. Editor

O SSC4Cloud Editor possui três visões, onde os especialistas de negócio, segurança e serviços podem modelar e configurar a composição com requisitos de segurança de forma independente. A classe *MoSCEditor* (Figura 3.5) é responsável por criar as visões de negócio, segurança e serviço, representadas pelas classes *BpmnDiagramEditor*, *QualityRequirements-Composite* e *ServiceViewComposite*, respectivamente (Figura 3.5). Além disto, ela deve garantir que todas as modificações salvas no editor sejam sincronizadas com o repositório de

artefatos localizado na nuvem, por meio do método *saveMoSCFile()*, chamado todas as vezes que um diagrama é salvo.

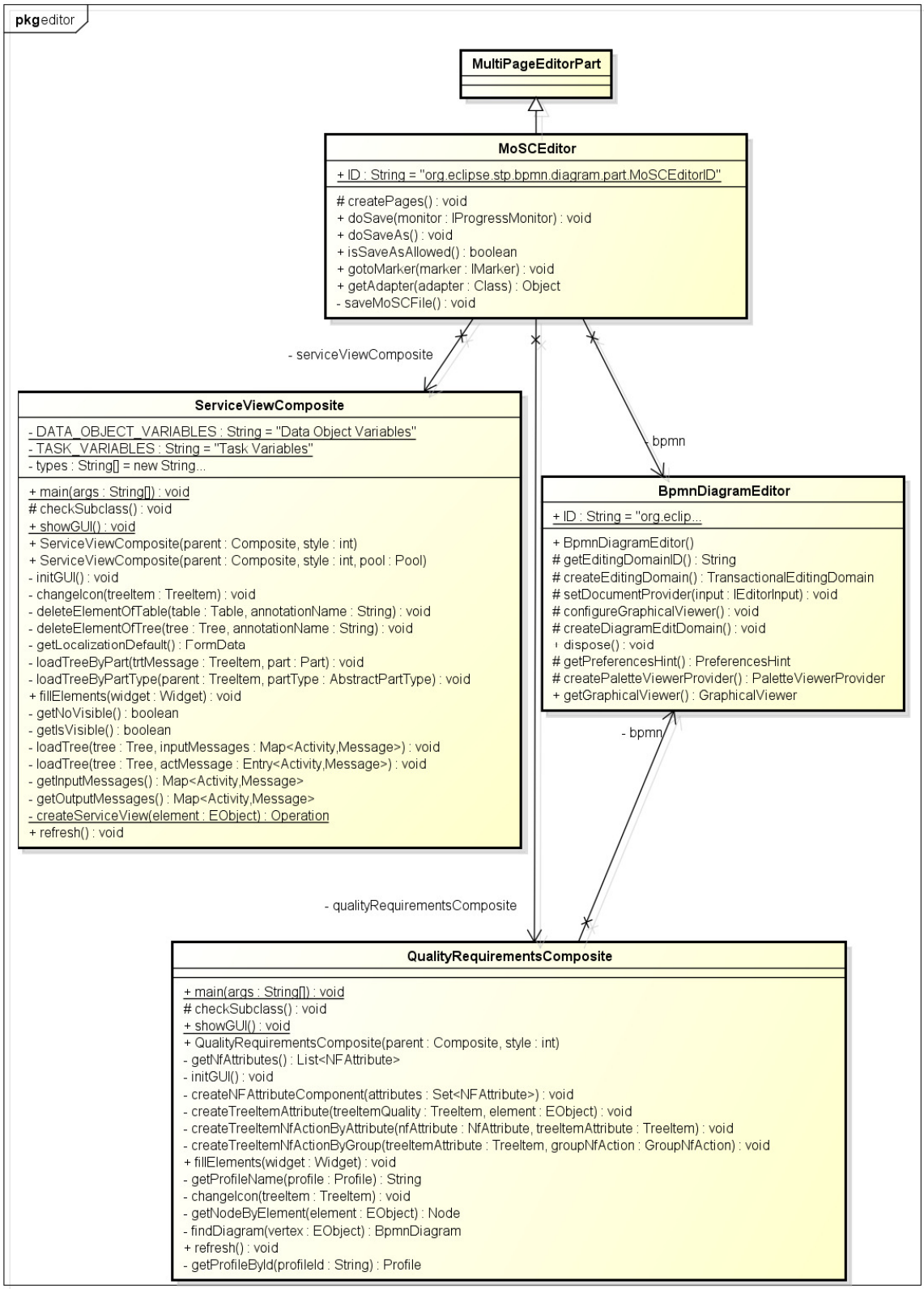


Figura 3.5 - Classes Responsáveis Pela Criação das Visões de BPMN, Segurança e Serviço

O *BpmnDiagramEditor* representa a visão para o analista de processo de negócio modelar composições utilizando BPMN com anotações gráficas de requisitos de segurança exigida no processo modelado. Nesta visão existe uma palheta, criada pelo método *createPalleteViewerProvider()*, com todos os elementos gráficos, tanto de BPMN quanto de requisitos de segurança, tal como *NF-Actions*, grupos de *NF-Actions* e *NF-Attribute*.

Utilizando a visão de segurança, representada pela classe *QualityRequirements-Composite,* o especialista de segurança poderá configurar os requisitos de segurança exigidos na composição. Além de configurar, esta visão permite adicionar e remover requisitos de segurança no BPMN utilizando uma referência ao *BpmnDiagramEditor* para manter a visão de negócio atualizada conforme suas modificações. Desta forma, qualquer modificação efetuada na visão de segurança será refletida na visão de negócio.

Além do processo de negócio e dos requisitos de segurança, a composição precisa ser configurada para invocar os serviços e operações de forma correta. Todas as configurações provenientes dos serviços serão feitas na visão de serviço (classe *ServiceViewComposite*). Esta visão permite ao especialista de serviço associar os elementos de atividade do BPMN às requisições de *Web Services* adequadas, fazer os *assign* das variáveis, representados pelo elemento *Data Objets* do BPMN e configurar o fluxo do processo, controlado pelos *Gateways*, para o processo de negócio ser executada corretamente na *engine* de orquestração conforme especificado na visão de BPMN.

3.4.2. Authentication

Para garantir que apenas usuários cadastrados e com contas válidas usem os serviços providos pela SSC4Cloud, todas as requisições entre o editor e a nuvem deverão ser autenticadas e a ferramenta deve prover meios para isso acontecer.

A autenticação do usuário na nuvem será feita por *login* e senha obtidos previamente e informados nas preferências do SSC4Cloud Editor. Sempre que o usuário acessar a ferramenta suas informações serão passadas para o serviço de autenticação na nuvem (método *authenticate()* da classe *AuthenticationServiceImpl*) (Figura 3.6) para serem validados. Caso os dados estejam corretos, o serviço retornará um *token* (Figura 3.6), relacionado ao *tenant* (cliente) que será usado em todas as operações realizadas entre o editor e os serviços da

nuvem. Quando o serviço de autenticação gera um *token* para o usuário, é mantido na nuvem outro *token* representado pela classe *LocalTenant* contendo o último acesso do usuário para controlar o tempo de inatividade da ferramenta e com isso forçar o usuário a obter um novo token para continuar utilizando os serviços da nuvem.

O *token* deve ter uma identificação única para representar a instância da ferramenta no módulo de autenticação da nuvem. Esta identificação é representada pela classe *TokenID* que tem o atributo *tokenContent*, utilizado na identificação de cada *token*.

Uma organização pode conter vários setores e departamentos onde cada um poderá utilizar a ferramenta por meio de diferentes *tenants*. Para garantir esta hierarquia, um *tenant*, representado pela classe *Tenant*, pode possuir outros *tenants* como participantes (*participants*) e esses participantes terão como atributo o *tenant* ao qual eles pertencem (*parent*).

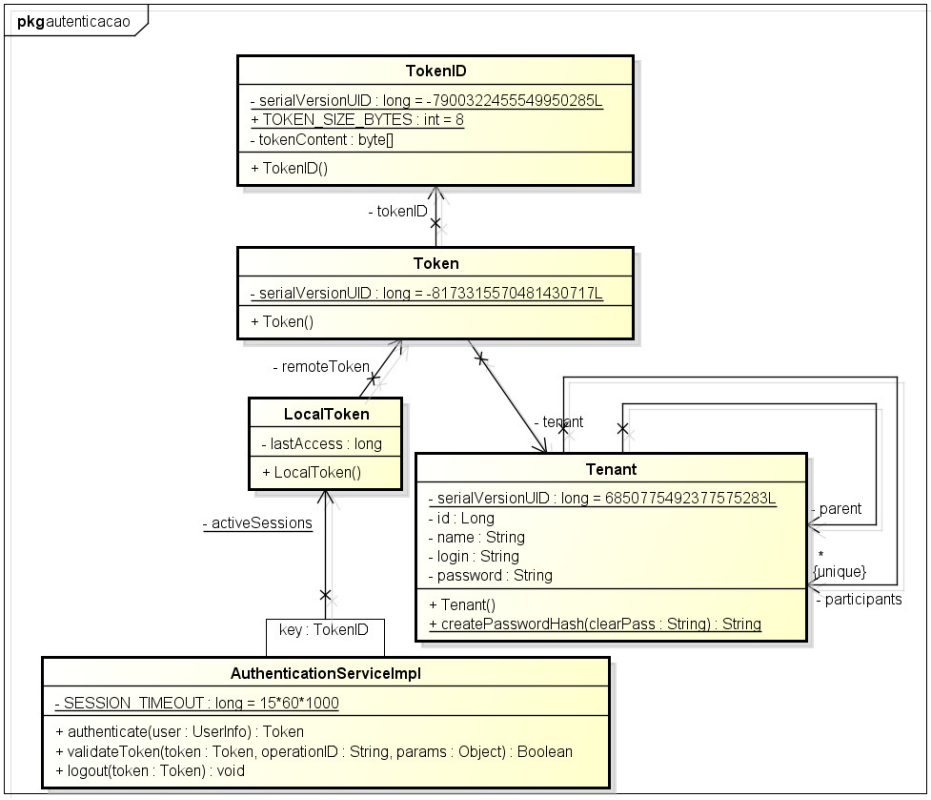


Figura 3.6 – Classes do Pacote de Autenticação

3.4.3. Security

Este pacote (ver Figura 3.7) contém todas as classes criadas para os analistas de segurança criarem diferentes configurações de segurança (*profiles*), que possuem nome, um

NF-Attribute primário, no caso Segurança e uma coleção de *NFDefaultConfiguration*, passíveis de reutilização em diferentes diagramas de processos de negócio e até compartilhadas com diferentes *tenants* (usuários). Um *NF-Attribute* pode está relacionado a um *NF-Attribute* primário, representado na classe pelo atributo *parent*.

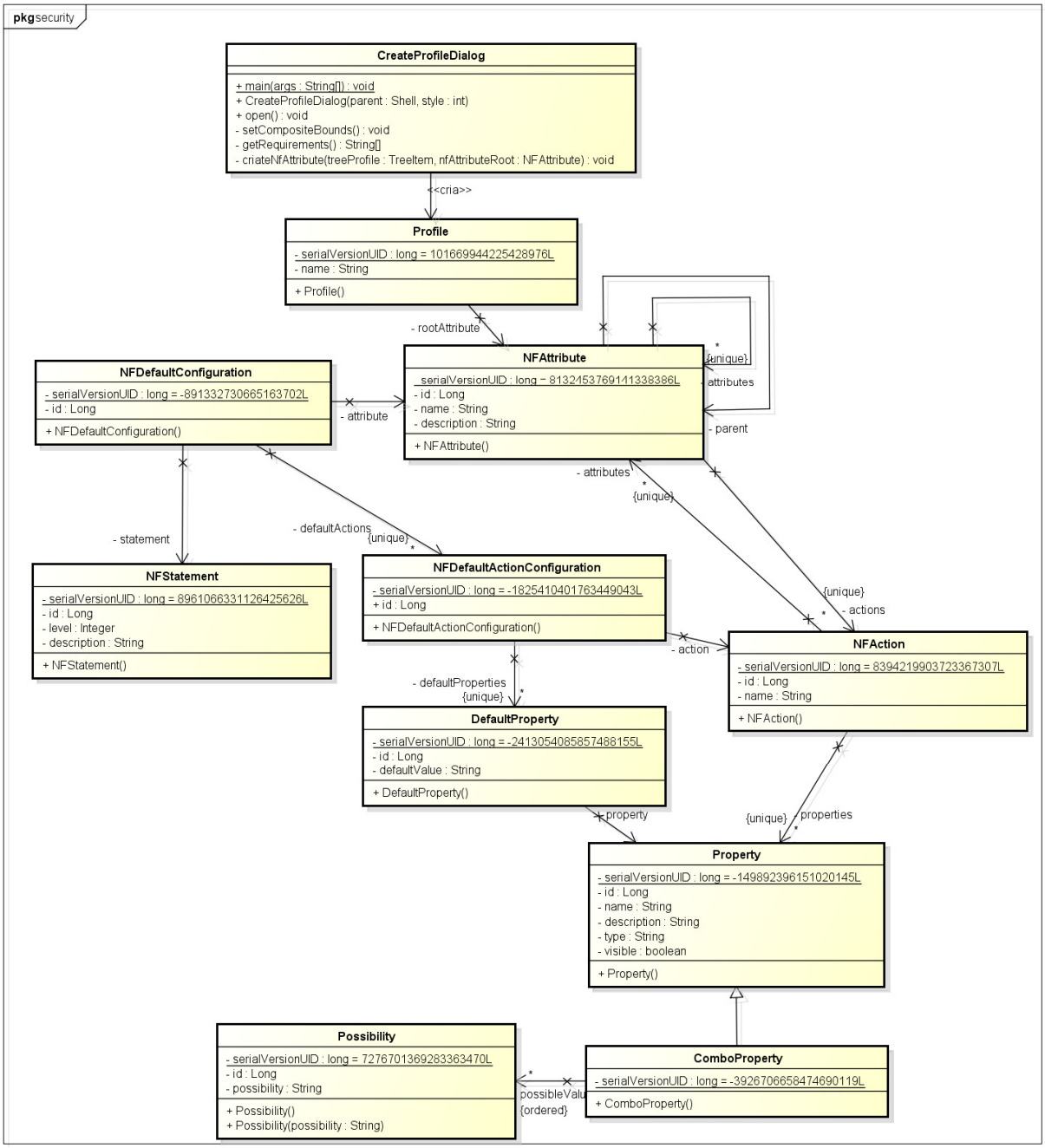


Figura 3.7 - Diagrama de Classe do Pacote *Security*

NFDefaultConfiguration é uma classe composta por um *NF-Attribute* relacionado à segurança (e.g., Confidencialidade) e é associada a um *NF-Statement*, que representa o nível desta Confidencialidade (e.g., “alta confidencialidade”) desejado.

Uma propriedade (*Property*) possui como atributos seu nome, uma descrição sobre ela, seu tipo (*String*, *Boolean*, *Long*, *Map*, entre outros) e se ela é visível ou não para o usuário. Uma propriedade pode ter mais de uma opção de escolha pré-determinada e para representar estas opções foi criado a classe *ComboProperty* (Figura 3.7), filha da classe *Property*, que possui como atributo uma coleção de *Possibility* que representa os possíveis valores a serem escolhidos na modelagem, representado no editor como um elemento do tipo *Combobox*.

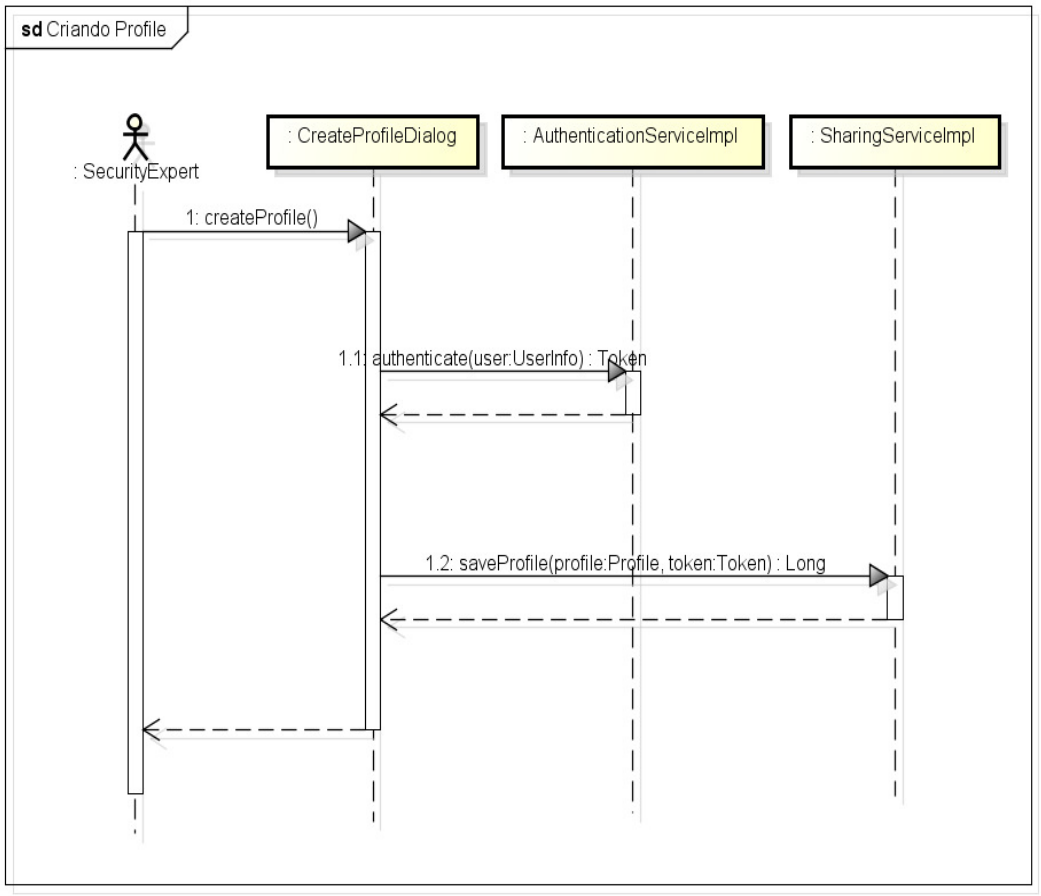


Figura 3.8 - Diagrama de Seqüência para Criação de *Profile* de Segurança

Todas as chamadas de serviços deverão passar como argumento o *token* obtido na autenticação do usuário na nuvem. A Figura 3.8 mostra o diagrama de seqüência da operação de criação de *profile* pelo especialista de segurança, onde a primeira operação (1.1) deverá ser a autenticação do usuário na SSC4Cloud passando como argumento suas informações (*login* e senha) e posteriormente a invocação da operação de criação do *profile* (1.2) que possui como argumentos o *profile* a ser criado e o *token* do usuário obtido na autenticação.

3.4.4. Sharing

Todas as entidades compartilhadas entre usuários deverão estender da classe *Shareable* (Figura 3.9), que tem como atributos o dono da entidade e a lista de usuários com acesso a ela.

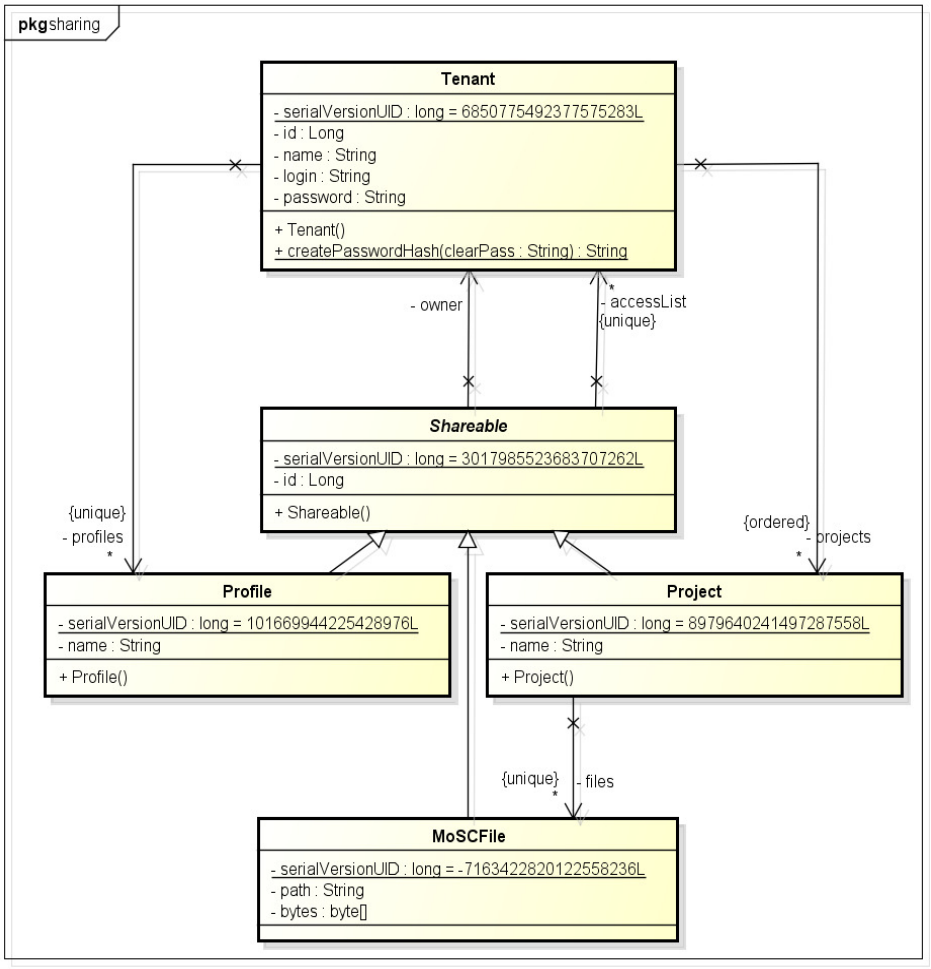


Figura 3.9 - Diagrama de Classe das Entidades que Podem ser Compartilhadas

Uma das propostas deste trabalho é prover aos analistas de negócio e segurança a capacidade de compartilhar projetos de processos de negócios, assim como os processos de negócios neles contidos, e *profiles* de segurança a fim de fornecer a reusabilidade e o compartilhamento de conhecimento entre os usuários da SSC4Cloud.

Um projeto é uma representação lógica usada para classificar os processos de negócio dos usuários, análogo a um diretório de arquivos de um sistema operacional. Todo processo de negócio deverá estar associado a um projeto do usuário e um projeto poderá

conter vários processos de negócio. Todos os processos de negócio são representados pela classe *MoSCFile* (Figura 3.9) possuindo como atributos o *path* do arquivo e seus *bytes* que serão utilizados para remontar localmente o processo de negócio quando necessário.

3.4.5. Engine

Após a composição ser criada utilizando o BPMN e com os serviços e requisitos de segurança anotados adequadamente, ela poderá ser executada na *engine* de orquestração localizada na nuvem. O *Engine Module* é responsável por instalar, iniciar, reiniciar e parar uma composição utilizando chamadas de serviço por meio da classe *ManagementService* (Figura 3.10).

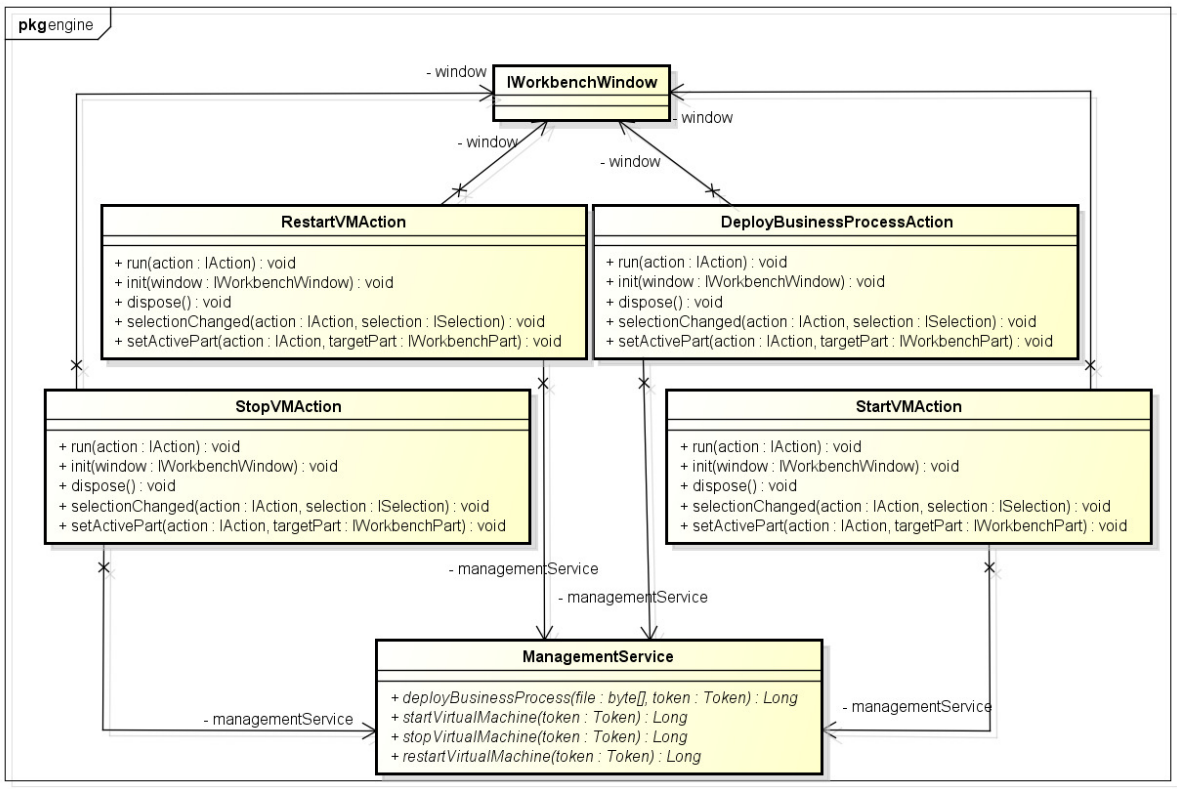


Figura 3.10 – Diagrama de Classe do Pacote *Engine*

As classes na Figura 3.10 representam essas funcionalidades do editor, onde a classe *DeployBusinessProcessAction* instala uma processo de negócio, a classe *StartVMAction* inicia um processo de negócio, a *RestartVMAction* reinicia e a *StopVMAction* pára um processo de negócio na nuvem através de invocações de serviços realizados pela classe *ManagementService*.

3.5 Implementação

Nesta seção serão vistos os detalhes de implementação do projeto proposto apresentado na seção anterior.

O SSC4Cloud Editor possui os módulos *BPMN*, *Security*, *Service*, *Sharing*, *Engine* e *User* e a implementação desses módulos foi feita utilizando as seguintes tecnologias: linguagem *Java 2 Standard Edition* 5.0, Eclipse 3.5 e Spring Framework 2.5. O editor foi desenvolvido como *plug-in* para a plataforma Eclipse (ECLIPSE, b).

3.5.1. User Module

Para garantir o isolamento dos clientes, todo usuário do editor precisa ser autenticado na SSC4Cloud antes de consumir seus serviços. Para isso acontecer, o editor precisa prover um meio para os usuários informarem seus dados, necessários para autenticação e localização dos serviços. Com esta finalidade, foi criada uma nova preferência (Figura 3.11) no SSC4Cloud Editor onde o usuário informa seu *login* e senha para serem utilizados na sua autenticação e o *host* onde os serviços remotos da nuvem estão localizados.

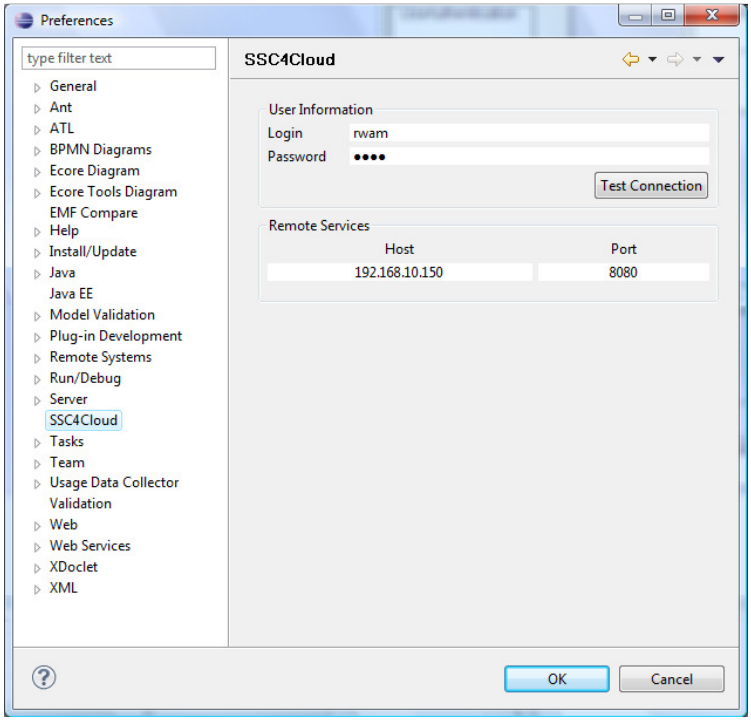


Figura 3.11 - Tela de Preferências do SSC4Cloud Editor

A Listagem 3.1 mostra a classe *PreferencesUtil,* que implementa os métodos para a interação do editor aos serviços da nuvem. O método `getConnection` (14- 19) devolve uma instância da classe *Connection*, responsável por prover todos os serviços disponível na nuvem e utilizada em todas invocações do editor à SSC4Cloud. Caso já exista uma instância de *Connection*, ela será devolvida para o solicitante. Caso esta instância não exista, uma nova conexão será criada pelo método `newConnection` (20-31), que chama o método createPropertiesConnection (Linha 29), da classe *ConnectionFactory* (fábrica de *Connection* que utiliza o *Spring Framework* para criar o objeto de conexão e encapsular todas as chamadas de serviços remotos) para criá-la.

Listagem 3.1 – Classe *PreferencesUtil*

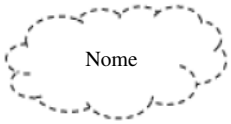
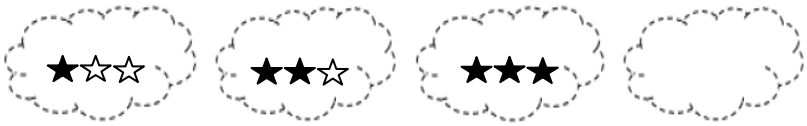
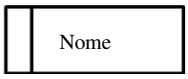
```
1   public class PreferencesUtil {
2    ...
3        public static synchronized Token getToken() {
4             if (localToken == null) {
5                  final AuthenticationService authenticationService
    = PreferencesUtil.getConnection().getAuthenticationService();
6                  try {
7                      localToken =
                       authenticationService.authenticate(
                       PreferencesUtil.getUserInfoFromPreferences());
8                  } catch (AuthenticationException exp) {
9                      exp.printStackTrace();
10                 }
11            }
12            return localToken;
13       }
14       public static Connection getConnection() {
15            if (connection == null) {
16                 newConnection();
17            }
18            return connection;
19       }
20       public static Connection newConnection() {
21            localToken = null;
22            connection = null;
23            IPreferenceStore preferenceStore =
                Activator.getDefault().getPreferenceStore();
24            final String host = preferenceStore.getString(
                PreferenceConstants.REMOTE_SERVICE_HOST);
25            final int port = preferenceStore.getInt(
                PreferenceConstants.REMOTE_SERVICE_PORT);
26            Properties props = new Properties();
27            props.put(ConnectionFactory.HOST, host);
28            props.put(ConnectionFactory.PORT, port);
29            connection = ConnectionFactory.getInstance()
                .createPropertiesConnection(props);
30            return connection;
31       }
40  }
```

Além de fornecer uma instância da classe *Connection*, encapsulado pelo *Spring Framework* para invocação remota de serviço, a classe *PreferencesUtil* possui o método `getToken` (11-21), que devolve um *token* para autenticação do usuário na nuvem. Este método verifica se já existe algum *token* instanciado e, caso contrário, obtém uma instância de *Connection* e chama a operação de autenticação do *authenticationService* (Linha 15), passando o *login* e *password* do usuário obtidos da preferência do Eclipse.

3.5.2. BPMN Module

O módulo de *BPMN* do *SSC4Cloud Editor* (ver Figura 3.3) permite que o analista de negócio crie processos utilizando BPMN e insira anotações segurança em suas modelagens. A Tabela 3.1 apresenta a representação gráfica dos elementos de segurança.

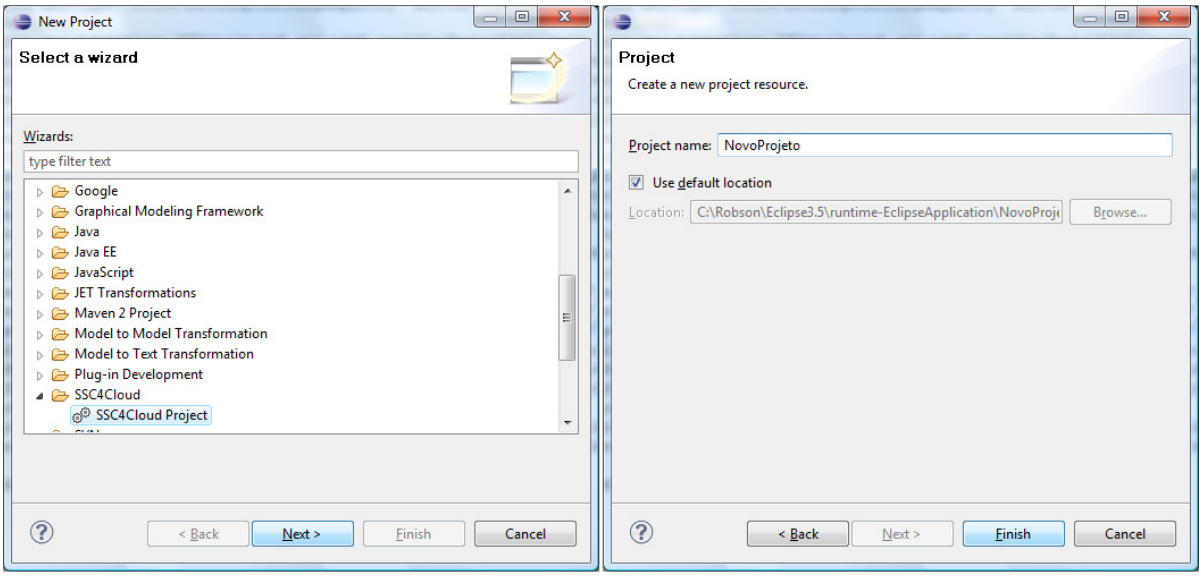
Tabela 3.1 - Principais Abstrações de Segurança

Abstração	Notação
NF-Attribute	
NF-Statement	
NF-Action	

O *NF-Attribute* é representado graficamente no editor com a forma de uma nuvem tracejada e com o nome do requisito de segurança que ele está representando. Para informar o nível da segurança, os *NF-Statements* são representados por três estrelas localizadas no centro dos *NF-Attributes* que, dependendo do seu preenchimento, irá representar uma segurança baixa (apenas a primeira estrela preenchida), segurança média (com as duas primeiras estrelas preenchidas) e segurança alta (todas as estrelas preenchidas). Uma *NF-Statement* é

considerada customizada quando ela é configurada em tempo de modelagem e neste caso ela não tem representação gráfica. No editor, as *NF-Actions* são representadas graficamente por um retângulo com o nome no centro.

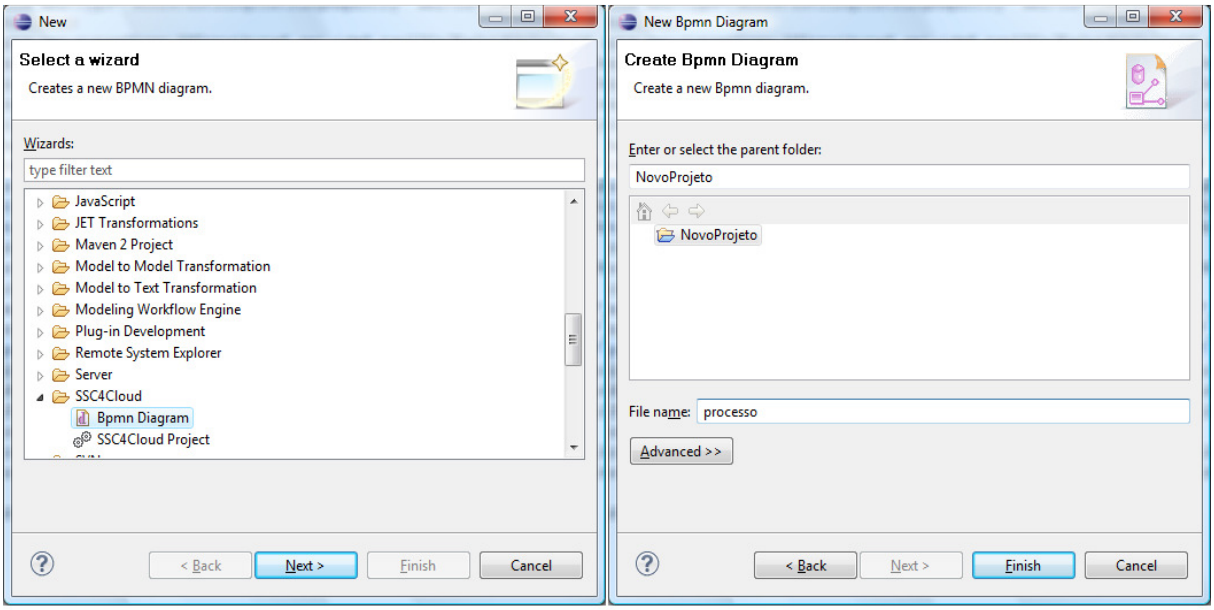
Todo processo de negócio deverá estar associado a um projeto do tipo SSC4Cloud. Para criar esses projetos foi desenvolvido na ferramenta um *wizard* (Figura 3.12) que, de forma simples e transparente para o usuário, cria o projeto localmente e remotamente na nuvem. Para criar o projeto local, o *wizard* utiliza o nome do projeto fornecido pelo usuário e o local ao qual ele deseja que o projeto seja salvo. Porém, na nuvem é utilizado apenas o nome do projeto que será associado ao *tenant* do usuário ao qual ele pertence.



a) Tela para selecionar o tipo do projeto **b) Tela para informar o nome do projeto**

Figura 3.12 - *Wizard* para Criar um Novo Projeto SSC4Cloud

Da mesma forma que existe um *wizard* para criar um novo projeto do tipo SSC4Cloud, também existe no editor um para criar diagramas BPMN. Todo diagrama é associado a um projeto, previamente criado, e é representado por dois artefatos: um responsável pelas informações gráficas do diagrama (e.g., posições dos elementos gráficos, tamanho e cores das fontes dos textos), e outro que contem todos os detalhes de associação e anotações do diagrama.



a) Tela para selecionar o diagrama de BPMN

b) Tela para informar o nome do processo

Figura 3.13 - *Wizard* para Criação de Novos Diagramas BPMN

O *BPMN Module* dispõe de uma visão (Figura 3.14) responsável por fornecer aos analistas de negócio recursos para modelagem de BPMN com as devidas anotações gráficas de segurança. Para utilizar as abstrações de segurança, os analistas utilizam os elementos gráficos localizados na paleta . Como exemplo, a Figura 3.14 mostra o SSC4Cloud Editor na visão de negócio com um simples diagrama BPMN para aluguel de veículo contendo quatro atividades BPMN (“*Reservar Carro*”, “*Enviar SMS de Indisponibilidade*”, “*Processar Pagamento*” e “*Enviar SMS de Confirmação*”), onde a atividade “*Reservar Carro*” tem como requisito não-funcional a confidencialidade que será implementada pela ação de autenticação e a atividade “*Processar Pagamento*” também com requisito de confidencialidade. Nesta primeira versão do diagrama o analista de negócio especifica os requisitos funcionais e não-funcionais do processo sem nenhuma configuração específica, tanto dos serviços quanto da segurança, deixando a composição ainda não executável na *engine* de orquestração da SSC4Cloud.

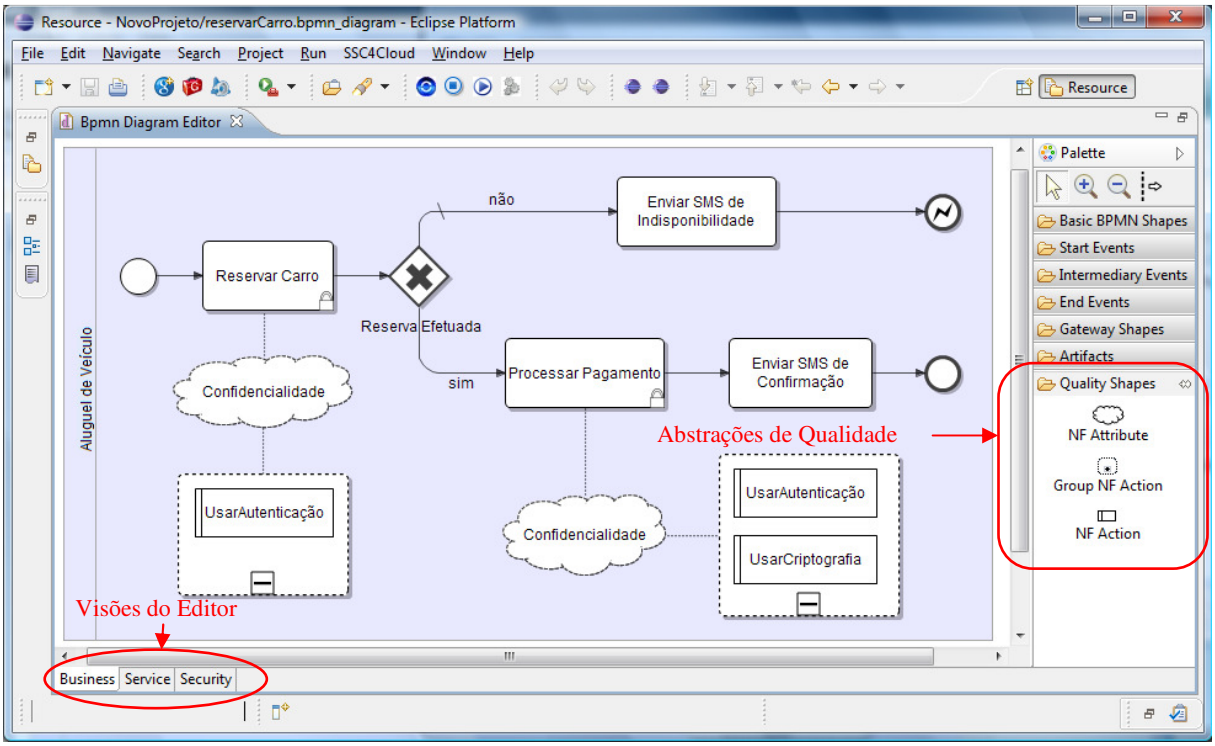


Figura 3.14 - BPMN Editor: Visão de Negócio

Depois da criação do diagrama, todas as modificações feitas nele deverão refletir na nuvem para manter a sincronização entre os dois ambientes (local e remoto). Para isso, o método `saveMoSCFile` (Listagem 3.2) é invocado sempre que um processo de negócio é salvo pelo usuário. Este método obtém os dois arquivos referentes ao BPMN que estão sendo salvos (2-5) e chama a operação `saveFile` do serviço SharingService da nuvem, encapsulado pelo objeto *connection* obtido através do método `getConnection()` da classe PreferencesUtil (Linha 20).

Listagem 3.2 – Método para Salvar na *Cloud* os Arquivos do Processo de Negócio

```
1   private void saveMoSCFile() throws IOException, CoreException,
    AuthenticationException {
2       final IFile diagram_file = ((FileEditorInput)
          this.getEditorInput()).getFile();
3       final String fileName = diagram_file.getName().
          replace(diagram_file.getFileExtension(), "bpmn");
4       final IFile bpmn_file =
          diagram_file.getWorkspace().getRoot().
          getFile(diagram_file.getParent().
          getFullPath().append(fileName));
5       final IFile[] files = new IFile[]{diagram_file,
           bpmn_file};
6       for (IFile iFile : files) {
7           MoSCFile moscFile = null;
8           String path = iFile.getFullPath().toString();
```

```
9            if (LocalFileCache.getInstance().containsFile(path)) {
10               moscFile =
                   LocalFileCache.getInstance().getFile(path);
11           } else {
12               moscFile = new MoSCFile();
13               moscFile.setPath(path);
14               LocalFileCache.getInstance().putFile(moscFile);
15           }
16           final InputStream contents = iFile.getContents();
17           byte[] contentBytes=new byte[contents.available()];
18           contents.read(contentBytes);
19           moscFile.setBytes(contentBytes);
20           Long idFile =
               PreferencesUtil.getConnection().getSharingService().
               saveFile(moscFile, PreferencesUtil.getToken());
21            moscFile.setId(idFile);
22       }
23   }
```

Com os diagramas também disponíveis na nuvem, usuários poderão removê-los do seu computador local e posteriormente recuperá-los através da funcionalidade de importação de projetos do editor (ver Figura 3.15). Nela é possível importar qualquer projeto criado pelo próprio *tenant* ou projetos criados por outros usuários, mas compartilhados com ele através do módulo de compartilhamento.

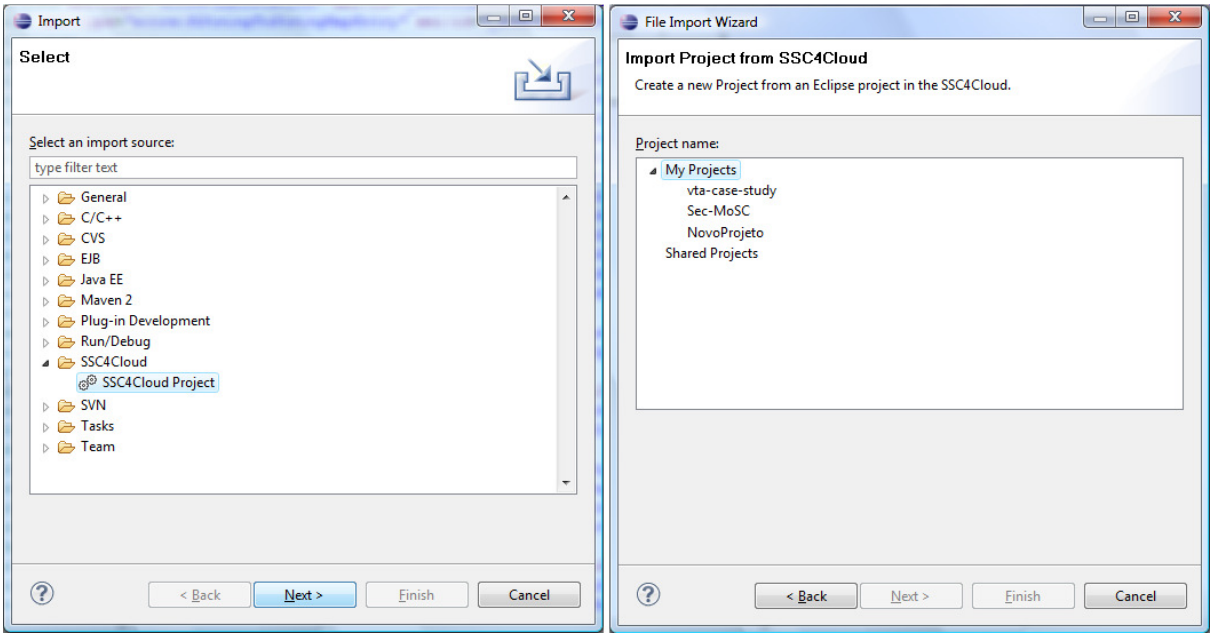


Figura 3.15 – *Wizard* para Importar Projetos da Nuvem

Os módulos detalhados a seguir são responsáveis por deixar os diagramas modelados capazes de serem executados em uma *engine* de orquestração e compartilhados entre os usuários da SSC4Cloud.

3.5.3. Security Module

Como foi dito na seção anterior, especialistas em processos de negócios podem inserir anotações de segurança em suas modelagens utilizando os elementos gráficos das abstrações de segurança localizados na paleta do editor (Figura 3.14). Porém, essas anotações inicialmente não contêm informações suficientes para serem executadas na SSC4Cloud por serem apenas anotações gráficas que precisarão ser configuradas na visão de segurança do editor por um especialista de segurança.

Para garantir que apenas *NF-Actions* suportadas pela SSC4Cloud sejam utilizadas pelos especialistas de segurança e de forma personalizada, foi criado no SSC4Cloud Editor o *Security Module* para possibilitar a criação de *profiles* de segurança, utilizando uma lista de *NF-Actions* pré-cadastradas e disponíveis no repositório de segurança da nuvem, que poderão ser utilizados pelos especialistas na visão de segurança do editor (Figura 3.16).

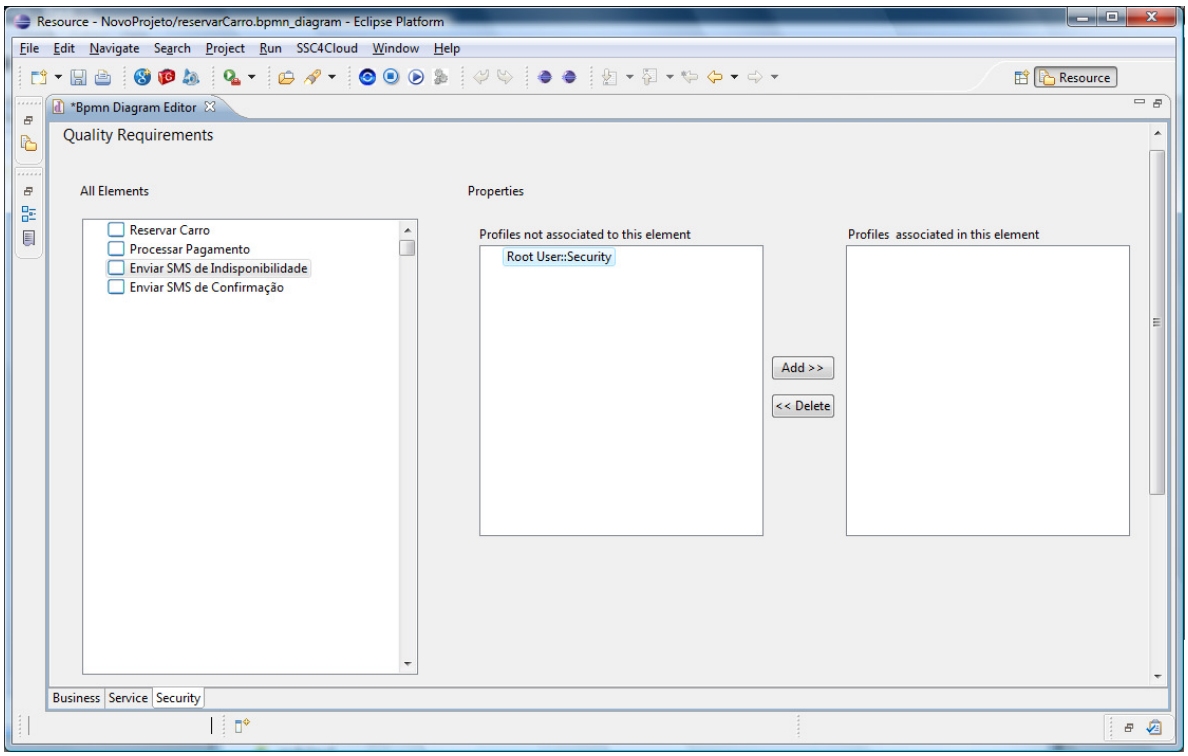


Figura 3.16 - Tela da Visão de Segurança

Um *profile* consiste de uma descrição e do requisito não-funcional (*NF-Requirement*) suportado pela ferramenta. No repositório, o requisito *Security* está associado aos seus *NF-Attributes* (*Audit, Authentication, Authorization, Availability, Business Security Requirement, Confidentiality, Integrity e Non Repudiation of Origin*) e com isso eles automaticamente farão parte de todos os *profiles* criados para este requisito.

A Figura 3.17 mostra um *profile* denominado *Security* criado no SSC4Cloud Editor. Por ser um *profile* de segurança, ele já é associado, na sua criação, aos *NF-Attributes Audit*, *Authentication*, *Authorization*, *Availability*, *Business Security Requirement*, *Confidentiality*, *Integrity* e *Non Repudiation of Origin*.

A personalização do *profile* se dá com a criação das *NF-Statements*, onde um *NF-Attribute* pode conter mais de um *NF-Statement*. Porém, o que diferencia um *NF-Statement* de outro é o conjunto de *NF-Actions* associado a elas com suas respectivas propriedades configuradas. A Figura 3.18 ilustra o atributo *Confidentiality* com três *NF-Statement* (*Medium Statement*, *High Statement* e *Low Statement*), inicialmente sem nenhuma *NF-Action* associada a elas. Ao criar um *NF-Statement* é informado o nível de segurança ao qual ela será interpretado pelo usuário.

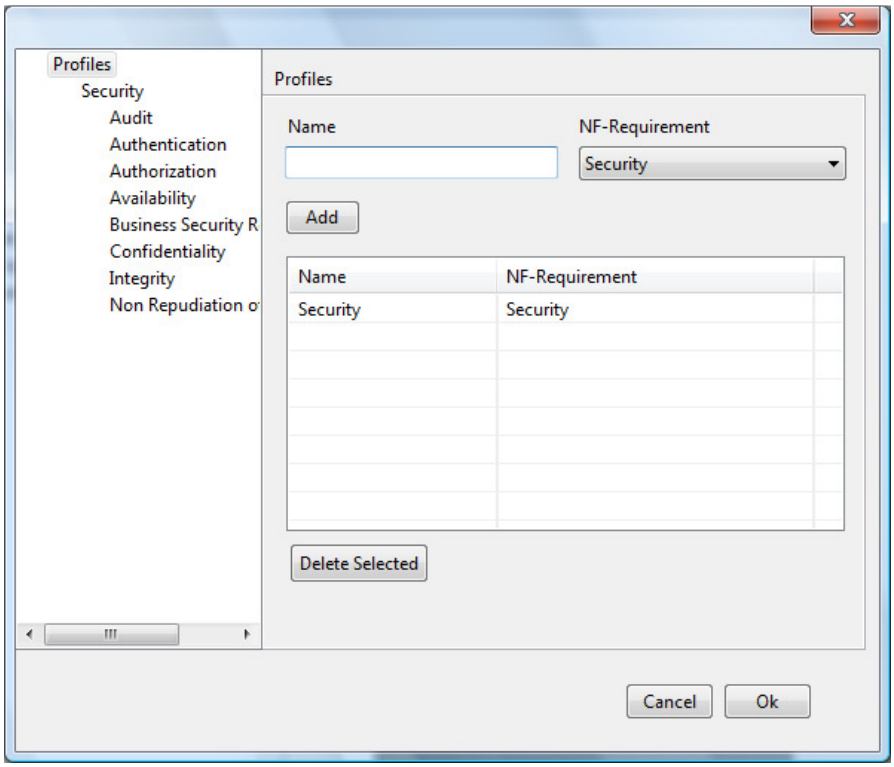


Figura 3.17 - Tela de Criação de *Profiles*

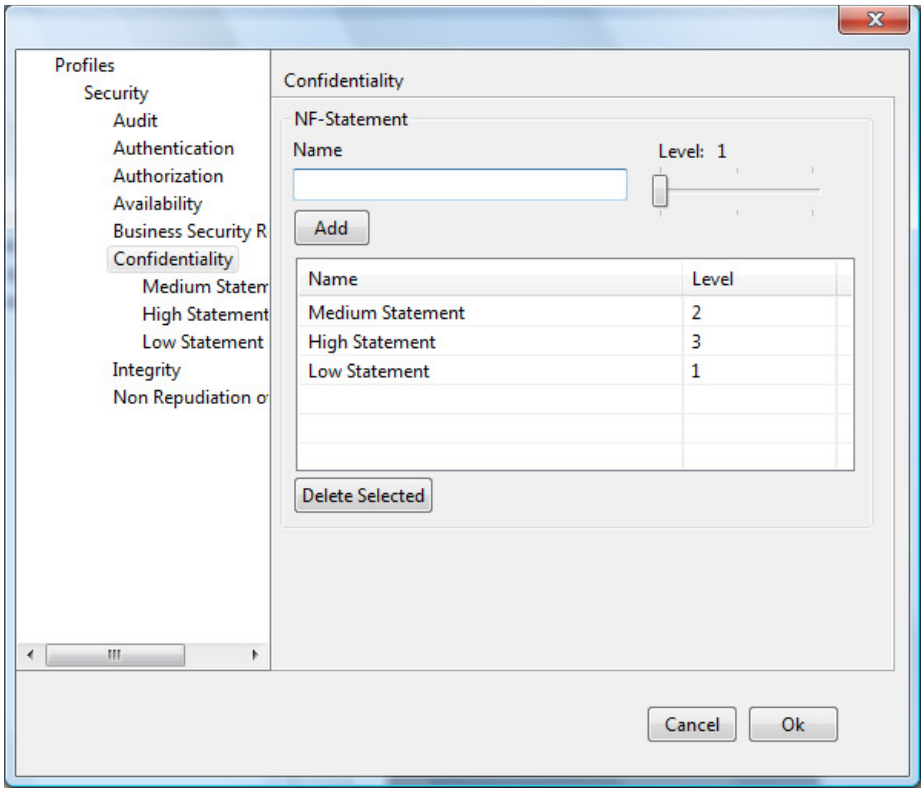


Figura 3.18 - *NF-Statements* de *Confidentiality*

Com a utilização dos níveis em forma de números (ver Tabela 3.2), é possível identificar o grau de segurança que a *NF-Statement* representa para o usuário, onde o nível 1 (*L1*) representa uma segurança mais baixa, o nível 2 (*L2*) uma segurança média e nível 3 (*L3*) uma segurança alta. Desta forma, o usuário poderá atribuir qualquer nome para representar seus *NF-Statements* e informar o seu nível de segurança através do atributo *level*.

Tabela 3.2 - Classificação dos Níveis

Nível	Descrição	Notação gráfica
1	Baixa segurança	★☆☆
2	Segurança média	★★☆
3	Alta segurança	★★★

Cada *NF-Attribute* possui com conjunto de *NF-Actions* cadastradas no repositório da nuvem. O *NF-Attribute Confidentiality* possui seis *NF-Actions* (*DeleteInformation*, *ClassifyInformation*, *UseAccessControl*, *RestrictAccess*, *UseAuthentication*, *UseCriptography*) que podem ser associadas aos *NF-Statements*. A Figura 3.19 mostra as NF-Actions *UseAuthentication* e *UseCriptography* associadas ao *NF-Statement "Medium Statement"* do *NF-Attribute* "*Confidentiality"*. Também pode ser visto nesta figura que cada *NF-Statement* de *Confidentiality* tem uma lista de *NF-Actions* diferente, conforme configurado pelo especialista.

Cada *NF-Action* cadastrada no repositório da nuvem, obtida pelo editor, possui uma lista de propriedades, que podem ser do tipo *Property* ou *ComboProperty* (Figura 3.7). Quando a propriedade é do tipo *Property* o editor cria um campo de texto para o usuário inserir um valor e quando é do tipo *ComboProperty* é criado no editor um campo do tipo *combo* onde o usuário poderá selecionar uma das opções pré-cadastradas no repositório. Esses valores informados nas propriedades no cadastro dos *profiles* serão utilizados como valor *default* para quando forem utilizadas nos processos de negócios. No entanto, eles poderão ser alterados caso o usuário ache necessário.

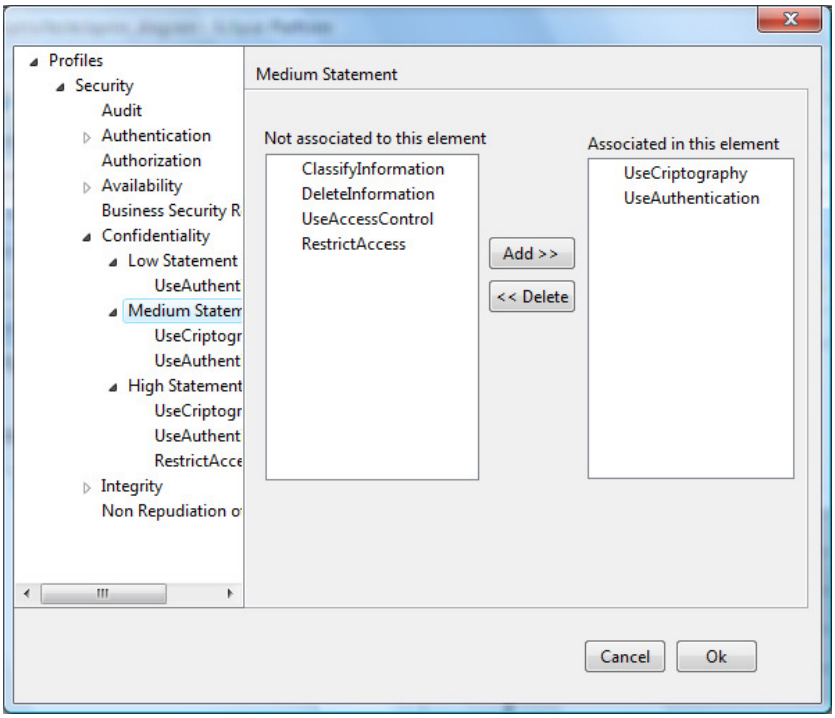


Figura 3.19 - *NF-Actions* Associadas à *Medium Statement* de *Confidentiality*

A Figura 3.20 mostra as propriedades da *NF-Action* "*UseAuthentication*", *Use Digital Signature*, *Session Time* e *Token Type*. Como pode ser visto, as propriedades "*Use*

Digital Signature" e "*Token Type*" são do tipo *ComboProperty* e desta forma o editor criou um *combo* com a lista de possibilidades que podem ser escolhidas pelo usuário. Já a propriedade *Session Time*, do tipo *Property*, possui um campo de texto onde será possível informar qualquer valor inteiro para ela.

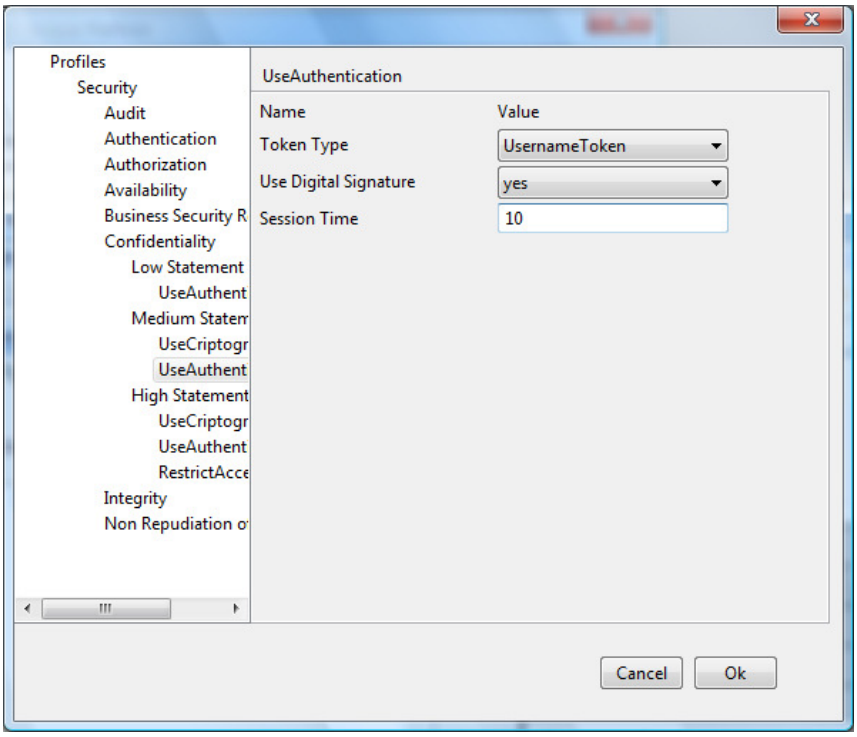


Figura 3.20 - Propriedades do *NF-Action "UseAuthentication"*

A especificação de *profiles* também pode ser expressa em XML, como mostra a Listagem 3.3:

Listagem 3.3 - Representação de *Profile* em XML

```
1    <Profiles tenantID="4EDF74CC-FECF-4e69-B54F-0F4AF03FB33F">
2     <Profile name="Security"
3      <NF-Attributes>
4       <NF-Attribute name ="Confidentiality">
5        <NF-Statement name="Low Statement">
6           <NF-Action name="UseAuthentication">
7              <property name="Session Time" value="10"/>
8              <property name="Token Type" value="UsernameToken"/>
9              <property name="Use Digital Signature" value="no"/>
10          </NF-Action>
11       </NF-Statement>
12       <NF-Statement name="Medium Statement">
13          <NF-Action name="UseAuthentication">
14             <property name="Session Time" value="10"/>
15             <property name="Token Type" value="UsernameToken"/>
16             <property name="Use Digital Signature"value="yes"/>
17          </NF-Action>
```

```
18          <NF-Action name="UseCryptography">
19             <property name="Algorithm" value="TripleDesRsa15"/>
20             <property name="Encryption Type"
                   value="Asymmetric"/>
21             <property name="Encrypted Message Parts"
                  value="Header and Body"/>
22          </NF-Action>
23       </NF-Statement>
24       <NF-Statement name="High Statement">
25          <NF-Action name="UseAuthentication">
26             <property name="Session Time" value="10"/>
27             <property name="Token Type" value="UsernameToken"/>
28             <property name="Use Digital Signature"
                   value="yes"/>
29          </NF-Action>
30          <NF-Action name="UseCryptography">
31             <property name="Algorithm" value="TripleDesRsa15"/>
32             <property name="Encryption Type"
                  value="Asymmetric"/>
33             <property name="Encrypted Message Parts"
                  value="Header and Body"/>
34          </NF-Action>
35         <NF-Action name="RestrictAccess">
36             <property name="PolicyType" value="deny"/>
37             <property name="IP" value="200.20.10.0"/>
38          </NF-Action>
39       </NF-Statement>
40     </NF-Attributes>
41    </Profile>
42   </Profiles>
```

Este *profile* de segurança é uma representação em XML do que foi definido no SSC4Cloud Editor (Figura 3.20). Três *NF-Statements* foram definidos para o *NF-Attribute* "*Confidentiality"* (4-40): *Low Statement*, *Medium Statement* e *High Statement*. O *NF-Statement* "Low Statement" (5-11) apenas contém a implementação da NF-Action "*UseAuthentication*" (6-10), a *NF-Statement Medium Statement* (12-23) contém as implementações das *NF-Actions UseCriptography* e *UseAuthentication*, e o *NF-Statement High Statement* (24-39) contém as implementações das *NF-Actions UseAuthentication*, *UseCriptography* e *RestrictAccess*. Finalmente, cada *NF-Action* tem um conjunto de *NF-Properties* (par de valor), como <*Algorithm*, *TripleDesRsa15*> (19) e <*Encryption Type*, *Asymmetric*> (20).

Após sua criação, um *profile* poderá ser compartilhado com outros usuários (ver Seção 3.5.5), e utilizado nos diagramas BPMN através da visão de segurança (Figura 3.16). Esta visão permite inserir em cada atividade BPMN requisitos de segurança através dos

profiles que o usuário tem acesso, *profiles* esses criados por ele ou compartilhados para ele por outros *tenants*.

As anotações de segurança feitas pelos analistas de negócio na visão de BPMN não poderão ser utilizadas pela *engine* de orquestração para implementação das *NF-Actions* antes da sua configuração. Porém, com base nessas informações iniciais, especialista em segurança usando o SSC4Cloud Editor consulta o repositório e insere as devidas *NF-Actions* suportadas e configura-as de forma que possam ser implementadas pela *engine*.

A Figura 3.21 mostra o *NF-Statement* "*Medium Statement*" do *NF-Attributo* "*Confidentiality*" do *profile* "*Security*" sendo inserido na tarefa "*Processar Pagamento*" do diagrama BPMN modelado. Este *NF-Statement* do *profile Security* possui duas *NF-Actions* que são inseridas automaticamente na tarefa com suas devidas propriedades *default*.

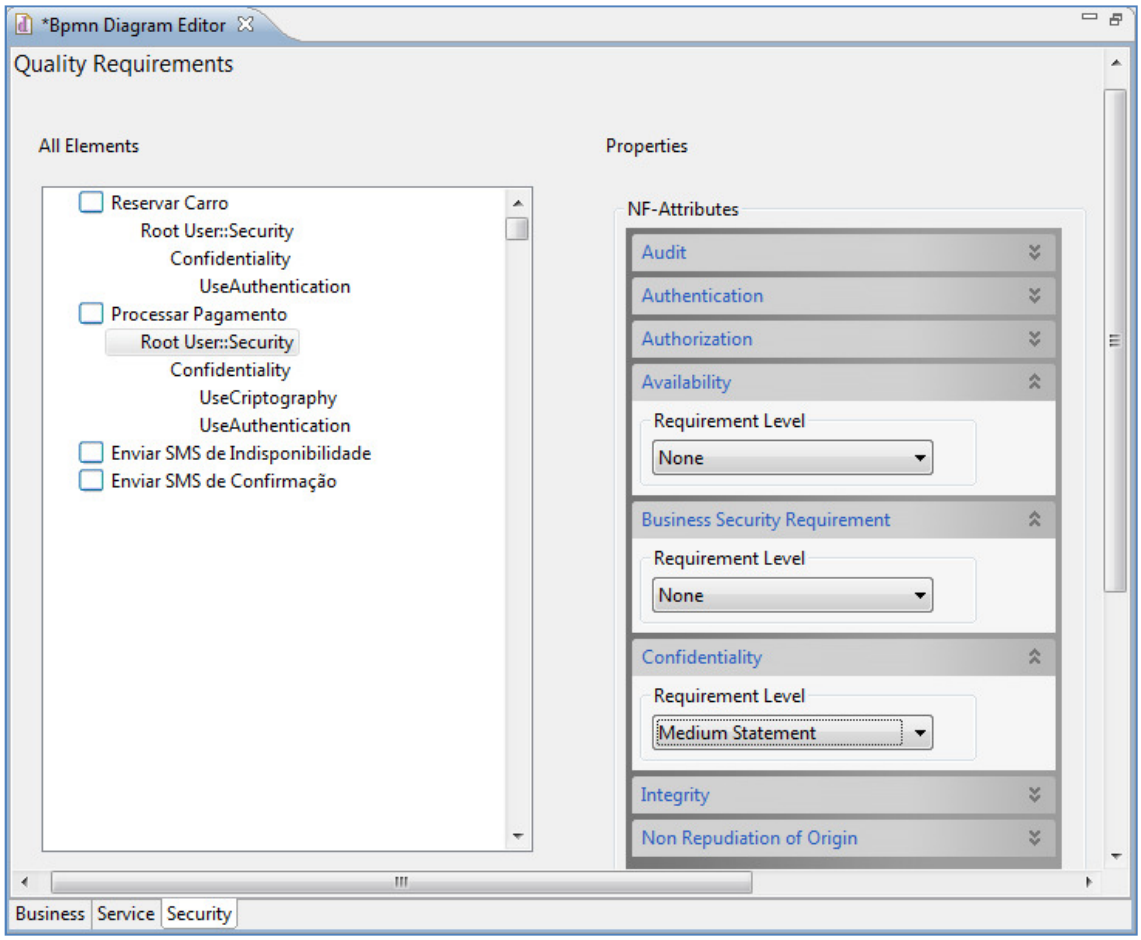


Figura 3.21 - Adicionando Média Confidencialidade na "*Tarefa 1*" do Diagrama BPMN

Caso o usuário tenha a necessidade de remover ou adicionar *NF-Actions* que não estejam inseridas no *NF-Statement* selecionado, ele poderá fazer livremente, conforme mostra a Figura 3.22.

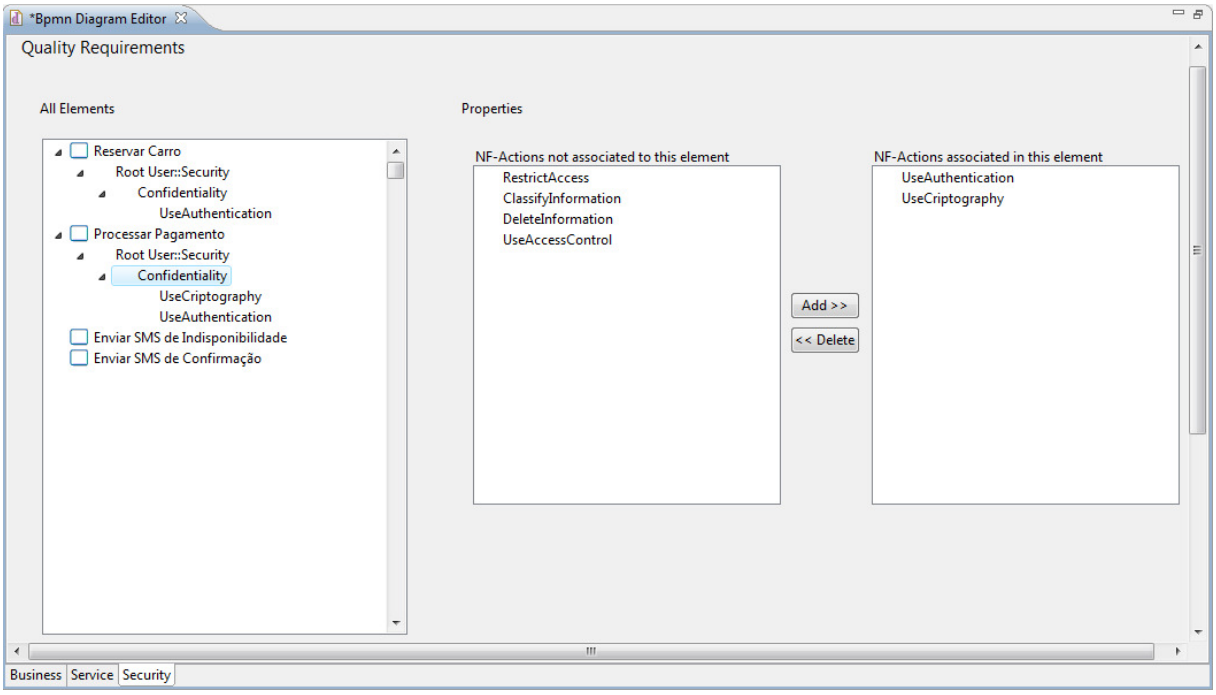


Figura 3.22 - Tela para Remover e Adicionar *NF-Actions* à *NF-Statement* Selecionada

Em tempo de modelagem, o usuário também está apto a modificar as propriedades das *NF-Actions*. Cada serviço invocado pela composição tem características diferentes, como por exemplo, serviços com algoritmos de criptografia diferentes. Ao selecionar uma *NF-Action* (visão de segurança), o especialista terá acesso a todas as propriedades deste elemento. Assim como na criação dos *profiles* de segurança, nesta visão ele poderá selecionar as opções previamente cadastradas no repositório, caso a propriedade seja do tipo *ComboProperty,* ou informar livremente o valor, obedecendo a característica desta propriedade, caso ela seja do tipo *Property*, conforme pode ser visto na Figura 3.23.

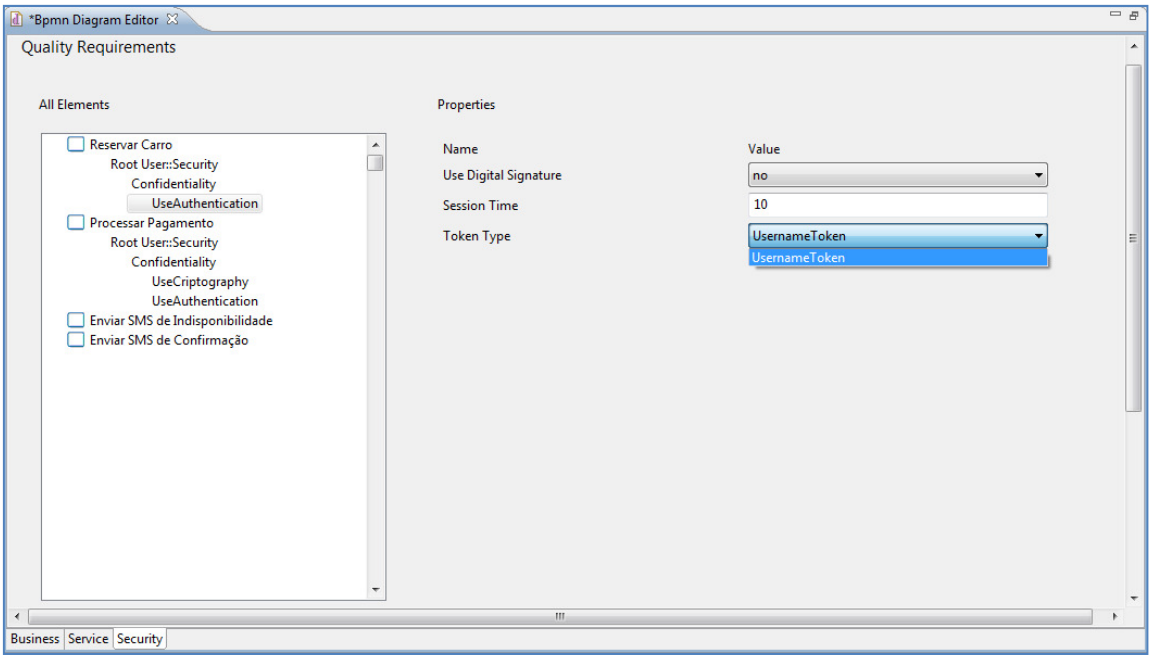


Figura 3.23 - Configuração de *NF-Actions* no Diagrama

Todas as modificações feitas no diagrama através da visão de segurança serão refletidas também na visão de BPMN. A Figura 3.24 mostra o diagrama BPMN da Figura 3.14 após as modificações feitas pelo especialista de segurança. No novo diagrama, os nomes das *NF-Attributes* e *NF-Actions* foram alterados conforme estão cadastrados no repositório de segurança, como exemplo, o *NF-Attributo* Confidencialidade, criado pelo especialista de negócio, passou a se chamar *Confidentiality* e a *NF-Action UsarCriptografia* tornou-se *UseCriptography*.

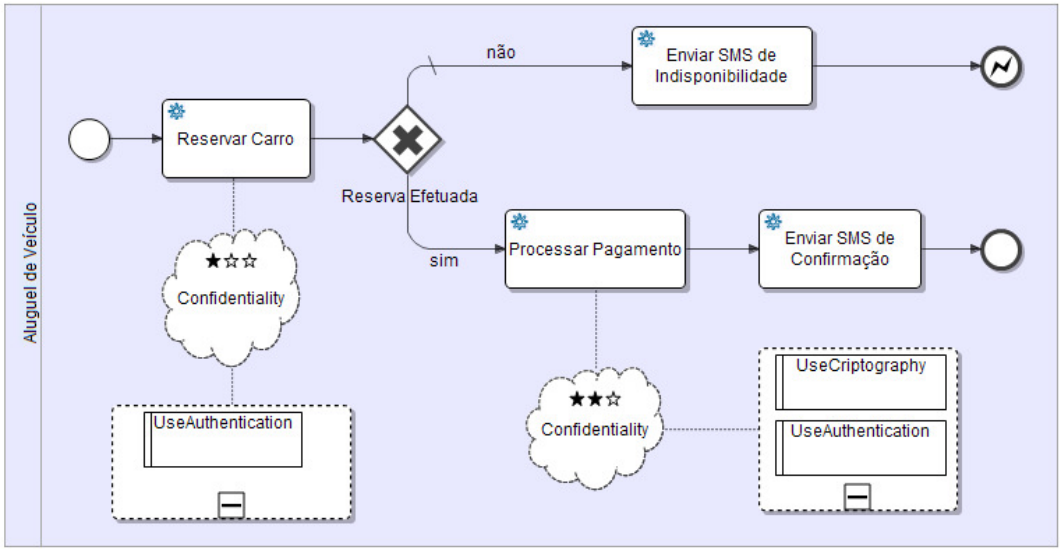


Figura 3.24 - Diagrama BPMN com Abstrações de Segurança Inseridas na Visão de Segurança

O *Security Module* permite a especificação de requisitos de segurança implementáveis nas *engines* de orquestrações disponíveis na SSC4Cloud. Esta especificação é feita através dos *profiles* de segurança que poderão ser criados e reutilizados em diversos diagramas BPMN sem desprezar as particularidades que cada serviço disponível na web possui e sem deixar inconsistência entre as visões de segurança e de negócio.

3.5.4. Service Module

Assim como os processos de negócios modelados e os requisitos de segurança configurados no SSC4Cloud Editor, especialistas em serviço precisam incluir no diagrama informações adicionais sobre os serviços que serão usados para realizar as atividades BPMN. Essas informações adicionais serão utilizadas para geração automática do WS-BPEL (a partir do BPMN) de forma que a *engine* de orquestração, localizada na nuvem, seja capaz de executá-la.

Para evitar problemas de compatibilidade entre os serviços e os requisitos de segurança exigidos na composição, o SSC4Cloud possui um repositório contendo todos os serviços relacionados às *NF-Actions* suportadas por ele e que serão obtidos pelo editor por meio de um *Web Service*. Portanto, é de responsabilidade deste módulo do SSC4Cloud Editor fornecer uma visão para configuração dos serviços e da própria composição, chamada *Service* (Figura 3.25).

Uma composição de serviço é disponibilizada da mesma forma que um *Web Service* simples, através de sua WSDL, e deve possuir o nome da operação ao qual a composição deverá ser invocada, com seus respectivos argumentos de entrada e retorno. No SSC4Cloud Editor, uma composição é representada por um *Pool* BPMN e a visão de serviços permite a configuração desta composição, informando todos os parâmetros necessários para criar sua WSDL.

Uma tarefa BPMN que irá representar uma chamada de serviço deverá ter seu tipo modificado para *Service* (Figura 3.26) para o tradutor do SSC4Cloud saber quais tarefas BPMN serão traduzidas em *invokes* BPEL. Cada atividade de BPMN do tipo *Service* deverá ser associada a um *Web Service* provido pela SSC4Cloud.

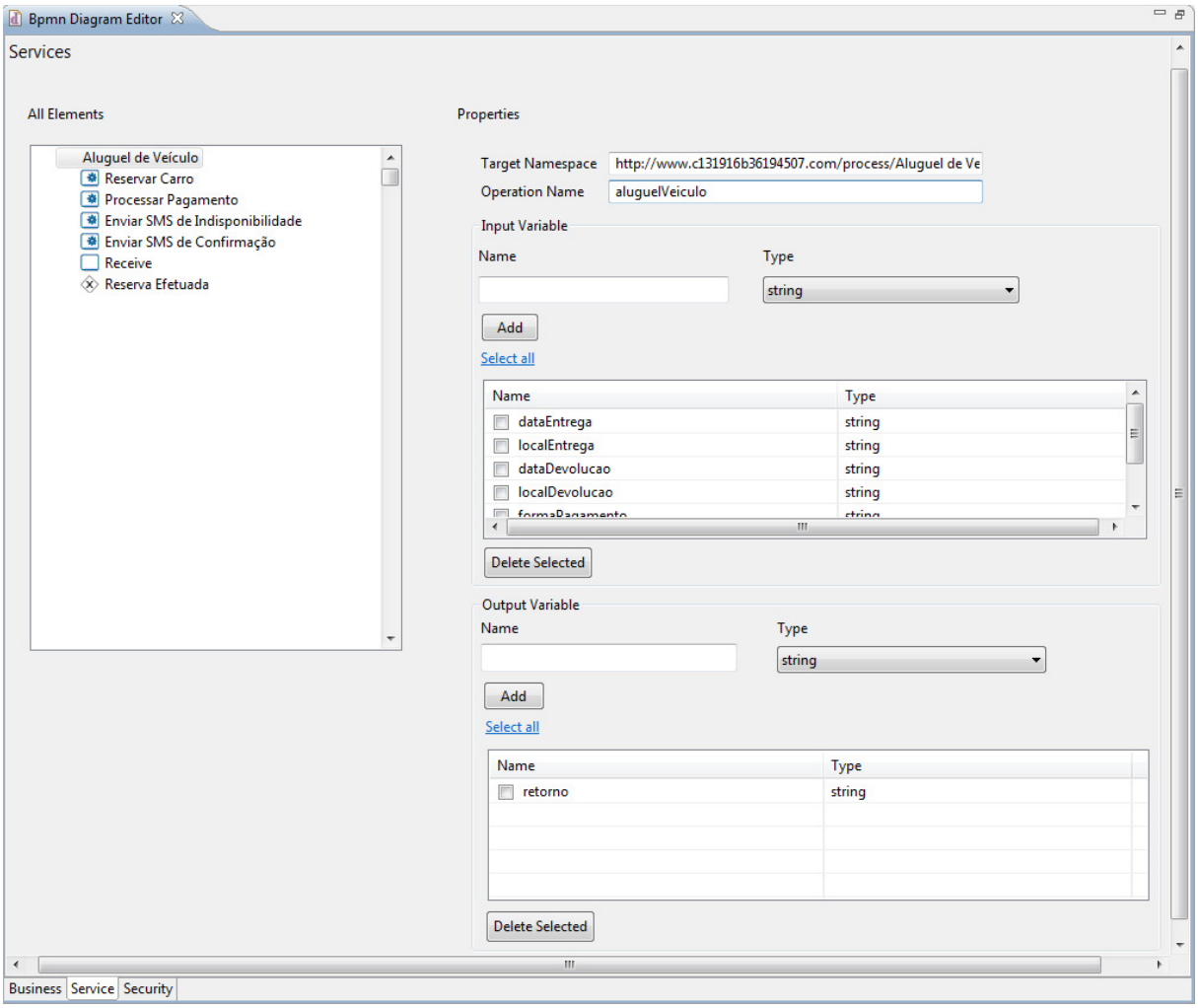


Figura 3.25 - Visão de Serviço: Configuração da Composição

O *Service Module* possui uma funcionalidade para buscar os serviços disponíveis no repositório utilizando filtros, como tipo de negócio, nome da empresa provedora ou descrição do serviço (Figura 3.26). Além desses, também é utilizado como argumento nas buscas as *NF-Actions* associadas à tarefa selecionada para garantir que apenas serviços que implementem essas ações sejam selecionados. Após a seleção do serviço, todas as informações necessárias para a invocação dele serão obtidas pelo editor e anotado no diagrama utilizado posteriormente pelo usuário e pelo tradutor BPMN/BPEL da SSC4Cloud.

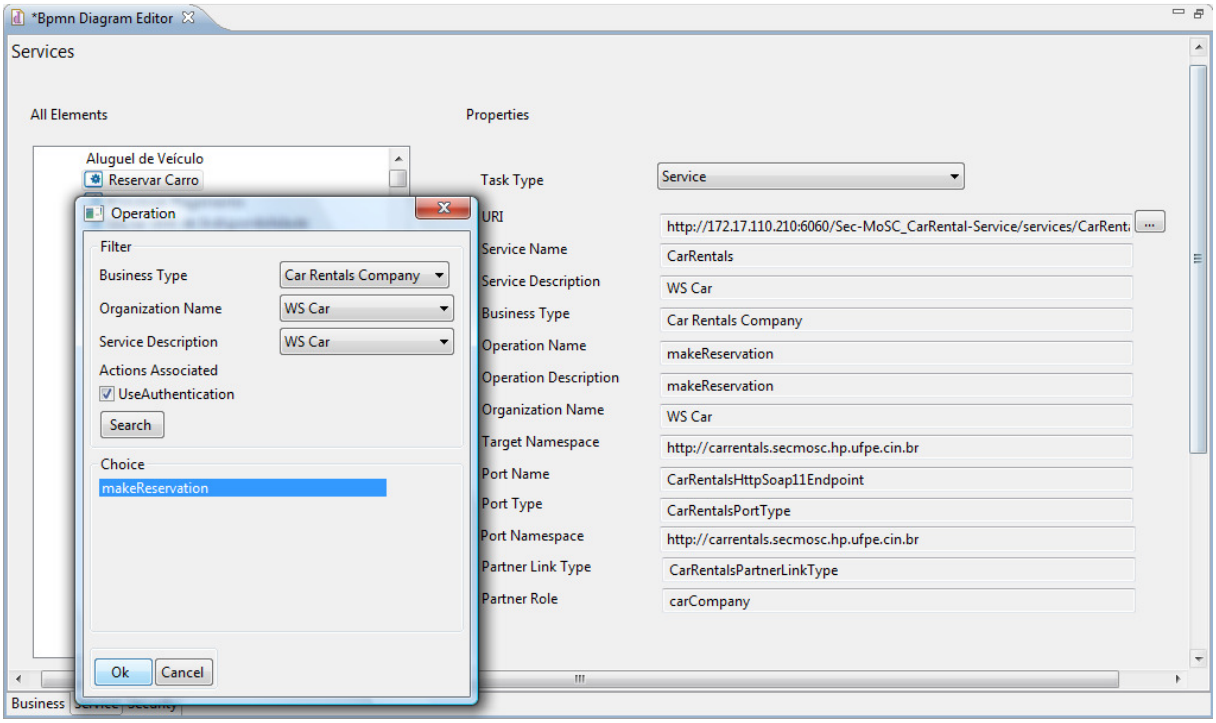


Figura 3.26 - Busca de Serviços Cadastrados no Repositório SSC4Cloud

Uma operação de um *Web Service* poderá possuir argumentos de entrada, informados no ato de sua invocação e valores de retorno que poderão ser utilizados na composição. Em WS-BPEL, uma variável poderá receber valores de outras variáveis ou constantes através da operação *assign*. Para representar as operações de *assign*, o SSC4Cloud Editor utiliza o elemento *Data Object* do BPMN associado às atividades BPMN da composição que representam as invocações de serviço.

A Figura 3.27 mostra o diagrama de "Aluguel de Veículo" com as devidas modificações para possibilitar as operações de *assign* das variáveis da composição. Neste novo diagrama foi adicionada uma nova tarefa (*Receive*) que irá representar o recebimento dos dados da invocação da composição e quatro *Data Objects* que irão representar todas as variáveis utilizadas na composição.

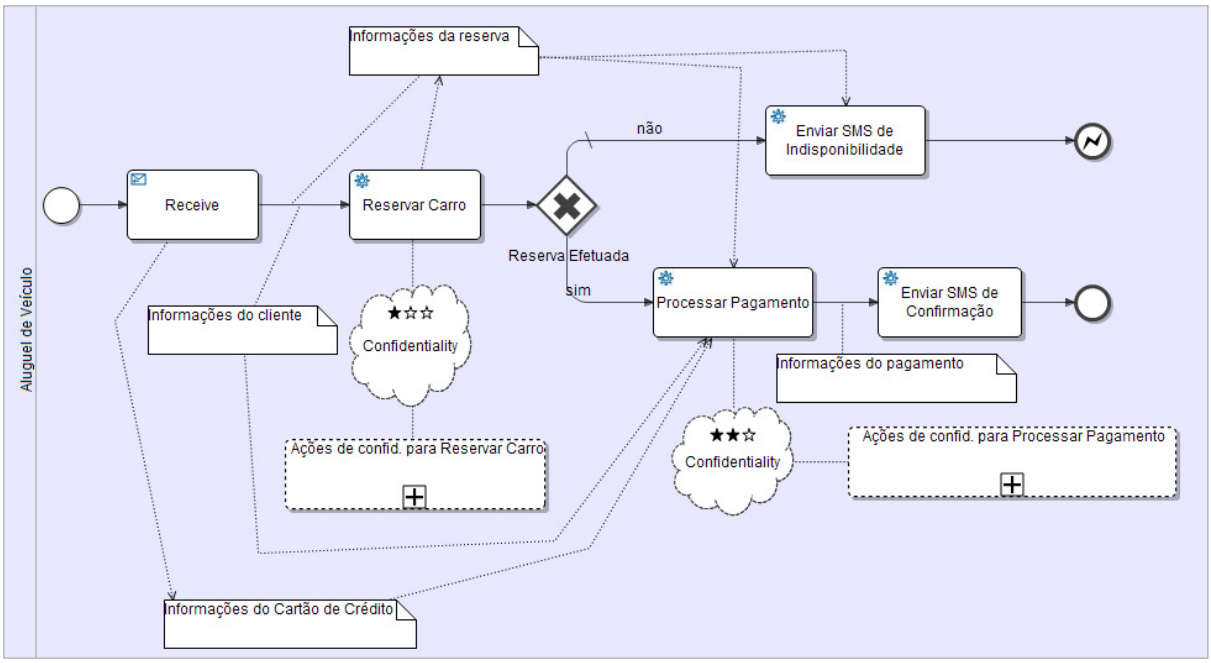


Figura 3.27 - Diagrama BPMN com *Data Objects* para Representar Troca de Dados Entre Tarefas

Após as representações gráficas serem feitas no diagrama BPMN para representar o fluxo dos dados envolvidos na composição, as associações de variáveis poderão ser efetuadas pelos especialistas de serviço. Ao selecionar um *DataObject*, localizado no canto esquerdo da visão de serviço, conforme Figura 3.28, o usuário poderá visualizar todas as variáveis disponíveis para efetuar *assign* de variável (no campo *From* as variáveis de origem e no campo *To* as possíveis variáveis de destino). O *assign* é feito selecionando uma variável de origem juntamente com uma de destino.

A Figura 3.28 mostra ainda a criação de um *assign* no *Data Object* "*Informações do Cliente*" onde a variável *nomeCliente* proveniente do parâmetro de entrada da própria composição é associada ao parâmetro *customerName* da operação *makeReservationRequest*.

Em BPMN um diagrama pode ter vários fluxos de sequência representados pelo elemento *Gateway*. Este elemento possui como atributo o fluxo *default*, representado por uma linha perpendicular ao elemento de seqüência, que será seguido caso todas as condições dos outros fluxos pertencentes a ele não sejam aceitas.

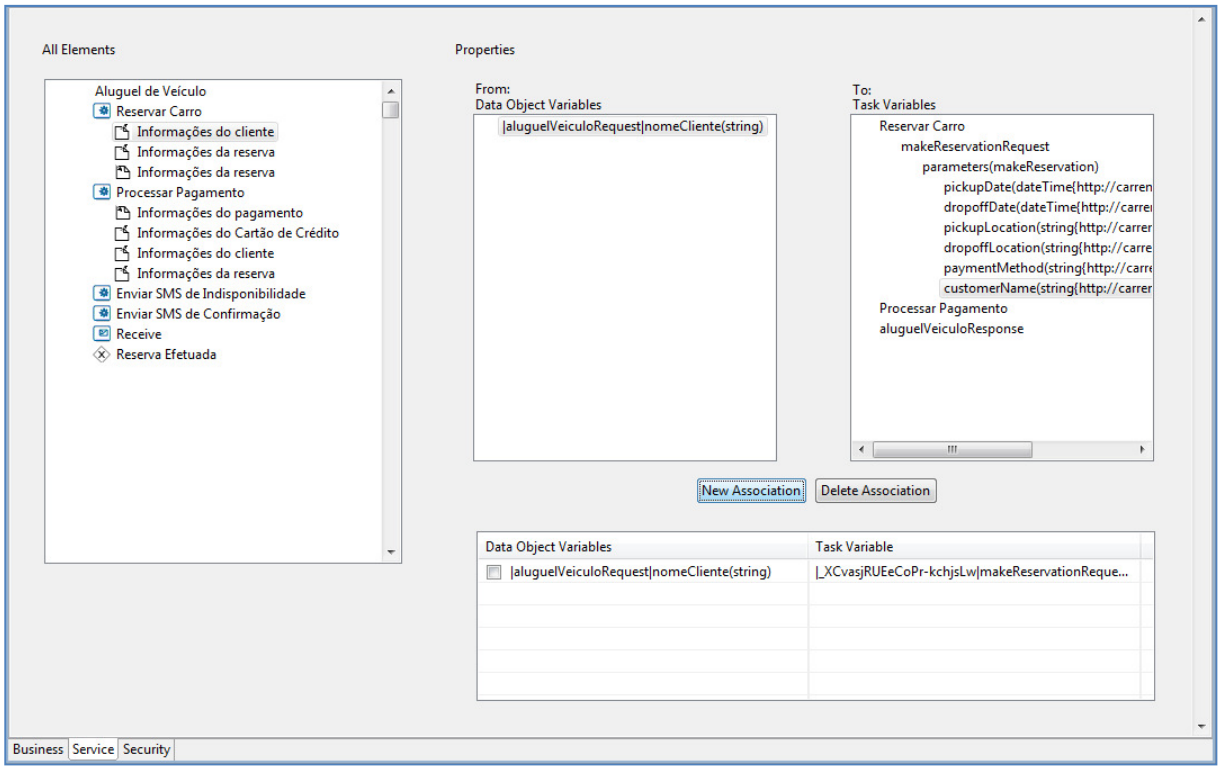


Figura 3.28 - *Assign* de Variáveis

No SSC4Cloud Editor um *Gateway* deverá possuir apenas dois fluxos, o principal, seguido caso uma determinada condição seja aceita e o fluxo *default* que será selecionado se a condição dada não for aceita. A composição "Aluguel Veículo" (Figura 3.27) possui um *Gateway* que verifica se a reserva foi realmente efetuada. Caso ela seja efetuada, o fluxo segue normalmente para tentar efetuar o pagamento da locação, mas caso ocorra algum problema o fluxo é seguindo pelo caminho *default* que irá enviar uma mensagem para o cliente informando o problema na reserva.

Como dito anteriormente, um *Gateway* precisa de uma condição para saber qual fluxo o diagrama deve tomar. A visão de serviço possui uma funcionalidade para que o usuário possa informar essa condição. Quando o especialista seleciona um *Gateway*, o editor mostra os parâmetros a serem configurados para este elemento. Nele o usuário informará a variável, obtidas dos elementos *Data Objects*, utilizada na condição do fluxo principal, o operador da condição e o valor que será comparado ao da variável.

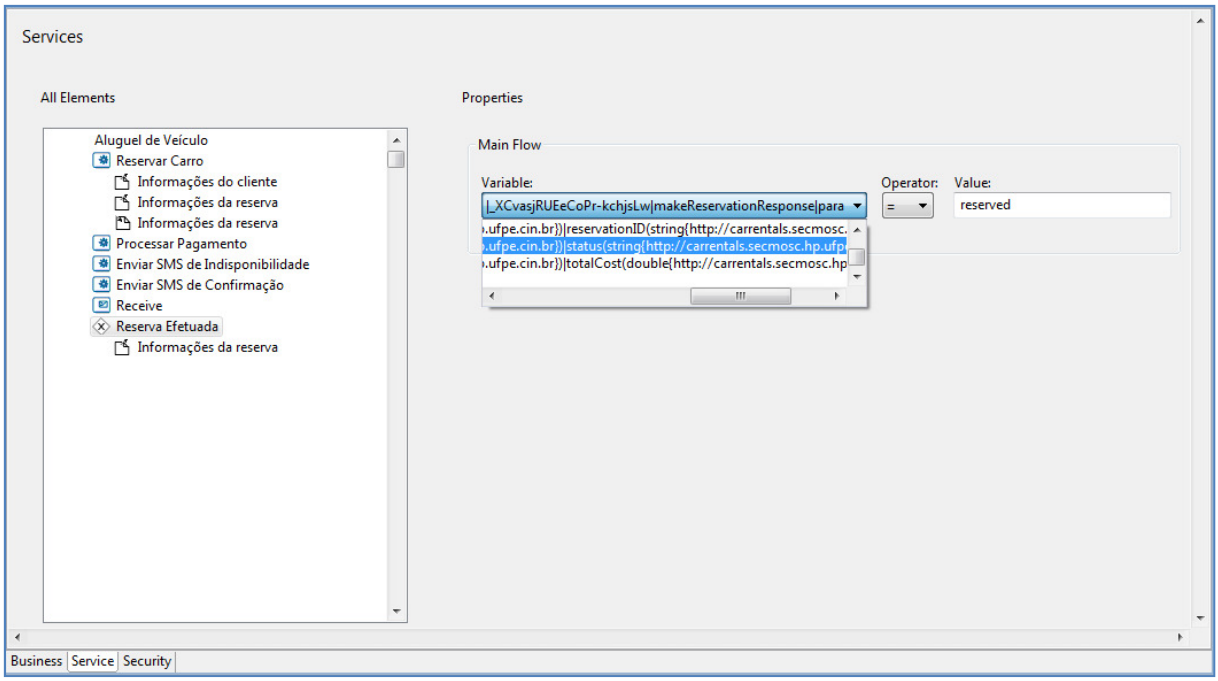


Figura 3.29 - Configuração de *Gateway*

Como foi visto, esse módulo provê ao especialista de serviço todos os meios para ele configurar a composição deixando-a executável na *engine* de orquestração da SSC4Cloud.

3.5.5. Sharing Module

O módulo de compartilhamento (*Sharing Module*) do SSC4Cloud Editor é responsável por fornecer a funcionalidade de compartilhamento de conhecimento e experiência, de segurança e de modelagem de BPMN, com outros usuários (*tenants*) da nuvem.

Neste módulo o usuário dispõe de uma tela onde ele poderá visualizar todos os projetos e seus respectivos processos de negócios e *profiles* de segurança criados por ele (Figura 3.30). Um *tenant* que obtiver um *profile* de segurança compartilhado poderá utilizá-lo na modelagem dos requisitos de segurança de suas composições. Porém, não poderá fazer nenhuma modificação na configuração desse *profile*.

No compartilhamento de processos de negócio o usuário que obtiver um processo compartilhado poderá fazer suas devidas modificações para o seu negócio. Porém, essas modificações não serão replicadas no diagrama original, mas sim em uma cópia criada pelo editor para o usuário em questão.

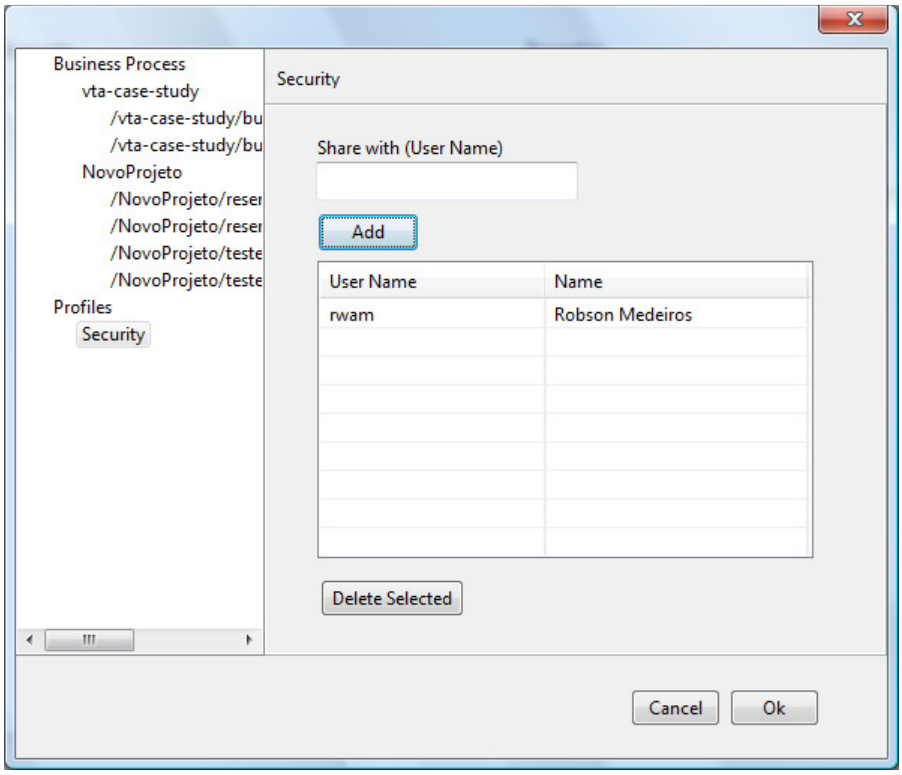


Figura 3.30 - Tela de Compartilhamento de *Profiles* de Segurança e Processos de Negócio

Portanto, usuários podem compartilhar *profiles* de segurança e processos de negócio, porém em modo somente leitura, onde não será possível fazer modificações nas suas configurações originais.

3.5.6. Engine Module

Com os processos de negócio devidamente criados e configurados, os usuários poderão gerenciar suas execuções na nuvem através do *Engine Module* do SSC4Cloud Editor.

Este módulo dispõe de quatro serviços, onde poderá instalar, iniciar, parar ou pausar um processo de negócio na nuvem, utilizando para isso quatro botões disponíveis no editor (Figura 3.31) que tem como finalidade chamar os serviços providos pela nuvem.

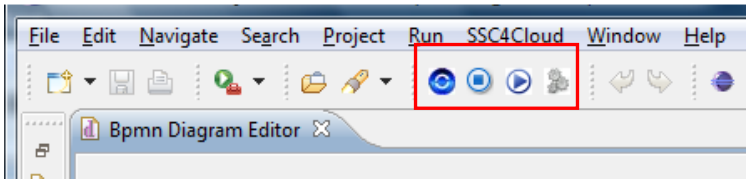


Figura 3.31 - Botões de Gerenciamento de Execução de Composição

Ao instalar um processo de negócio na nuvem, o SSC4Cloud utiliza o arquivo BPMN anotado e gerado pelo editor para geração de artefatos independente de plataforma de execução (Figura 3.32).

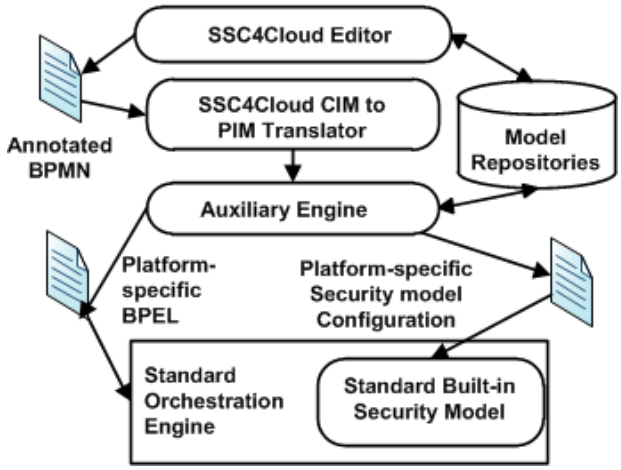


Figura 3.32 - Arquitetura do Ambiente de Desenvolvimento Orientado a Modelo do SSC4Cloud

Conforme a arquitetura orientado a modelo do SSC4Cloud, o SSC4Cloud Editor gera o BPMN anotado com os requisitos de segurança e serviços da composição que servirá como insumo para geração de artefatos independentes de plataforma pelo módulo de tradução e passados para a Engine Auxiliar, que por sua vez gera todos os artefatos necessários para a execução do processo, com seus requisitos não funcionais, para uma engine de orquestração específica, tal como o Apache ODE (APACHE ODE).

3.6 Considerações Finais

Este capítulo apresentou o *SSC4Cloud Editor*, uma ferramenta para modelagem de processos de negócio com anotações de segurança para uma execução na nuvem. Foram apresentados os requisitos que serviram de base para a definição da arquitetura assim como a própria arquitetura. Por fim, foi feita um detalhamento da implementação, incluindo todos os módulos nos quais o *SSC4Cloud Editor* foi estruturado.

4 ESTUDO DE CASO: AGÊNCIA DE VIAGENS VIRTUAL

Este capítulo apresenta o estudo de caso utilizado para demonstrar a abordagem proposta neste trabalho.

4.1 Agência de Viagem Virtual

Virtual Travel Agency (VTA) (STOLLBERG, LAUSEN, *et al.*, 2004) é uma agência que disponibiliza serviços de turismo através de um portal da Internet para clientes interessados em pacotes de viagens. A agência funciona como uma interface (portal) entre usuários finais e as empresas que provêem os serviços propriamente ditos, como por exemplo, hotéis, locadoras de automóveis e companhias aéreas. Todos esses serviços também disponibilizados na Internet por meio de *Web Services*.

Aqui será visto todo o processo para executar o caso de uso de compra de passagem aérea nacional do VTA, desde a criação de *profiles* de segurança e do modelo do processo de negócio até a geração dos artefatos de execução pela *engine* auxiliar, localizada na nuvem. Este caso de uso consiste na compra de uma passagem aérea nacional pelo cliente através do portal da VTA. O portal VTA processa a requisição e pagamento e envia um SMS de confirmação. A Tabela 4.1 mostra os atores do caso de uso e seus papeis.

Tabela 4.1 - Atores e Papeis do Caso de Uso

Ator	Descrição	Papel
Usuário	Usuário final que solicita um serviço provido pelo VTA.	Interagir com o VTA através do portal.
Provedores de Serviço Comercial	Companhias que fornecem serviços específicos para o VTA (e.g. companhias aéreas).	Prover serviços na forma de *Web Service*, para isso deverá ter um contrato de uso com a VTA.
VTA	Companhia que fornece serviços de turismo agregado.	Fornecer serviços de turismo para os usuários através de uma interface.

O caso de uso "*Comprar passagem aérea nacional*" possui os cenários conforme descritos abaixo:

Cenário Principal de Sucesso

1. Usuário acessa o portal VTA e seleciona a opção de viagem nacional;
2. O portal VTA traz a página web necessária para a compra de uma passagem aérea nacional;
3. Usuário entra com as informações sobre o trecho desejado;
4. Usuário entra com as informações de pagamento;
5. O portal VTA envia a requisição para a companhia área;
6. A companhia aérea verifica a disponibilidade do vôo;
7. O portal VTA recebe o resultado do processamento da companhia aérea;
8. O portal VTA envia a requisição de autorização de pagamento à administradora de cartão de crédito;
9. Administradora de cartão de crédito processa a solicitação de pagamento;
10. Portal VTA recebe o resultado do processamento do pagamento;
11. Portal VTA faz a reserva da passagem área junto com a companhia aérea;
12. O portal VTA envia o bilhete eletrônico para o usuário via SMS.

Extensões

No passo 5, se não existir nenhum vôo que satisfaça a requisição do usuário:

1. O usuário receberá uma mensagem do portal informando a indisponibilidade do vôo.

No passo 8, se o pagamento não for autorizado:

1. O usuário receberá uma mensagem do portal informando da impossibilidade de pagamento.

Requisitos de Segurança

1. O portal VTA deverá criptografar as informações de pagamento antes de enviá-las para a administradora de cartão de crédito;
2. O portal VTA e seus parceiros deverão utilizar certificados digitais válidos;

3. Mecanismos de autenticação deverão ser utilizados durante qualquer comunicação do portal VTA com seus parceiros;
4. Parceiros do portal VTA podem desejar que seus *Web Services* devam ser acessados somente de requisições vindas de certos endereços IP.

4.2 Modelagem

Para modelar o estudo de caso "*Comprar passagem aérea nacional*" do VTA, a ferramenta SSC4Cloud Editor foi utilizada por três especialista: negócio, segurança e serviço. O especialista de negócio criou o modelo de processo de negócio com base nos requisitos funcionais e no fluxo de atividade do cenário do estudo de caso. Para modelagem dos requisitos de segurança, o especialista de segurança utilizou as abstrações definidas na Seção 3.1 e criou os *profiles* de segurança através do módulo de segurança do SSC4Cloud Editor.

As *NF-Actions* utilizadas para implementar os requisitos de segurança foram:

- *UseCryptography* : usada para implementar a criptografia dos dados na comunicação entre o portal VTA e as companhias de cartão de crédito;
- *RestrictAccess* : usada para restringir o acesso ao portal VTA ou restringir acesso à serviços de parceiros através da sua localização determinada pelo endereço IP (*Internet Protocol*) (POSTEL, 1981);
- *UseAuthentication* : usada para implementar a necessidade de autenticação entre o portal VTA e seus parceiros.

Para atender a estes requisitos de segurança foram criados dois *NF-Statements*, "*Low Statement*" e "*Medium Statement*" referentes a *NF-Attribute Confidenciality*. No "*Low Statement*" o especialista inseriu a *NF-Action RestrictAccess* e no *Medium Statement* as *NF-Actions UseCriptography* e *UseAuthentication*, conforme pode ser visto na Figura 4.1.

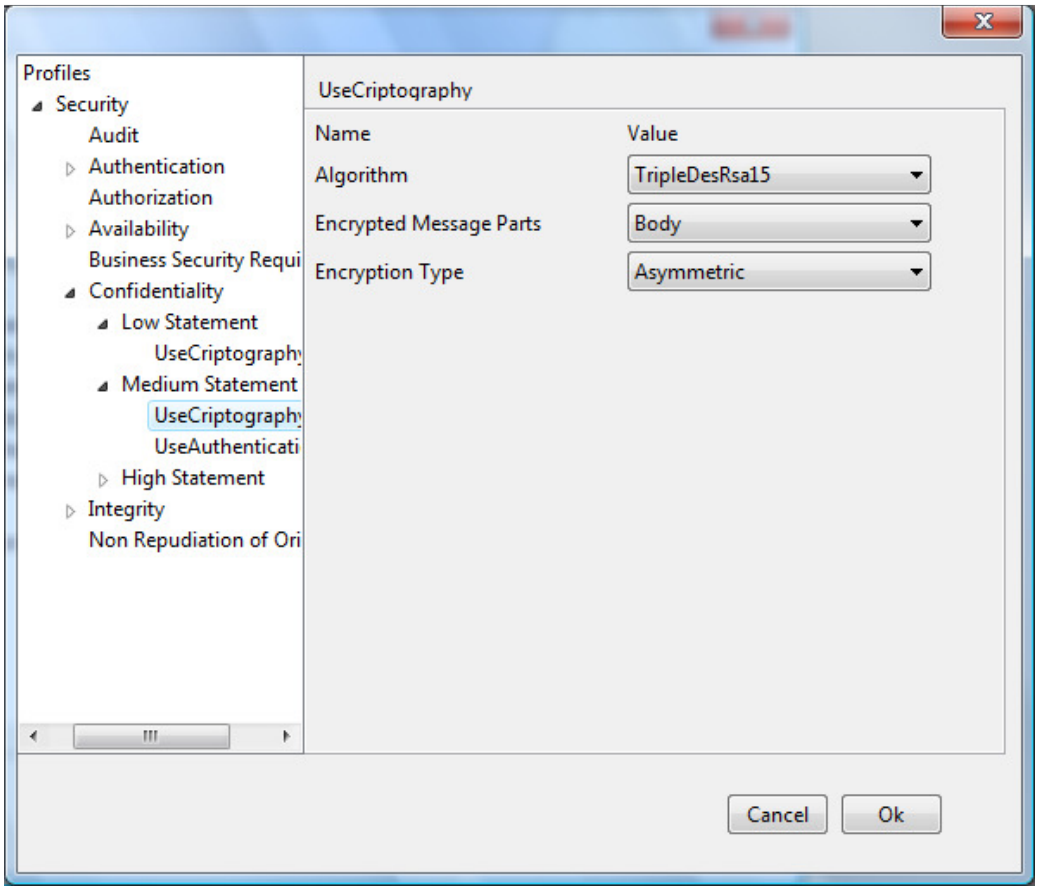


Figura 4.1 - *Profile* para o Estudo de Caso Comprar Passagem Aérea Nacional

As propriedades dessas *NF-Actions*, estabelecidas pelo especialista de segurança, foram:

- *RestrictAccess*:
  - IP: **209.93.4.0 - 217.114.221.0**
  - *polyceType*: ***allow***
- *UseAuthentication*
  - *Session Time*: **3600000**
  - *Use Digital Signature*: ***Yes***
  - *Token Type*: ***UsernameToken***
- *UseCriptography*
  - *Encrypted Message Parts*: ***Body***
  - *Encryption Type*: ***Asymmetric***
  - *Algorithm*: ***TripleDesRsa15***

Com o *profile* de segurança criado, o especialista de segurança o compartilhou com o especialista de negócio que, com isso, pôde inserir na modelagem todos os requisitos exigidos no caso de uso, tanto funcionais como não-funcionais.

A Figura 4.2 mostra a modelagem da composição do caso de uso "*Comprar Passagem Aérea Nacional*" do VTA com os requisitos funcionais e não-funcionais.

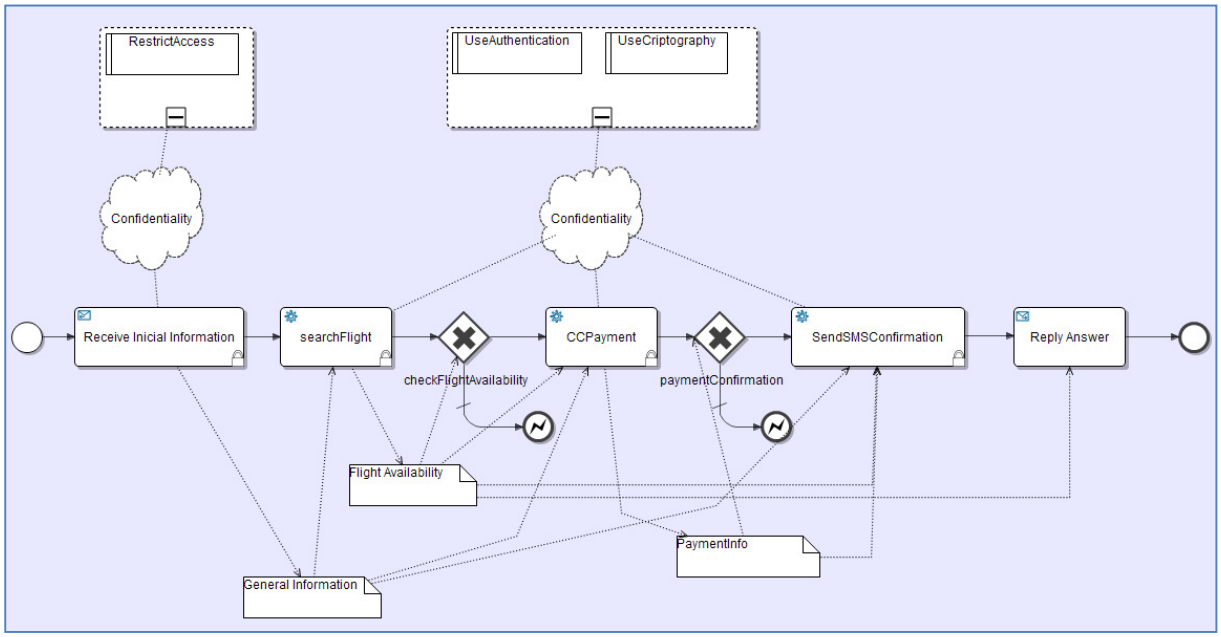


Figura 4.2 - Processo de Negócio com Requisitos de Segurança

Com o diagrama modelado e os requisitos de segurança configurados, o especialista de serviços, através da visão de serviço, configurou a interface da composição, informando o nome da operação, argumentos de entrada e variável de saída, conforme Figura 4.3.

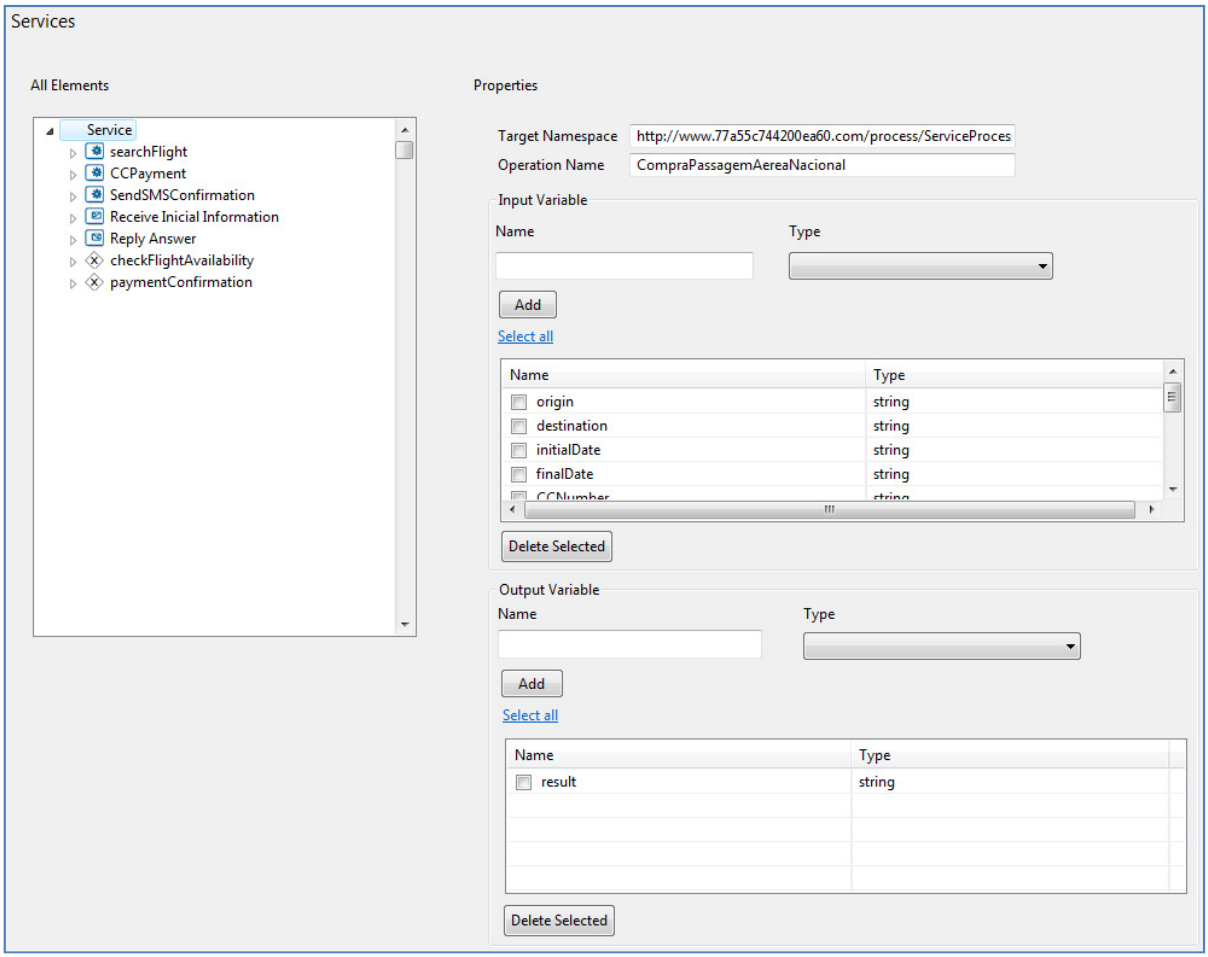


Figura 4.3 - Configuração da Composição

As configurações efetuadas na interface da composição através do editor geram anotações no arquivo BPMN. A Listagem 4.1 mostra o arquivo BPMN anotado gerado através das configurações efetuadas conforme a Figura 4.3.

Listagem 4.1 – BPMN Anotado com as Configurações da Interface da Composição

```
    ...
1     <eAnnotations xmi:type="ecore:EAnnotation"
    xmi:id="_ss33YLmSEd6hIsEHyXmPrQ"source="service">
2       <details xmi:type="ecore:EStringToStringMapEntry"
        xmi:id="_stBoYLmSEd6hIsEHyXmPrQ" key="operation_name"
        value="CompraPassagemAereaNacional"/>
3       <details xmi:type="ecore:EStringToStringMapEntry"
    xmi:id="_ovyVAM8CEd6E4N-mIbFN_A" key="target_namespace"
    value="http://www.77a55c744200ea60.com/process/ServiceProcess"/>
4       <details xmi:type="ecore:EStringToStringMapEntry"
    xmi:id="_uqnUsM8CEd6E4N-mIbFN_A" key="uri"
    value="http://172.17.110.210:6060/Sec-MoSC_GDS...."/>
5     </eAnnotations>

6     <eAnnotations xmi:type="ecore:EAnnotation"
        xmi:id="_15nssMMHEd6h0OEYQYulGw" source="interface_input">
```

```
7         <details xmi:type="ecore:EStringToStringMapEntry"
            xmi:id="_15nsscMHEd6h0OEYQYulGw" key="origin"
      value="string"/>
8         <details xmi:type="ecore:EStringToStringMapEntry"
            xmi:id="_2mLQsMMHEd6h0OEYQYulGw" key="destination"
      value="string"/>
9         <details xmi:type="ecore:EStringToStringMapEntry"
            xmi:id="_4eutwMMHEd6h0OEYQYulGw" key="initialDate"
      value="string"/>
10        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_5oWBMMMHEd6h0OEYQYulGw" key="finalDate"
      value="string"/>
11        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_7SIpgMMHEd6h0OEYQYulGw" key="CCNumber"
      value="string"/>
12        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_78dY4MMHEd6h0OEYQYulGw" key="CCName"
      value="string"/>
13        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_85QYUMMHEd6h0OEYQYulGw" key="CCExpiration"
      value="string"/>
14        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_vFGcMMMYEd6h0OEYQYulGw"key="cellPhoneNumber"
      value="string"/>
15        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_R5avsMMZEd6h0OEYQYulGw" key="address"
      value="string"/>
16        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_SfkJsMMZEd6h0OEYQYulGw" key="zipcode"
      value="string"/>
17        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_SnL88MPPEd6MW7dZL2kVvg" key="email"
      value="string"/>
18        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_TfznAMPPEd6MW7dZL2kVvg" key="countryCode"
      value="string"/>
19      </eAnnotations>

20      <eAnnotations xmi:type="ecore:EAnnotation"
          xmi:id="_rixGMMPOEd6MW7dZL2kVvg" source="interface_output">
21        <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_rixGMcPOEd6MW7dZL2kVvg" key="result"
      value="string"/>
22      </eAnnotations>

23    ...
```

A Linha 2 contém o nome da operação da composição (`CompraPassagemAereaNacional`) usada para iniciar sua execução. O *namespace* da composição está sendo representado na Linha 3 onde, para esta composição, foi denominada de `http://www.77a55c744200ea60.com/process/ServiceProcess;` a linha 4 o endereço onde o serviço será acessível. As variáveis de entrada da composição estão inseridas neste arquivo nas Linhas 6-19 e as de saída encontram-se na linha 21 com o nome `result` do tipo *string*.

Todas as tarefas BPMN foram associadas aos respectivos serviços previamente cadastrados na nuvem através de suas WSDL (Figura 4.4). Esses serviços são localizados e inseridos na tarefa por meio da visão de serviço pelos especialistas de serviço.

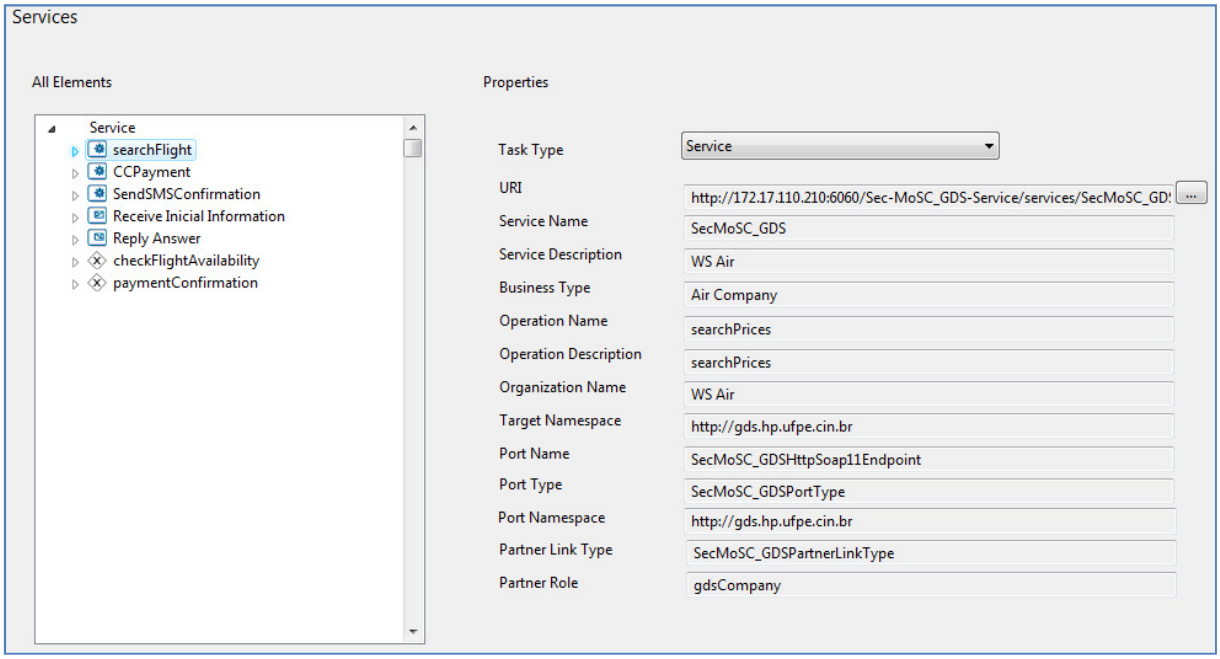


Figura 4.4 - Configuração do Serviço da Tarefa *searchFlight*

Assim como na configuração da interface da composição, as configurações dos serviços também geram anotações no arquivo BPMN.

Listagem 4.2 – BPMN Anotado com as Configurações do Serviço *serchFlight*

```
1   ...
2   <vertices xmi:type="bpmn:Activity" name="searchFlight"...>
3     <eAnnotations xmi:type="ecore:EAnnotation"
       xmi:id="_tMLQQLmXEd6UmLQomVegcA" source="service">
4     <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMLQQbmXEd6UmLQomVegcA" key="service_name"
    value="SecMoSC_GDS"/>
5     <details xmi:type="ecore:EStringToStringMapEntry"
      xmi:id="_tMLQQrmXEd6UmLQomVegcA"key="service_description"
    value="WS Air"/>
6     <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMLQQ7mXEd6UmLQomVegcA" key="uri"
    value="http://172.17.110.210:6060/Sec-MoSC_GDS-
    Service/services/SecMoSC_GDS.SecMoSC_GDSHttpSoap11Endpoint/"/>
7     <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMLQRLmXEd6UmLQomVegcA" key="business_type"
    value="Air Company"/>
8     <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMLQRbmXEd6UmLQomVegcA" key="operation_name"
    value="searchPrices"/>
9     <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMUaMLmXEd6UmLQomVegcA" key="operation_description"
```

```
    value="searchPrices"/>
10    <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMUaMbmXEd6UmLQomVegcA" key="organization_name"
    value="WS Air"/>
11    <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMUaMrmXEd6UmLQomVegcA" key="target_namespace"
    value="http://gds.hp.ufpe.cin.br"/>
12    <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMUaM7mXEd6UmLQomVegcA" key="port_name"
    value="SecMoSC_GDSHttpSoap11Endpoint"/>
13    <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMUaNLmXEd6UmLQomVegcA" key="port_type"
    value="SecMoSC_GDSPortType"/>
14    <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMUaNbmXEd6UmLQomVegcA" key="partner_link_type"
    value="SecMoSC_GDSPartnerLinkType"/>
15    <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="_tMUaNrmXEd6UmLQomVegcA" key="partner_role"
    value="gdsCompany"/>
16    <details xmi:type="ecore:EStringToStringMapEntry"
       xmi:id="__0_qQMsWEd6zmqxONp343Q" key="port_namespace"
    value="http://gds.hp.ufpe.cin.br"/>
17  </eAnnotations>
18  ...
19  </vertices>
```

A Linha 3 informa que a atividade searchFlight é do tipo `service` e será transformada em uma ação de *invoke* do WS-BPEL. As Linhas 04-17 referem-se às propriedades do serviço selecionado para esta atividade. Esses valores são obtidos do repositório da SSC4Cloud e serão utilizados nas chamadas dos respectivos *web services*.

Para efetuar os *assigns* das variáveis, primeiramente o usuário deve atribuir nos elementos *DataObjects* do BPMN as variáveis as quais ela irá representar. Esta atribuição só pode ocorrer entre as tarefas e os *Data Objects*. No estudo de caso, o *DataObject* "*General Information"* recebeu todas as variáveis referentes a interface da composição, conforme Figura 4.5. Neste instante, qualquer atividade que tenha necessidade de obter algum dado proveniente de alguma variável de entrada da composição deverá utilizar este elemento.

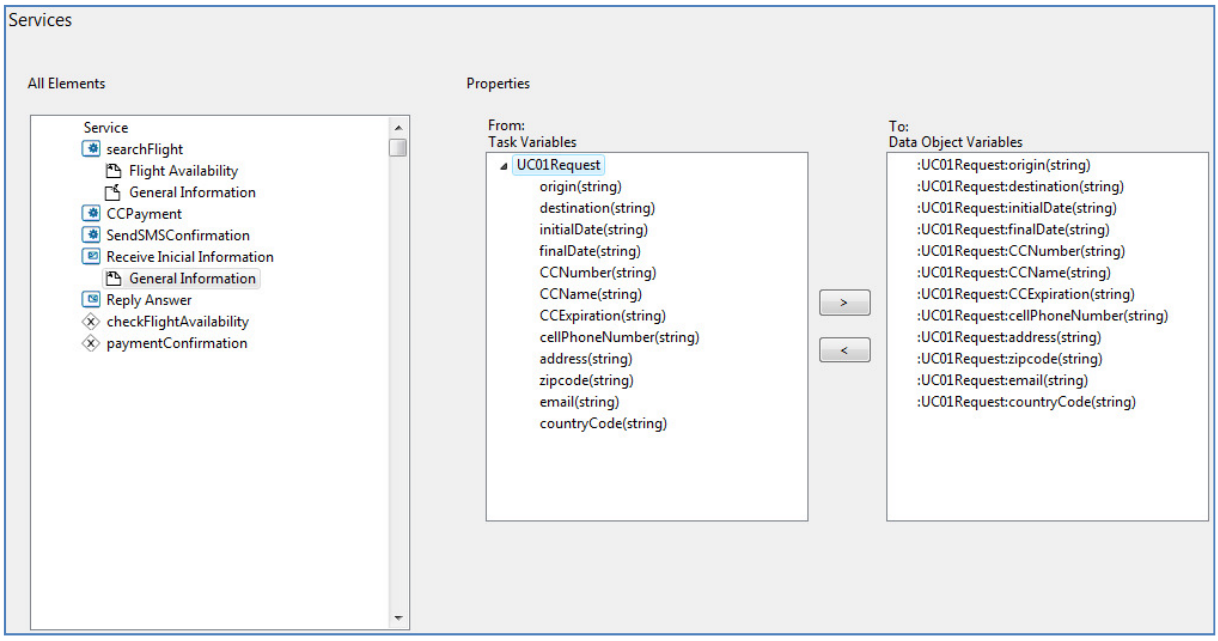


Figura 4.5 - Associação das Variáveis de Entrada da Composição ao *DataObject General Information*

As anotações geradas no *DataObject* "*General Information*" do arquivo BPMN podem ser vistas na Listagem 4.3. Nas 3-16 encontram-se todas as assinaturas das variáveis adicionadas a este elemento: `origin` representa o aeroporto de origem (4); `destination` representa o aeroporto de destino da viagem (5); `initialDate` e `finalDate` (6-7) definem a data de início e fim da viagem, respectivamente; `CCNumber`, `CCName` e `CCExpiration` representam o número do cartão de crédito, o nome impresso no cartão e a data que o cartão expira, respectivamente; e `cellPhoneNumber`, `zipcode`, `email` e `countryCode` definem o endereço, o CEP, o e-mail e o código do pais, respectivamente.

Listagem 4.3 – BPMN Anotado com as Configurações Efetuadas no *Data Object* Após a Associação de Variáveis

```
1  . . .
2  <artifacts xmi:type="bpmn:DataObject" xmi:id="_fkm_l0EBEeC-
   xewWlIfNPg"
              iD="_l9nbALmbEd6UmLQomVegcA" name="General
   Information">
3    <eAnnotations xmi:type="ecore:EAnnotation"
      xmi:id="_f0MmQMPOEd6MW7dZL2kVvg" source="assign">
4      <details xmi:type="ecore:EStringToStringMapEntry"
        xmi:id="_R68g0MsIEd6Vz_v8VGA_MQ"
        key=":UC01Request:origin(string)"
        value=":UC01Request:origin(string)"/>
5      <details xmi:type="ecore:EStringToStringMapEntry"
        xmi:id="_SQG4gMsIEd6Vz_v8VGA_MQ"
        key=":UC01Request:destination(string)"
        value=":UC01Request:destination(string)"/>
```

```
6       <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_SaPE4MsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:initialDate(string)"
         value=":UC01Request:initialDate(string)"/>
7       <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_Sk9uMMsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:finalDate(string)"
         value=":UC01Request:finalDate(string)"/>
8       <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_SvPrkMsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:CCNumber(string)"
         value=":UC01Request:CCNumber(string)"/>
9       <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_S5-U4MsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:CCName(string)"
         value=":UC01Request:CCName(string)"/>
10      <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_TD0NYMsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:CCExpiration(string)"
         value=":UC01Request:CCExpiration(string)"/>
11      <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_TP4TcMsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:cellPhoneNumber(string)"
         value=":UC01Request:cellPhoneNumber(string)"/>
12      <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_Tf0z8MsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:address(string)"
         value=":UC01Request:address(string)"/>
13      <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_Trl_EMsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:zipcode(string)"
         value=":UC01Request:zipcode(string)"/>
14      <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_T39nIMsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:email(string)"
         value=":UC01Request:email(string)"/>
15      <details xmi:type="ecore:EStringToStringMapEntry"
         xmi:id="_UECUQMsIEd6Vz_v8VGA_MQ"
         key=":UC01Request:countryCode(string)"
         value=":UC01Request:countryCode(string)"/>
16    </eAnnotations>

17   ...

18 </artifacts>

19 ...
```

Após a associação das variáveis nos *Data Objects*, essas poderão ser utilizadas como argumentos dos parâmetros dos serviços invocados pela composição. Para isso, todas as atividades que tenham a necessidade de obter o valor de alguma variável representada no *Data Object* deverá ser associada a este elemento. O serviço *searchFlight* possui dois argumentos, o aeroporto destino e o aeroporto origem da viagem que foram obtidos da interface da composição e representadas no *DataObject* "*General Information*". Para fazer a associação entre essas variáveis, o especialista de serviço, utilizando a visão de serviço, seleciona as variáveis que irão ser associadas, origem e destino, e adiciona a atividade, assim

como mostra Figura 4.6. Nela o usuário associa o conteúdo da variável *origin* representado no *Data Object* "*General Information*" ao parâmetro de entrada *from* do serviço *searchFlight*.

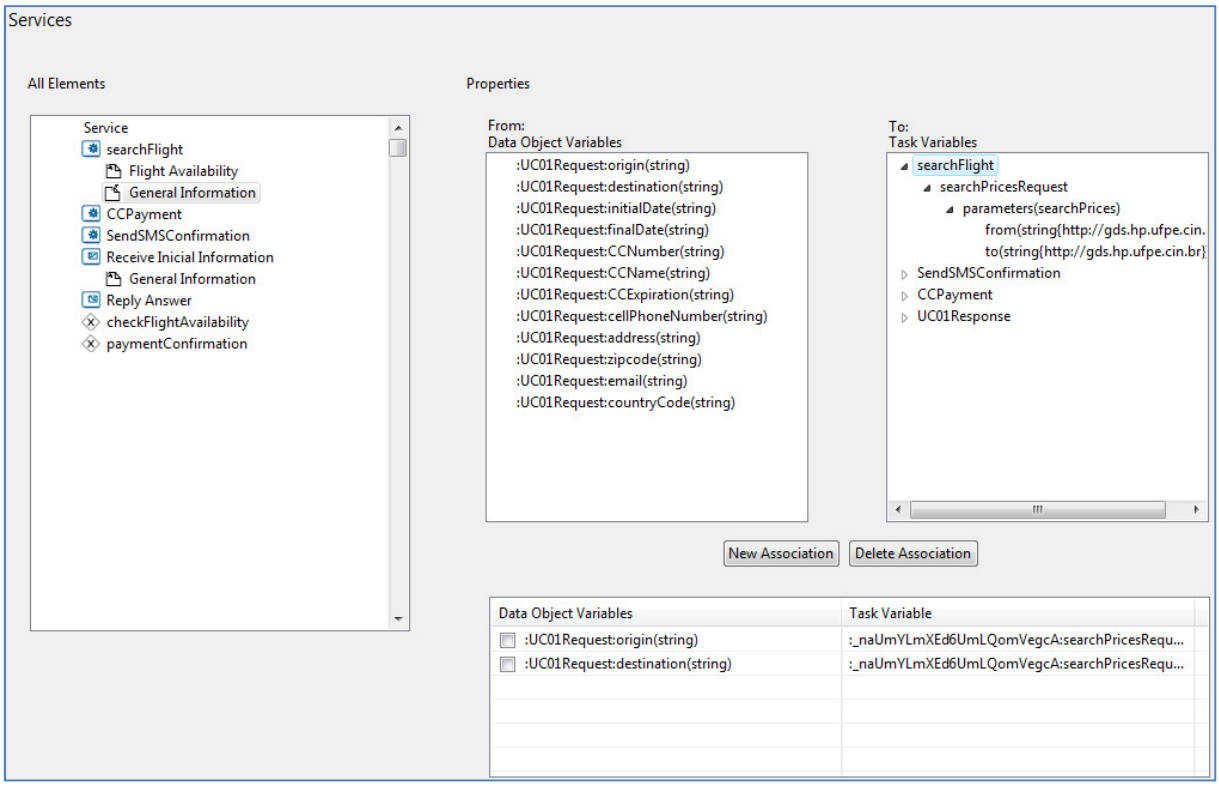


Figura 4.6 - Associação de Variáveis

Nesta associação entre variáveis, o editor gera anotações no arquivo BPMN, conforme Listagem 4.4.

Listagem 4.4 - BPMN Anotado com as Configurações da Associação Entre Variáveis

```
1 . . .
2 <vertices xmi:type="bpmn:Activity" name="searchFlight" ...>
3    <eAnnotations xmi:type="ecore:EAnnotation" source="assign"
  ...>
4      <details key=":_naUmYLmXEd6UmLQomVegcA:searchPricesRequest:
  parameters:from(string{http://www.w3.org/2001/XMLSchema})"
  value=":UC01Request:origin(string)" .../>
5      <details key=":_naUmYLmXEd6UmLQomVegcA:searchPricesRequest:
  parameters:to(string{http://www.w3.org/2001/XMLSchema})"
                        value=":UC01Request:destination(string)"
  .../>
6    </eAnnotations>
7     ...
8 </vertices>
9 ...
```

A Linha 4 refere-se a associação do valor da variável *origin* da composição à variável *from* da operação *searchPricesRequest* do serviço *searchFlight*. A Linha 5 associa o valor da variável *destination* da interface da composição à variável *to* da operação *searchPricesRequest*.

Além das chamadas de serviços, o fluxo da composição também deve ser configurado. O fluxo da composição é controlado através dos elementos Gateways de BPMN. O estudo de caso foi modelado com dois gateways, *checkFlightAvailability* e *paymentConfirmation*, onde o fluxo continuará de forma normal caso haja disponibilidade de vôo e o pagamento seja confirmado, respectivamente. Para poder ter acesso as variáveis contidas nos *Data Objects*, os *gateways* também deverão ser associados a esses elementos. Com a associação efetuada, *gateways* poderão ter acessos às variáveis contidas nos *Data Objects* e com isso fazer operações relacionadas as a variável e uma constante.

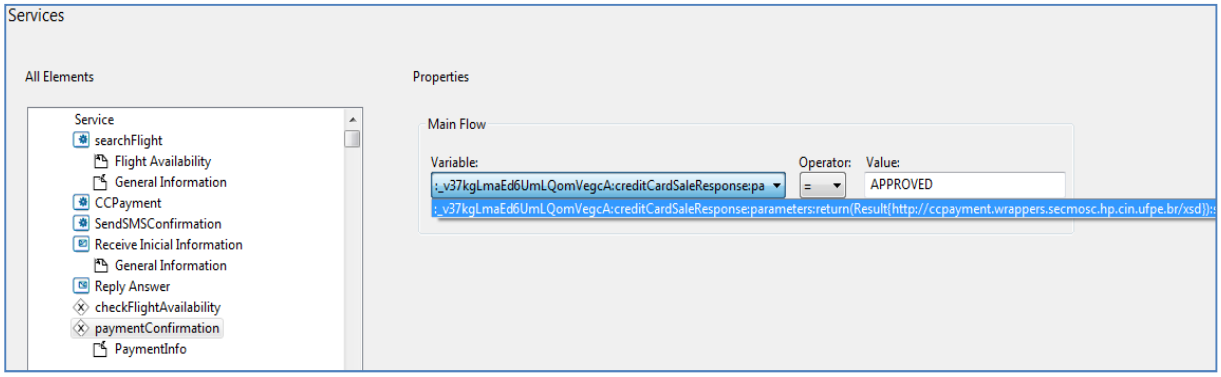


Figura 4.7 - Configuração de *Gateway*

A Figura 4.7 mostra o *gateway paymentConfirmation* sendo configurado pelo especialista de serviço. Nele, o especialista selecionou a variável de retorno denominada *return* da operação *creditCardSale* e comparou com a *string* "*APPROVED*" através do operador igual (=). Desta forma, caso o retorno da operação seja "*APPROVED*" o fluxo da composição segue o caminho principal e caso contrário, o caminho a ser seguido na composição será o representado pelo elemento de seqüência com a propriedade *dafault* associado a este *gateway*, no caso ao final da execução da composição.

Assim como as outras configurações, a configuração do *gateway* também modifica o arquivo BPMN, inserindo as anotações que serão utilizadas na geração do WS-BPEL da composição (Listagem 4.5).

Listagem 4.5 - BPMN Anotado com as Configurações do G*atwway paymentConfirmation*

```
1 . . .
2 <vertices xmi:type="bpmn:Activity"  name="paymentConfirmation"
              activityType="GatewayDataBasedExclusive" ...>
3       <eAnnotations xmi:type="ecore:EAnnotation"
  xmi:id="_dlg94MMuEd6RN4F9kLuLwA" source="condition">
4         <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_dlg94cMuEd6RN4F9kLuLwA" key="variable"
  value=":_v37kgLmaEd6UmLQomVegcA:creditCardSaleResponse:parameters:
  return(Result{http://ccpayment.wrappers.secmosc.hp.cin.ufpe.br/xsd
  }):status(string{http://www.w3.org/2001/XMLSchema})"/>
5         <details xmi:type="ecore:EStringToStringMapEntry"
           xmi:id="_dvziUMMuEd6RN4F9kLuLwA" key="operator"
  value="="/>
6         <details xmi:type="ecore:EStringToStringMapEntry"
  xmi:id="_d-_04MMuEd6RN4F9kLuLwA" key="value" value="APPROVED"/>
7       </eAnnotations>
8 </vertices>
9 ...
```

Nesta listagem, a Linha 4 informa qual a variável que será utilizada na operação, neste caso a variável *return* da resposta da operação *creditCardSale*. A Linha 5 informa o operador utilizado na operação, que para este *gateway* foi utilizado o operador de igual (=). E por fim, na linha 6, a constante "*APPROVED*" que será comparada a variável selecionada.

4.3 Execução

Após a composição ser modelada e configurada adequadamente pelos especialistas, ela deverá ser executada na *engine* de orquestração localizada na nuvem com base no que encontra-se anotado no arquivo BPMN. O caso de uso "*Comprar passagem aérea nacional"* foi executado na infra-estrutura da SSC4Cloud, localizada na nuvem da Open Cirrus, através da *engine Apache ODE* (APACHE ODE).

Quando o usuário do editor clica no botão para iniciar o processo de negócio (Figura 3.31), o respectivo serviço da SSC4Cloud será chamado passando o arquivo BPMN anotado que será recebido e posteriormente traduzido para artefatos independentes de plataforma através do módulo de tradução, artefatos referentes as *NF-Actions* utilizadas na composição e ao processo de negócio e serviço.

A atividade "*Receive Inicial Information"* possui o requisito de confidencialidade que será implementado através do uso da *NF-Action RestrictAccess*, conforme Figura 4.2. A

Listagem 4.6 mostra o artefato independente de plataforma gerado para esta *NF-Action* com base no BPMN anotado gerado pelo SSC4Cloud Editor.

Listagem 4.6 - Configuração Genérica da *NF-Action RestrictAccess*

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <nf-requirements xmlns="http://www.cin.ufpe.br/sec-mosc/nf-
   requirements">
3       <nf-statements>
4         <nf-statement name="Customize"
             taskID="_0celZdllEd6se5R3-QLMYg">
5            <nf-actions>
6               <nf-action name="RestrictAccess"
                 nf-attribute="Security\Confidentiality">
7                  <nf-properties>
8                     <nf-property name="policyType"
                          value="allow" />
9                     <nf-property name="IP"
                          value="209.93.4.0 - 217.114.221.0" />
10                 </nf-properties>
11              </nf-action>
12           </nf-actions>
13        </nf-statement>
14      <!-- Addtional NFStatements -->
15      </nf-statements>
16 </nf-requirements>
```

- As Linhas 4-13 referem-se ao *NF-Statement* que a *NF-Action RestrictAccess* está associada;
- As Linhas 5-12 possuem o conjunto de *NF-Actions* associados ao *NF-Statement*;
- As Linhas 6-11 representam a *NF-Action RestrictAccess*. A Linha 6 contem o nome da *NF-Action* e a qual *NF-Attribute* está associada;
- As propriedades referentes a *NF-Action* encontram-se nas Linhas 7-10. A Linha 8 representa a propriedade *policyType* e o valor associado a ela, no caso *allow* e na linha 9 os IP's que terão acesso à composição.

Este arquivo genérico de segurança, além da *NF-Action RestrictAccess*, também conterá os dados referentes a *UseAuthentication* e *UseCriptography*, ambos referentes ao *NF-Attribute Confidentiality* associado as atividades *searchFligth*, *CCPayment* e *SendSMSConfirmation*.

Para o processo de negócio, o módulo de tradução gera dois artefatos independentes de plataforma. Esses artefatos serão utilizados como insumos para geração da WSDL (Anexo I) e da WS-BPEL (Anexo II) da composição pela *Engine* Auxiliar.

4.4 Considerações Finais

Este capítulo apresentou um estudo de caso para validar a proposta deste trabalho. Inicialmente foi apresentado e detalhado o caso de uso de compra de passagem aérea nacional. Em seguida foram apresentados a realização e a sua execução no ambiente da nuvem.

5 TRABALHOS RELACIONADOS

Este capítulo apresenta e analisa os principais trabalhos ao que está sendo proposto nesta dissertação.

5.1 Especificação de Requisitos de Segurança no Processo de Negócio

Atualmente, diversas pesquisas vêm sendo desenvolvidas relacionadas à modelagem de requisitos não funcionais, principalmente segurança, em processos de negócios. Muitas ferramentas vêm adotando o padrão BPMN (*Business Process Model Notation*) (OMG, 2009) como padrão para especificar processos de negócios.

Rodríguez (RODRÍGUEZ, FERNÁNDEZ-MEDINA e PIATTINI, 2007) propôs uma extensão BPMN para incorporar requisitos de segurança em diagramas de negócio em nível de modelagem. Sua abordagem utiliza o símbolo de cadeado contendo uma letra maiúscula no centro para representar graficamente os requisitos de segurança (Não-Repúdio, Detecção de Ataques Maliciosos, Integridade, Privacidade e Controle de Acesso) nos elementos do BPMN, conforme Tabela 5.1.

Tabela 5.1 - Notação de Segurança Proposto por Rodríguez (RODRÍGUEZ, FERNÁNDEZ-MEDINA e PIATTINI, 2007)

Requisito	Notação
Requisito de Segurança	
Não-Repúdio	
Detecção de Ataques Maliciosos	
Integridade	
Privacidade	
Controle de Acesso	

No caso de Integridade e Privacidade, Rodríguez incluiu a letra x para representar o nível de segurança desses requisitos, que pode ser representado por **I** para baixo (*low*), **m** para médio (*medium*) e **h** para alto (*high*). Comparando com a proposta desta dissertação, os cadeados seriam comparáveis com os *NF-Attributes* e o nível de segurança, simbolizados pela letra x, seriam os *NF-Statements*. Uma importante contribuição de Rodríguez é a pesquisa de como representar graficamente requisitos de segurança em Processos de Negócios, fazendo com que especialistas de negócios possam integrar sua visão sobre segurança do negócio nas suas modelagens.

Assim como Rodríguez, Menzel (MENZEL, THOMAS e MEINEL, 2009) propõe uma abordagem para representar graficamente requisitos de segurança no nível de modelagem com uso do elemento cadeado com representação de nível de segurança (Tabela 5.2). O principal objetivo da proposta de Menzel é facilitar a geração de configuração de segurança baseado no que foi definido no processo de negócio.

Tabela 5.2 - Escala de Segurança

Notação	Classificação	Descrição
	Extremo	Põe em perigo a vida humana ou ameaçando a empresa
	Muito Alto	Graves conseqüências financeiras ou de segurança
	Alto	Impacto nos serviços e reputação do cliente
	Médio	Afeta a missão da empresa
	Baixo	Mínimos prejuízos financeiros e pouco impactos no negócio
	Insignificante	Sem relevância de segurança

Menzel também representa graficamente o grau de confiança entres os parceiros de uma composição de *Web services* através de um elemento gráfico obedecendo a seguinte classificação:

Tabela 5.3 - Classificação de Confiança

Tipo	***Confiança***	***Conhecido por***
Operacional	Absoluta	Mesma organização

B2B	Alta	Contrato
B2C	Baixa	Reputação
WebPresence	Nenhum	Desconhecido

Um ponto muito importante da abordagem de Menzel que se destaca em relação a anterior é a preocupação em não só representar os requisitos de segurança no BPMN, mas também em traduzir estes requisitos de segurança em algo concreto, implementável. Para isto, Menzel transforma os requisitos de segurança anotado no BPMN em especificações, tais como SSL, *WS-Trust* (OASIS, 2007b) e WS-Security (OASIS, 2006), utilizando padrões de segurança como possibilidade para determinar o protocolo e os mecanismos de segurança apropriados para cada abordagem.

Em seu recente trabalho (MENZEL, WARSCHOFSKY e MEINEL, 2010), o autor também propôs a idéia de "*profiles*" que se assemelha com a idéia de *NF-Statement* apresentada nesta dissertação. Os *profiles* são utilizados para expressar segurança no processo pelos analistas de negócios sem que haja uma preocupação com detalhes técnicos. Como exemplo, a Tabela 5.4 possui dois *profiles*: *High* e *Low*.

Tabela 5.4 - Exemplo de P*rofile* do Trabalho de Menzel

Profile	***Security Mechanisms***
High	X509-Token
Low	UserName-Token, X509-Token

A abordagem proposta por Menzel (MENZEL, THOMAS e MEINEL, 2009) possui alguns pontos em comum com a metodologia SSC4Cloud, como a definição de abstrações de segurança, a extensão da notação gráfica para o BPMN e a geração de arquivos de configuração para execução em uma engine de orquestração. Porém, um diferencial da *SSC4Cloud Editor* é a liberdade que o usuário tem de criar e configurar seus próprios "*profiles*" de segurança e de compartilhar conhecimentos entre si através dos *profiles* e dos modelos de processo de negócio criados por eles.

Na proposta de Menzel, no entanto, não fica claro como os requisitos de segurança serão de fato implementados, pois, embora ele utilize padrões de segurança, na prática, diferentes analistas podem ter diferentes interpretações do que seria, por exemplo, uma grave conseqüência financeira para sua empresa.

5.2 Editor ORYX

Oryx (ORYX) é um ambiente de modelagem baseado na Web que permite criação e compartilhamento de diagramas em diferentes linguagens e notações, inclusive BPMN nas versões 1.2 e 2.0. Essa ferramenta provê autenticação de usuário através de *OpenID* (OPENID) e por ser Web, não há a necessidade de instalação de nenhum aplicativo na máquina local do usuário. Todos os diagramas são salvos na base de dados do Oryx e caso seja de interesse do usuário, ele possibilita a exportação de diagramas para diferentes formatos, como exemplo, WS-BPEL, que poderá ser executado posteriormente em alguma engine de execução.

Assim como o SSC4Cloud Editor, o Oryx provê uma palheta contendo todos os elementos BPMN para facilitar a modelagem por parte dos usuários de forma mais intuitiva, conforme Figura 5.1. Porém, ele não possui visões para configuração de serviços web assim como também não possui abstrações nem mecanismos para representar requisitos não funcionais no diagrama.

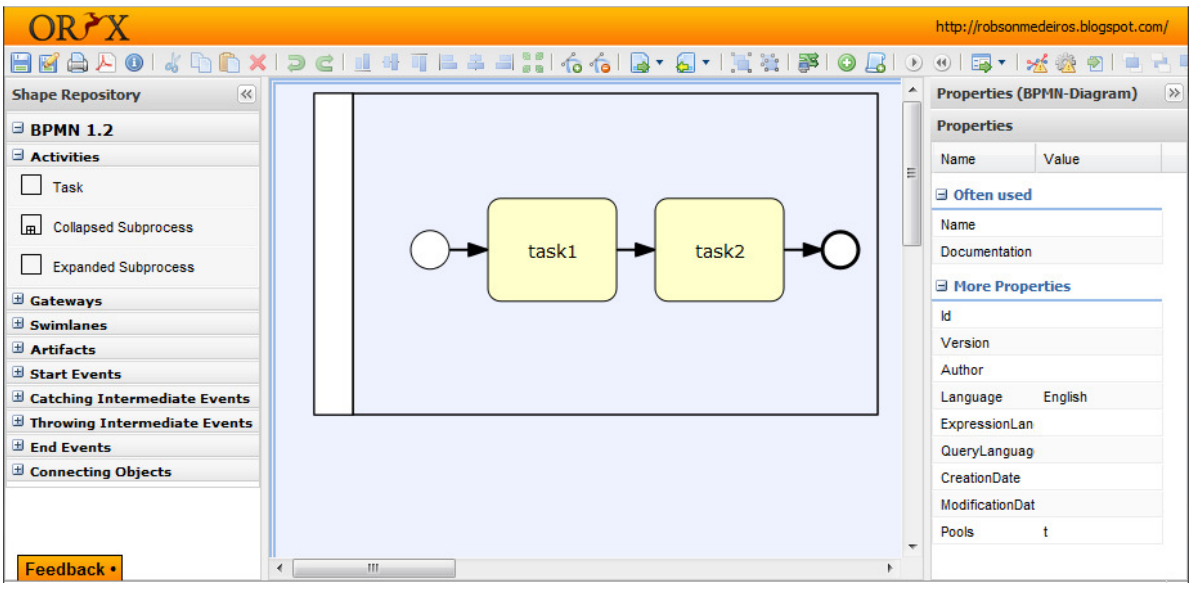


Figura 5.1 - Tela do Editor de BPMN do Oryx

Atualmente existe um grupo de pesquisa da *Hasso Plattner Institut* (HPI) (HASSO-PLATTNER-INSTITUT, 2011) desenvolvendo uma pesquisa para criar uma *engine* de orquestração para executar na nuvem os processos de negócio modelados no Oryx, denominada *Oryx Engine* (ORYX, 2010). Porém, como dito, este projeto ainda encontra-se em fase de pesquisa.

O grande diferencial do Oryx em relação ao SSC4Cloud Editor é sua plataforma web e quantidade de linguagens de modelagem suportadas. Assim como no trabalho dessa dissertação, ela possibilita compartilhamento de processos de negócios entre seus usuários e, pelo fato de ser Web, as modificações feitas por um usuário são percebidas pelos demais usuários que visualizam o mesmo artefato no mesmo instante e com compartilhamento não só de leitura, mas também de escrita.

Porém, diferentemente do SSC4Cloud, o Oryx é uma ferramenta apenas de modelagem. O único suporte à execução fornecido pela ferramenta é a possibilidade de exportação de diagramas feitos em BPMN para WS-BPEL.

5.3 eClarus Business Process Modeler for SOA Architects

Assim como o SSC4Cloud Editor, o *eClarus Business Process Modeler for SOA Architects* (ECLARUS SOFTWARE) é um editor desenvolvido na plataforma Eclipse para modelagem de processos de negócios em BPMN. Por ser desenvolvido na plataforma Eclipse, o eClarus é facilmente integrado com outras ferramentas de desenvolvimento através de *Plug-in* (Figura 5.2). Além disso, ele provê geração de código executável em WS-BPEL que pode ser executado em diferentes *engines* de orquestração, como Apache ODE. O eClarus também possui um simulador que permite analistas de negócios avaliarem o impacto que novos processos ou modificações em processos existentes podem causar.

Em comparação com a proposta desse trabalho, o eClarus também provê um ambiente de modelagem e configuração de processos de negócios utilizando BPMN com configurações de serviço e geração de artefatos executáveis para *engine* de orquestração. Um grande diferencial do eClaru é a sua funcionalidade de simulação, que fornece informações, com base em cenários hipotéticos, para que analistas de negócios possam encontrar uma ótima condição para a execução do seu processo de negócio.

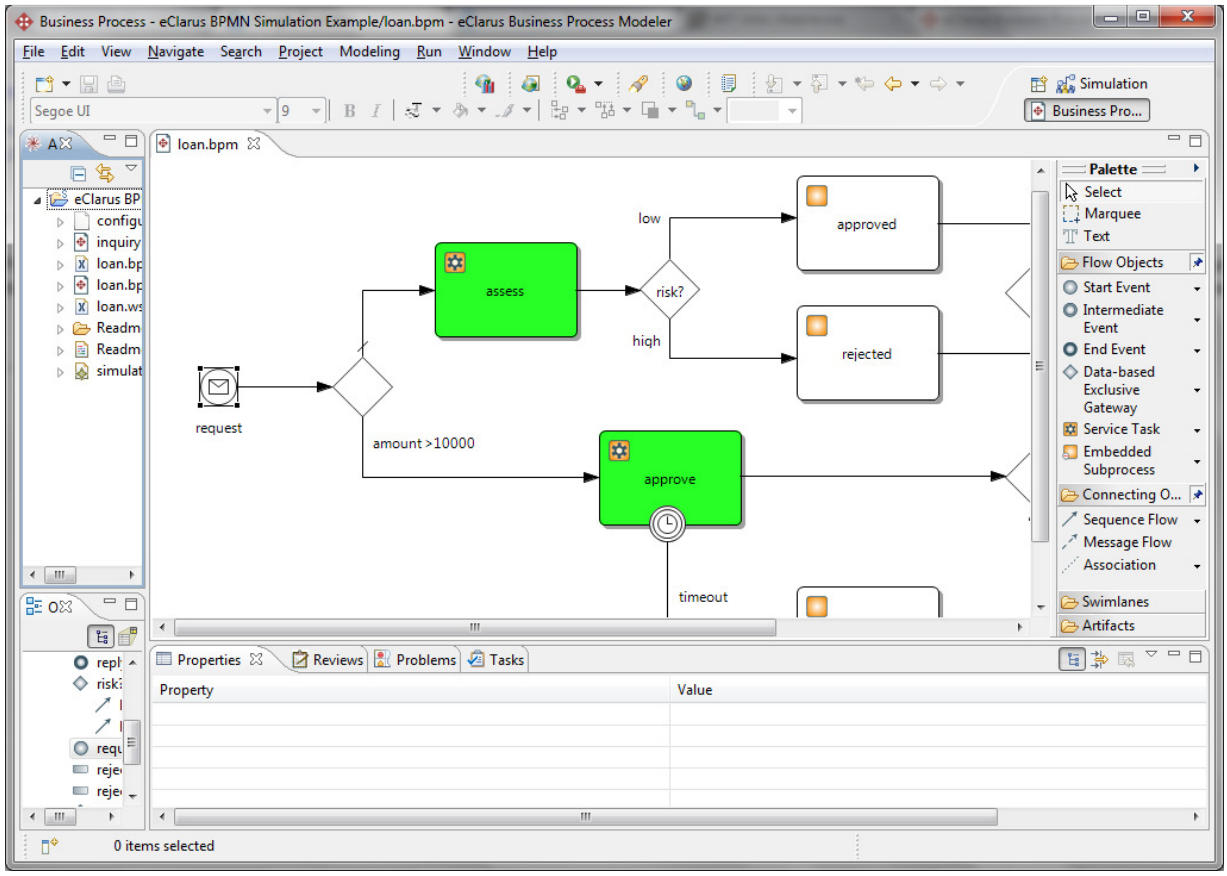


Figura 5.2 - Tela do Editor eClarus

No entanto, esta ferramenta não provê meios para compartilhamento de processos de negócio. O único meio de compartilhamento é através de troca de artefatos de forma direta, fora da ferramenta. Também não é possível anotar graficamente na modelagem informações sobre requisitos de segurança exigidos na composição e muito menos fornecer recursos para que a execução da composição seja feita de forma segura. No eClarus, informações do processo de negócio e de serviços encontram-se na mesma visão e, normalmente, analistas de negócio não tem conhecimento suficiente para configurações referentes a execução do processo, diferente da proposta dessa dissertação que provê visões para cada especialidade (negócio, segurança e serviço).

5.4 Oracle SOA Suite 11g

O SOA *Suite* 11g (ORACLE) possui um conjunto de componentes de infra-estrutura que possibilita a modelagem, orquestração e gerenciamento de aplicações baseadas em SOA. Como principais características, essa solução permite composição de serviços através de troca

de mensagem, descoberta de serviços, orquestração de composição, gerenciamento e segurança de *Web Services*, criação de regras de negócio e monitoramento das atividades de negócio com possibilidade de disparar eventos conforme regras estabelecidas.

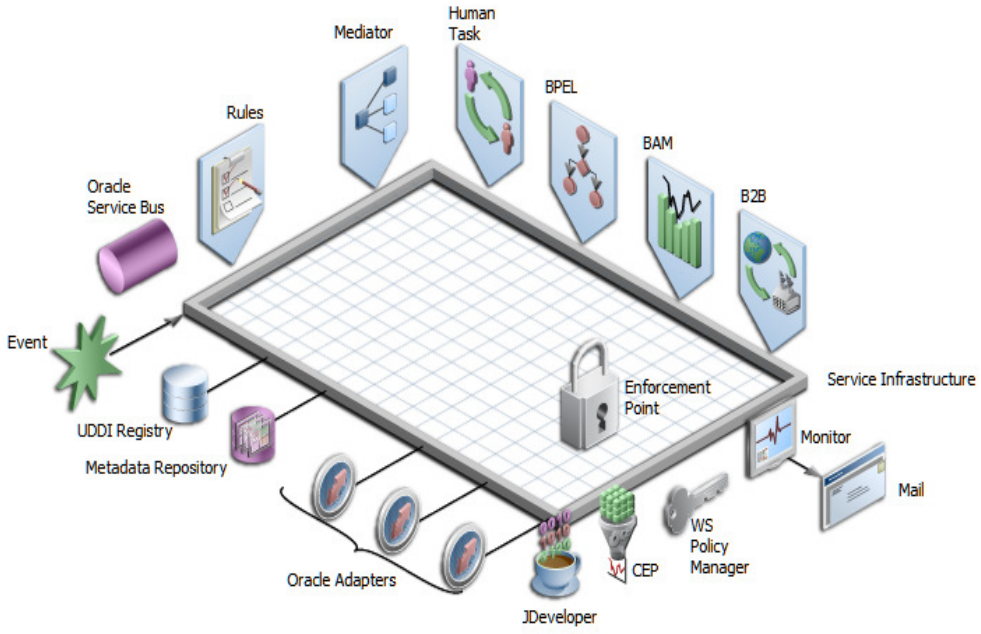


Figura 5.3 - Visão Geral dos Componentes do SOA *Suite*

A Figura 5.3 mostra os componentes do SOA *Suite*. O JDeveloper (Figura 5.4) é o componente do SOA *Suite* responsável pela modelagem de processos através de BPEL e BPMN e que se assemelha à ferramenta proposta nessa dissertação. Nele, analistas de negócios podem criar, descobrir, testar, implantar e manter seus aplicativos compostos baseados em serviços. A ferramenta também possui um simulador para que analistas possam analisar comportamentos da composição com base em situações hipotéticas.

Com a composição criada, ela poderá ser implantada no *BPEL Process Manager* que irá executar a composição conforme modelado, inclusive na nuvem utilizando a infra-estrutura da Amazon EC2 (PALVANKAR, 2010). Com o Oracle SOA *Suite* também é possível reforçar requisitos de segurança e auditoria nas composições através do componente Oracle WSM *Policy Manager*. Ele fornece mecanismos para assinatura de mensagem, criptografia, autenticação, e prover controle de acesso na comunicação da composição. Porém, essa configuração não é feita através ferramenta JDeveloper, mas sim no Oracle Enterprise Manager que é um ambiente de gerenciamento de aplicações do SOA *Suite*.

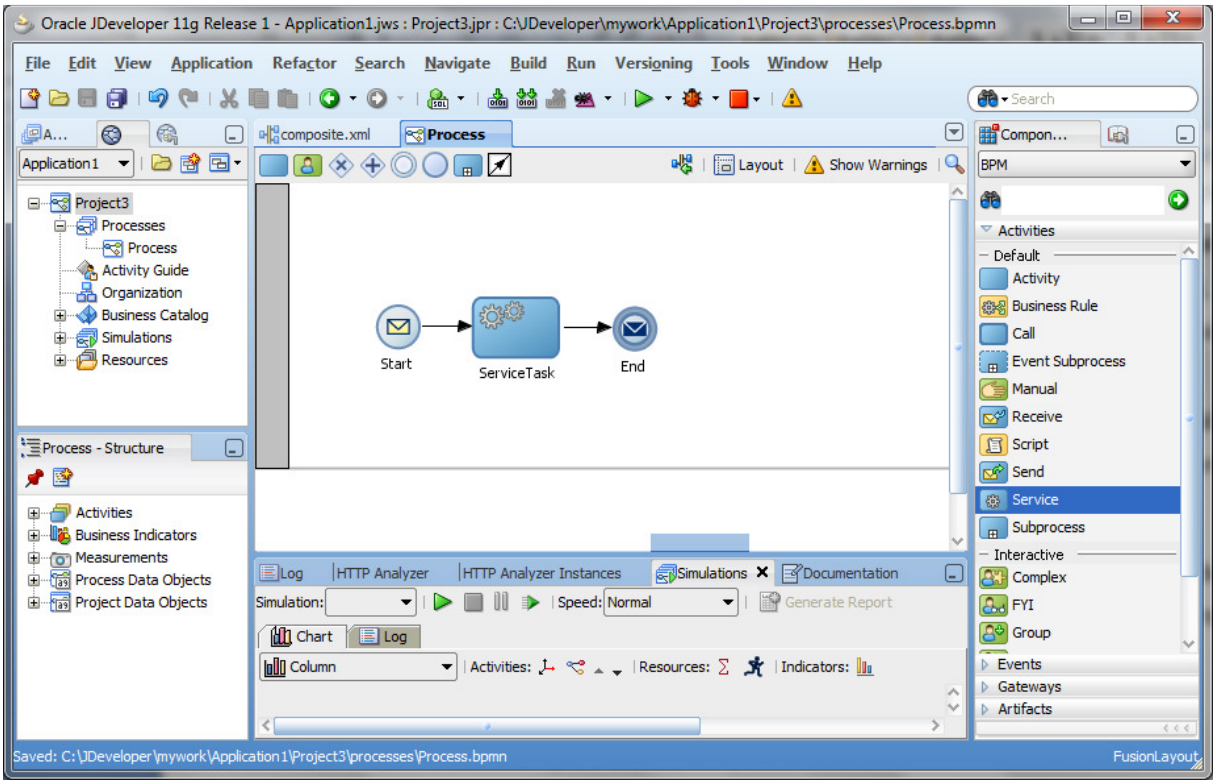


Figura 5.4 - Tela do JDeveloper

Como pôde ser visto, o SOA *Suite* se destaca pelo número de componentes e, conseqüentemente, funcionalidades que ele possui, desde a modelagem, com possibilidade de simulação, até a execução, com monitoramento, segurança e auditoria. Porém, na modelagem, através do JDeveloper, especialistas de negócio não possuem mecanismos para expressar requisitos de segurança. Além disso, configurações referentes aos serviços são feitas na mesma visão da modelagem, diferentemente do SSC4Cloud Editor que provê diferentes visões para que analistas de negócios só se preocupe com o a modelagem do negócio.

5.5 Comparação

Neste capítulo foram apresentados alguns trabalhos relacionados à modelagem de processos de negócios e anotação de segurança em BPMN. Para apresentar uma comparação mais clara, a Tabela 5.5 resume a comparação entre os trabalhos relacionados e esta dissertação.

Tabela 5.5 - Comparação Entre os Trabalhos Relacionados

	Suporte a Requisitos de Segurança	Presença de Visões para Criação e Configuração de Composição por Especialidade (Negócio, Serviço e Segurança)	Integração com Ambientes de Nuvem	Compartilhamento de Artefatos no Nível de Modelagem
Rodríguez (RODRÍGUEZ, FERNÁNDEZ-MEDINA e PIATTINI, 2007)	Sim	Não	Não	Não
Menzel (MENZEL, THOMAS e MEINEL, 2009)	Sim	Não	Não	Não
Oryx (ORYX)	Não	Não	Não	Sim
eClarus (ECLARUS SOFTWARE)	Não	Não	Não	Não
SOA *Suite* 11 g (ORACLE)	Não	Não	Sim	Sim
SSC4Cloud Editor (trabalho desta dissertação)	Sim	Sim	Sim	Sim

Com isso, pode-se observar que apenas Menzel (MENZEL, THOMAS e MEINEL, 2009) e Rodríguez (RODRÍGUEZ, FERNÁNDEZ-MEDINA e PIATTINI, 2007), além da SSC4Cloud Editor, representam requisitos de segurança no nível de modelagem, possibilitando que analistas de negócios consigam expressar a segurança da composição sem preocupação com a implementação desses requisitos. Embora Rodríguez tenha se preocupado com a modelagem, ele não especifica como esses requisitos serão transformados em configurações executáveis em *engines* de orquestração.

Embora os trabalhos eClarus, Oryx e SOA permitam a transformação de processo de negócio em linguagem executável, eles não suportam a especificação de segurança no nível de modelagem. Apenas o SOA *Suite* tem mecanismos para configurações desses requisitos.

Em relação ao compartilhamento de artefatos, os trabalhos Oryx e SOA *Suite* possuem mecanismos de compartilhamento de processos de negócios modelados em seus editores assim como o SSC4Cloud Editor. Porém, apenas o SSC4Cloud Editor possui a funcionalidade de compartilhamento de conhecimento de segurança através dos seus *profiles* e integrados com um ambiente de nuvem.

Nenhum dos trabalhos relacionados citados nesta seção possui visões para criação de processos de negócio e configuração de requisitos de segurança e serviços.

Com base à integração com ambientes de nuvem, atualmente apenas o SSC4Cloud Editor, através da infra-estrutura SSC4Cloud, e o SOA *Suite* possuem um ambiente de execução localizado na nuvem. O Oryx possui um projeto para desenvolvimento de uma engine de orquestração para executar neste ambiente, denominada Oryx *Engine*, mas no momento encontra-se em fase de pesquisa.

Portanto, observa-se que existem trabalhos que provêm possibilidade de anotar segurança em processos de negócios, trabalhos que executam composição de serviço na nuvem e que compartinham processos de negócio entre usuários, porém nenhum desses trabalhos tem todas essas funcionalidades integradas. O SSC4Cloud Editor provê um ambiente integrado, com todas essas funcionalidades, onde *stakeholders* podem trabalhar de forma colaborativa para modelar processos de negócio com anotações de segurança provenientes de configurações predefinidas através de *profiles* de segurança e executá-los em um ambiente na nuvem que garante a implementação dos requisitos de segurança anotados. Além disso, ele também possibilita que usuários compartilhem conhecimento, tanto de modelagem de processo de negócio, como o SOA Suite da Oracle faz, mas também de segurança através dos *profiles*.

5.6 Considerações Finais

Este capítulo apresentou os principais trabalhos relacionados ao tema desta dissertação, trabalhos esses voltados à modelagem de processos de negócio, requisitos de

segurança, transformação de modelo de processo de negócio em linguagem executável e compartilhamento de artefatos de modelagem e segurança. Em seguida foi apresentada uma comparação desses trabalhos e a proposta dessa dissertação, onde foi apresentada uma tabela comparativa.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo apresenta as conclusões, as principais contribuições deste trabalho, suas limitações e os trabalhos futuros que poderão ser realizados dentro desta linha de pesquisa.

6.1 Conclusões

A infra-estrutura de suporte à composição de serviços (e.g., as *engines* de orquestração), as próprias questões relacionadas à composição de serviços (e.g., linguagens de composição, estratégias de composição) e o gerenciamento destas composições têm recebido muita atenção na computação. Mais recentemente, a necessidade de associação de requisitos não-funcionais à composição e a sua realização através de mecanismos de segurança reais (providos pela infra-estrutura) também tem sido objeto de diversas pesquisas.

Neste cenário, uma evolução importante que está em curso é a execução das composições de serviços em ambientes de nuvem. A possibilidade de compartilhamento de conhecimento de processos de negócio e requisitos não-funcionais, associado as facilidades providas pelas nuvens são a origem deste crescente interesse de associação de composição e nuvem.

A partir deste cenário, foi proposto a SSC4Cloud Editor, um editor de processo de negócio em BPMN com suporte a requisitos de segurança. Vale salientar que este editor está inserido em num contexto onde há necessidade de tratar requisitos de segurança desde a modelagem até sua execução em um ambiente localizado na computação em nuvem. A proposta do SSC4Cloud Editor visa prover um ambiente de modelagem de processos de negócios com anotações de segurança onde analistas de negócios, especialistas em serviços e em segurança possam criar composições de serviços de forma colaborativa e independente e que sejam de fato executadas em uma engine de orquestração. Além disso, o SSC4Cloud Editor também provê um ambiente onde usuários, sejam eles analistas de negócio ou especialistas em segurança, possam compartilhar seus conhecimentos nas suas áreas com outros usuários para promoverem um maior reuso de processos de negócios e de configurações de segurança.

A proposta foi implementada utilizando a plataforma Eclipse e a linguagem de programação Java e validada através do estudo de caso de um portal de uma agência de viagem.

6.2 Contribuições

As principais contribuições deste trabalho estão relacionadas à modelagem de processos de negócio com anotação de segurança em um ambiente colaborativo e ao compartilhamento de conhecimento, tanto de processo de negócio como de segurança, através da nuvem. Anotação de requisitos de segurança em nível de modelagem de processo de negócios com suporte a execução na computação em nuvem faz com que especialistas de negócios consigam expressar na modelagem todo seu conhecimento sobre a segurança exigida na composição e no relacionamento entre os seus parceiros e executem seus processos em um ambiente elástico, com possibilidade de aumentar ou diminuir os recursos computacionais conforme sua demanda. A configuração de requisitos funcionais e não-funcionais de forma independente e interativa, através de visões específicas para cada especialista, propicia aos usuários um ambiente mais independente e fácil para trabalhar.

A configuração personalizada de requisitos de segurança em forma de *profiles* com base nas abstrações *NF-Statement*, *NF-Attribute* e *NF-Action* é outra grande contribuição deste trabalho. Com base nisso, especialistas de segurança podem criar seus *profiles* personalizados, para uso geral ou específico, que podem ser reutilizados em diferentes composições e compartilhados entre diversos usuários.

6.3 Limitações

Por questões de escopo e de objetivos da dissertação, alguns aspectos não foram abordados nesse trabalho:

- Apenas o requisito não-funcional segurança é suportado por este trabalho. Outros requisitos, como por exemplo, desempenho, também são importantes para o bom funcionamento de composições de serviços;

- O SSC4Cloud Editor não possui um ambiente de monitoramento para detecção de violação dos requisitos de segurança e análise;
- Este trabalho não provê colaboração em tempo real e de modelagem, onde usuários com acesso ao mesmo processo de negócio possam trabalhar e discutir juntos o modelo mais adequado para o negócio usando BPMN e as abstrações de segurança providas pelo editor;
- O SSC4Cloud Editor não possui um ambiente de simulação para que analistas possam simular o comportamento da composição antes de sua execução;
- Atualmente o editor funciona localmente, fora da nuvem e com isso não aproveita os benefícios oferecidos por esse ambiente, diferentimento do restante da solução SSC4Cloud que encontra-se na nuvem; e
- O SSC4Cloud Editor não trata da segurança individual dos dados. As anotações de segurança referem-se às invocações dos serviços e consequentemente a todos os dados utilizados nesta invocação.

6.4 Trabalhos Futuros

Com base neste trabalho, diversos outros podem ser desenvolvidos. Inicialmente, pretende-se incluir mais requisitos não-funcionais, como desempenho e evolução e verificar se as abstrações hoje existentes são suficientes para representá-los e propor novas, caso necessário.

Para tornar a solução mais robusta, será necessário incluí-la no ambiente da computação em nuvem. Mas, para isso, é necessário a mudança de sua plataforma do Eclipse para Web. Com esta modificação vislumbra-se deixar a colaboração da modelagem em tempo real, onde mais de um analista poderá, ao mesmo tempo, modelar um processo de negócio para deixá-lo o mais adequado possível ao negócio, como também na criação de *profiles* de segurança.

Na proposta atual, após o processo de negócio ser executado não é possível visualizar sua execução, seguindo-a passo a passo e interagir com ele, por exemplo, fornecendo informações em tempo de execução em tarefas do tipo usuário, tarefas que necessitam da interação humana.

Também se vislumbra incluir a funcionalidade para simular a execução do processo de negócio, de forma que analistas possam verificar e adaptar o processo de negócio para um modelo mais adequado conforme dados hipotéticos. Além disso, pretende-se criar um ambiente de monitoramento dos requisitos não funcionais para detectar violações dos requisitos especificados na modelagem, hoje não presente na solução SSC4Cloud.

Finalmente, está sendo planejada também uma avaliação de usabilidade para detecção de falhas e melhorias do SSC4Cloud Editor. É vislumbrado provar que tal abordagem não adiciona uma maior complexidade na elaboração de processos de negócios, fazendo com que o tempo gasto na modelagem seja muito elevado, assim como a da configuração da composição de serviços e dos requisitos de segurança.

REFERÊNCIAS

AMAZON. Amazon Elastic Compute Cloud (Amazon EC2). **Amazon Web Services**, a. Disponivel em: <http://aws.amazon.com/ec2/>. Acesso em: 25 fev. 2010.

AMAZON. Amazon Simple Storage Service (Amazon S3). **Amazon Web Services**, b. Disponivel em: <http://aws.amazon.com/s3/>. Acesso em: 24 fev. 2010.

APACHE ODE. Apache ODE. Disponivel em: <http://ode.apache.org/>. Acesso em: 22 jan. 2011.

ARMBRUST, M. et al. **Above the Clouds: A Berkeley View of Cloud Computing**. Electrical Engineering and Computer Sciences University of California at Berkeley. [S.l.]. 2009.

ATKINSON, B. et al. Web Services Security (WS-Security). **msdn**, 2002. Disponivel em: <http://msdn.microsoft.com/en-us/library/ms951257.aspx>. Acesso em: 15 maio 2010.

CHONG, R.; CARRARO, G.; WOLTER, R. MSDN Architecture Center. **MSDN Microsoft**, 2006. Disponivel em: <http://msdn.microsoft.com/en-us/architecture/aa479086.aspx>. Acesso em: 23 mar. 2010.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Sistemas Distribuídos:** Conceitos e Projetos. Tradução de J. Tortello. 4. ed. Porto Alegre: Bookman, 2007.

DAMASCENO, J. C. SecMosc-Engine: Um Ambiente de Suporte à Execução Segura de Composição de Sserviços. **Dissertação de Mestrado, Universidade Federal de Pernambuco**, 2010.

DAMASCENO, J. C. et al. Towards Generating Richer Code by Binding Security Abstractions to BPMN Task Types. **Revista de Informática Teórica e Aplicada**, p. 97-98, 2009.

DAMASCENO, J. et al. Modeling and Executing Business Processes with Annotated Security Requirements in the Cloud (Submetido). **IEEE International Conference on Web Service**, Washington DC, EUA, 4 jul. 2011.

ECLARUS SOFTWARE. Disponivel em: <http://www.eclarus.com/>. Acesso em: 12 fev. 2011.

ECLIPSE. BPMN Modeler. **Eclipse**, 2009. Disponivel em: <http://www.eclipse.org/bpmn/>. Acesso em: 28 mar. 2009.

ECLIPSE. Eclipse Platform. **Eclipse**, b. Disponivel em: <http://www.eclipse.org/platform/>. Acesso em: 04 mar. 2009.

EFFINGER, P.; SIEBENHALLER, M.; KAUFMANN, M. An Interactive Layout Tool for BPMN. **Commerce and Enterprise Computing**, 2009. 399-406.

ERL, T. **Service-Oriented Architecture:** Concepts, Technology, and Design. [S.l.]: Prentice Hall PTR, 2005.

FURTADO, C. et al. **Arquitetura Orientada a Serviço - Conceituação**. Universidade Federal do Estado do Rio de Janeiro. Rio de Janeiro. 2009.

GONG, C. et al. The Characteristics of Cloud Computing. **39th International Conference on Parallel Processing Workshops**, 2010. 275-279.

GOOGLE. Press Center, 2006. Disponivel em: <http://www.google.com/press/podium/ses2006.html>. Acesso em: 12 jan. 2011.

GOOGLE. Google App Engine. **Google**. Disponivel em: <http://code.google.com/intl/en/appengine/>. Acesso em: 05 mar. 2010.

GROSSO, W. **Java RMI**. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2001.

HASSO-PLATTNER-INSTITUT. Home. **HPI**, 2011. Disponivel em: <http://www.hpi.uni-potsdam.de/>. Acesso em: 12 jan. 2011.

IBM. Implementing WS-Security, 2003. Disponivel em: <http://www.ibm.com/developerworks/webservices/library/ws-security.html>. Acesso em: 6 mar. 2010.

JOPERA. JOpera. Disponivel em: <http://www.jopera.org/>. Acesso em: 22 abr. 2010.

KUMAR, S. et al. Security Enforcement using PKI in Semantic Web. **Computer Information Systems and Industrial Management Applications**, 2010. 392-397.

LAVARACK, T.; COETZEE, M. Considering web services security policy compatibility. **Information Security for South Africa (ISSA)**, 2010. 1-8.

LOURIDAS, P. SOAP and Web Services. **Software, IEEE**, v. 23, p. 62-67, dez. 2006.

MEDEIROS, R. et al. SSC4Cloud Tooling: An Integrated Environment for the Development of Business Processes with Security Requirements in the Cloud (Submetido). **IEEE World Congress on Services**, Washington DC, USA, 4 jul 2011.

MENZEL, M.; THOMAS, I.; MEINEL, C. Security Requirements Specification in Service-Oriented Business Process Management. **International Conference on Availability, Reliability and Security (ARES 2009)**, Fukuoka, Japan, 2009. 41-48.

MENZEL, M.; WARSCHOFSKY, R.; MEINEL, C. A Pattern-driven Generation of Security Policies for Service-oriented Architectures. **IEEE International Conference on Web Services**, 2010. 243 - 250.

MICROSOFT..NET Framework Overview. **Microsoft**, 2009. Disponivel em: <http://www.microsoft.com/net/overview.aspx>. Acesso em: 18 mar. 2011.

MIHINDUKULASOORIYA, N. Understanding WS – Security Policy Language. **WSO2**, 2008. Disponivel em: <http://wso2.org/library/3132>. Acesso em: 08 jan. 2011.

NETBEANS. NetBeans IDE. Disponivel em: <http://netbeans.org/>. Acesso em: 11 jan. 2011.

NORDBOTTEN, N. A. XML and Web Services Security Standards. **IEEE COMMUNICATIONS SURVEYS & TUTORIALS**, 11, 2009. 4-21.

NURMI, D. et al. The Eucalyptus Open-Source Cloud-Computing System. **9th IEEE International Symposium on Cluster Computing and the Grid**, Shanghai, China, 2009. 124-131.

OASIS. UDDI Spec TC, 2004. Disponivel em: <http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>. Acesso em: 06 jan. 2011.

OASIS. Web Services Security: SOAP Message Security 1.1 , 2006. Disponivel em: <http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>. Acesso em: 10 abr. 2010.

OASIS. Web Services Business Process Execution Language Version 2.0, 2007. Disponivel em: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>. Acesso em: 07 jan. 2011.

OASIS. WS-Trust Version 1.3, 2007b. Disponivel em: <http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html>. Acesso em: 15 out. 2009.

OMG. Business Process Modeling Notation (BPMN), 2009. Disponivel em: <http://www.omg.org/spec/BPMN/1.2/PDF>. Acesso em: 07 jan. 2011.

OMG. CORBA. **OMG**, 2011. Disponivel em: <http://www.corba.org/>. Acesso em: 18 mar. 2011.

OMG. OMG IDL, 2011. Disponivel em: <http://www.omg.org/gettingstarted/omg_idl.htm>. Acesso em: 18 mar 2011.

ONYSZKO, T. Secure Socket Layer. **WindowSecurity.com**, 2002. Disponivel em: <http://www.windowsecurity.com/articles/Secure_Socket_Layer.html>. Acesso em: 09 jan. 2011.

OPEN CIRRUS. Home. **Open Cirrus(TM)**, 2010. Disponivel em: <https://opencirrus.org/>. Acesso em: 15 abr. 2010.

OPENID. **OpenID**. Disponivel em: <http://openid.net/>. Acesso em: 25 jan. 2011.

OPENNEBULA. **OpenNebula**, 2011. Disponivel em: <http://opennebula.org/>. Acesso em: 11 abr. 2011.

OPPLIGER, R. **SSL and TLS:** Theory and Practice (Information Security and Privacy). [S.l.]: Arteck House, 2009.

ORACLE. Java EE. **Oracle**, 2011. Disponivel em: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>. Acesso em: 18 mar. 2011.

ORACLE. Remote Method Invocation Home. **Oracle Technology Network**, 2011. Disponivel em: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html>. Acesso em: 21 jan. 2011.

ORACLE. Oracle SOA Suite. **Oracle**. Disponivel em: <http://www.oracle.com/technetwork/middleware/soasuite/overview/index.html>. Acesso em: 05 abr. 2010.

ORT, E. Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools, 2005. Disponivel em: <http://www.oracle.com/technetwork/java/features-2005-135172.html>. Acesso em: 4 jan. 2011.

ORYX. **Oryx Engine**, 2010. Disponivel em: <http://bpt.hpi.uni-potsdam.de/Oryx/Engine>. Acesso em: 21 fev 2011.

ORYX. The Oryx Project. **ORYX**. Disponivel em: <http://bpt.hpi.uni-potsdam.de/Oryx/>. Acesso em: 12 maio 2010.

PALVANKAR, P. Oracle BPM, 2010. Disponivel em: <http://blogs.oracle.com/bpm/2010/06/bpm_11gr1_now_available_on_ama.html>. Acesso em: 15 mar 2011.

PAPAZOGLOU, M. P.; HEUVEL, W.-J. Service oriented architectures: approaches, technologies and research issues. **VLDB Journal, Springer-Verlag**, 2007.

PENG, J. et al. Comparison of Several Cloud Computing Platforms. **Second International Symposium on Information Science and Engineering**, 2009. 23-27.

POSTEL, J. Internet Protocol. **RFC:** 791, 1981. Disponivel em: <http://www.ietf.org/rfc/rfc791.txt>.

RODRÍGUEZ, A.; FERNÁNDEZ-MEDINA, E.; PIATTINI, M. A BPMN Extension for the modeling of Security Requirements in Business Processes. **IEICE - Trans. Inf. Syst.**, v. E90–D, n. 4, p. 745 - 752, abril 2007.

ROSA, N. S. NFi: An Architecture-based Approach for Treating Non-Functional Properties. **PhD thesis, Federal University of Pernambuco**, 2001.

SILVA, B. L. B. et al. An Approach for Reducing the Gap between BPMN Models and Implementation Artifacts. **Revista de Informática Teórica e Aplicada**, p. 83-86, 2009.

SINGH, A. An Introduction to Virtualization. **Kernelthread**, 2004. Disponivel em: <http://www.kernelthread.com/publications/virtualization/>. Acesso em: 12 jan. 2011.

SOUZA, A. R. S. et al. Incorporating Security Requirements into Service Composition: From Modelling to Execution. **ICSOC/ServiceWave 2009: The 7th International Joint Conference on Service Oriented Computing**, Stockholm, 5900, 2009. 373-388.

SOUZA, A. R. S. et al. Sec-MoSC Tooling - Incorporating Security Requirements into Service Composition. **ICSOC/ServiceWave 2009: The 7th International Joint Conference on Service Oriented Computing**, Stockholm, 5900, 2009. 649-650.

SPRINGSOURCE. **Springsource Community**. Disponivel em: <http://www.springsource.org/>. Acesso em: 09 jun. 2010.

STEPHENSON, B. et al. Deployment Experience Using Open Cirrus for a Cloud-based Business Process Modeling and Execution Environment (Submetido). **Open Cirrus Summit 2011**, Moscow, Russia, 01 jun. 2011.

STOLLBERG, M. et al. D3.2 WSMO Use Case Modeling and Testing. **Web Service Modeling Ontology**, 2004. Disponivel em: <http://www.wsmo.org/2004/d3/d3.2/20041004/>. Acesso em: 20 fev. 2009.

TAURION, C. **Cloud Computing - Computação em Nuvem:** Transformando o mundo da Tecnologia da Informação. [S.l.]: Brasport, 2009.

VINOSKI, S. Corba: Integrating diverse applications within distributed heterogeneous environments. **IEEE Communications Magazine**, v. 35, p. 46–55, fev. 1997.

W3C. Web Services Description Language (WSDL). **W3C**, 2001. Disponivel em: <http://www.w3.org/TR/wsdl>. Acesso em: 18 mar. 2011.

W3C. Web Services Architecture, 2004. Disponivel em: <http://www.w3.org/TR/ws-arch/>. Acesso em: 6 jan. 2011.

W3C. SOAP. **W3C**, 2007. Disponivel em: <http://www.w3.org/TR/soap12-part1/>. Acesso em: 18 mar. 2011.

W3C. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition), 2007. Disponivel em: <http://www.w3.org/TR/soap12-part1/>. Acesso em: 06 jan. 2011.

W3C. Extensible Markup Language (XML). **W3C**, 2008. Disponivel em: <http://www.w3.org/XML/>. Acesso em: 18 mar. 2011.

WANG, L. et al. Cloud Computing: a Perspective Study. **New Generation Computing**, v. 28, p. 137-146, 2010.

WHITE, A. S. Introduction to BPMN. **OMG**, 2004. Disponivel em: <http://www.bpmn.org/Documents/Introduction_to_BPMN.pdf>. Acesso em: 03 maio 2010.

ZENG, L. et al. QoS-Aware Middleware for Web Services Composition. **IEEE Transactions on Software Engineering**, v. 30, p. 311-327, maio 2004.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **J Internet Service Application**, 2010. 7–18.

ZHANG, S. et al. Cloud Computing Research and Development Trend. **Second International Conference on Future Networks**, 2010. 93-97.

ANEXO I: WSDL DA COMPOSIÇÃO DE SERVIÇO DE COMPRA DE PASSAGEM AÉREA NACIONAL

```
<wsdl:definitions
targetNamespace="http://samplePaper.windows.cin.ufpe.br.wsdl"

     xmlns="http://schemas.xmlsoap.org/wsdl/"

     xmlns:tns="http://samplePaper.windows.cin.ufpe.br.wsdl"
xmlns:wns="http://samplePaper.windows.cin.ufpe.br"
     xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3. org/2001/XMLSchema-instance"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">

     <wsdl:types>
          <xsd:schema attributeFormDefault="qualified"
elementFormDefault="qualified"
               xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.77a55c744200ea60.com/process/ServiceProc
ess">
               <xsd:element
name="CompraPassagemAereaNacionalRequestType">
                    <xsd:complexType>

                         <xsd:sequence>
                              <xsd:element name="origin"
type="xsd:string" />
                              <xsd:element name="destination"
type="xsd:string" />
                              <xsd:element name="initialDate"
type="xsd:string" />
                              <xsd:element name="finalDate"
type="xsd:string" />
                              <xsd:element name="CCNumber"
type="xsd:string" />
                              <xsd:element name="CCName"
type="xsd:string" />
                              <xsd:element name="CCExpiration"
type="xsd:string" />
                              <xsd:element name="cellPhoneNumber"
type="xsd:string" />
                              <xsd:element name="address"
type="xsd:string" />

                              <xsd:element name="zipcode"
type="xsd:string" />
                              <xsd:element name="email"
type="xsd:string" />

                              <xsd:element name="countryCode"
type="xsd:string" />
                         </xsd:sequence>
                    </xsd:complexType>
               </xsd:element>
               <xsd:element
name="compraPassagemAereaNacionalResponseType">
```

```
                    <xsd:complexType>
                         <xsd:sequence>
                              <xsd:element name="string"
type="xsd:string" />
                         </xsd:sequence>
                    </xsd:complexType>
               </xsd:element>
          </xsd:schema>
     </wsdl:types>
     <wsdl:message name="compraPassagemAereaNacionalRequestMessage">
          <wsdl:part name="compraPassagemAereaNacionalRequestPart"
element="tns:compraPassagemAereaNacionalRequestType" />
     </wsdl:message>
     <wsdl:message
name="compraPassagemAereaNacionalResponseMessage">
          <wsdl:part name="compraPassagemAereaNacionalResponsePart"
element="tns:compraPassagemAereaNacionalResponseType" />
     </wsdl:message>
     <wsdl:portType name="CompraPassagemAereaNacionalPortType">
          <wsdl:operation name="CompraPassagemAereaNacional">
               <wsdl:input
message="tns:compraPassagemAereaNacionalRequestMessage" />
               <wsdl:output
message="tns:compraPassagemAereaNacionalResponseMessage" />
          </wsdl:operation>
     </wsdl:portType>
     <wsdl:binding name="CompraPassagemAereaNacionalServiceBinding"
type="tns:CompraPassagemAereaNacionalServicePortType">
          <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
          <wsdl:operation name="CompraPassagemAereaNacional">
               <soap:operation soapAction="" style="document" />
               <wsdl:input> <soap:body use="literal" />
</wsdl:input>
               <wsdl:output> <soap:body use="literal" />
</wsdl:output>
          </wsdl:operation>
     </wsdl:binding>
     <wsdl:service name="CompraPassagemAereaNacional">
          <wsdl:port name="CompraPassagemAereaNacionalServicePort"
binding="tns:CompraPassagemAereaNacionalServiceBinding">
               <soap:address
location="http://www.77a55c744200ea60.com/process/ServiceProcess" />
          </wsdl:port>
     </wsdl:service>
     <plnk:partnerLinkType
name="CompraPassagemAereaNacionalPartnerLinkType">
          <plnk:role name="CompraPassagemAereaNacionalServiceRole"
portType="tns:CompraPassagemAereaNacionalServicePortType" />
```

```
    </plnk:partnerLinkType>
</wsdl:definitions>
```

ANEXO II: WS-BPEL DA COMPOSIÇÃO DE SERVIÇO DE COMPRA DE PASSAGEM AÉREA NACIONAL

```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process name="CompraPassagemAereaNacional"
exitOnStandardFault="yes" suppressJoinFailure="yes"
    targetNamespace="http://samplePaper.windows.cin.ufpe.br"
    xmlns:air="http://air.services.secmosc.cin.ufpe.br"
xmlns:atomic="http://ode.apache.org/atomicScope"
    xmlns:bpmn="http://www.intalio.com/bpms"
xmlns:bpws="http://docs.oasis-
open.org/wsbpel/2.0/process/executable"
    xmlns:pay="http://payment.services.secmosc.cin.ufpe.br"
xmlns:pnlk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"

    xmlns:tns="http://samplePaper.windows.cin.ufpe.br.wsdl"
xmlns:tns1="http://core.secmosc.cin.ufpe.br/xsd"
    xmlns:vprop="http://docs.oasis-open.org/wsbpel/2.0/varprop"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
        location="VTA.wsdl"
namespace="http://samplePaper.windows.cin.ufpe.br.wsdl"/>
    <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
        location="WSAirService.wsdl"
namespace="http://air.services.secmosc.cin.ufpe.br"/>
    <bpws:import importType="http://schemas.xmlsoap.org/wsdl/"

        location="WSPaymentService.wsdl"
namespace="http://payment.services.secmosc.cin.ufpe.br"/>
    <bpws:partnerLinks>

        <bpws:partnerLink myRole="CompraPassagemAereaNacionalRole"
name="initializer"
partnerLinkType="tns:CompraPassagemAereaNacionalPartnerLinkType"/>
        <bpws:partnerLink name="airInvoke"
            partnerLinkType="air:AirServiceServicePartnerLinkType"
partnerRole="airCompany"/>
        <bpws:partnerLink name="payInvoke"
            partnerLinkType="pay:PaymentServicePartnerLinkType"
partnerRole="payCompany"/>

    </bpws:partnerLinks>

    <bpws:variables>
        <bpws:variable
messageType="tns:buyNationalTicketRequestMessage"
name="serviceRequest"/>
        <bpws:variable

            messageType="tns:buyNationalTicketResponseMessage"
name="serviceResponse"/>
        <bpws:variable
messageType="air:checkFlightAvailabilitRequest"
name="airCheckRequest"/>
```

```
        <bpws:variable
messageType="air:checkFlightAvailabilitResponse"
name="airCheckResponse"/>
        <bpws:variable messageType="air:confirmFlightRequest"
name="airConfirmationRequest"/>

        <bpws:variable messageType="air:confirmFlightResponse"
name="airConfirmationResponse"/>
        <bpws:variable messageType="pay:processPaymentRequest"
name="paymentRequest"/>
        <bpws:variable messageType="pay:processPaymentResponse"
name="paymentResponse"/>
    </bpws:variables>
    <bpws:sequence name="compraBilheteNacional">
        <bpws:receive createInstance="yes" name="recebeSolicitacao"
            operation="buyNationalTicket" partnerLink="initializer"
            portType="tns:CompraPassagemAereaNacionalPortType"
variable="serviceRequest"/>
        <bpws:assign validate="no">
            <bpws:copy>
                <bpws:from>
                    <bpws:literal>
                        <checkFlightAvailabilit
xmlns="http://air.services.secmosc.cin.ufpe.br">
                            <flightrequest
xmlns="http://air.services.secmosc.cin.ufpe.br">
                                <arrivalAirport
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <arrivalDate
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <departureAirport
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <departureDate
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <passengersNumber
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                            </flightrequest>
                        </checkFlightAvailabilit>
                    </bpws:literal>
                </bpws:from>

<bpws:to><![CDATA[$airCheckRequest.parameters]]></bpws:to>
            </bpws:copy>
            <bpws:copy>

<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:flightrequest_arrivalAirport]]></bpws:from>

<bpws:to><![CDATA[$airCheckRequest.parameters/air:flightrequest/tns1
:arrivalAirport]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
```

```
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:flightrequest_arrivalDate]]></bpws:from>
<bpws:to><![CDATA[$airCheckRequest.parameters/air:flightrequest/tns1
:arrivalDate]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:flightrequest_departureAirport]]></bpws:from>
<bpws:to><![CDATA[$airCheckRequest.parameters/air:flightrequest/tns1
:departureAirport]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:flightrequest_departureDate]]></bpws:from>
<bpws:to><![CDATA[$airCheckRequest.parameters/air:flightrequest/tns1
:departureDate]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:flightrequest_passengersNumber]]></bpws:from>
<bpws:to><![CDATA[$airCheckRequest.parameters/air:flightrequest/tns1
:passengersNumber]]></bpws:to>
            </bpws:copy>
        </bpws:assign>
        <bpws:invoke inputVariable="airCheckRequest"
name="verificaVoos"
            operation="checkFlightAvailabilit"
            outputVariable="airCheckResponse"
partnerLink="airInvoke" portType="air:AirServicePortType"/>
        <bpws:assign validate="no">
            <bpws:copy>
                <bpws:from>
                    <bpws:literal>

                        <confirmFlight
xmlns="http://air.services.secmosc.cin.ufpe.br">
                            <person
xmlns="http://air.services.secmosc.cin.ufpe.br">
                                <name
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <personID
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                            </person>
                            <flightrequest
xmlns="http://air.services.secmosc.cin.ufpe.br">
                                <arrivalAirport
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <arrivalDate
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <departureAirport
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <departureDate
```

```
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                    <passengersNumber
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                </flightrequest>
                                <flightAvailabilit
xmlns="http://air.services.secmosc.cin.ufpe.br">
                                    <available
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                    <embarkTaxes
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                    <ticketCost
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                </flightAvailabilit>
                            </confirmFlight>
                        </bpws:literal>
                    </bpws:from>

<bpws:to><![CDATA[$airConfirmationRequest.parameters]]></bpws:to>
                </bpws:copy>
                <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:person_name]]></bpws:from>
<bpws:to><![CDATA[$airConfirmationRequest.parameters/air:person/tns1
:name]]></bpws:to>
                </bpws:copy>
                <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:person_personID]]></bpws:from>
<bpws:to><![CDATA[$airConfirmationRequest.parameters/air:person/tns1
:personID]]></bpws:to>
                </bpws:copy>
                <bpws:copy>
<bpws:from><![CDATA[$airCheckRequest.parameters/air:flightrequest]]>
</bpws:from>
<bpws:to><![CDATA[$airConfirmationRequest.parameters/air:flightreque
st]]></bpws:to>
                </bpws:copy>
                <bpws:copy>
<bpws:from><![CDATA[$airCheckResponse.parameters/air:return]]></bpws
:from>
<bpws:to><![CDATA[$airConfirmationRequest.parameters/air:flightAvail
abilit]]></bpws:to>
                </bpws:copy>
            </bpws:assign>
            <bpws:invoke inputVariable="airConfirmationRequest"
                name="confirmaVoo" operation="confirmFlight"
                outputVariable="airConfirmationResponse"
                partnerLink="airInvoke"
portType="air:AirServicePortType"/>
            <bpws:assign validate="no">
```

```
            <bpws:copy>
                <bpws:from>
                    <bpws:literal>
                        <processPayment
xmlns="http://payment.services.secmosc.cin.ufpe.br">
                            <paymentRequest
xmlns="http://payment.services.secmosc.cin.ufpe.br">
                                <amount
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                <cardInformation
xmlns="http://core.secmosc.cin.ufpe.br/xsd">
                                    <cardCode
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                    <cardExpiration
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                    <cardHolder
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                    <cardNumber
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                                </cardInformation>
                                <merchant
xmlns="http://core.secmosc.cin.ufpe.br/xsd"/>
                            </paymentRequest>
                        </processPayment>
                    </bpws:literal>
                </bpws:from>
<bpws:to><![CDATA[$paymentRequest.parameters]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$airConfirmationRequest.parameters/air:flightAva
ilabilit/tns1:ticketCost]]></bpws:from>
<bpws:to><![CDATA[$paymentRequest.parameters/pay:paymentRequest/tns1
:amount]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:cardInfo_cardNumber]]></bpws:from>
<bpws:to><![CDATA[$paymentRequest.parameters/pay:paymentRequest/tns1
:cardInformation/tns1:cardNumber]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:cardInfo_cardHolder]]></bpws:from>

<bpws:to><![CDATA[$paymentRequest.parameters/pay:paymentRequest/tns1
:cardInformation/tns1:cardHolder]]></bpws:to>

            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:cardInfo_cardExpiration]]></bpws:from>
```

```
<bpws:to><![CDATA[$paymentRequest.parameters/pay:paymentRequest/tns1
:cardInformation/tns1:cardExpiration]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$serviceRequest.buyNationalTicketRequestPart/tns
:cardInfo_cardCode]]></bpws:from>
<bpws:to><![CDATA[$paymentRequest.parameters/pay:paymentRequest/tns1
:cardInformation/tns1:cardCode]]></bpws:to>
            </bpws:copy>
        </bpws:assign>
        <bpws:invoke inputVariable="paymentRequest"
            name="processaPagamento" operation="processPayment"
            outputVariable="paymentResponse" partnerLink="payInvoke"
portType="pay:PaymentServicePortType"/>
        <bpws:assign validate="no">
            <bpws:copy>
                <bpws:from>
                    <bpws:literal>
                        <buyNationalTicketResponseType
xmlns="http://samplePaper.windows.cin.ufpe.br.wsdl">
                            <flightNumber/>
                            <flightOrder/>
                            <approved/>
                            <authorizationDate/>
                            <authorizationNumber/>
                        </buyNationalTicketResponseType>
                    </bpws:literal>
                </bpws:from>
<bpws:to><![CDATA[$serviceResponse.buyNationalTicketResponsePart]]><
/bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$airConfirmationResponse.parameters/air:return/t
ns1:flightNumber/text()]]></bpws:from>
<bpws:to><![CDATA[$serviceResponse.buyNationalTicketResponsePart/tns
:flightNumber]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$airConfirmationResponse.parameters/air:return/t
ns1:flightOrder/text()]]></bpws:from>
<bpws:to><![CDATA[$serviceResponse.buyNationalTicketResponsePart/tns
:flightOrder]]></bpws:to>
            </bpws:copy>
            <bpws:copy>
<bpws:from><![CDATA[$paymentResponse.parameters/pay:return/tns1:appr
oved/text()]]></bpws:from>
<bpws:to><![CDATA[$serviceResponse.buyNationalTicketResponsePart/tns
```

```
:approved]]></bpws:to>

            </bpws:copy>

            <bpws:copy>
<bpws:from><![CDATA[$paymentResponse.parameters/pay:return/tns1:auth
orizationDate/text()]]></bpws:from>
<bpws:to><![CDATA[$serviceResponse.buyNationalTicketResponsePart/tns
:authorizationDate]]></bpws:to>

            </bpws:copy>

            <bpws:copy>
<bpws:from><![CDATA[$paymentResponse.parameters/pay:return/tns1:auth
orizationNumber/text()]]></bpws:from>
<bpws:to><![CDATA[$serviceResponse.buyNationalTicketResponsePart/tns
:authorizationNumber]]></bpws:to>

            </bpws:copy>

        </bpws:assign>
        <bpws:reply name="enviaMensagem"
operation="buyNationalTicket"
            partnerLink="initializer"
portType="tns:CompraPassagemAereaNacionalPortType"
variable="serviceResponse"/>

    </bpws:sequence>

</bpws:process>
```