

INTERNET DE SERVIÇOS: ESTUDO,  
ANÁLISE E PROPOSTA DE  
ARQUITETURA CONCEITUAL.

AGOSTINHO MANUEL VAZ

Dezembro 2011

# Internet de Serviços: Estudo, Análise e Proposta de Arquitetura Conceitual.

AGOSTINHO MANUEL VAZ

Dissertação apresentada ao Instituto Nacional de Telecomunicações, como parte dos requisitos para obtenção do Título de Mestre em Telecomunicações.

Orientador: PROF. DR. ANTÔNIO MARCOS ALBERTI

Santa Rita do Sapucaí  
2011

Vaz, Agostinho Manuel

V393i

Internet de Serviços: Estudo, Análise e Proposta de Arquitetura  
Conceitual. / Agostinho Manuel Vaz. – Santa Rita do Sapucaí, 2011.  
88 p.

Orientador: Prof. Dr. Antônio Marcos Alberti.  
Dissertação de Mestrado – Engenharia de Telecomunicações –  
Instituto Nacional de Telecomunicações – INATEL.

Inclui bibliografia.

1. Serviço 2. Internet de Serviços 3. Computação orientada a  
Serviços 4. Arquiteturas orientadas a Serviço 5. SLA 6. Engenharia de  
Telecomunicações. I. Alberti, Antônio Marcos. II. Instituto Nacional de  
Telecomunicações – INATEL. III. Título.

CDU 621.39

Dissertação defendida e aprovada em 19/12/2011, pela comissão julgadora:

---

Prof. Dr. Antônio Marcos Alberti (Orientador) - Inatel

---

Prof. Dr. Eduardo Cerqueira - UFPA

---

Prof. Dr. Carlos Alberto Ynoguti - INATEL

---

**Prof. Dr. Luciano Leonel Mendes - Coordenador do Curso de Mestrado**

Este trabalho é dedicado às duas pessoas mais importantes da minha vida: Wilma Celeste e Teixeira Vaz. Amo-vos.

*“The course of true love never did run smooth”*  
William Shakespeare.

# Agradecimentos

A minha mãe, Angélica Agostinho, por ter me dado o dom da vida e pelo apoio sempre presente durante toda a minha vida. Ao Teixeira Lukenie, meu filho, pela alegria que me dá todos os dias desde que nasceu e pela força que me transmite para superar as dificuldades.

A Wilma Celeste Fernandes, o meu muito obrigado pelo apoio, pela compreensão, pelo incentivo, pela ajuda e pela dedicação a mim, durante todos estes anos que estamos juntos. Este curso de mestrado não teria começado e, principalmente, não teria terminado sem tua companhia ao meu lado. Não importa o que acontecer serás sempre a mulher da minha vida. Te amo muito.

Ao Dr. Antônio Marcos Alberti, meu orientador, pelo seu sábio acompanhamento, atenção e compreensão durante todo o período de dissertação. Ao Instituto Nacional de Telecomunicações e a todos os professores, pela oportunidade singular de poder frequentar um curso de mestrado de alto nível.

Aos amigos e colegas de mestrado pela ajuda prestada.

# Sumário

<b>LISTA DE FIGURAS</b>	<b>vii</b>
<b>LISTA DE TABELAS</b>	<b>viii</b>
<b>LISTA DE ABREVIATURAS E SIGLAS</b>	<b>ix</b>
<b>CAPÍTULO 1 - INTRODUÇÃO</b>	<b>1</b>
1.1 Contextualização e Motivação. . . . .	1
1.2 Objetivo do Trabalho. . . . .	4
1.3 Organização do Trabalho. . . . .	4
<b>CAPÍTULO 2 – COMPUTAÇÃO ORIENTADA A SERVIÇOS</b>	<b>5</b>
2.1 Serviços: Conceitos, Valor e Características. . . . .	5
2.1.1 Conceitos de Serviço . . . . .	5
2.1.2 Valor Econômico do Serviço. . . . .	7
2.1.3 Características Intrínsecas do Serviço. . . . .	8
2.2 A Visão do Software como um Serviço. . . . .	11
2.3 Computação Orientada a Serviços. . . . .	13
2.4 Arquitetura Orientada a Serviços. . . . .	16
2.4.1 Enterprise Service Bus. . . . .	18
2.5 Benefícios da Implementação do SOA. . . . .	19
2.6 Negociação em Ambientes Orientados a Serviços. . . . .	20
2.6.1 Característica do Mercado de Serviços. . . . .	21
2.6.2 Negociação. . . . .	22



2.6.2.1	Pré – negociação. . . . .	22
2.6.2.2	Fase de Negociação. . . . .	24
2.6.2.3	Prestação do Serviço. . . . .	25
	<b>CAPÍTULO 3 – TRABALHOS RELACIONADOS</b>	<b>26</b>
3.1	SOA4ALL. . . . .	26
3.1.1	Incorporação dos Princípios da Web no SOA. . . . .	27
3.1.2	Incorporação da Web Semântica no SOA. . . . .	28
3.1.3.	Incorporação da Ciência do Contexto no SOA. . . . .	28
3.1.4	Incorporação da Web 2.0 no SOA. . . . .	29
3.1.5	Arquitetura SOA4ALL. . . . .	31
3.1.5.1	Barramento de Serviço Distribuído. . . . .	31
3.1.5.2	SOA4ALL Studio. . . . .	32
3.1.5.3	Plataforma de Serviço. . . . .	33
3.2	SLA@SOI. . . . .	33
3.2.1	Objetivos Funcionais. . . . .	35
3.2.2	Gerenciamento de SLA. . . . .	36
3.2.3	Principais Componentes da Arquitetura. . . . .	37
	<b>CAPÍTULO 4 – GENERALIZED INDIRECTION RESOLUTION SYSTEM – GIRS</b>	<b>41</b>
4.1	Desacoplamento da Identificação da Localização. . . . .	41
4.2	Redes Centradas na Informação. . . . .	42
4.3	Indireção. . . . .	43
4.4	Distributed Hash Tables. . . . .	44
4.5	GIRS. . . . .	46
4.5.1	Estrutura de Mapeamento para o GIRS. . . . .	47
4.5.2	Estrutura do Descritor. . . . .	47
4.5.3	Mecanismos de Resolução. . . . .	48
4.5.4	Escalabilidade. . . . .	49
4.5.5	Localização e Recuperação de Conteúdo. . . . .	49

<b>CAPÍTULO 5 – INTERNET DE SERVIÇOS BASEADA NO GIRS</b>	<b>51</b>
5.1 Tabela de Mapeamentos para a Internet de Serviços. ....	53
5.2 Mecanismo de Resolução. ....	55
5.3 Mecanismos de Publicação, Assinatura e Notificação. ....	56
5.4 Estrutura do Descritor. ....	57
5.5 Ajuste de Nomes. ....	57
5.6 Localização e Contratação de um Serviço. ....	58
5.6.1 Localização. ....	59
5.6.2 Contratação. ....	60
5.6.2.1 Estrutura de um SLA. ....	61
5.7 Discussões sobre Questões de Segurança, Confiança e de Dependência.	64
<b>CAPÍTULO 6 - CONCLUSÕES</b>	<b>65</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>67</b>

## Lista de Figuras

Figura 1. Termos associados a serviços.....	05
Figura 2. Contribuição do PIB brasileiro de cada setor da Economia.....	08
Figura 3. Características Intrínsecas dos serviços.....	09
Figura 4. Arquitetura básica do SOA.....	17
Figura 5. Exemplo da função de um ESB no SOA.....	19
Figura 6. Fases da negociação de Serviços.....	22
Figura 7. Tecnologias incorporadas pelo SOA4ALL.....	27
Figura 8. Arquitetura Simplificada do SOA4ALL.....	31
Figura 9. Exemplo de gerenciamento de um SLA.....	35
Figura 10. Componentes Principais do SLA@SOI.....	37
Figura 11. Exemplo de gerenciamento de SLA com os componentes da arquitetura.....	40
Figura 12. Principais vantagens das DHTs.....	45
Figura 13. Estrutura do GIRS.....	46
Figura 14. Estrutura dos mapeamentos para o GIRS.....	47
Figura 15. Estrutura do descritor para o GIRS.....	48
Figura 16. Localização e recuperação de conteúdo usando o GIRS.....	49
Figura 17. Estrutura do mapeamento para Internet de Serviços com base no GSIR.....	52
Figura 18. Ajuste e codificação de nomes.....	58
Figura 19. Exemplo de um processo de localização de serviço.....	59
Figura 20. Estrutura de SLA.....	61
Figura 21. Exemplo do processo de contratação de um serviço.....	63

## Lista de Tabelas

Tabela 1. Benefícios da integração da <i>Web</i> 2.0 e SOA4ALL.....	30
Tabela 2. Exemplos de mapeamento para a Internet de Serviço. ....	54

# Lista de Abreviaturas e Siglas

<b>ACK</b>	<i>Acknowledgment</i>
<b>AJAX</b>	<i>Asynchronous Javascript and XML</i>
<b>ASP</b>	<i>Application Server Provider</i>
<b>BM</b>	<i>Business Manager</i>
<b>CCN</b>	<i>Content Centric Network</i>
<b>CORBA</b>	<i>Common Object Request Broker Architecture</i>
<b>DCOM</b>	<i>Distributed Component Object Model</i>
<b>DHT</b>	<i>Distributed Hash Table</i>
<b>DSB</b>	<i>Distributed Service Bus</i>
<b>ESB</b>	<i>Enterprise Service Bus</i>
<b>FIA</b>	<i>Future Internet Assembly</i>
<b>FP7</b>	<i>European Union Framework Programme 7</i>
<b>FISO</b>	<i>Future Internet Service Offer</i>
<b>GIRS</b>	<i>Generalized Indirection Resolution System</i>
<b>GPS</b>	<i>Global Position System</i>
<b>HIP</b>	<i>Host Identity Protocol</i>
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i>
<b>IoS</b>	<i>Internet of Services</i>
<b>IP</b>	<i>Internet Protocol</i>
<b>JRMI</b>	<i>Java Remote Method Invocation</i>
<b>LISP</b>	<i>Locator/ID Split Protocol</i>
<b>MA</b>	<i>Management Agents</i>
<b>MB</b>	<i>Mecanismo de Busca</i>
<b>MILSA</b>	<i>Mobility and Multihoming Supporting Identifier Locator Split</i>
<b>NRS</b>	<i>Name Resolution System</i>
<b>NetInf</b>	<i>Network of Information</i>
<b>NESSI</b>	<i>Networked European Software and Services Initiative</i>
<b>PARC</b>	<i>Palo Alto Research Center Incorporated</i>

<b>PIB</b>	<i>Produto Interno Bruto</i>
<b>PSIRP</b>	<i>Publish/Subscribe Internetworking Routing Paradigm</i>
<b>QoE</b>	<i>Quality of Experience</i>
<b>QoS</b>	<i>Quality of Service</i>
<b>RPC</b>	<i>Remote Procedure Call</i>
<b>SLA</b>	<i>Service Level Agreement</i>
<b>SOC</b>	<i>Service Oriented Computing</i>
<b>SOI</b>	<i>Service Oriented Infrastructure</i>
<b>SOAP</b>	<i>Simple Object Access Protocol</i>
<b>SOA</b>	<i>Service Oriented Architecture</i>
<b>SOA4ALL</b>	<i>Service Oriented Architecture for All</i>
<b>SIR</b>	<i>Search and Indirection Resolution</i>
<b>SE</b>	<i>Service Evaluation</i>
<b>SM</b>	<i>Service Manager</i>
<b>SLAM</b>	<i>SLA Manager</i>
<b>TI</b>	<i>Tecnologia de Informação</i>
<b>TTL</b>	<i>Time To Live</i>
<b>WWW</b>	<i>World Wide Web</i>
<b>XML</b>	<i>Extensive Markup Language</i>

## Resumo

Este trabalho estuda as principais características das arquiteturas de Internet de Serviços. Estuda ainda, algumas abordagens já propostas para este paradigma de Internet do Futuro. A construção dinâmica e flexível de aplicações, a partir da combinação autonômica de diferentes serviços distribuídos é um dos objetivos da Internet de Serviços. Assim, propõe-se uma Arquitetura de Internet de Serviços conceitual que permite que os serviços descubram uns aos outros, estabeleçam contratos e se auto-organizem de forma distribuída e transparente.

**Palavras-chave:** Serviço, Internet de Serviços, computação orientada a serviços, arquiteturas orientadas a serviço, SLA.

# Abstract

This work studies the main features of the Internet of Service's architectures. Also, it studies some proposed approaches to this paradigm of the Future Internet. The dynamic and flexible construction of application, from the autonomic combination of different distributed services is one of the goals of Internet of Service. Thus, we propose a Conceptual Architecture for Internet of Service which enables services to discover each other, establish contracts and self-organize in distributed and transparent way.

**Keywords:** Service, Internet of Service, service-oriented computing, service-oriented architectures, SLA.



# Capítulo 1

## Introdução

### 1.1 Contextualização e Motivação

Com pouco mais de dois bilhões de usuários, ou aproximadamente 30.2% da população mundial (INTERNET USAGE STATISTIC, 2011), pode-se afirmar que a Internet é o mais bem sucedido artefato de engenharia que o ser humano já criou. Concebida inicialmente como uma rede militar de interconexão de computadores, rapidamente evoluiu para uma plataforma global provendo ampla comunicação através de protocolos padronizados e abertos. Com o surgimento da *World Wide Web* (WWW), ela se popularizou e se transformou em uma ferramenta usada para os mais diversos fins, que vão desde meio de comunicação até infraestrutura de interação social.

A Internet mudou de forma profunda e irreversível nossa sociedade e nossa economia. Se hoje ela é elemento crucial na economia, a Internet do futuro irá desempenhar um papel ainda mais vital em todos os processos de negócio. Vai tornar-se uma ferramenta de produtividade por excelência (REDING, V., 2008), uma vez que a maior parte dos processos de negócio envolverá serviços na Internet.

Hoje, todos os países desenvolvidos têm suas economias baseadas no setor de serviços, em termos de números de trabalhadores empregados e receitas geradas. Enquanto que os primeiros serviços foram certamente prestados por seres humanos para seres humanos, os avanços nos sistemas de computação nos últimos 40 anos permitiram que os computadores prestassem serviços aos homens. As Tecnologias de Informação (TI), de forma geral, e a Internet, de forma muito particular, têm contribuído significativamente para a evolução do setor de serviços, conseqüentemente para a evolução da economia mundial. A cada inovação na área de TI, criam-se soluções que permitem a execução automática de atividades feitas anteriormente pelo homem. A popularização da Internet tem permitido uma interação remota e generalizada entre os seres humanos e os diversos

dispositivos computacionais disponíveis, como, computadores, *smartphones*, *tablets*, dentre outros.

A evolução tecnológica do final dos anos 90 levou o conceito de serviço para a *web*, através dos chamados *Web Services* (BOOTH et al, 2004) que tem por objetivo apoiar a interoperabilidade entre os diferentes sistemas computacionais. Esta interoperabilidade requer serviços que sejam autônomos e independentes de plataforma computacional. Ela requer que os serviços sejam descritos, publicados, descobertos e invocados utilizando protocolos padrões. O foco desta tecnologia centrou-se na definição de interfaces entre serviços com uma perspectiva técnica.

Devido a esta perspectiva estritamente técnica, até poucos anos atrás, o uso de serviços através da Internet ficou restrito a profissionais de TI e aos departamentos de tecnologias de informação dentro das grandes empresas.

Com o uso cada vez maior da Internet nos mais diversos ramos, houve a necessidade de se ampliar o conceito de serviço na rede para um nível superior, não apenas técnico ou restrito à tecnologia de informação, mas sim trazer para a rede também os serviços do mundo real ou serviços do dia-a-dia, como por exemplo, educação. Isto implica na pesquisa e desenvolvimento de técnicas de representação e modelamento digital de serviços.

O modelamento e representação digital dos serviços dão origem ao paradigma da Internet de Serviços. A Internet de Serviços tem como objetivo pesquisar e desenvolver novas teorias, modelos, arquiteturas e tecnologias para fornecer soluções eficientes que permitam que usuários não profissionais criem, comercializem e consumam serviços disponíveis na Internet (CARDOSO et al, 2011). Na Internet de Serviços, *software*, plataformas computacionais, infraestrutura, banco de dados, tudo é visto como serviço.

Este conceito de “tudo como serviço” nos leva ao paradigma de Computação Orientada a Serviço. Este paradigma defende que os serviços sejam usados para apoio ao

desenvolvimento de aplicações complexas de forma rápida, a baixo custo e distribuídas. Este paradigma é independente de linguagens de programação ou sistemas computacionais, permitindo que todos, desde grandes empresas até pequenos usuários, exponham suas competências e aptidões através da Internet.

O paradigma de computação orientada a serviço é extremamente amplo e complexo, pois abrange diversas tecnologias, conceitos e protocolos que têm em suas origens áreas como sistemas distribuídos, redes de computadores, arquitetura de computadores, engenharia de *software*, banco de dados, linguagens de programação, segurança, inteligência artificial, dentre outras, e que se relacionam de forma complexa (PAPAZOGLU et al, 2006). Implica dizer que, somente através da adoção de uma abordagem holística para este paradigma, é que se poderá aproveitar todo o seu potencial.

A chave para este paradigma são as Arquiteturas Orientadas a Serviço (SOA, do Inglês, *Service Oriented Architectures*) (CHANNABASAVIAIAH et al 2004) (PAPAZOGLU, et al., 2006). Estas arquiteturas são formas lógicas de construção de sistemas de *software* para prestar serviços a qualquer usuário final ou outros serviços distribuídos em uma rede, através da publicação e descoberta de interfaces de serviço. Isto permite criar infraestruturas flexíveis e dinâmicas que se adaptam rapidamente a qualquer tipo de mudança.

Iniciativas como o NESSI (*Networked European Software and Services Initiative*) (NESSI, 2011) e o FISO (*Future Internet Service Offer*) (DOMINGUE, 2009), ambas financiados pela União Européia são exemplos da união entre a indústria, academia e governos de todo mundo para o desenvolvimento de arquiteturas de referencia para a Internet do Futuro, de forma geral, e da Internet de Serviços, em particular.

## **1.2 Objetivo do Trabalho**

O objetivo deste trabalho é o estudo e a análise das principais características de arquiteturas de serviços que irão compor a futura Internet de Serviços e estudar também algumas abordagens já propostas para este paradigma de Internet. Pretende-se ainda, a ampliação do Sistema Generalizado de Resolução de Indireções<sup>1</sup> proposto por (MAGALHÃES, 2011) na direção de uma arquitetura convergente para a Internet de Informações e Serviços. A arquitetura proposta trata as questões relativas à identificação única, nomeação, localização e contratação de entidades de serviços no escopo do Sistema Generalizado de Resolução de Indireções.

## **1.3 Organização do Trabalho**

Esta dissertação está organizada da seguinte maneira: O Capítulo 2 apresenta o paradigma de computação orientada a serviço, as características e propósitos deste paradigma e sua implementação, através das arquiteturas orientadas a serviço. Neste capítulo ainda, para melhor compreensão deste paradigma, apresentamos também os principais conceitos associados ao serviço, suas características e seu valor econômico. O Capítulo 3 contém o estudo de duas propostas de arquitetura para a Internet de Serviço. No Capítulo 4 é abordado o Sistema Generalizado de Resolução de Indireções, que será usado como base para a proposta deste trabalho. No Capítulo 5 tem-se a proposta principal desta dissertação, uma abordagem convergente para a Internet de Informações e Serviços, com um estudo de caso para exemplificar a localização e contratação de serviços em ambientes altamente dinâmicos e distribuídos. O Capítulo 6 apresentará as conclusões deste trabalho e algumas propostas para trabalhos futuros.

---

<sup>1</sup> O Sistema Generalizado de Resolução de Indireções será apresentado no Capítulo 4. Esse sistema provê a resolução de mapeamentos entre nomes.

## Capítulo 2

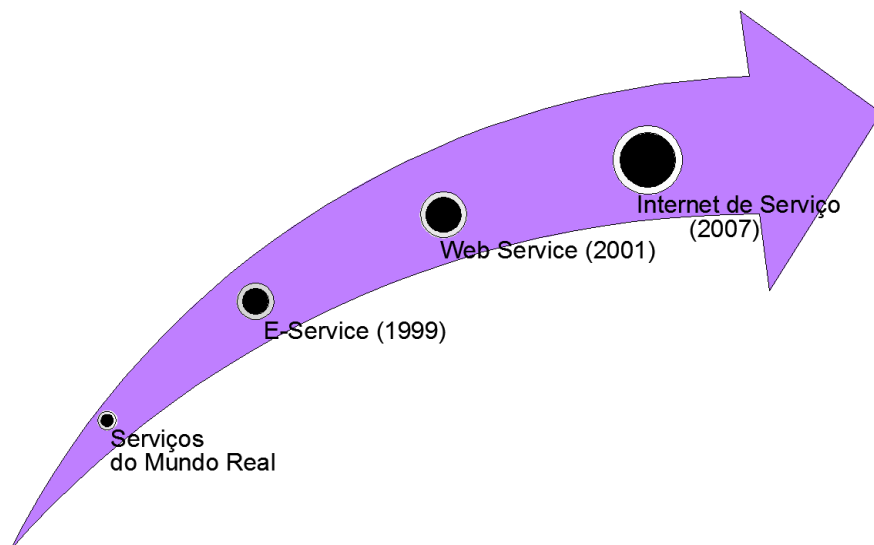
# Computação Orientada a Serviços

### 2.1 Serviços: Conceitos, Valor e Características

O uso generalizado do termo serviço para se referir a situações conceitualmente distintas, cria, geralmente, alguma confusão. Para evitar qualquer ambigüidade, esta seção apresentará os principais conceitos associados a este termo, suas características intrínsecas e o valor que eles agregam à economia de mercado baseada em serviços digitais na Internet.

#### 2.1.1 Conceito de Serviço

Os termos mais relevantes associados ao conceito de serviço, conforme introduzidos pela comunidade científica e industrial (CARDOSO, et al., 2011) ao longo dos anos, estão ilustrados na Figura 1.



**Figura1:** Termos associados a serviço.

**Serviços do mundo real:** O termo serviços do mundo real, isto é, serviços do dia-dia ou simplesmente, serviço, é usado para se referir a qualquer tipo de serviço que pode ser encontrado na sociedade. Por causa da sua diversidade e heterogeneidade é tradicionalmente difícil definir o significado desta palavra. Segundo (HOLANDA, A., 2008), serviço é o produto da atividade humana destinado à satisfação de necessidades humanas, mas que não apresenta o aspecto de um bem material, como por exemplo, transporte, educação, assistência médica, etc. Segundo (CARDOSO, et al., 2011), no mundo dos negócios, é um conjunto de atividades intangíveis que geralmente ocorrem nas interações entre um prestador e um consumidor, onde o foco não está nos protocolos, padrões ou programas, mas na busca constante de melhoria na experiência do consumidor com o serviço.

**E-Service:** São serviços para os quais redes de comunicação, tal como a Internet, são usadas como canal para a interação entre os consumidores e serviços remotos. Praticamente, qualquer serviço pode ser transformado em um *e-service* se ele puder ser invocado através de uma rede. Eles são independentes da linguagem de programação utilizada para definir suas funcionalidades, interfaces ou propriedades não funcionais. Assim como os serviços do mundo real, sua definição é bastante ampla. (HULL, et al., 2003) define um *e-service* como, uma coleção de programas residentes na rede, acessíveis por protocolos padronizados, em que suas funcionalidades podem ser descobertas e integradas a aplicativos ou compostos na formação de serviços mais complexos. O principal requisito de um *e-service*, é a possibilidade de invocação e interação remota com o serviço.

**Web-services:** (BOOTH et al, 2004) define *web services* como “um sistema de *software* projetado para suportar a interoperabilidade nas interações máquina-máquina na rede”. Podemos ainda definir os *web services* como uma tecnologia que permite a comunicação entre aplicativos independentemente de sua plataforma e linguagem. Atualmente, os *web services* são elementos essenciais para a computação, pois ela está baseada em aplicativos heterogêneos, autônomos e distribuídos. Tecnologias como *Remote Procedure Call* (RPC), *Common Object Request Broker Architecture* (CORBA),

*Distributed Component Object Model* (DCOM) e *Java Remote Method Invocation* (JRMI), são exemplos de algumas tecnologias que faziam a interoperabilidade dos sistemas antes dos *web services*. Elas tinham desvantagens significativas para o desenvolvimento de aplicações distribuídas, tais como incompatibilidade entre os fornecedores, complexidade e o custo das soluções.

**Internet de Serviços:** Segundo (CARDOSO, et al. 2011) os serviços que serão oferecidos neste novo paradigma de Internet serão serviços representados com estruturas digitais, especificações e normas a serem negociadas na Internet de Serviços (IoS, sigla em Inglês de *Internet of Service*).

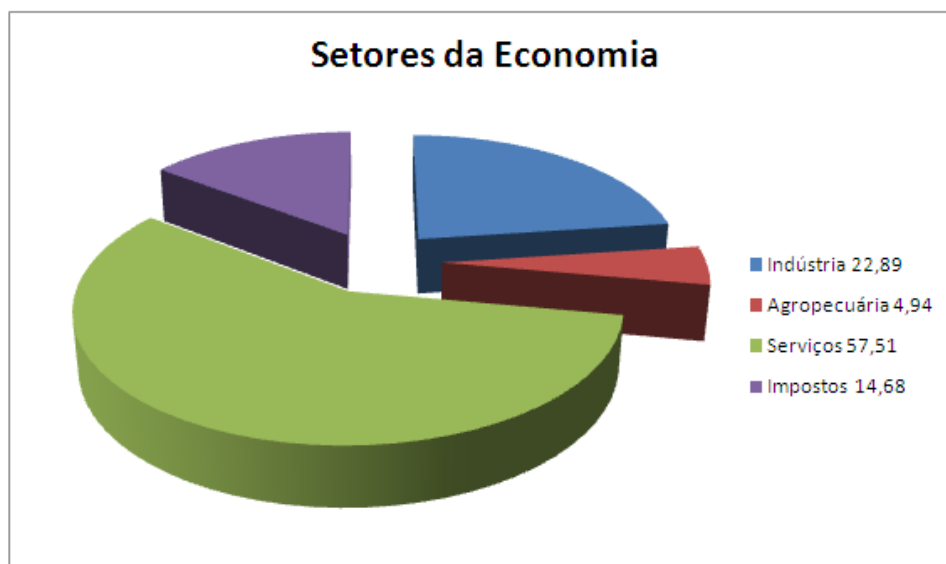
Duas principais características tornam os serviços baseados na IoS distintos dos demais tipos de serviços já apresentados: primeiro esta noção de serviço não se limita aos serviços relacionados a TI, mas também aos serviços do mundo real; segundo, as partes interessadas em tais serviços, não são apenas os profissionais de tecnologia, mas usuários leigos na parte técnica dos serviços. Assim, os serviços baseados na Internet de Serviços servem para um propósito duplo, pois podem ser utilizados diretamente pelos consumidores, mas também podem ser invocados por sistemas técnicos para acessar funcionalidades de negócios que são fornecidos pelos provedores de serviços.

Enquanto, os *e-services* e *web services* são vistos como entidades, principalmente, tecnológicas, a Internet de Serviços trará a representação dos serviços um passo a frente. O desenvolvimento de soluções para a IoS é mais elaborada, pois ela deve levar em conta todas as características específicas dos serviços do mundo real.

### **2.1.2 Valor Econômico do Serviço**

O setor de serviço, também chamado em economia como, terceiro setor, é o novo motor da economia mundial. Este setor envolve a comercialização de produtos em geral e a prestação de serviços às empresas e consumidores.

O Brasil é considerado hoje como a oitava maior economia do mundo. Este setor representa 57,51 % do Produto Interno Bruto (PIB) nacional (LACERDA, L., 2011), o que mostra sua importância. A Figura 2 representa a contribuição de cada setor na economia brasileira.



**Figura 2:** Contribuição no PIB brasileiro de cada setor da Economia.

Com base no valor econômico e importância dos serviços, a criação de uma infraestrutura para a aplicação efetiva de uma Internet de Serviços é de extrema importância para o fornecimento de soluções para a criação de um verdadeiro mercado para a prestação de serviços sem fronteiras.

### 2.1.3 Características Intrínsecas do Serviço

A compreensão da natureza dos serviços existentes no mundo real é fundamental para qualquer proposta de solução para a Internet de Serviços, pois serão estes que serão digitalizados e representados adequadamente na nova Internet. O objetivo é permitir a comercialização de serviços através da rede. A Figura 3 mostra as características intrínsecas dos serviços do mundo real (CARDOSO, et al. 2011), como, intangibilidade, bipolaridade, variabilidade, etc.





**Figura 3:** *Características Intrínsecas dos Serviços.*

**Intangíveis:** Serviços são intangíveis, uma vez que não existem materialmente. Isto não impede que seus resultados possam ser vistos e avaliados. Devido a esta característica, é difícil criar especificações adequadas para modelar e definir os atributos que descrevem objetivamente um serviço. É fundamental que as características e aspectos fundamentais dos serviços do mundo real estejam presentes no modelo de Internet de Serviços.

**Inseparáveis:** Normalmente a prestação e consumo do serviço ocorrem em paralelo. Isto implica que um rigoroso equilíbrio de oferta e demanda deve ser alcançada, senão, serviços serão perdidos ou consumidores terão necessidade de esperar pela disponibilidade do serviço. A existência de mecanismos que providenciam um balanceamento entre a oferta e demanda é fundamental (CARDOSO, et al. 2011).

**Imersivos:** Serviços são geralmente executados em colaboração entre os fornecedores e os consumidores, e por vezes, é difícil determinar responsabilidades pelo

sucesso e/ou fracasso do mesmo. A definição de Acordos de Nível de Serviço (SLA, sigla em Inglês de *Service Level Agreement*) é de extrema relevância.

**Bipolar ou híbridos:** Serviços são geralmente executados com uma mistura de recursos humanos e tecnológicos. Enquanto abordagens para monitorar os recursos puramente tecnológicos estão disponíveis, soluções para monitorar o envolvimento humano na execução dos serviços e a complexa relação entre as dimensões humanas e tecnológicas estão sendo estudadas no contexto de ambientes altamente distribuídos, autônomos e heterogêneos como a IoS (CARDOSO, et al. 2011).

**Variável:** Ao contrário dos produtos que têm alto grau de padronização, os serviços são, na maior parte das vezes, feitos especificamente para cada consumidor ou grupo deles e são, portanto, heterogêneos. Organizações se diferenciam na oferta de produtos e serviços, mas as variações entre produtos são bem menores do que, entre os serviços das diferentes empresas. A qualidade e consistência dos serviços estão sujeitos a uma considerável variabilidade, uma vez que dependem de seres humanos. Atitudes humanas, como, comportamento, estilo, habilidades, emoções, entre outras, variam de pessoa para pessoa, e são difíceis de gerenciar e avaliar (CARDOSO, et al. 2011).

**Não Ostensivos:** Devido à propriedade de intangibilidade, a maior parte dos serviços também não é ostensiva, isto é, não pode ser ostentado visualmente. Por exemplo, não se pode exibir uma consulta medica como se ostenta uma peça de arte ou jóia.

**Longa duração:** comercializar produtos requer baixo nível de interação entre fornecedores e consumidores. Já os serviços requerem uma interação entre consumidor e provedor de longa duração, desde a localização, a contratação e durante o prazo de vigência do contrato do serviço<sup>2</sup>.

---

<sup>2</sup> O contrato de serviço será apresentado no item 2.4. Trata-se de um acordo entre as partes prestadora e de usufruto de um dado serviço.

**Desacoplados:** O processo de descoberta, seleção, invocação, execução e rescisão, é, resumidamente, o ciclo de vida de um serviço. Na Internet de Serviços cada uma destas fases é desacoplada e pode se posicionar em qualquer lugar no espectro do serviço.

**Perecível:** Mais uma vez, devido ao fato dos serviços serem intangíveis, normalmente não é possível armazená-los, isto é, a capacidade não utilizada de um serviço não pode ser armazenada para ser comercializada futuramente. Esta impossibilidade de armazenamento de serviços traz um desafio extra para a Internet de Serviço, pois esta requer uma otimização na utilização dos recursos disponibilizados por um serviço.

**Qualitativos:** A evidência física ou os produtos tangíveis originados pela execução de um serviço fornecem os parâmetros que permitem medir a qualidade do serviço. Esta característica é novamente um desafio para a Internet de Serviço. Segundo (CARDOSO, et al. 2011), duas perguntas podem ser feitas neste momento: Como identificar quais aspectos da execução de um serviço podem ser usados para avaliar a qualidade dos serviços, tanto quantitativamente como qualitativamente? E como a Internet de Serviços pode devidamente atender e gerir as expectativas dos consumidores?

Todas estas características dos serviços do mundo real devem ser representadas e modeladas digitalmente nas propostas para Internet de Serviços, em particular, como de forma geral, em todas as propostas de Internet do Futuro.

## **2.2 A Visão do *Software* como um Serviço**

Comprar ou alugar? Esta questão aplica-se não apenas a carros ou casas, mas agora também a *software* (ELFATATRY & LAYZELL, 2004). Tradicionalmente, *softwares* têm sido concebidos, desenvolvidos e entregues como produto, mas com o advento da computação orientada a serviços, surgiu o interesse de desenvolver *software* como serviço.

A principal vantagem desta abordagem é o fraco acoplamento entre os problemas do negócio e as soluções de *software* associadas, aumentando assim, a capacidade de desenvolver sistemas quando os requisitos de negócio mudam.

Segundo (ELFATATRY & LAYZELL, 2004), o conceito de *software* como um serviço é relativamente simples: “Não compre *software*, apenas use-o como e quando precisar dele”. Este conceito, apesar de revolucionário, apareceu pela primeira vez com o modelo de Provedores de Aplicações de Serviço (ASP, sigla em Inglês de *Application Service Provider*), muito utilizados na década de 90.

Um ASP é uma entidade que implanta, hospeda e gerencia o acesso a um pacote de aplicativos e soluções de serviços a partir de um *Data Center* centralizado para clientes distribuídos em uma rede. As aplicações são oferecidas ou entregues através de uma rede de comunicação com base em assinaturas ou aluguel. Em essência, os ASPs eram a maneira que as empresas tinham de terceirizar algumas ou todas as necessidades de tecnologia de informação na década de 90. O modelo ASP sofre de grandes limitações, como as restrições no desenvolvimento de aplicações altamente interativas e dificuldade de prover aplicações específicas para cada cliente. Isto resultou em arquiteturas frágeis, monolíticas e não reutilizáveis baseadas nos princípios do acoplamento forte<sup>3</sup>.

Com a computação orientada a serviços, a visão de *software* como serviço, encontra-se em outro estágio, arquitetado para interações assíncronas fracamente acopladas, com base em padrões aceitos por todas as plataformas computacionais, como o XML (*eXtensible Markup Language*), tornando o acesso e a comunicação das aplicações mais fáceis via Internet.

### **2.3 Computação Orientada a Serviços**

---

<sup>3</sup> Acoplamento é a medida de quanto duas entidades estão relacionadas, dependentes ou ainda têm conhecimentos uma da outra. O princípio de acoplamento fraco defende que o desenvolvimento de aplicações esteja totalmente dissociado com problemas específicos de negócios, aumentando a capacidade de adaptação das aplicações para novos requisitos.

A Computação Orientada a Serviços (SOC, sigla em Inglês de *Service-Oriented Computing*) é o paradigma de computação que utiliza serviços como o elemento fundamental para o desenvolvimento de aplicações/soluções. Serviços são autônomos, independentes de plataforma computacional, fracamente acoplados e podem ser descritos, publicados e descobertos de várias formas. Estas características permitem o desenvolvimento de aplicações, usando serviços, de forma rápida, a baixo custo, com interoperabilidade e distribuídas maciçamente, que podem desempenhar diversas funções, que vão desde uma simples requisição até um complexo processo de negócio (PAPAZOGLU, et al., 2006).

O SOC se baseia na idéia de que aplicações podem ser construídas de forma flexível e dinâmica, a partir da composição de *softwares* distribuídos disponíveis em rede, organizados para a realização de alguma tarefa.

Para a existência do paradigma de computação orientado a serviço três requisitos básicos devem ser observados:

- **Neutralidade Tecnológica:** Serviços devem ser invocados por meio de tecnologias padronizadas, disponíveis em quase todos os ambientes de tecnologia de informação. Isto implica que os mecanismos de invocação dos serviços devem estar em conformidade com os padrões amplamente aceites.

- **Acoplamento Fraco:** Os serviços não devem exigir dos seus clientes, qualquer conhecimento de sua estrutura interna, e os clientes não podem exigir do serviço o conhecimento do contexto em que ele será usado.

- **Localização Transparente:** A localização, contratação e invocação de um serviço deve ser independente de sua localização física na rede. Os serviços devem ter suas definições e informações em algum repositório, acessível para o maior número de clientes possível.

Os serviços são oferecidos por prestadores que possuem as implementações destes e fornecem a descrição dos mesmos. Os clientes podem ser outras aplicações ou processos internos da aplicação. Os serviços podem ser simples ou compostos. Os serviços compostos envolvem a montagem do serviço, através do acesso e combinação de funções e informações de vários provedores de serviços. Assim, todos os serviços devem ser criados de forma que sejam facilmente integrados a outros serviços distribuídos (PAPAZOGLU, et al., 2006).

As aplicações baseadas em serviços (*service-based applications*) são desenvolvidas por um conjunto de serviços independentes que interagem, oferecendo interfaces bem definidas aos seus potenciais utilizadores. Isto é feito sem a necessidade de acoplamento forte entre as aplicações parceiras, nem de acordos pré-determinados a serem postos antes do uso do serviço.

A adoção do paradigma de computação orientada a serviços possui os principais objetivos:

- **Reusabilidade:** Este foi o principal objetivo da engenharia de *software* durante muitos anos. No contexto do SOC, a capacidade de compor novos serviços reutilizando os serviços já existentes, fornece inúmeras vantagens para as empresas, como o custo dos investimentos, melhora na qualidade do *software*, e a diminuição do *time-to market* (tempo que leva desde a concepção até o produto chegar ao mercado).

- **Flexibilidade e Agilidade nos Negócios:** O atual cenário de negócios exige que as empresas modifiquem continuamente seus sistemas de acordo com as mudanças do mercado, para atender as novas exigências, requisitos e outros fatores (Carter, 2007). Assim, a arquitetura das corporações deve ser facilmente reconfigurável. As características do paradigma de computação orientada a serviços permitem um melhor alinhamento entre a tecnologia de informação e os negócios, com a integração de novos serviços e a reutilização dos serviços já existentes, a fim de responder rapidamente às mudanças do mercado (ENDREI, et al, 2004)

- **Múltiplos Prestadores de Serviço:** O paradigma orientado a serviços permite aos clientes, aplicações ou empresas, a escolha entre os vários fornecedores de um serviço específico. Assim, as organizações são sempre livres para mudar, substituir ou adicionar tecnologias.

Os *web services* são hoje o ambiente mais próximos da visão orientada a serviços. Eles influenciaram diversos conceitos de orientação a serviços, como, serviços fracamente acoplados e abstração de serviços (ERL, 2007). Assim como o paradigma de computação orientado a serviços, os *web services* expõem suas características de programação através da Internet, usando linguagens e protocolos padrões e podem ser implementados através de uma interface baseada em padrões abertos para a Internet (exemplo, interfaces XML). (FREMANTLE et al 2002) define os seguintes atributos para os *web services*:

**Protocolos Padrões da Internet:** O uso do SOAP<sup>4</sup> (*Simple Object Access Protocol*) sobre o HTTP (*Hypertext Transfer Protocol*) no projeto dos *web services* permite a sua utilização em ambientes heterogêneos.

**Interoperabilidade:** O SOAP fornece um padrão comum para a troca de mensagens padronizadas XML. Para tanto, define um modelo de mensagens e ações relacionadas.

**Baseado em XML:** O uso do XML contribui para a interoperabilidade através da criação de um padrão comum de criação de documentos.

**Modular:** Os componentes do serviço são reutilizáveis, permitindo a composição de um serviço através de múltiplos serviços diferentes.

---

<sup>4</sup> O SOAP é um protocolo para troca de informações estruturadas em um ambiente distribuído e descentralizado usando o XML. É constituído por três partes distintas: Um envelope, que define o tipo de mensagem e como processá-la; um conjunto de regras codificadas para expressar instâncias dos tipos de dados definidos na aplicação; e uma convenção para representar chamadas de procedimentos (MITRA, et al., 2007).

**Descritivos:** Os serviços devem ser incorporados com uma descrição padrão, permitindo assim, a identificação da interface do serviço.

**Implementação Independente:** A interface de serviço está disponível de forma independente da implementação final do serviço.

**Publicáveis:** As descrições do serviço devem ser publicadas e disponibilizadas em repositórios onde os clientes podem encontrar e usar o serviço.

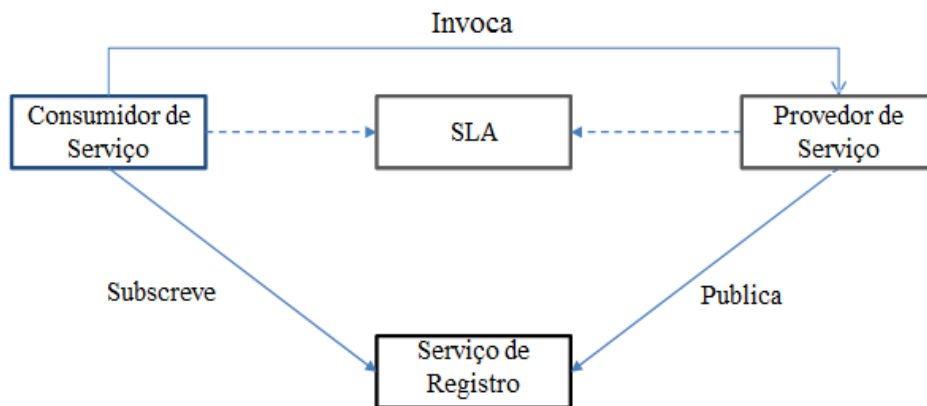
## 2.4 Arquitetura Orientada a Serviços

A Arquitetura Orientada a Serviços (SOA, sigla em Inglês de *Service Oriented Architecture*) é a implementação prática de um modelo de Internet de Serviços. Segundo (CHANNABASAVIAIAH et al 2004), uma SOA “é uma arquitetura de aplicativos na qual todas as funções são definidas como serviços independentes com interfaces bem definidas, que podem ser invocadas em sequências definidas para formar processos de negócio”. Ainda segundo (CHANNABASAVIAIAH et al 2004), esta definição diz que: a) Todas as funções organizacionais são definidas como serviço, tais como, funções do negócio, transações do negócio e funções do sistema; b) Todos os serviços trabalham como “caixas pretas”, isto é, componentes externos não precisam entender como os serviços funcionam e estão apenas preocupados com a sua contribuição no resultado final do serviço; c) As interfaces dos serviços são invocadas independentemente de sua localização, isto é, não importa se são interfaces de serviços locais ou acessadas remotamente, o tipo de conexão usado, o protocolo ou infraestrutura utilizada para a invocação dos serviços.

De forma geral, pode-se afirmar que uma Arquitetura Orientada a Serviços é uma forma de reorganização prévia da infraestrutura de suporte e das aplicações, em um conjunto de serviços interconectados acessíveis através de interfaces e protocolos padronizados. Esta abordagem de arquitetura é particularmente interessante quando múltiplas aplicações que rodam em tecnologias e plataformas variadas precisam se comunicar (PAPAZOGLU, et al., 2006).



A arquitetura básica do SOA envolve o relacionamento entre três participantes: O Provedor de Serviço, o Consumidor do Serviço e o Serviço de Registro. Este relacionamento envolve basicamente as operações de publicação, de descoberta, contratação e invocação como mostra a Figura 4. O termo SLA significa *Service Level Agreement* e será abordado no Capítulo 4.



**Figura 4:** Arquitetura básica do SOA.

O Provedor de Serviço é a entidade que recebe e executa as solicitações dos clientes. Pode ser um aplicativo legado ou ainda um componente ou entidade de outro *software*, que publica no serviço de registro sua descrição, tais como, funcionalidades funcionais (os serviços ou funções que a aplicação deve ser capaz de fornecer) e não-funcionais (os aspectos gerais da aplicação ou serviço como custo, performance, confiabilidade entre outros). O Consumidor do Serviço ou Cliente pode ser uma aplicação, serviço ou entidade de *software* que precisa de um determinado serviço; e o Serviço de Registro ou *Broker*, que é uma terceira entidade responsável por manter as publicações das descrições de serviço.

Como mostra ainda a Figura 4, existe também o SLA ou Contrato de Serviço, que especifica as condições de interação entre os provedores de serviço e clientes. Este contrato especifica, por exemplo, os formatos das mensagens, pré-condições e pós-condições de uso do serviço, e também pode descrever os atributos não funcionais dos provedores de serviço.

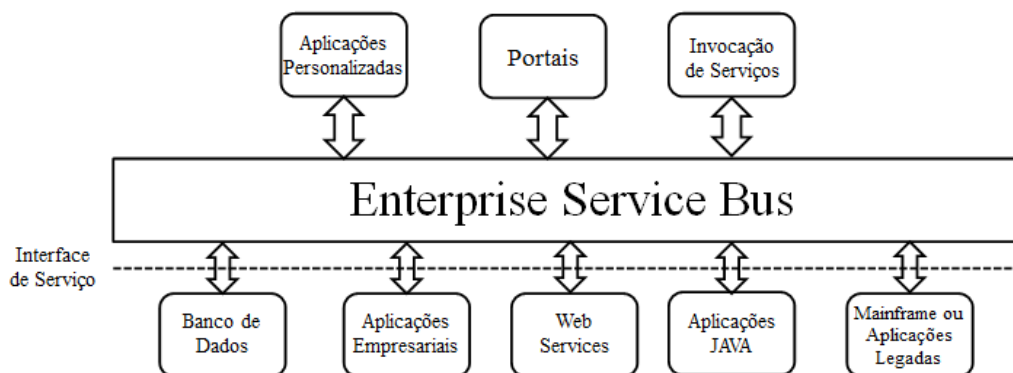
### 2.4.1 *Enterprise Service Bus (ESB)*

O ESB pode ser considerado o *backbone* da infraestrutura SOA. O ESB é o responsável por permitir a comunicação entre os consumidores e os provedores de serviço. Esta comunicação embora pareça simples, envolve varias tarefas como, fornecimento de conectividade, transformação dos dados, roteamento, monitoramento, registro e questões de segurança e confiança (JOSUTTIS, 2007).

Segundo (PAPAZOGLU et al., 2006), o ESB é “uma infraestrutura de mensagem baseada num padrão aberto projetada para permitir a implementação, implantação e gestão de soluções baseadas no SOA”. Um ESB contém ferramentas que permitem a comunicação entre aplicações utilizando protocolos padrões da rede, resolvendo os problemas de interoperabilidade entre aplicativos executados em diferentes plataformas e usando formatos de dados incompatíveis.

Como se vê, a principal função do ESB é proporcionar a interoperabilidade. A transformação dos dados é uma das partes fundamentais para garantia da interoperabilidade entre as aplicações, devido às diferentes plataformas e linguagens de programação. Para resolver esta questão, normalmente, usa-se a abordagem de se introduzir um formato específico para todas as plataformas individuais. Nas plataformas *web services*, por exemplo, esse formato é geralmente o SOAP.

A Figura 5, adaptada de (PAPAZOGLU & VAN DEN HEUVEL, 2007), mostra, simplificadamente, a arquitetura de um ESB que integra várias aplicações, como, sistemas legados, *web services*, entre outros.



**Figura 5:** Exemplo da função de um ESB no SOA (PAPAZOGLU & VAN DEN HEUVEL, 2007).

Como se pode ver na Figura 5, a interface fornecida pelo ESB às diversas aplicações e sistemas é a chave para a interoperabilidade do SOA. Ela define parâmetros, como por exemplo, segurança, confiabilidade, necessários ao SOA. No entanto, tecnicamente e conceitualmente, os ESB podem diferir de forma significativa. Por exemplo, uma solução SOA pode não envolver qualquer ferramenta específica de um ESB, ou seja, apenas o uso de um protocolo de comunicação pode ser suficiente. Neste caso o ESB pode delegar uma série de tarefas para os provedores e os consumidores dos serviços. Portanto, o uso ou não de todas as ferramentas disponibilizadas pelo ESB depende de cada solução SOA (PAPAZOGLU, et al., 2006).

## 2.5 Benefícios da Implementação do SOA

O SOA permite o desenvolvimento de novos aplicativos dinâmicos que abordam questões de negócios que são fundamentais para o crescimento e competitividade das empresas. Ela fornece uma estrutura através do qual é possível simplificar a criação e gestão de sistemas e aplicativos integrados, de forma a alinhar as vantagens da tecnologia de informação com o modelo de negócio. Segundo (MODI, G., 2004), os principais benefícios da implementação do SOA para as empresas são:

**Melhoria no processo de tomada de decisão:** Ao agregar às aplicações um conjunto de informações e serviços relativos ao negócio, a SOA permite que se tomem decisões mais precisas e mais abrangentes.

**Aplicações mais produtivas e flexíveis:** Permite que as empresas aproveitem as vantagens das tecnologias existentes, incluindo sistemas e aplicações legadas, sem a necessidade de soluções proprietárias.

**Desenvolvimento de aplicações de forma rápida e rentável:** Projetos de serviços padronizados permitem criar um repositório de serviços reutilizáveis que podem ser combinados e compostos em outras aplicações de acordo com a necessidade de novos negócios.

**Aplicações seguras e gerenciáveis:** O fornecimento de infraestrutura comum para desenvolvimento seguro de serviços torna mais fácil a adição de novos serviços e capacidades que mapeiam processos críticos aos negócios.

## 2.6 Negociação em Ambientes Orientados a Serviço

Tradicionalmente, *softwares* ou aplicações são especificados e implementados em termos de um único conjunto de requisitos funcionais, uniformes para todos os usuários, sejam eles individuais ou coletivos.

No modelo de computação orientada a serviços, um mesmo serviço pode ser entregue com qualidade, valor e condições diferentes, acomodando diferentes formas de uso para usuários ou clientes específicos.

No mundo global, altamente dinâmico, onde o mercado de oferta e demanda está em constante mutação, à entrega de serviços padrões para todos os usuários, sem levar em conta suas características, é um fator de desvantagem para o fornecedor (ELFATATRY & LAYZELL,2004). Para evitar tal desvantagem, a negociação no momento de solicitação

de um serviço, é a melhor solução. Em outras palavras, a negociação entre prestadores de serviço e os clientes ou consumidores, é a chave para o modelo de orientação a serviço.

### 2.6.1 Característica do Mercado de Serviços

O mercado é o lugar onde compradores e vendedores se reúnem para a troca de valores (bens ou serviços). Os elementos básicos de um mercado (digital ou físico) são os participantes, os produtos trocados e as regras do mesmo (ELFATATRY & LAYZELL, 2004).

O mercado é o elemento básico do modelo de orientação a serviço, pois é nele onde prestadores, consumidores e uma terceira entidade, como por exemplo, um *broker*, anunciam, solicitam e negociam serviços. Segundo (ELFATATRY & LAYZELL, 2004) as principais características de um mercado para o modelo SOC são:

**Escala:** Escalabilidade é uma questão crítica no ambiente SOC, pois este modelo requer serviços otimizados para atender as necessidades específicas dos consumidores, que serão muitos nas redes futuras. Este mercado é potencialmente grande, caracterizado por um número grande de pequenos prestadores de serviço. Para o consumidor obter o melhor serviço, significa negociar com todos os fornecedores possíveis do serviço requerido, o que significa um desafio para a escalabilidade dos sistemas.

**Tempo:** Neste modelo, os serviços são negociados em cada solicitação. Isto implica que as negociações devem ser concluídas em um período de tempo muito curto, para não afetar a eficiência da aplicação. (ELFATATRY & LAYZELL, 2004) defendem que, uma forma de minimizar este problema de negociação a cada solicitação é o estabelecimento de um contrato que cubra a possibilidade de utilização futura do serviço.

**Não conhecimento:** Nos ambientes SOC os negociadores (fornecedores ou consumidores) não precisam ter um perfeito conhecimento sobre a outra parte ou o ambiente em que serviço está inserido. Isto implica na formulação de regras de confiança,

que podem ser feitas internamente pelos negociadores, ou por entidades criadas especialmente para tratar de questões relacionadas com a confiança nestes ambientes.

**Volatilidade:** Com a entrada e saída livre dos provedores e consumidores, o mercado muda de forma contínua, influenciando na oferta e demanda dos serviços, e conseqüentemente no valor dos mesmos. Há ainda, a possibilidade dos serviços não estarem disponíveis, sempre que os consumidores requisitarem.

### 2.6.2 Negociação

Estabelecidas às características principais do mercado de serviços, volta-se ao processo de negociação, propriamente dito. No SOC este processo pode ser dividido em três fases: Pré-negociação, Negociação e Entrega do serviço, como ilustra a Figura 6.

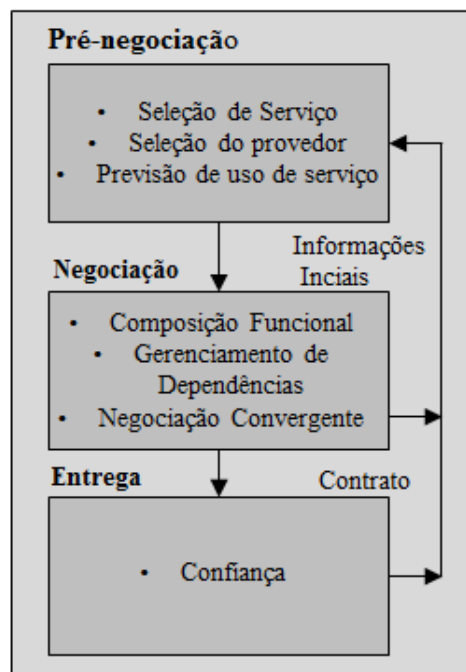


Figura 6: Fases da negociação de serviços.

#### Pré-negociação

É a fase que se ocupa na produção de informações para a fase principal de negociação. A pré-negociação pode ser dividida nas seguintes partes:

**Seleção do Serviço:** A seleção do serviço em ambientes SOC depende dos seus atributos funcionais e não funcionais. Normalmente, existe um atributo chave, que seu valor é considerado dominante para a escolha dos outros atributos. Considere, por exemplo, um serviço de chamadas via Internet, que possui, entre outros, atributos como preço, qualidade de voz (medido em número de pacotes perdidos) e confiabilidade (medido pelo número de vezes que o serviço está disponível por semana). Se o preço for o atributo dominante, o consumidor que pagar o valor mais baixo por chamada pode estar sujeito a uma perda grande de pacotes, afetando, portanto a qualidade de sua ligação e disponibilidade do serviço. O atributo dominante deve levar em consideração todos os outros atributos para se alcançar o mínimo de satisfação do serviço (ELFATATRY & LAYZELL, 2004).

**Seleção do Provedor.** Num mercado potencialmente enorme, cada invocação de serviços pode envolver um novo provedor. No entanto, o consumidor pode manter uma pequena lista de fornecedores preferenciais. Tal lista é atualizada continuamente com base nas informações obtidas nas fases de negociação e entrega dos serviços.

**Previsão de Uso do Serviço:** Segundo (ELFATATRY & LAYZELL, 2004), em uma aplicação baseada em serviços onde a maioria ou todas as funcionalidades são entregues como serviços, a negociação resulta em uma sobrecarga de comunicação e pode afetar significativamente o seu desempenho. É possível prever as suas necessidades e temporariamente criar contratos que podem ser assinados se e quando o serviço for invocado. Por exemplo, considere um serviço de processador de texto. A aplicação pode proativamente entrar em contato com fornecedores de serviço de verificação ortográfica e negociar os termos e condições para tal serviço assim que o usuário começar a editar um documento. Quando o usuário invoca o serviço de verificação ortográfica, o contrato é assinado.

## **Fase de Negociação**

Na sua forma mais simples, a negociação pode ser considerada como a troca de mensagens entre os provedores e consumidores de serviço, com o objetivo de chegar a um acordo, formalizado por um contrato. Cada contrato possui a descrição do serviço negociado, além de uma série de condições do serviço.

A negociação envolve, basicamente, as seguintes fases: composição funcional, gerenciamento de dependências e negociação convergente.

**Composição Funcional:** A composição de funcionalidades, num mercado amplo com múltiplos fornecedores de serviços, é uma questão essencial. A montagem da aplicação a partir de numerosas e diversas fontes requer, entre muitas outras coisas, levar em consideração os efeitos finais desta composição sobre a qualidade final da aplicação.

**Gerenciamento de Dependências e Incertezas:** Diferentemente dos negócios tradicionais executados normalmente de modo previsível e repetitivo, a composição de serviços complexos, em ambientes orientados a serviço, são diretamente afetados pela dinâmica destes ambientes e lidam, fortemente, com as questões de incerteza e dependência que devem ser tratados adequadamente durante o processo de negociação.

**Negociação Convergente:** Em ambientes onde o tempo é uma questão crítica, como nos orientados a serviço, as negociações devem ser finalizadas no mais curto espaço de tempo. Segundo (ELFATATRY & LAYZELL,2004) dois mecanismos podem ajudar para se chegar a este objetivo: protocolos de negociação e estratégias de negociação.

Protocolos de negociação são usados para sincronizar o comportamento das entidades participantes, de forma que atividades ou atitudes que possam gerar desperdício de tempo sejam minimizadas, proporcionando negociações mais rápidas e ágeis. Tanto



regras simples, como a definição do tempo de duração máxima da negociação, ou regras complexas, como o estabelecimento de penalidades para as entidades que tenham comportamentos que desperdicem tempo, ajudam a alcançar este objetivo.

A abordagem de incorporação de estratégias vislumbra duas técnicas que ajudam a conseguir uma melhor convergência na negociação: A abordagem heurística e a abordagem da argumentação. A heurística utiliza diferentes técnicas de aprendizagem que permite a uma entidade gerar ofertas de serviços, que muito provavelmente serão aceitas, com base em um histórico de negociação da outra entidade. A abordagem de argumentação depende do uso de técnicas que convençam a outra entidade a aceitar a oferta (ELFATATRY & LAYZELL, 2004).

### **Prestação do Serviço**

Quando o resultado da fase de negociação for um contrato, a próxima fase é a prestação do serviço, que envolve a implementação dos termos acordados. Segundo (ELFATATRY & LAYZELL, 2004) é a fase em que o cumprimento das promessas é testada e os valores de confiança são atualizados.

A confiança é uma questão importante na seleção do prestador de serviço, uma vez que, nos ambientes orientados a serviço, os consumidores podem decidir mudar de fonte sempre que for necessário. Desta forma, pode-se afirmar que a confiança é a percepção da capacidade de um provedor de serviço cumprir o que foi acordado no contrato de serviço. Nota-se, que a confiança é um conceito relativo, pois um provedor pode ser confiável em um contexto e não confiável em outro.

Os conceitos de confiança podem ser desenvolvidos internamente nas arquiteturas, com base na experiência progressa de um determinado cliente ou consumidor ou podem ser desenvolvidas por entidades terceirizadas. Exemplos de entidades terceirizadas de serviços de confiança são as empresas de classificação de crédito para pessoa física, como SPC e SERASA.

# Capítulo 3

## Trabalhos Relacionados

### 3.1 SOA4ALL

O *Service Oriented Architectures For All* (SOA4ALL) ou Arquitetura Orientada a Serviço para Todos, é um projeto do *European Seventh Framework Programme* (FP7).

Segundo os idealizadores deste projeto, apesar do sucesso inicial das pesquisas e desenvolvimento de arquiteturas orientadas a serviço (SOA) para apoio aos grandes prestadores de serviços, estas soluções permanecem, em sua maioria, específicas para cada prestador, distanciando-se cada vez mais do objetivo principal do paradigma de computação orientada a serviço, que é a criação de sistemas flexíveis e distribuídos globalmente, tornando-se assim, necessário repensar o SOA.

O SOA4ALL é o resultado deste esforço, e tem como o objetivo a materialização de um mundo onde bilhões de entidades expõem ou consomem serviços ou parte deles, através de tecnologias avançadas para a *web* (SOA4ALL, 2011). O projeto ambiciona a criação de um *framework* e infraestrutura de *software* visando à integração do SOA com quatro avanços técnicos complementares e revolucionários que emergiram nas últimas décadas (*web*, *web 2.0*, *web* semântica e tecnologias cientes do contexto), tal como ilustra a Figura 7.

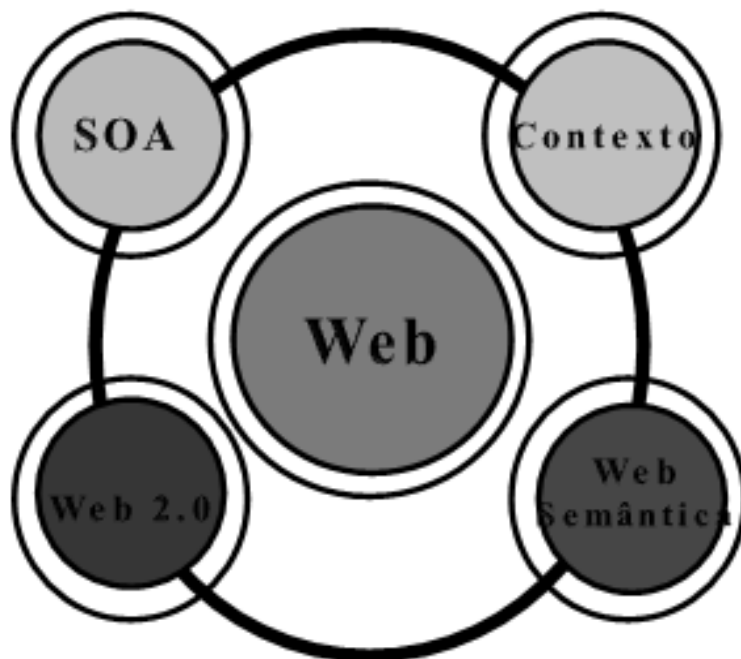


Figura 7: Tecnologias incorporadas pelo SOA4ALL (SOA4ALL, 2011).

### 3.1.1 Incorporação dos Princípios da *Web* no SOA

Com o objetivo de permitir que o SOA se expanda fora dos limites dos grandes prestadores de serviço, o projeto adota e expande os princípios que fizeram da *web* o maior avanço tecnológico mundial. Assim os princípios *design* do projeto são:

- Abertura: Introduce o conceito de *prosumer*, em que todos os usuários podem contribuir como provedores ou consumidores, independentemente de ser uma grande empresa ou simples usuários.
- Usabilidade: Permite o oferecimento de interfaces simples para todos os usuários, abstraindo-se assim os detalhes técnicos, possibilitando que os serviços estejam disponíveis e possam ser usados independentemente de sua plataforma subjacente, ou seja, plataforma de implementação em *software*.

- Interoperabilidade: Providenciada pela integração das diferentes soluções proprietárias e sistemas legados através de padrões e protocolos abertos.
- Descentralização: permitir que os serviços sejam criados, modificados e retirados de operação de forma descentralizada, sem a necessidade de uma autoridade central.
- Publicação: Em vez de uso de mensagens direcionadas, o projeto prevê o uso de publicações para a comunicação entre os provedores de serviço e os clientes.

### **3.1.2 Incorporação da *Web Semântica* no SOA**

Operações como, descoberta, seleção, composição, negociação e invocação de serviços, em ambiente com bilhões de serviços precisam de uma infraestrutura robusta e autogerenciável. Desta forma, é necessário fornecer descrições semânticas para os serviços disponíveis, criar localizadores de serviços com base nestas descrições e criar, ainda, mecanismos de mapeamento para permitir a invocação destes serviços. É importante que as descrições e serviços estejam mapeados corretamente para permitir a localização e invocação dinâmica dos serviços (SOA4ALL, 2011).

### **3.1.3 Incorporação da *Ciência do Contexto* no SOA**

“Contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Entidade é uma pessoa, lugar ou objeto que é considerado relevante para a interação entre um usuário e uma aplicação incluindo os mesmos, o usuário e a aplicação.” (SOA4ALL, 2011). Esses conceitos são relevantes para qualquer iniciativa em SOA.

No SOA4ALL, existe o *Framework* de Adaptação Contextual do Serviço que fornece uma plataforma genérica de apoio à adaptação dos serviços nos diversos contextos, de acordo com suas informações relevantes. É importante observar que a informação chave para o contexto depende de cada serviço. Por exemplo, a localização do consumidor pode ser a informação contextual chave para alguns serviços (por exemplo, serviços de entrega de encomendas expressas) ou considerada uma informação totalmente irrelevante para outros (por exemplo, serviço de entrega de mensagens via Internet).

O SOA4ALL irá incorporar contextos no SOA como uma forma de personalizar o consumo e prestação de serviços em escala global, apartir da perspectiva do usuário e das expectativas do provedor. A mistura de tecnologias de semântica e de contexto na infraestrutura SOA4ALL será um fator essencial de adaptação dinâmica dos serviços ao seu contexto de uso. Para facilitar a implantação de serviços semânticos em uma escala global, o contexto irá abranger uma série de aspectos que vão desde informações do sistema e sua localização, até contextos sociais e normas legais (SOA4ALL, 2011).

### **3.1.4 Incorporação da *Web 2.0* no SOA**

“*Web 2.0* é uma revolução na indústria de computadores causada pela mudança da Internet para uma nova plataforma, e uma tentativa de entender as regras para obter sucesso nesta nova plataforma<sup>5</sup>. Em outras palavras, a regra mais importante é desenvolver aplicativos que aproveitem a rede para se tornarem melhores, à medida que forem usados pelas pessoas, aproveitando a inteligência coletiva” (Tim O’Reilly, 2004). Por exemplo, criar aplicativos que aprendam com as pessoas a medida que sejam utilizados e divulguem o que aprenderam com seus pares.

Basicamente, sua idéia principal foi conectar diferentes usuários permitindo que eles troquem informações e agreguem valor na construção de aplicativos, redes e comunidades. SOA4ALL usará a tecnologia *Web 2.0* em sua interface de usuário,

---

<sup>5</sup> Novas arquiteturas para a troca e processamento de informações.

denominada *STUDIO SOA4ALL*, através do uso da tecnologia AJAX (sigla em Inglês de *Asynchronous Javascript and XML*), que é elemento central desta interface. Esta tecnologia permite ao projeto fornecer aplicações com interfaces que reagem instantaneamente às solicitações dos usuários, em vez de recarregar as páginas em cada clique (SOA4ALL, 2011).

Os princípios da *Web 2.0* serão usados para melhorar drasticamente a usabilidade do SOA visando o consumo e prestação de serviços e irão facilitar a criação dinâmica de novas funcionalidades compostas.

A Tabela 1, traduzida de (SOA4ALL, 2011), ilustra o resultado da integração entre a *Web 2.0* e o SOA que pode ser alcançada pelo SOA4ALL.

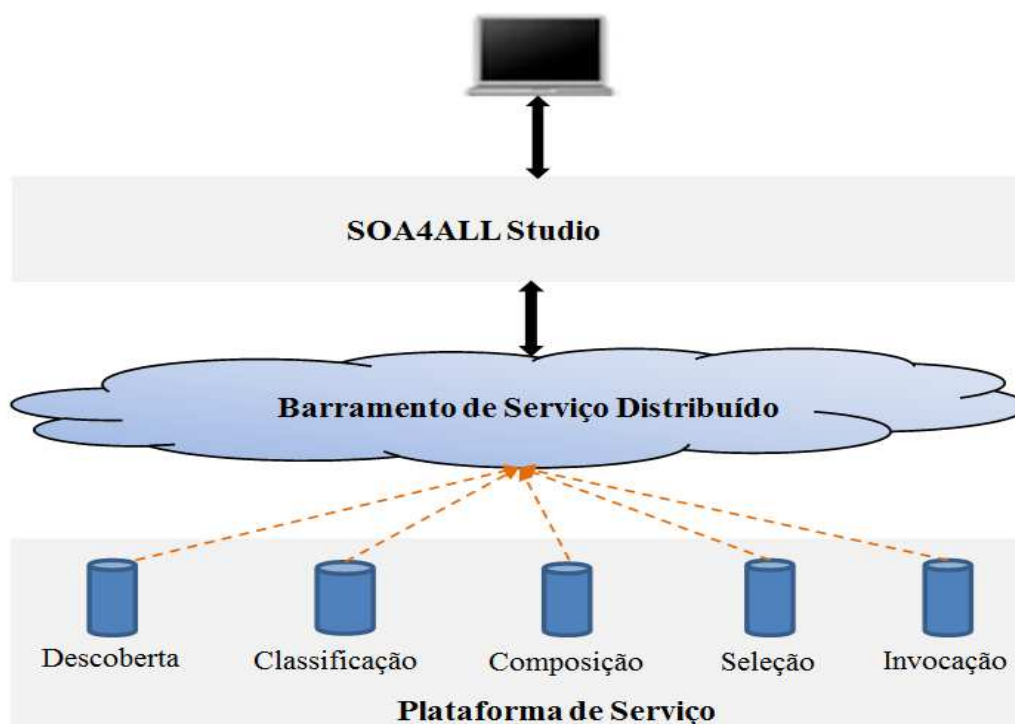
	<b>Web 2.0</b>	<b>SOA</b>	<b>SOA4ALL</b>
<b>Público alvo</b>	Consumidor	Empresas	Todos
<b>Controle</b>	Descentralizado	Centralizado	Descentralizado, mas com centralização local.
<b>Organização</b>	Não gerenciado	Gerenciado	Não gerenciado, mas com gerenciamento local.
<b>Tipos de dispositivos</b>	Diferentes tipos	Servidores	Diferentes tipos
<b>Número de dispositivos</b>	Muitos	Intermediário	Muitos
<b>Conectividade</b>	Global	Media	Global
<b>Demanda</b>	Enorme	Grande	Enorme
<b>Taxa de mudança</b>	Rápida	Lenta	Rápida

**Tabela 1:** *Benefícios da integração da web 2.0 e SOA para o SOA4ALL*

Embora a *Web 2.0* seja utilizada no projeto do SOA4ALL, principalmente na interação com os clientes, ela também terá um forte impacto na classificação, implantação, contexto, montagem e até na negociação dos serviços.

### 3.1.5 Arquitetura SOA4ALL

A arquitetura geral do projeto é construída em torno de três principais componentes: SOA4ALL *Studio*, Barramento de Serviço Distribuído e a Plataforma de Serviço. A Figura 8, adaptada de (PEDRINACI, C. et al., 2009) ilustra os componentes principais da arquitetura SOA4ALL.



**Figura 8:** *Arquitetura Simplificada do SOA4ALL.*

#### **Barramento de Serviço Distribuído**

É o componente central da arquitetura e atua como a infraestrutura central de comunicação e integração. O Barramento de Serviço Distribuído (DSB, sigla em Inglês de *Distributed Service Bus*) amplia as tradicionais tecnologias ESB, tornando os ambientes computacionais abertos, altamente distribuídos e de grande escalabilidade, permitindo a implementação de registros de serviços distribuídos, a realização de integração de

barramentos e a melhoria dos protocolos de coordenação e comunicação por meio da introdução dos espaços semânticos (PEDRINACI, C. et al., 2009).

O espaço semântico é usado como uma memória compartilhada na arquitetura permitindo construir repositórios, como o acesso aos dados de monitorização, evoluindo os tradicionais barramentos para uma infraestrutura de publicação que permite uma comunicação assíncrona ainda não disponível nas atuais tecnologias abertas e acessíveis de ESB.

Resumidamente, pode-se afirmar que o DSB é projetado para prover uma infraestrutura central de serviços (integração, comunicação e armazenamento) no SOA4ALL. Este é um princípio importante para plataformas de prestação de serviços escaláveis, pois ele permite a comunicação eficiente e transparente, por meio do compartilhamento ou troca de qualquer tipo de dado entre entidades ou partes distintas (SOA4ALL 2011).

### **SOA4ALL *Studio***

É plataforma que permite a criação, provisionamento, consumo e análise dos serviços que são publicados para o SOA4ALL. É dividido em três componentes principais: A Plataforma de Provisionamento, que permite que o SOA4ALL oculte muito da complexidade da composição dos serviços e proporciona aos usuários notação semântica suficiente para a construção de serviços compostos sob medida; a Plataforma de Consumo, que é a interface dos consumidores ou aplicações cliente com os serviços. Esta plataforma permite-lhes formalizar suas necessidades, definir e refinar os objetivos usados para descobrir e invocar serviços que atendam suas necessidades. A Plataforma de Análise é a responsável por obter e processar as informações do subsistema de monitoramento do barramento, com o objetivo de extrair o significado destas informações (PEDRINACI, C. ET al., 2009).

### **Plataforma de Serviço**



Providenciam funcionalidades básicas relacionadas ao serviço, como descoberta, classificação, seleção, composição e invocação do serviço. Estas funcionalidades são usadas pelo SOA4ALL *Studio* para oferecer aos clientes a funcionalidade mais adequada, enquanto que a combinação de suas atividades são coordenadas através do DSB.

“O conjunto DSB, SOA4ALL *Studio* e Plataformas de Serviços torna a arquitetura inovadora, totalmente baseada na *web* e permite serviços habilitados pela *web*. O projeto do SOA4ALL permite a entrega de serviços globais a nível de barramento, permite o acesso a serviços na *web* via *Studio* e permite o processamento, o gerenciamento e a manutenção dos serviços através da Plataforma de Serviços” (SOA4ALL 2011).

### 3.2 SLA@SOI

O SLA@SOI (*Service Level Agreement at Service Oriented Infrastructures*) é um dos projetos afiliados ao FISO (*Future Internet Service Offer*) que é um dos *cluster* da iniciativa da União Européia para a construção da Internet do Futuro, o FIA (*Future Internet Assembly*). É um consorcio formado pela indústria, acadêmicos e institutos de pesquisa de toda Europa com o objetivo de pesquisar e criar tecnologias que possam incorporar infraestruturas cientes do SLA<sup>6</sup> para a economia de mercado (SLA@SOI , 2011).

O projeto de pesquisa SLA @ SOI visa proporcionar um marco importante para a evolução no caminho de uma economia orientada a serviços, onde serviços baseados em TI possam ser facilmente negociados como bens econômicos, ou seja, em condições bem definidas e confiáveis, e com custos claramente associados (WOLFGANG, et al., 2011).

---

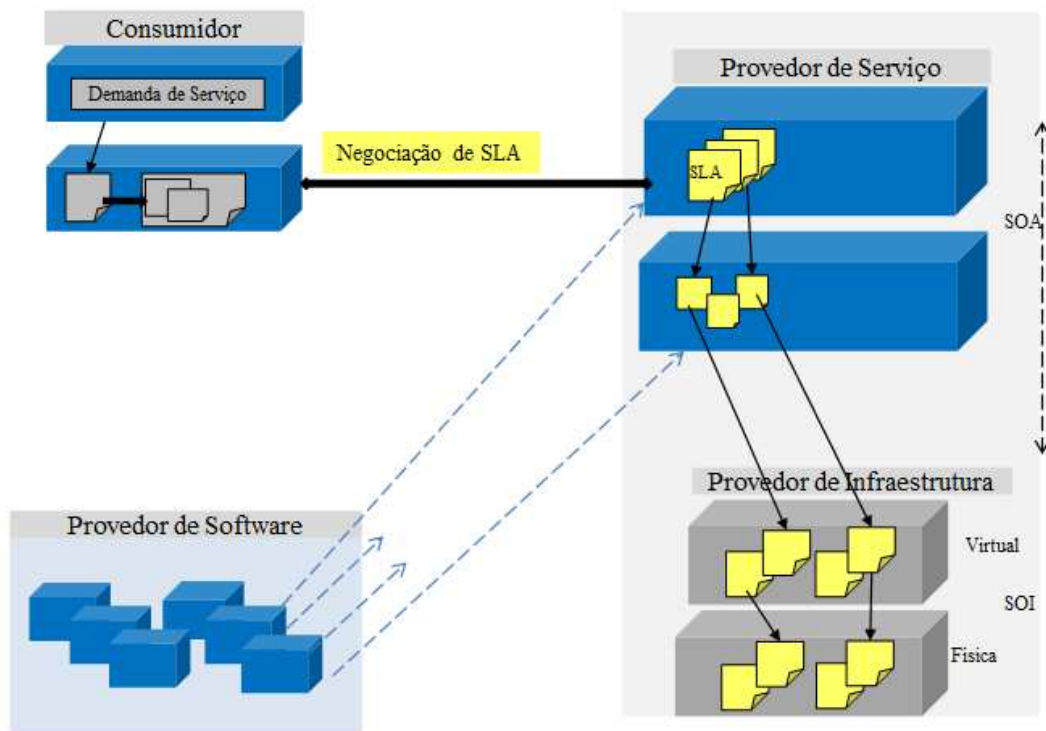
<sup>6</sup> O projeto SLA@SOI providência uma infraestrutura para a prestação de serviços de acordo com o SLA negociado, isto é, a infraestrutura deve ser capaz de dividir seus recursos entre os diversos serviços com base no SLA, permitindo que alguns serviços recebam mais ou menos recursos.

Este projeto tem como visão a “criação de uma infraestrutura de negócio orientada a serviços, fortalecendo de forma flexível e confiável a economia de serviços” (WOLFGANG, et al., 2011). Para alcançar estes objetivos, todos os processos envolvidos nesta infraestrutura devem possuir as seguintes características:

- **Previsibilidade e Confiabilidade:** Os requisitos de qualidade de serviço podem ser previstos e implementados em tempo de execução do mesmo.
- **Gerenciamento Transparente de SLA:** Acordos de nível de serviço, que definem as condições exatas em que os serviços serão prestados ou/e consumidos devem ser geridos com transparência em toda a cadeia de negócio e pilha de TI.
- **Automação:** Todo o processo de negociação de SLAs, execução e acompanhamento dos serviços serão automatizados, permitindo a execução de serviços de forma dinâmica e escalável.

Basicamente, a idéia do projeto é permitir que um prestador de serviço ofereça serviços com SLAs diferenciados, confiáveis e ajustáveis, e possam negociar estes SLAs de forma automatizada com os clientes.

Esta idéia gera o desafio de se criar uma infraestrutura orientada a serviços (SOI, sigla em Inglês de *Service Oriented Infrastructure*) para suporte ao gerenciamento consistente dos SLAs em todas as camadas de uma pilha de TI, e em todas as perspectivas das diferentes partes interessadas. A Figura 9 traduzida de (WOLFGANG, et al., 2011) apresenta uma visão simplificada de um processo de gerenciamento de SLA.



**Figura 9:** Exemplo de gerenciamento de um SLA.

A Figura 9 mostra que serviços podem ser compostos de outros serviços, fornecidos por outras entidades ou até mesmo por outros serviços. Assim, é necessário um mapeamento gradual dos requisitos de SLA de níveis superiores para os níveis inferiores, bem como a agregação das capacidades de nível inferior aos níveis mais elevados. Como mostrado na figura, o processo de gerenciamento global de um SLA inclui as diferentes partes interessadas, ou seja, clientes, provedores de serviços e provedores de infraestrutura.

### 3.2.1 Objetivos Funcionais

Segundo (WOLFGANG, et al., 2011) o principal objetivo funcional do *framework* de gerenciamento de SLA, proposto pelo projeto SLA@SOI, é fornecer uma solução genérica que:

- a) Suporte o gerenciamento de SLA em várias camadas.

b) Suporte diversos tipos de serviços (negócios, *software*, infraestrutura).

c) Cubra completamente todo o ciclo de vida dos serviços, com consistente interligação com o tempo do projeto, planejamento e aspectos gerenciamento.

d) Possa ser aplicada em domínio empresarial.

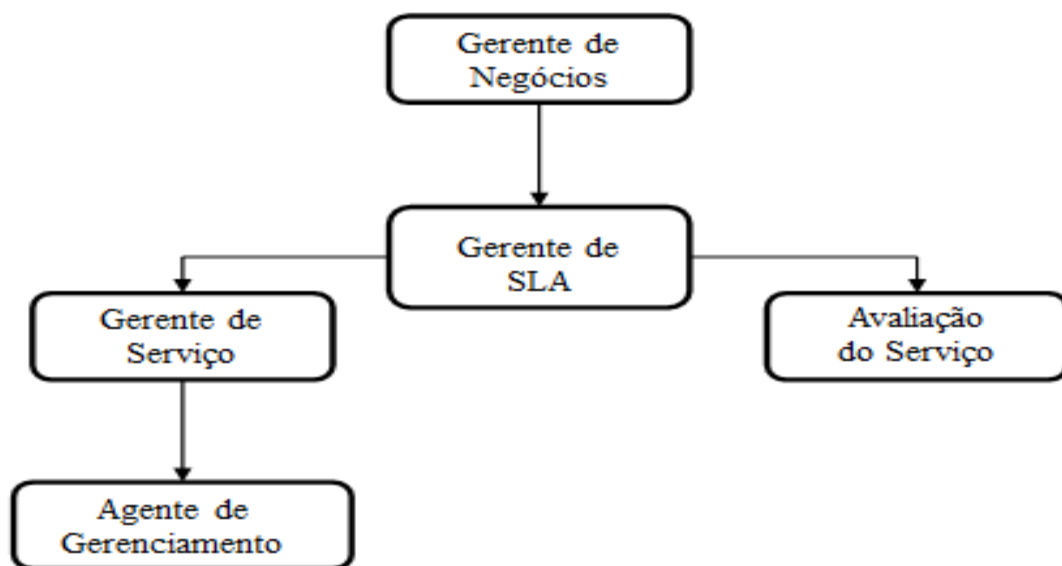
Para atingir esses objetivos, a arquitetura é baseada em três princípios de *design*. Primeiro, forte ênfase na separação clara entre gerenciamento de serviços e gerenciamento de SLA e apoio a uma estrutura de camadas de gerenciamento. Segundo, uma base sólida de meta-modelos comuns para SLAs e sua relação com os serviços. A construção de instâncias para serviços é um aspecto importante para suportar a invocação semântica entre os diferentes componentes do *framework*. Terceiro, escalabilidade e capacidade de adaptação são aspectos importantes a fim de suportar os diferentes domínios de uso (WOLFGANG, et al., 2011).

### **3.2.2 Gerenciamento de SLA**

O SLA@SOI, define o Gerenciamento de SLA como um sistema de gerenciamento de entregas de serviços, a fim de cumprir os requisitos de qualidade de serviço (QoS, sigla em Inglês de *Quality of Service*) especificados no SLA. O Gerenciamento de SLA cobre todas as fases do ciclo de vida do serviço: *Design* do modelo de SLA, assegurando que as garantias de QoS são realistas; *Negociação* do SLA, assegurando que as garantias acordadas de QoS são realizáveis; *Execução* do SLA, assegurando que as garantias de QoS estão sendo cumpridas satisfatoriamente; *Arquivamento* do SLA, garantindo que a experiência anterior estará disponível para futuros ciclos.

### 3.2.3 Principais Componentes da Arquitetura

Os principais componentes da arquitetura do projeto são: O Gerente de Negócios (BM, do Inglês *Business Manager*), o Gerente de SLA (SLAM, do Inglês *SLA Manager*), o Gerente de Serviços (SM, do Inglês *Service Manager*), o Agente de Gerenciamento (MA, do Inglês *Manageability Agent*) e o Serviço de Avaliação (SE, do Inglês *Service Evaluation*), como ilustra a Figura 10.



**Figura 10:** Componentes Principais do SLA@SOI.

A Figura 10 apresenta uma visão geral dos componentes principais da arquitetura e suas relações. O Gerenciador de Negócios é o componente principal, e é o responsável pelas informações relativas ao negócio e pelas decisões que vão orientar o negócio. Ele controla o Gerenciador de SLA, responsável pelos modelos de SLAs e pelos SLAs existentes. Usa o Gerenciador de Serviços para consultas de implementação de serviço e para provisionamento de atividades. O Gerenciador de Serviços é o responsável pelo gerenciamento da implementação do serviço que está sendo executado. Ele usa os Agentes de Gerenciamento para inicializar atividades de gerenciamento na hora de

execução do serviço. O Gerenciador de SLA conta com um Serviço de Avaliação para acompanhar os requisitos de qualidade prometidos.

Conforme (WOLFGANG, et al., (2011), o Gerenciador de Negócios é o responsável por impor todas as restrições no sistema, a fim de atender os objetivos e manter a relação entre o cliente e o provedor. Assim, ele obtém todas as informações necessárias, tais como preço (incluindo recompensas, promoções e descontos), informações sobre o perfil do cliente e as regras do negócio para que ele tome decisões ciente de custos/lucros. Os Gerenciadores de Negócios podem conter dados sensíveis que não podem ou não devem ser compartilhados entre os componentes. Dentre as várias funcionalidades destes, destacam-se:

- Busca e publicação de produtos.
- Gerenciamento de clientes e provedores de serviço.
- Negociação e estabelecimento de acordos com clientes e provedores de serviço.
- Notificação e aplicação de penalidades para clientes e provedores.

O Gerenciador de SLA é o responsável por gerenciar o conjunto de modelos de SLAs e SLAs em execução em seu domínio. As suas funcionalidades principais são:

- Busca e publicação de modelos de SLA.
- Negociação de SLAs, incluindo a conversão entre diferentes formatos de SLA.
- Planejamento, otimização e ajuste de SLAs.

O Gerenciador de Serviços é o responsável pelo gerenciamento dos elementos necessários para o instanciamento de um serviço. Ele possui informações sobre a estrutura de implementação e mantém registros dos serviços existentes. Suas funcionalidades são:

- Publicação da implementação do serviço.

- Manter um panorama dos serviços, incluindo os elementos necessários para instanciamento de uma implementação de serviço.
- Agendamento e reservas de instâncias de serviço.
- Provisionamento de serviços efetivamente.

O Agente de Gerenciamento atua como *gateway* para os recursos atuais. Agentes de Gerenciamento podem existir por recursos, por instâncias de serviço ou pelo conjunto destes. Suas funcionalidades são:

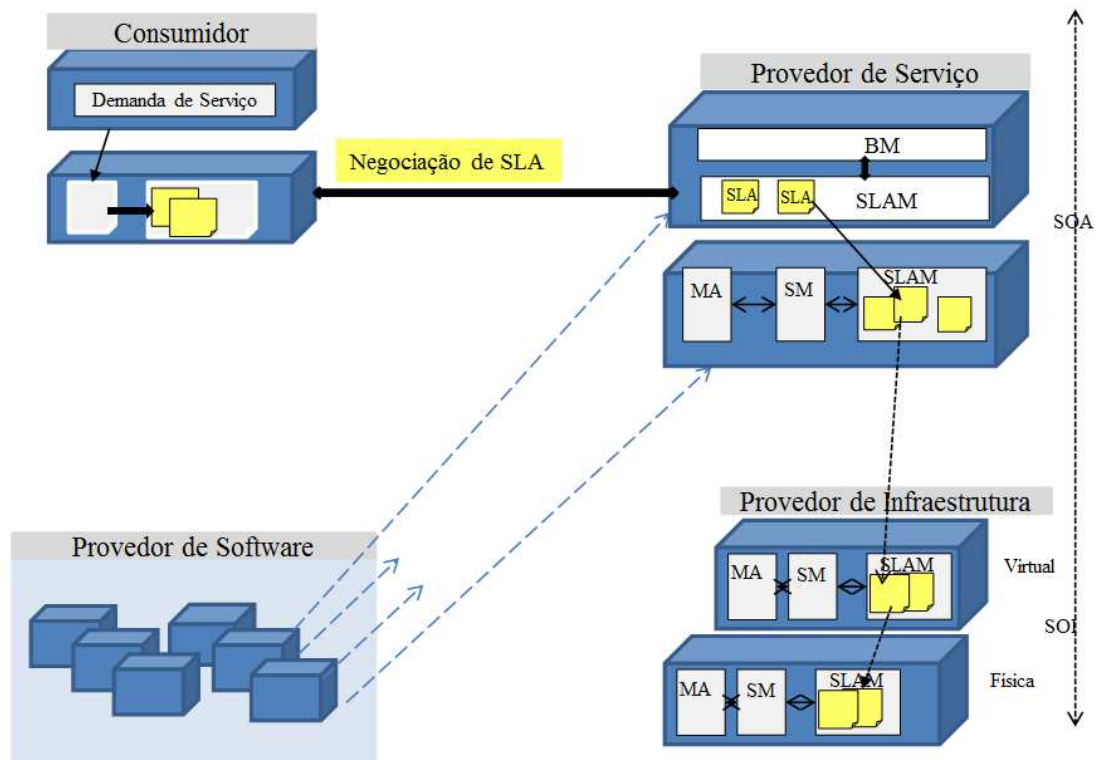
- Detecção e monitoramento do estado das instâncias de serviço e dos recursos.
- Busca e execução de ações de gerenciamento.

Por fim, para apoiar as ações proativas de gerenciamento, em todos os níveis, o *framework* possui o Avaliador de serviços. Ele se baseia em informações de apoio (tempo de *design*, tempo de execução ou histórico) sobre as características de qualidade do serviço. Ele providência uma avaliação da qualidade do serviço antes do serviço ser instanciado (WOLFGANG, et al., 2011).

Apesar de todos os componentes terem suas responsabilidades distintas, eles também precisam ter algum entendimento geral da arquitetura. Assim, os tipos de serviço devem ser entendidos pelos Gerenciadores de Serviço e pelos Avaliadores de Serviços; a identidade da implementação de serviço, também, deve ser compreendida pelos Gerenciadores de Serviço e Avaliadores de Serviço, embora ambos os componentes possam invocar modelos de dados diferentes a fim de lidar com a implementação do serviço (WOLFGANG, et al., 2011).

Os termos do SLA e os instrumentos de monitorização disponíveis devem ser compreendidos por Gerenciadores de SLA e Gerenciadores de Serviço, e o processo de negociação do SLA devem ser comumente compreendido pelos Gerenciadores de Negócio e Gerenciadores de SLA (WOLFGANG, et al., 2011).

Finalmente, mostra-se na Figura 11 a relação entre os principais componentes e a visão de gerenciamento apresentada na Figura 9.



**Figura 11:** Exemplo de gerenciamento de SLA com os componentes da arquitetura.

A Figura 11 ilustra, exatamente, as interações dos componentes da arquitetura, no processo de gerenciamento de SLA. Note, que o Avaliador de Serviços não foi mostrado a fim de reduzir a complexidade da mesma. No entanto, este componente pode ser associado a cada um dos Gerenciadores de SLA apresentados.



# Capítulo 4

## Generalized Indirection Resolution System-GIRS

Neste capítulo, apresenta-se o *Generalized Indirection Resolution System* (GIRS), um esquema proposto por (MAGALHÃES, B., 2011), para tratar indireções em propostas de Internet do Futuro, em especial nas Redes Centradas em Informação (JACOBSON, et al., 2009). Esta proposta utiliza o conceito de *Distributed Hash Tables* (DHTs) para a formação, manipulação e armazenamento dos mapeamentos entre os atributos da rede (nomes, identificadores, localizadores e descritores) tanto para os objetos de conteúdo, quanto para os elementos responsáveis pelo armazenamento destes objetos.

### 4.1 Desacoplamento da Identificação da Localização

A Internet criada nos anos 70 tinha como principal objetivo a interconexão de *hosts* fixos. Atualmente, ela é usada como fonte de informações, como meio de comunicação, como local de interação social através das redes sociais, e para muitos outros fins. Deste modo, sua arquitetura original tem sido constantemente “remendada” para dar suporte a todas estas novas aplicações.

O *Internet Protocol* (IP) (POSTEL, 1981) sobrecarrega as funções de localização e identificação dos *hosts*, ou seja, o endereço IP serve para as duas coisas. Esta sobrecarga causa inúmeros constrangimentos para a Internet atual, principalmente para suporte à mobilidade e a realização de *multihoming*.

Com o aumento exponencial dos dispositivos móveis com acesso a Internet, o suporte à mobilidade é um dos requisitos básicos para a rede atual e para as redes de nova geração. A mobilidade garante que os usuários possam se mover na sua rede local ou fora

dela, isto é, mudar de rede, sem que haja perda de comunicação. O *multihoming* possibilita a existência de várias conexões simultâneas para acesso à Internet, com o objetivo de oferecer redundância e melhorar a escolha do provedor.

Nos últimos anos, o desacoplamento entre a identificação e a localização tem aparecido como uma solução importante para estes problemas. Assim, inúmeras propostas foram feitas na literatura, como, o *Locator ID Separation Protocol* LISP (FARANACCI, et al., 2011), o *Host Identity Protocol* HIP (MOSKOWITZ & NIKANDER, 2006), e o *Mobility and Multihoming Supporting Identifier Locator Split Architecture* MILSA (JINLI, et AL., 2008). Este desacoplamento é considerado uma questão essencial nos projetos de arquitetura de redes de nova geração como, por exemplo, no projeto Akari (HARAI, et AL., 2007).

#### **4.2 Redes Centradas na Informação**

A rede atual foi projetada com base no paradigma *host-centric*, isto é, centrada na interconexão de *hosts*, em que os terminais da rede estavam no centro do projeto. Deste modo, todas as funções complexas foram movidas para os terminais da rede, mantendo o núcleo o mais simples possível.

Hoje a rede é usada principalmente para transferência de conteúdos e/ou informações, e devido a este fato, (JACOBSON, et al., 2009) defende a mudança do paradigma *host-centric* para o *information-centric* (centrado na informação), as chamadas redes de terceira geração. Segundo (JACOBSON, et al., 2009) a primeira geração tratava da interligação através de fios da infraestrutura fixa; a segunda foi à fase da popularização da Internet através da *World Wide Web*; e a terceira geração são as redes centradas na informação. Nesta geração, todas as ações da rede têm como objetivo a troca e o processamento da informação, independentemente da sua localização.

As arquiteturas *Network of Information-NetInf* (AHLGREN, et al., 2010) (OHLMAN, et al., 2009), fomentada pelo FP7 (*European Union Framework Programme*),

o *Content Centric Network-CCN* (JACOBSON, et al., 2009), proposta pelo PARC (*Palo Alto Research Center Incorporated*), e o *Publish/Subscribe Internetworking Routing Paradigm-PSIRP* (AIN, et al., 2008), também fomentada pelo FP7, estão entre as principais abordagens para as redes centradas na informação.

### 4.3 Indireções

A arquitetura original da Internet foi projetada para oferecer comunicação ponto a ponto entre *hosts* fixos. Com sua evolução para uma infraestrutura de comunicação global, novas funcionalidades, tais como, mobilidade, *multicast* e *anycast*<sup>7</sup> têm sido implementadas.

Esta implementação tem encontrado limitações, porque o paradigma de comunicação ponto a ponto, que assume que a comunicação se dá entre um transmissor e um receptor fixos na rede e que se conhecem mutuamente, não é apropriado para estas novas funcionalidades. Por exemplo, a mobilidade remove a suposição de que os *hosts* são fixos e o *multicast* remove o pressuposto de que existe apenas um receptor.

Para contornar estas limitações usa-se uma técnica simples, mas poderosa: Indireção. Esta técnica assume que um ponto de indireção físico ou lógico se interpõe entre o transmissor ou transmissores e o receptor ou receptores, retransmitindo o tráfego entre eles. Comunicando-se através do ponto de indireção e não diretamente, as entidades envolvidas na comunicação podem abstrair a localização e o número de transmissores e receptores (STOICA, 2002).

Todas as propostas de solução para o desacoplamento entre a identificação e localização, assim como as propostas de redes centradas em informação utilizam o conceito de indireção.

---

<sup>7</sup> *Anycast* consiste em rotear um conteúdo a partir da fonte mais próxima.

#### 4.4 Distributed Hash Tables (DHTs).

As DHTs são uma classe de sistemas distribuídos descentralizados que prestam serviço de pesquisa baseados em tabelas *hash* distribuídas. Pares <chave, valor> são armazenados nas HTs e qualquer nó participante pode eficientemente recuperar um valor associado a uma dada chave. Na DHT, a responsabilidade de manter o mapeamento entre as chaves e valores é distribuída entre os nós, de forma que mudanças no conjunto de nós participantes causam o mínimo de desordem. Isso permite que as DHTs se estendam a um número extremamente grande de nós e gerenciem chegadas, saídas e falhas dos nós (DISTRIBUTED HASH TABLE, 2011).

Para melhorar a escalabilidade, a responsabilidade pela manutenção das tabelas de mapeamento é distribuída entre os nós participantes. Esta distribuição de mapeamentos torna as DHTs bastante eficientes, pois os nós não precisam conhecer todos os seus vizinhos, e o alcance a um nó qualquer é feito com um número relativamente pequeno de saltos na topologia lógica da rede de HTs, variando conforme o algoritmo de roteamento utilizado nesta topologia.

As DHTs formam um infraestrutura que pode ser usada para criar outros serviços complexos, tais como sistemas de arquivos distribuídos, compartilhamento de arquivos *peer-to-peer*, sistemas de distribuição de conteúdo, web *caching* cooperativo, *multicast*, *anycast*, serviços de nome de domínio e mensagens instantâneas (DISTRIBUTED HASH TABLE, 2011). Portanto, o uso desta tecnologia para criar sistemas de mapeamento, devido as suas propriedades inerentes de autogerenciamento, auto-configuração, robustez, tolerância a falhas e alta escalabilidade, é bastante promissora e desejável (LIU, et al., 2009).

Segundo (HANKA, et al., 2008), o mecanismo de tabelas DHTs é o conceito ideal para ultrapassar as maiores limitações da Internet atual, em especial nas seguintes áreas:

- Escalabilidade: Uso das DHTs para suportar o roteamento.
- Descentralização: Mecanismos distribuídos de buscas, gerenciamento e localização de entidades.
- Mobilidade: facilitado pelo mapeamento dinâmico entre os identificadores e localizadores.

A Figura 12, redesenhada de (HANKA, et al., 2008), resume as principais vantagens da tecnologia DHT.



**Figura 12:** Principais vantagens das DHTs (Hanka, et al., 2008).

Ela permite uma maior escalabilidade, descentralização e mobilidade, melhorando a pesquisa e mapeamento dos *hosts*. Como os *hosts* nas arquiteturas baseadas em DHTs são cientes da localização de alguns dos seus pares, eles otimizam o tráfego de informações na rede lógica e consequentemente diminuem os custos e o atraso do mesmo. As propriedades de auto-organização desta tecnologia diminuem o custo de manutenção e operação, melhorando a resiliência e permitindo a autocura devido à natureza distribuída dos *hosts*.



Este sistema suporta naturalmente a mobilidade, pois os localizadores dos *hosts* são obtidos através dos seus identificadores, e como o identificador do *host* pode ser associado a vários localizadores, este sistema também suporta o *multihoming*.

#### 4.5.1 Estrutura de Mapeamento para o GIRS

Nomes, identificadores, localizadores e descritores dos objetos de conteúdo e dos *hosts* são relacionados através de mapeamentos que são registrados nas DHTs. A Figura 14 ilustra a estrutura do mapeamento para o GIRS.

<b>Chave (512 bits)</b> (Hash da fonte do mapeamento)
Tipo do Mapeamento
Lista de Valores Destino do Mapeamento
Tempo de Vida

**Figura 14:** Estrutura dos mapeamentos para o GIRS (MAGALHÃES, B., 2011).

Como mostra a Figura 14, a chave é o resultado do *hashing* do objeto fonte de mapeamento, e ela nos dá o valor de entrada para as pesquisas de valor associado nas DHTs. O tipo de mapeamento representa a classe e o estilo de resolução entre as entidades envolvidas na pesquisa. A lista de valores é o valor resultante de um mapeamento associado a uma dada chave e o Tempo de Vida é usado para determinar a expiração e a atualização dos mapeamentos.

#### 4.5.2 Estrutura do Descritor para o GIRS

Um descritor é um objeto que possui todas as especificações do objeto fonte do mapeamento. Os atributos dos descritores dependem de cada objeto, não existindo regras

para estes mesmos atributos, que podem representar informações de classificação do objeto (imagem, som, vídeo), formato do arquivo (JPEG, MP3), além de atributos do mundo real, tal como a posição em um sistema de posicionamento global (GPS do Inglês *Global Positioning System*). É importante que as especificações dos descritores estejam em linguagem natural, para que os usuários e aplicações os localizem com base na semântica dos descritores. A Figura 15 mostra a estrutura de um descritor para este sistema.

ID
Lista de Identificadores de Padrões Relacionados
Lista de Atributos
Localizador do Descritor
Tempo de Vida - TTL (Time to Live)

**Figura 15:** Estrutura do descritor para o GIRS (MAGALHÃES, B., 2011)

Como mostra a Figura 15, todo descritor também possui um ID, resultado do *hashing* do seu padrão binário; uma lista ou apenas um identificador dos objetos descritos; uma lista de atributos, responsável pela apresentação do conjunto de atributos relacionados a este objeto; o localizador do *host* que hospeda este descritor; e um tempo de vida para parâmetros de atualização e expiração (MAGALHÃES, 2011).

#### 4.5.3 Mecanismos de Resolução

Segundo (MAGALHÃES, 2011) o mecanismo de resolução de indireções do GIRS subdivide-se em três blocos: Mecanismo de Busca (MB), que faz a pesquisa e recuperação, com base na semântica, metadados e atributos das entidades; Sistema de Resolução de Nomes ou *Name Resolution System* (NRS), que faz o mapeamento entre identificadores e localizadores dos objetos; e o *Split Host* que faz o mapeamento entre os identificadores e os localizadores de *hosts*.



#### 4.5.4 Escalabilidade

A escalabilidade é um dos requisitos principais para os projetos de Internet do Futuro. Estas redes devem crescer sem comprometer o seu projeto original, o que não aconteceu com a rede IP, possibilitando a introdução de novas funcionalidades à medida que novas tecnologias forem surgindo.

Devido às vantagens das DHTs, como por exemplo, a descentralização, seu uso torna-se bastante promissor em redes centradas em informação. (MAGALHÃES, 2011) frisa que com a tecnologia DHT podem-se formar topologias sobrepostas independentes das redes físicas existentes, podendo assim criar diferentes sistemas autônomos ou ainda criar uma organização hierárquica de acordo com a necessidade do escopo a ser alcançado.

#### 4.5.5 Localização e Recuperação de Conteúdo

A Figura 16 ilustra um estudo de caso genérico apresentado por (MAGALHÃES, 2011) que utiliza a arquitetura proposta para o processo de localização e recuperação de um objeto de conteúdo no contexto das redes centradas na informação.

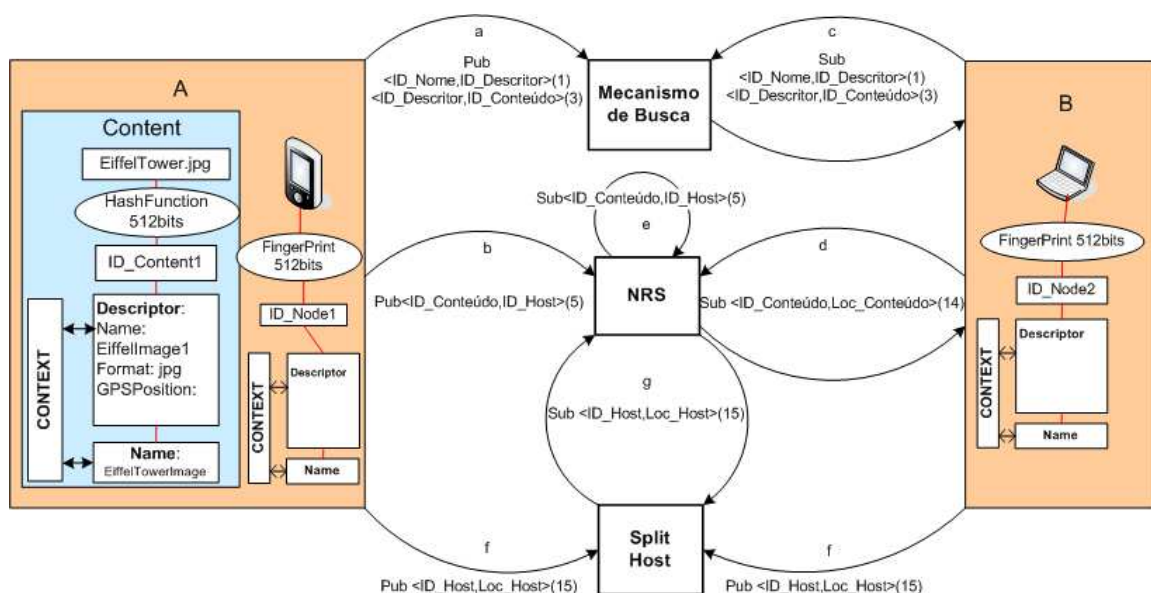


Figura 16: Localização e recuperação de conteúdo usando o GIRS (MAGALHÃES, B., 2011).

Neste exemplo, de forma resumida, (MAGALHÃES, 2011) apresenta um cenário, no qual um nó B está interessado numa foto da torre Eiffel, disponível em outro nó, o nó A. Este nó, que possui a foto, publica o nome e o descritor junto ao mecanismo de busca (passo a), e informa a posse do conteúdo ao sistema de resolução de nomes (NRS) (passo b). Desta forma, o nó assinante, procura pelo identificador único do conteúdo no sistema de busca baseando-se no nome deste e nos atributos do descritor (passo c). Com base nos resultados desta busca, o usuário elege um resultado com um identificador para o conteúdo interessado e o mesmo é usado como parâmetro para a assinatura do conteúdo no NRS (passo d).

Ainda segundo (MAGALHÃES, 2011), no NRS é feito o mapeamento entre o identificador do conteúdo e o nó responsável por esse conteúdo (passo e). Assim o NRS pode transmitir diretamente a mensagem Sub <Conteúdo, ID-Conteúdo> para o nó publicador através do encaminhamento direto. Anterior a este passo, é necessário que o NRS interaja com algum mecanismo de desacoplamento de identificação e localização de *hosts*, a fim de fazer o mapeamento dinâmico entre o identificador e o localizador do nó publicador (passo f e g) e desta forma, suportar a mobilidade e *multihoming* dos *hosts*.

Como se nota, este sistema de resolução de indireções possibilita o desacoplamento dos identificadores e localizadores das entidades, suportando naturalmente a mobilidade e *multihoming*. Possibilita a identificação única das entidades, o uso de nomes e descritores destas entidades em linguagem natural, com base no seu contexto e semântica, e a independência da tecnologia de conexão, podendo ser usada em redes atuais ou futuras.

# Capítulo 5

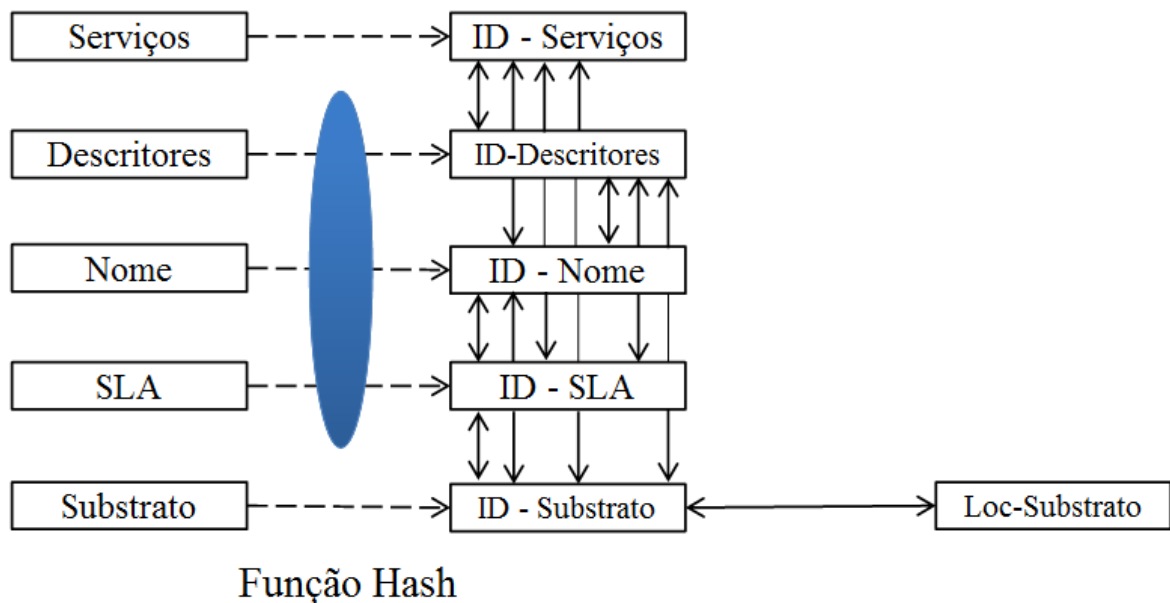
## Internet de Serviços baseada no *Generalized Indirection Resolution System* (GIRS)

Na rede atual, os serviços devem ser registrados pelo provedor de serviços e devem ser analisados pelo usuário antes de serem invocados. Na Internet de Serviços, os serviços são anunciados de forma semelhante aos conteúdos, nas redes de informação, e ao invocar um serviço, ele é descoberto pelo nome e não pelo endereço de um servidor específico. Assim, a solicitação de serviço é feita por uma mensagem de interesse para o elemento da rede que suporta o serviço de descoberta.

Neste trabalho, propomos o uso do GIRS para apoio a Internet de Serviços. Assim sendo, pretende-se: a) Utilizar o GIRS para apoio aos serviços como um todo e não apenas a localização e recuperação de conteúdo, como originalmente proposto por (MAGALHAES, B., 2011). Com isto, o GIRS passa a suportar a localização, recuperação e processamento de conteúdos antes e/ ou depois destes chegarem aos usuários; b) Ampliar o uso dos nomes, não só para conteúdos, mas também para os serviços invocados; c) tratar Conteúdo e Serviços como entidades distintas, implicando na necessidade de identificação para ambas as entidades; d) Obter um esquema de nomeação para Serviços, assim como já é feito para conteúdo em (MAGALHAES, B., 2011), introduzindo-se o Objeto de Serviço.

Neste caso, considera-se que, todos os serviços possuem: Um identificador único que os identifica, também chamado de ID; um Nome legível em linguagem natural, para facilitar seu uso; um Objeto Descritor, que contém todas as especificações do serviço; um mapeamento para o Localizador de Substrato, que permite localizar os serviços na rede;

e um Contrato entre o provedor de serviços e o usuário ou consumidor, também chamado de *Service Level Agreement*. A Figura 17 mostra a estrutura de uma Internet de Serviços baseado no GIRS.



**Figura 17:** Estrutura do mapeamento para Internet de Serviços com base no GIRS. A função hash é usada para os objetos de informação.

Na Figura 17 tem-se uma ilustração da estrutura proposta neste trabalho. À direita estão os elementos que compõem a arquitetura: Serviço, Descritores, Nome, SLA e o Substrato. Estes elementos da arquitetura são identificados unicamente (ID-Serviços; ID-Descritores; ID-Nome; ID-SLA e ID-Substrato), como mostra a coluna a seguir, para que este sistema de indireção possa ser usado para mapeamento dos atributos tanto de serviços, quanto de substratos.

Alguns mecanismos terão que ser usados para criar os identificadores únicos dos serviços. Por exemplo, o próprio sistema operacional poderia criar este ID. Assim, com o serviço identificado unicamente pode-se criar um objeto que o descreve para uso geral, contendo todas as especificações do mesmo, e posteriormente criar-se um nome para este mesmo serviço.

De forma semelhante ao GIRS, os elementos do substrato, que podem ser *hosts*, servidores, ou qualquer outro equipamento que esteja executando o serviço, são identificados unicamente a partir da combinação única de características da máquina, e atribui-se um nome legível para o mesmo. Este esquema permite separar os identificadores e localizadores do substrato, diferentemente do que acontece nas redes atuais. Com esta separação, os localizadores do substrato são encontrados através de seus identificadores únicos, de forma dinâmica.

### 5.1 Tabela de Mapeamento para a Internet de Serviços

Nesta seção, utiliza-se a flexibilidade da tabela de mapeamento GIRS para mostrar as relações ilustradas na Figura 17. Os mapeamentos são implementados através de registros em tabelas *hash* distribuídas. A Tabela 2 contém exemplos de alguns mapeamentos para Internet de Serviços com as suas chaves de entrada, uma previsão da quantidade e valores de saída, e o tempo de vida destes mapeamentos (*Time To Live-TTL*).

Mapeamento	Chave	Valor (XML)			
		Tipo	Quant	Lista de Valores	TTL
<ID-Serviço; ID-Descrição>	ID-Serviço	1	1	ID-Descrição1; ID do descritor associado a um serviço.	1
<ID-Descrição; ID-Serviço>	ID-Descrição	2	5	ID-Serviço1 ,..., ID-Serviço5 ; Lista de serviços associados a um dado descritor relativo ao nome “transcodificador”.	1
<ID-Serviço; ID-Nome>	ID-Serviço	3	2	ID-Nome1; ID-Nome2; IDs de nomes associados a um único serviço.	1
<ID-Nome; ID-Serviço>	ID-Nome	4	7	ID-Serviço1 ,..., ID-Serviço7; Lista de serviços que possuem exatamente o mesmo nome.	1
<ID-Nome; ID-Descrição>	ID-Nome	5	7	ID-Descrição1,...,ID-Descrição7; Lista dos descritores de objetos de serviços associados ao ID de Nome “transcodificador”.	1
<ID-Descrição; ID-Nome>	ID-Descrição	6	2	ID-Nome1; ID-Nome2; IDs de Nomes associados a um descritor como “Transcodificação”, “Filragem”.	1

<ID-Serviço; ID-Substrato>	ID-Serviço	7	4	ID-Substrato1 ,..., ID-Substrato4; Lista dos identificadores de Substrato onde se encontra executando o serviço requisitado.	1
<ID-Substrato; ID-Serviço>	ID-Substrato	8	15	ID-Serviço1, ..., ID-Serviço15; Lista de serviços encontrados em um dado Substrato.	1
<ID-Descriptor; ID-Substrato>	ID-Descriptor	9	5	ID-Substrato1, ..., ID-Substrato5; Lista de identificadores de Substrato que possuem o descriptor de um determinado serviço.	1
<ID-Substrato; ID-Descriptor>	ID-Substrato	10	6	ID-Descriptor1, ..., ID-Descriptor6; Lista de descritores disponíveis no Substrato identificado pelo ID-Substrato.	1
<ID-Substrato; Loc-Substrato>	ID-Substrato	11	1	Loc1-Substrato; Endereço na rede do Substrato.	1
<ID-SLA; ID-Serviço>	ID-SLA	12	3	ID-Serviço1, ..., ID-Serviço3 Lista de identificadores de Serviços que fazem parte do SLA com este ID.	1

**Tabela 2:** Exemplos de mapeamentos para a Internet de Serviço que podem ser acomodados no GIRS.

A Tabela 2 relaciona 12 tipos de mapeamentos que representam as ligações dinâmicas entre os identificadores de Serviços, Descritores, Nomes, SLA e identificadores e localizadores de Elementos de Substrato, como ilustrado na Figura 17.

Para exemplificar, o mapeamento 1 é utilizado para se obter o identificador do descriptor (ID-Descriptor) relacionado a um dado serviço (ID-Serviço). No mapeamento 2 deseja-se obter um a lista de serviços associados a um dado descriptor, assim entra-se com a chave ID-Descriptor e se obtém esta lista (ID-Serviço 1, ..., ID-Serviço 5). O mapeamento 11 retorna o localizador do elemento de substrato associado a este identificador de substrato (ID-Substrato). O mapeamento 12 retorna 3 identificadores de substrato que armazenam o SLA relacionado à chave consultada (ID-SLA).

Esta abordagem é totalmente flexível e possibilita a inserção de mapeamentos entre várias entidades, tais como, conteúdo no caso do GIRS, serviços neste trabalho, usuários, fluxos de informações, dentre outras.

## 5.2 Mecanismo de Resolução

Diferentemente do GIRS, que utiliza três módulos para o mecanismo de resolução de indireções, neste trabalho tem-se apenas um, que denomina-se de forma genérica como, *Search and Indirection Resolution* (SIR). Este trabalho repensa a proposta GIRS considerando que os serviços desta arquitetura também fazem parte da Internet de Serviços. Assim, o serviço SIR é o responsável pela pesquisa e recuperação de entidades, que podem ser serviços e/ ou conteúdos, com base nas especificações contidas nos descritores dos mesmos, e é responsável ainda, pelo mapeamento entre os identificadores e os localizadores do substrato que possuem tais serviços.

Outra função desempenhada pelo SIR é a notificação dos elementos envolvidos na publicação ou assinatura de algum mapeamento, sempre que for requerido. A notificação nada mais é do que uma mensagem de aviso enviado pelo SIR para informar que um dado mapeamento foi publicado ou assinado após a autenticação dos pares.

## 5.3 Mecanismos de Publicação, Assinatura e Notificação

O paradigma publica/assina (*publish/subscribe*) é implementado a partir da publicação de um objeto de serviço. A publicação é feita ao registrar a identificação única do serviço no SIR, e a assinatura por sua vez, é feita quando algum interessado solicita ao SIR algum serviço publicado. A localização e contratação de um serviço só é possível, portanto, depois que seu objeto for publicado. Neste trabalho, usam-se os comandos **Pub**, para fazer a publicação, e **Sub** para fazer a assinatura.

Existem duas formas de se fazer publicações/assinaturas: com notificação ou sem notificação. Com notificação, quando é necessário que o SIR notifique outra entidade a respeito da publicação/assinatura de algum mapeamento, e possui a seguinte estrutura: **Pub/Sub ( ID-Fonte; ID-Notificação; < Mapeamento > )**, onde; ID-Fonte, é a identificação da entidade que publicando/assinando no SIR e receberá de volta, uma mensagem (ACK) confirmando que o mapeamento foi publicado/assinado com sucesso; O

campo ID-Notificação é uma identificação da entidade que será notificada pelo SIR a respeito da publicação/assinatura do referido mapeamento.

Por outro lado, as publicações/assinaturas sem notificação são aquelas em que não existe a necessidade de notificação de outra entidade, possuindo a seguinte estrutura: **Pub/Sub ( ID-Fonte < Mapeamento > )**. Claramente, só há a necessidade de termos o ID-Fonte como parâmetro, que receberá a mensagem (ACK) de confirmação pelo SIR, pois é uma publicação/assinatura sem notificação.

Nesta proposta, as publicações/assinaturas sem notificação são realizadas durante o processo de localização e escolha dos serviços, e os com notificação aparecem durante o processo de contratação dos mesmos, como se mostra no caso de estudo apresentado na Seção 5.6.

#### **5.4 Estrutura do Descritor**

O descritor é o objeto que contém todas as especificações do serviço que ele descreve. Genericamente, ele revela informações do serviço prestado, tais como: nome do serviço, como por exemplo, “transcodificação de vídeo digital”; o tipo de serviço, por exemplo, o tipo de transcodificação, heterogênea ou homogênea; se homogênea, que tipo de transcodificação homogênea este serviço faz (ajuste de taxa de bits, espacial ou temporal), (REGIS, ET AL., 2007), além de varias outras informações. Não existe um padrão para as especificações de serviço, exigindo apenas que elas sejam em linguagem natural, para permitir que os usuários ou aplicações clientes escolham o serviço com base no contexto e na semântica destas especificações.

Neste trabalho, usaremos a estrutura genérica de descritores usada por (MAGALHAES, B., 2011), no GIRS.

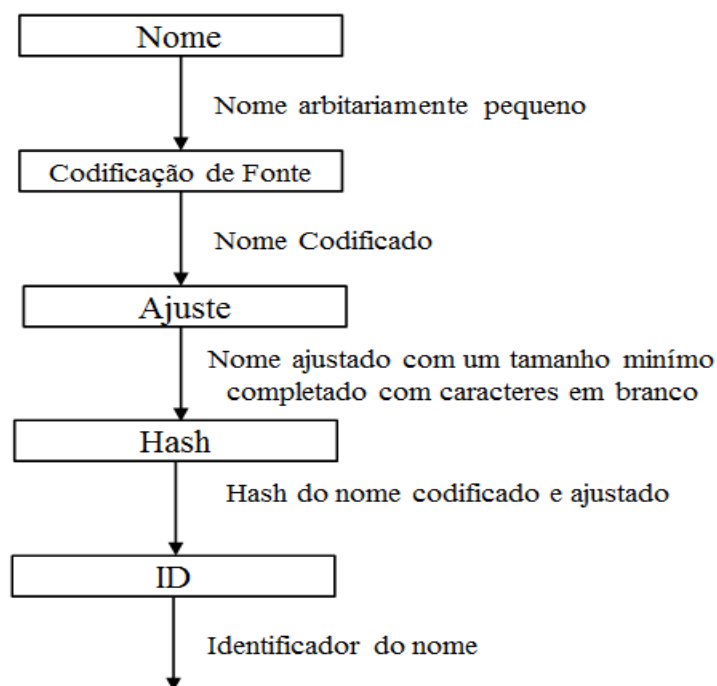
#### **5.5 Ajuste de Nomes**

Assim como no mundo real, a concorrência é o motor do sistema econômico mundial, exigindo dos diferentes competidores, serviços e produtos cada vez melhores. Na



Internet de Serviços ela também deverá existir. Assim, a existência de diferentes serviços concorrentes pode ocasionar a existência de numerosos serviços homônimos. Portanto, exige-se que os nomes dos serviços tenham um alto grau semântico.

Propõem-se neste trabalho que os nomes de serviços passem por uma codificação de fonte e um ajuste de nome. A codificação é usada para a padronização na nomeação dos serviços e o ajuste padroniza o nome em um tamanho mínimo estabelecido – evitando assim o uso de nomes pequenos demais. A Figura 18 ilustra o processo de criação de nomes, desenvolvido em (MAGALHAES, B., 2011).



**Figura 18:** Ajuste e codificação de nomes (MAGALHAES, B., 2011).

## 5.6 Localização e Contratação de um Serviço

Nesta parte do trabalho, apresenta-se um estudo de um caso genérico, para a localização de um objeto de serviço e contratação de um serviço no escopo da proposta do

SIR. É importante salientar que a localização de um determinado serviço por parte do usuário ou consumidor não implica na sua contratação automática, sendo necessário o estabelecimento de um acordo (SLA) entre as partes, isto é, entre o usuário ou aplicação cliente e o provedor deste serviço. Assim, para melhor entendimento desta proposta dividiremos este estudo de caso em duas partes, na localização do serviço e na contratação do mesmo.

É de extrema importância ressaltar que antes de todo o processo de localização e contratação de serviço existe toda uma gama de questões relativas à segurança e confiança entre as entidades que publicam e as que assinam estes serviços, mas para a diminuição da complexidade deste nosso exemplo, abstrai-se que estas questões estão em outros elementos da arquitetura, portanto, fora do escopo deste trabalho. Propõe-se, também, que na Internet de Serviço, que os usuários ou clientes de um determinado serviço, publiquem avaliações sobre o mesmo, com base na sua experiência, para o estabelecimento de relações de confiança para outros serviços clientes.

### **5.6.1 Localização**

A construção dinâmica e flexível de aplicações, a partir da combinação de diferentes serviços distribuídos é uma das metas da Internet de Serviços. Desta forma, a localização do conjunto de serviços que vão compor a aplicação é a primeira tarefa a ser realizada.

A Figura 19 ilustra o processo de localização de um determinado objeto de serviço, por parte de uma aplicação na IoS.

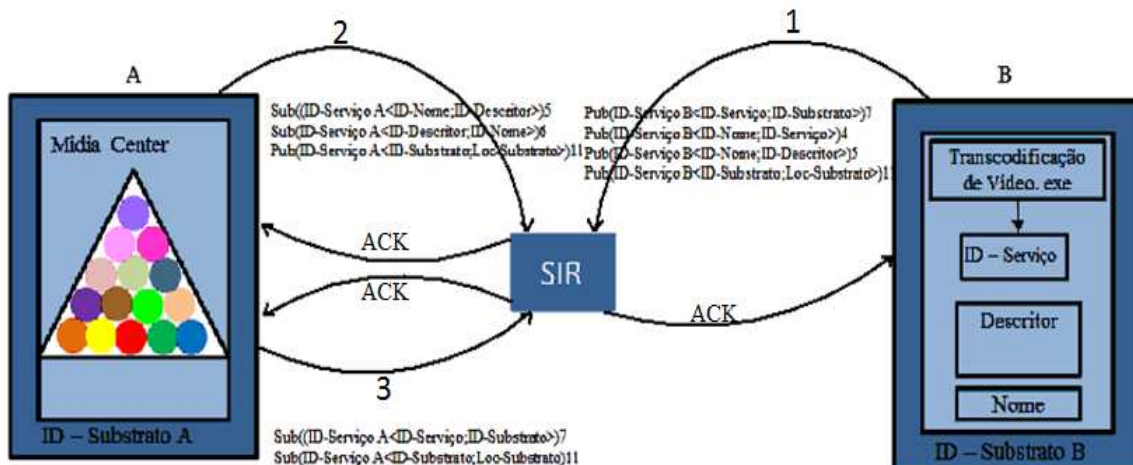


Figura 19: Exemplo de um processo de localização de serviço.

Na Figura 19 tem-se o *host* A, que é uma aplicação de Media Center, interessado em um serviço de transcodificação de vídeo oferecido pelo *host* B. Considera-se que o *host* responsável pelo serviço desejado (*host* B) deve publicar no SIR o serviço que oferece (mapeamento 7 da Tabela 2), o nome do serviço (mapeamento 4 da Tabela 2), seu descritor (mapeamento 5 da Tabela 2) e sua localização (mapeamento 11 da Tabela 2) representados pelo passo 1 na Figura 19.

Com o serviço já publicado, o *host* assinante, isto é, o *host* A, faz uma busca pelo mesmo, baseando-se no nome do serviço e/ou nas especificações contidas no descritor (mapeamentos 5 e 6 da Tabela 2) representado pelo passo 2. Para tanto, o *host* A deve formular nomes adequados semanticamente e assiná-los (etapa não mostrada na Figura 19). Ele deve adivinhar o nome publicado pelo candidato ou já ter conhecimento deste nome por outros meios. Embora a variedade de nomes possíveis seja muito grande, os nomes podem ser escolhidos de forma a melhor representar os serviços existentes. Algoritmos para ajudar na atribuição desses nomes podem ser utilizados. Se houver algum serviço que encaixe exatamente no nome formulado, o ID do serviço poderá ser obtido a partir do mapeamento 4 da Tabela 2. E a partir do ID do serviço desejado, o mapeamento 4 da Tabela 2 permite obter o descritor exato do serviço candidato a auto-organização semântica. Com base nestas informações, o nó assinante escolhe um dos candidatos (mapeamento 7 da Tabela 2) e localiza onde estão hospedados (mapeamento 11 da Tabela 2), ações estas representadas pelo passo 3 na Figura 19.

Aqui, na eleição dos candidatos, temos a questão da confiança presente. Assumimos aqui que esta escolha pode ser feita não apenas pelas informações disponíveis nos descritores dos candidatos mas também pelo acesso a relatórios de confiança gerados por outros utilizadores anteriores do serviço em questão.

A localização dinâmica do elemento de substrato onde está hospedado o serviço, feito no mapeamento 11 da Tabela 2, permite que este esquema de tratamento de indireções para a Internet de Serviço suporte a mobilidade e o *multihoming*, uma vez que existe o desacoplamento lógico entre o identificador e o localizador do substrato.

### **5.6.2 Contratação**

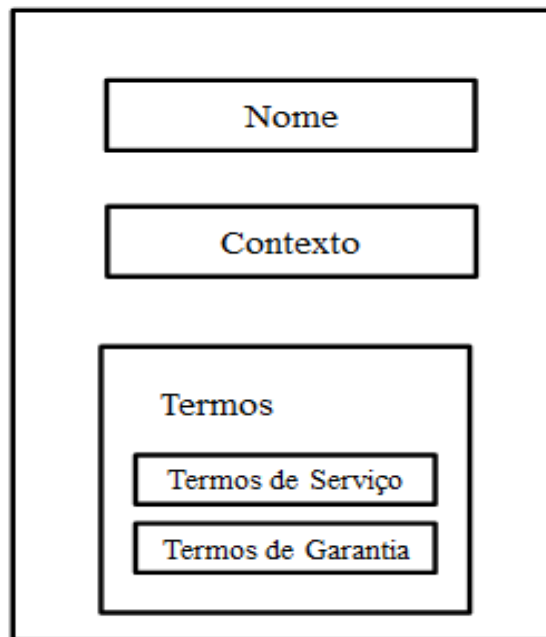
Com a localização do elemento do substrato que possui o serviço feito pelo mapeamento 11 da Tabela 2, o *host* assinante procede então para a assinatura do SLA com o serviço publicador. Novamente, o processo se dará via SIR. Antes de apresentar os detalhes deste processo, é importante apresentar o SLA. O SLA é um contrato de todos os recursos/condições que o cliente espera receber por um determinado serviço. Importa ressaltar que não se refere apenas às funcionalidades entregues pelo serviço, isto é, com a qualidade de serviço, mas também com a qualidade de experiência do usuário (QoE, sigla em Inglês de *Quality of Experience*).

Um SLA especifica um ou mais objetivos em nível de serviço, tanto para as necessidades do cliente como para as garantias por parte do prestador de serviço para a qualidade de atendimento e/ou sobre a disponibilidade dos recursos. Por exemplo, um SLA pode dar garantias quanto ao tempo limite de resposta do serviço ou a disponibilidade do serviço, e alternativamente, fornecer garantias quanto à disponibilidade de recursos mínimos, como memória, CPU, armazenamento etc.

### **Estrutura de um SLA**

Um SLA deve conter parâmetros objetivos e mensuráveis que o provedor de serviço se comprometa a atender. A Figura 20 é uma versão de (ANDRIEUX, et al.,2007)

para estabelecimento de SLA para *Web Services*, e ilustra a estrutura de um SLA. Opcionalmente podemos ou não, nomear o SLA, em linguagem natural. O contexto é o campo que contém os metadados para o acordo inteiro, tais como: nomes, identificação e localização dos participantes; tempo de vida do acordo e outros dados.



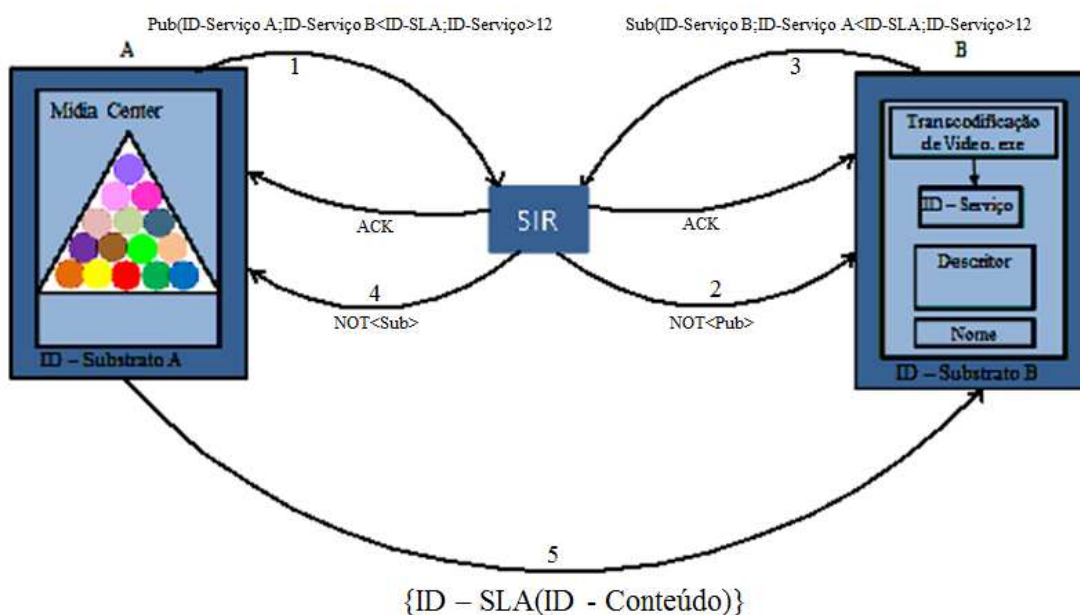
**Figura 20:** *Estrutura de SLA.*

No bloco referente aos termos tem-se: Termos de Serviço, que descrevem as funcionalidades de serviço que serão entregues no âmbito do acordo, por exemplo, se o provedor de serviço possui mais de uma funcionalidade (transcodifica vídeo e filtra áudio, por exemplo). Este termo diz qual delas será entregue ao usuário, no nosso caso, transcodificação; Termos de Garantia é o campo que define os termos que vão garantir a qualidade de serviço, associados com a descrição do mesmo, tal qual definidos pelos termos de serviço. Referem-se basicamente à disponibilidade de serviço, os requisitos de desempenho e os tempos e prazos de duração do acordo, tais como tempo de recuperação de falha ou tempo máximo de indisponibilidade do serviço.

Para estabelecimento do SLA, usa-se neste trabalho a forma de negociação de rodada única, sem contra-ofertas. Isto é, uma parte faz uma oferta de acordo, e a outra parte

simplesmente aceita ou rejeita. Note, que em nenhum momento acima, define-se qual das partes faz a oferta do SLA (cliente ou provedor). Isto possibilita a flexibilidade da proposta, pois em certos casos, é conveniente que o cliente define os termos do acordo, e em outros o provedor do serviço.

Neste trabalho, assumi-se que é a aplicação cliente responsável pela montagem do SLA, como mostra a Figura 21. Assim, depois do passo 3 da Figura 19, que determina a localização do *host* que executa o serviço, o *host* A publica no SIR, o SLA (mapeamento 12 da Tabela 2) representado pelo passo 1 na Figura 21. Após esta publicação, o SIR notifica o *host* B, (passo 2), da publicação do SLA. Desta forma, o *host* B assina o SLA, mapeamento 12 da Tabela 2, representado pelo passo 3. Feita a assinatura, o *host* A é notificado pelo SIR, passo 4, e pode finalmente encaminhar a mensagem  $\langle \text{ID-SLA (ID-Conteúdo)} \rangle$  (passo 5), que especifica que, com base no contrato assinado (ID-SLA), o serviço B deve fazer o tratamento adequado de um arquivo de conteúdo, no caso atual, transcodificar, o conteúdo identificado pelo respectivo ID-Conteúdo.



**Figura 21:** Exemplo do processo de contratação de um serviço.

No caso do *host* B não assinar o SLA proposto, o *host* A usará os próximos candidatos obtidos no passo 2 da Figura 19. Apesar disto ser possível, estas situações são minimizadas, pois o processo de escolha dos candidatos, envolve, como já se disse, a análise cuidadosa dos descritores dos objetos de serviços. Em condições normais, são escolhidos aqueles candidatos que seus descritores apresentam maior capacidade para a execução do serviço pretendido.

Como se mostrou, o SLA é fundamental para a existência da Internet de Serviços, mas é importante sublinhar, que ele deve ser feito de forma flexível para atender à diversidade de clientes e provedores de serviço. Assim, a ausência ou mesmo um SLA fraco ou frouxo, torna difícil o processo de cobrança de qualidade dos provedores de serviço, o que implicará na qualidade da aplicação. Mas, por outro lado, um SLA extremamente rigoroso, eleva o custo do serviço ou pode provocar a realização de promessas não cumpridas, o que inviabilizaria a construção de uma aplicação cliente. Assim, é necessário pesar na profundidade da negociação. E mais, no cenário de Internet de Serviços apresentado, fica evidente a necessidade de incluir as etapas de negociação e contratação, pois somente com elas os parceiros poder[ao] alinhar seus interesses, objetivos, e verificar se a qualidade dos serviços prestados atende os requisitos contratados.

### **5.7 Discussões sobre Questões de Segurança, Confiança e de Dependência.**

Nas redes atuais, a segurança é baseada nos *hosts*, e por consequência a segurança e confiança no serviço é baseado na sua localização. Desta forma, ao invocar um serviço, ele é especificado pelo endereço de um servidor específico. Com o acoplamento entre o identificador e o localizador dos *hosts* nas redes IPs, estes serviços ficam indisponíveis quando ocorre um evento de mobilidade. Na Internet de Serviços isto não ocorre, pois os serviços são invocados através dos seus identificadores unicamente atribuídos na rede.

Com base na experiência de cada usuário do serviço, podem ser gerados relatórios de confiança para o provedor de serviço, como forma de demonstração da confiança do mesmo. Ou seja, o SLA pode ser avaliado após o serviço, assim como acontece hoje, quando se faz uma compra na *web*.

Como visto, no estudo de caso apresentado neste trabalho, um dos objetivos principais da Internet de Serviços é a construção, dinâmica e flexível, de aplicações a partir da combinação dinâmica e contextualizada de diferentes serviços distribuídos. Esta combinação, dos diferentes serviços, é regulada através do SLA. O gerenciamento destes serviços, que colaboram entre si, é uma tarefa difícil, pois eles dependem uns dos outros. Esta dependência leva a propagação de falhas, isto é, se um dos serviços falhar, o desempenho global da aplicação fica comprometido. Desta forma, os diferentes SLA devem ser negociados de forma que garantam uma boa colaboração entre os serviços. Assim, alterações em um SLA já negociado podem exigir alterações em outros SLAs da aplicação.



## Capítulo 6

### Conclusões

Neste trabalho foram apresentados os requisitos básicos do paradigma de Computação Orientado a Serviços e as principais características das Arquiteturas Orientadas a Serviço. Algumas propostas de arquiteturas para a Internet de Serviços foram igualmente estudadas.

Embora as arquiteturas de Internet de Serviços propostas já tenham alcançado resultados importantes, muitas questões fundamentais para a Internet do futuro não são abordadas nelas, tais como, o desacoplamento dos identificadores e localizadores das entidades de serviço, a identificação única destas entidades e, principalmente, o suporte ao estabelecimento do SLA. A partir desta constatação, foi proposta neste trabalho uma Arquitetura de Internet de Serviços baseado no GIRS. O Sistema de Internet de Serviços baseado no GIRS, permite:

- ✓ O desacoplamento de identificadores e localizadores das entidades de serviço, para suporte a mobilidade destas entidades.
- ✓ A identificação única das entidades de serviço e criação de nomes e descritores, diminuindo drasticamente a possibilidade de existência de serviços homônimos e permitindo a otimização da busca por esses serviços.

- ✓ A total independência da tecnologia de interconexão, podendo ser aplicada as redes IP atuais, bem como as futuras abordagens de Internet do Futuro.
- ✓ A inserção do SLA como parte integrante da arquitetura.

De forma geral, propõe-se com esta arquitetura uma solução geral e unificada para o paradigma de Internet de Serviços, permitindo que estes sejam localizados, invocados e contratados independentemente de sua localização física na rede, e que esta invocação seja feita por tecnologias abertas e padronizadas e principalmente o fraco acoplamento entre os serviços e seus consumidores ou clientes.

Naturalmente, algumas questões, fora do escopo deste trabalho, foram identificadas para a evolução desta arquitetura, e são deixadas como sugestões para trabalhos futuros, tais como:

- ✓ Desenvolvimento de um código fonte que implemente a arquitetura conceitual projetada, e de um ambiente de teste para o Sistema de Internet baseado no GIRS.
- ✓ Criação e implementação de um mecanismo eficiente para tratar das questões de identificação única das entidades de serviço.
- ✓ Criação de mecanismos internos a arquitetura ou entidades independentes para tratar das questões de segurança e confiança entre as entidades envolvidas na negociação.
- ✓ Estudo e análise ampla sobre o ciclo de vida das entidades de serviço, para tratar de questões como a invocação (nascimento) e a liberação (morte) destas entidades.

- ✓ Desenvolvimento de um código para estabelecimento de SLA na IoS, assim como existe na SLA *Language* (ANDRIEUX, et al.,2007) voltada aos *web services*.

# REFERÊNCIAS BIBLIOGRÁFICAS

**Internet Usage Statistic.** 2011. Disponível em:

<http://www.internetworldstats.com/stats.htm>. Último acesso em: 21 de Setembro de 2011.

REDING, Viviane. **The Future of Internet: A compendium of European Projects on ICT Research Suported by the 7<sup>th</sup> Framework Programme for RTD.** 2008.

BOOTH, David. et al. **Web Services Architecture: W3C Working Group Note.** Fevereiro de 2004. Disponível em <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211>.

CARDOSO Jorge, WINKLER Matthias, VOIGT Konrad, BERTHOLD Henrike. **IoS-Based Services, Platform Services, SLA and Models for the Internet of Services.** Book Chapter: Software and Data Technologies. Springer Berlin Heidelberg, Junho de 2011, pag 3-17.

CHANNABASAVIAIAH Kishore, HOLLEY Kerrie, TUGGLE Edward M. **Migrating to a Service-Oriented Architecture.** IBM White Paper, Abril de 2004. Disponível em [ftp://129.35.224.15/software/info/openenvironment/G224-7298-00\\_Final.pdf](ftp://129.35.224.15/software/info/openenvironment/G224-7298-00_Final.pdf).

PAPAZOGLU Michael P., TRAVERSO Paolo, DUSTDAR Schahram, LEYMANN Frank, KRÄMER Bernd J. **Service-Oriented Computing: Research Roadmap.** Dagstuhl Senimar Proceedings, Maio de 2006.

**NESSI 2011.** Disponível em:

[http://www.nessi-europe.com/files/ResearchPapers/NESSI\\_2005\\_FoundingVision.pdf](http://www.nessi-europe.com/files/ResearchPapers/NESSI_2005_FoundingVision.pdf).

Último acesso em: 28 de Setembro de 2011.

DOMINGUE, John. **Future Internet Service Offer: An Overview**. Abril 2009. Disponível em [http://services.future-internet.eu/images/e/e0/Future\\_Internet\\_Service\\_Offer\\_v4.pdf](http://services.future-internet.eu/images/e/e0/Future_Internet_Service_Offer_v4.pdf)

MAGALHÃES, Bruno. **Desacoplamento Identificadores e Localizadores: Uma Comparação de Abordagens e Proposta de Arquitetura**. Dissertação de conclusão do curso de Mestrado em Telecomunicações do Inatel, Fevereiro 2011.

de HOLANDA, Aurélio B. **Dicionário Aurélio da Língua Portuguesa 3ª edição**. Editora Nova Fronteira, Rio de Janeiro 1999.

HULL Richard, BENEDIKT Michael, CRISTOPHIDES Vassilis, SU Jianwen. **E-Services: A Look Behind the Curtain**. 2003. Disponível em <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.1501>.

LACERDA Luciana. **Serviços é o Setor da Economia que mais Cresce**. Disponível em: <http://g1.globo.com/economia-e-negocios/noticia/.html>. Último acesso em: 21 de Setembro de 2011.

ELFATRY Ahmed, LAYZELL Paul. **Negotiating in Service-Oriented Environments**. Communications of the ACM, Vol. 47, N° 8, Agosto de 2004. Disponível em: [http://mmlab.ceid.upatras.gr/courses/ais\\_site/files/3%5CNegotiating%20in%20service-oriented%20environments.pdf](http://mmlab.ceid.upatras.gr/courses/ais_site/files/3%5CNegotiating%20in%20service-oriented%20environments.pdf)

CARTER, Sandy. **The New Language of Business: SOA & Web 2.0**. IBM Press, Fevereiro de 2007.

ANDREI, Mark. et al. **Patterns: Service-Oriented Architecture and Web Services**. IBM RedBooks, Abril de 2004, pag 17-30. Disponível em <http://www.immagic.com/eLibrary/ARCHIVES/GENERAL/IBM/SG246303.pdf>.

ERL, Thomas. **SOA Principles of Service Design**. The Prentice Hall Service-Oriented Computing Series, Julho de 2007, pag 37-54.

FREMANTLE Paul, WEERAWARANA Sanjiva, KHALAF Rania. **Enterprise Services**. Communication of the ACM, Vol. 45, Nº 10. Outubro de 2002, pag 77-82.

MITRA Nilo, LAFON Yves. **SOAP Version 1.2: W3C Recommendation**. Abril de 2007. Disponível em <http://www.w3.org/TR/soap12-part0>.

JOSUTTIS, Nicolai M. **SOA in Practice: The Art of Distributed System Design**. O Reilly Books, 1ª ed, Agosto de 2007, pag 47-56.

PAPAZOGLU Michael P., VAN DEN HEUVEL Willen-Jan. **Service Oriented Architectures: Approaches, Technologies and Research Issues**. Springer-Verlag, Março de 2007. Disponível em:

<ftp://ftp-sop.inria.fr/members/Didier.Parigot/PDFarticle/francesco/Papazoglou2007a.pdf>

MODI Gurpreet S. **Service Oriented Architecture & Web 2.0**. 2004. Disponível em [http://www.gsmodi.com/files/SOA\\_Web2\\_Report.pdf](http://www.gsmodi.com/files/SOA_Web2_Report.pdf).

**SOA4ALL 2011**. Disponível em: <http://www.soa4all.eu>. Último acesso em 22 de Setembro de 2011.

PEDRINACCI Carlos, KRUMMENACHER Reto, NORTON Barry, SIMPERL Elena. **SOA4All: Enabling Web-scale Service Economies**. 2009. Disponível em: <http://soa4all.eu/pubs/icsc2009.pdf>.

**SLA@SOI 2011**. Disponível em: <http://sla-at-soi.eu>. Último acesso em: 20 de Setembro de 2011.

WOLFGANG Theilmann, GUINEA Sam, BROSCHE Franz, YAHYPOUR Ramin. **Deliverable D.A1a Framework Architecture-Evaluated Architecture**.2011. Disponível em <http://sla-at-soi.eu/wp-content/uploads/2009/07/D.A1a-M26-FrameworkArchitecture.pdf>

JACOBSON Van, et al. **Networking Named Content**. Proceedings of the 5<sup>th</sup> international conference on Emerging networking experiments and technologies (CoNext). ACM Press, New York, 2009.

POSTEL, Jon. **RFC 791 – Internet Protocol**. 1981. Disponível em <http://www.faqs.org/rfcs/rfc791.html>.

FARANACCI, D.; FULLER, V.; MEYER, D; LEWIS, D. **Locator/ID Separation Protocol (LISP) Draft-IETF-LISP-15**. Julho de 2011. Disponível em: <http://tools.ietf.org/html/draft-ietf-lisp-15>.

MOSKOWITZ, R.; NIKANDER, P. **Host Identity Protocol (HIP) Architecture RFC 4423**. Maio de 2006. Disponível em: <http://www.ietf.org/rfc/rfc4423.txt>

JINLI Pan, SUBHARTHI Paul, RAJ Jain, MIC Bowman. **MILSA: A Mobility and Multihoming Supporting Identifier Locator Split Architecture for Naming in the Next Generation Internet**. Proceedings of IEEE Global Conference (GLOBECOM) 2008. Disponível em: <http://www.cse.wustl.edu/~jain/papers/ftp/milsa.pdf>.

HARAI Hiroaki, et al. **Akari Architecture Conceptual Design for Generation Network**. Agosto de 2009. Disponível em [http://akari-project.nict.go.jp/eng/concept-design/AKARI\\_fulltext\\_e\\_preliminary\\_ver2.pdf](http://akari-project.nict.go.jp/eng/concept-design/AKARI_fulltext_e_preliminary_ver2.pdf).

AHLGREN Bengt, D'AMBROSIO Matteo, DANNEWITZ Christian. **Second NetInf Architecture Description Deliverable 6.2**. Junho de 2010. Disponível em: <http://www.4ward-project.eu/index.php?id=192>.

OHLMAN Börje, et al. **First NetInf Architecture Description Deliverable 6.1**. Fevereiro de 2009. Disponível em: <http://www.4ward-project.eu/index.php?id=62>

AIN Mark, et al. **PSIRP-Publish Subscribe Internet Routing Paradigm Deliverable 2.2: Conceptual Architecture of PSIRP Including Subcomponent Descriptions**. Agosto de 2008. Disponível em: [http://psirp.org/files/Deliverables/FP7-INFISO-ICT-216173-PSIRP-D2.2\\_ConceptualArchitecture\\_v1.1.pdf](http://psirp.org/files/Deliverables/FP7-INFISO-ICT-216173-PSIRP-D2.2_ConceptualArchitecture_v1.1.pdf)

STOICA, Ion, et al. **Internet Indirection Infrastructure**. 2002. Disponível em: <http://www.cs.berkeley.edu/~istoica/papers/i3-sigcomm02.pdf>

WIKIPEDIA. **Distributed Hash Table**. Disponível em: [http://en.wikipedia.org/wiki/Distributed\\_hash\\_table](http://en.wikipedia.org/wiki/Distributed_hash_table). Último Acesso em: 20 de Julho de 2011.

LIU Shi, BI Jun, WANG Yangyang. **DHTs-Based Mapping System for Identifier and Locator**. Proceedings of the First International Conference on Advances in Future Internet (AFIN) 2009.

HANKA Oliver, SPLEIB Christoph, KUNZMANN Gerald, EBERSPACHER Jorg. **A DHT-Inspired Clean-Slate Approach for the Next Generation Internet**. In the 2<sup>nd</sup> GI/ITG KuVs Workshop on The Future Internet. 2008.

REGIS Carlos D., et al. **Transcodificação de Vídeo Digital para Receptores Portáteis**. 4<sup>o</sup> Simpósio de Excelência em Gestão e Tecnologia (SEGET) 2009. Disponível em: [http://www.aedb.br/seget/artigos07/1288\\_1288\\_transcodificacao\\_seget\\_final.pdf](http://www.aedb.br/seget/artigos07/1288_1288_transcodificacao_seget_final.pdf).

ANDRIEUX Alain, et al. **Web Services Agreement Specification (WS-Agreement)**. 2007. Disponível em: <http://www.ogf.org/documents/GFD.107.pdf>.