



UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

**Segurança de aplicações empresariais
em arquitecturas de serviços**

Miguel Filipe Leitão Pardal

(Licenciado)

Dissertação para obtenção do Grau de Mestre
em Engenharia Informática e de Computadores

Orientador: Doutor Alberto Manuel Ramos da Cunha

Júri

Presidente: Doutor José Manuel da Costa Alves Marques

Vogais: Doutor André Ventura da Cruz Marnôto Zúquete

Doutor Alberto Manuel Ramos da Cunha

Setembro de 2006

Resumo

As organizações procuram agilizar os seus sistemas de informação para melhor responder ao permanente desafio de adaptação aos requisitos de negócio. As arquitecturas de serviços e os Web Services são uma proposta para estruturar os sistemas com maior flexibilidade, reutilização e interoperabilidade. No entanto, as importantes normas e implementações de segurança não foram ainda suficientemente avaliadas na prática.

Esta tese avalia a Web Services Security com um caso de estudo real, complexo e com valor: a compra e venda de imóvel. Foi realizado um protótipo que evidenciou insuficiências das implementações disponíveis.

O contributo mais significativo desta tese é o retrato actual e completo da tecnologia de Web Services, com uma avaliação aprofundada das normas e implementações de segurança.

Palavras Chave:

Arquitecturas de Serviços

Web Services

Segurança

Sistemas de Informação Empresariais

Integração de Aplicações Empresariais

Sistemas Distribuídos

Abstract

Organizations want to make their information systems more agile so they can better answer the challenge of continuous adaptation to business requirements. Service architectures and Web Services are a proposal to structure systems with greater flexibility, reuse and interoperability. However, the important security standards and implementations have yet to be sufficiently evaluated in practical uses.

This thesis evaluates Web Services Security with a complex and valuable business case study: real-estate transactions. A prototype evidenced several insufficiencies in the available implementations.

The most relevant contribution of this thesis is the up-to-date and complete description of Web Services technology, with an insightful assessment of security standards and implementations.

Keywords:

Service-Oriented Architectures

Web Services

Security

Enterprise Information Systems

Enterprise Applications Integration

Distributed Systems

Agradecimentos

Começo por agradecer ao meu orientador, Professor Alberto Cunha, por toda a disponibilidade ao longo destes anos e pelo constante enriquecimento intelectual proporcionado pelas nossas reuniões, saudavelmente temperadas com pragmatismo e sentido de humor.

Um obrigado especial a todo o júri, pela prontidão com que leu o meu trabalho e pelos comentários que o tornaram melhor.

Agradeço ao Departamento de Engenharia Informática do Instituto Superior Técnico, pelas condições de trabalho e pelo privilégio de poder aprender e ensinar. Agradeço também à Link Consulting o apoio complementar que me deu.

Agradeço às equipas de Sistemas Distribuídos. Professores Alves Marques, Paulo Guedes e Rodrigo Rodrigues, pela sua liderança, motivação e exigência. Aos colegas das práticas, pelo seu esforço e dedicação. A todos os alunos, por nos incentivarem a todos a fazer sempre mais e melhor. Um abraço ao Nuno Santos, Sérgio Fernandes e Jorge Martins pela “fusão” a frio.

Para chegar até aqui, tive muitos anjos da guarda. José João, Miguel Panão e Ricardo Ataíde, obrigado por me mostrarem o ‘bright side of life’. Miguel Sousa e João Neves, caros LEICanos, sem vocês não seria o Engenheiro que sou hoje. Amigos da “Eunice” (Unisys), o vosso profissionalismo e empenho no ‘delivery’ com um sorriso nos lábios é inesquecível. Marta Guerra, obrigado pela parceria indispensável em alguns dos melhores trabalhos que já fiz. Helena Simões, obrigado pela simpatia e competência. N’anhos, obrigado.

Finalmente, quero agradecer à minha família que sempre acreditou em mim e me apoiou. Pai e mãe, devo-vos tudo. Ana e João, vocês são os meus irmãos muito queridos, obrigado pela paciência. Avós e tios, mesmo longe estão sempre perto do meu coração. Sogros, obrigado por me tratarem como um filho. Joana, és o raio de luz da minha vida que enches todos os dias com reflexos de ouro. Obrigado por tudo, amo-te!

Lisboa, 8 de Setembro de 2006

Miguel Filipe Leitão Pardal

“The city central computer told you?”

*R2-D2, you know better than to **trust** a strange computer!”*

“Star Wars: The Empire Strikes Back”

George Lucas, 1980

Índice

RESUMO.....	I
ABSTRACT.....	III
AGRADECIMENTOS	V
ÍNDICE.....	IX
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABELAS.....	XV
ÍNDICE DE EXEMPLOS	XVII
1. INTRODUÇÃO	1
1.1. Área da tese.....	2
1.2. Objectivo.....	2
1.3. Metodologia	3
1.4. Organização do texto	3
2. ENQUADRAMENTO	5
2.1. Sistemas de informação empresariais	5
2.1.1. <i>Negócio</i>	5
2.1.2. <i>Sistemas de informação</i>	5
2.1.3. <i>Aplicações empresariais</i>	7
2.2. Serviços.....	8
2.2.1. <i>Características</i>	8
2.2.2. <i>Arquitectura orientada a serviços</i>	9
2.2.3. <i>Tecnologia relacionada</i>	10
2.3. Requisitos de sistemas de informação.....	10
2.3.1. <i>Perspectivas</i>	10
2.3.2. <i>Tipos de requisitos</i>	11
2.4. Segurança.....	13
2.4.1. <i>Agente, acção e recurso</i>	13
2.4.2. <i>Ataques</i>	13
2.4.3. <i>Defesas</i>	14
2.4.4. <i>Agente, acção e recurso no contexto de um serviço</i>	16

2.4.5. <i>Confiança</i>	16
2.5. <i>Resumo</i>	18
3. PLATAFORMA DE SERVIÇOS	21
3.1. <i>Web Services</i>	21
3.1.1. <i>Princípios técnicos fundamentais</i>	22
3.1.2. <i>Normas</i>	22
3.2. <i>Plataforma base</i>	23
3.2.1. <i>Representação de dados</i>	24
3.2.2. <i>Interoperabilidade</i>	24
3.2.3. <i>Transporte</i>	25
3.2.4. <i>Mensagem</i>	26
3.2.5. <i>Contrato</i>	28
3.2.6. <i>Descoberta</i>	30
3.3. <i>Implementações da plataforma</i>	30
3.3.1. <i>Antes dos Web Services</i>	30
3.3.2. <i>Primeira geração (Dot Net, Axis, JAX-RPC)</i>	31
3.3.3. <i>Segunda geração (WSE, Axis2, JAX-WS)</i>	31
3.3.4. <i>Terceira geração (WCF, WSIT)</i>	32
3.4. <i>Plataforma estendida</i>	33
3.4.1. <i>Segurança</i>	33
3.4.2. <i>Mensagens fíáveis</i>	33
3.4.3. <i>Transacções</i>	33
3.4.4. <i>Processos de negócio</i>	34
3.4.5. <i>Gestão</i>	35
3.5. <i>Edifício normativo</i>	36
3.6. <i>Resumo</i>	38
4. SEGURANÇA DE SERVIÇOS	39
4.1. <i>Problemas a resolver</i>	39
4.2. <i>Mecanismos</i>	40

4.2.1. Autenticação	40
4.2.2. Protecção de mensagens.....	41
4.2.3. Autorização de acesso a recursos.....	41
4.2.4. Protecção de domínios de segurança	42
4.3. Modelo conceptual	42
4.4. Normas	43
4.4.1. XML-Signature e XML-Encryption.....	44
4.4.2. WS-Security.....	44
4.4.3. WS-Trust	45
4.4.4. WS-SecureConversation	45
4.4.5. WS-Federation.....	45
4.4.6. WS-SecurityPolicy	45
4.4.7. SAML (Security Assertion Markup Language).....	48
4.4.8. XAdES (XML Advanced Electronic Signatures).....	50
4.4.9. XACML (Extensible Access Control Markup Language).....	51
4.4.10. REL (Rights Expression Language).....	51
4.4.11. XKMS (XML Key Management Specification)	51
4.5. Implementações disponíveis	51
4.6. Trabalho relacionado.....	52
4.7. Objectivos de avaliação	53
4.8. Resumo	53
5. CASO DE ESTUDO	55
5.1. Motivação.....	55
5.2. Descrição.....	56
5.2.1. Contexto organizacional.....	56
5.2.2. Processo de negócio	57
5.2.3. Cenários.....	60
5.2.4. Sistema de informação estruturado em serviços	63
5.2.5. Protótipo.....	66
5.3. Implementação	70

5.3.1. Ensaios.....	70
5.3.2. Protótipo.....	73
5.4. Resumo	76
6. AVALIAÇÃO	77
6.1. Desenvolvimento de serviços.....	77
6.1.1. Ferramentas.....	77
6.1.2. Vinculação dinâmica	77
6.1.3. Separação de vinculação de dados e funcional.....	78
6.1.4. Esquemas de dados abertos.....	79
6.1.5. Limitações da conversão de dados	79
6.2. Protecção de serviços	80
6.2.1. Segurança no transporte e segurança na mensagem	80
6.2.2. Autenticação	82
6.2.3. Autorização.....	83
6.2.4. Protecção de mensagens.....	84
6.2.5. Configuração	84
6.3. Resumo	87
7. CONCLUSÃO	89
7.1. Contributos.....	89
7.2. Trabalho futuro	93
7.3. Comentário final	94
8. BIBLIOGRAFIA.....	97
A. TRADUÇÕES, SIGLAS E ABREVIATURAS UTILIZADAS	107
B. ORGANIZAÇÕES DE NORMALIZAÇÃO	111
C. POLÍTICAS WS-POLICY	113
D. PROCESSO DE COMPRA E VENDA DE IMÓVEL	119
E. LEGISLAÇÃO PORTUGUESA	129

Índice de Figuras

Figura 1 – Exemplos de diferentes tipos de sistemas de informação nos vários níveis e funções da organização [Laudon02].	6
Figura 2 - Perspectivas de um sistema de informação.....	11
Figura 3 – Agente, acção e recurso de uma aplicação informática.	13
Figura 4 – Ataques ao agente, acção e recurso de uma aplicação informática.....	13
Figura 5 – Defesas de agente, acção e recurso de uma aplicação informática.	14
Figura 6 – Agente, acção e recurso no contexto de um serviço.	16
Figura 7 – Modelo de confiança entre duas entidades.....	17
Figura 8 – Modelo de confiança com três entidades.	17
Figura 9 – Modelo de confiança com quatro entidades.....	18
Figura 10 – Fases de interacção do cliente com o serviço.....	21
Figura 11 – Classificação das normas de Web Services em categorias [Pardal06].....	23
Figura 12 – Normas base dos Web Services [Pardal06].	24
Figura 13 – Mediação entre organizações de normalização e indústria desempenhado pela WS-I [WSI05].	25
Figura 14 – Forma normal de uma WS-Policy [Schlimmer06].....	29
Figura 15 – Normas de Web Services. Adaptado de [Pardal06].	36
Figura 16 – Normas de Web Services suportadas pelo JAX-RPC.	37
Figura 17 – Fases de interacção do cliente com o serviço, com indicação das tecnologias.	38
Figura 18 – Problemas a resolver na segurança de serviços.....	39
Figura 19 – Autenticação directa, quando o cliente e o serviço têm uma relação de confiança [Hogg05].	40
Figura 20 – Autenticação com intermediário, quando o cliente e o serviço não têm uma relação de confiança directa entre si [Hogg05]......	41
Figura 21 – Protecção da fronteira de um domínio de segurança por um nó intermediário [Hogg05]. ..	42
Figura 22 – Modelo de base para Web Services seguros [IBM02].	43
Figura 23 – Normas de segurança para serviços Web [Hogg05].	44

Figura 24 – Processo de desenvolvimento para serviços seguros [Gutierrez05].....	52
Figura 25 – Cadeia de valor do mercado imobiliário.	56
Figura 26 – Forças competitivas do mercado imobiliário português.	57
Figura 27 – Resumo do processo de negócio da compra e venda de imóvel.	58
Figura 28 – Resumo das entidades informacionais da compra e venda de imóvel.	59
Figura 29 – Cenários exemplificativos da compra e venda de imóvel.	60
Figura 30 – Serviços da “consulta de licença de habitação”.	64
Figura 31 – Serviços da “assinatura de contrato-promessa de compra e venda”.	65
Figura 32 – Serviços da “validação de documentos exigidos para escritura”.	66
Figura 33 – Diagrama de colaboração entre serviços no cenário “assinatura de contrato de compra e venda” durante a vinculação.	67
Figura 34 – Diagrama de colaboração entre serviços no cenário “assinatura de contrato de compra e venda” durante a invocação.	67
Figura 35 – Chaves do cenário “assinatura de contrato de compra e venda”.	69

Índice de Tabelas

Tabela 1 – Níveis de segurança em assinaturas digitais XML.	50
Tabela 2 – Normas suportadas nas implementações disponíveis de serviços seguros.	51
Tabela 3 – Relação entre mecanismos e normas de serviço seguros.	53
Tabela 4 – Relação entre os cenários e a avaliação da segurança.	62
Tabela 5 – Vantagens e desvantagens da segurança no transporte e da segurança na mensagem.	81
Tabela 6 – Resultados da avaliação dos mecanismos de autenticação.	82
Tabela 7 – Resultados da avaliação dos mecanismos de autorização.	83
Tabela 8 – Resultados da avaliação dos mecanismos de protecção de mensagens.	84
Tabela 9 – Comparação dos mecanismos de configuração da segurança.	87

Índice de Exemplos

Exemplo 1 – WS-SecurityPolicy [Kaler05].	46
Exemplo 2 – Asserção SAML de autenticação.	48
Exemplo 3 – Asserção SAML de atributos de utilizador.	49
Exemplo 4 – Asserção SAML de autorização.....	49

1. Introdução

Neste início de século XXI, a *Internet* assumiu definitivamente o seu lugar como infra-estrutura principal da “sociedade do conhecimento”. A rede pública de grande escala proporciona um ambiente de negócio aberto e dinâmico, onde as *tecnologias de informação e comunicação* permitem novas e diferentes formas de trabalhar e criar valor.

As palavras *dados*, *informação* e *conhecimento* são por vezes usadas indistintamente, no entanto, cada uma tem significado próprio. Os *dados* são sequências de símbolos que representam factos ou eventos. A *informação* é um conjunto de dados com um contexto de interpretação humano. As pessoas têm *conhecimento* quando usam informação para realizar tarefas específicas, que implicam aprendizagem e experiência. Os *sistemas de informação*, suportados por computadores e redes, facilitam as recolhas de dados, permitem a apresentação e manipulação de informação e apoiam as pessoas em trabalho de conhecimento [Laudon02].

Os sistemas de informação empresariais são essenciais para as organizações na forma como se relacionam com clientes, fornecedores e parceiros. No mundo digital, o ritmo de mudança destas relações é mais rápido, o que significa que os sistemas têm que ser capazes de se adaptar mais depressa a mudança de requisitos de negócio. Os *requisitos* são *funcionais*, quando ditam o que o sistema faz, ou *não funcionais*, quando ditam qualidades do sistema, como a segurança.

Para dar resposta à necessidade de maior agilidade dos sistemas dos seus clientes empresariais, os “vértices” da indústria informática – Microsoft, IBM, Sun e Oracle – propõem actualmente a tecnologia de Web Services (WS) e as arquitecturas orientadas a serviços. Os *serviços* são uma forma de estruturar os sistemas de informação empresariais que têm como objectivo maximizar a *flexibilidade*, *reutilização* e *interoperabilidade*. A flexibilidade, em particular, traduz-se na facilidade de composição funcional dos serviços e na possibilidade de configuração de aspectos não funcionais em tempo de instalação ou mesmo em tempo de execução.

As implementações da plataforma de Web Services em Microsoft Dot Net e Java, têm tecnologias de base já consolidadas, como XML, SOAP, WSDL, UDDI, e extensões ainda experimentais. Uma das extensões consideradas mais prioritárias é a *segurança*, designada por WS-Security, pois é uma condição indispensável para sistemas que manipulam informação com valor. Para este fim, foram propostas normas e implementações de serviços seguros que, no entanto, não foram ainda suficientemente avaliadas em utilizações reais e complexas. A ênfase da segurança de Web Services não é desenvolver novas técnicas de segurança mas sim encontrar formas de integrar as tecnologias que

já existem e cujo uso está consolidado na prática, como acontece com certificados digitais X.509, Kerberos, etc.

Esta dissertação centra-se na *avaliação de serviços seguros*, ou seja, na avaliação das normas propostas e das implementações disponíveis da WS-Security. A abordagem é baseada num caso de estudo, real e complexo, para permitir uma verdadeira avaliação da tecnologia.

O caso de estudo é a “compra e venda de imóvel” e os resultados foram obtidos com a implementação de um protótipo. O objectivo inicial era escolher uma implementação de serviços seguros para realizar o protótipo que fosse completa no suporte às normas. No entanto, as diversas limitações encontradas nas implementações disponíveis obrigaram a realizar previamente ensaios práticos em todas elas. Estes ensaios, apesar de não terem sido inicialmente planeados, contribuíram significativamente para os resultados obtidos.

Os principais contributos desta tese são:

- O levantamento da tecnologia de Web Services, abrangendo em largura as normas e implementações de toda a plataforma, e em profundidade a segurança;
- A avaliação da tecnologia de serviços seguros a partir do caso de estudo;
- A identificação dos mecanismos necessários para permitir a implementação de segurança em plataformas de serviços.

1.1. Área da tese

As áreas científicas deste trabalho são os sistemas de informação no âmbito de utilização empresarial e a segurança de sistemas distribuídos.

1.2. Objectivo

O objectivo principal foi a avaliação das normas e implementações de segurança para Web Services. Este objectivo foi decomposto nos seguintes objectivos instrumentais:

- Caracterização dos sistemas de informação estruturados em serviços e dos desafios para garantir a sua segurança;
- Estudo da plataforma tecnológica de Web Services, com ênfase nas normas de segurança;
- Análise de um caso de estudo real e complexo com desenvolvimento de protótipo para avaliação da tecnologia de serviços seguros.

1.3. Metodologia

A metodologia adoptada para avaliação foi a realização de um protótipo de um caso de estudo, precedido pela realização de ensaios. O caso de estudo avalia o *desenvolvimento de serviços* como sistemas de informação empresariais flexíveis, e a *protecção de serviços* nos mecanismos de autenticação, autorização, protecção de mensagens e configuração.

1.4. Organização do texto

O documento está estruturado em oito capítulos.

- **Capítulo 1 – Introdução** – motivação, área científica da tese, objectivo e metodologia;
- **Capítulo 2 – Enquadramento** – definição de conceitos sobre sistemas de informação empresariais, serviços, requisitos e segurança;
- **Capítulo 3 – Plataforma de serviços** – levantamento das normas e implementações da plataforma de Web Services, abrangendo tecnologias de base e extensões;
- **Capítulo 4 – Segurança de serviços** – análise das normas e implementações de segurança para Web Services e definição dos objectivos de avaliação;
- **Capítulo 5 – Caso de estudo** – contexto organizacional, processo de negócio, cenários exemplificativos e protótipo;
- **Capítulo 6 – Avaliação** – apreciação dos resultados obtidos com o protótipo do caso de estudo no que respeita a desenvolvimento e protecção de serviços;
- **Capítulo 7 – Conclusão** – apresentação das conclusões, incluindo as contribuições, o trabalho futuro e o comentário final;
- **Capítulo 8 – Bibliografia.**

2. Enquadramento

Neste capítulo apresentam-se os conceitos que enquadram a restante dissertação, abrangendo os sistemas de informação empresariais, os serviços, os requisitos de negócio e a segurança informática.

2.1. Sistemas de informação empresariais

Os sistemas de informação empresariais são construídos para facilitar a manipulação de informação de acordo com as necessidades do negócio. As aplicações enfrentam um conjunto de desafios que tornam a sua implementação tecnicamente relevante.

2.1.1. Negócio

O *cliente* é o centro do negócio, pois é ele quem requer produtos ou serviços a um *fornecedor*, mediante pagamento ou contracto prévio. O *valor* é a avaliação por parte do cliente da capacidade geral do produto ou serviço para satisfazer as suas necessidades [Kotler99].

O fornecedor encabeça uma *cadeia de valor*, para a qual contribuem várias organizações. O *processo de negócio* é o modo particular como as organizações se coordenam e se organizam em actividades de trabalho, informação e conhecimento para produzir o produto ou serviço [Laudon02]. Uma *entidade informacional* é um conjunto de informação de negócio utilizada nos processos [Inmon93].

Os *sistemas de informação* facilitam a recolha de dados, a apresentação de informação e apoiam o trabalho de conhecimento das pessoas. Como tal, acrescentam valor aos processos de negócio, o que motiva as organizações a procurarem ter mais e melhores sistemas [Laudon02].

2.1.2. Sistemas de informação

Um *sistema de informação* é formado por vários componentes funcionais interrelacionados que trabalham em conjunto para recolher, processar, armazenar e disseminar dados e informação para suportar a tomada de decisões, a coordenação, o controlo, a análise e a visualização [Laudon02].

Uma *aplicação* informática é um conjunto de componentes de um sistema de informação que são executados através de programas de computador com repositórios persistentes de dados.

A Figura 1 mostra algumas das tarefas de uma organização que podem ser realizadas com sistemas de informação, nos vários níveis e funções da organização.

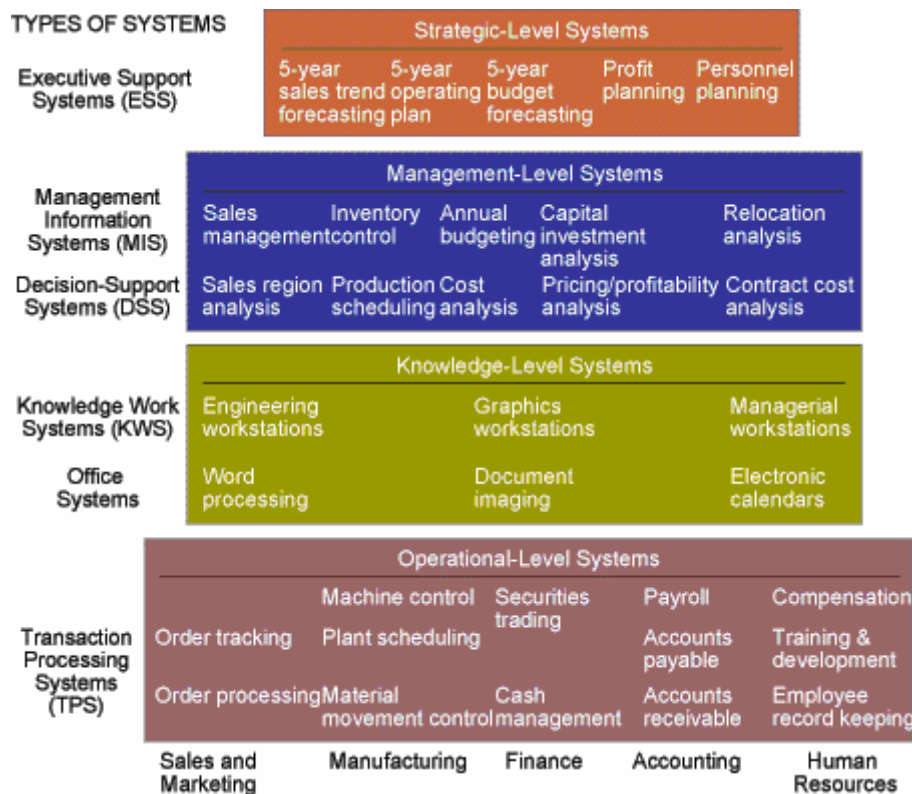


Figura 1 – Exemplos de diferentes tipos de sistemas de informação nos vários níveis e funções da organização [Laudon02].

Os sistemas de informação podem ser classificados de acordo com a utilização que permitem:

- *Sistemas operacionais* (TPS – Transaction Processing Systems) – sistemas do nível operacional, que registam as transacções elementares necessárias aos processos de negócio;
- *Sistemas de escritório* (OWS – Office Work Systems) – sistemas de apoio a trabalho de escritório;
- *Sistemas de trabalho de conhecimento* (KWS – Knowledge Work Systems) – sistemas de apoio a trabalhadores especializados em áreas de conhecimento, na criação e integração de novo conhecimento na organização;
- *Sistemas de apoio à decisão* (DSS – Decision Support Systems) – sistemas ao nível de gestão que combinam dados e modelos analíticos para apoiar decisões não rotineiras;
- *Sistemas de informação de gestão* (MIS – Management Information Systems) – sistemas ao nível de gestão utilizados para funções de planeamento, controlo e decisão baseados em relatórios rotineiros de resumo ou de excepção;

- *Sistemas de apoio executivo* (ESS – Executive Support Systems) – sistemas ao nível estratégico concebidos para auxiliar em decisões não rotineiras, utilizando técnicas de comunicação e representação gráfica.

A necessidade de ter diferentes aplicações para diferentes utilizações, coloca o desafio de gerir correctamente a informação da organização, nomeadamente para evitar problemas graves de incoerência e redundância. Este problema é difícil de resolver, especialmente se apenas existir uma visão local a cada sistema. Uma *arquitectura de sistemas de informação* formula uma visão global coerente para o conjunto de sistemas de informação da organização, com o objectivo de alinhar as aplicações com os processos de negócio, para melhorar a gestão dos recursos informacionais [Spewak93].

2.1.3. Aplicações empresariais

As aplicações empresariais distinguem-se de outras aplicações informáticas nos seguintes aspectos [Fowler02]:

- Complexidade e quantidade de *dados*;
- Diversidade e quantidade de *utilizadores*;
- Complexidade das *regras de negócio*;
- Necessidades de *integração*, dentro e fora da organização;
- Variedade e complexidade das *ferramentas* de desenvolvimento.

Os *dados* das aplicações empresariais são estruturalmente complexos, existem em grandes quantidades e têm que ser persistentes. Além disso, têm que ser representados de diferentes formas para diferentes usos, o que normalmente implica replicação. Adicionalmente, têm que ser acedidos concorrentemente por vários utilizadores, o que implica gestão de transacções para garantir a sua consistência.

As interfaces utilizador são também vastas, com dezenas de ecrãs de interacção, com *utilizadores* ocasionais ou regulares e com diferentes níveis de qualificação técnica.

As *regras de negócio* são complexas e existem em grande quantidade. Por este motivo, a lógica de negócio pode aparentar “não ter lógica nenhuma”, no sentido em que existem muitas condições e casos especiais que se combinam, implicando testes muito extensivos.

As necessidades de *integração* surgem porque as aplicações empresariais raramente estão isoladas, tendo que se ligar a outras aplicações. Para dificultar, as outras aplicações provavelmente foram construídas em momentos diferentes, com tecnologias diferentes. Também poderão já existir integrações anteriores, que usam ainda mais tecnologia diversa. Todo este problema agudiza-se em

integrações com parceiros de negócio. Mesmo que o problema da tecnologia seja resolvido ou minorado, continuam a existir problemas na semântica dos processos de negócio e das entidades informacionais.

Como consequência dos outros aspectos, as *ferramentas* necessárias para o desenvolvimento e manutenção das aplicações empresariais são várias e potencialmente complexas, exigindo pessoal técnico devidamente qualificado para as operar.

2.2. Serviços

A implementação de aplicações empresariais implica vários desafios técnicos. Os serviços são uma proposta de tecnologia que pretende ser mais adequada à sua construção.

2.2.1. Características

Uma aplicação baseada em serviços pretende ser *mais ágil*, ou seja, com *menores custos de adaptação*, tendo em conta as circunstâncias da sua utilização e a constante evolução dos requisitos de negócio ao longo do tempo [Endrei04].

Um *serviço* é uma unidade básica de disponibilização de recursos funcionais e informacionais para aplicações empresariais. Para dar resposta aos problemas das aplicações empresariais, os serviços pretendem garantir as seguintes características: *flexibilidade, reutilização e interoperabilidade*.

Para aumentar a flexibilidade, deve existir compromisso mínimo entre as partes envolvidas, pretendendo-se que clientes e servidores fiquem *fracamente interligados*¹. Esta fraca dependência pode ser conseguida com comunicação exclusivamente por mensagens, com encapsulamento de recursos e com autonomia em relação a outros serviços. Por exemplo, a utilização de comunicação com mensagens assíncronas permite uma maior independência de funcionamento entre duas aplicações.

Para aumentar a reutilização, os serviços devem disponibilizar funcionalidades com *granularidade grossa*, bem maior do que objectos individuais de negócio, de forma a que faça sempre mais sentido usar o que existe em vez de fazer de novo, devido ao custo associado.

Para garantir a interoperabilidade, os serviços são *baseados em normas* e são independentes de implementações e plataformas computacionais específicas. É também fundamental que a tecnologia tenha *licenciamento aberto*.

¹ *loosely coupled*.

2.2.2. Arquitectura orientada a serviços

Uma *arquitectura orientada a serviços* (SOA – Service-Oriented Architecture) tem objectivos similares aos de uma arquitectura de sistemas de informação “clássica”, mas usa o serviço como unidade básica do seu modelo, definindo interfaces e mensagens para aceder aos recursos.

Para construir uma arquitectura de serviços bem sucedida, é necessário identificar primeiro quais são as entidades informacionais e quem é responsável pela sua gestão. Depois deve ser definido e disponibilizado um serviço como única forma de acesso à informação para garantir a coerência [Sousa04].

Os serviços podem depois ser orquestrados ou coreografados em processos de negócio. A *orquestração* define o fluxo de controlo e de dados na invocação de serviços. A *coreografia* é mais abstracta do que a orquestração, optando por distinguir os fluxos de dados públicos dos privados, especificando as pré e pós-condições para a troca de mensagens. Por esta razão, a coreografia é mais adequada para efectuar a colaboração de serviços entre organizações distintas [Papazoglou03].

Na arquitectura de serviços é dada grande ênfase à gestão da meta-informação. A componente central desta gestão é o *registo de serviços* que funciona como repositório de nomes, localizações e descrições dos serviços. Para usar um serviço publicado, um cliente pesquisa o registo, descobre os contratos, vincula-se a eles e invoca o serviço correspondente. Segue-se a sequência “pesquisar-vincular-invocar”² [Baglietto02].

Para suportar uma arquitectura de serviços à escala da organização, existem propostas para uma *central de serviços* (ESB – Enterprise Service Bus)³, responsável pelo controlo, fluxo e eventuais traduções de mensagens entre serviços, suportando vários protocolos de comunicação. Esta abordagem surge como uma evolução orientada a serviços dos sistemas de filas de mensagens assíncronas e persistentes. A central de serviços agrega as seguintes responsabilidades: registo de serviços, gestão dos serviços e gestão do contexto e fluxo dos processos de negócio em execução [Endrei04].

A arquitectura de serviços está fortemente associada aos *Web Services*, mas os mesmos conceitos podem ser aplicados a outras tecnologias.

²“*find-bind-invoke*”

³ Apesar da designação de “central” de serviços, o ESB pode ser distribuído com vários servidores para garantir maior disponibilidade.

2.2.3. Tecnologia relacionada

A principal tecnologia de serviços são os *Web Services*, que resultam de uma iniciativa da indústria de tecnologias de informação e comunicação para criar uma plataforma universal de serviços para aplicações empresariais.

O ebXML [Grangard01] é uma tecnologia paralela aos *Web Services* com alguma sobreposição de normas e objectivos. O ebXML usa o mesmo formato de mensagens que os *Web Services*, mas tem registo e orquestração próprios. O ebXML está especializado em aplicações negócio-para-negócio, na evolução de EDI [Guerra05].

Para além dos *Web Services* e do ebXML, existem outras tecnologias orientadas a serviços: JINI e Grid Computing, que não se aplicam directamente a aplicações empresariais. Os principais objectivos do JINI são a descoberta automática de serviços e a auto-reparação. A concepção está orientada à utilização de dispositivos móveis em redes ad-hoc [Tanenbaum03]. Os objectivos do Grid Computing são a descoberta automática e o alistar dinâmico de recursos computacionais para resolver um problema, enquanto que nas aplicações empresariais as máquinas são dimensionadas à partida, a população de utilizadores é bem definida e o nível de qualidade de serviço é acordado previamente [Sheil06].

A tecnologia de sistemas distribuídos que antecedeu os *Web Services* foi a CORBA [OMG91] cuja formulação era orientada a objectos e não a serviços. A adopção da CORBA não foi a esperada, devido sobretudo a falhas técnicas nas normas e implementações, a bibliotecas de programação excessivamente complexas e a insuficiências na segurança [Henning06].

2.3. Requisitos de sistemas de informação

Um sistema de informação é definido por requisitos e é avaliado na medida em que os satisfaz ou não.

2.3.1. Perspectivas

A *framework de Zachman* [Zachman87], que surge aqui adaptada, define perspectivas para descrever um sistema de informação:

- Negócio (conceptual);
- Sistema (lógica);
- Tecnologia (física).

A Figura 2 resume as perspectivas, os seus principais artefactos, e as relações que existem entre as perspectivas: traduzir e detalhar.

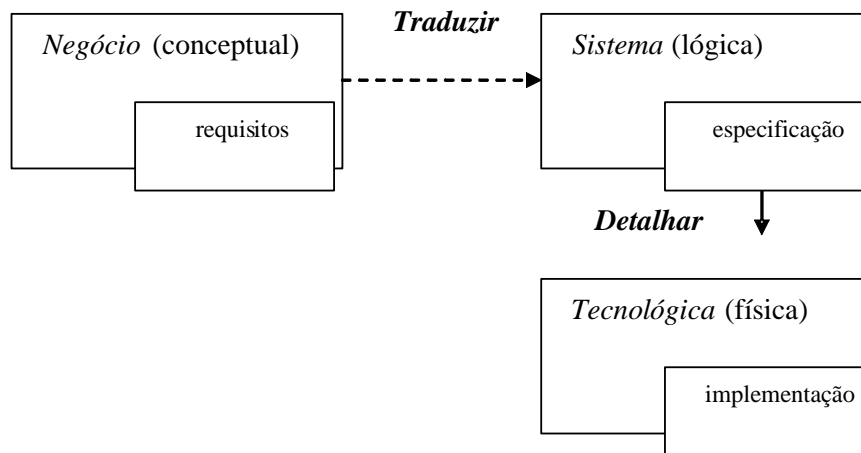


Figura 2 - Perspectivas de um sistema de informação.

A perspectiva de *negócio* (conceptual) descreve o sistema com entidades informacionais (ex. Cliente, Fornecedor) e com processos de negócio (ex. Fazer Encomenda, Aceitar Reclamação). Os processos definem os *requisitos*.

A perspectiva de *sistema* (lógica) descreve o sistema formal ou semi-formalmente, mas ainda de forma abstracta. Ou seja, é produzida a *especificação*.

A perspectiva de *tecnologia* (física) descreve a concretização do sistema em bases de dados e programas concretos. Ou seja, é produzida a *implementação*.

A perspectiva de negócio é *traduzida* para a perspectiva de sistema, ou seja, os conceitos de negócio informais são representados com objectos ou com outro formalismo. Isto é feito de forma não automática.

A perspectiva de sistema é *detalhada* na perspectiva de tecnologia, onde a descrição abstracta é vinculada a tecnologia concreta. Por exemplo, os objectos abstractos são implementados com objectos Java.

O sistema de informação cumpre os objectivos para que foi desenhado, quando as perspectivas estão alinhadas, ou seja, os *requisitos* são satisfeitos pela forma como foi efectuada a *especificação* e depois a *implementação*.

2.3.2. Tipos de requisitos

Os requisitos de um sistema de informação podem ser divididos em dois tipos: funcionais ou não funcionais. Os *requisitos funcionais* dizem o que o sistema deve ser capaz de fazer. Por exemplo, o sistema deverá ser capaz de efectuar encomendas electrónicas, validando o inventário de produtos. Os *requisitos não funcionais* dizem quais as qualidades que o sistema deverá apresentar no seu

funcionamento. Por exemplo, o sistema deverá garantir a confidencialidade das encomendas e deverá permitir o acesso ao inventário apenas a utilizadores autorizados.

Os requisitos não funcionais abrangem:

- Segurança;
- Fiabilidade e tolerância a faltas;
- Supervisão: gestão e controlo;
- Desempenho;
- Ergonomia e usabilidade.

Os diferentes requisitos não funcionais têm que ser equilibrados entre si na especificação e implementação do sistema, pois muitas vezes são contraditórios. Por exemplo, o sistema de encomendas seria mais fácil de utilizar se não precisasse de senhas de acesso, mas seria certamente menos seguro.

A flexibilidade pretendida para a implementação dos sistemas com serviços relaciona-se principalmente com os requisitos não funcionais, pois as qualidades dos serviços devem adaptar-se às circunstâncias da sua invocação. Por exemplo, no que respeita à segurança, deve ser possível ajustar o nível de protecção ao valor dos dados do serviço, bem como ao cliente que o invoca e à infra-estrutura que está a ser utilizada. Por exemplo, um sistema de encomendas poderá aceitar a autenticação de um utilizador com uma verificação de senha ou com uma credencial de um outro domínio de segurança no qual confie.

Esta flexibilidade só vai ser possível de garantir se a plataforma de serviços permitir que a implementação da parte funcional do serviço seja independente da parte não funcional [Baligand04]. Os requisitos funcionais devem ser implementados como *componentes*, que se podem estruturar e compor em procedimentos⁴. Os requisitos não funcionais devem ser implementados como *aspectos*, que permitem acrescentar comportamento de forma independente dos componentes. A forma de o conseguir passa pela utilização de *padrões de desenho* [Gamma95] que flexibilizam a estrutura do sistema e as relações entre os componentes, ou mesmo por novos paradigmas de programação, como as *linguagens orientadas a aspectos* (AOP – Aspect Oriented Programming) [Kiczales97].

⁴ Procedimento, método ou função conforme o paradigma de programação.

2.4. Segurança

A *segurança* define-se como a protecção de algo que é valioso contra apropriação ou utilização indevida. Nos sistemas de informação, a protecção incide sobre os dados e a utilização de recursos computacionais e comunicacionais [Marques98].

2.4.1. Agente, acção e recurso

Tradicionalmente, a segurança de uma aplicação informática é representada com os conceitos abstractos de: agente, acção e recurso. O *agente*⁵ pode ser uma pessoa, organização ou programa informático. A *acção* é a funcionalidade. O *recurso* são os dados e outros meios necessários à acção. A Figura 3 representa visualmente os três conceitos.

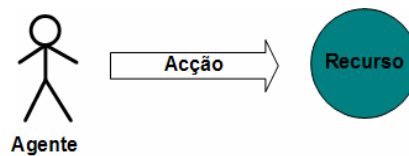


Figura 3 – Agente, acção e recurso de uma aplicação informática.

2.4.2. Ataques

A aplicação pode sofrer ataques que exploram vulnerabilidades da sua implementação ao nível do agente, da acção ou do recurso, tal como representado na Figura 4. Por exemplo, o agente pode ser personificado por alguém que assume a sua identidade; a acção pode ser manipulada para executar uma funcionalidade não autorizada; o recurso pode ficar indisponível devido a muitas falsas tentativas de acesso em simultâneo.

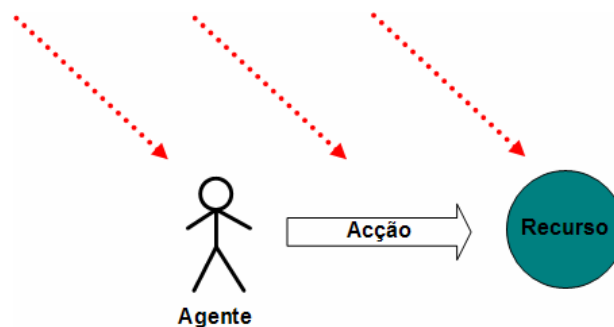


Figura 4 – Ataques ao agente, acção e recurso de uma aplicação informática.

⁵ Do inglês, *principal*.

2.4.3. Defesas

A *política* de segurança define o que deve ser defendido e de quem, e os *mecanismos* que concretizam a defesa.

Política de segurança

A política é definida após uma análise de riscos da aplicação e descreve sucintamente a estratégia de protecção contra um determinado conjunto de ataques, respondendo às questões:

- O que devemos proteger?
- De quem?
- Qual o custo da protecção?

Depois de definida a política de segurança, é necessário identificar qual a *base computacional de confiança* (TCB – Trusted Computing Base), que define o conjunto de mecanismos e recursos necessários para implementar a política [Marques98].

Mecanismos de segurança

Os mecanismos implementam e verificam a política dificultando a realização de ataques. As propriedades que os mecanismos primitivos de segurança garantem são as seguintes (ver Figura 5):

- Autenticação dos agentes;
- Autorização e não-repúdio das acções;
- Disponibilidade, integridade e confidencialidade dos recursos.

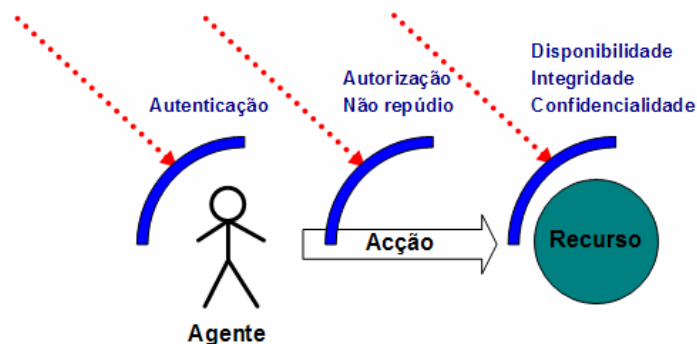


Figura 5 – Defesas de agente, acção e recurso de uma aplicação informática.

A *autenticação* é a verificação de identidade do agente. A *autorização* é verificação do direito de um agente para executar uma acção sobre um recurso. O *não-repúdio* é a existência de provas de que uma acção foi executada por um agente sobre um recurso. As provas têm que garantir a autenticação do agente e a integridade dos dados e das próprias provas. A *disponibilidade* é definida como a probabilidade do recurso estar disponível num instante no tempo, admitindo que ocorreram falhas

acidentais ou provocadas, mas que foram recuperadas através de manutenção adequada. A *integridade* garante que os dados não são modificados de forma não autorizada. A *confidencialidade* garante que os dados não são lidos de forma não autorizada [Anderson01].

A base computacional de confiança de um sistema distribuído é constituída por:

- *Canais de comunicação seguros* – baseados em protocolos criptográficos e na necessária gestão de chaves;
- *Autenticação dos agentes* – baseados na posse de chave, por exemplo, através de uma assinatura digital;
- *Autorização* – com *listas de controlo de acessos* associadas aos recursos ou *capacidades* associadas aos agentes.

Criptografia

A *criptografia* [Smith97] define formas de transformar dados em claro em dados cifrados que são difíceis de inverter sem a chave. Um *algoritmo criptográfico simétrico* usa a mesma chave, designada por chave secreta, para cifrar e decifrar dados. Um *algoritmo criptográfico assimétrico* usa pares de chaves diferentes, designadas por chave pública e por chave privada, para cifrar e decifrar.

Uma *assinatura digital* é um anexo de um documento ou mensagem que permite garantir a integridade do conteúdo e a autenticação do produtor desses dados. A assinatura é produzida criptograficamente a partir do documento ou mensagem usando uma chave privada. A verificação de integridade e autenticação usa a chave pública correspondente.

Um *certificado digital* é um documento digitalmente assinado por uma autoridade de certificação para garantir integridade e autenticidade do seu conteúdo. Um *certificado de chave pública* contém uma chave pública e o nome da entidade a que pertence, juntamente com outros atributos da entidade certificada e do próprio certificado. Uma *credencial* é um certificado que atesta a autenticação de um agente ou a autorização de utilização de um recurso. Um *comprovativo* é um certificado que atesta a ocorrência de um facto passado. Uma *autoridade de certificação* realiza as seguintes operações:

- Verificação de afirmações do certificado;
- Codificação e assinatura do certificado;
- Armazenamento de certificado em directório ou outro repositório;
- Renovação de chave e actualização de certificado;
- Revogação de certificado.

2.4.4. Agente, acção e recurso no contexto de um serviço

A representação com agente, acção e recurso é adequada para aplicações informáticas tradicionais, mas a distribuição, o encapsulamento e a granularidade dos serviços torna-a demasiado simplista.

A Figura 6 apresenta uma representação possível para um serviço. O *agente do cliente* faz a invocação enviando um *pedido* pela rede. No servidor, a mensagem é recebida pelo *agente do serviço*, que vai executar acções sobre recursos. No fim, é enviada a *resposta*.

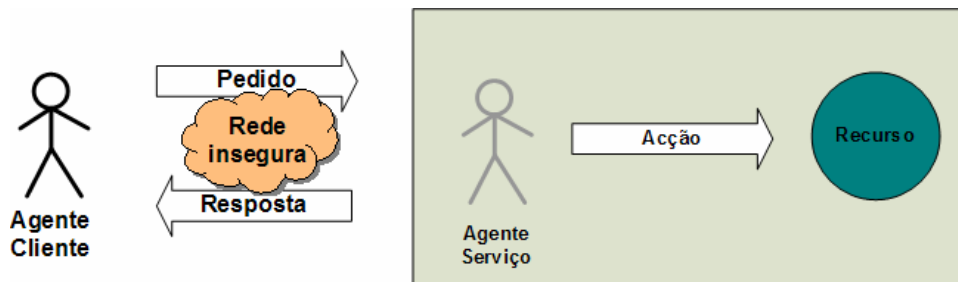


Figura 6 – Agente, acção e recurso no contexto de um serviço.

A interacção entre o cliente e o serviço é efectuada sobre uma rede de comunicação que normalmente não garante a protecção dos dados. O princípio da suspeição mútua⁶ obriga à validação cuidadosa de todas as acções.

2.4.5. Confiança

A partir do momento em que as aplicações de negócio usam recursos em localizações remotas ou pertencentes a outras entidades, a confiança torna-se um bem necessário.

Definição

A *confiança* é uma crença quantificada de um *confiante* na competência de um *confiado* para um dado fim [Essin98]. Por exemplo, um utilizador da Web confia na chave pública de uma empresa contida num certificado digital. O utilizador é o confiante. A autoridade de certificação que emitiu o certificado é a confiada. Por intermédio desta, a empresa certificada é também confiada.

Uma entidade em quem se confia é alguém que nos pode prejudicar (trair) sem que se possa fazer nada contra isso a não ser minorar os danos. Uma falha de um confiado pode quebrar a política de segurança definida para o sistema.

A confiança pode ser baseada na verificação de factos ou não. Existe sempre um limiar em que a confiança é baseada em experiências anteriores. Por exemplo, um utilizador confia no certificado de

⁶ Nunca se deve confiar no interlocutor, quer seja cliente ou servidor.

uma empresa não porque se tenha deslocado pessoalmente à autoridade certificadora para verificar se os seus processos são de facto seguros, mas sim porque já utilizou anteriormente o certificado e o sistema funcionou sempre como esperado. No dia em que não funcione, a confiança pode quebrar-se.

Modelos de confiança

Um *modelo de confiança* [Gaston03] permite representar as relações de confiança necessárias para a colaboração de diferentes entidades num sistema. O *modelo de confiança entre duas entidades* representado na Figura 7 é o mais simples. O cliente envia um pedido ao servidor. O servidor avalia o pedido e a confiança que tem no cliente e decide aceitar ou rejeitar.

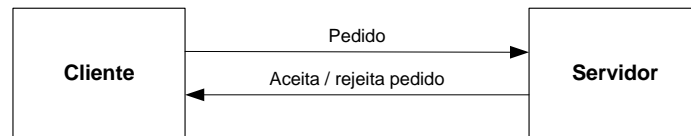


Figura 7 – Modelo de confiança entre duas entidades.

A confiança entre duas entidades pode desenvolver-se por si só, mas é um processo gradual que demora tempo. Quando a escala é alargada ou se pretendem tempos mais reduzidos para estabelecer confiança é necessário outro modelo. O *modelo de três entidades* representado na Figura 8 inclui uma entidade terceira confiada, a quem o servidor pede referências sobre o cliente. Com base nas referências, o servidor decide aceitar ou rejeitar o pedido.

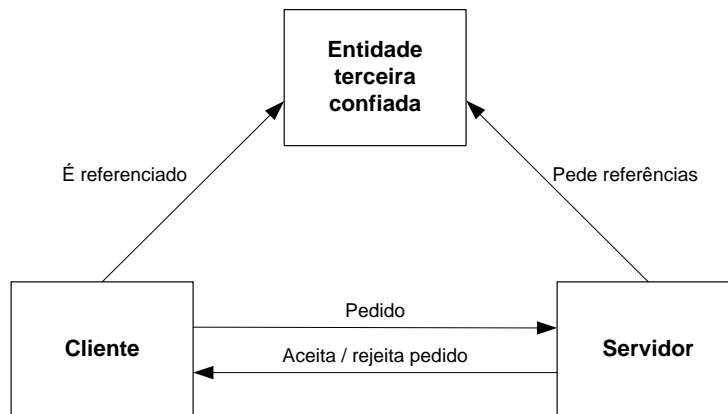


Figura 8 – Modelo de confiança com três entidades.

Para que este modelo possa ser aplicado, a entidade terceira terá que, pelo menos, já confiar no cliente e ser confiada pelo servidor. Adicionalmente poderá também confiar no servidor e ser confiada pelo cliente. Uma entidade terceira confiada consegue transmitir confiança dentro de um domínio de segurança em que a sua autoridade é reconhecida. No entanto, este modelo não permite estender confiança a diferentes domínios. O *modelo de quatro entidades* representado na Figura 9 inclui duas entidades terceiras, cada uma associada a um domínio de segurança.

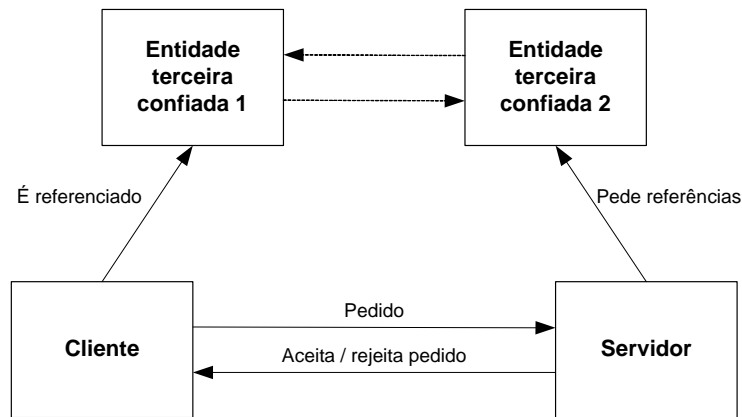


Figura 9 – Modelo de confiança com quatro entidades.

Para que um cliente e servidor de diferentes domínios possam interagir, é necessário que as entidades terceiras tenham uma relação prévia de confiança. Depois disso o servidor pode pedir referências à entidade terceira do seu domínio, que pede referências à entidade terceira do domínio do cliente. O servidor analisa as referências obtidas e decide aceitar ou rejeitar o pedido. Este modelo de quatro entidades pode ser generalizado, construindo-se uma cadeia de referências de confiança com comprimento arbitrário. No fim, é sempre o servidor que analisa a cadeia produzida e que decide se aceita ou rejeita o pedido.

2.5. Resumo

Neste capítulo enquadra-se a dissertação nos sistemas de informação empresariais.

Os sistemas de informação são importantes para as organizações, pois permitem-lhes agilizar os seus processos de negócio, e desta forma, servir melhor os seus clientes. No entanto, as aplicações empresariais, pela sua natureza, colocam uma série de desafios de implementação cuja resolução não é trivial. Destacam-se a gestão da informação e a mudança constante das regras de negócio que têm impacto nas integrações com outras aplicações.

Os serviços foram propostos como elementos estruturantes das aplicações empresariais, sendo uma tecnologia mais flexível, com maior potencial de reutilização e com interoperabilidade. Os serviços são a unidade de representação, quer ao nível de arquitectura, onde estão propostas as SOA, quer ao nível da tecnologia, onde estão propostos os Web Services.

Para descrever um sistema de informação empresarial, é necessário um modelo adequado, que permita representar requisitos funcionais e não funcionais, com implementação em componentes e aspectos, respectivamente.

As propriedades das aplicações seguras são a autenticação de agentes, a autorização e não-repúdio de acções e a protecção de recursos. Ao usar serviços, estas propriedades devem continuar a ser garantidas, mas a estrutura do sistema é mais elaborada do que o simples agente, acção e recurso. Os pedidos são efectuados por um agente no cliente que envia mensagens por redes não necessariamente seguras até ao serviço, onde são recebidas por um agente do serviço que efectua a validação e a execução, decompondo o pedido de maior granularidade em várias acções mais elementares que actuam sobre diferentes recursos.

No próximo capítulo é apresentada a plataforma de Web Services.

3. Plataforma de serviços

Neste capítulo descreve-se em detalhe a plataforma de Web Services, através das normas e das implementações existentes. Existem as normas base, que já estão consolidadas e que são suportadas de forma praticamente universal, e as extensões WS-*, que na sua maioria, estão ainda em estados mais atrasados de desenvolvimento e de aceitação [Hogg04]. As implementações existentes são agrupadas em “gerações” para analisar as suas capacidades efectivas face às previstas nas normas.

3.1. Web Services

Do ponto de vista técnico, um Web Service define uma *interface funcional* que pode ser invocada remotamente para dar acesso a recursos. A designação Web deve-se à inspiração na World Wide Web e na forma como os recursos são referenciados, por *identificadores uniformes de recursos* (URIs – Uniform Resource Identifiers)⁷. O cenário apresentado na Figura 10, resume a visão de funcionamento dos serviços, tirando partido de todas as suas potencialidades.

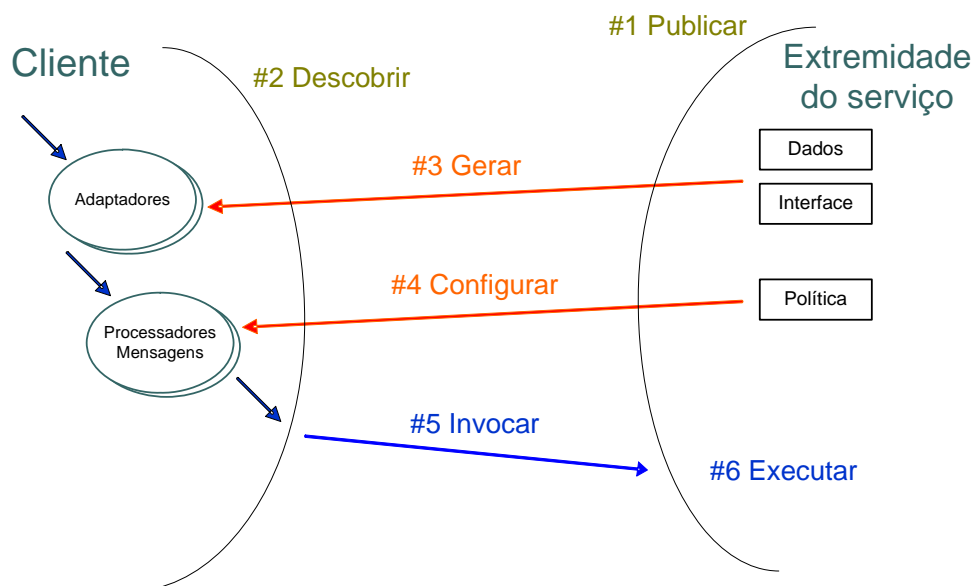


Figura 10 – Fases de interação do cliente com o serviço.

As fases de interação do cliente com o serviço são a *descoberta* (#1 publicar, #2 descobrir), a *vinculação* (#3 gerar, #4 configurar) e a *invocação* (#5 invocar, #6 executar). A extremidade de serviço é criada e publicada no registo de serviços. O cliente pesquisa o registo para descobrir a localização do

⁷ Existem duas variantes, o *Uniform Resource Locator (URL)* que indica como e onde aceder ao recurso e o *Uniform Resource Name (URN)* que é globalmente único e independente da localização.

serviço e para aceder a toda a meta-informação. O contrato de dados descreve os tipos de dados. O contrato de interface descreve as mensagens e permite ao cliente gerar *adaptadores* que vão converter os seus tipos de dados para a representação das mensagens. Adicionalmente, o cliente obtém o contrato de política e vai configurar processadores de mensagens e bibliotecas de suporte a requisitos não funcionais. O cliente invoca o serviço, enviando uma mensagem. A mensagem é recebida e verificada e o serviço é executado.

3.1.1. Princípios técnicos fundamentais

Para dar coerência global ao desenho da plataforma de Web Services, foram definidos os seguintes *princípios técnicos fundamentais* [Curbera05] que devem ser respeitados pelas normas e implementações:

- *Orientação a mensagens* – os serviços comunicam exclusivamente por mensagens, que têm um tempo de vida útil que se pode estender para além do acto de transmissão num dado transporte;
- *Encapsulamento* – os serviços são descritos em contratos normalizados e públicos, mas a sua implementação é mantida privada;
- *Autonomia* – cada serviço pode ser gerido de forma individual e tem o mínimo de dependências para outros serviços;
- *Composição de protocolos* – os protocolos utilizados pelos serviços são estruturados em blocos que podem ser compostos à medida das necessidades efectivas de uma dada aplicação;
- *Interoperabilidade baseada em normas* – não se assume nenhum pressuposto para além dos que são explicitados nas normas.

Dentro destes princípios, a plataforma de serviços terá que ser capaz de satisfazer *requisitos não funcionais* – segurança, mensagens fiáveis, transacções, etc. – para poder ser efectivamente utilizada em processos de negócio com valor [Kreger03].

3.1.2. Normas

A plataforma de Web Services é definida de forma modular através de *normas*, que especificam as regras técnicas que as implementações têm que respeitar. A Figura 11 classifica as normas em categorias: representação de dados, interoperabilidade, transporte, mensagem, contrato, descoberta, segurança, mensagens fiáveis, transacções, processos de negócio e gestão [Pardal06].

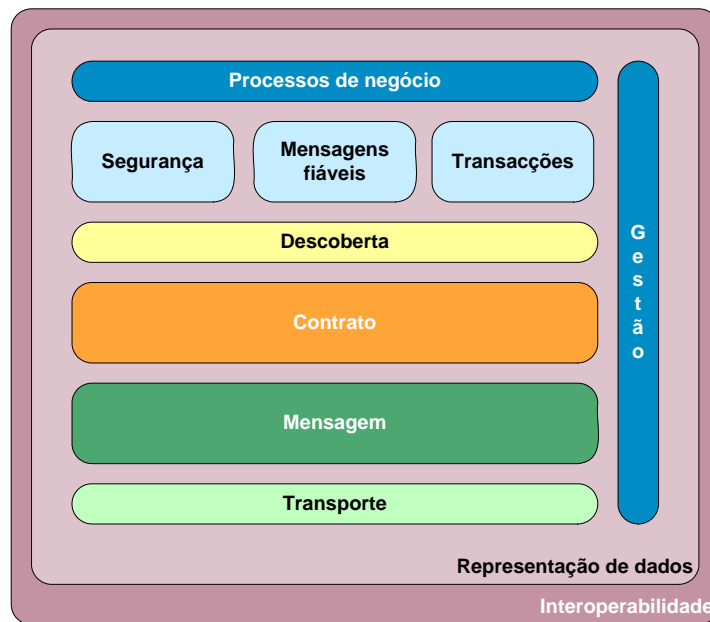


Figura 11 – Classificação das normas de Web Services em categorias [Pardal06].

Existem vários *processos de normalização* actualmente em curso para cobrir todo o espaço de requisitos para aplicações empresariais com serviços Web [Wahli05]. As normas são propostas inicialmente por empresas, sendo depois discutidas e finalizadas em *organizações de normalização* como o IETF, o W3C, a OASIS, o WS-I ou outras. O Anexo B tem mais informação sobre estas organizações.

As principais empresas promotoras até à data têm sido: a Microsoft [Microsoft05], a IBM [IBM05], a Sun [Sun05] e a Oracle [Oracle05], acompanhadas pela generalidade da indústria de tecnologias de informação e comunicação.

3.2. Plataforma base

As normas base da plataforma são identificadas na Figura 12 e englobam: a representação de dados, a interoperabilidade, o transporte, a mensagem, o contrato e a descoberta. Com esta base é possível satisfazer a generalidade de requisitos funcionais dos serviços.

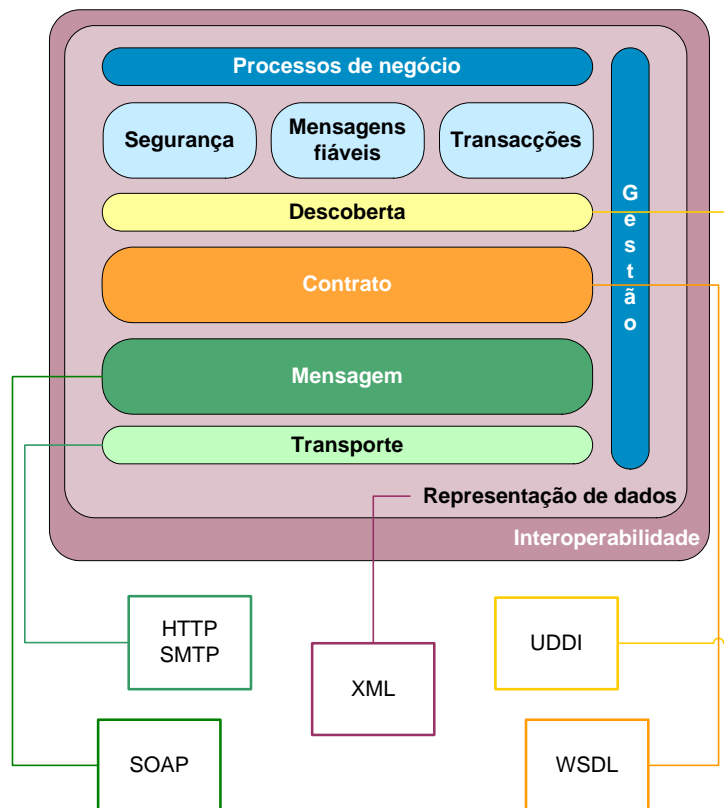


Figura 12 – Normas base dos Web Services [Pardal06].

As secções seguintes detalham estas normas base e outras que permitem construir serviços.

3.2.1. Representação de dados

O problema da heterogeneidade na representação de dados é transversal a todas as outras normas e foi resolvido com XML [Bray04] como formato canónico e com XML Schema (XSD) [Fallside04] para especificar formatos de documentos. O XML é uma linguagem textual de etiquetas que permite representar dados de uma forma estruturada e auto-descritiva. O XML Schema é uma gramática para definir esquemas de documentos, incluindo: elementos, atributos, ordem, cardinalidade, tipos de dados e valores por omissão. Os documentos que respeitam a sintaxe da XML dizem-se *bem-formatados* e os que adicionalmente respeitam um esquema dizem-se *válidos*. O XML Schema permite especificar e validar os tipos de dados que são transportados em mensagens de serviços. O XPath é uma sintaxe para referenciar elementos ou atributos dentro de um documento XML.

3.2.2. Interoperabilidade

O problema da interoperabilidade vai para além da representação de dados em XML e centra-se na compatibilidade de diferentes implementações das mesmas normas.

A *Web Services Interoperability Organization (WS-I)* [WSI05] é uma organização que junta os principais fornecedores de ferramentas e define perfis de interoperabilidade. Cada perfil abrange um conjunto de normas e fornece: orientações à implementação, aplicações de exemplo e testes para aferir o cumprimento da especificação. Actualmente existe um perfil de base (*WS-I Basic Profile*) [Ferris04] e um perfil de segurança (*WS-I Basic Security Profile*) [Barbir05]. A Figura 13 representa visualmente o papel de mediação desempenhado pela WS-I entre as organizações de normalização e a indústria.

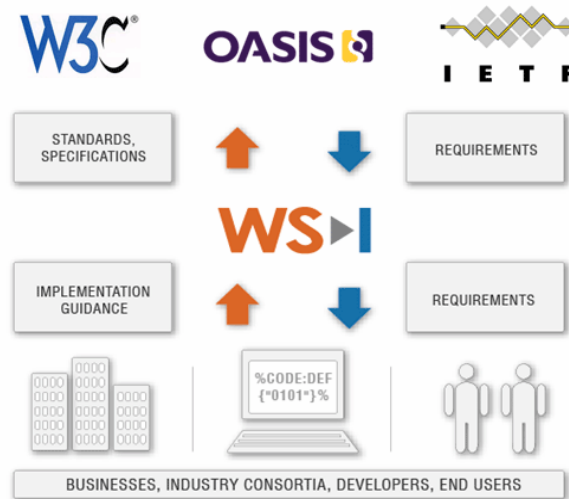


Figura 13 – Mediação entre organizações de normalização e indústria desempenhado pela WS-I [WSI05].

O *WS-DeviceProfile* (Device Profile for Web Services) [Schlimmer05b] não é definido pela WS-I e tem um âmbito diferente. Neste caso pretende-se orientar a escolha de um subconjunto de normas para implementação em dispositivos com recursos limitados, tentando encontrar um equilíbrio entre as capacidades ricas que estão disponíveis e aquelas que são mais importantes [Cabrera04].

3.2.3. Transporte

As normas de transporte resolvem o problema de estabelecer um canal de comunicação entre o cliente e servidor. A comunicação pode ser síncrona ou assíncrona. Os transportes mais comuns são o HTTP [Fielding99] e o SMTP [Klensin01]. Existem também implementações que recorrem a sistemas de filas de mensagens assíncronas, sendo no entanto soluções proprietárias.

Adicionalmente, o transporte pode oferecer garantias de fiabilidade ou segurança. Por exemplo, o HTTPS [Rescorla00] e HTTPR [Parr02] são alternativas para aumentar, respectivamente, a segurança e fiabilidade da comunicação com HTTP.

3.2.4. Mensagem

As normas de mensagem definem a estrutura das unidades de comunicação e as formas como são trocadas entre serviços.

Os conceitos elementares das mensagens são: mensagem, operação, interface, vínculo de interface, extremidade e serviço [Schlimmer02]. Uma *mensagem* é a unidade básica de comunicação entre um cliente e o Web Service. Uma *operação* é uma sequência de mensagens relacionadas com uma única invocação. Uma *interface* é um agrupamento lógico de operações, que define um tipo abstracto e independente do protocolo de transporte e do formato dos dados. Um *vínculo de interface* (interface binding) associa a interface ao protocolo de transporte concreto e ao formato de dados a usar. Uma *extremidade* (endpoint) indica a localização da interface de serviço vinculada a um dado protocolo e um dado formato de dados. O *endereço* na rede é especificado através de um URI. Finalmente, um *serviço* é uma colecção de extremidades.

SOAP

O SOAP [Gudgin03] é uma das normas mais importantes dos Web Services, pois define o protocolo de mensagens para os serviços e garante a independência do transporte utilizado. As definições SOAP incluem o formato das mensagens, a serialização de dados e os vários modos de interacção entre cliente e servidor.

O *envelope SOAP* é o documento XML da mensagem, que separa o *cabeçalho* com dados de sistema para serem processados pela plataforma, do *corpo* com dados de negócio para serem processados pelas aplicações. O cabeçalho permite a composição de protocolos, pois cada elemento de extensão indica a sua versão e a opcionalidade da sua interpretação. O corpo contém os dados de negócio da mensagem ou então o elemento *SOAP Fault* com informação de erro.

A serialização de dados pode ser efectuada com as *regras de codificação SOAP* (encoding rules) ou com os tipos XML Schema. A segunda alternativa é actualmente a preferida.

Os *modelos de interacção* entre cliente e servidor previstos no SOAP incluem: mensagem unidireccional, pedido e resposta⁸, notificação simples (callback), e notificação com resposta.

Qualquer agente que envie ou receba mensagens é denominado por *nó SOAP*. O nó que envia a mensagem pela primeira vez é o *emissor original*. O nó final que consome e processa a mensagem é o *receptor último*. Qualquer nó que processe a mensagem entre ambos é um *intermediário*. A colecção de

⁸ Corresponde a uma chamada de procedimento remoto (RPC – Remote Procedure Call)

nós intermédios atravessados pela mensagem é denominada como o *caminho da mensagem* [Cabrera04].

A *semântica* de uma chamada remota de procedimento em SOAP depende do transporte utilizado. Por exemplo, se for usado HTTP, o procedimento é executado “no máximo uma vez”.

MTOM/XOP

Para transportar dados binários em SOAP usa-se MTOM/XOP que substitui as abordagens anteriores baseadas em anexos aos envelopes: SOAP with Attachments (SwA) [Barton00] e WS-Attachments/DIME [Nielsen02b] [Cabrera04]. O XOP (XML-binary Optimized Packaging) [Gudgin05a] é uma codificação de dados que permite intercalar dados binários opacos com XML tradicional baseado em texto, através de um formato MIME [Freed96] com múltiplas partes, sem recorrer a codificação de base 64. O SOAP MTOM (Message Transmission Optimization Method) [Gudgin05b] especifica como usar XOP em mensagens SOAP.

WS-Addressing

O WS-Addressing [Box04a] permite fazer o *endereçamento* e *encaminhamento* de mensagens SOAP de forma independente do transporte. O WS-Addressing define elementos de cabeçalho para identificar a mensagem, os agentes envolvidos e a acção de processamento pretendida. O conceito fundamental é a *referência de extremidade*, que permite um endereçamento mais fino do que o permitido por um único URI, com propriedades e parâmetros adicionais [Cabrera04].

WS-Enumeration

O WS-Enumeration [Geller04a] permite criar uma sessão para enumerar sequências de dados, de forma gradual, englobando vários pedidos e respostas.

WS-Eventing / WS-Notification

A *notificação assíncrona* de eventos com Web Services permite evitar consultas sucessivas (polling) a serviços através da interacção entre subscritores, gestores de subscrição, produtores e consumidores de eventos. Existem duas propostas concorrentes para notificações assíncronas: o WS-Eventing [Geller04c] da Microsoft e o WS-Notification [Graham04] da IBM.

WS-Polling

O WS-Polling [Davis05] define mecanismos para fazer consultas sucessivas quando não é possível usar as notificações assíncronas, devido às duas extremidades da comunicação não terem uma ligação

bidireccional entre ambas, por exemplo, devido a uma firewall. Nestes casos, uma das extremidades tem que periodicamente estabelecer ligação para verificar se há novidades.

3.2.5. Contrato

As normas de contrato entre o cliente e o servidor resolvem o problema da descrição da interface, da política e dos recursos do serviço.

WSDL

A Web Services Description Language (WSDL) [Booth05] especifica a interface do serviço. Os tipos de dados são descritos com XML Schema. A *interface* define um contrato abstracto com operações e mensagens. O *vínculo de interface* define um contrato concreto indicando a representação das mensagens e o transporte a usar. Estes contratos são usados por ferramentas de desenvolvimento para gerar *adaptadores de invocação* (stubs).

A versão 1.1 da WSDL é a mais utilizada e está centrada na chamada remota de procedimento. A versão 2.0 permite explicitar o *padrão de troca de mensagens* (MEP – Message Exchange Pattern), bem como o mecanismo de correlação entre mensagens.

A WSDL é necessária mas não é suficiente para descrever um contrato de serviço. Além da interface, é necessário descrever também aspectos operacionais, como a versão do SOAP a utilizar, aspectos comerciais, como os custos de utilização, e aspectos não funcionais, como a segurança [Kreger03].

WS-Policy

A WS-Policy [Schlimmer06] é uma linguagem para definir políticas de serviços. A *política* define os requisitos adicionais que têm que ser cumpridos pelo cliente e pelo prestador do serviço para que a interacção entre ambos possa acontecer. O objectivo é que a negociação de requisitos e a configuração possa ser feita de forma automática [Samaranayake06].

A gramática de asserções e operadores de política é especificada na WS-PolicyAssertions. Cada asserção é satisfeita se e só se o requisito que especifica foi cumprido pelo cliente e pelo prestador. Os operadores permitem combinar várias asserções. A forma de anexar políticas ao serviço, por exemplo, através da WSDL ou da UDDI, é especificada no WS-PolicyAttachment.

A forma normal de uma WS-Policy tem a estrutura apresentada na Figura 14. Nesta forma, é necessário escolher exactamente uma das alternativas contidas dentro de “ExactlyOne” e é necessário satisfazer todas as asserções dentro de “All”.


```
<wsp:Policy ... >
  <wsp:ExactlyOne>
    ( <wsp:All>
      ( <Assertion ...> ... </Assertion> ) *
    </wsp:All> ) *
  </wsp:ExactlyOne>
</wsp:Policy>
```

Figura 14 – Forma normal de uma WS-Policy [Schlimmer06].

Basta ao cliente suportar uma das alternativas, mesmo que desconheça o vocabulário das restantes. Esta característica é muito importante para permitir a evolução das políticas sem comprometer o funcionamento de versões anteriores.

As *asserções* de políticas são opacas para o motor de processamento, sendo depois necessário definir significados específicos para elas, em normas externas à WS-Policy.

Os *operadores* de políticas são: a normalização, a intersecção e a junção. A *normalização* é o processo de converter uma política para a forma normal. Esta conversão preserva o significado lógico da política original. A *intersecção* é o processo de isolar as alternativas de política de duas políticas e é útil no cenário cliente-servidor. Se a intersecção for vazia, então o cliente não consegue interagir com o serviço. A *junção* de política é o processo de criação de uma única política a partir de duas políticas. É uma operação útil no servidor, para juntar a política do serviço a outras existentes em âmbitos mais alargados, obtendo-se assim a *política efectiva* do serviço [Samaranayake06]. As operações com políticas são detalhadas no Anexo C.

WS-PolicyConstraints

A WS-PolicyConstraints [Anderson05] está a ser desenvolvida pela Sun Microsystems e propõe a interpretação de políticas com um motor único, em vez de motores específicos para cada domínio não funcional, como a segurança, por exemplo. Para já, ainda não existem implementações disponíveis.

WS-Transfer / WS-ResourceFramework

Os *recursos* dos serviços são entidades informacionais, com representação em XML, identificadas por um URI e que podem ser criadas, lidas, actualizadas e apagadas. Para descrever e gerir explicitamente estes recursos existem duas abordagens concorrentes: WS-Transfer [Geller04b] da Microsoft e WS-ResourceFramework [Czajkowski04] da IBM.

3.2.6. Descoberta

As normas de descoberta definem formas de publicar e pesquisar serviços.

UDDI

A UDDI (Universal Description, Discovery and Integration) [Clement04] define um directório, que permite a publicação e a pesquisa de serviços de forma dinâmica. O domínio de informação da UDDI permite três tipos de pesquisa: por tipo de serviço no que respeita a área de negócio e capacidades pretendidas (páginas amarelas), por contactos da empresa (páginas brancas), e por pontos de acesso públicos ao serviço (páginas verdes).

WS-MEX

O WS-MEX (WS-MetadataExchange) [Curbera04] define um protocolo de acesso aos contratos WSDL e WS-Policy de um serviço e outros eventuais, permitindo assim aos serviços serem auto-descritivos.

WS-Inspection

A WS-Inspection [Ballinger01] define documentos de inspecção com referências para descrições. Os documentos de inspecção podem referenciar registos em directórios UDDI, mas podem também indicar outros tipos de descrição.

WS-Discovery

A WS-Discovery [Schlimmer05a] define protocolos de descoberta de serviços associados a dispositivos, através de multi-difusão de mensagens em redes locais.

3.3. Implementações da plataforma

Nas secções anteriores foram descritas algumas das mais importantes normas de Web Services mas não foram mencionadas implementações concretas da plataforma. Nesta secção serão referidas as principais implementações e as suas capacidades, numa perspectiva de evolução até à actualidade e fazendo-se uma breve antevisão das próximas versões.

3.3.1. Antes dos Web Services

Os Web Services são uma tecnologia nova, mas não são uma tecnologia inovadora. Já antes existia tecnologia de suporte à distribuição com modelos de programação idênticos ou até mais sofisticados, como é o caso do DCE [Lockhart94], CORBA [OMG91], DCOM [Eddon98] e Java RMI [Sun97]. A diferença está na ênfase, que no caso dos Web Services, se centra na flexibilidade, reutilização e interoperabilidade.

O ponto de partida para os Web Services foi, sem dúvida, o surgimento da linguagem XML em 1998, como forma de representação de dados. As primeiras utilizações de XML foram “ilhas de dados” para preencher tabelas em páginas HTML. Rapidamente se propuseram formas simples de trocar apenas XML entre cliente e servidor, cuja mais bem sucedida foi o XML-RPC [Dumbill01]. A ênfase inicial deste esforço, muito ligado ao desenvolvimento de páginas Web, foi a simplicidade.

No entanto, a utilização de XML para transporte de dados sofreu uma revolução em 2002, quando a Microsoft lançou a plataforma Dot Net, com uma tecnologia de chamada de procedimento remoto, de aplicações para aplicações, baseada em XML, designada por Web Services. Esta entrada “pela porta grande” fez dos Web Services uma tecnologia de topo, que ascendeu rapidamente na lista de prioridades da restante indústria, concorrente da Microsoft. Teve início a primeira geração de implementações da plataforma de Web Services.

3.3.2. Primeira geração (Dot Net, Axis, JAX-RPC)

A primeira implementação de Web Services foi o Microsoft Dot Net [MacDonald03]. Seguiu-se depois uma implementação de código aberto na plataforma Java, o Apache Axis [Graham01]. No seguimento deste primeiro esforço, a Sun Microsystems liderou a implementação de referência para a plataforma Java, o JAX-RPC [McGovern03], que foi depois implementado em produtos da Sun, IBM, Oracle e BEA.

Os “traços genéticos” desta geração são: a utilização das normas SOAP 1.0/1.1, WSDL 1.0/1.1, UDDI 1.0/2.0 e o transporte HTTP praticamente em exclusivo.

Os principais defeitos apontados a esta geração são: a ênfase excessiva no modelo de chamada remota de procedimento sem suporte para invocações assíncronas, a utilização das regras de codificação SOAP para representar os dados em vez de XML Schemas impossibilitando a partilha de esquemas de dados e a ausência de endereçamento independente de HTTP.

As primeiras implementações de requisitos não funcionais que surgiram eram demasiado limitadas e não se podem considerar suficientemente maduras para uso em produção.

3.3.3. Segunda geração (WSE, Axis2, JAX-WS)

A segunda geração, actualmente em vigor, foi mais uma vez iniciada pela Microsoft, com o lançamento do WSE (Web Services Enhancements) que foi pioneiro na disponibilização de tecnologias de segurança e encaminhamento de mensagens independente do transporte. Houve também uma preocupação em tornar o processo de configuração declarativo e “amigável”.

A plataforma Java respondeu com uma melhoria do JAX-RPC, renomeado para JAX-WS 2 e integrado com a tecnologia JAX-B 2 para vinculação de dados Java para XML e vice-versa. Esta versão suporta

interacções assíncronas e outros tipos de transporte. O endereçamento independente do transporte ainda não está integrado na biblioteca. A implementação de referência destas novas normas está disponível no pacote JWSDP 2.0, da Sun Microsystems.

Ainda na plataforma Java, surgiu o Apache Axis2 em código aberto, com algumas propostas inovadoras, para permitir múltiplos transportes e também a configuração de múltiplos módulos externos para requisitos não funcionais. A implementação está ainda muito instável, precisando de muitos testes e de amadurecimento do seu código base.

Os “traços genéticos” desta geração são a utilização das normas: SOAP 1.1/1.2, WSDL 1.1/2.0. A codificação de dados usa XML Schema. A UDDI tem sido substituída por abordagens de auto-descrição dos serviços. As novas versões suportam invocação assíncrona de serviços e têm endereçamento independente do transporte, onde se mantém predominante o HTTP. Nesta geração existem também disponíveis implementações de requisitos não funcionais em bibliotecas isoladas, principalmente para segurança.

3.3.4. Terceira geração (WCF, WSIT)

A terceira geração, ainda em fase de desenvolvimento, não está disponível mas promete maior interoperabilidade entre as plataformas concorrentes: Dot Net e Java. A Microsoft vai lançar a Windows Communications Foundation (WCF) e a Sun Microsystems acompanha com o projecto WSIT (Web Services Interoperability Technology / Project Tango).

O WCF vai consolidar a aposta da Microsoft nos Web Services, assistindo-se a uma substituição de outros tipos de tecnologias de distribuição, nomeadamente, o Dot Net Remoting, por uma alternativa baseada em SOAP com codificação binária mais eficiente. Existe também uma aposta na auto-configuração com políticas, totalmente orientada aos serviços e aos seus princípios, sob o lema *endereços, vínculos e contratos* (ABC – Addresses, Bindings and Contracts).

A WSIT é baseada na arquitectura do JAX-WS 2, que está a ser estendida para encaixar perfeitamente com a WCF, em resultado de um acordo de cooperação entre a Microsoft e a Sun. A IBM e a Oracle, seguem, mas com maior distância mantendo para já os seus produtos na primeira geração. A Oracle tem uma aposta forte na orquestração de processos de negócio, com a implementação líder de orquestração de serviços.

O Apache Axis2 promete seguir com uma implementação mais aberta, mas para já tem ainda um esforço de consolidação pela frente, antes de se poder aferir as suas capacidades efectivas.

Os “traços genéticos” desta geração são a utilização do SOAP 1.1/1.2, da WSDL 2.0 com suporte para diversos padrões de trocas de mensagens (síncronas, assíncronas, pedido-resposta, notificação, etc), auto-descrição dos serviços, independência do transporte e suporte para filas de mensagens e

implementações de requisitos não funcionais com configuração automática por política integradas na plataforma.

3.4. Plataforma estendida

A maior parte das extensões à plataforma pretendem satisfazer requisitos não funcionais dos serviços: a segurança, as mensagens fiáveis e as transacções. O objectivo é depois tirar partido de toda esta tecnologia para orquestrar processos de negócio, com gestão integrada dos serviços e das infra-estruturas.

Cada extensão é opcional e pode ser composta com outras, à medida das necessidades da aplicação. As condições e obrigações adicionais do cliente e do serviço são descritas na política.

3.4.1. Segurança

As normas de segurança especificam mecanismos de protecção para que os serviços Web possam ser utilizados em aplicações que manipulam informação com valor. Existem duas abordagens para a segurança: no transporte ou na mensagem.

A principal norma de segurança na mensagem para serviços é a WS-Security, à volta da qual se centram outras normas especializadas, pois é ela que faz a ligação entre as tecnologias de segurança informática já existentes e a tecnologia de Web Services. A WS-Security [Nadalin04] permite a troca de tokens de segurança e garantir a integridade, a autenticação e a confidencialidade de mensagens SOAP. A WS-SecurityPolicy [Kaler05] especializa a WS-Policy para políticas de segurança.

As tecnologias de segurança para serviços são amplamente detalhadas no Capítulo 4: Segurança de serviços.

3.4.2. Mensagens fiáveis

Para dar fiabilidade à comunicação com SOAP de forma independente do transporte em aspectos como a entrega garantida, a eliminação de repetições e a correcta ordenação, existem duas propostas concorrentes: a WS-Reliability [Iwasa04] da Oracle, Sun Microsystems e a WS-ReliableMessaging [Ferris05] da IBM, Microsoft.

3.4.3. Transacções

As normas de transacções permitem ter semânticas bem definidas para os resultados de várias interacções entre serviços com recursos informacionais distribuídos. Para isso assumem modelos de faltas temporárias e recuperáveis para as máquinas e comunicações. Existem dois enquadramentos de

normas concorrentes para transacções em serviços Web: a WS-Coordination [Feingold05a] da Microsoft, IBM e a WS-CompositeApplicationFramework [Little03a] da Oracle, Sun Microsystems.

A WS-Coordination define uma base para a coordenação de serviços, que depois é especializada de formas alternativas entre si. Para situações em que os recursos podem ficar cativos durante toda a duração da transacção, existe a WS-AtomicTransaction [Feingold05b] que garante as propriedades ACID – atomicidade, consistência, isolamento e durabilidade – através de protocolos de consenso alternativos: conclusão e compromisso em duas fases (volátil e persistente). Para situações mais demoradas em que os recursos não podem ficar trancados, existe a WS-BusinessActivity [Feingold05c] que permite relaxar as propriedades transaccionais.

A WS-CompositeApplicationFramework está estruturada em três partes. A WS-Context que define operações de gestão simplificada de contexto. A WS-CoordinationFramework que especifica o coordenador para gerir o ciclo de vida do contexto e para garantir a entrega de mensagens aos participantes. A WS-TransactionManagement que define protocolos de gestão de transacções que abrangem compromisso em duas fases, transacções de longa duração e fluxos de processos de negócio, com a preocupação de garantir interoperabilidade entre diferentes gestores transaccionais.

3.4.4. Processos de negócio

As normas de processos de negócio pretendem cobrir a necessidade de ferramentas de desenvolvimento cujo nível de abstracção de conceitos seja menos técnico e mais próximo do negócio e das preocupações das pessoas da organização. Existem abordagens distintas e potencialmente complementares para este problema. Mendling [Mendling04] apresenta uma discussão mais alargada.

A WS-BPEL [Thatte03] da Microsoft e IBM é baseada em orquestração, sendo o processo representado por um grafo, em que os nós são actividades e os arcos são fluxos de controlo e de informação, permitindo a composição de serviços mais simples.

A WS-CDL [Kavantzias04] da Oracle faz a coreografia de processos de forma declarativa, especificando pré-condições e pós-condições para a execução de actividades, podendo a forma concreta como o processo é executado variar, desde que as condições continuem a ser satisfeitas.

O ASAP [Fuller05] permite definir processos através do encadeamento de serviços assíncronos, com possibilidade de intervenção humana. Como o serviço assíncrono se executa independentemente de quem o invocou, pode ser consultado várias vezes durante a sua vida, para ser modificado ou cancelado. O ASAP prevê formas de pedir informação actualizada ao serviço e de receber notificações.

A WS-RP (Web Services for Remote Portlets) [Kropp03] especifica formas de integrar serviços Web na interface utilizador de um portal, que agrega conteúdos de diversas fontes.

A WS-XL (Web Services Experience Language) [Diaz02] é um modelo de componentes para serviços Web, que permite construir aplicações através da composição de outras e com distribuição por múltiplos canais de distribuição: Internet, redes móveis, etc. Todos os serviços componentes WSXL implementam operações base para a gestão do ciclo de vida, processamento das entradas do utilizador e apresentação de resultados.

3.4.5. Gestão

À medida que mais processos de negócio da organização forem sendo suportados por serviços, mais importante se tornará a gestão de toda a plataforma que tem duas facetas: a gestão dos serviços em si e a gestão das máquinas e redes de dados que os suportam. Neste momento o esforço de normalização está concentrado na segunda faceta, existindo duas normas concorrentes: a WS-Management da Microsoft e a WS-DistributedManagement da IBM.

A WS-Management [Geller04d] define os requisitos mínimos de implementação para um conjunto de operações comuns para a gestão de sistemas, que permitem: descobrir a presença de recursos, gerir parâmetros de configuração dos recursos, enumerar entradas em registos de actividade (logs), subscrever eventos de notificação assíncrona e suportar operações de gestão específicas.

A WS-DistributedManagement [Sedukhin05] também especifica operações de gestão dos recursos dos sistemas. O WS-Provisioning [Woods03] define regras de interoperabilidade entre sistemas de aprovisionamento para a atribuição de recursos e privilégios a utilizadores.

3.5. Edifício normativo

O edifício normativo dos Web Services é vasto e tem várias alternativas. A Figura 15 apresenta todas as normas identificadas em cada categoria. As actuais indefinições são assinaladas com setas e interrogações.

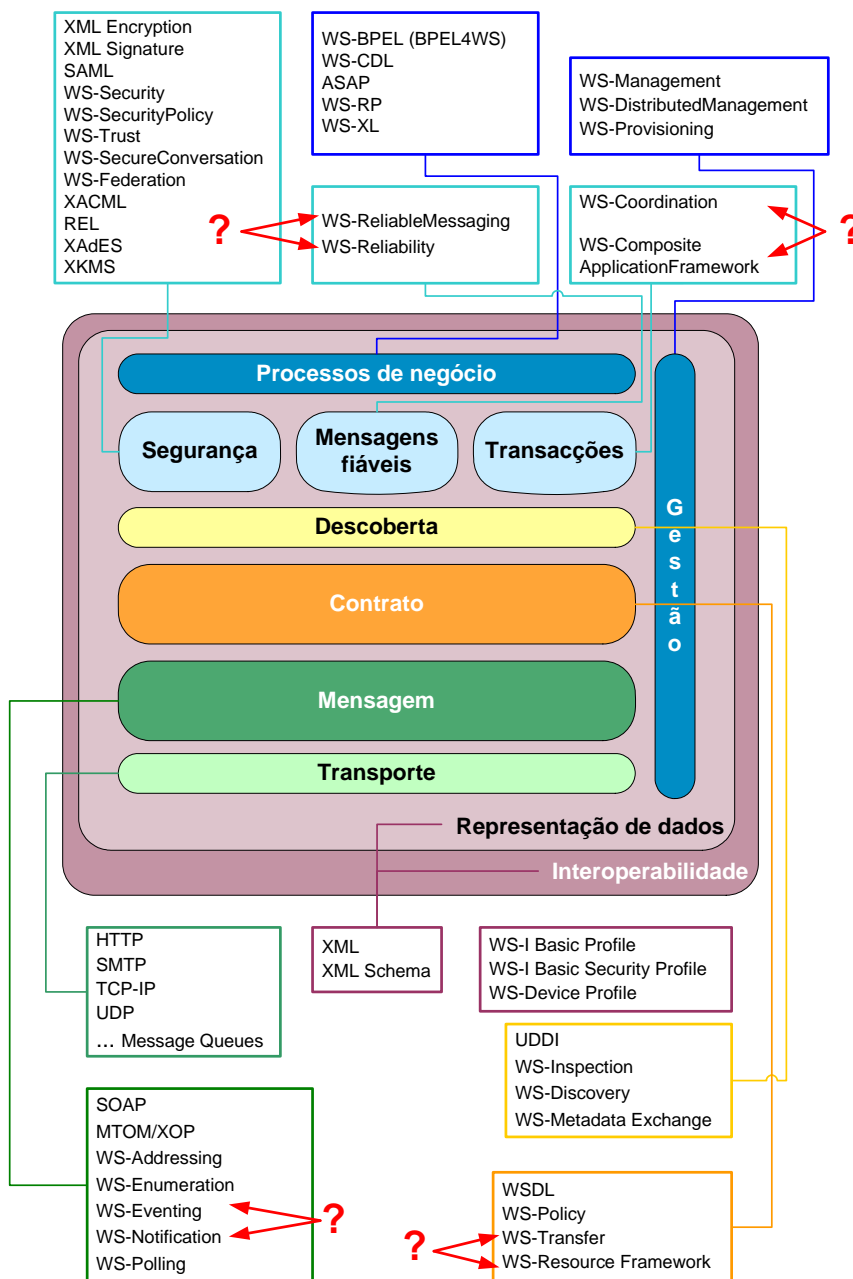


Figura 15 – Normas de Web Services. Adaptado de [Pardal06].

Cada implementação da plataforma define o seu próprio edifício com as normas que suporta. Por exemplo, as normas suportadas pela implementação referência do JAX-RPC [McGovern03], da primeira geração para a plataforma Java, são apresentadas na Figura 16.

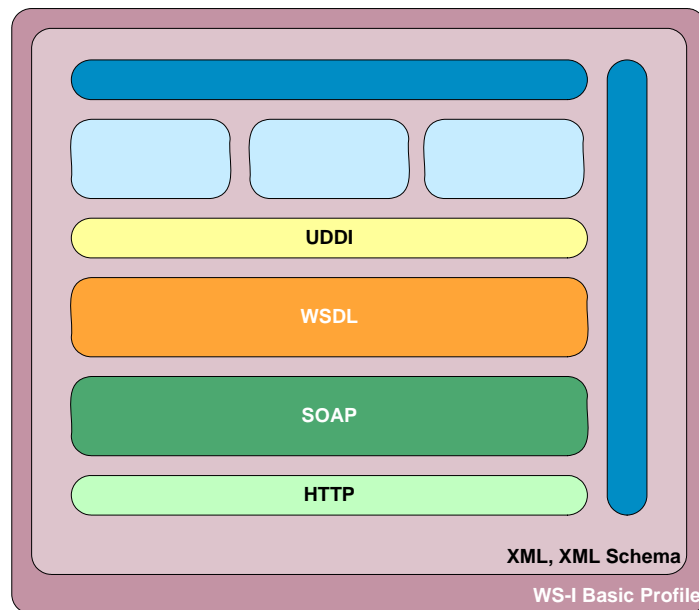


Figura 16 – Normas de Web Services suportadas pelo JAX-RPC.

O JAX-RPC suporta apenas serviços com transporte HTTP, mensagens SOAP, contrato WSDL e publicação em registo UDDI. A representação de dados usa XML e XML Schema. A implementação respeita o perfil de interoperabilidade WS-I Basic Profile, o que garante que será capaz de “falar” com outras implementações que também o suportem.

3.6. Resumo

Este capítulo descreveu a plataforma de Web Services, apresentando normas e implementações. A Figura 17 é uma actualização da Figura 10, onde se substituíram os termos genéricos pelos nomes das tecnologias entretanto apresentadas.

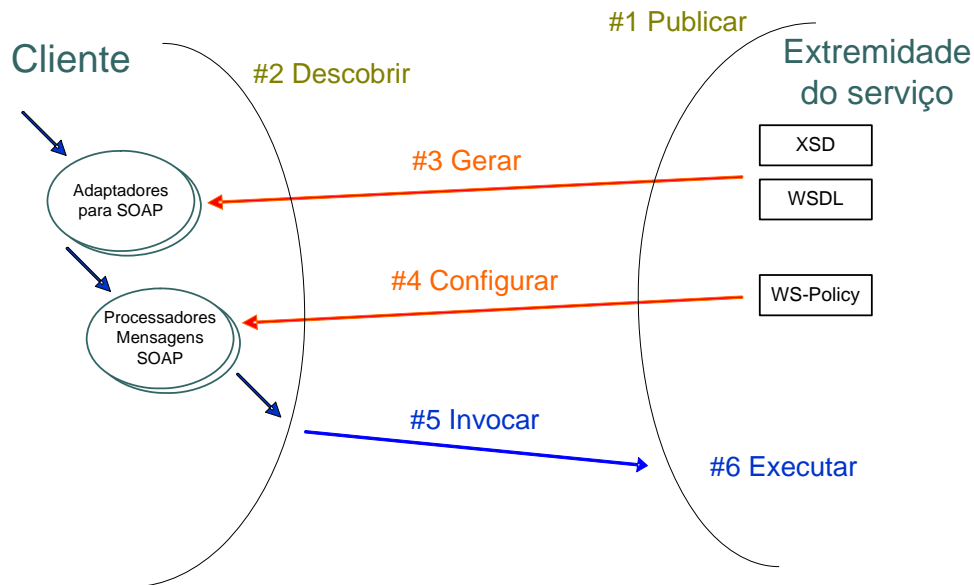


Figura 17 – Fases de interacção do cliente com o serviço, com indicação das tecnologias.

A extremidade de serviço é criada e publicada num directório UDDI. O cliente pesquisa o UDDI para descobrir a localização do serviço. O WSDL descreve o formato das mensagens SOAP e o XSD os dados, permitindo gerar adaptadores (stubs) que vão converter os tipos de dados. Adicionalmente, o cliente obtém a WS-Policy que estipula que as mensagens para o serviço devem ser assinadas. O cliente configura um processador de mensagem SOAP que usa uma biblioteca WS-Security para efectuar a assinatura digital da mensagem. O cliente faz a invocação do serviço com HTTP, enviando a mensagem assinada. A mensagem é recebida pelo serviço, que utiliza uma biblioteca WS-Security para verificar a assinatura. De seguida, o serviço é executado.

O tema do próximo capítulo é a segurança de serviços, onde as tecnologias de protecção são analisadas em detalhe.

4. Segurança de serviços

Neste capítulo é discutida a segurança de serviços, começando pelos problemas existentes e pelos mecanismos necessários para os resolver. Depois é analisada a proposta de segurança Web Services Security incluindo o modelo conceptual, as normas e as implementações disponíveis. O estado da arte é apresentado com um resumo de artigos publicados em conferências com trabalho relacionado. Finalmente são identificados os objectivos de avaliação.

4.1. Problemas a resolver

O perfil de interoperabilidade WS-I para segurança [Schwarz05] enumera os ataques que devem ser considerados sob a responsabilidade da segurança de serviços. A segurança de serviços deve preocupar-se com os ataques de falsificação e repetição de mensagens. De fora ficam os ataques às chaves e algoritmos criptográficos, à rede e aos servidores, bem como o uso de canais encobertos e os erros de programação das aplicações.

Os principais problemas a resolver pela segurança de serviços são apresentados na Figura 18 e são a protecção das mensagens, o controlo de acessos e a flexibilidade de configuração.

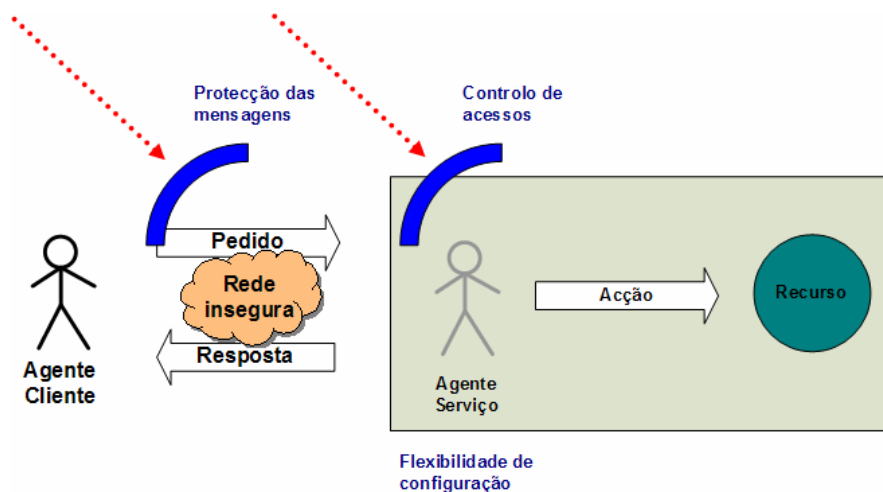


Figura 18 – Problemas a resolver na segurança de serviços.

Os serviços têm que ser seguros para que possam de facto ser a unidade estruturante de aplicações que manipulam valor significativo para as organizações. No entanto, a segurança não pode ser conseguida comprometendo a flexibilidade, reutilização e interoperabilidade que são a sua motivação originária dos serviços.

4.2. Mecanismos

Tendo em conta os problemas a resolver, os mecanismos de segurança genericamente necessários para serviços são os seguintes [Hogg05]:

- Autenticação:
 - Directa;
 - Com intermediário;
- Protecção de mensagens:
 - Confidencialidade;
 - Origem e integridade dos dados;
- Autorização de acesso a recursos:
 - Baseada em credencial de autenticação:
 - Controlo de acesso e capacidades;
 - Personificação e delegação;
 - Baseada em credencial de autorização;
- Protecção de domínios de segurança:
 - Validação de mensagens;
 - Detecção de mensagens repetidas;
 - Filtragem de excepções.

4.2.1. Autenticação

A autenticação pode ser efectuada directamente ou com intermediário.

Na *autenticação directa* o cliente e o serviço participam numa relação de confiança entre si com um modelo de duas entidades, que lhes permite trocar e validar chaves, que são válidas nesse domínio de segurança.

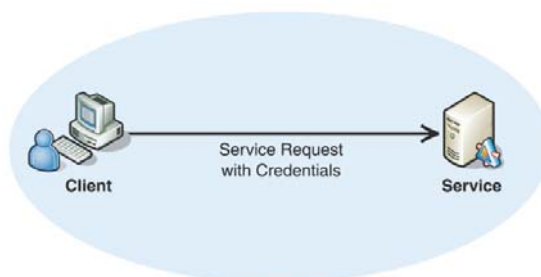


Figura 19 – Autenticação directa, quando o cliente e o serviço têm uma relação de confiança [Hogg05].

Na *autenticação com intermediário* uma entidade terceira confiada está entre o cliente e o serviço. Usa-se um modelo de confiança de três ou mais entidades.

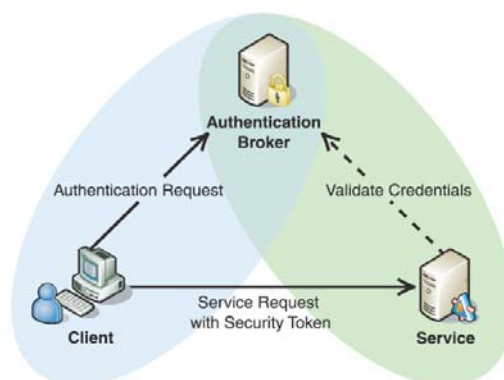


Figura 20 – Autenticação com intermediário, quando o cliente e o serviço não têm uma relação de confiança directa entre si [Hogg05].

4.2.2. Protecção de mensagens

Para garantir a *confidencialidade* de mensagens, é necessário cifrar os dados, total ou parcialmente. Para efectuar a cifra pode ser usada criptografia de chave simétrica ou criptografia de chave assimétrica.

Para garantir a *origem e integridade dos dados*, é necessário assinar os dados. A técnica mais comum é a assinatura digital com chaves assimétricas, precisamente pela diferenciação entre chave pública e chave privada, que serve de base ao não-repúdio de acções. Se apenas se pretender assegurar a integridade dos dados, pode utilizar-se uma assinatura com uma chave simétrica partilhada entre as partes.

4.2.3. Autorização de acesso a recursos

A autorização pode ser *baseada em autenticação* prévia, sendo depois verificado se existe permissão de acesso ao recurso.

A autorização com *personificação* consiste em assumir temporariamente uma outra identidade para conseguir aceder ao recurso. Na autorização com *delegação*, existe também uma personificação, mas para um propósito bem definido e não se perdendo a referência da verdadeira identidade do sujeito. Na personificação, a identidade do agente do cliente perde-se e fica só o registo do agente do serviço. Na delegação, a identidade do agente do cliente fica registada nas acções que são efectuadas pelo agente do serviço.

O uso de uma *credencial de autorização* permite aceder a um dado recurso, sem que exista uma autenticação prévia.

4.2.4. Protecção de domínios de segurança

Quando existe a necessidade de clientes efectuarem pedidos a serviços que estão num domínio de segurança diferente, é prudente ter algumas protecções a nível aplicacional, que podem ser implementadas por um nó intermediário, colocado na fronteira e que intervém ao nível das mensagens SOAP dos serviços.

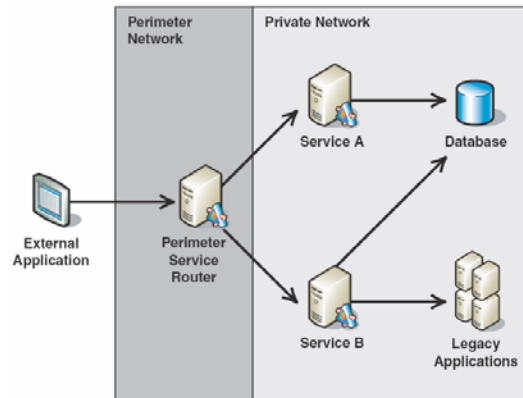


Figura 21 – Protecção da fronteira de um domínio de segurança por um nó intermediário [Hogg05].

O intermediário pode efectuar:

- A *validação de mensagens* – confirmando que estão bem formadas;
- A *deteção de mensagens repetidas* – com validações temporais e mantendo um registo temporário dos pedidos mais recentes, para evitar ataques por repetição de mensagens já transmitidas. Desta forma, evita-se que cada serviço individual tenha que ter esta preocupação;
- A *filtragem de excepções* – garantindo que, em casos de erro, não é devolvida informação interna sensível do serviço. Por exemplo, em vez de ser devolvido: “erro no acesso à tabela ‘CLIENTE’ da base de dados...”, seria devolvido: “de momento não é possível satisfazer o seu pedido, por favor tente mais tarde”.

4.3. Modelo conceptual

O modelo base de serviços seguros foi proposto inicialmente pela IBM e Microsoft [IBM02] e faz actualmente parte da norma OASIS. O modelo descreve a forma como se processam as invocações que se pretendem seguras.

Cada serviço tem uma política que define as condições de acesso e os *tokens de segurança* exigidos. Por exemplo, um serviço pode exigir que uma mensagem que recebe seja cifrada e que contenha um token com utilizador e senha. Se a mensagem chegar sem provas suficientes, o serviço pode ignorar ou rejeitar a mensagem.

A Figura 22 ilustra o modelo de serviços seguros. As setas representam comunicação entre extremidades de serviços.

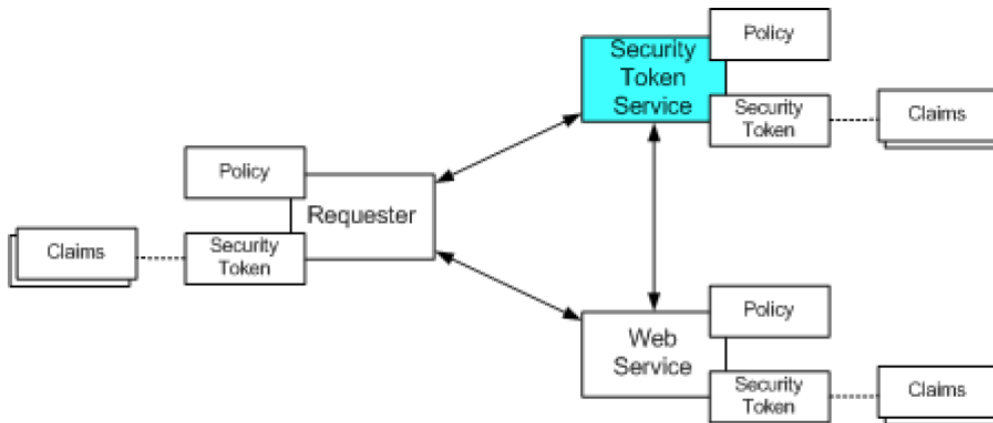


Figura 22 – Modelo de base para Web Services seguros [IBM02].

Quando o emissor da mensagem não cumpre as condições exigidas e não tem capacidade de as gerar por si, pode tentar obter tokens contactando Serviços de Tokens de Segurança (STS). Estes podem, por sua vez, exigir um outro conjunto de condições. Desta forma, os STS podem ser intermediários de segurança entre diferentes domínios, aplicando-se um modelo de confiança de três ou mais entidades.

Este modelo genérico – com políticas, condições e tokens – abstrai vários modelos mais específicos, tais como: segurança baseada em identidade, listas de controlo de acesso, capacidades, etc. O modelo permite também o uso conjunto de tecnologias existentes, como certificados de chave pública X.509 [Housley99], bilhetes de chave partilhada Kerberos [Kohl93] e resumos criptográficos de senhas.

4.4. Normas

A tecnologia de segurança para Web Services está alinhada com os princípios técnicos fundamentais, nomeadamente, a composição de protocolos e a interoperabilidade baseada em normas. As mensagens SOAP [Gudgin03] são extensíveis, permitindo incorporar informação de segurança em cabeçalhos específicos para esse efeito. A Figura 23 apresenta as normas de segurança, devidamente enquadradas na plataforma.

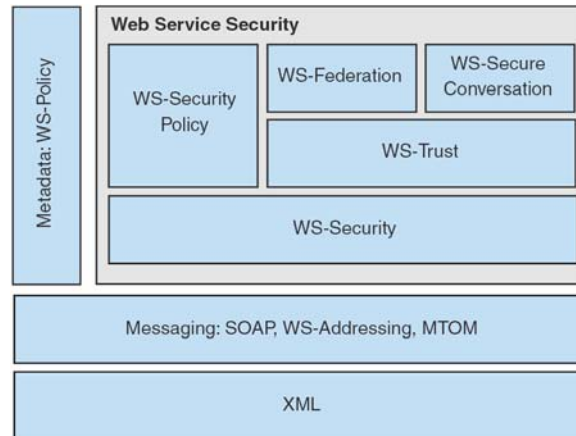


Figura 23 – Normas de segurança para serviços Web [Hogg05].

Nas secções seguintes, são descritas as características essenciais das normas de base da segurança: XML-Signature, XML-Encryption; das normas específicas de serviços: WS-Security, WS-SecurityPolicy, WS-Trust, WS-SecureConversation e WS-Federation; e de outras normas complementares: SAML, XACML, REL, XAdES e XKMS.

4.4.1. XML-Signature e XML-Encryption

A XML-Signature [Eastlake02a] e XML-Encryption [Eastlake02b] são duas normas que definem como assinar e cifrar documentos XML, respectivamente. Ambas podem ser aplicadas selectivamente a partes da mensagem ou a conteúdos externos referenciados nos documentos. A diferença mais notória entre ambas é que elemento de assinatura (Signature) *referencia* o que está a ser assinado enquanto que o elemento de dados cifrados (EncryptedData) *contém* o que está a ser cifrado [Rosenberg04].

4.4.2. WS-Security

A WS-Security [Nadalin04] permite proteger a mensagem SOAP e transportar tokens de segurança, definindo o modelo base de segurança descrito anteriormente.

A WS-Security permite as seguintes operações sobre a mensagem SOAP:

- Acrescentar data e identificação à mensagem;
- Enviar tokens de segurança no cabeçalho;
- Usar XML-Signature para assinar toda ou parte da mensagem e enviar a assinatura no cabeçalho;
- Usar XML-Encryption para cifrar toda ou parte da mensagem;
- Enviar chaves criptográficas ou referências no cabeçalho.

Existem actualmente duas versões de Web Services Security: a 1.0 (2004) e a 1.1 (2006). A versão 1.1 estende a 1.0 ao nível dos XML Schemas, corrigindo alguns problemas, e acrescentando tokens de segurança para Kerberos, SAML e REL.

4.4.3. WS-Trust

A WS-Trust [Gudgin05d] define um modelo de confiança com operações para adquirir, emitir, renovar e validar tokens de segurança e formas de criar novas relações de confiança através de serviços intermediários.

Os serviços intermediários são designados por serviço de tokens de segurança (STS). O pedido de token de segurança contém credenciais do cliente a ser autenticado, por exemplo, nome de utilizador e senha. A resposta ao pedido de token de segurança contém um token, por exemplo, com o formato SAML.

4.4.4. WS-SecureConversation

A WS-SecureConversation [Gudgin05c] permite que dois serviços estabeleçam uma sessão segura entre si para trocarem várias mensagens. Assim, é possível tirar partido de segredos partilhados para derivar chaves simétricas de sessão e ter segurança de forma mais eficiente e robusta.

O contexto de segurança da sessão é representado como um token de segurança designado por Security Context Token (SCT).

4.4.5. WS-Federation

A WS-Federation [Kaler03] define federações de serviços para partilhar informação sobre identidade, atributos, autenticação e autorização entre diferentes domínios de confiança.

4.4.6. WS-SecurityPolicy

A WS-SecurityPolicy [Kaler05] é uma especialização da WS-Policy para definir políticas de segurança que descrevem a forma como as mensagens devem ser tornadas seguras. A política pode aplicar-se a mensagens individuais, a operações ou a toda a extremidade do serviço.

As asserções WS-SecurityPolicy referem-se às funcionalidades de segurança de WS-Security, WS-Trust e WS-SecureConversation. Podem também referir segurança no transporte, como HTTPS.

As asserções descrevem os algoritmos de cifra suportados, as partes das mensagens a assinar ou cifrar, os tokens de segurança que são aceites pelo serviço e outros. Por exemplo, uma política pode especificar que a mensagem seja assinada com uma chave de certificado digital, e outra pode ditar que a autenticação seja feita com a chave de um bilhete Kerberos.

O Exemplo 1 é a WS-SecurityPolicy de um serviço. Esta política descreve apenas uma configuração, mas poderia conter diferentes alternativas.

Exemplo 1 – WS-SecurityPolicy [Kaler05].

(01)	<wsp:Policy>
(02)	<sp:SymmetricBinding>
(03)	<wsp:Policy>
(04)	<sp:ProtectionToken>
(05)	<wsp:Policy>
(06)	<sp:KerberosV5APREQToken
	sp:IncludeToken=".../IncludeToken/Once" />
(07)	</wsp:Policy>
(08)	</sp:ProtectionToken>
(09)	<sp:SignBeforeEncrypting />
(10)	<sp:EncryptSignature />
(11)	</wsp:Policy>
(12)	</sp:SymmetricBinding>
(13)	<sp:SignedParts>
(14)	<sp:Body/>
(15)	<sp:Header
	Namespace="http://schemas.xmlsoap.org/ws/2004/08/addressing" />
(16)	</sp:SignedParts>
(17)	<sp:EncryptedParts>
(18)	<sp:Body/>
(19)	</sp:EncryptedParts>
(20)	</wsp:Policy>

A linha (01) indica que se trata de uma política e que todas as asserções nela contidas têm que ser satisfeitas. A linha (02) indica um vínculo de segurança de criptografia simétrica. A linha (04) indica o token de segurança a usar para protecção e na linha (06) é especificado o token Kerberos V5 APREQ, que deve ser usado por ambas as entidades para a protecção da troca de mensagens. A linha (09) indica que as assinaturas devem ser geradas a partir dos dados em claro em vez do texto cifrado. A linha (10) indica que a assinatura deve ser cifrada. As linhas (13) a (16) indicam que partes da mensagem devem ser abrangidas pela assinatura principal, neste caso o soap:Body indicado pela linha (14) e todos os cabeçalhos SOAP dentro do espaço de nomes do WS-Addressing, indicado pela linha (15). As linhas (17) a (19) indicam que partes da mensagem devem ser cifradas; neste caso apenas o soap:Body, indicado pela linha (18).

Tal como o exemplo ilustra, existem duas secções principais na definição da política: o vínculo de segurança e a protecção.

A secção da política que define o *vínculo de segurança* (security binding) pode ser: criptografia simétrica (SymmetricBinding), criptografia assimétrica (AsymmetricBinding) e segurança no transporte (TransportBinding). Cada vínculo condiciona os tokens de segurança que podem ser utilizados, a transferência de chaves e a estrutura dos cabeçalhos de segurança.

A asserção *SymmetricBinding* assume interacção segura baseada em criptografia simétrica entre o emissor e o receptor com WS-Security. Pode ter uma das seguintes combinações de tokens de segurança, mutuamente exclusivas:

- EncryptionToken (cifra) e SignatureToken (assinatura);
- ProtectionToken (cifra e assinatura em simultâneo).

Se existir mensagem de resposta, seguem-se as mesmas indicações. As sub-asserções disponíveis para detalhar a configuração são: AlgorithmSuite (algoritmos criptográficos a utilizar), Layout (ordenação dos elementos de segurança), IncludeTimestamp (incluir marca temporal na mensagem), EncryptBeforeSign (cifrar antes de assinar), EncryptSignature (cifrar assinatura), ProtectTokens (incluir os tokens de segurança na assinatura) e OnlySignEntireHeadersAndBody (assinar apenas corpo e cabeçalhos que não os de segurança).

A asserção *AsymmetricBinding* assume interacção baseada em criptografia assimétrica com WS-Security. O InitiatorToken é usado para assinar do emissor para o receptor e para cifrar do receptor para o emissor. O RecipientToken é usado para cifrar do emissor para o receptor e para assinar do receptor para o emissor. As sub-asserções disponíveis são idênticas às da asserção *SymmetricBinding*.

A asserção *TransportBinding* indica que a segurança é garantida pelo transporte e que a correlação de segurança é assegurada por meios externos à WS-Security.

As asserções de *tokens de segurança* identificam quais são os tokens aceites e qual o processamento que lhes deve ser dado. Os tokens podem ser: HttpsToken, UsernameToken, X509Token, KerberosToken, SAMLToken e RelToken. Por exemplo, o token HttpsToken indica o uso de HTTPS, podendo também especificar se o certificado cliente é obrigatório. Existem também tokens relacionados com emissão de credenciais WS-Trust e sessões de segurança WS-SecureConversation, que são: IssuedToken, SpnegoContextToken, SecurityContextToken e SecureConversationToken.

A propriedade de processamento dos tokens mais importante é a TokenInclusion, que indica como o token de segurança deve ser usado:

- Never (nunca) – o token nunca pode ser transportado em mensagens, podendo apenas ser referenciado;
- Once (uma vez) – o token deve ser transportado uma vez na primeira mensagem, mas depois não mais. Esta opção é usada para iniciar sessões seguras;
- AlwaysToRecipient (sempre para o receptor) – o token deve ser enviado sempre que o emissor enviar uma mensagem para o receptor. O mesmo já não deve acontecer na resposta;
- Always (sempre) – o token deve ser sempre enviado nas mensagens.

A secção da política que define a *protecção* inclui:

- Integridade – indica o que deve ser assinado;
- Confidencialidade – indica o que deve ser cifrado;
- Elementos obrigatórios – que partes da mensagem têm que existir.

A granularidade da protecção pode ser especificada para todo o corpo ou cabeçalho SOAP, ou para elementos especificados com XPath.

4.4.7. SAML (Security Assertion Markup Language)

A SAML (Security Assertion Markup Language) [Cantor04] permite representar factos de segurança em formato XML. A especificação define um protocolo de comunicação e um formato de asserções, que são independentes um do outro. O *protocolo de comunicação* permite obter e validar asserções. As *asserções* permitem expressar autenticação, atributos e autorização de agentes com identidade num domínio de segurança. Os exemplos seguintes ilustram a estrutura das asserções SAML para uma autenticação, atributos e autorização, respectivamente.

Exemplo 2 – Asserção SAML de autenticação.

(01)	<Assertion>
(02)	<Conditions NotBefore="2006-07-22T12:02:00Z" NotOnOrAfter="2006-07-22T13:02:00Z">
(03)	<AudienceRestrictionCondition>
(04)	<Audience>http://www.example.com/Members</Audience>
(05)	</AudienceRestrictionCondition>
(06)	</Conditions>
(07)	<Advice>
(08)	<AssertionIDReference>id</AssertionIDReference>
(09)	<Assertion>...</Assertion>
(10)	</Advice>
(11)	<AuthenticationStatement AuthenticationMethod="urn:ietf:rfc:2246"
(12)	AuthenticationInstant="2006-07-22T12:02:00Z">
(13)	<Subject>
(14)	<NameIdentifier
(15)	Format="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress"
(16)	user@example.com
(17)	</NameIdentifier>
(18)	</Subject>
(19)	</AuthenticationStatement>
(20)	</ds:Signature>...</ds:Signature>
(20)	</Assertion>

Esta asserção comunica que o utilizador com o endereço de correio electrónico user@example.com foi autenticado com sucesso.

A linha (01) indica que se trata de uma asserção. As linhas (02) a (06) indicam as condições, que definem limites temporais e os destinatários da asserção. As linhas (07) a (10) contém informação adicional, podendo referir outras asserções, etc. As linhas (11) a (18) indicam que se trata de uma autenticação, que neste caso foi realizada com um certificado cliente SSL. Finalmente, a linha (19) contém a assinatura da entidade emissora da asserção de autenticação, para garantir a autenticidade e integridade.

Exemplo 3 – Asserção SAML de atributos de utilizador.

(01)	<Assertion>
(02)	<AttributeStatement>
(03)	<Subject>
(04)	<NameIdentifier
	Format="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
(05)	user@example.com
(06)	</NameIdentifier>
(07)	</Subject>
(08)	<Attribute AttributeName="PaidStatus"
(09)	AttributeNamespace="http://company.com">
(10)	<AttributeValue>PaidUp</AttributeValue>
(11)	</Attribute>
(12)	<Attribute AttributeName="CreditLimit"
(13)	AttributeNamespace="http://company.com">
(14)	<AttributeValue xsi:type="my:type">
(15)	<my:amount currency="EUR">500.00</my:amount>
(16)	</AttributeValue>
(17)	</Attribute>
(18)	</AttributeStatement>
(19)	<ds:Signature>...</ds:Signature>
(20)	</Assertion>

Esta asserção informa que o utilizador user@example.com tem as contas em dia e diz qual o limite de crédito que tem disponível. Os atributos são específicos a cada aplicação, que os qualifica num espaço de nomes próprio.

A linha (01) indica que se trata de uma asserção. A linha (02) indica que se trata da asserção de atributos. As linhas (03) a (07) especificam o sujeito a que se referem os atributos. As linhas (08) a (11) indicam o valor do atributo 'PaidStatus'. As linhas (12) a (17) indicam o valor do atributo 'CreditLimit'. A linha (19) é a assinatura da entidade emissora da asserção.

Exemplo 4 – Asserção SAML de autorização.

(01)	<Assertion>
(02)	<Conditions NotBefore="2006-07-22T12:02:00Z" NotOnOrAfter="2006-07-22T13:02:00Z">
(03)	</Conditions>
(04)	<AuthorizationDecisionStatement Resource="http://www.company.com/info"
(05)	Decision="Permit">
(06)	<Subject>
(07)	<NameIdentifier
	Format="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">
(08)	user@example.com
(09)	</NameIdentifier>
(10)	</Subject>
(11)	<Action Namespace="urn:oasis:names:tc:SAML:1.0:action:rwdc">Read</Action>
(12)	</AuthorizationDecisionStatement>
(13)	<AuthorizationStatement Resource="http://www.company.com/register.cgi"
(14)	Decision="Permit">
(15)	<Subject>...</Subject>
(16)	<Action Namespace="urn:oasis:names:tc:SAML:1.0:action:rwdc">Execute</Action>
(17)	</AuthorizationStatement>
(18)	<ds:Signature>...</ds:Signature>
(19)	</Assertion>

Esta asserção afirma que o utilizador user@example.com está autorizado a consultar a página <http://www.company.com/info> e que pode submeter o formulário <http://www.company.com/register.cgi>. Ambas as decisões foram permitir (Permit), mas poderiam ter sido negar (Deny) ou indeterminadas (Indeterminate).

A linha (01) indica que se trata de uma asserção. A linha (02) indica a validade temporal da asserção. As linhas (04) a (12) autorizam a leitura do recurso. As linhas (13) a (17) autorizam a execução do programa que processa o formulário. A linha (18) contém a assinatura da entidade emissora da asserção de autorização.

As asserções podem ser anexadas a mensagens e podem ser aceites em diferentes domínios de segurança, desde que, directa ou indirectamente, se confie no emissor. Qualquer entidade pode emitir asserções. Cabe depois ao receptor da asserção, verificá-la e decidir se confia no seu conteúdo ou não. O mecanismo mais usual para atribuir confiança a asserções é a assinatura digital.

4.4.8. XAdES (XML Advanced Electronic Signatures)

As assinaturas XAdES (XML Advanced Electronic Signatures) [Cruellas03] estendem as assinaturas XML-Signature para permitir o não-repúdio e o armazenamento de longo prazo. Existem diferentes níveis de segurança em assinaturas digitais XML, que são sucintamente descritos na Tabela 1.

Tabela 1 – Níveis de segurança em assinaturas digitais XML.

Nível	Conteúdo
XML-Signature	Documento + assinatura
XAdES	... + Data da assinatura + Referência do certificado de chave pública correspondente à chave privada usada para assinar + referência para o domínio / política de assinatura + (opcionalmente: local de assinatura + papel desempenhado pelo assinante + meta-dados + marcas temporais dos dados)
XAdES-T (Time-stamped)	... + Carimbo temporal da assinatura, emitido por uma autoridade de certificação de data
XAdES-C (Complete)	... + Referências para a cadeia de certificação e revogação
XAdES-X (eXtended)	... + Carimbo temporal da assinatura e das referências para validação da assinatura, emitido por uma autoridade de certificação de data
XAdES-X-L (eXtended Long-term)	... + Cópia da cadeia de certificação e revogação
XAdES-A (Archival)	... + Assinatura de toda a informação, efectuada pela autoridade de arquivo

A utilização de carimbos temporais emitidos por autoridades de certificação de data é essencial para permitir o não-repúdio. A autoridade não precisa de ter acesso aos dados, pois o carimbo é feito sobre a assinatura, que por sua vez já contém o resumo do documento. Para manter a propriedade do não-repúdio é necessário efectuar novos carimbos temporais, sempre que se renovarem as chaves da autoridade de certificação de data [Epifanio05].

4.4.9. XACML (Extensible Access Control Markup Language)

A XACML (eXtensible Access Control Markup Language) [Moses05] especifica regras de acesso à informação que permitem: o controlo fino de actividades autorizadas, a análise de características do cliente, a autorização baseada em classes de actividades e a introspecção de conteúdos.

4.4.10. REL (Rights Expression Language)

A REL (Rights Expression Language) [DeMartini04] especifica formas de proteger os direitos de autor de conteúdos acessíveis através de serviços.

4.4.11. XKMS (XML Key Management Specification)

A XKMS (XML Key Management Specification) [HallamBaker05] define protocolos para distribuir e registar chaves públicas, adequadas ao uso na XML-Signature e na XML-Encryption. A XKMS dá acesso à autoridade de certificação, o que possibilita operações de confirmação e revogação de chaves.

4.5. Implementações disponíveis

As principais implementações disponíveis de segurança para serviços são:

- WSE 3 (Web Services Enhancements 3) para Microsoft Dot Net 2 [Microsoft05c];
- WSS4J (Web Services Security for Java), sobre Apache Axis2, para Java [Apache06b];
- XWSS (XML and Web Services Security), sobre JAX-WS 2, disponível no pacote JWSDP 2.0, para Java [Sun06].

A Tabela 2 indica quais as normas suportadas por cada implementação.

Tabela 2 – Normas suportadas nas implementações disponíveis de serviços seguros.

Fornecedor	Implementação	Normas suportadas
Microsoft	WSE 3: Dot Net Framework 2.0, Visual Studio 2005, Web Services Enhancements 3.0	WS-Security: Username, X.509, Kerberos WS-Secure Conversation, WS-Trust SAML ⁹
Apache (código aberto)	WSS4J: Apache Axis2, Rampart module of Web Services Security for Java (WSS4J)	WS-Security: Username, X.509 WS-Policy SAML
Sun Microsystems	XWSS: Java Web Services Developer Pack 2.0, XML and Web Services Security 2.0	WS-Security: Username, X.509 SAML

⁹ O ambiente Microsoft suporta experimentalmente as asserções SAML, mas não o protocolo de comunicação.

4.6. Trabalho relacionado

Os serviços têm demonstrado ser um tópico de interesse para a comunidade científica, existindo actualmente três conferências internacionais anuais:

- ICWS (IEEE International Conference on Web Services), que se realiza desde 2004;
- ECOWS (IEEE European Conference on Web Services), que se realiza desde 2005;
- NWeSP (International Conference on Next Generation Web Services Practices), que se realiza também desde 2005.

A nível de modelo de aplicações seguras estruturadas em serviços, Gutierrez [Gutierrez05] deparou-se com o problema de existirem diversas normas de serviços mas não existir uma metodologia de desenvolvimento para aplicá-las. No seu artigo propõe um processo que acompanha todo o ciclo de desenvolvimento, com possibilidade de seguir o mapeamento dos requisitos na arquitectura e depois na tecnologia utilizada. A Figura 24 resume a abordagem proposta.

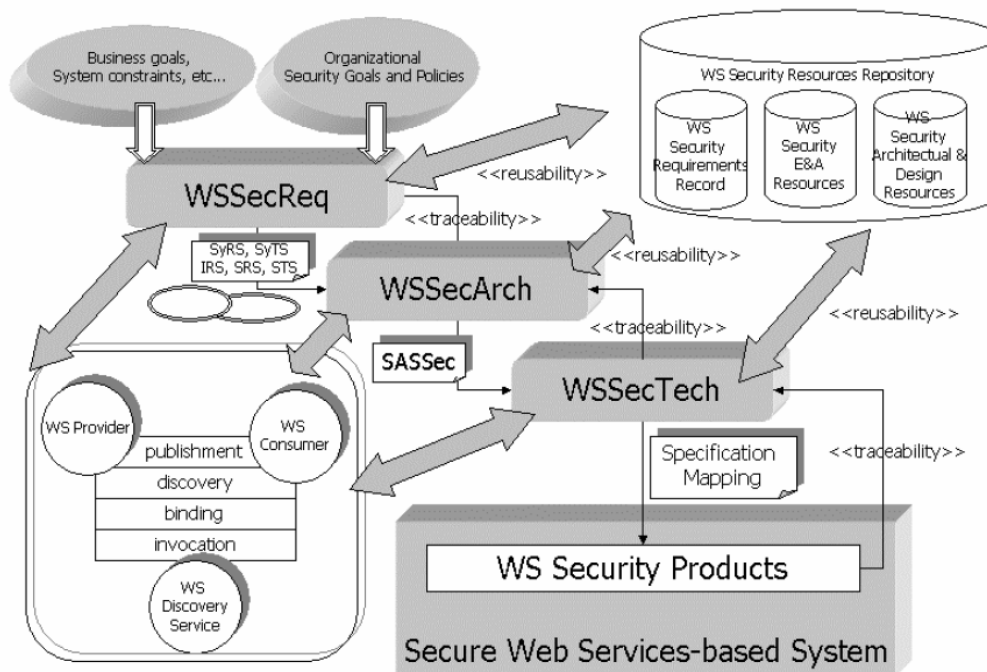


Figura 24 – Processo de desenvolvimento para serviços seguros [Gutierrez05].

A delegação em serviços, ou seja, a noção que a identidade do agente cliente deve ser usada para autorização e mantida no contexto de execução de serviços no âmbito de um processo de negócio, é um problema reconhecido para o qual estão a ser efectuadas várias propostas. Bengtsson [Bengtsson05] propõe que a orquestração segura de serviços seja baseada no registo, acompanhamento e autenticação das identidades assumidas pelos serviços ao longo da sua execução. Mello [Mello05] propõe uma abordagem de federação entre domínios para a troca de informação de segurança para autenticação e

autorização. Vecchio [Vecchio05] propõe um sistema de gestão de credenciais para serviços centradas no utilizador, tendo realizado uma implementação multi-plataforma com WS-Security e WS-Trust que permite a emissão e troca de credenciais, sendo a delegação apontada como trabalho futuro relevante. Wang [Wang05] propõe uma extensão à SAML, definindo asserções de delegação, cuja segurança é baseada em assinaturas XML com certificados X.509.

Finalmente, no que respeita à validação de configurações de segurança, Bhargavan [Bhargavan05] apresenta o verificador de configuração – ‘Policy advisor’ – disponibilizado no “WSE 3 / Dot Net 2”, discutindo o modelo formal da ferramenta e de outras similares.

4.7. Objectivos de avaliação

O modelo, normas e implementações apresentados constituem uma proposta concreta para dar resposta aos problemas de segurança dos serviços. O principal objecto da avaliação dos serviços seguros a apresentar nos capítulos seguintes são as *normas* na forma como especificam os seguintes mecanismos:

- Autenticação (AUTN);
- Autorização (AUR);
- Protecção das mensagens (PROT);
- Configuração (CONF).

Para além das normas, vão também avaliar-se as *implementações* e respectivas ferramentas de desenvolvimento. A Tabela 3 relaciona as normas com os mecanismos de segurança. Apenas são referidas as normas suportadas nas implementações disponíveis.

Tabela 3 – Relação entre mecanismos e normas de serviço seguros.

Tecnologia	Autenticação (AUTN)	Autorização (AUR)	Protecção das mensagens (PROT)	Configuração (CONF)
WS-Security	X		X	
SAML	X	X		
WS-SecurityPolicy				X

4.8. Resumo

Neste capítulo enumeraram-se os problemas a resolver na segurança de serviços e descreveram-se em detalhe as normas e implementações disponíveis. A segurança de serviços incide sobre a protecção das mensagens, o controlo de acessos e a configuração automática. Foi analisado o trabalho científico relacionado e concluiu-se com a identificação de objectivos de avaliação dos serviços seguros.

O tema do próximo capítulo é o caso de estudo.

5. Caso de estudo

Este capítulo descreve o caso de estudo “compra e venda de imóvel”. São apresentadas a motivação inicial para o caso, a descrição do problema, o desenho da solução para cenários exemplificativos e a implementação do protótipo de um dos cenários.

5.1. Motivação

O caso de estudo tem como objectivo efectuar uma *avaliação do desenvolvimento de serviços*, como sistemas de informação empresariais flexíveis, e da *protecção de serviços*, nos mecanismos de autenticação, autorização, protecção de mensagens e configuração especificados nas normas. A *metodologia* adoptada foi a realização de um protótipo com ensaios prévios das implementações disponíveis.

Os *critérios* definidos para a *escolha do caso* foram:

- *Ter valor* – para evidenciar as necessidades de segurança de forma inequívoca;
- *Ser real* – para que os requisitos fossem ditados por necessidades efectivas do negócio de forma não enviesada pela tecnologia;
- *Ser familiar* – para que o contexto aplicacional fosse facilmente perceptível e para que as conclusões pudessem ser transpostas para outros domínios.

Tendo em conta os critérios enunciados, o caso escolhido foi a “compra e venda de imóvel”. O valor a proteger está na informação de registo dos imóveis e nos montantes que estão a ser negociados. A realidade do caso reflecte-se no levantamento de requisitos a partir do processo de negócio. A familiaridade deve-se ao facto de muitas pessoas conhecerem ou já terem participado em processos para aquisição de habitação própria.

No futuro as transacções de compra e venda de imóveis no mercado imobiliário serão suportadas de forma mais eficaz com meios electrónicos. Para que a confiança das partes interessadas seja preservada será necessária uma correcta implementação e gestão dos mecanismos de segurança para protecção dos valores envolvidos em todo o processo. O caso é rico em situações que permitem explorar as potencialidades da tecnologia de serviços seguros.

5.2. Descrição

A descrição do caso inclui o contexto organizacional, o processo de negócio, o sistema de informação estruturado em serviços para um conjunto de cenários exemplificativos e a especificação completa do cenário do protótipo.

5.2.1. Contexto organizacional

O contexto organizacional da compra e venda de imóveis é descrito com o modelo da cadeia de valor e com o modelo das forças competitivas.

Cadeia de valor

A *cadeia de valor* [Laudon02] é formada pelas actividades que produzem produtos ou serviços e que são realizadas por diferentes organizações. A cadeia de valor para o imobiliário é apresentada na Figura 25.

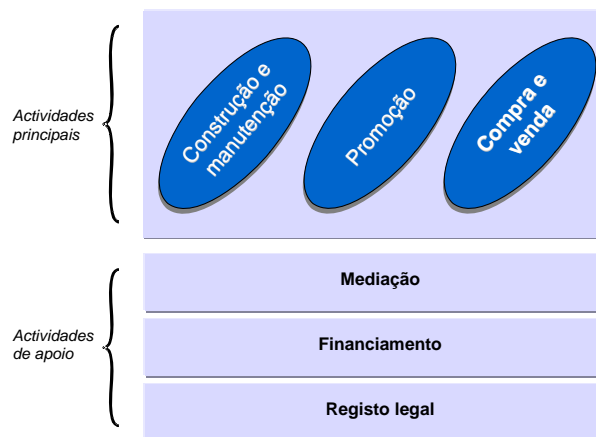


Figura 25 – Cadeia de valor do mercado imobiliário.

As *actividades principais* da cadeia – construção e manutenção, promoção, compra e venda – acrescentam valor directamente. As *actividades de apoio* – mediação, financiamento e registo legal – tornam as actividades principais possíveis ou facilitam-nas, acrescentando valor indirectamente.

Forças competitivas

O modelo das *forças competitivas*, proposto por Porter [Porter04], permite descrever a influência de forças externas na estratégia e competitividade de uma indústria. A Figura 26 ilustra o modelo aplicado às organizações que actuam no mercado imobiliário nacional.

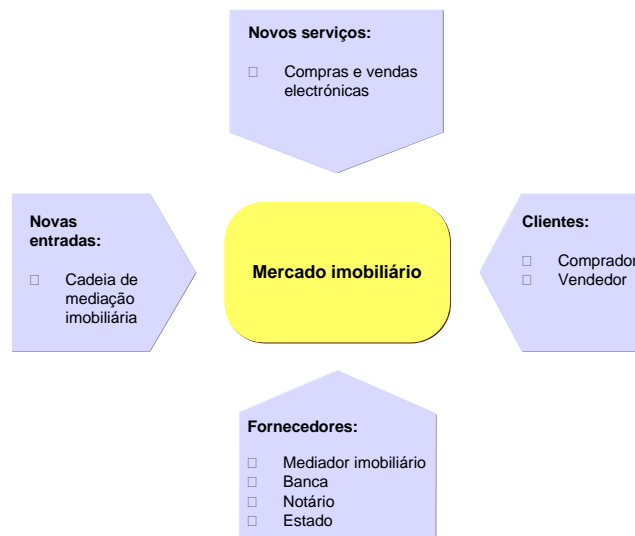


Figura 26 – Forças competitivas do mercado imobiliário português.

Consideraram-se como *clientes* o comprador e o vendedor do imóvel, pois são eles que justificam a existência do mercado. Os *fornecedores* são o mediador imobiliário, a banca, o notário e o estado. O mediador dá apoio ao vendedor e ao comprador. A banca financia, em muitos casos, a operação. O notário efectua as escrituras públicas. Os organismos do estado como: a conservatória de registo predial, a câmara municipal, as finanças e os tribunais, prestam informação oficial e permitem que a venda seja legalmente aceite. Uma *nova entrada* no mercado são as cadeias de mediação imobiliária, que agregam mais informação e que executam as actividades de apoio de forma mais moderna e optimizada. A sua entrada criou uma força para *novos serviços*, no sentido de agilizar os processos de compra e venda.

5.2.2. Processo de negócio

O processo de negócio da compra e venda de imóveis é descrito pelos actores envolvidos, pelos sub-processos e pelas entidades informacionais. A abordagem seguida é adaptada de Spewak [Spewak93], tratando-se de uma descrição de alto nível que, propositadamente, omite muitos detalhes para se focar no que é essencial. Os resultados são apresentados resumidamente nas subsecções seguintes e detalhadamente no Anexo D.

Actores

Foram identificados os seguintes *actores* relevantes para o processo de compra e venda de imóvel: Vendedor, Comprador, Mediador Imobiliário, Notário, Conservatória do Registo Predial, Finanças, Município, Tribunal, Banco e Seguradora.

Processos

Foram identificados os *processos de negócio* (PN) para o processo de compra e venda de imóvel apresentados na Figura 27.

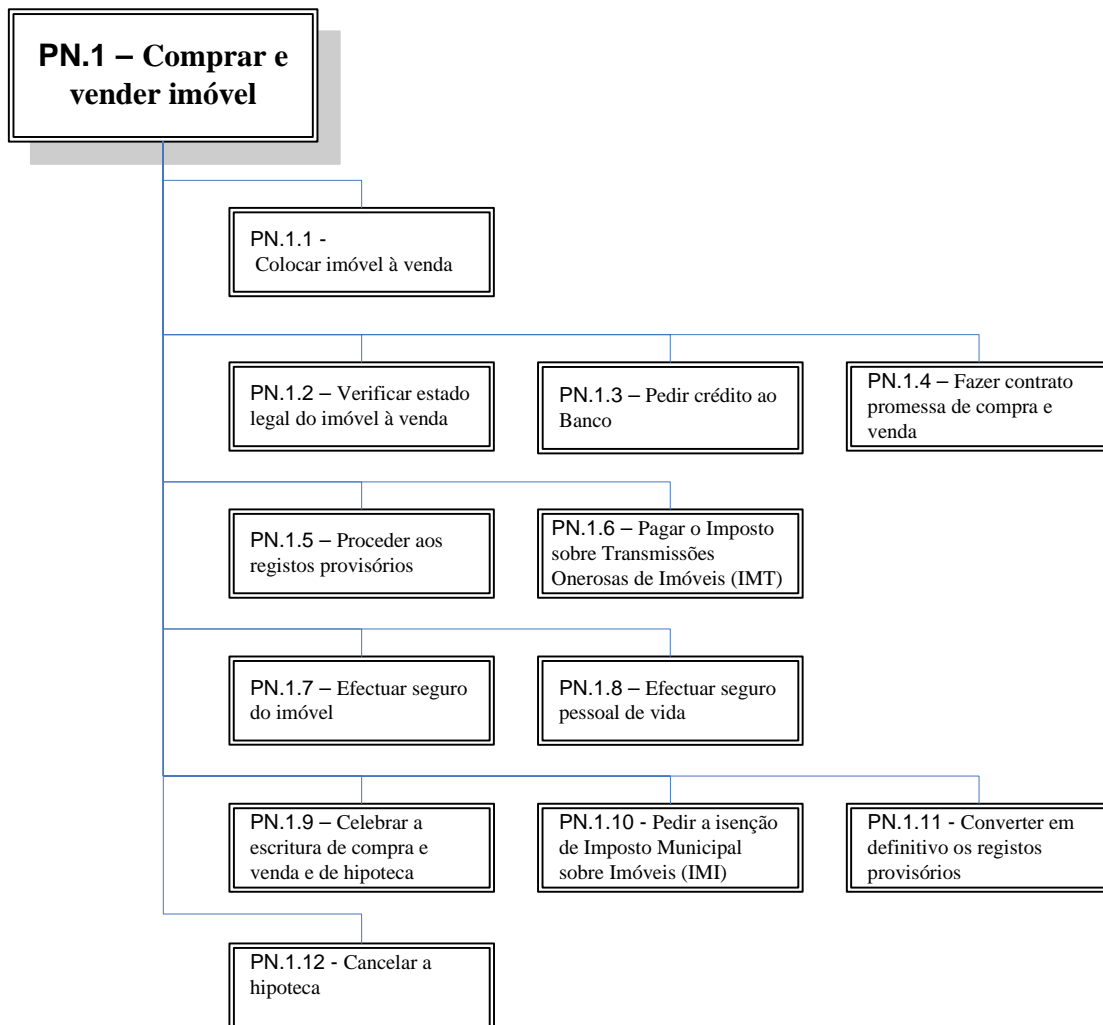


Figura 27 – Resumo do processo de negócio da compra e venda de imóvel.

Entidades informacionais

Foram identificadas as *entidades informacionais* (EI) usadas no processo de compra e venda de imóvel apresentadas na Figura 28.

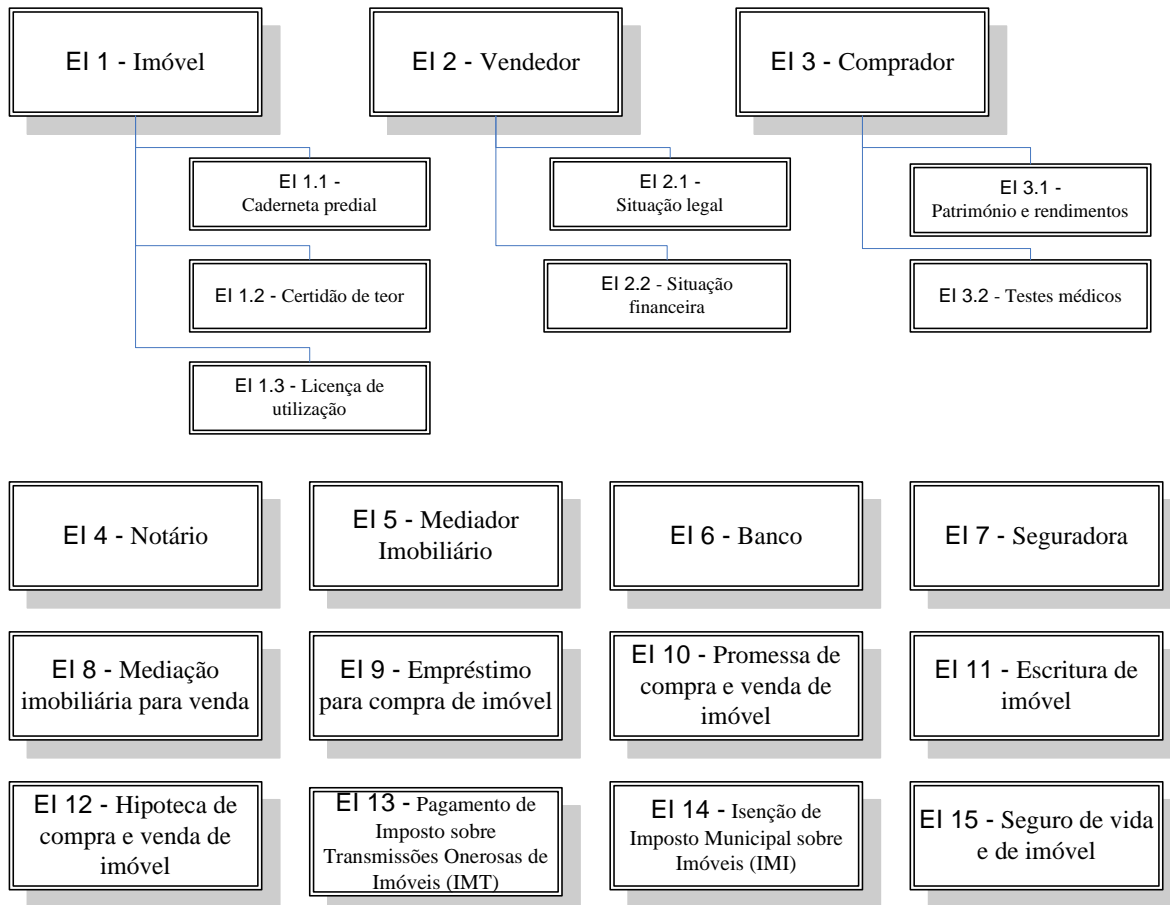


Figura 28 – Resumo das entidades informacionais da compra e venda de imóvel.

5.2.3. Cenários

Os cenários são excertos do processo de negócio de compra e venda de imóvel que exemplificam as interações entre os vários actores. A Figura 29 apresenta os cenários e os actores envolvidos.

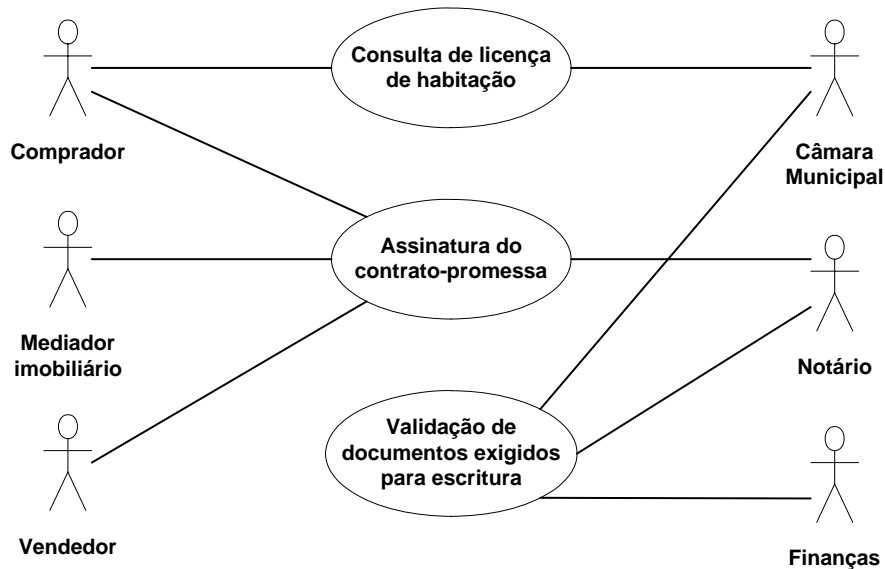


Figura 29 – Cenários exemplificativos da compra e venda de imóvel.

Cada cenário corresponde a um caso de uso UML [Fowler99], com:

- Objectivo – o fim que se pretende atingir no cenário;
- Actores – entidades envolvidas;
- Pré-condições – condições que se assumem como verdadeiras à partida;
- Cenário principal – descrição do cenário principal;
- Extensões – descrição de variantes do cenário principal.

Cenário da “Consulta de licença de habitação”

<u>Objectivo</u>	O Comprador contacta o Município para consultar a licença de habitação do imóvel.
<u>Actores</u>	Comprador, Município
<u>Pré-condições</u>	Os serviços electrónicos do Município são acedidos através de uma Associação de Municípios.
<u>Cenário principal</u>	O Comprador contacta a Associação de Municípios. A Associação encaminha o pedido para o Município competente, acessível dentro da rede da Associação. O pedido é devidamente credenciado. O Município verifica a credencial, processa o pedido e responde à Associação, que encaminha a resposta ao Comprador.

<u>Extensões</u>	A autorização pode ser concedida num domínio de segurança externo à Associação de Municípios.
------------------	---

O cenário principal de “Consulta de licença de habitação” permite avaliar:

- A autorização dos acessos (AUTR);
- A utilização de políticas para suportar, por configuração (CONF):
 - O atravessamento de uma fronteira organizacional (o pedido externo é validado à entrada e convertido para um pedido interno);
 - A adaptação tendo em conta os dados de instância do pedido.

A extensão permite avaliar:

- A portabilidade de credenciais de autorização (AUTR).

Cenário da “Assinatura de contrato-promessa de compra e venda”

<u>Objectivo</u>	O Vendedor e o Comprador querem assinar um contrato-promessa de compra e venda com reconhecimento por Notário.
<u>Actores</u>	Vendedor, Comprador, Notário, Mediador Imobiliário
<u>Pré-condições</u>	O Vendedor e o Comprador têm uma identidade digital certificada e são capazes de assinar documentos digitalmente. O Notário é capaz de reconhecer as assinaturas do Vendedor e do Comprador de domínios de segurança diferentes.
<u>Cenário principal</u>	O Vendedor e Comprador negociam o contrato. O Vendedor submete o contrato-promessa e a sua assinatura do mesmo. O Comprador submete o contrato-promessa e a sua assinatura do mesmo. O Notário compara os contratos submetidos e confirma que são iguais. O Notário valida a assinatura do Vendedor e a assinatura do Comprador. O Notário assina o contrato e devolve uma cópia ao Vendedor e ao Comprador.
<u>Extensões</u>	O contrato pode ser mantido confidencial em trânsito. A protecção pode abranger apenas parte do contrato. A assinatura do Vendedor pode ser apresentada por um Mediador Imobiliário, que receba delegação para o efeito.

O cenário principal de “Assinatura de contrato-promessa de compra e venda” permite avaliar:

- A autenticação (AUTN) e integridade (PROT);
- A portabilidade de credenciais de autenticação (AUTN);
- A negociação dinâmica de uma política aceitável por todas as partes (CONF).

As extensões permitem avaliar:

- A confidencialidade (PROT);
- A granularidade da protecção de confidencialidade e integridade (PROT);
- A delegação como forma especial de autorização (AUTR).

Cenário da “Validação de documentos exigidos para escritura”

<u>Objectivo</u>	O Notário verifica a autenticidade de alguns dos documentos necessários para a celebração de uma escritura de compra e venda de imóvel.
<u>Actores</u>	Notário, Finanças, Município
<u>Pré-condições</u>	Os documentos a validar são a Caderneta Predial das Finanças e a Licença de Habitação do Município, que são emitidos por entidades em domínios de confiança diferentes.
<u>Cenário principal</u>	O Notário analisa as credenciais associadas ao documento para as validar. No decorrer do processo, se necessário, poderá contactar o emissor de cada credencial para obter mais informação relevante para a validação.

O cenário de “Validação de documentos exigidos para escritura” permite avaliar:

- A autenticação (AUTN) e integridade dos documentos (PROT);
- A portabilidade das credenciais de autenticação do documento, que atestam a sua origem (AUTN);
- A expressividade das políticas para definir a configuração (CONF).

Resumo dos cenários

A Tabela 4 resume os cenários e o que cada um deles permite avaliar relativamente à segurança.

Tabela 4 – Relação entre os cenários e a avaliação da segurança.

Cenário	Autenticação (AUTN)	Autorização (AUTR)	Protecção das mensagens (PROT)	Configuração (CONF)
Consulta de licença de habitação		X		X
Assinatura do contrato-promessa de compra e venda	X	X	X	X
Validação de documentos exigidos para escritura	X			X

5.2.4. Sistema de informação estruturado em serviços

O sistema de informação é descrito pelos serviços que o constituem e pela forma como são combinados para realizar partes do processo de negócio.

De seguida apresenta-se o procedimento usado para especificar os serviços nos cenários exemplificativos do processo de negócio. A metodologia foi estendida a partir da proposta de Guerra e Pardal [Guerra04], baseada na arquitectura empresarial de sistemas de informação.

Procedimento de especificação dos serviços

A especificação dos serviços tem duas fases. A primeira fase consiste na *definição abstracta dos serviços*, tendo apenas em conta as necessidades do processo de negócio. A segunda fase consiste na *definição concreta dos serviços*, ajustando o modelo proposto à realidade dos recursos disponíveis para a implementação.

A primeira fase da especificação de serviços inicia-se a partir da descrição do processo de negócio com actores, processos e entidades informacionais:

- Identificar o actor responsável pela criação e actualização de cada entidade informacional;
- Definir uma vista de cada entidade informacional no contexto do processo de negócio, reutilizando ou produzindo esquemas de dados (XML Schemas);
- Decompor o processo em partes, e para cada parte detalhar e identificar as acções de negócio que se realizam, indicando: emissor, destinatário, acção e dados trocados;
- Definir interfaces do serviço (WSDL) de cada actor, tirando partido dos esquemas de dados definidos (XML Schemas) na especificação das entradas, saídas e excepções;
- Definir requisitos não funcionais do serviço (WS-Policy) de cada actor.

A segunda fase da especificação dos serviços usa os esquemas de dados, as interfaces e as políticas definidas:

- Identificar a realidade dos sistemas de informação envolvidos: ambientes de execução, bibliotecas, aplicações, ficheiros, bases de dados, distribuição física e as interfaces disponíveis para integração aos diferentes níveis: utilizador, programação e dados;
- Seleccionar a plataforma de serviços a usar para implementar os serviços;
- Disponibilizar a meta-informação – esquemas, interfaces e políticas – localmente nos serviços (auto-descrição) ou através de um directório de serviços;

- Implementar a funcionalidade do serviço, tirando partido das interfaces de integração disponíveis, especificando:
 - Componentes – para implementar as funcionalidades dos serviços, usando aplicações existentes e novos desenvolvimentos;
 - Fontes de dados – para construir as vistas das entidades informacionais, a disponibilizar nos serviços;
 - Transportes para as mensagens – para suportar a comunicação;
- Implementar os aspectos não funcionais do serviço (ex. segurança), usando as capacidades disponíveis na plataforma de serviços seleccionada.

Serviços dos cenários

Para cada cenário exemplificativo foram identificados os serviços necessários à sua implementação. Nas figuras seguintes são apresentados os diagramas UML de colaboração e as definições abstractas das interfaces com entradas, saídas e excepções.

A Figura 30 descreve os serviços da “Consulta de licença de habitação”. O Comprador pede para consultar a licença de habitação. Existem serviços para a Associação de Municípios e para o Município.

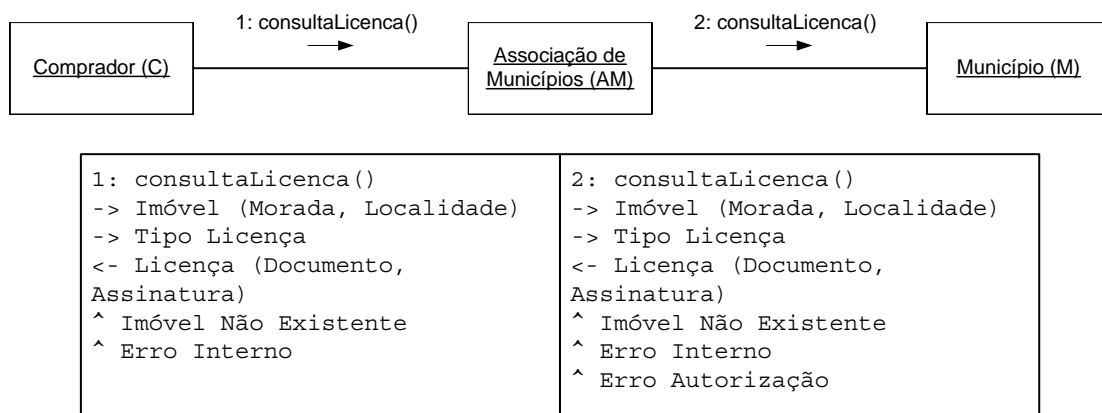
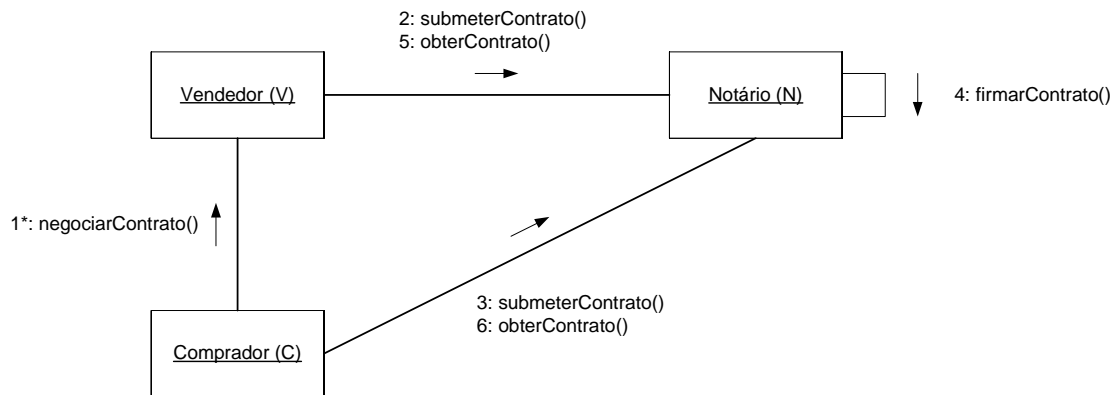


Figura 30 – Serviços da “consulta de licença de habitação”.

A Figura 31 apresenta os serviços da “Assinatura de contrato-promessa de compra e venda”. O Vendedor e Comprador negociam o contrato, que depois submetem, assinado, no Notário. Existem serviços para Vendedor e Notário.



<pre> 1*: negociarContrato() -> Proposta Contrato <- Aceitação Proposta <- ID Contrato <- Contra-proposta ^ Erro Interno </pre>	<pre> 2,3: submeterContrato() -> ID Contrato -> Contrato -> Assinatura ^ ContratoNaoAceite ^ Erro Interno </pre>
<pre> 4: firmarContrato() -> ID Contrato -> Contrato -> Assinaturas[] <- Contrato <- Assinatura ^ Erro Interno </pre>	<pre> 5,6: obterContrato() -> ID Contrato <- Contrato <- Assinatura ^ ContratoAindaNaoFirmado ^ ErroFirmarContrato ^ Erro Interno </pre>

Figura 31 – Serviços da “assinatura de contrato-promessa de compra e venda”.

A Figura 32 descreve os serviços da “Validação de documentos exigidos para escritura”. O Comprador submete documentos cuja autenticidade é verificada pelo Notário, contactando as Finanças e o Município, se necessário. Existem serviços para o Notário, para as Finanças e para o Município.

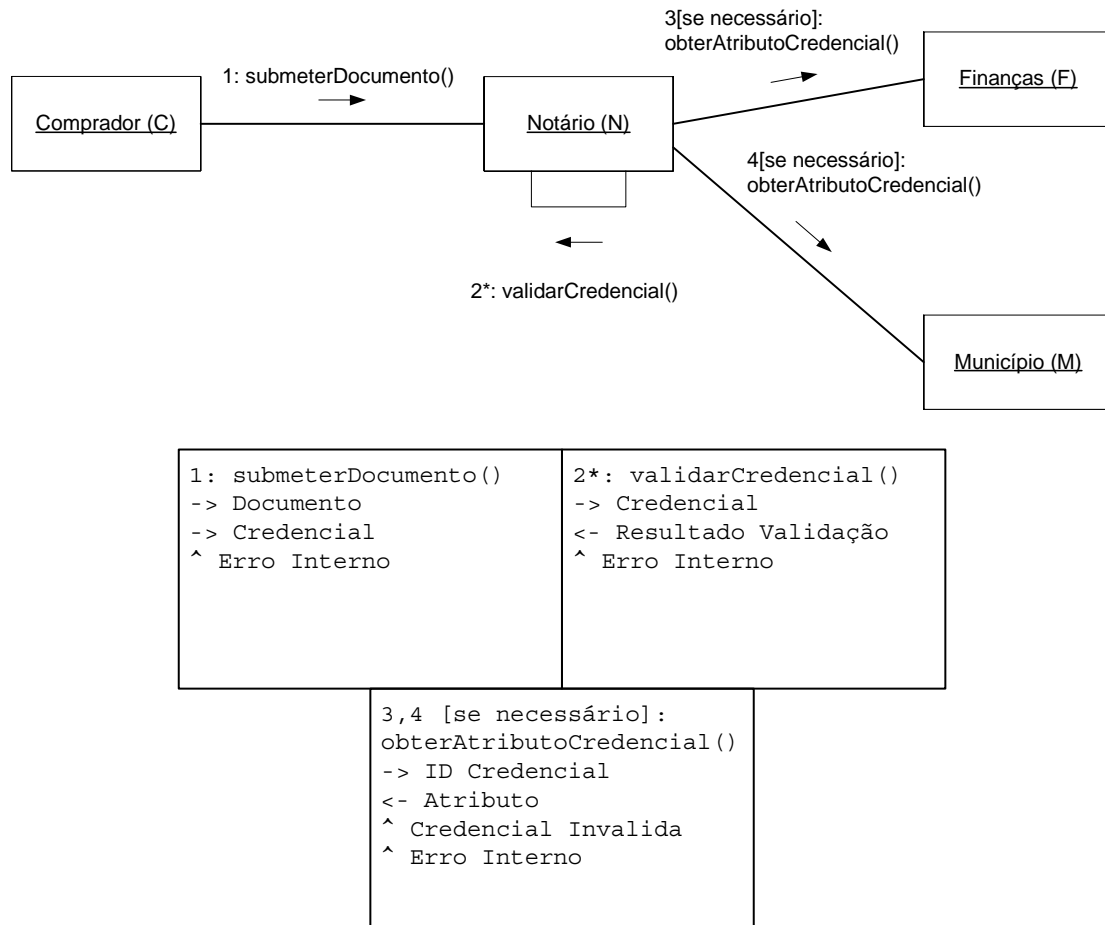


Figura 32 – Serviços da “validação de documentos exigidos para escritura”.

5.2.5. Protótipo

O cenário escolhido para o protótipo foi a “assinatura de contrato-promessa de compra e venda”, por ser o mais interessante e abrangente do ponto de vista da avaliação, como se pode observar na Tabela 4.

No protótipo, cada actor é representado por um serviço. Os serviços existentes são: o Comprador (C), o Vendedor (V), o Notário (N), as Finanças (F) e o Registo Civil (RC).

V e C negociam o contrato de compra e venda entre si, que depois submetem, assinado individualmente, a N. N verifica os contratos submetidos, autenticando V através de RC e C através de F.

Cada serviço define uma extremidade para *vinculação*, que permite obter meta-informação sobre o serviço, e uma extremidade para *invocação*, que permite executar o serviço. A vinculação é representada na Figura 33 e a invocação na Figura 34.

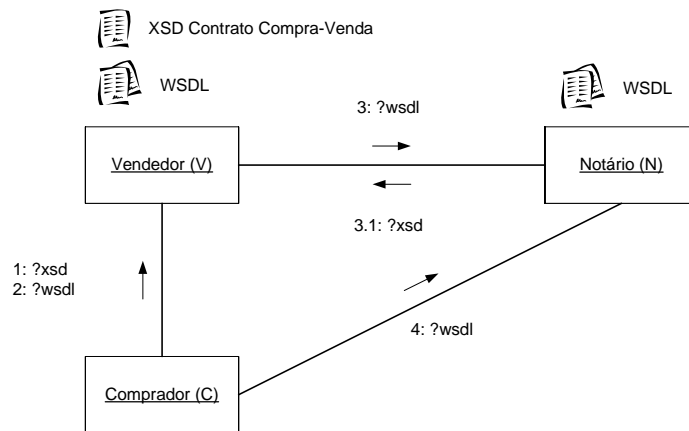


Figura 33 – Diagrama de colaboração entre serviços no cenário “assinatura de contrato de compra e venda” durante a vinculação.

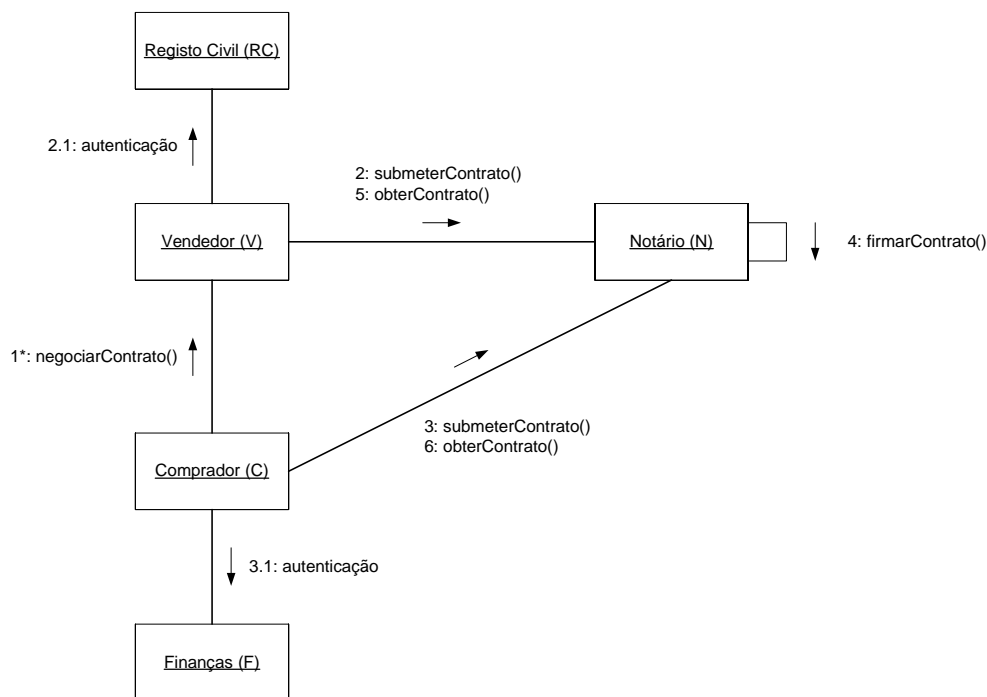


Figura 34 – Diagrama de colaboração entre serviços no cenário “assinatura de contrato de compra e venda” durante a invocação.

A execução do protótipo, com vinculação e invocação dinâmicas, seria a seguinte, se não existissem limitações na tecnologia:

- Vinculação entre C e V:
 - V é disponibilizado;

- C localiza V;
- C estabelece confiança em V;
- V disponibiliza o esquema do contrato de compra e venda (XSD);
- V disponibiliza a interface funcional (WSDL) que inclui o esquema das mensagens;
- C gera adaptadores para invocação do serviço;
- V disponibiliza política (WS-Policy);
- C configura processadores de mensagens para satisfazerem requisitos de segurança.
- Invocação de V por C para negociar contrato:
 - C propõe contrato a V;
 - V aceita ou contra-propõe;
 - C e V geram identificador para contrato.
- Vinculação entre C e N:
 - N é disponibilizado;
 - C localiza N;
 - C estabelece confiança em N;
 - V propõe esquema do contrato de compra e venda (XSD);
 - N estabelece confiança em V;
 - N disponibiliza a interface funcional (WSDL) que inclui o esquema das mensagens;
 - C gera adaptadores para invocação;
 - N disponibiliza política (WS-Policy);
 - C configura processadores de mensagens para satisfazerem requisitos de segurança.
- Vinculação entre V e N (idêntica à anterior, substituindo C por V).
- Invocação de N por C para submeter contrato:
 - C submete cópia do contrato.
- Invocação de N por V para submeter contrato (idêntica à anterior, substituindo C por V).
- Validação do contrato por N:
 - N usa esquema do contrato de compra e venda (XSD) proposto por V;

- N instancia validador adequado ao tipo de contrato;
- N valida cópia do contrato submetida por C;
- N valida cópia do contrato submetida por V;
- N firma contrato.
- Invocação de N por C para obter contrato firmado:
 - C pede contrato fornecendo o identificador;
 - N devolve contrato.
- Invocação de N por V para obter contrato firmado (idêntica à anterior, substituindo C por V).

O estabelecimento de confiança entre serviços é efectuado com base na distribuição de chaves, simétricas e assimétricas, apresentada na Figura 35.

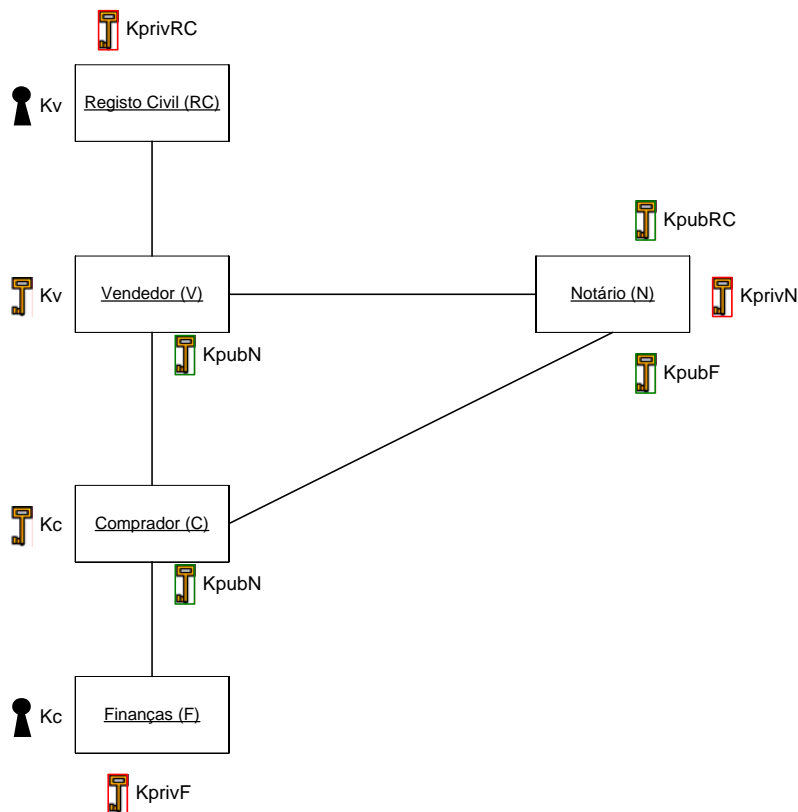


Figura 35 – Chaves do cenário “assinatura de contrato de compra e venda”.

C tem uma chave secreta, que F consegue verificar. V tem uma chave secreta, que RC consegue verificar. N tem um par de chaves, cuja chave pública foi distribuída a C e V de forma segura. F tem um par de chaves, cuja chave pública foi distribuída a N de forma segura. RC tem um par de chaves, cuja chave pública foi distribuída a N de forma segura.

Em termos de relações de confiança, V e C confiam em N para firmar o contrato e N confia em RC e F para identificar e autenticar V e C, respectivamente.

5.3. Implementação

A metodologia de avaliação adoptada foi a implementação de um protótipo. O plano inicial era escolher a implementação disponível com suporte para todas as normas, no entanto, verificou-se que nenhuma satisfazia esta condição. Foram necessários ensaios prévios para testar quais os mecanismos de segurança efectivamente disponíveis. Durante esta fase foram registados diversos apontamentos técnicos sobre as capacidades e limitações de cada implementação, que fundamentaram depois a escolha de uma delas para a realização do protótipo.

5.3.1. Ensaios

Para testar a tecnologia de serviços seguros foram realizados *ensaios* nas implementações:

- “WSE 3 sobre Dot Net 2”;
- “WSS4J sobre Axis2 sobre Java”;
- “XWSS sobre JAX-WS 2 sobre Java”.

Os ensaios foram realizados primeiro no “WSE 3 / Dot Net 2”, depois no “WSS4J / Axis2 / Java” e por último no “XWSS / JAX-WS 2 / Java”.

WSE 3 / Dot Net 2

Os ensaios realizados no “WSE 3 / Dot Net 2” foram os seguintes:

- Utilização da biblioteca System.Security.Cryptography para cifra e decifra de dados;
- Cliente e servidor Web Services simples, tipo HelloWorld;
- Cliente e servidor Web Service assíncronos;
- Cliente e servidor Web Service com invocação de método unidireccional;
- Cliente de Web Service Java;
- Utilização do registo de mensagens SOAP;
- Utilização de MTOM para otimizar a transmissão de mensagens com dados binários;
- Utilização de configuração declarativa de segurança WS-Security:
 - Autenticação de servidor com certificado digital X.509;

- Autenticação de cliente com nome e senha;
- Autenticação de cliente com certificado digital X.509;
- Utilização de WS-SecureConversation para que o certificado do cliente só fosse enviado na primeira mensagem da sessão;
- Autorização com base em conta de utilizador Windows;
- Selecção programática da configuração WS-Security a usar;
- Desenvolvimento de elemento de configuração à medida;
- Instalação do pacote de extensão SAML Security Token Service QuickStart.

Os ensaios concluíram que não eram suportadas as seguintes funcionalidades:

- WS-Policy, WS-SecurityPolicy;
- Autorização independente do sistema operativo;
- Asserções SAML.

As referências utilizadas nestes ensaios foram: [MacDonald03] e [Microsoft05c].

WSS4J / Axis2 / Java

Os ensaios realizados no “WSS4J / Axis2 / Java” foram os seguintes:

- Programas de manipulação XML com a biblioteca AXIOM 1.0:
 - Criação de novos documentos;
 - Leitura e validação de documentos existentes;
- Desenvolvimento de serviços com a biblioteca Axis2 1.0:
 - Com manipulação directa de XML;
 - Com classes Java vinculadas aos dados;
- Desenvolvimento de clientes com a biblioteca Axis2 1.0:
 - Com manipulação directa de XML;
 - Com classes de invocação geradas por ferramenta a partir da definição WSDL;
- Invocações síncronas e assíncronas de serviços;
- Operações com políticas WS-Policy usando a biblioteca Commons Policy 1.0:
 - Normalização;

- Junção;
- Intersecção.
- Instalação do módulo de segurança WSS4J.

Os ensaios concluíram que o Axis2 é, actualmente, muito instável no desenvolvimento de serviços simples, mesmo sem segurança. Por exemplo, não é possível desenvolver serviços que usem vectores ou excepções.

As referências utilizadas nestes ensaios foram: [Apache06] e [Samaranayake06].

XWSS / JAX-WS 2 / Java

Os ensaios realizados no “XWSS / JAX-WS 2 / Java” foram os seguintes:

- Vinculação de dados Java com XML através da biblioteca JAX-B 2:
 - Geração de código Java a partir de XML Schema, para leitura e escrita de documentos;
 - Utilização do elemento de extensão Any do XML Schema;
- Assinatura digital XML com a biblioteca Java XML Digital Signature API (xmldsig), nas variantes:
 - ‘Detached’, que assina dados que são externos à assinatura;
 - ‘Enveloping’, que assina dados contidos na estrutura da assinatura;
 - ‘Enveloped’, onde a assinatura está contida na estrutura que está a ser assinada;
- Desenvolvimento de clientes e serviços JAX-WS 2:
 - A partir de WSDL;
 - A partir de classes Java;
 - Em ambas as situações anteriores, utilização de JAX-WS Handlers para interceptar as mensagens SOAP;
- Desenvolvimento de clientes e serviços JAX-WS 2 com segurança XWSS, nas seguintes configurações:
 - Autenticação e autorização com utilizador e senha;
 - Colocação e verificação de marcas temporais;
 - Assinatura de mensagens com chave privada e verificação com certificado digital de chave pública X.509;

- Cifra de mensagens com chave privada e decifra com certificado digital de chave pública X.509;
- Cifra e decifra de mensagem com chave simétrica partilhada;
- Configuração de segurança diferenciada por operação do serviço (cada operação do serviço tem uma configuração de segurança própria);
- Utilização e verificação de asserções SAML de autenticação;
- Combinações das configurações anteriores;
- Invocação de serviço com segurança no transporte usando HTTPS com certificado digital do servidor Tomcat:
 - Autenticação do cliente com utilizador e senha;
 - Autenticação do cliente com certificado digital.

Os ensaios concluíram que não eram suportadas as seguintes funcionalidades:

- WS-Policy, WS-SecurityPolicy.

A referência utilizada nestes ensaios foi: [Sun06].

antInclude: extensões para a ferramenta de construção

Para os ensaios sobre Java, foram desenvolvidas extensões para a ferramenta de construção Ant.

O antInclude [Pardal04] é uma biblioteca de extensões para a ferramenta Apache Ant [Apache05]. O seu objectivo é normalizar os processos de construção – configuração, compilação, instalação, etc. – de projectos, permitindo reutilização e simplificação.

O antInclude foi originalmente desenvolvido para apoiar projectos das disciplinas de “Sistemas Distribuídos”, “Engenharia de Software” e “Integração de Sistemas de Informação” do Instituto Superior Técnico, estando disponível gratuitamente sob licença GNU LGPL.

No decorrer dos ensaios, foi desenvolvida uma nova versão do antInclude, com suporte para Axis2 e para JAX-WS 2 com XWSS.

5.3.2. Protótipo

Após a realização dos ensaios, a tecnologia escolhida para concretizar o protótipo foi a “XWSS / JAX-WS 2 / Java” em conjunto com a biblioteca de processamento de políticas proveniente de “Axis2 / Java”. Esta combinação foi utilizada por cobrir o maior número de normas de segurança para serviços. O “WSE 3 / Dot Net” não suporta WS-Policy cuja avaliação era fundamental. Uma vantagem adicional

do “XWSS / JAX-WS 2 / Java” é ter o código fonte disponível, o que permitiu análises mais aprofundadas, sempre que necessário.

O ambiente de desenvolvimento utilizado incluiu as seguintes ferramentas: Sun Microsystems Java Developer Kit 1.5.0_06, Apache Ant 1.6.5, Eclipse Web Tools Platform 1.0 e Altova XMLSpy 2006.

O ambiente de execução foi composto por: Sun Microsystems Java Runtime Environment 1.5.0_06, Sun Microsystems Java Web Services Developer Pack 2.0 (actualizado com as versões finais de JAX-WS 2, JAX-B 2 e XWSS), Apache Tomcat 5, Sun Microsystems UDDI Registry e Apache Commons Policy 1.0.

O protótipo foi desenvolvido por etapas que são descritas sucintamente de seguida.

1ª etapa – serviços de negócio sem segurança

A primeira etapa consistiu na implementação das funcionalidades de negócio sem preocupações de segurança. Foram realizadas as seguintes tarefas:

- Definição de esquema XSD para o contrato promessa de compra e venda e contratos exemplo para teste;
- Definição das interfaces WSDL com esquemas XSD para as mensagens, directamente em XML. O esquema do contrato promessa foi reutilizado por cópia;
- Implementação de esqueleto das aplicações a partir de WSDL: ServicoVendedor, TesteServicoVendedor, ServicoNotario, TesteServicoNotario e Comprador;
- Implementação das funcionalidades: negociação de contrato de compra e venda, submissão ao notário e obtenção de contrato firmado.

Para apoio à edição e validação de XSD e WSDL foi usada a ferramenta XMLSpy.

2ª etapa – gestão da meta-informação

Durante o desenvolvimento do código de negócio foram identificados vários aspectos a melhorar, que implicavam uma gestão autónoma da meta-informação dos serviços, com uma abordagem semelhante à proposta na norma WS-MetadataExchange. Foram efectuadas as seguintes tarefas:

- Definição de nomes de rede lógicos para os vários actores: *financas.gov*, *registocivil.gov*, *vendedor.org*, *comprador.org* e *notario.org*;
- Implementação de aplicações Web para disponibilização de XSD e WSDL: *VendedorMetaInfo* e *NotarioMetaInfo*;
- Actualização dos serviços para uso da meta-informação por referência;

- Utilização partilhada do esquema do contrato promessa, o que permitiu a partilha de tipos de dados entre os serviços, sem a necessidade de conversões;
- Actualização do esquema do contrato promessa para permitir incorporar dados adicionais não especificados através do elemento de extensibilidade `xsd:any`. Esta alteração permitiu a inclusão de uma assinatura digital XML de forma opaca para o restante código da aplicação;
- Revisão da implementação dos serviços para reforçar a separação entre o código de interface e o código com a lógica da aplicação.

3ª etapa – distribuição de chaves de segurança

Antes de iniciar a implementação da segurança foi necessário gerar e distribuir chaves:

- Definição de utilizadores e senhas para Comprador e ServicoVendedor;
- Geração de par de chaves assimétricas RSA para ServicoNotario, Finanças e Registo Civil;
- Geração de certificados digitais de chave pública para ServicoNotario, Finanças e Registo Civil;
- Criação de repositórios de chaves e repositórios de certificados confiados para Comprador, ServicoVendedor e ServicoNotario.

Para gerar as chaves e certificados foi utilizada a ferramenta keytool do Java.

4ª etapa – autenticação directa

Nesta etapa iniciou-se a utilização dos mecanismos de segurança XWSS para proteger os serviços. Foi efectuada a autenticação directa de Comprador e ServicoVendedor pelo ServicoNotario:

- Envio e validação de utilizador e resumo da senha;
- Protecção do resumo da senha com cifra usando a chave pública do servidor;
- Envio e validação de marca temporal.

5ª etapa – autenticação com intermediário

O objectivo nesta etapa foi substituir a autenticação directa com utilizador e senha por autenticação SAML, com asserções emitidas por serviços externos. Foi conseguido o envio e validação da asserção SAML de autenticação preenchida com os dados necessários. No entanto, a implementação SAML do XWSS não permitiu a verificação automática da assinatura da asserção.

6ª etapa – protecção da integridade da mensagem

Feita a autenticação, o objectivo seguinte foi a protecção da integridade das mensagens que transportavam o contrato. Partindo da autenticação directa, o mecanismo de segurança pretendido era a protecção da mensagem com um resumo cifrado com chave simétrica (MAC – Message Authentication Code) usando a senha partilhada. No entanto, este mecanismo não existe no XWSS, apesar de estar previsto na norma WS-Security. Como implementação alternativa, foram gerados pares de chaves assimétricas para o Comprador e Vendedor, que permitiram o envio das mensagens com assinatura e a verificação baseada em certificado digital de chave pública X.509.

7ª e última etapa – negociação da configuração de segurança

A configuração XWSS não usa WS-Policy, mas sim um formato próprio.

Para o protótipo foram usados exemplos de políticas de teste de um ‘workshop’ de interoperabilidade [Microsoft06b], tendo sido realizada a operação de intersecção de duas políticas: cliente e servidor. As políticas usadas descreviam apenas um subconjunto dos requisitos de segurança necessários. No fim faltou usar a política negociada para gerar a configuração XWSS. Para aplicar a nova configuração seria necessário reinstalar o serviço no servidor Tomcat.

5.4. Resumo

Neste capítulo foi apresentado o caso de estudo para a avaliação da tecnologia de serviços seguros. O caso da “compra e venda de imóvel” foi escolhido por ter requisitos de segurança reais e exigentes. A descrição do caso foi enquadrada no contexto organizacional e usou um modelo orientado a processos. Foram detalhados três cenários exemplificativos da complexidade do caso, dos quais foi seleccionado o mais abrangente para se realizar o protótipo.

O desenvolvimento do protótipo foi antecedido por ensaios nas principais implementações disponíveis, tendo depois sido escolhida a combinação que abrangia o maior número de normas de serviços seguros.

No capítulo seguinte é apresentada a avaliação da tecnologia de serviços seguros com base nos resultados dos ensaios e do protótipo do caso de estudo.

6. Avaliação

Este capítulo apresenta a avaliação efectuada a partir dos resultados do caso de estudo. A avaliação incide sobre o *desenvolvimento e protecção de serviços*.

6.1. Desenvolvimento de serviços

Nas primeiras etapas do protótipo foram implementados os serviços de negócio sem segurança, o que permitiu avaliar as ferramentas de desenvolvimento de serviços.

6.1.1. Ferramentas

Para a especificação dos esquemas de dados em XML Schema utilizou-se o editor Altova XMLSpy que facilitou a edição, visualização e validação dos documentos. A mesma ferramenta foi utilizada para a especificação das interfaces dos serviços em WSDL e das políticas WS-Policy.

Os contratos foram definidos com um editor de texto para permitir a manipulação directa de XML Schema, WSDL e WS-Policy e desse modo garantir o respeito pela independência de plataforma na avaliação. A definição directa dos contratos não é fácil, porque a sua estrutura tem vários níveis de indirectão, que resultam de estar optimizada para processamento automático e não para utilização humana.

As ferramentas utilizadas para o desenvolvimento dos serviços foram o JAX-B 2 e o JAX-WS 2. O JAX-B 2 permitiu o mapeamento de dados entre XML e Java. O JAX-WS 2 permitiu o desenvolvimento dos clientes e servidores de serviços. Ambas as ferramentas estão interligadas, pois o JAX-WS 2 usa o JAX-B 2 para converter os dados do serviço para mensagens XML.

A configuração do JAX-B 2 e JAX-WS 2 é baseada em anotações acrescentadas ao código e em ficheiros de configuração, o que obriga a algum esforço de manter as definições coerentes. A ferramenta de construção Ant com as extensões desenvolvidas para o antIncludes simplificou estas tarefas.

6.1.2. Vinculação dinâmica

A vinculação dinâmica (dynamic binding) é uma das promessas da tecnologia de serviços. Para permitir a sua avaliação, as fases de vinculação entre serviços foram descritas explicitamente na especificação do protótipo. A vinculação entre cliente e serviço pode decompor-se em três partes:

- A *vinculação de dados*, que usa o contrato XML Schema para mapear tipos de dados;

- A *vinculação funcional*, que usa o contrato WSDL para mapear as operações disponíveis e as formas de invocação;
- A *vinculação não funcional*, que usa o contrato WS-Policy para configurar a invocação de serviço para satisfazer requisitos adicionais.

A vinculação diz-se *estática* se for efectuada em tempo de desenvolvimento ou de instalação, e *dinâmica* se efectuada em tempo de execução.

Os ensaios e o protótipo permitiram concluir que o JAX-B 2 permite a vinculação de dados estática e dinâmica, e que o JAX-WS 2 permite a vinculação funcional estática e dinâmica. A vinculação não funcional é apenas efectuada de forma estática, e usa um formato próprio, diferente de WS-Policy.

No protótipo todas as vinculações foram efectuadas de forma estática. A vinculação dinâmica de dados e funcional foi considerada pouco interessante, porque apesar do JAX-WS 2 permitir a inspecção de extremidades de serviços, seria depois necessário um contexto de interpretação para atribuir significado às operações disponibilizadas, o que tornava o programa dependente de um utilizador humano ou com lógica de invocação demasiado complexa para o protótipo e para a maior parte das aplicações empresariais. A vinculação não funcional dinâmica seria mais interessante, porque permitiria adequar os requisitos de invocação do serviço às circunstâncias, mantendo-se a funcionalidade. Este caso é diferente da vinculação funcional, porque enquanto que o domínio funcional da aplicação é aberto a diferentes áreas de negócio, o domínio não funcional é mais fechado com um conjunto restrito de alternativas suportadas pela plataforma, que poderiam ser escolhidas de forma automática.

6.1.3. Separação de vinculação de dados e funcional

O JAX-WS 2 permite a separação da vinculação de dados (XSD) e da vinculação funcional (WSDL), com cada contrato num documento autónomo.

Durante o desenvolvimento do protótipo foi detectada uma situação que ilustra a importância da boa gestão da meta-informação dos serviços. O esquema de dados do contrato promessa de compra e venda era partilhado pelos três serviços principais: Comprador, Vendedor e Notário. Na primeira implementação dos serviços a partilha foi efectuada por cópia do XSD do contrato promessa para cada serviço, sendo cada cópia referenciada no WSDL respectivo. Ao gerar os adaptadores de invocação JAX-WS 2, eram criadas classes Java para o contrato idênticas mas com tipos diferentes e incompatíveis. Sendo assim, para passar o contrato recebido do Comprador para o Notário, era necessário efectuar a tradução de Java para XML e novamente para Java para obter o contrato na representação certa. Este procedimento ineficiente e deslegante complicava significativamente a implementação da aplicação. Na segunda implementação o problema foi resolvido tirando partido da vinculação configurável (custom binding) do JAX-WS 2, eliminando as cópias e passando o XSD do

contrato promessa a ser referenciado directamente e mapeado para o mesmo pacote Java. Esta modificação teve também a vantagem de ter obrigado a explicitar que o Vendedor era o dono do esquema do contrato promessa.

O mecanismo da vinculação configurável é muito útil, pois a partilha de esquema de dados é uma situação recorrente em aplicações de negócio. O “WSE 3 / Dot Net 2” e o “WSS4J / Axis2 / Java” não suportam este mapeamento flexível.

6.1.4. Esquemas de dados abertos

Os esquemas de dados podem ser abertos, o que significa que contêm elementos `xsd:any` que permitem aos documentos ter dados adicionais. Desta forma é possível estender esquemas sem forçar a actualização de aplicações que usem versões anteriores. Outra utilidade do elemento de extensibilidade `xsd:any` é o transporte de itens de informação opacos para a aplicação.

No protótipo o esquema do contrato foi definido com um elemento de extensibilidade que permitia acrescentar XML no fim. Desta forma foi possível acrescentar os elementos de uma assinatura digital sem afectar o funcionamento anterior da aplicação de negócio.

O JAX-B 2 trata os elementos de extensibilidade dos esquemas – `xsd:any` – apresentando-os à aplicação como documentos XML ou convertendo-os para classes Java, caso exista mapeamento para o espaço de nomes dos elementos adicionais.

6.1.5. Limitações da conversão de dados

O JAX-WS 2 e JAX-B 2 permitiram a conversão dos esquemas de dados XML para tipos de dados Java de forma muito satisfatória, abrangendo: tipos simples, tipos complexos (estruturas), vectores e excepções. No entanto, durante a realização do protótipo, foram detectadas algumas limitações inerentes ao modelo de conversão de dados.

No protótipo pretendia-se que o contrato tivesse uma assinatura digital independente da protecção aplicada às mensagens SOAP. Neste caso teria sido útil dispor da flexibilidade de usar objectos Java para a lógica de negócio ou XML para a criação e verificação de assinatura, sem ter o custo das várias conversões de XML para objectos e vice-versa.

Uma alternativa à conversão seria o embrulho (wrapping) do documento XML. Em vez de serem geradas classes Java que copiam o conteúdo do documento, como faz o JAX-B 2, seriam geradas interfaces Java para navegação no documento XML por referência. Esta alternativa não está actualmente disponível. A abordagem mais semelhante existente é a utilização de expressões XPath para navegar no documento XML.




Outro problema da conversão é implicar que o documento XML seja lido integralmente e construído em memória. Neste aspecto, o AXIOM do “WSS4J / Axis2 / Java” faz um processamento gradual do XML, que permite otimizar o desempenho no processamento de grandes mensagens, apenas com um pequeno aumento de complexidade do código da aplicação, que passa a ter que se preocupar com o estado de leitura do documento. O JAX-B 2 não segue esta abordagem.

6.2. Protecção de serviços

Os ensaios realizados e as restantes etapas de desenvolvimento do protótipo permitiram avaliar as normas e implementações de segurança.

Em primeiro lugar foram avaliadas as vantagens e desvantagens de usar segurança no transporte ou segurança na mensagem. Depois a avaliação incidiu sobre os mecanismos de autenticação, autorização, protecção de mensagens e configuração.

Nas tabelas de avaliação são usados os seguintes símbolos:

-  – “Suporta totalmente”;
-  – “Suporta parcialmente”;
-  – “Não suporta”.

6.2.1. Segurança no transporte e segurança na mensagem

Em muitos casos, o transporte seguro de mensagens pode ser suficiente para dar protecção adequada aos serviços. Esta abordagem é também designada por segurança *ponto-a-ponto* (point-to-point) porque o âmbito de protecção cobre apenas a ligação entre dois pontos na rede. O exemplo mais comum de protecção no transporte é a utilização de HTTPS.

No entanto, se as mensagens dos serviços passarem por intermediários ou exigirem confidencialidade persistente, então é necessário tornar seguras as próprias mensagens ou partes delas. Esta abordagem é também designada por segurança *extremo-a-extremo* (end-to-end) porque o âmbito de protecção vai desde o emissor original até ao receptor último, com partes que podem ser destinadas a intermediários ao longo do caminho da mensagem. A tecnologia para protecção de mensagens SOAP é a WS-Security.

O protótipo do caso de estudo usou apenas segurança na mensagem com WS-Security, mas foram também realizados ensaios com segurança no transporte com HTTPS. Deste modo, foram avaliadas ambas as abordagens.

A principal vantagem da segurança no transporte é a simplicidade, pois o serviço limita-se a tirar partido de uma capacidade oferecida pelo servidor aplicacional¹⁰. Uma desvantagem é a granularidade da autenticação, porque existe um único certificado digital para todo o servidor e um único conjunto de certificados clientes confiados, o que significa que estas definições não podem ser diferenciadas para cada serviço. Outra desvantagem é a protecção igual para todos os dados, o que significa que pode ser fraca demais para alguns e forte demais para outros. Além disso, não é possível expor apenas parte da informação a intermediários de auditoria.

A principal vantagem da segurança na mensagem é permitir proteger individualmente partes da mensagem. Além disso, os tokens de segurança permitem transportar chaves, certificados e credenciais diversas. Por exemplo, no protótipo, usa-se WS-Security com credenciais SAML. Outra vantagem é permitir a independência de servidor e de transporte. A principal desvantagem é o menor desempenho, com processamento mais lento e com maior utilização de memória, e o aumento significativo da dimensão das mensagens.

A Tabela 5 resume as vantagens e desvantagens da segurança no transporte e da segurança na mensagem.

Tabela 5 – Vantagens e desvantagens da segurança no transporte e da segurança na mensagem.

Segurança	No transporte	Na mensagem
Vantagens	<ul style="list-style-type: none"> • Simplicidade 	<ul style="list-style-type: none"> • Protecção individual de elementos da mensagem • Permite o transporte de tokens de segurança • Independência de servidor aplicacional • Independência de transporte
Desvantagens	<ul style="list-style-type: none"> • Granularidade de autenticação demasiado grossa • Protecção igual para todos os dados • Expõe tudo ou nada aos intermediários 	<ul style="list-style-type: none"> • Pior desempenho • Mensagens maiores e com legibilidade reduzida

É possível combinar ambas as abordagens para tentar obter as vantagens de cada uma. Uma prática comum é assinar a mensagem para garantir integridade e autenticidade e usar HTTPS para garantir a confidencialidade. Esta solução é mais flexível que a segurança no transporte simples, e tem um desempenho melhor que a cifra ao nível de mensagem [Adams04].

¹⁰ Neste caso, o servidor Apache Tomcat.

6.2.2. Autenticação

A autenticação responde à questão: qual a verdadeira identidade do cliente ou do serviço?

As duas normas mais relevantes para a autenticação são a WS-Security, que permite transportar o token de autenticação, e a SAML, que define o formato da credencial.













Normalmente, o mecanismo de autenticação é a primeira escolha de segurança, porque depois facilita a autorização e a protecção das mensagens. Foi esta a abordagem seguida no protótipo, tendo-se testado duas configurações alternativas de autenticação.

A primeira implementação foi baseada em autenticação directa. O Comprador e o Vendedor tinham um utilizador com senha no Notário. O Notário autenticava-se perante o Comprador e o Vendedor com um certificado digital. O token de autenticação utilizado neste caso continha o nome do utilizador e o resumo da senha. Esta solução por si só não era segura, porque era possível capturar o token e repeti-lo em mensagens diferentes. Para proteger o token de autenticação, foi acrescentada uma marca temporal e foi efectuada a cifra com a chave pública do Notário, para garantir que apenas ele conseguia ler o token. A resposta do Notário foi depois assinada com a chave privada correspondente ao certificado digital. Esta implementação foi efectuada com sucesso no protótipo usando o XWSS.

A segunda implementação foi baseada em autenticação com intermediário. O Comprador autenticava-se nas Finanças que depois emitiam uma asserção SAML de autenticação. O Vendedor fazia o mesmo, mas no Registo Civil. Ambas as asserções eram assinadas, para garantir a sua autenticidade. Cabia depois ao Notário verificar a assinatura das credenciais e decidir se as aceitava como forma de autenticação. Esta segunda implementação foi efectuada parcialmente com o XWSS. As credenciais não eram assinadas, mas eram emitidas e o seu conteúdo era lido. Na prática, esta implementação não garante segurança alguma.

A Tabela 6 resume os resultados da avaliação dos mecanismos de autenticação.

Tabela 6 – Resultados da avaliação dos mecanismos de autenticação.

Autenticação	XWSS	WSS4J	WSE 3
Utilizador-senha			
Certificados X.509			
Asserção SAML			
Kerberos			

Podemos concluir que o utilizador-senha e os certificados X.509 são os mecanismos de autenticação mais suportados, com implementações no XWSS, WSS4J e WSE 3. A SAML permite a autenticação

entre domínios, mas ainda não tem implementações disponíveis que sejam seguras. A autenticação com Kerberos é suportada apenas pelo WSE 3.

As asserções SAML são interessantes, pois permitem diferentes formas de identificação do domínio e do utilizador. A segurança é baseada em certificados, mas em vez de cada cliente ter um certificado próprio, basta confiar no certificado da autoridade de autenticação do domínio e depois verificar a asserção emitida.

6.2.3. Autorização

A autorização responde à questão: a acção sobre o recurso pode ser efectuada? O principal desafio da autorização é a granularidade, pois quanto mais fina for a distinção de recursos mais complexo é expressar a autorização.










A avaliação aqui apresentada foi baseada nos resultados dos ensaios, pois a extensão do cenário do protótipo para avaliar a autorização não foi concretizada por não existirem mecanismos de autorização disponíveis no “XWSS / JAX-WS 2 / Java”.

Os mecanismos de autorização propostos para serviços são praticamente todos baseados em autenticação prévia, podendo ser vistos como confirmação de atributos associados à identidade.

A única implementação disponível de autorização é o WSE 3, que permite inclusivamente a delegação. As verificações são efectuadas através dos mecanismos de autorização do sistema Windows, tirando partido do directório ActiveDirectory de utilizadores da organização.

A Tabela 7 resume os resultados da avaliação dos mecanismos de autorização.

Tabela 7 – Resultados da avaliação dos mecanismos de autorização.

Autorização	XWSS	WSS4J	WSE 3
Baseada em autenticação			
Usando mecanismos do sistema operativo			
Asserção SAML			

Apesar de não existirem implementações disponíveis, foram identificadas duas abordagens para autorização baseada em SAML. A primeira abordagem usaria uma asserção de autorização, onde a acção de negócio seria identificada por um URI. A segunda abordagem usaria uma asserção de atributos, definidos pela aplicação e associados a uma identidade, para expressar factos de negócio relevantes para a decisão de autorização.

6.2.4. Protecção de mensagens

A protecção de mensagens responde à questão: como proteger o conteúdo das mensagens dos serviços? A protecção pode garantir a integridade, a confidencialidade e a origem dos dados.

No protótipo, a protecção de mensagens foi efectuada com certificados digitais.













O XWSS utilizado no protótipo permite a assinatura digital, para garantir integridade e origem dos dados, e a cifra, para garantir a confidencialidade. A assinatura é efectuada com chave privada associada a um certificado digital, mas não são suportados resumos cifrados com chave simétrica (MAC). A cifra pode ser efectuada com chave pública de certificado digital ou então com chave simétrica. A granularidade da protecção pode incidir sobre: toda a mensagem, sobre o corpo ou cabeçalho, ou sobre conjuntos de elementos XML especificados por XPath ou por espaço de nomes. Uma falha do XWSS é não permitir verificar mensagens repetidas, através de uma opção de configuração, obrigando a desenvolvimento à medida que poderá, em muitos casos, não ser efectuado. Outra falha é não proteger as mensagens de erro.

As capacidades do WSS4J são equivalentes às do XWSS.

Os ensaios realizados com o WSE 3 permitiram confirmar que são suportados todos os casos da XWSS e adicionalmente também sessões seguras com WS-SecureConversation. Adicionalmente, o WSE 3 suporta a validação de mensagens, a verificação de repetições e a filtragem de excepções através de um intermediário de protecção de domínio de segurança, que pode ser modificado se necessário.

A Tabela 8 resume os resultados da avaliação dos mecanismos de protecção de mensagens.

Tabela 8 – Resultados da avaliação dos mecanismos de protecção de mensagens.

Protecção de mensagens	XWSS	WSS4J	WSE 3
Assinatura digital			
Cifra			
Detecção de repetições			
Sessão de segurança (WS-SecureConversation)			

6.2.5. Configuração

A configuração responde à questão: como configurar o cliente e serviço para a interacção segura? A configuração é definida pela vinculação de segurança entre cliente e serviço e está condicionada à expressividade do contrato. Actualmente nenhuma das implementações suporta a vinculação de

segurança e existem diferentes e interessantes abordagens à configuração ainda em fase de experimentação.

Vinculação de segurança

A vinculação não funcional entre cliente e serviço define a configuração de segurança, descrita por um contrato WS-Policy com vocabulário WS-SecurityPolicy. O cliente e o serviço negociam, por intersecção das suas políticas, uma alternativa suportada pelos dois, que depois dita a configuração da plataforma de cada um durante a invocação.

A *política* descreve os requisitos de forma declarativa. A *configuração* escolhe e parametriza os mecanismos necessários para cumprir a política.

Nenhuma das implementações actualmente disponíveis – XWSS, WSS4J e WSE 3 – permite realizar a negociação de políticas de forma integrada. No protótipo, a negociação de políticas foi efectuada para exemplos simples, de forma autónoma usando a biblioteca Apache Commons Policy 1.0.

A configuração de segurança pode ser definida em tempo de desenvolvimento, instalação ou execução. As implementações disponíveis apenas permitem a configuração no desenvolvimento e na instalação. O XWSS e o WSE 3 permitem a obtenção de alguns parâmetros de segurança na execução, mas não a definição do tipo de configuração.

No protótipo, a configuração de segurança foi definida de forma estática, usando o formato de configuração próprio do XWSS.

Confiança na meta-informação

A política do serviço é muito importante pois determina a configuração de segurança utilizada.

Um problema muitas vezes ignorado é a integridade e autenticidade da meta-informação. Se em vinculações estáticas esta abordagem pode ser aceitável, já em vinculações dinâmicas é uma vulnerabilidade de segurança significativa.

No protótipo o problema foi ignorado, por se usar apenas vinculação estática, mas foram identificadas duas formas de proteger a meta-informação. Uma hipótese seria usar segurança no transporte dos contratos. Outra hipótese seria acrescentar uma assinatura digital a cada contrato, que depois seria verificada.

Expressividade da WS-SecurityPolicy

A WS-SecurityPolicy permite descrever os parâmetros da configuração de segurança, como os algoritmos criptográficos a usar, as dimensões de chaves, etc. e quais as partes das mensagens a proteger.

A política define o tipo dos parâmetros mas não define instâncias concretas. Por exemplo, a política consegue estipular que a autenticação é feita com certificado digital, mas não consegue indicar exactamente que certificado deve ser usado.

A especificação de protecção pode englobar toda a mensagem SOAP, ou então apenas partes especificadas por XPath ou por espaço de nomes. A WS-PolicyAttachment permite definir políticas a aplicar a todo o serviço ou a operações individuais.

Uma das vantagens de ter uma política descrita em formato normalizado e interpretável por máquina é a possibilidade de usar analisadores para detectar erros frequentes de configuração de segurança. O WSE 3 disponibiliza a ferramenta 'Policy advisor' que analisa a configuração e detecta problemas típicos.

Âmbito da política de segurança

Finalmente, importa perceber onde termina a política e onde começa a coreografia de negócio. Os limites da política são os limites da configuração de segurança, que tem um objectivo específico, com um conjunto relativamente pequeno de alternativas disponíveis. Neste sentido, a política tem um âmbito fechado. A coreografia incide sobre a parte funcional do serviço e por isso o seu âmbito é aberto.

Diferentes abordagens à configuração

Cada uma das implementações XWSS, WSS4J e WSE 3 está a experimentar diferentes abordagens à configuração de segurança. Todas as abordagens usam formatos de configuração próprios e incompatíveis. A ideia é que a negociação será efectuada numa linguagem normalizada como a WS-SecurityPolicy, mas depois cada implementação vai gerar automaticamente uma configuração no seu formato próprio, para configurar os seus mecanismos.

O modelo de configuração do "XWSS / JAX-WS 2 / Java" é baseado num ficheiro de configuração incluído com o serviço durante a instalação. O ficheiro especifica requisitos para mensagens à entrada e à saída do serviço. Os detalhes de configuração, como por exemplo o certificado a usar para assinar as mensagens à saída, são especificados directamente no ficheiro ou então pedidos à aplicação através de uma classe de 'callback'. Esta classe permite uma cooperação entre aplicação e plataforma, sendo invocada para obter parâmetros, mas também para efectuar validações de tokens de informação, como senhas e marcas temporais.

O aspecto mais interessante da configuração do "WSS4J / Axis2 / Java" é o conceito de módulos para suporte a requisitos não funcionais, que são distribuídos em pacotes. Cada módulo inclui um conjunto de interceptores de mensagens SOAP com restrições de configuração que indicam a ordem relativa em que devem ser invocados na cadeia de processamento da mensagem, relativamente aos outros módulos. Isto permite, por exemplo, ao módulo de segurança, indicar que é o primeiro a ser invocado à entrada de mensagens, e o último à saída. Cada serviço especifica depois que módulos quer activar,

especificando também parâmetros de configuração. Neste caso, a configuração de segurança não fica separada da restante configuração do serviço, ficando no mesmo descritor. A principal vantagem dos módulos é simplificar a construção de uma plataforma à medida para dar resposta a requisitos não funcionais.

Finalmente, o “WSE 3 / Dot Net 2” apresenta a configuração mais fácil de usar. A configuração de um serviço é descrita por um ficheiro, designado por política, mas que se trata de um formato próprio de configuração e não de WS-Policy. A edição do ficheiro pode ser feita directa ou através de ecrãs de configuração (wizards). Neste ficheiro podem ser especificados cenários pré-definidos (turn-key scenarios)¹¹ como: utilizador com senha, certificados X.509 e Kerberos; com parâmetros por omissão, que podem ser ajustados em pormenor. Existe também a possibilidade de programar elementos de configuração à medida (custom assertions) para satisfação de requisitos mais específicos.

Comparação

A Tabela 9 compara os mecanismos de configuração das implementações avaliadas.

Tabela 9 – Comparação dos mecanismos de configuração da segurança.

Configuração	XWSS	WSS4J	WSE 3
Utiliza WS-Policy?	Não	Sim	Não
Utiliza WS-SecurityPolicy?	Não	Não	Não
Formato de configuração	Próprio	Próprio	Próprio
Definição de configuração	Instalação	Instalação	Instalação
Mais valias	‘Callback’ para obter parâmetros em tempo de execução	Módulos para simplificar instalação	Cenários pré-definidos Elementos de configuração à medida

6.3. Resumo

Neste capítulo foi apresentada a avaliação do desenvolvimento e protecção de serviços tendo por base os ensaios realizados e o protótipo do caso de estudo.

No que respeita a desenvolvimento de serviços foi avaliada em detalhe a vinculação dinâmica e a importância da gestão da meta-informação dos serviços.

No que respeita à protecção de serviços, começou-se por analisar a segurança no transporte em comparação com a segurança nas mensagens e, em termos gerais, conclui-se que a segurança no transporte é mais simples de utilizar, mas a segurança na mensagem é mais expressiva e flexível. De

¹¹ *Turn-key scenarios* pode traduzir-se livremente por *cenários chave-na-mão*.

seguida foram analisados os mecanismos de autenticação, autorização e protecção de mensagens, centrando-se a atenção nas normas WS-Security e SAML. Para terminar, a avaliação incidiu sobre a política e configuração de segurança do serviço, com apreciação da norma WS-SecurityPolicy e das diferentes abordagens para a configuração.

O próximo capítulo fecha a dissertação com as conclusões.

7. Conclusão

O capítulo final da dissertação apresenta os contributos da tese e identifica as oportunidades de trabalho futuro. Para terminar é apresentado um comentário final.

7.1. Contributos

Os contributos foram vários ao longo da dissertação e são descritos de seguida. No fim da secção, são resumidos os principais.

O *enquadramento* caracteriza os sistemas de informação empresariais e justificou a forma como os serviços – arquitectura e tecnologia – pretendem responder ao desafio de permanente mudança, onde os dados da organização são estáveis mas os processos de negócio são variáveis. A este respeito, conclui-se que:

- As características chave dos serviços são a flexibilidade, reutilização e interoperabilidade;
- O negócio define requisitos, que são traduzidos em especificações que depois são detalhadas em implementações;
- Os requisitos são funcionais quando dizem o que o sistema faz, e são não funcionais quando dizem quais as qualidades que o sistema apresenta no seu funcionamento;
- Os requisitos de segurança são não funcionais;
- A implementação de requisitos não funcionais deve ser efectuada como um aspecto independente da implementação dos requisitos funcionais.

De seguida, foi apresentado um retrato em largura da *plataforma de serviços*, descrevendo as normas propostas e as principais implementações disponíveis. O levantamento efectuado permite concluir que:

- As normas base da plataforma – representação de dados, transporte, mensagem, contrato e descoberta – estão disponíveis em diversas implementações, com interoperabilidade entre a maior parte delas;
- As normas de extensão – segurança, mensagens fiáveis, transacções, processos de negócio e gestão – estão ainda mais atrasadas;
- A segurança, pela sua importância decisiva para serviços que manipulam informação com valor, já tem modelo conceptual, normas e implementações disponíveis.

A *segurança de serviços* foi analisada em profundidade com a identificação das necessidades de protecção mais relevantes. Conclui-se que:

- A segurança dos serviços está centrada na protecção das mensagens, no controlo de acessos e na configuração que se pretende automática e dinâmica;
- As normas propõem mecanismos de autenticação, autorização, protecção de mensagens e de configuração;
- As principais implementações disponíveis são: “WSE 3 / Dot Net 2”, “WSS4J / Axis2 / Java” e “XWSS / JAX-WS 2 / Java”;
- As normas que podem ser avaliadas nas implementações disponíveis são: WS-Security, SAML e WS-SecurityPolicy.

Para avaliar as normas e implementações de serviços seguros foi necessário encontrar um *caso de estudo* que, pelo seu realismo e complexidade, as pudesse pôr à prova a sério, e que não fosse um “problema brinquedo” à medida das soluções propostas. Foi escolhido o processo de “compra e venda de imóvel” por ser real, complexo e por manipular informação com valor elevado, envolvendo diversos actores. O caso foi enquadrado no contexto organizacional e realizou-se uma descrição alto nível do processo. Depois foram escolhidos três cenários exemplificativos, que foram modelados em detalhe para identificar os seus requisitos funcionais e de segurança. Finalmente, foi escolhido o cenário de “assinatura de contrato-promessa de compra e venda” para a realização do protótipo. A descrição do caso de estudo permitiu concluir que:

- A representação de um processo de negócio em serviços não é trivial, nomeadamente no que respeita à tradução de requisitos;
- Para descrever requisitos é essencial identificar os principais actores do sistema;
- A plataforma de serviços deve permitir a satisfação de requisitos funcionais e não funcionais;
- A especificação deve indicar o que vai ser implementado pela plataforma/segurança e o que vai ser implementado pela aplicação/serviço/negócio;
- A abordagem centrada no modelo XML, com especificação directa dos contratos de dados, funcional e não funcional, é mais adequada para o desenvolvimento de serviços do que as alternativas centradas em objectos Java e Dot Net, pois estas últimas efectuem um mapeamento de conceitos do seu modelo para XML, que torna os contratos menos explícitos e portanto, mais difíceis de gerir e manter;

- A especificação do comportamento dos serviços deve também incluir as vinculações além das invocações, para permitir explorar todas as potencialidades da tecnologia durante o desenvolvimento.

Antes da realização do protótipo foram realizados diversos ensaios para aferir na prática as capacidades das implementações de serviços seguros disponíveis. O protótipo do caso de estudo foi depois realizado na implementação “XWSS / JAX-WS 2 / Java” em diversas etapas. Os resultados dos ensaios e do protótipo foram usados para fazer a *avaliação*, que permitiu concluir que:

- A segurança no transporte é mais simples de utilizar, a segurança nas mensagens é mais expressiva e flexível;
- A implementação de serviços seguros mais avançada e robusta é “WSE 3 / Dot Net 2”, mas não suporta políticas WS-Policy nem asserções SAML;
- “WSS4J / Axis2 / Java” promete suportar tudo, mas está ainda muito instável. A documentação é pouca e, em geral, de má qualidade;
- “XWSS / JAX-WS 2 / Java” é razoavelmente robusto, suporta os mecanismos mais importantes, tem uma arquitectura extensível e tem o código fonte aberto;
- Não existe nenhuma implementação de serviços seguros que permita avaliar simultaneamente os mecanismos de autenticação, autorização, protecção de mensagens e configuração. A autenticação é bem suportada, a autorização não é suportada, a protecção de mensagens é bem suportada e a configuração é suportada apenas em tempo de instalação;
- Não existe nenhuma implementação de serviços seguros capaz de efectuar a vinculação a partir de política, gerando a configuração de forma automática;
- A fronteira entre o que é realizado pela plataforma e pela aplicação é difícil de traçar. No protótipo colocaram-se os seguintes problemas:
 - A assinatura do contrato é visível para o negócio?
 - O que deve ser assinado: o contrato ou a mensagem que transporta o contrato?
 - Como validar e ler credenciais SAML que contêm atributos de negócio?
- A fronteira entre os mecanismos de segurança do sistema operativo e da plataforma é também difícil de traçar. A segurança de serviços tem um modelo abstracto que permite integrar sistemas diversos para construir mecanismos de mais alto nível para trocas de chave, autenticação, autorização, auditoria e outras operações nas quais se deposita confiança. No entanto, não foi aprofundado quando e como se deve fazê-lo;

- A gestão da meta-informação é importante. A reutilização dos esquemas de dados e a sua disponibilização em catálogos de informação, pode ser a base de uma arquitectura de informação empresarial para integração, como a proposta no artigo de Guerra e Pardal [Guerra04].

Finalmente, conclui-se que uma plataforma de serviços deve disponibilizar os seguintes mecanismos para permitir uma implementação de segurança:

- Declaração de requisitos de segurança com política;
- Declaração da configuração de segurança;
- Gestão de contextos de execução;
- Intercepção do processamento de mensagens;
- Intercepção do processamento de operações.

A *política* do serviço deve expressar os requisitos de segurança de forma declarativa¹². Esta capacidade é necessária, por exemplo, para declarar que um serviço pode ser invocado com segurança no transporte ou com segurança na mensagem.

A *configuração* deve indicar quais os mecanismos de segurança a activar e quais os seus parâmetros, que poderão ser pedidos à aplicação em tempo de execução. Esta capacidade é requerida, por exemplo, para indicar qual a chave a usar para assinar as mensagens enviadas por um serviço.

A *gestão de contextos* permite manter variáveis de estado da segurança com diferentes âmbitos e indexações. Os contextos permitem também a partilha de informação de processamento entre plataforma, implementação de segurança e aplicação. Alguns dos contextos que deverão existir são: aplicação, sessão, operação e tarefa. Por exemplo, o contexto de sessão permite guardar uma chave utilizada para cifrar um conjunto de mensagens. Cada contexto deverá dar acesso à política e configuração de segurança em vigor, de acordo com o seu âmbito. Por exemplo, o contexto de operação poderá disponibilizar qual a política e configuração efectivas que foram negociadas com o cliente para essa execução da operação.

A *intercepção do processamento de mensagens* deve permitir aceder ao seu conteúdo – cabeçalho e corpo – e influenciar o seu encaminhamento. Esta capacidade torna possível, por exemplo, a assinatura de uma mensagem à saída e a verificação de mensagem à chegada, sendo rejeitada em caso de assinatura inválida. O mais comum é o processamento de mensagens ser em cadeia, mas existem

¹² Declarativa no sentido em que são indicadas as condições que têm que ser satisfeitas e não a forma como devem ser satisfeitas.

implementações que permitem fluxos mais elaborados, como o SPEF (SOAP Profile Enabling Framework) [Malek05] da Fujitsu.

A *intercepção do processamento de operações* deve permitir influenciar quais os objectos de negócio, acesso a dados e adaptadores de invocação remota que são instanciados durante a execução da operação do serviço, para permitir acrescentar ou modificar o comportamento existente, sempre que necessário. Isto obriga a que o código da aplicação efectue a instanciação de objectos através de fábricas e que os objectos de negócio sejam encapsulados com interfaces para poderem ser substituídos transparentemente¹³. Esta capacidade é necessária, por exemplo, para implementar mecanismos genéricos de autorização.

Em resumo, os *principais contributos* desta tese são:

- Retrato abrangendo em largura as normas e implementações de toda a plataforma de Web Services, e em profundidade as normas e implementações de segurança;
- Avaliação da tecnologia de serviços seguros a partir de ensaios efectuados a todas as implementações disponíveis e com o protótipo de caso de estudo;
- Identificação dos mecanismos de uma plataforma de serviços necessários para implementação de segurança.

7.2. Trabalho futuro

Foram identificados muitos tópicos de interesse a explorar em trabalho futuro.

Um dos desafios será *manter a actualização do levantamento de normas* de modo a acompanhar a evolução da tecnologia de serviços, em especial no que respeita à segurança.

O levantamento de normas realizado pode servir de base a outros trabalhos de âmbito similar, com *casos de estudo para avaliar tecnologias de mensagens fiáveis e de transacções*. Se assim for, será possível confirmar se os mecanismos identificados para suportar a implementação de segurança na plataforma são também suficientes para outros requisitos não funcionais.

Existe também trabalho a desenvolver nos aspectos horizontais da plataforma de serviços, ou seja, em *normas para áreas de negócio*. Futuramente faz sentido desenvolver normas sectoriais para a Banca, Seguros, Saúde, etc. para facilitar os projectos de integração e aumentar a reutilização efectiva.

¹³ Utilização dos padrões de desenho 'factory' e 'decorator', respectivamente [Gamma95].

Outro grande desafio é propor *metodologias para modelação de processos de negócio em serviços*, incidindo inicialmente apenas sobre requisitos funcionais, mas podendo depois ser estendidas para requisitos de segurança. A metodologia utilizada no protótipo funcionou bem, mas é um cenário de âmbito demasiado limitado para efectuar uma avaliação correcta. Seria interessante aplicá-la a um sistema de maior dimensão. Para os requisitos de segurança, seria também interessante ter uma forma de definir a política nos vários níveis de abstracção: requisitos, especificação e implementação.

Nas implementações da tecnologia de serviços, são necessárias *ferramentas mais centradas no modelo XML dos serviços*, em vez de se centrarem no modelo de objectos Java ou Dot Net. O principal objectivo das novas ferramentas deve ser tornar mais simples o acesso ao XML e não escondê-lo. Uma proposta concreta é o embrulho (wrapping) de documentos XML que foi descrita no capítulo da avaliação. As ferramentas JAX-WS 2 e JAX-B 2 representam uma evolução significativa neste sentido.

Para além das ferramentas mais orientadas a XML, existe outra ideia a experimentar para tornar a fronteira entre plataforma/segurança e aplicação/serviço/negócio menos rígida, que é o *relatório de segurança*. A ideia é que a plataforma deve realizar as verificações criptográficas de todos os tokens de segurança das mensagens e produzir um relatório, indicando o que fez e como fez, com detalhe dos parâmetros tais como chaves, certificados, etc. A aplicação depois analisa o relatório e toma a decisão de confiar ou não. Desta forma, pode também aceder aos dados da segurança que sejam relevantes para o negócio, como atributos de uma asserção SAML, por exemplo.

Finalmente, seria interessante *incorporar a tecnologia de segurança nos adaptadores dos motores de integração* [Linthicum00], para permitir estabelecer ligações com aplicações remotas usando Web Services seguros. Alguns exemplos destes motores são o Microsoft BizTalk ou o IBM Websphere ESB, que além de adaptadores para ligar a outras aplicações têm também filas de mensagens assíncronas e transformadores para compensar diferenças entre esquemas de dados.

7.3. Comentário final

“Those who cannot remember the past are condemned to repeat it.” George Santayana

Já várias vezes no passado a indústria de tecnologias de informação e comunicação apresentou soluções “definitivas” para “todos” os problemas dos seus clientes empresariais. Mais recentemente, na década de 1990, foi proposta a CORBA, que hoje se pode considerar como um falhanço comercial e técnico. Henning acompanhou a história desta tecnologia de perto e recentemente publicou um artigo [Henning06], onde resumiu algumas boas práticas a seguir na definição de normas:

1. A norma não deve ser inovadora, mas sim consagrar melhores práticas;
2. Nenhuma norma deve ser aprovada sem pelo menos uma implementação de referência;

3. Nenhuma norma deve ser aprovada sem ter sido usada em projectos de complexidade realística;
4. A evolução da norma deve ser aberta a sugestões da comunidade, mas decidida por um “núcleo duro”, comprometido com a visão original do sistema.

O ponto 1 é parcialmente satisfeito pelos Web Services pois em muitas das normas aproveitam ideias já consolidadas em ciência informática, nomeadamente de *sistemas distribuídos* – invocações remotas, filas de mensagens, gestão de nomes, segurança e transacções – de *linguagens de programação orientadas a objectos* – encapsulamento e polimorfismo – e de *componentes* – uso obrigatório de interfaces e meta-informação disponível em tempo de execução. Noutros casos, as normas propõem abordagens ainda não suficientemente consolidadas, como acontece por exemplo com a norma REL que incide sobre a protecção de direitos de autor no acesso a informação.

O ponto 2 é parte integrante do processo de normalização seguido até agora para os Web Services, em que a aprovação de normas é precedida por implementações práticas, de um ou mais promotores, que são testadas em conjunto em workshops de interoperabilidade.

O ponto 3 não se pode considerar totalmente conseguido, pois apesar de já existirem muitos projectos reais que usam as tecnologias de base da plataforma, no que respeita às extensões, só agora começam a surgir contributos como esta tese, que avaliam as normas e implementações de segurança com casos de estudo reais e exigentes.

Finalmente, no ponto 4, existem algumas indefinições quanto ao futuro, mas até agora toda a construção da plataforma tem sido conduzida pelas quatro grandes empresas Microsoft, IBM, Oracle e Sun que, para já, se mantêm alinhadas estrategicamente, apenas com divergências de pormenor.

Os Web Services seguem no bom caminho, mas será ainda necessário esperar para ver se a visão dos serviços se concretiza na totalidade. Actualmente já se pode afirmar que os Web Services simplificam a integração entre ambientes Java e Dot Net e, no que respeita à segurança, suportam os cenários mais comuns por configuração, mas apenas de forma estática, em tempo de instalação. As futuras implementações prometem melhorias significativas. O rumo está traçado.

8. Bibliografia

- [Adams04] Adams, H., *Best Practices for Web services, Part 11: Web services security*, IBM Developer Works, 2004
<http://www.ibm.com/developerworks/webservices/library/ws-best11/>
- [Anderson01] Anderson, R., *Security Engineering*, Wiley, 2001
- [Anderson05] Anderson, A., *WS-Security policy profile of WS-PolicyConstraints*, OASIS, 2005 <http://research.sun.com/projects/xacml/ws-security-profile-of-ws-policy-constraints-wd-04.pdf>
- [Apache05] Apache, *Apache Ant web site*, 2005 <http://ant.apache.org/>
- [Apache06] Apache, *Apache Axis2 1.0 web site*, 2006
http://ws.apache.org/axis2/1_0/index.html
- [Apache06b] Apache, *Securing SOAP Messages with WSS4J*, 2006
http://ws.apache.org/axis2/modules/rampart/1_0/security-module.html
- [Baglietto02] Baglietto, P.; Maresca, M.; Parodi, A. & Zingirian, N., *Deployment of Service Oriented Architecture for a Business Community*, Proceedings of the Sixth International Enterprise Distributed Object Computing Conference (EDOC, IEEE Computer Society, 2002
<http://ieeexplore.ieee.org/iel5/8230/25384/01137718.pdf?tp=&arnumber=1137718&isnumber=25384>
- [Baligand04] Baligand, F. & Monfort, V., *A concrete solution for web services adaptability using policies and aspects*, ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing, ACM Press, 2004, 134-142
- [Ballinger01] Ballinger, K.; Brittenham, P.; Malhotra, A.; Nagy, W.A. & Pharies, S., *Web Services Inspection Language (WS-Inspection) 1.0*, Microsoft, IBM, 2001
<http://www-128.ibm.com/developerworks/library/specification/ws-wsilspec/>
- [Barbir05] Barbir, A.; Gudgin, M.; McIntosh, M. & Morrison, K.S., *WS-I Basic Security Profile Version 1.0*, WS-I, Nortel Networks, Microsoft, IBM, Layer 7, 2005
<http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>
- [Barton00] Barton, J.J.; Thatte, S. & Nielsen, H.F., *SOAP Messages with Attachments*, W3C, Hewlett Packard Labs, Microsoft, 2000
<http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211>
- [Bengtsson05] Bengtsson, A. & Westerdahl, L., *Secure Choreography of Cooperating Web Services*, Proceedings of the Third European Conference on Web Services (ECOWS, IEEE Computer Society, 2005
http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=33570&arnumber=1595725&count=29&index=19
- [Bhargavan05] Bhargavan, K.; Fournet, C.; Gordon, A.D. & O'Shea, G., *An advisor for web services security policies*, SWS '05: Proceedings of the 2005 workshop on Secure web services, ACM Press, 2005, 1-9

- [Booth05] Booth, D. & Liu, C.K., *Web Services Description Language (WSDL) Version 2.0*, W3C, Hewlett-Packard, SAP Labs, Booth05
<http://www.w3.org/TR/2005/WD-wsdl20-primer-20050803>
- [Box04a] Box, D. & Curbera, F., *Web Services Addressing (WS-Addressing)*, W3C, Microsoft, IBM, BEA, SAP, 2004
<http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/>
- [Bray04] Bray, T.; Paoli, J.; McQueen, C.M.S.; Maler, E. & Yergeau, F., *Extensible Markup Language (XML) 1.0 (Third Edition)*, W3C, Textuality and Netscape, Microsoft, Sun Microsystems, 2004 <http://www.w3.org/TR/2004/REC-xml-20040204>
- [Cabrera04] Cabrera, L.F.; Kurt, C. & Box, D., *An Introduction to the Web Services Architecture and Its Specifications Version 2.0*, MSDN, 2004
- [Cantor04] Cantor, S.; Kemp, J.; Philpott, R. & Maler, E., *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, OASIS, Internet2, Nokia, RSA Security, Sun Microsystems, 2004
<http://xml.coverpages.org/SAML-core-20-CD-01.pdf>
- [Clement04] Clement, L.; Hatley, A.; von Riegen, C. & Rogers, T., *UDDI Version 3.0.2*, OASIS, Systinet, IBM, SAP AG, Computer Associates, 2004
<http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>
- [Cruellas03] Cruellas, J.C.; Karlinger, G.; Pinkas, D. & Ross, J., *XML Advanced Electronic Signatures (XAdES)*, W3C, UPC, IAIK, Bull, Security and Standards, 2003
<http://www.w3.org/TR/2003/NOTE-XAdES-20030220/>
- [Curbera04] Curbera, F. & Schlimmer, J., *Web Services Metadata Exchange (WS-MetadataExchange)*, MSDN, Microsoft, IBM, Computer Associates, SAP, BEA Systems, Sun Microsystems, webMethods, 2004
<http://msdn.microsoft.com/ws/2004/09/ws-metadataexchange/>
- [Curbera05] Curbera, F.; Leymann, F.; Storey, T.; Ferguson, D. & Weerawarana, S., *Web Services Platform Architecture: Soap, WSDL, WS-Policy, WS-Addressing, WS-Bpel, WS-Reliable Messaging and More*, Prentice Hall, 2005
- [Czajkowski04] Czajkowski, K.; Ferguson, D.F.; Foster, I.; Frey, J.; Graham, S.; Sedukhin, I.; Snelling, D.; Tuecke, S. & Vambenepe, W., *The WS-Resource Framework Version 1.0*, Computer Associates International, Inc., Fujitsu Limited, Hewlett-Packard Development Company, International Business Machines Corporation and The University of Chicago, 2004 <http://www.globus.org/wsrf/specs/ws-wsrf.pdf>
- [Davis05] Davis, D., *Web Services Polling (WS-Polling)*, W3C, IBM, 2005
<http://www.w3.org/Submission/2005/SUBM-ws-polling-20051026/>
- [DeMartini04] DeMartini, T.; Nadalin, A.; Kaler, C.; Monzillo, R. & Baker, P.H., *Web Services Security Rights Expression Language (REL) Token Profile*, OASIS, ContentGuard, Inc., IBM, Microsoft Corporation, Sun Microsystems, Verisign, 2004 <http://docs.oasis-open.org/wss/oasis-wss-rel-token-profile-1.0.pdf>

- [Diaz02] Diaz, A.; Lucassen, J. & Wiecha, C.F., (*WSXL*) *Web Service Experience Language Version 2*, IBM, 2002
<http://www.ibm.com/developerworks/library/specification/ws-wsxl/>
- [Dumbill01] Dumbill, E.; Johnston, J. & Laurent, S.S., *Programming Web Services with XML-RPC*, O'Reilly, 2001
- [Eastlake02a] Eastlake, D.; Reagle, J. & Solo, D., *XML-Signature Syntax and Processing*, W3C, 2002 <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>
- [Eastlake02b] Eastlake, D. & Reagle, J., *XML Encryption Syntax and Processing*, W3C, 2002 <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>
- [Eddon98] Eddon, G. & Eddon, H., *Inside Distributed COM*, Microsoft Press, 1998
- [Endrei04] Endrei, M.; Ang, J.; Arsanjani, A.; Chua, S.; Comte, P.; Krogdahl; Luo, M. & Newling, T., *Service-Oriented Architecture and Web Services*, IBM RedBooks, 2004 <http://publib-b.boulder.ibm.com/abstracts/sg246303.html?Open>
- [Epifanio05] Epifânio, T. & Rasteiro, R., *Segurança num Sistema de Gestão Documental*, Trabalho Final de Curso, Licenciatura em Engenharia Informática e de Computadores, Instituto Superior Técnico, 2005
- [Essin98] Essin, D.J., *Patterns of trust and policy*, NSPW '97: Proceedings of the 1997 workshop on New security paradigms, ACM Press, 1997, 38-47
- [Fallside04] Fallside, D.C. & Walmsley, P., *XML Schema Part 0: Primer Second Edition*, W3C, 2004 <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>
- [Feingold05a] Feingold, M., *Web Services Coordination (WS-Coordination) Version 1.0*, IBM, Microsoft, Hitachi, Arjuna Technologies, IONA, 2005
<http://www.ibm.com/developerworks/library/specification/ws-tx/>
- [Feingold05b] Feingold, M., *Web Services Atomic Transaction (WS-AtomicTransaction) Version 1.0*, IBM, Microsoft, Hitachi, Arjuna Technologies, IONA, 2005
<http://www.ibm.com/developerworks/library/specification/ws-tx/>
- [Feingold05c] Feingold, M., *Web Services Business Activity Framework (WS-BusinessActivity) Version 1.0*, IBM, Microsoft, Hitachi, Arjuna Technologies, IONA, 2005 <http://www.ibm.com/developerworks/library/specification/ws-tx/>
- [Ferris04] Ferris, C.; Liu, C.K.; Nottingham, M.; Yendluri, P.; Gudgin, M.; Ballinger, K. & Ehnebuske, D., *WS-I Basic Profile Version 1.1*, WS-I, Microsoft, IBM, SAP, BEA Systems, webMethods, 2004 <http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html>
- [Ferris05] Ferris, C. & Langworthy, D., *Web Services Reliable Messaging Protocol (WS-ReliableMessaging)*, Microsoft, IBM, BEA, TIBCO Software, 2005
<http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-ReliableMessaging.pdf>

- [Fielding99] Fielding, R.; Gettys, J.; Mogul, J.; Frystyk, H.; Masinter, L.; Leach, P. & Lee, T.B., *Hypertext Transfer Protocol -- HTTP/1.1*, IETF, 1999
<http://www.w3.org/Protocols/rfc2616/rfc2616.txt>
- [Fowler02] Fowler, M.; Rice, D.; Foemmel, M.; Hieatt, E.; Mee, R. & Stafford, R., *Patterns of Enterprise Application Architecture*, Addison Wesley, 2002
- [Fowler99] Fowler, M. & Scott, K., *UML Distilled*, Addison-Wesley, 1999
- [Freed96] Freed, N. & Borenstein, N., *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, IETF, 1996
<http://www.ietf.org/rfc/rfc2045.txt>
- [Fuller05] Fuller, J.; Krishnan, M.; Swenson, K. & Ricker, J., *Asynchronous Service Access Protocol (ASAP) Version 1.0*, OASIS, 2005 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=asap
- [Gamma95] Gamma, E.; Helm, R.; Johnson, R. & Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995
- [Gaston03] Gaston, L., *Open Smart Card Infrastructure for Europe v2*, 2003
- [Geller04a] Geller, A., *Web Service Enumeration (WS-Enumeration)*, Microsoft, Systinet, Sonic Software, BEA, Computer Associates, 2004
<http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-enumeration.pdf>
- [Geller04b] Geller, A., *Web Service Transfer (WS-Transfer)*, Microsoft, Systinet, Sonic Software, BEA, Computer Associates, 2004
<http://msdn.microsoft.com/ws/2004/09/ws-transfer/>
- [Geller04c] Geller, A., *Web Services Eventing (WS-Eventing)*, Microsoft, IBM, TIBCO Software, BEA Systems, Computer Associates, Sun Microsystems, 2004
<http://www-128.ibm.com/developerworks/webservices/library/specification/ws-eventing/>
- [Geller04d] Geller, A., *Web Services for Management (WS-Management)*, Microsoft, Sun, Intel, AMD, Dell, 1994 <http://msdn.microsoft.com/library/en-us/dnglobspec/html/ws-management1004.pdf>
- [Graham01] Graham, S.; Simeonov, S.; Boubez, T.; Davis, D.; Daniels, G.; Nakamura, Y. & Neyama, R., *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*, Sams Publishing, 2001
- [Graham04] Graham, S. & Niblett, P., *Web Services Notification (WS-Notification) Version 1.0*, IBM, Sonic Software, TIBCO Software, Akamai Technologies, SAP AG, Globus, Argonne National Laboratory, Hewlett-Packard, 2004
<http://ifr.sap.com/ws-notification/ws-notification.pdf>

- [Grangard01] Grangard, A., *ebXML Technical Architecture Specification v1.0.4*, EDI France, DataChannel, XML Global Technologies, TIE, ATPCO, GIP-MDS, Group 8760, DataAccess Technologies, Military Traffic Management Command, US Army, Sun Microsystems, SWIFT, Worldspan, Chungnam National University, OMG, General Motors, eProcessSolutions, KPMG Consulting, MITRE, IBM, NIST, Encoda Systems, Inc., 2001 <http://ebxml.org/specs/index.htm>
- [Gudgin03] Gudgin, M.; Hadley, M.; Mendelsohn, N.; Moreau, J. & Nielsen, H.F., *SOAP Version 1.2 Part 1: Messaging Framework*, W3C, Microsoft, Sun Microsystems, IBM, Canon, 2003 <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>
- [Gudgin05a] Gudgin, M.; Mendelsohn, N.; Nottingham, M. & Ruellan, H., *XML-binary Optimized Packaging*, W3C, Microsoft, IBM, BEA, Canon, 2005 <http://www.w3.org/TR/2005/REC-xop10-20050125/>
- [Gudgin05b] Gudgin, M.; Mendelsohn, N.; Nottingham, M. & Ruellan, H., *SOAP Message Transmission Optimization Mechanism*, W3C, Microsoft, IBM, BEA, Canon, 2005 <http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/>
- [Gudgin05c] Gudgin, M. & Nadalin, A., *Web Services Secure Conversation Language (WS-SecureConversation)*, Microsoft, IBM, OpenNetwork, Layer 7, Computer Associates, VeriSign, BEA, RSA Security, Ping Identity, Actional, Computer Associates, 2005 <http://www.ibm.com/developerworks/library/specification/ws-secon/>
- [Gudgin05d] Gudgin, M. & Nadalin, A., *Web Services Trust Language (WS-Trust)*, Microsoft, IBM, OpenNetwork, Layer 7, Computer Associates, VeriSign, BEA, Oblix, Reactivity, RSA Security, Ping Identity, VeriSign, Actional, 2005 <http://www.ibm.com/developerworks/library/specification/ws-trust/>
- [Guerra04] Guerra, M.; Pardal, M. & da Silva, M.M., *An Integration Methodology based on the Enterprise Architecture*, Proc. of the 2004 Conference of the UK Academy for Information Systems (UKAIS 2004), 2004 <http://mflpar.googlepages.com/GuerraPardalUkais2004.pdf>
- [Guerra05] Guerra, M.N.C., *Suporte Electrónico de Processos na Cadeia de Aprovisionamento de Produtos Perecíveis*, Tese de Mestrado em Engenharia Informática e de Computadores, Instituto Superior Técnico, 2005
- [Gutierrez05] Gutierrez, C.; Medina, E.F. & Piattini, M., *Web services enterprise security architecture: a case study*, SWS '05: Proceedings of the 2005 workshop on Secure web services, ACM Press, 2005, 10-19
- [HallamBaker05] Baker, P.H. & Mysore, S.H., *XML Key Management Specification (XKMS 2.0) Version 2.0*, W3C, Verisign, 2005 <http://www.w3.org/TR/2005/REC-xkms2-20050628/>
- [Henning06] Henning, M., *The Rise and Fall of CORBA*, ACM Queue vol. 4, no. 5, 2006 <http://www.acmqueue.com/modules.php?name=Content&pa=showpage&pid=396>

- [Hogg04] Hogg, K.; Chilcott, P.; Nolan, M. & Srinivasan, B., Castro, V.E. (eds.), *An Evaluation of Web Services in the Design of a B2B Application*, 27th Australasian Computer Science Conference, The University of Otago, Dunedin, New Zealand. *Conferences in Research and Practice in Information Technology*, 2004, 26
<http://portal.acm.org/citation.cfm?id=979962&coll=ACM&dl=ACM&CFID=51181327&CFTOKEN=45118188>
- [Hogg05] Hogg, J.; Smith, D.; Chong, F.; Taylor, D.; Wall, L. & Slater, P., *Web Service Security Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0*, Microsoft, 2005
- [Housley99] Housley, R.; Ford, W.; Polk, W. & Solo, D., *Internet X.509 Public Key Infrastructure*, IETF, 1999 <http://www.ietf.org/rfc/rfc2459.txt>
- [IBM02] IBM & Microsoft, *Security in a Web Services World: A Proposed Architecture and Roadmap Version 1.0*, IBM, Microsoft, 2002
<http://www.ibm.com/developerworks/library/specification/ws-secmap/>
- [IBM05] IBM, *About IBM*, IBM Web Site, 2005 <http://www.ibm.com/ibm/us/>
- [Inmon93] Inmon, W., *Data Architecture - The Information Paradigm - 2nd edition*, QED Technical Publishing Group, 1993
- [Iwasa04] Iwasa, K., *Web Services Reliable Messaging TC WS-Reliability 1.1*, OASIS, Fujitsu Limited, Novell, Inc., Oracle Corporation, Sun Microsystems, 2004
<http://docs.oasis-open.org/wsrn/ws-reliability/v1.1>
- [Kaler03] Kaler, C. & Nadalin, A., *Web Services Federation Language (WSFederation) Version 1.0*, OASIS, Microsoft, IBM, VeriSign, BEA, RSA Security, VeriSign, 2003 <http://www.ibm.com/developerworks/library/specification/ws-fed/>
- [Kaler05] Kaler, C. & Nadalin, A., *Web Services Security Policy Language (WS-SecurityPolicy) Version 1.1*, Microsoft, IBM, VeriSign, RSA Security, 2005
<http://www.ibm.com/developerworks/library/specification/ws-secpol/>
- [Kavantzias04] Kavantzias, N.; Burdett, D. & Ritzinger, G., *Web Services Choreography Description Language Version 1.0*, W3C, Oracle, Commerce One, Novell, 2004 <http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/>
- [Kiczales97] Kiczales, G.; Lamping, J.; Mendhekar, A.; Maeda, C.; Lopes, C.V.; Loingtier, J. & Irwin, J., *Aspect-Oriented Programming*, Proceedings of the European Conference on Object-Oriented Programming (ECOOP), Finland, Springer-Verlag, 1997, *LNCS 1241*
<http://www2.parc.com/csl/groups/sda/publications/papers/Kiczales-ECOOP97/for-web.pdf>
- [Klensin01] Klensin, J., *Simple Mail Transfer Protocol*, IETF, 2001
<http://www.ietf.org/rfc/rfc2821.txt>
- [Kohl93] J. Kohl, C.N., *The Kerberos Network Authentication Service (V5)*, IETF, 1993
<http://www.ietf.org/rfc/rfc1510.txt>

- [Kotler99] Kotler, P.; Armstrong, G.; Saunders, J. & Wong, V., *Principles of Marketing 2nd Edition*, Prentice Hall Europe, 1999
- [Kreger03] Kreger, H., *Fulfilling the web services promise*, Communications of the ACM, COMMUNICATIONS OF THE ACM June 2003/Vol. 46, No. 6 29, 2003, 46
<http://portal.acm.org/citation.cfm?id=777334&coll=ACM&dl=ACM&CFID=48963334&CFTOKEN=98490806>
- [Kropp03] Kropp, A.; Leue, C. & Thompson, R., *Web Services for Remote Portlets Specification*, OASIS, IBM, Vignette Corporation, 2003 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp
- [Laudon02] Laudon, K. & Laudon, J., *Management Information Systems*, Pearson Prentice-Hall, 2002
- [Linthicum00] Linthicum, D.S., *Enterprise Application Integration*, Addison-Wesley, 2000
- [Little03a] Little, M., *Web Services Composite Application Framework (WS-CAF) version 1.0*, Sun, Oracle, IONA, Arjuna, Fujitsu, 2003
http://www.arjuna.com/library/specs/ws_caf_1-0/WS-CAF-Primer.pdf
- [Lockhart94] Lockhart, H., *OSF DCE Guide to Developing Distributed Applications*, McGraw-Hill, 1994
- [MacDonald03] MacDonald, M., *Microsoft .NET Distributed Applications: Integrating XML Web Services and .NET Remoting*, Microsoft Press, 2003
- [Malek05] Malek, H.B. & Durand, J., Kitsuregawa, M. (eds.), *A SOAP Container Model for e-Business Messaging Requirements*, Proceedings of WISE 2005, Springer-Verlag, 2005, LNCS 3806, 643
- [Marques98] Marques, J.A. & Guedes, P., *Tecnologia de Sistemas Distribuídos*, FCA, 1998
- [McGovern03] McGovern, J.; Tyagi, S.; Stevens, M. & Matthew, S., *Java Web Services Architecture*, Morgan Kaufmann, 2003
- [Mello05] Mello E., Fraga J., *Mediation of Trust across Web Services*, Proceedings of the IEEE International Conference on Web Services (ICWS), 2005
http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=32665&arnumber=1530842&count=129&index=72
- [Mendling04] Mendling, J.; uttgens, N. & Neumann, M., *A Comparison of XML Interchange Formats for Business Process Modelling*, Proceedings of EMISA 2004 - Information Systems in E-Business and E-Government. LNI. 2004., 2004
citeseer.ist.psu.edu/mendling04comparison.html
- [Microsoft05] Microsoft, *Company Information*, Microsoft Web Site, 2005
<http://www.microsoft.com/mscorp/info/>
- [Microsoft05c] Microsoft, *Microsoft Web Services Enhancements (WSE) 3.0 documentation*, 2005
<http://msdn.microsoft.com/webservices/webservices/building/wse/default.aspx>

- [Microsoft06b] Microsoft, *WS-Policy Interop Workshop*, Microsoft Web Site, 2006
<http://msdn.microsoft.com/webservices/community/workshops/wspolicygermany042006.aspx>
- [Moses05] Moses, T., *eXtensible Access Control Markup Language (XACML) Version 2.0*, OASIS, Entrust, 2005 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
- [Nadalin04] Anthony Nadalin, C.K., *Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)*, OASIS, IBM, Microsoft, Verisign, Sun, 2004
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
- [Nielsen02b] Nielsen, H.F.; Christensen, E. & Farrell, J., *WS-Attachments*, Microsoft, IBM, 2002 <http://msdn.microsoft.com/library/en-us/dnglobspec/html/draft-nielsen-dime-soap-01.txt>
- [Nolan04] Nolan, P., *Understand WS-Policy processing*, IBM Developer Works, 2004
<http://www.ibm.com/developerworks/webservices/library/ws-policy.html>
- [OMG91] OMG, *The Common Object Request Broker: Architecture and Specification (CORBA)*, 1991
- [Oracle05] Oracle, *Company Information*, Oracle Web Site, 2005
<http://www.oracle.com/corporate/about.html>
- [Papazoglou03] Papazoglou, M.P., *Service -Oriented Computing: Concepts, Characteristics and Directions*, Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE, 2003
<http://ieeexplore.ieee.org/iel5/8885/28063/01254461.pdf?tp=&arnumber=1254461&isnumber=28063>
- [Pardal04] Pardal, M., *antInclude*, antInclude web site, 2004
<http://mega.ist.utl.pt/~mflpar/antInclude>
- [Pardal06] Pardal, M., *Em construção: uma análise ao estado actual da plataforma de Serviços Web para negócio electrónico*, XATA2006, XML: Aplicações e Tecnologias Associadas, 2006
<http://mflpar.googlepages.com/PardalWsXata2006.pdf>
- [Parr02] Parr, F., *HTTPR Specification*, IBM, 2002
<http://www.ibm.com/developerworks/webservices/library/ws-httpspec/>
- [Porter04] Porter, M.E., *Competitive Strategy*, Free Press, 2004
- [Rescorla00] Rescorla, E., *HTTP Over TLS*, IETF, 2000 <http://www.ietf.org/rfc/rfc2818.txt>
- [Rosenberg04] Rosenberg, J. & Remy, D., *Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature and XML Encryption*, SAMS, 2004
- [Samaranayake06] Samaranayake, S., *Web services Policy - Why, What & How*, WSO2 Oxygen Tank, 2006
http://www.wso2.net/2006/01/web_services_policy_why_what_how

- [Schlimmer02] Schlimmer, J.C., *Web Services Description Requirements*, W3C, Microsoft, 2002 <http://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028>
- [Schlimmer05a] Schlimmer, J., *Web Services Dynamic Discovery (WS-Discovery)*, Microsoft, BEA Systems, Canon, Intel, webMethods, 2005
<http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-Discovery.pdf>
- [Schlimmer05b] Schlimmer, J., *Devices Profile for Web Services*, Microsoft, Ricoh, Intel, Lexmark, 2005
<http://specs.xmlsoap.org/ws/2005/05/devprof/devicesprofile.pdf>
- [Schlimmer06] Schlimmer, J., *Web Services Policy Framework (WSPolicy) Version 1.2*, Microsoft, IBM, VeriSign, Sonic Software, SAP, BEA Systems, 2006
<http://specs.xmlsoap.org/ws/2004/09/policy/ws-policy.pdf>
- [Schwarz05] Schwarz, J.; Hartman, B.; Nadalin, A.; Kaler, C.; Davis, M.; Hirsch, F. & Morrison, K.S., *Security Challenges, Threats and Countermeasures Version 1.0*, WS-I, Microsoft, IBM, Oracle, DataPower, Sarvega, Nokia Corporation, Layer 7, 2005 <http://www.ws-i.org/Profiles/BasicSecurity/SecurityChallenges-1.0.pdf>
- [Sedukhin05] Sedukhin, I. & Vambenepe, W., *Web Services Distributed Management: Management of Web Services (WSDM-MOWS) 1.0 and Management Using Web Services (MUWS 1.0)*, OASIS, 2005 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm
- [Sheil06] Sheil, H., *Can't we just keep it simple? Use SOAs to add real value, not complexity, to Java enterprise applications*, JavaWorld, 2006
http://www.javaworld.com/javaworld/jw-01-2006/jw-0109-soa_p.html
- [Smith97] Smith, R.E., *Internet Cryptography*, Addison Wesley, 1997
- [Sousa04] Sousa, P.; Pereira, C.M. & Marques, J.A., *Enterprise Architecture Alignment Heuristics*, Microsoft Architects Journal, 2004, 4, 34-39
- [Spewak93] Spewak, S. & Hill, S., *Enterprise Architecture Planning*, John Wiley & Sons, 1993
- [Sun05] Sun, *About Sun Microsystems*, Sun Web Site, 2005
<http://www.sun.com/aboutsun/>
- [Sun06] Sun, *Java Web Services Developer Pack*, Sun Microsystems Web Site, 2006
<http://java.sun.com/webservices/>
- [Sun97] Sun, *Java Remote Method Invocation (RMI)*, Sun Microsystems Web Site, 1997 <http://java.sun.com/products/jdk/rmi/index.jsp>
- [Tanenbaum03] Tanenbaum, A.S. & van Steen, M., *Distributed Systems - principles and paradigms*, Prentice Hall, 2003
- [Thatte03] Thatte, S., *Business Process Execution Language for Web Services Version 1.1*, Microsoft, IBM, Siebel Systems, BEA, SAP, 2003
<http://www.ibm.com/developerworks/library/specification/ws-bpel/>

- [Vecchio05] Vecchio, D.D.; Humphrey, M.; Basney, J. & Nagaratnam, N., *CredEx: User-Centric Credential Management for Grid and Web Services*, Proceedings of the IEEE International Conference on Web Services (ICWS), 2005
http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=32665&arnumber=1530793&count=129&index=25
- [Wahli05] Wahli, U.; Kjaer, T.; Robertson, B.; Satoh, F.; Schneider, F.; Szczeponik, W. & Whyley, C., *Web Services Handbook Development and Deployment*, IBM RedBooks, 2005
- [Wang05] Wang, J.; Vecchio, D.D. & Humphrey, M., *Extending the Security Assertion Markup Language to Support Delegation for Web Services and Grid Services*, Proceedings of the IEEE International Conference on Web Services (ICWS), 2005
- [Wilkes05] Wilkes, L., *The Web Services Protocol Stack*, Web Services Roadmap, 2005
<http://roadmap.cbdiforum.com/reports/protocols/>
- [Woods03] Woods, G. & Gullotta, T., *Web Services Provisioning*, IBM Web Site, 2003
<http://www.ibm.com/developerworks/library/specification/ws-provis/>
- [WSI05] WSI, *Interoperability: Ensuring the Success of Web Services - an overview of WS-I*, WS-I Web Site, 2005 <http://www.ws-i.org/about/Default.aspx>
- [Zachman87] Zachman, J., *Zachman Framework for Enterprise Architecture*, ZIFA, 1987

A. Traduções, siglas e abreviaturas utilizadas

Sigla / abreviatura	Significado
AOP	Aspect Oriented Programming
ASAP	Asynchronous Services Access Protocol
AUTN	Autenticação
AUTR	Autorização
AXIOM	Axis Object Model
CGI	Common Gateway Interface
CONF	Configuração
CORBA	Common Object Request Broker Architecture
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
DIME	Direct Internet Message Encapsulation
ebXML	electronic business XML
EDI	Electronic Data Interchange
EI	Entidade Informacional
ESB	Enterprise Service Bus
HTML	HyperText Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPR	HTTP Reliable
HTTPS	HTTP sobre SSL
IP	Internet Protocol
JAX-B	Java Architecture for XML Data Binding
JAX-RPC	Java APIs for XML-based RPC
JAX-WS	Java API for XML Web Services
JWSDP	Java Web Services Developer Pack
MAC	Message Authentication Code
MEP	Message Exchange Pattern
MIME	Multi-Purpose Internet Mail Extensions
MTOM	Message Transmission Optimization Mechanism
OWS	Office Work System
PN	Processo de Negócio
PROT	Protecção de mensagens
REL	Rights Expression Language
RMI (Java)	Java Remote Method Invocation
RPC	Remote Procedure Call
SAML	Security Assertion Markup Language
SCT	Security Context Token
SMTP	Simple Mail Transfer Protocol
SOA	Service-Oriented Architecture

Sigla / abreviatura	Significado
SOAP	Service Oriented Architecture Protocol , Simple Object Access Protocol
SPEF	SOAP Profile Enabling Framework
SSL	Secure Sockets Layer
STS	Security Token Service, Serviço de Tokens de Segurança
SwA	SOAP with Attachments
TCP	Transport Control Protocol
TLS	Transport Layer Security
TPS	Transaction Processing System
UBL	Universal Business Language
UDDI	Universal Description, Discovery, and Integration
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
UUID	Universal Unique Identifier
WCF	Windows Communication Foundation
WS	Web Service(s)
WS-BPEL	Web Services Business Process Execution Language
WS-CDL	Web Services Choreography Description Language
WS-CTX	Web Service Context
WSDL	Web Service Description Language
WSE	Web Services Enhancements
WS-I	Web Services Interoperability Organization
WSIT	Web Services Interoperability Technology (Project Tango)
WS-MEX	WS-MetadataExchange
WS-RP	Web Services for Remote Portlets
WSS4J	Web Services Security For Java
WS-XL	WS Experience Language
WWW	World Wide Web
XACML	Extensible Access Control Markup Language
XAdES	XML Advanced Electronic Signatures
XKMS	XML Key Management Specification
XML	eXtensible Markup Language
XOP	XML-binary Optimized Packaging
Xpath	XML Path
XSD	XML Schema Definition
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformations
XWSS	XML and Web Services Security

Termo traduzido	Termo original
adaptadores (de invocação)	stubs
agente	principal
arquitectura orientada a serviços	service oriented architecture
asserção	assertion
autoridade de certificação de datas	timestamp authority (TSA)
caminho da mensagem (SOAP)	(SOAP) message path
cenários pré-definidos (chave-na-mão)	turn-key scenarios
central de serviços	enterprise service bus (ESB)
chamada de procedimento remoto	remote procedure call (RPC)
código aberto	open-source
consultas sucessivas	polling
dados em série	streaming data
elemento de configuração à medida	custom assertion
embrulho	wrapping
emissor original (da mensagem SOAP)	original sender (of SOAP message)
entidade	principal
extremidade	endpoint
extremo-a-extremo	end-to-end
fracamente interligados	loosely coupled
identificador uniforme de recurso	uniform resource identifier (URI)
intermediação de confiança	trust brokering
intermediário	broker
intermediário (da mensagem SOAP)	intermediary (of SOAP message)
intersecção de política	policy intersection
junção de política	policy merge
mensagem unidireccional	one-way message
multi-difusão	multi-cast
nó SOAP	SOAP node
normalização de política	policy normalization
notificação	callback
padrão de troca de mensagens	message exchange pattern (MEP)
política	policy
política efectiva	effective policy
ponto-a-ponto	point-to-point
programação orientada a aspectos	aspect oriented programming
publicar, pesquisar, vincular e invocar	publish-find-bind-invoke
qualidade de serviço	quality of service (QoS)
receptor último (da mensagem SOAP)	ultimate receiver (of SOAP message)
registo de actividades	log, logging
regras de codificação SOAP	SOAP encoding rules
sistemas de filas de mensagens	messaging oriented middleware (MOM)

Termo traduzido	Termo original
vinculação configurável	custom binding
vínculo	binding
vínculo de interface	interface binding

B. Organizações de normalização

B.1. IETF

A Internet Engineering Task Force (IETF) é uma comunidade internacional de grande dimensão que junta engenheiros de redes, operadores, fabricantes e investigadores preocupados com a operação e evolução da arquitectura da Internet. Está aberta a quaisquer indivíduos interessados.

O trabalho técnico é feito em grupos de trabalho, organizados por assunto de investigação. O principal meio de comunicação entre os grupos são listas de correio electrónico.

Os documentos Requests for Comments (RFC) são notas técnicas e organizacionais sobre a Internet que se iniciaram em 1969. Estas notas discutem muitos aspectos de redes de computadores, incluindo protocolos, procedimentos, programas e conceitos, bem como notas de reuniões e opiniões.

Normas: HTTP, URI (URL e URN), TCP, IP entre outras

Endereço: <http://www.ietf.org>

B.2. W3C

O World Wide Web Consortium (W3C) foi criado em Outubro de 1994 para potenciar a World Wide Web, desenvolvendo protocolos comuns que promovessem a sua evolução e garantissem a sua interoperabilidade. O W3C é internacional e tem aproximadamente 350 organizações membro incluindo universidades e as principais empresas de tecnologias de informação.

Normas: HTML, XML, XML Schema, XML Encryption, XML Signature, XML Key Management, XSL, XSLT, WSDL, SOAP entre outras

Endereço: <http://www.w3.org/>

B.3. OASIS

A Organization for the Advancement of Structured Information Standards (OASIS) é um consórcio internacional sem fins lucrativos criado em 1993 para conduzir o desenvolvimento, convergência e adopção de normas de negócio electrónico. Parte significativa do seu esforço tem sido concentrado nas normas de Web Services. A OASIS tem mais de 4000 participantes representando acima de 600 organizações e membros individuais em 100 países.

Normas: ebXML, XACML, SAML, UDDI, UBL, WS-Reliability, WS-Security 2004

Endereço: <http://www.oasis-open.org>

B.4. WS-I

A Web Services Interoperability Organization (WS-I) é uma organização criada por iniciativa da indústria informática para promover a interoperabilidade de Web Services entre diferentes plataformas, aplicações e linguagens de programação.

A WS-I disponibiliza recomendações e recursos para apoio ao desenvolvimento de soluções com interoperabilidade, definindo perfis que agregam conjuntos de normas sobre aspectos específicos da arquitectura (por exemplo, sobre segurança).

Endereço: <http://www.ws-i.org>

B.5. JCP

A Java Community Process (JCP) é a organização internacional e aberta, criada em 1998, que suporta o processo de desenvolvimento e revisão de especificações da tecnologia Java designadas por Java Specification Requests (JSR), juntamente com a disponibilização de implementações de referência e de ferramentas de teste.

A empresa norte-americana Sun Microsystems é o membro principal do comité executivo da JCP, que é responsável por coordenar os diferentes trabalhos em curso na organização.

Endereço: <http://www.jcp.org>

C. Políticas WS-Policy

Neste anexo são apresentados os seguintes itens complementares: terminologia associada a políticas, políticas na forma compacta, políticas vazia e nula e exemplos das operações normalização, intersecção e junção. Referências: [Schlimmer06], [Nolan04], [Samaranayake06].

C.1. Terminologia

Uma *política* WS-Policy é uma colecção de alternativas. Uma *asserção* representa individualmente um requisito, uma capacidade ou uma outra propriedade ou comportamento. Um *tipo de asserção* representa uma classe de asserções e implica um esquema para a asserção e para a semântica própria da asserção. Um *vocabulário* é o conjunto de tipos de asserções usados na política. O *sujeito* da política é uma entidade (extremidade, mensagem, recurso) ao qual a política está associada. Um *âmbito* da política é uma colecção de sujeitos aos quais a política se pode aplicar. Um *anexo* de política é um mecanismo para associar uma política a um ou mais âmbitos.

Uma asserção é suportada por um cliente se e só se o cliente satisfaz o requisito correspondente. Uma alternativa de política é suportada pelo cliente se e só se o cliente suporta todas as asserções da política. Uma política é suportada pelo cliente se e só se o cliente suporta pelo menos uma das alternativas da política.

C.2. Políticas compactas

Na forma normal as políticas podem ficar extensas. Por isso, a norma define três formas de expressar políticas de forma mais compacta:

- Atributo de opcionalidade – permite representar compactamente duas alternativas de política: uma com a asserção opcional e outra sem ela;
- Políticas aninhadas – definições de políticas dentro de outras definições de políticas;
- Inclusão de políticas por referência – obtenção de políticas definidas em documentos XML externos.

As políticas compactas são sempre reduzíveis à forma normal.

C.3. Política vazia e política nula

Existem dois casos especiais de políticas: A política vazia e a política nula.

A *política vazia* tem o seguinte aspecto:

```
<wsp:Policy ... >
  <wsp:ExactlyOne>
    <wsp:All/>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Como a única alternativa All está vazia, isto indica que não são requeridas nenhuma asserções, ou seja, que não é necessário fazer nada mais para interagir com o serviço.

A *política nula* tem o seguinte aspecto:

```
<wsp:Policy ... >
  <wsp:ExactlyOne/>
</wsp:Policy>
```

Como não existem alternativas contidas dentro de ExactlyOne, a política nula indica que não existe nenhuma alternativa aceitável para interagir com o serviço.

C.4. Normalização

A normalização é o processo de converter uma política para a forma normal. Esta conversão preserva o significado lógico da política original. O elemento `wsp:Policy` é equivalente ao elemento `wsp:All`.

Exemplo de política inicial não normalizada:

```
<wsp:Policy ... >
  <wsp:All>
    <wsp:ExactlyOne>
      <nsSecurityAssertion wsp:Optional="true"/>
      <nsReliableMessagingAssertion/>
    </wsp:ExactlyOne>
    <nsTransactionAssertion/>
    <nsAuditAssertion/>
  </wsp:All>
</wsp:Policy>
```

Expandindo o atributo `wsp:Optional` são criadas duas entradas para cada ocorrência da opcionalidade, uma onde existe, outra onde não existe, tendo o `ExactlyOne` para as agrupar como alternativas.

```
<nsSecurityAssertion wsp:Optional="true"/>
```

Fica como:

```
<wsp:ExactlyOne>
  <nsSecurityAssertion/>
  <wsp:All/>
</wsp:ExactlyOne>
```

Ou seja, a política fica:

```
<wsp:All>
  <wsp:ExactlyOne>
    <wsp:ExactlyOne>
      <nsSecurityAssertion/>
```

```

        <wsp:All/>
    </wsp:ExactlyOne>
    <nsReliableMessagingAssertion/>
</wsp:ExactlyOne>
<nsTransactionAssertion/>
<nsAuditAssertion/>
</wsp:All>

```

Colapsando os operadores de política, fica-se apenas com um único `wsp:ExactlyOne` contendo as asserções dentro de operadores `wsp:All`.

```

<wsp:All>
  <wsp:ExactlyOne>
    <nsSecurityAssertion/>
    <wsp:All/>
    <nsReliableMessagingAssertion/>
  </wsp:ExactlyOne>
  <nsTransactionAssertion/>
  <nsAuditAssertion/>
</wsp:All>

```

Repetindo o processo de colapso chega-se à situação em que se unifica apenas com um único `wsp:ExactlyOne`. Para inverter o `wsp:All` e `wsp:ExactlyOne`, usa-se a propriedade associativa. As asserções de `wsp:ExactlyOne` são combinadas com as asserções do `wsp:All` de fora.

```

<wsp:ExactlyOne>
  <wsp:All>
    <nsSecurityAssertion/>
    <nsTransactionAssertion/>
    <nsAuditAssertion/>
  </wsp:All>
  <wsp:All>
    <wsp:All/>
    <nsTransactionAssertion/>
    <nsAuditAssertion/>
  </wsp:All>
  <wsp:All>
    <nsReliableMessagingAssertion/>
    <nsTransactionAssertion/>
    <nsAuditAssertion/>
  </wsp:All>
</wsp:ExactlyOne>

```

Finalmente, para chegar à forma normal, podem-se limpar todos os `wsp:All` vazios. A política normal tem três alternativas:

```

<wsp:ExactlyOne>
  <wsp:All>
    <nsSecurityAssertion/>
    <nsTransactionAssertion/>
    <nsAuditAssertion/>
  </wsp:All>
  <wsp:All>
    <nsTransactionAssertion/>
    <nsAuditAssertion/>
  </wsp:All>
  <wsp:All>
    <nsReliableMessagingAssertion/>
    <nsTransactionAssertion/>
    <nsAuditAssertion/>
  </wsp:All>
</wsp:ExactlyOne>

```

Se existir uma política que contém outras políticas aninhadas, esta hierarquia mantém-se, no entanto, em cada nível apenas é apresentada uma alternativa.

C.5. Intersecção

A intersecção é o processo de isolar as alternativas de política de duas políticas que duas entidades compreendem. Isto é útil no cenário cliente-servidor. Se a intersecção for vazia, então o cliente não consegue interagir com o serviço.

Uma dificuldade é conseguir que o motor genérico de políticas determine se duas asserções são iguais. O significado só é completamente conhecido pelo domínio que as define. É portanto necessário conhecimento de domínio para completar o processo. O processo inicia-se com a normalização da políticas. Depois as políticas são analisadas uma alternativa de cada vez. A intenção da primeira fase é eliminar as alternativas claramente diferentes. Isto é feito examinando os vocabulários das duas alternativas. Se os vocabulários não forem iguais, então as alternativas são claramente diferentes e podem ser descartadas da intersecção.

Política A normalizada:

```
<wsp:Policy wsu:Id="Provider_Policy"...>
  <wsp:ExactlyOne>
    <wsp:All>
      <nsSecurityAssertion level="high"/>
      <nsReliableMessagingAssertion/>
    </wsp:All>
    <wsp:All>
      <nsSecurityAssertion level="medium"/>
      <nsTransactionAssertion/>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Política B normalizada:

```
<wsp:Policy wsu:Id="Requester_Policy"...>
  <wsp:ExactlyOne>
    <wsp:All>
      <nsSecurityAssertion/>
      <nsReliableMessagingAssertion timeout="100"/>
      <nsReliableMessagingAssertion retries="3"/>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

O processo de intersecção pega em cada alternativa All. Apenas os nomes das asserções são comparados. Isto significa que a alternativa única da política B apenas emparelha com a primeira alternativa da política A (ambas têm os mesmos nomes). Estas duas alternativas são combinadas para produzir a intersecção.

```
<wsp:Policy ... >
  <wsp:All>
    <nsSecurityAssertion level="high"/>
    <nsSecurityAssertion/>
    <nsReliableMessagingAssertion timeout="100"/>
    <nsReliableMessagingAssertion retries="3"/>
    <nsReliableMessagingAssertion/>
  </wsp:All>
</wsp:Policy>
```


Cabe agora ao motor de políticas interpretar esta alternativa e fundir as asserções com nomes idênticos em asserções que tenham significado para os módulos de processamento. O resultado poderia ser algo como:

```
<wsp:Policy ... >
  <wsp:All>
    <nsSecurityAssertion level="high"/>
    <nsReliableMessagingAssertion timeout="100" retries="3"/>
  </wsp:All>
</wsp:Policy>
```

Seria também nesta fase que seriam detectadas eventuais contradições entre as políticas.

C.6. Junção

A junção de política é o processo de criação de uma única política a partir de duas políticas tendo por base todas as alternativas com as quais as entidades envolvidas concordam. É uma operação útil no servidor, para juntar as política do serviço com a do servidor em que se executa. À política resultante da junção é dada a designação de *política efectiva*.

O processo de junção é realizado entre políticas que tinham sido convertidas para a forma normal. As alternativas de cada política são combinadas para formar a nova alternativa de junção. O processo de combinação pega numa alternativa de uma política e combina-a com todas as alternativas da segunda e assim para as restantes da primeira.

Política P1 normalizada:

```
<wsp:Policy wsu:id="P1"... >
  <wsp:ExactlyOne>
    <wsp:All>
      <nsSecurityAssertion/>
    </wsp:All>
    <wsp:All>
      <nsReliableMessagingAssertion/>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Política P2 normalizada:

```
<wsp:Policy wsu:id="P2"... >
  <wsp:ExactlyOne>
    <wsp:All>
      <nsTransactionAssertion/>
    </wsp:All>
    <wsp:All>
      <nsAuditAssertion/>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Para obter a política de junção, partindo das políticas normalizadas, cada alternativa de P1 é combinada com cada alternativa de P2.

```
<wsp:Policy wsu:Id="P1_Merged_with_P2"...>
  <wsp:ExactlyOne>
    <wsp:All>
      <nsSecurityAssertion/>
      <nsTransactionAssertion/>
    </wsp:All>
    <wsp:All>
      <nsSecurityAssertion/>
      <nsAuditAssertion/>
    </wsp:All>
    <wsp:All>
      <nsReliableMessagingAssertion/>
      <nsTransactionAssertion/>
    </wsp:All>
    <wsp:All>
      <nsReliableMessagingAssertion/>
      <nsAuditAssertion/>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

É responsabilidade do motor de políticas determinar o significado de alternativas que resultem na duplicação de asserções. Existe também o potencial de se criar um grande número de alternativas, especialmente, se o atributo opcional for muito usado.

D. Processo de compra e venda de imóvel

Este anexo descreve em detalhe a metodologia e os resultados obtidos para a modelação do processo de negócio de compra e venda de imóvel, cujo resumo foi apresentado na dissertação.

D.1. Metodologia

A metodologia adoptada para a modelação de processos consistiu nos seguintes passos:

1. Identificação e definição dos *actores* envolvidos;
2. Identificação dos *processos de negócio*:
 - 2.1. Levantamento detalhado de processos para cada actor;
 - 2.2. Combinação das diferentes perspectivas dos actores, utilizando os objectivos dos processos como critério agregador;
 - 2.3. Ajuste do nível de detalhe, tendo em consideração o objectivo da modelação;
 - 2.4. Definição dos processos de negócio;
3. Identificação das *entidades informacionais*:
 - 3.1. Listagem conjunta de entradas e saídas de um grupo de processos;
 - 3.2. Agregação dos itens de informação por afinidade;
 - 3.3. Definição de entidades candidatas, com indicação do respectivo dono;
 - 3.4. Agregação ou separação de entidades candidatas;
 - 3.5. Ajuste do nível de detalhe das entidades e seus atributos, tendo em conta o objectivo da modelação;
 - 3.6. Definição das entidades informacionais;
4. Verificação da coerência global entre os processos e as entidades informacionais.

A opção de capturar as actividades através de uma *visão orientada aos processos* tem como principal vantagem em relação a uma visão orientada às funções internas da organização, partir de um ponto de vista externo identificando situações a melhorar que acrescentam valor efectivo aos clientes.

Os actores são papéis desempenhados pelos intervenientes nos processos de negócio. Ao usar actores para levantar o processo conseguem-se combinar diferentes perspectivas sobre a mesma realidade complexa.

Os processos de negócio estão estruturados hierarquicamente por objectivos, pelo que um sub-processo corresponde a um sub objectivo do objectivo do processo pai.

As entidades informacionais identificam e definem a informação fundamental para o negócio de forma independente das aplicações, sendo uma condição necessária mas não suficiente para uma gestão correcta da informação ao nível de toda a organização.

D.2. Actores

A.1	Vendedor
<i>Descrição sumária</i>	Pessoa em cujo nome está legalmente registada a propriedade do imóvel, que agora pretende vender.
A.2	Comprador
<i>Descrição sumária</i>	Pessoa que pretende comprar o imóvel.
A.3	Mediador imobiliário
<i>Descrição sumária</i>	Pessoa ou empresa que desempenha a actividade comercial sob contrato escrito onde se obriga, mediante o recebimento de um determinado preço, a conseguir um interessado para a compra e venda de imóveis, prestando os serviços necessários à realização dessas operações.
A.4	Notário
<i>Descrição sumária</i>	Entidade pública ou privada dependente do Ministério da Justiça (Direcção Geral e Registos de Notariado) que tem por missão realizar certos actos e contratos ou verificar as condições legais dos mesmos, nomeadamente a compra e venda de imóveis e a constituição de hipoteca, as quais estão sujeitas a escritura pública.
A.5	Conservatória do Registo Predial
<i>Descrição sumária</i>	Serviço público, dependente do Ministério da Justiça (Direcção Geral e Registos de Notariado), onde é registada a informação essencial relativa aos bens imóveis (urbanos e rústicos), designadamente a sua localização e confrontações, a sua composição e a identificação dos sucessivos proprietários. Existem em todos os concelhos do país, sendo várias nas grandes cidades.
A.6	Finanças
<i>Descrição sumária</i>	Serviço público responsável pela cobrança de impostos.
A.7	Câmara Municipal
<i>Descrição sumária</i>	Órgão executivo de um município (concelho), responsável pela gestão e licenciamento da sua circunscrição territorial.
A.8	Tribunal
<i>Descrição sumária</i>	Órgão de soberania com competência para administrar a justiça.
A.9	Banco
<i>Descrição sumária</i>	Instituição cuja actividade consiste na realização de operações financeiras e na prestação de serviços financeiros, dos quais, os mais comuns são a concessão de crédito e o recebimento de depósitos dos clientes, que remunera.
A.10	Seguradora
<i>Descrição sumária</i>	Entidade legalmente autorizada a exercer a actividade seguradora e que subscreve, com o tomador, o contrato de seguro

D.3. Processos de negócio

PN.1	Comprar e vender imóvel
<i>Actores</i>	Vendedor, Comprador, Notário, Conservatória do Registo Predial, Banco, Mediador Imobiliário, Finanças, Câmara Municipal
<i>Descrição sumária</i>	<p>O <i>Vendedor</i> coloca o imóvel à venda através de um <i>Mediador Imobiliário</i>.</p> <p>O <i>Comprador</i> interessa-se pelo imóvel, confere a sua legalidade e decide a compra.</p> <p>O <i>Vendedor</i> e o <i>Comprador</i> assinam um contrato promessa de compra e venda do imóvel, no <i>Notário</i>.</p> <p>O <i>Comprador</i> pede crédito ao <i>Banco</i>.</p> <p>O <i>Comprador</i> efectua na <i>Conservatória do Registo Predial</i> o registo provisório do imóvel e da hipoteca.</p> <p>O <i>Comprador</i> paga o IMT (Imposto de Transmissões Onerosas de Imóveis) nas <i>Finanças</i>.</p> <p>O <i>Vendedor</i>, o <i>Comprador</i> e o <i>Banco</i> celebram no <i>Notário</i> a escritura de compra e venda e de hipoteca.</p> <p>O <i>Comprador</i> pede nas <i>Finanças</i> a isenção do IMI (Imposto Municipal sobre Imóveis), caso satisfaça as condições legais para tal.</p> <p>O <i>Banco</i> efectua a conversão definitiva dos registos provisórios do imóvel.</p> <p>Após a liquidação total do empréstimo, o <i>Banco</i> renuncia à hipoteca e o <i>Comprador</i> deve contactar a <i>Conservatória do Registo Predial</i> para cancelar o registo hipotecário.</p>
<i>Notas</i>	<p>A participação do <i>Mediador Imobiliário</i> é opcional, podendo o <i>Vendedor</i> e o <i>Comprador</i> interagir directamente.</p> <p>A participação do <i>Banco</i> é também opcional.</p>
<i>Entradas</i>	Imóvel
<i>Saídas</i>	Imóvel com novo proprietário
<i>Requisitos</i>	Os actos realizados devem ser autênticos e legalmente aceites.

PN.1.1	Colocar imóvel à venda
<i>Actores</i>	Vendedor, Mediador Imobiliário
<i>Descrição sumária</i>	<p>O <i>Vendedor</i> coloca o imóvel à venda através de um <i>Mediador Imobiliário</i>, assinando um contrato para o efeito.</p> <p>O <i>Vendedor</i> define as condições para a venda do imóvel.</p> <p>O <i>Mediador Imobiliário</i> divulga as condições de venda e procura interessados, publicitando e mostrando o imóvel até conseguir um interessado na sua compra.</p>
<i>Notas</i>	O <i>Vendedor</i> pode optar por não recorrer aos serviços de um <i>Mediador Imobiliário</i> .
<i>Entradas</i>	Imóvel
<i>Saídas</i>	Contrato de mediação imobiliária. Condições de venda do imóvel.
<i>Requisitos</i>	<p>O contrato de mediação deve ser autêntico e legalmente aceite.</p> <p>A divulgação das condições de venda deve ser fidedigna.</p> <p>O <i>Comprador</i> deve poder confirmar que o <i>Mediador</i> está legalmente autorizado pelo <i>Vendedor</i>.</p>

PN.1.2	Verificar estado legal do imóvel à venda
<i>Actores</i>	Comprador, Conservatória do Registo Predial, Finanças, Câmara Municipal, Tribunal
<i>Descrição sumária</i>	<p>Antes de iniciar o processo de compra de imóvel, o <i>Comprador</i> deverá ter alguns cuidados legais na escolha, consultando um conjunto de entidades públicas para verificar se tudo está bem.</p> <p>O <i>Comprador</i> deverá confirmar com a <i>Conservatória do Registo Predial</i> da área do imóvel, se: o vendedor é o verdadeiro proprietário, isto é, se tem o imóvel registado em seu nome; não existem hipotecas ou penhoras a favor de terceiros; o imóvel ou fracção não está sujeito a qualquer usufruto a favor de terceiros.</p> <p>O <i>Comprador</i> deverá confirmar com as <i>Finanças</i> da área do imóvel, se: o imóvel se encontra devoluto, isto é, livre de herdeiros com direito de preferência ou inquilinos; o IMI está em dia, no caso de compra em 2ª mão, ou se o imóvel responde pelo pagamento deste imposto.</p> <p>O <i>Comprador</i> deverá confirmar com a <i>Câmara Municipal</i>, se: já foram emitidas as Licenças de Construção e de Habitação; no caso de aquisição de terreno para construção de futura habitação, se o terreno se situa em zona autorizada para construções urbanas, isto é, se foi emitido Alvará de Loteamento; caso o terreno tenha autorização para construção, quais as características da habitação que pode construir.</p>

	O <i>Comprador</i> deverá confirmar com o <i>Tribunal</i> se: o <i>Vendedor</i> não está em situação de falência, nem corre contra ele prática de crime doloso que torne ineficaz a alienação de bens próprios; decidir comprar o imóvel em planta, averiguar se os promotores têm capacidade e idoneidade para levar o empreendimento a bom termo.
<i>Notas</i>	Cada uma das verificações é opcional, correndo o <i>Comprador</i> o risco de investir num negócio danoso.
<i>Entradas</i>	Imóvel
<i>Saídas</i>	Confirmação do registo de propriedade do imóvel. Confirmação de estado devoluto do imóvel. Confirmação de IMI em dia. Confirmação de licença de utilização. Confirmação de situação financeira e legal do <i>Vendedor</i> .
<i>Requisitos</i>	Cada uma das confirmações tem que ser autêntica.

PN.1.3	Pedir crédito ao banco
<i>Actores</i>	Comprador, Banco
<i>Descrição sumária</i>	O <i>Comprador</i> vai pedir crédito ao <i>Banco</i> . Com base no valor do empréstimo pretendido e em dados elementares relativos ao imóvel a adquirir e ao nível de rendimento do <i>Comprador</i> , o <i>Banco</i> dá uma resposta de princípio acerca da viabilidade do pedido. De seguida, o <i>Banco</i> vai solicitar um conjunto de documentação e procederá à avaliação do imóvel. Posteriormente, comunica a decisão definitiva sobre a concessão e as condições do empréstimo.
<i>Notas</i>	
<i>Entradas</i>	Imóvel. Identificação, património e rendimentos do Comprador.
<i>Saídas</i>	Avaliação do imóvel. Proposta de crédito.
<i>Requisitos</i>	A integridade da proposta de crédito deve ser assegurada (montantes e datas).

PN.1.4	Fazer contrato promessa de compra e venda
<i>Actores</i>	Vendedor, Comprador, Notário
<i>Descrição sumária</i>	O <i>Vendedor</i> propõe um contrato promessa ao <i>Comprador</i> . O <i>Comprador</i> analisa as cláusulas do contrato e decide. Caso exista acordo das partes, o <i>Comprador</i> e o <i>Vendedor</i> assinam o contrato promessa de compra e venda, com assinaturas reconhecidas por <i>Notário</i> . O <i>Comprador</i> pode exigir comprovativos do estado legal do imóvel para a assinatura do contrato, como a licença de utilização, por exemplo.
<i>Notas</i>	
<i>Entradas</i>	Imóvel. Identificação do <i>Vendedor</i> e do <i>Comprador</i> . Modelo de contrato promessa de compra e venda, com todas as cláusulas explicitadas (preço e forma de pagamento; valor do sinal; prazo máximo de escritura; juros por atrasos; condições como a dependência da concessão de crédito; etc.)
<i>Saídas</i>	Contrato promessa de compra e venda assinado pelo <i>Vendedor</i> e pelo <i>Comprador</i> .
<i>Requisitos</i>	A integridade da informação de todo o contrato tem que ser assegurada. As assinaturas do <i>Vendedor</i> e do <i>Comprador</i> têm que ser autênticas. O não-repúdio das assinaturas tem que ser garantido (por exemplo, para permitir accionar as cláusulas de não cumprimento).

PN.1.5	Proceder aos registos provisórios
<i>Actores</i>	Comprador, Vendedor, Banco, Conservatória do Registo Predial, Finanças
<i>Descrição sumária</i>	Após a assinatura do contrato-promessa de compra e venda e após a autorização do empréstimo, o <i>Comprador</i> terá de proceder aos registos provisórios de aquisição e de hipoteca, na <i>Conservatória do Registo Predial</i> da área do imóvel. O registo de hipoteca deverá ser efectuado nos termos indicados na minuta a fornecer pelo <i>Banco</i> para esse efeito. Para além destes registos, o comprador deverá também solicitar, na <i>Conservatória do Registo Predial</i> , uma Certidão de Teor de todos os registos em vigor relativos ao imóvel a adquirir. Para qualquer destes fins, terá de apresentar a Caderneta Predial das <i>Finanças</i> . O <i>Vendedor</i> tem que autorizar o pedido de registo provisório.

<i>Notas</i>	
<i>Entradas</i>	Caderneta Predial do imóvel. Minuta do registo de hipoteca.
<i>Saídas</i>	Registo provisório de aquisição de imóvel. Registo provisório de hipoteca de imóvel. Certidão de Teor.
<i>Requisitos</i>	A integridade e autenticidade dos registos provisórios tem que ser garantida.

PN.1.6	Pagar o Imposto sobre Transmissões Onerosas de Imóveis (IMT)
<i>Actores</i>	Comprador, Finanças
<i>Descrição sumária</i>	O <i>Comprador</i> efectua o pagamento do IMT nas <i>Finanças</i> .
<i>Notas</i>	Existem casos de isenção deste imposto.
<i>Entradas</i>	Imóvel. Declaração para efeitos de pagamento do imposto.
<i>Saídas</i>	Comprovativo do pagamento do IMT.
<i>Requisitos</i>	A integridade e autenticidade do comprovativo tem que ser assegurada.

PN.1.7	Efectuar seguro do imóvel
<i>Actores</i>	Comprador, Seguradora
<i>Descrição sumária</i>	O <i>Comprador</i> efectua com a <i>Seguradora</i> um contrato de seguro para o imóvel.
<i>Notas</i>	As coberturas do seguro de imóvel são variáveis, mas existem mínimos previstos na lei.
<i>Entradas</i>	Imóvel. Identificação do <i>Comprador</i> . Proposta de contrato de seguro de imóvel.
<i>Saídas</i>	Contrato do seguro de imóvel. Apólice do seguro de imóvel.
<i>Requisitos</i>	A integridade e autenticidade do contrato e da apólice tem que ser assegurada.

PN.1.8	Efectuar seguro pessoal de vida
<i>Actores</i>	Comprador, Seguradora
<i>Descrição sumária</i>	O <i>Comprador</i> pede à <i>Seguradora</i> a realização de um seguro de vida. A <i>Seguradora</i> pode requerer ao <i>Comprador</i> testes médicos. O <i>Comprador</i> efectua com a <i>Seguradora</i> um contrato de seguro de vida.
<i>Notas</i>	O seguro de vida pessoal pode ser exigido pelo Banco ao <i>Comprador</i> na concessão de crédito.
<i>Entradas</i>	Identificação do <i>Comprador</i> . Resultado dos testes médicos. Proposta de contrato de seguro de vida.
<i>Saídas</i>	Contrato do seguro de vida. Apólice do seguro de vida.
<i>Requisitos</i>	A integridade e autenticidade do contrato e da apólice tem que ser assegurada.

PN.1.9	Celebrar a escritura de compra e venda e de hipoteca
<i>Actores</i>	Comprador, Vendedor, Banco, Notário
<i>Descrição sumária</i>	O <i>Vendedor</i> , o <i>Comprador</i> e o <i>Banco</i> celebram no <i>Notário</i> a escritura de compra e venda e de hipoteca. No momento da escritura, têm lugar dois contratos distintos. O primeiro é o contrato de compra e venda, através do qual o <i>Comprador</i> passa a ser o proprietário juridicamente reconhecido do imóvel. O outro contrato – de mútuo com hipoteca – é celebrado entre o <i>Comprador</i> (devedor) e o <i>Banco</i> (credor) e estipula tudo o que se relaciona com a dívida contraída (o seu valor, taxa de juro, prazos de pagamento, etc.). Após a celebração deste último contrato, o <i>Banco</i> liberta o montante autorizado, permitindo ao <i>Comprador</i> pagar ao <i>Vendedor</i> a parcela do valor da transacção que ainda faltava pagar.
<i>Notas</i>	A realização da hipoteca e a participação do <i>Banco</i> apenas existem se tiver sido pedido um empréstimo. A escritura descreve a compra e venda juntamente com a hipoteca, sendo o mesmo

	documento assinado por todas as partes.
<i>Entradas</i>	Identificação do <i>Vendedor</i> e do <i>Comprador</i> . Contrato promessa de compra e venda do imóvel. Certidão de teor de todos os registos em vigor (onde já constem o registo provisório de aquisição e o de hipoteca). Caderneta Predial urbana ou certidão do pedido de inscrição na matriz passada pelas repartições de <i>Finanças</i> . Licença de utilização ou prova de que a mesma foi requerida à <i>Câmara Municipal</i> . Apólices do seguro do imóvel e de vida (quando obrigatório). Documento comprovativo do pagamento do IMT.
<i>Saídas</i>	Escritura de compra e venda do imóvel. Escritura de hipoteca do imóvel.
<i>Requisitos</i>	A integridade da informação de toda a escritura tem que ser assegurada. As assinaturas do <i>Vendedor</i> , do <i>Comprador</i> e do <i>Banco</i> têm que ser autênticas. O não-repúdio das assinaturas tem que ser garantido.

PN.1.10	Pedir a isenção de Imposto Municipal sobre Imóveis (IMI)
<i>Actores</i>	Comprador, Finanças, Câmara Municipal
<i>Descrição sumária</i>	Dentro do prazo de 60 dias após a realização da escritura, o <i>Comprador</i> requer nas <i>Finanças</i> a isenção do IMI. A isenção é posteriormente comunicada à <i>Câmara Municipal</i> .
<i>Notas</i>	Existem condições determinadas na lei para a concessão da isenção do IMI.
<i>Entradas</i>	Imóvel. Caderneta Predial. Escritura de compra e venda de imóvel. Declaração de se tratar de habitação própria.
<i>Saídas</i>	Caderneta Predial actualizada. Comprovativo de isenção de IMI.
<i>Requisitos</i>	A actualização da caderneta deve respeitar os registos anteriores nela efectuados.

PN.1.11	Converter em definitivo os registos provisórios
<i>Actores</i>	Comprador, Banco, Conservatória do Registo Predial
<i>Descrição sumária</i>	Depois da escritura de compra e venda e de hipoteca, o <i>Banco</i> procede à conversão em definitivo dos registos provisórios, junto da <i>Conservatória do Registo Predial</i> .
<i>Notas</i>	O processo terá que ser efectuado pelo <i>Comprador</i> , no caso de não existir nenhum empréstimo.
<i>Entradas</i>	Escritura de compra e venda do imóvel. Escritura de hipoteca do imóvel (se houver empréstimo).
<i>Saídas</i>	Certidão de Teor.
<i>Requisitos</i>	A integridade e autenticidade dos registos tem que ser garantida.

PN.1.12	Cancelar a hipoteca
<i>Actores</i>	Comprador, Banco, Conservatória do Registo Predial
<i>Descrição sumária</i>	Após a liquidação total do empréstimo, o <i>Banco</i> emite um documento em que renuncia à hipoteca que foi constituída a seu favor ("distrata da hipoteca") e em que declara liquidada a dívida, deixando o <i>Banco</i> de exercer quaisquer direitos sobre a casa. Este documento deverá ser entregue pelo proprietário na <i>Conservatória do Registo Predial</i> , para efeitos de cancelamento do registo hipotecário. O cancelamento da hipoteca constitui o último passo que envolve a compra de uma habitação com recurso ao crédito.
<i>Entradas</i>	Escritura de compra e venda do imóvel. Escritura de hipoteca do imóvel. Distrata da hipoteca.
<i>Saídas</i>	Certidão de Teor.
<i>Requisitos</i>	A integridade da informação tem que ser assegurada. As assinaturas do <i>Vendedor</i> , do <i>Comprador</i> e do <i>Banco</i> têm que ser autênticas. O não-repúdio das assinaturas tem que ser garantido.

D.4. Entidades informacionais

EI.1	Imóvel
<i>Dono</i>	Finanças, Conservatória do Registo Predial, Câmara Municipal
<i>Gestor</i>	Finanças, Conservatória do Registo Predial, Câmara Municipal
<i>Descrição</i>	Prédio rústico ou urbano, água, árvore, arbusto e frutos naturais enquanto estiverem ligados ao solo, os direitos inerentes a estas coisas e as partes integrantes dos prédios rústicos e urbanos que estejam ligadas materialmente com carácter de permanência.
<i>Atributos identificadores</i>	Identificação imóvel (Concelho, Freguesia, Rua, Número, Fração).

EI.1.1	Imóvel – Caderneta predial
<i>Dono</i>	Finanças
<i>Gestor</i>	Finanças
<i>Descrição</i>	Documento que funciona como uma espécie de “bilhete de identidade” do imóvel. É emitido pelo Serviço de Finanças e comprova a sua inscrição na matriz, identifica a sua localização, a composição, a área, o proprietário e o valor patrimonial tributável. Sempre que seja necessário efectuar um registo na Conservatória do Registo Predial é solicitada a apresentação da caderneta predial actualizada pelo Serviço de Finanças.
<i>Atributos</i>	Referência Imóvel. Proprietário. Valor patrimonial.

EI.1.2	Imóvel – Certidão de teor
<i>Dono</i>	Conservatória do Registo Predial
<i>Gestor</i>	Conservatória do Registo Predial
<i>Descrição</i>	Documento usualmente designado por “certidão de teor” e emitido pela Conservatória do Registo Predial que certifica todos os registos efectuados em relação ao imóvel: localização, composição, proprietários, ónus, transmissões, etc. O registo de aquisição é uma anotação na Conservatória do Registo Predial da aquisição de determinado imóvel e respectiva transmissão de propriedade. O registo de hipoteca é uma anotação na Conservatória do Registo Predial da constituição de uma hipoteca sobre um imóvel. Para obtenção de um empréstimo de habitação, efectua-se previamente um registo provisório de hipoteca que é convertido em definitivo após a assinatura do contrato de mútuo e hipoteca. O registo provisório, se entretanto não for convertido, caduca ao fim de seis meses.
<i>Atributos</i>	Referência Imóvel. Proprietário. Registo provisório de aquisição. Registo provisório de hipoteca. Registo definitivo de aquisição. Registo definitivo de distrate de hipoteca.

EI.1.3	Imóvel – Licença de utilização
<i>Dono</i>	Câmara Municipal
<i>Gestor</i>	Câmara Municipal
<i>Descrição</i>	Documento emitido pela Câmara Municipal da área onde se situa o imóvel, que atesta a habitabilidade do mesmo, depois de verificado o cumprimento das condições legais exigíveis para a sua emissão.
<i>Atributos</i>	Referência Imóvel. Planta de localização. Licença de utilização. Alvará de licença de construção. Requerimento de licença de utilização.

EI.2	Vendedor
<i>Dono</i>	Tribunal, Finanças
<i>Gestor</i>	Tribunal, Finanças

<i>Descrição</i>	Informação associada à pessoa em cujo nome está legalmente registada a propriedade do imóvel, que agora pretende vender.
<i>Atributos</i>	Identificação civil (bilhete de identidade - BI). Identificação fiscal (número de identificação fiscal - NIF). Situação legal. Situação financeira.

EI.2.1	Vendedor - Situação legal
<i>Dono</i>	Tribunal
<i>Gestor</i>	Tribunal
<i>Descrição</i>	Dados legais sobre a prática de crimes dolosos que tornem ineficaz a alienação de bens próprios.
<i>Atributos</i>	Referência Vendedor. Descrição da situação legal.

EI.2.2	Vendedor - Situação financeira
<i>Dono</i>	Finanças
<i>Gestor</i>	Finanças
<i>Descrição</i>	Dados descritivos da situação financeira do Vendedor. que referem se está em situação de falência.
<i>Atributos</i>	Referência Vendedor. Dados da situação financeira (bens, dívidas).

EI.3	Comprador
<i>Dono</i>	Finanças, Seguradora
<i>Gestor</i>	Finanças, Seguradora
<i>Descrição</i>	Informação da pessoa que pretende comprar o imóvel.
<i>Atributos identificadores</i>	Identificação civil (bilhete de identidade - BI) Identificação fiscal (número de identificação fiscal - NIF)

EI.3.1	Comprador – Património e rendimentos
<i>Dono</i>	Finanças
<i>Gestor</i>	Finanças
<i>Descrição</i>	Inventariação do património do Comprador.
<i>Atributos</i>	Referência Comprador. Património. Rendimentos

EI.3.2	Comprador - Testes médicos
<i>Dono</i>	Seguradora
<i>Gestor</i>	Seguradora
<i>Descrição</i>	Dados de saúde relevantes para a contratação de seguros de vida necessários à concessão de crédito.
<i>Atributos</i>	Referência Comprador. Dados médicos.

EI.4	Notário
<i>Dono</i>	(Direcção Geral e Registos de Notariado)
<i>Gestor</i>	(Direcção Geral e Registos de Notariado)
<i>Descrição</i>	Informação do notário.
<i>Atributos</i>	Identificação fiscal (número de identificação fiscal - NIF). Autorização do Ministério da Justiça (Direcção Geral e Registos de Notariado). Localização (distrito, concelho).

EI.5	Mediador Imobiliário
<i>Dono</i>	(Instituto dos Mercados de Obras Públicas e Particulares e do Imobiliário)
<i>Gestor</i>	(Instituto dos Mercados de Obras Públicas e Particulares e do Imobiliário)
<i>Descrição</i>	Informação do mediador imobiliário.

<i>Atributos</i>	Identificação fiscal (número de identificação fiscal - NIF). Licenciamento do Instituto dos Mercados de Obras Públicas e Particulares e do Imobiliário (IMOPPI), sob tutela do Ministério das Obras Públicas, Transportes e Comunicação. Apólice de seguro de responsabilidade civil. Contactos.
------------------	---

EI.6	Banco
<i>Dono</i>	(Banco de Portugal)
<i>Gestor</i>	(Banco de Portugal)
<i>Descrição</i>	Informação sobre o Banco.
<i>Atributos</i>	Identificação comercial (nome registado, sede, capital social). Identificação fiscal (número de identificação fiscal - NIF).

EI.7	Seguradora
<i>Dono</i>	(Instituto de Seguros de Portugal)
<i>Gestor</i>	(Instituto de Seguros de Portugal)
<i>Descrição</i>	Informação sobre a Seguradora.
<i>Atributos</i>	Identificação comercial (nome registado, sede, capital social). Identificação fiscal (número de identificação fiscal - NIF).

EI.8	Mediação imobiliária para venda
<i>Dono</i>	Vendedor
<i>Gestor</i>	Mediador Imobiliário
<i>Descrição</i>	Informação resultante da relação entre o Vendedor e o Mediador Imobiliário, para a venda de imóvel.
<i>Atributos identificadores</i>	Referência Vendedor. Referência Mediador Imobiliário. Referência Imóvel. Contrato de mediação. Condições de venda.

EI.9	Empréstimo para compra de imóvel
<i>Dono</i>	Banco
<i>Gestor</i>	Banco
<i>Descrição</i>	Informação resultante da relação entre o Banco e o Comprador do imóvel. É um contrato em que fica registado o acordo estabelecido entre o Banco (mutuante) e o seu Cliente (mutuário) relativo a um financiamento e onde se especificam todas as suas condições (montante, prazos, taxas de juro, etc.). Pode tomar a forma de escritura pública.
<i>Atributos identificadores</i>	Referência Banco. Referência Comprador. Referência Imóvel. Avaliação do imóvel. Proposta de crédito.

EI.10	Promessa de compra e venda de imóvel
<i>Dono</i>	Comprador
<i>Gestor</i>	Notário
<i>Descrição</i>	Informação resultante da relação entre o Vendedor e Comprador, sustentada por um Notário. A promessa normalmente inclui um sinal, que é um valor que o Comprador entrega ao Vendedor depois de tomar a decisão de compra. Constitui o início do pagamento da habitação e funciona como garantia do interesse do Comprador.
<i>Atributos identificadores</i>	Referência Vendedor. Referência Comprador. Referência Notário. Referência Imóvel. Modelo do contrato promessa de compra e venda (cláusulas). Contrato promessa de compra e venda.

EI.11	Escritura de imóvel
<i>Dono</i>	Comprador
<i>Gestor</i>	Notário
<i>Descrição</i>	Informação resultante da relação entre o Vendedor e Comprador para a transferência do imóvel, sustentada por um Notário. A escritura é o contrato pelo qual se transmite o bem de um proprietário para outro através de um documento escrito e assinado por ambas as partes perante o Notário.
<i>Atributos identificadores</i>	Referência Vendedor. Referência Comprador. Referência Notário. Referência Imóvel. Escritura de compra e venda de imóvel.

EI.12	Hipoteca de compra e venda de imóvel
<i>Dono</i>	Banco
<i>Gestor</i>	Notário
<i>Descrição</i>	Informação resultante da relação entre o Banco e Comprador para a hipoteca do imóvel comprado, sustentada por um Notário. A hipoteca é uma garantia real que confere ao credor o direito de ser pago pelo valor do imóvel pertencente ao devedor, com preferência sobre os demais credores.
<i>Atributos identificadores</i>	Referência Banco. Referência Comprador. Referência Notário. Referência Imóvel. Minuta registo hipoteca. Escritura de hipoteca.

EI.13	Pagamento de Imposto sobre Transmissões Onerosas de Imóveis (IMT)
<i>Dono</i>	Finanças
<i>Gestor</i>	Finanças
<i>Descrição</i>	Informação resultante da relação entre o Comprador e as Finanças para o pagamento do IMT. O IMT é um imposto a pagar de uma só vez antes da escritura e incide sobre o maior dos seguintes valores: o constante do acto ou do contrato ou o valor patrimonial tributário.
<i>Atributos identificadores</i>	Referência Comprador. Referência Imóvel. Comprovativo do pagamento do IMT.

EI.14	Isenção de Imposto Municipal sobre Imóveis (IMI)
<i>Dono</i>	Câmara Municipal
<i>Gestor</i>	Finanças
<i>Descrição</i>	Informação resultante da relação entre o Comprador, as Finanças e a Câmara Municipal, para a isenção de IMI. O IMI é um imposto municipal anual, que incide sobre o valor patrimonial tributário da habitação. A habitação própria permanente pode ser isenta deste imposto por um período de 3 a 6 anos, consoante o seu valor patrimonial tributário devendo ser efectivamente afecta àquele fim no prazo de 6 meses após a escritura.
<i>Atributos identificadores</i>	Referência Comprador. Referência Imóvel. Comprovativo de isenção de IMI.

EI.15	Seguro de vida e de imóvel
<i>Dono</i>	Comprador
<i>Gestor</i>	Seguradora
<i>Descrição</i>	Informação do seguro de vida do Comprador.
<i>Atributos</i>	Referência Comprador. Referência Imóvel. Proposta de contrato de seguro de vida; Contrato e apólice de seguro de vida. Proposta de contrato de seguro de imóvel; Contrato e apólice de seguro de imóvel.

E. Legislação portuguesa

O Ministério da Justiça de Portugal define o papel de conservatórias de registos e de notários, no que respeita a pessoas e bens. As *conservatórias* efectuam o registo de pessoas e bens. Os *notários* efectuam o registo de transacções de bens.

As *referências legais* utilizadas para este anexo foram:

- Portal do Ministério da Justiça de Portugal: <http://www.mj.gov.pt>
- Dicionário Jurídico – Jurinform: <http://www.lexportugal.com/LexPortugal/>
- Verbo Jurídico – Portal de Direito: <http://www.verbojuridico.net/>
- APN – Associação Portuguesa de Notários: http://www.geocities.com/apn_notarios/
- Associação Sindical dos Oficiais dos Registos e Notariado (ASOR): <http://www.asor.pt>

E.1. Conservatórias

As conservatórias, emitem e gerem os registos diversos:

- Certidão do *registo civil*: bilhete de identidade, passaporte, segurança social ou processo de casamento;
- Certidão do *registo comercial* – As certidões do registo comercial visam publicitar a situação jurídica dos comerciantes individuais, das sociedades comerciais, das sociedades civis sob forma comercial, das empresas públicas, entre outras, tendo como objectivo a segurança do comércio jurídico;
- Certidão do *registo predial* – Através das certidões de registo predial pretende-se obter informação sobre a situação jurídica dos prédios, designadamente sobre quem é o proprietário ou se está hipotecado. As certidões de prédios descritos dizem respeito aos imóveis já registados na conservatória. As certidões de prédios não descritos informam se determinado prédio está ou não registado e, estando, a quem pertence e que ónus ou encargos incidem sobre o mesmo.

E.2. Notário

O notário é o jurista a cujos documentos escritos, elaborados no exercício da sua função, é conferida *fé pública*, o qual reveste a dupla qualidade de *oficial público*, uma vez que confere autenticidade aos documentos e assegura o seu arquivamento, e de *profissional liberal*, já que actua de forma independente, imparcial e por livre escolha dos interessados. O notário está sujeito à fiscalização e

acção disciplinar do Ministro da Justiça e dos órgãos competentes da Ordem dos Notários (cfr. artigo 1º e seguintes do Decreto-Lei Nº 26/2004 de 4 de Fevereiro e artigo 1º do Código do Notariado). O local onde é desempenhada a função notarial é designado por *Cartório Notarial*.

E.3. Documentos electrónicos e assinaturas

O *documento* é o objecto elaborado pelo homem com o fim de reproduzir ou representar uma pessoa, coisa ou facto (cfr. artigo 362º do Código Civil). (Dir. Civil) É a declaração corporizada em escrito, ou registada em disco, fita gravada ou qualquer outro meio técnico, inteligível para a generalidade das pessoas ou para um certo círculo de pessoas, que, permitindo reconhecer o emitente, é idónea para provar facto juridicamente relevante, quer tal destino lhe seja dado no momento da sua emissão quer posteriormente; e bem assim o sinal materialmente feito, dado ou posto numa coisa para provar facto juridicamente relevante e que permite reconhecer a generalidade das pessoas ou a um certo círculo de pessoas o seu destino e a prova que dele resulta (cfr. artigo 255º do Código Penal de 1995). (Dir. Penal)

O *documento electrónico* é o documento elaborado mediante processamento electrónico de dados (cfr. artigo 2º do Decreto-Lei Nº 290-D/1999, de 2 de Agosto).

O *certificado de assinatura* é o documento electrónico autenticado com assinatura digital e que certifique a titularidade de uma chave pública e o prazo de validade da mesma chave (cfr. artigo 2º do Decreto-Lei Nº 290-D/1999, de 3 de Abril).

A *assinatura digital* é o processo de assinatura electrónica baseado em sistema criptográfico assimétrico composto de um algoritmo ou série de algoritmos, mediante o qual é gerado um par de chaves assimétricas exclusivas e interdependentes, uma das quais privada e outra pública, e que permite ao titular usar a chave privada para declarar a autoria do documento electrónico ao qual a assinatura é aposta e concordância com o seu conteúdo, e ao declaratório usar a chave pública para verificar se a assinatura foi criada mediante o uso da correspondente chave privada e se o documento electrónico foi alterado depois de aposta a assinatura (cfr. artigo 2º do Decreto-Lei Nº 290-D/1999, de 2 de Agosto).

A *assinatura electrónica* é o resultado de um processamento electrónico de dados susceptível de constituir objecto de direito individual e exclusivo e de ser utilizado para dar a conhecer a autoria de um documento electrónico ao qual seja aposta, de modo que:

- i. identifique de forma unívoca o titular como autor do documento;
- ii. a sua aposição ao documento dependa apenas da vontade do titular;
- iii. a sua conexão com o documento permita detectar toda e qualquer alteração superveniente do conteúdo deste.

(cfr. artigo 2º do Decreto-Lei Nº 290-D/1999, de 2 de Agosto).

A *chave pública* é o elemento do par de chaves assimétricas destinado a ser divulgado, com o qual se verifica a assinatura digital aposta no documento electrónico pelo titular do par de chaves assimétricas, ou se cifra um documento electrónico a transmitir ao titular do mesmo par de chaves (cfr. artigo 2º do Decreto-Lei Nº 290-D/1999, de 2 de Agosto).

A *chave privada* é o elemento do par de chaves assimétricas destinado a ser conhecido apenas pelo seu titular, mediante o qual se apõe a assinatura digital no documento electrónico, ou se decifra um documento electrónico previamente cifrado com a correspondente chave pública (cfr. artigo 2º do Decreto-Lei Nº 290-D/1999, de 2 de Agosto).

A *credenciação* é o acto pelo qual é reconhecido a uma entidade que o solicite e que exerça actividade de entidade certificadora referida na alínea h) deste artigo o preenchimento dos requisitos definidos no presente diploma para os efeitos nele previstos.

A *autoridade credenciadora* é a entidade competente para a credenciação e fiscalização das entidades certificadoras

A *entidade certificadora* é a entidade ou pessoa singular ou colectiva credenciada que cria ou fornece meios para a criação das chaves, emite os certificados de assinatura, assegura a respectiva publicidade e presta outros serviços relativos a assinaturas digitais;

Um *certificado de assinatura* é um documento electrónico autenticado com assinatura digital e que certifique a titularidade de uma chave pública e o prazo de validade da mesma chave;

A *validação cronológica*: declaração de entidade certificadora que atesta a data e hora da criação, expedição ou recepção de um documento electrónico;

O *endereço electrónico* é a identificação de um equipamento informático adequado para receber e arquivar documentos electrónicos.