UNIVERSITY OF SOUTHAMPTON

Engineering a Semantic Web Trust Infrastructure

by

Marcus D. Cobden

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Physical Sciences and Engineering
Electronics and Computer Science

August 2014

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Marcus D. Cobden

The ability to judge the trustworthiness of information is an important and challenging problem in the field of Semantic Web research. In this thesis, we take an end-to-end look at the challenges posed by trust on the Semantic Web, and present contributions in three areas: a Semantic Web identity vocabulary, a system for bootstrapping trust environments, and a framework for trust-aware information management.

Typically Semantic Web agents, which consume and produce information, are not described with sufficient information to permit those interacting with them to make good judgements of trustworthiness. A descriptive vocabulary for agent identity is required to enable effective inter-agent discourse, and the growth of trust and reputation within the Semantic Web; we therefore present such a foundational identity ontology for describing web-based agents.

It is anticipated that the Semantic Web will suffer from a trust network bootstrapping problem. In this thesis, we propose a novel approach which harnesses open data to bootstrap trust in new trust environments. This approach brings together public records published by a range of trusted institutions in order to encourage trust in identities within new environments.

Information integrity and provenance are both critical prerequisites for well-founded judgements of information trustworthiness. We propose a modification to the RDF Named Graph data model in order to address serious representational limitations with the named graph proposal, which affect the ability to cleanly represent claims and provenance records. Next, we propose a novel graph-based approach for recording the provenance of derived information. This approach offers computational and memory savings while maintaining the ability to answer graph-level provenance questions. In addition, it allows new optimisations such as strategies to avoid needless repeat computation, and a delta-based storage strategy which avoids data duplication.

Contents

List of Figures

List of Tables

Listings

Declaration of Authorship

I, **Marcus D. Cobden**,

declare that the thesis entitled

Engineering a Semantic Web Trust Infrastructure

and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research.

I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- none of this work has been published before submission.

Signed: _____

Date: _____

Acknowledgements

Firstly I would like to thank my supervisors, Nick and Les, for their guidance and assistance throughout the course of my PhD, as well as Hugh Glaser, who was also part of my supervisory team prior to his retirement. I also thank the University of Southampton for giving me the opportunity to undertake this PhD, and securing funding for me to do so.

I am thankful for my friends, colleagues and fellow students who have guided, encouraged and enlightened me throughout the course of my studies.

I would like to show my gratitude to my family for their support and encouragement throughout, and for the opportunities they have given me in life.

Above all, I am truly indebted to and thankful for the support, advice, encouragement and patience of my fiancée Kate. My mere expression of thanks does not suffice to express my gratitude; without you I could not have got this far.

Acronyms

ACL Access Control List. 98

AI Artificial Intelligence. 56

CA Certificate Authority. 34, 35

CR Certified Reputation. 31

CRS Coreference Service. 56

DAI Distributed Artificial Intelligence. 29

DCMI Dublin Core Metadata Initiative. xxi

DNS Domain Name System. 71, 79, 87, 88, 116

EV Extended Validation. 35, 85, 89

FOAF Friend of a Friend. xxi, 54, 55, 62

HABIT Hierarchical And Bayesian Inferred Trust Model. 31

HTML Hypertext Markup Language. 40, 41, 45, 51, 63

HTTP Hypertext Transfer Protocol. 9, 40–42, 47–51, 60, 63, 64, 67, 71, 72, 79

HTTPS Hypertext Transfer Protocol Secure. 9, 37, 40, 88

IANA Internet Assigned Numbers Authority. 71

ICANN Internet Corporation for Assigned Names and Numbers. 71

IETF Internet Engineering Task Force. 40

IP Internet Protocol. 3, 40, 71, 87, 88

IQ Information Quality. 24

IRW Identity of Resources on the Web. xxi, 64, 66, 68

ISBN International Standard Book Number. 43

ITU-T ITU Telecommunication Standardization Sector. 34

JSON JavaScript Object Notation. 43, 44

LUBM Lehigh University Benchmark. 112

MAS Multi-Agent Systems. 15, 29, 55

N3 Notation 3. 44, 68, 76–78

NG4J Named Graphs API for Jena. 110

OWL The Web Ontology Language. xxi, 46, 47, 66, 112

PGP Pretty Good Privacy. 35

PKI Public Key Infrastructure. 9, 34–36, 38

POWDER Protocol for Web Description Resources. xxi, 65, 66, 72, 76

POWDER-S Semantic POWDER. 65, 66, 68, 72

RDF Resource Description Framework. 42–51, 53, 55, 57, 62, 70, 76–78, 98, 99, 101, 102, 106, 110, 113, 115, 117

RDFa RDF in Attributes. 45

RDFS RDF Schema. xxi, 46, 47, 51, 112

SPARQL SPARQL Protocol And RDF Query Language. 46, 47, 98

SSL Secure Socket Layer. 54, 85, 92

TAG Technical Architecture Group. 49–51

TLS Transport Layer Security. 10, 40

TMS Truth Maintenance System. 105

TRA Theory of Reasoned Action. 36

URI Uniform Resource Identifier. 41–43, 45, 46, 48–51, 54, 56, 62–64, 66, 68, 71, 72, 79, 80, 98, 102, 103

URL Uniform Resource Locator. 9, 42, 43, 45, 63, 64, 85

URN Uniform Resource Name. 43

W3C World Wide Web Consortium. 25, 40, 42, 46, 47, 49, 50, 53, 54, 94

WSDL Web Services Description Language. 66

WWW World Wide Web. 15, 36, 38–40, 57

XHTML Extensible Hypertext Markup Language. 40

XML Extensible Markup Language. 40, 43–45, 65, 72

Ontologies

Chapter 1

Introduction

Trust is an integral part of modern society: it plays a crucial role in our everyday lives and is the foundation upon which our societies are built. We unconsciously rely on trust in everything we do: when we buy our groceries, we trust that they were produced and handled in a safe environment; when we leave our homes, we trust that they will not be broken into while we are away; and when we get up each morning, we trust in our internal models of the world that getting up is a wise decision.

The World Wide Web is increasingly regarded as an important and convenient source of free information, despite the fact there are no guarantees that information published on the Web will be accurate or honest. In the physical world, the logistical hurdles of publishing, the reputations of publishing companies, and the threat of libel generally discourage the widespread publication of dishonest or grossly inaccurate information. On the Web, the barriers to content publishing and wide distribution are almost non-existent, and the prolificacy of free and anonymous publishing platforms has undermined the risks present in the offline world. Thus, it is all the more important that we are able to judge the trustworthiness of an information source, before we rely on its information. It is despite all this that we have come to increasingly regard the World Wide Web as an important source of free information.

The Semantic Web [Berners-Lee et al. 2001] promotes the notion of a Web of information, built upon a common data model and a foundation of World Wide Web technologies. The aim of the Semantic Web movement is to create a Web of information that is as powerfully interlinked as the Web of human-readable pages, but where the facts are not obscured by language and visual markup. It is hoped that this Web of interlinked data, the vocabularies which arise from it, and the technologies built around it, will allow easier information discovery and integration. In turn, it is hoped that this will enable the development of innovative applications and powerful information analysis.

As it is an uncontrolled environment, the Semantic Web will suffers from similar problems of trust to the World Wide Web. That is, we must expect some actors in this

environment to exhibit untrustworthy behaviour, as there are few incentives to discourage it. Trust remains an acknowledged research problem among the Semantic Web community [O'Hara et al. 2004, Richardson et al. 2003, Artz and Gil 2007]; however, we have yet to see Semantic Web systems which can judge the trustworthiness of real-world Semantic Web data routinely and effectively.

Against this background, in this work, we present a number of mechanisms which provide a foundation for the future growth of the trust on the Semantic Web, and better enable us to make reasoned judgements of trustworthiness.

We continue this chapter by first outlining the problem trust poses for the Semantic Web (Section 1.1) and identifying the key objectives of our work (Section 1.2). Next, in Section 1.3, we explain the contributions we have made to the state of the art. We then conclude this chapter in Section 1.4 with outline of the remainder of this thesis.

1.1 Trust and the Semantic Web

Briefly, we consider trust to be the act of relying on something in the face of risk, and trustworthiness to be a judgement of the degree of confidence that something can be relied upon in a specific risky context[1].

The Semantic Web is a digital environment built upon the technologies of the World Wide Web and the Internet. Berners-Lee et al.'s visionary article for the Semantic Web [Berners-Lee et al. 2001] proposed an ecosystem of intelligent artificial actors, 'Semantic Web agents', who are able to evaluate the trustworthiness of other agents and services, and are able to make intelligent informed decisions on behalf of their users. However, in this digital context, judging the trustworthiness of some entity or artefact is difficult.

In the context of the Semantic Web, there are two primary situations in which trustworthiness becomes important: when acting on information gleaned from the Semantic Web, or when relying on other Semantic Web agents. With little to no information about the identity and characteristics of other Semantic Web agents it is not possible to make effective assessments of their trustworthiness. In turn, it is also impossible to judge the trustworthiness of any information they may have provided you with, as there are no assurances that it was not maliciously altered.

Unlike face-to-face human interactions, where there are a multitude of factors (such as facial expressions, posture, attire and conversational manner) which we employ to gauge trustworthiness, in a digital context there are very few intrinsically observable characteristics on which trustworthiness can be judged. Web user-agent identifiers and

[1] We give this brief definition of trust and trustworthiness for the sake of clarity, in lieu of a fuller definition in Section 3.1.1.

Internet Protocol (IP) addresses are among these observable characteristics, however neither are likely to be of particular use for assessing trustworthiness. The former is provided on an honesty basis and thus is of little worth, and the latter, IP addresses, are unlikely to prove good predictors of individual trustworthiness as they cannot easily be resolved to a single individual due to the processes by which they are allocated.

In the digital environment that is the Semantic Web, the challenges of trust and trustworthiness are often technological as well as sociological in nature. In this thesis, we aim to better identify these challenges and, where possible, to also propose solutions to them.

1.2 Research Aims and Objectives

The ultimate aim of this thesis is to advance the state of the art in trust research for the Semantic Web, and to further the vision of the Semantic Web as an ecosystem of intelligent, and trust-aware, autonomous actors. Specifically, this work aims to advance the state of the art on three fronts:

1. **Understanding Trust**

   Our work aims to develop an understanding of trust, both to clarify the terms in which we will frame our research, and to improve how trust is understood in the wider picture of Semantic Web research.

   Trust is studied in a number of disciplines, including psychology, sociology, philosophy, and management. Each of these disciplines approaches trust from their own perspective. Perhaps as a result of this, Semantic Web research appears to have approached the subject of trust from subtly different directions.

   As part of this thesis, we aim to develop a clear understanding of trust, and, to some extent, reconcile the different approaches of existing Semantic Web research in this area.

2. **Enabling Trust**

   Next, we then aim to address what we believe to be the core challenges on the road towards a trust infrastructure for the Semantic Web.

   In particular we focus on the architectural issues whose resolution is crucial for the development of an effective trust ecosystem on the Semantic Web.

3. **Building and Harnessing Trust**

   Finally, we aim to demonstrate how we might begin to grow Trust within the Semantic Web, and to investigate what improvements we must make to our information management systems if we are to be able to make effective judgements of trustworthiness on Semantic Web information.

1.3 Research Contributions

Based on the aims and objectives described above, and our study of the research area, we present the following as our primary contributions in this thesis:

1. **Semantic Web identity vocabulary**

   The challenge of describing an identity on the Semantic Web is one which has been wisely avoided for some time, as general purpose solutions are difficult to get right. However, in order to perform effective judgements of trustworthiness, we need to be able to describe (and potentially also communicate) the behaviour and characteristics of other agents. Therefore we have developed an identity vocabulary to describe web-based agents, including Semantic Web agents. This vocabulary is intended to act as a foundation for the growth of trust and reputation systems for the Semantic Web.

2. **Grounding Trust in existing authorities**

   Trust is difficult to create; trustworthiness is usually earned by one's actions and thus gaining trust takes time and effort. In a new environment, such as the Semantic Web, where there are very few or no established trustworthy actors, every interaction involves more risk. As a consequence, the interactions required to build a trustworthy reputation in this type of environment involve more risk.

   To avoid the elevated risk, we present a website identity dashboard which demonstrates how we might harness trust transfer to 'bootstrap' trust in Web-based identities. Our system seeks to provide users with information which lends credibility to an identity and identifies its links with existing real-world entities. It is by drawing on information from trusted institutions, that it seeks to encourage trust transfer, and thereby help to bootstrap trust in an identity.

3. **Trust-aware information management**The ability to make sound judgements of trustworthiness depends entirely on having information of a known trustworthiness upon which to ground one's assessment. One of the advantages of Semantic Web systems is their ability to combine heterogeneous information and to draw new information from it.

   In this vein, we first propose an improvement to the Named Graph data model in order to permit advanced representations while maintaining data integrity. Then, we propose a novel approach to recording provenance information, which scales with the number of graphs, rather than the number of statements. This approach also enables computational and memory optimisations for information management, while maintaining the ability to answer questions of trustworthiness.

1.4 Thesis Outline

The remainder of this thesis is organised as follows: in Chapter 2, in order to illustrate the role of trust in the Semantic Web, we describe a number of fictional scenarios involving Semantic Web agents, in which trust plays some part. These scenarios highlight certain challenges and questions which have shaped the direction of our work.

Then, in Chapter 3, we study 'trust' in more detail, discussing what it means to trust, and what forms of trusting decision may be made. We also look at attempts to introduce trust mechanisms in artificial environments, and the role encryption algorithms have played (Section 3.4).

Next, Chapter 4 discusses the World Wide Web and the Semantic Web in greater detail, giving an introduction to their basic concepts and processes, before returning to the subject of trust in the context of the Semantic Web (Section 4.3). Concluding this chapter, we identify the key Semantic Web research challenges with respect to trust (Section 4.4).

Against this background, in Chapter 5, we present a web server identity vocabulary intended to form the foundation of a Semantic Web identity infrastructure. We begin this chapter by refining our requirements in this endeavour (Section 5.2) before we undertake a specialised review of related work in Section 5.3. Section 5.3.4 presents our web server identity vocabulary, providing a detailed description of its terms, and the rationale for our design decisions, particularly those related to co-reference (Section 5.5). Then, in Section 5.7, we review whether we have met our requirements.

Continuing, Chapter 6 proposes trust transfer as a means to address the chicken and egg bootstrapping problem faced by young trust networks, and presents a website identity service which demonstrates this approach. This service, presented in Section 6.3, employes identity information in order to improve the appropriate perception of entitativity, and thereby encourage trust transfer. This builds on our work thus far, providing a means by which we can bootstrap trust in the identities of Semantic Web agents. In Section 6.4 we discuss the potential of our prototype, the questions it raises, and its shortcomings and avenues for potential future work.

In Chapter 7, we examine the problem of making judgements of information trustworthiness. We identify the limitations of graph 'acceptance' and the implications this has on provenance records and the named graph data model (Section 7.1), before proposing a potential solution in Section 7.1.2. Extensive provenance records are necessary for sound assessments, however recording such information for inferred data has traditionally been costly (Sections 7.1, 7.2.3). We then propose a graph-based provenance recording approach which exhibits significantly better scaling behaviour than previous approaches (Section 7.2.4).

Finally, Chapter 8 concludes our work, recapitulating our contributions and discussing avenues for future work.

Chapter 2

Use Cases

In this chapter we present short scenarios describing interactions between people and hypothetical Semantic Web agents; intelligent artificial actors able to make intelligent informed decisions on behalf of their users. Each scenario aims to illustrate different aspects of their interactions, and the role of trust in each. We follow each scenario with a discussion of the key trust-related challenges it presents for the Semantic Web.

The first use case (Section 2.1) describes an e-commerce scenario in which Semantic Web agents manage searches and transactions on behalf of their owners. The second (Section 2.2) tells of a reputation scenario, where a cautious individual investigates the identity and reputation of potential gift vendors in order to determine their trustworthiness.

The third (Section 2.3) is an information management scenario, describing the challenges involved in acting on information of unknown quality, and reacting to errors. We examine each scenario in the context of our research objectives (Section 1.2), in order to identify the key challenges they present.

2.1 Semantic Web Agents and Services

Berners-Lee et al.'s article on the Semantic Web [Berners-Lee et al. 2001] described a vision of personal software agents which employed Semantic Web technologies for their users' benefit. This use case considers a similar scenario involving assistive Semantic Web Agents, in an business-to-business e-commerce situation.

2.1.1 Use Case #1

Susan, the owner of small retail company, owns a Semantic Web agent which publishes her catalogue online and also operates an e-commerce service interface.

Tom's personal Semantic Web agent is investigating the prices of certain goods on his behalf and, during the course of this activity, requests some information from Susan's agent. Alongside the information on goods, Tom's agent also records the means by which it came to hold this information, which includes a description of Susan's agent and records of the 'conversation' which took place.

The following day, Tom shares the results of the investigation - which showed that Susan offers the best price - with his colleague Hubert, including his agent's description of her agent's identity. Hubert's agent is able to contact Susan's agent using that description, and also be sufficiently confident that it is in fact the same agent – the description of her agent matches its observations. Hubert instructs his agent to purchase of a small quantity of the goods, as a trial purchase.

Meanwhile, due to the continued success of her business, and the increased demand on her Semantic Web agent, Susan upgrades her e-commerce agent from a single server to a powerful multi-server deployment.

On receipt of the goods Hubert and Tom decide that they are pleased with their quality and the level of service, and Hubert instructs his agent to purchase a larger quantity of the same item. When Hubert's agent attempts to place the second order, it notices that Susan's agent seems substantially different from its previous description. It decides that it cannot proceed with the transaction as it does not have enough confidence that they are the same agent, and must seek advice from Hubert.

Hubert contacts Susan directly in order to confirm the changes, and then instructs his agent that the identity of Susan's agent can be trusted. His agent records the change in apparent identity, and proceeds to place the order.

Hubert's agent shares the new identity information with Tom's agent, including the fact that Hubert has vouched for its legitimacy in this context. As Tom's agent has already been advised to trust Hubert's judgements, it will be able to conduct future business with Susan's agent confidently, without needing to request Tom's advice.

2.1.2 Key Challenges

The creation of effective personal software agents brings together a number of problems which are significant research challenges in their own right. In light of this, the key challenges we identify in this scenario are only those directly relevant to our study of trust and the Semantic Web. These challenges are as follows:

- **Describing identity**

  The ability to distinguish between different entities is a prerequisite of an effective trust system. One must be able to identify and record the definitive characteristics that identify a certain entity in order to be able to recognise it when it is encountered at a later date. Therefore, the ability to describe the identity of an entity to be a key challenge for Semantic Web agents.

- **Describing relationships**

  If agents are operating in concert we must be able to describe the relationship between them to some degree, so that we may factor it into our trusting decisions, where approriate.

- **Communicating identity**

  Next, if we wish to share our experiences of certain entities with others, we must be able to communicate our description of that entity's identity. If we cannot communicate a usable description of identity, shared records of our experiences become less useful as they cannot be linked to real entities or used to inform trusting decisions.

- **Comparing identities**

  Finally, assuming we can both effectively describe and communicate the identities of other actors, the last key challenge is the comparison of identity descriptions. If we encounter the same agent on two separate occasions, we must have some means by which we can confidently determine whether or not it is the same agent. Additionally, if we have received information about trustworthy agents from a trusted peer, we must be able to identify whether an agent was or was not featured in those recommendations.

Uniform Resource Locators (URLs) alone are not sufficient to solve these challenges; they are primarily a means of encoding an address, and generally do not themselves give any assurance of identity[1]. By analogy, you may have recorded an acquaintance's home address but you cannot be sure from the address alone whether or not they still live there.

In summary, from the scenario we described, we identified three key requirements for an identity infrastructure; the ability to describe, communicate and compare the identities of agents. As we argued above, an identity system is a prerequisite for an effective trust system, thus these requirements are important in ensuring that trust systems are built on a robust foundation.

[1]HTTP based URLs delegate ownership and identity to registries of Domain names or IP addresses. HTTPS URLs add to this some cryptographic assurance of identity through the Public Key Infrastructure (PKI) (Section 3.4.2.3), however the level of assurance varies with the type of certificate. Other URL schemes may have stronger identity assurances, however few are in common use.

2.2 Trusting Decisions

This scenario describes the process by which one might arrive at a trustworthiness judgement for a previously un-encountered web site.

2.2.1 Use Case #2

Martin is a cautious individual; when using the Web he takes the time to be confident in the identity and trustworthiness of a Web site before deciding to do business on it. In anticipation of the holiday season, Martin is on the lookout for novel gifts for his friends and family, and has found a number of new gift vendors.

The first vendor presents a Transport Layer Security (TLS) certificate, using it to encrypt the communications between itself and Martin, and assure its identity. Martin's browser warns him that he has never visited this website before, suggests that he first investigate its reputation further before continuing. None of Martin's peers report having any experience of this vendor, so he turns to other sources of information.

Consulting a number of online reputation repositories turns up a number of favourable reviews for this merchant in one review repository. Martin decides that he can trust this vendor as, in previous situations, he has found this repository to be accurate, and generally trusts the reviews of its contributors.

Martin moves on to the next vendor, who also presents a TLS certificate. This certificate, however, only certifies certain cryptographic keys for use at this address, and does not provide any assurances of identity, so Martin decides he must look for more information.

Unfortunately, Martin is unable to obtain any information on the vendor's legitimacy from reputation repositories or his peers After further investigation, Martin is able to associate the vendor's Web presence with a company registered with his national government, using publicly available records. Martin has past experience of this company from visiting their brick-and-mortar stores, and so is happy to place his trust in them.

Martin realises that his dilligent investigations have taken a considerable amount of time. He wishes that he could delegate the investigation of trustworthness to somebody else, so that he could focus on selecting the best gifts.

2.2.2 Key Challenges

This scenario describes a setting where a number of trusting decisions are to be made in the context of the World Wide Web. One of the visionary goals of the Semantic Web community is to work towards a world where we can construct autonomous artificial Semantic Web agents which can undertakes these kinds of tasks on a person's behalf.

While we described the actions of a human browsing the Web, without significant changes it could also describe the actions an autonomous artificial agent. The trusting decisions in the scenario are equally relevant for artificial agents.

We have highlighted the following separate research challenges from how these trusting decisions are approached and undertaken:

- **Assessing trustworthiness**

  The primary challenge here is the assessment of whether or not a vendor is trustworthy. In our scenario we have assumed that Martin has made this kind of assessment before, and thus already has his own personal criteria for this. Assessing trustworthiness is a complex task, there are many factors on which a decision could be based, and the degree of trustworthiness required may vary with the situation. The choice of factors, and the degree of trust needed, are generally highly subjective decisions.

- **Discovering reputation**

  We cannot expect every actor to already have sufficient knowledge to make a trust judgement of another actor they have never before encountered. It follows therefore that the Semantic Web needs reputation discovery mechanisms. This, in turn, requires solutions to the challenges which we highlighted in Section 2.1.2; identity description, communication and comparison.

- **Bootstrapping trust**

  Building a favourable reputation is a chicken-and-egg problem; it is difficult to gain one without interacting with others, and it is difficult to do so without a favourable reputation as it increases the risks involved.

  In order to boost the growth of reputation information on the Semantic Web, and thereby improve judgements of trustworthiness, we should seek ways to reduce the risks of interaction. One potential means for achieving this is to look for sources of identity and legitimacy which currently lie outside of the Semantic Web, which may be harnessed by Semantic Web agents.

From the scenario we described, we identified three key challenges; the judgement of trustworthiness, the sharing and discovery of reputation information, and the building of reputations, particularly in new environments.

2.3 Information Management

The World Wide Web is an open platform, internet connectivity and technical skills are the only barriers to publishing information on it. The Semantic Web, being built upon

the World Wide Web, is no less open. There no inbuilt controls on what content may be published on the Web and the Semantic Web; social norms and the threat of prosecution are the primary control mechanisms.

It follows then that information gleaned from the Semantic Web may be of varying, unknown or even questionable quality. Therefore, in order to effectively harness this uncontrolled information to inform decisions of trust, we require robust information management systems. This scenario describes an organisation using an information management and integration system to inform their strategic decisions in an environment where information may be of unknown quality.

2.3.1 Use Case #3

As an information analyst, Grace works with an information management system to deduce valuable observations and insights from the facts and information stored in the system's knowledge-base. While the system itself has some limited reasoning and deduction capabilities, there remain areas where humans are better at spotting changes, trends, or at combining seemingly-unrelated facts. When Grace, or the system's inbuilt reasoning unit, deduces new information, it is added to the system's knowledge-base.

All this information feeds into the decisions made by her employer's strategic planning team, who are responsible for putting this information to use.

New and fresh, information is regularly added to the system's knowledge-base by another team. This information comes from a range of publication sources; some sources are well informed and rigorous in their methods, and some gather information from rumours and anonymous sources. As a result, the quality of information may vary greatly, and both the information analysts, and the automated reasoning system, have to pay particular attention to the information's source and whether the information is corroborated by other sources.

In the event that information is discovered to be incorrect – be it through simple error or through subterfuge – Grace's employer wants be able to track the implications of that discovery. They want to be able to identify which of the planning team's decisions are affected, so that they may take corrective action where appropriate.

2.3.2 Key Challenges

One of the key benefits of Semantic Web technologies is their ability to combine data from heterogeneous sources and enable powerful analysis of it. This scenario highlights information management challenges which are a consequence of combining information whose quality is not known ahead of time.

The information management system we described already contains a complete record of how each piece of information came to exist in the system. With additional records of the planning team's decisions, and which pieces of information were key to them, Grace's employer will have all the information they need to track the implications of changes.

While the affected human deductions may need to be reviewed by analysts themselves, it should be possible for the automated reasoning system to validate whether its deductions are still supported by current facts without assistance.

If we wish to effectively work with information collated from various sources, we must have robust systems for handling this kind of task. Thus, we identify the following challenge from the above scenario:

- **Provenance and reasoning over heterogeneous data**

  When performing inference over a collection of documents of varying provenance is is important that adequate provenance information is maintained.

  With inadequate information, there may be no means by which to determine the original sources of a newly inferred piece of information. Without this ability, it would be become prohibitively difficult to correct errors in one's knowledge-base, as the number of included documents increases. The retraction of a single statement would require a complete re-computation of all inferred information.

  The lack of adequate provenance records would also significantly impact any judgement of trustworthiness. If records of the source of inferred information is not kept, we have no basis on which to judge its trustworthiness.

As we have argued, to make decisions based on reasoned trust – rather than blind faith – requires a certain degree of information. Thus with Semantic Web technologies we must take care to maintain adequate provenance data so that we have such information.

2.4 Summary

In this chapter we presented short use cases in order to illustrate the role trust may come to play in the Semantic Web, and to help shape the direction of our research. From these use cases we identified a number of research challenges, these broadly fall into the categories of identity, trust and reputation, and provenance. Some of these research challenges relate specifically to trust on the Semantic Web, however others are more general challenges faced by artificial trust systems.

Before we seek to address these research challenges, we must first develop a better understanding of trust, and mechanisms of trust. Thus, in the next chapter, we focus

on the concept of 'trust', first discussing forms of trust and the various uses of the term, before progressing to study existing work on trust systems in artificial environments.

Chapter 3

Trust

Trust is a concept which has been studied in a wide variety of disciplines, each of which has developed its own unique perspective and understanding of the term. In this chapter we review the literature surrounding trust in order to improve and clarify our understanding of the term and its relation to our work.

We begin by exploring the term 'trust' itself, looking at how it is understood and used as a term (Section 3.1). Against this background, we consider how we choose to trust, i.e. how trustworthiness is assessed (Section 3.2), both in the context of other actors, and in information. Next, in Section 3.3, we discuss the approaches which previous works have employed in order to model trust mathematically.

Continuing, Section 3.4 begins our investigation of trust in artificial environments; we first discuss trust mechanisms from the Multi-Agent Systems literature (Section 3.4), then continuing to discuss trust in digital environments – the forms it takes and the technologies which have been developed to underpin it (Section 3.4.2), and then finally, in Section 3.4.3, we discuss trust on the World Wide Web.

3.1 Trust and its Forms

There is no single widely accepted definition of trust; each discipline coins its own definition on different terms, taking different elements of the concept into consideration. For example, psychologists have considered trust a personal attribute [Erikson 1994], economists are likely to consider trust a result of a rational choice process [McKnight and Chervany 1996], and sociologists discuss the societal structures which are supported by trust [Shapiro 1987]. A number of definitions of trust are quoted below to illustrate these different perspectives.

"[Trust is] confidence in or reliance on some quality or attribute of a person or thing, or the truth of a statement." [Oxford English Dictionary 2011]

"[Trust is] the extent to which one is willing to ascribe good intentions to and have confidence in the words and action of the others." [Cook and Wall 1980]

"[Trust is] an individual's behavioural reliance on another person under a condition of risk." [Currall and Judge 1995]

"[Trust is] the expectation that arises, within a community of regular, honest, and cooperative behaviour, based on commonly shared norms, on the part of other members of that community." [Fukuyama 1995]

"[T]rust, in general, is taken as the belief (or a measure of it) that a person (the trustee) will act in the best interests of another (the truster) in a given situation, even when controls are unavailable and it may not be in the trustee's best interests to do so" [Marsh and Dibben 2005]

"Trust indicates the willingness of an agent to engage in a transaction in the absence of adequate safeguards." [Berger et al. 1995]

"Trust is "the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context". [Grandison and Sloman 2000]

"Trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)."[Olmedilla et al. 2005]

"(a) An individual is confronted with an ambiguous path, a path that can lead to an event perceived to be beneficial ($Va+$) or to an event perceived to be harmful ($Va-$); (b) he perceives that the occurrence of $Va+$ or $Va-$ is contingent on the behaviour of another person; (c) and he perceives the strength of $Va-$ to be greater than the strength of $Va+$.

If he chooses to take an ambiguous path with such properties, I shall say he makes a trusting choice; if he chooses not to take the path, he makes a distrustful choice."[Deutsch 1962]

There are two main themes among these definitions of trust; some describe trust as choice between action whereas others describe it as a bond. Those which define trust as a choice generally agree that it involves the question of whether or not to rely in the face of uncertainty. To further understand the concept of trust, we believe it is beneficial to consider the difference between the two main forms of the term, trust as an act, and trust as a bond. We explore these in the following sections.

3.1.1 An Act of Trust

Trusting is the act of relying on something in an uncertain environment, where it is perceived that the outcome of the situation is contingent on the behaviour of the trusted item. In most definitions of trust the expected potential variance in the outcome is described as the risk.

It is common for us to say that we have trust in some inanimate object – this usage raises some interesting questions. For example, what does it mean to place one's trust in a safety harness; our safety is certainly contingent on the behaviour of the harness, however, the behaviour of the harness, having of course no free will of its own, is entirely derived from our usage of it. Exploring this further; in reality we have not trusted the harness itself, instead we have trusted our judgement that the harness appears safe and then that those who were responsible for constructing and maintaining the harnesses have done so with due care and diligence.

The decision of whether or not to trust is a choice between different courses of action, of which one or more is a trusting path, and one or more is a path which does not rely on trust, which Deutsch named trusting choice and distrustful choice, respectively. When dealing with complex, multifaceted decisions, potential paths may include measures to decrease the degree of risk or selectively avoid particularly risky factors. Thus, it is often possible to take a trusting path which does not rely on trust in every respect. The degree of risk, the context of the choice, and the utility of potential outcomes all play a role in the decision of whether to trust, as such one must remember that the evaluation and weighting of these factors are inherently subjective.

Deutsch's definition of trust, which we quoted in Section 3.1, is the closest to our understanding of trust as an act, although we don't consider it necessary for the outcome $Va-$ to be harmful and greater in relative strength than the outcome $Va+$, only that it be less preferable than $Va+$.

To state this explicitly, in our opinion, *to trust is to choose an ambiguous path where the desired outcome is contingent on the past, present or future behaviour of another.* Trusting decisions may be contingent on past actions only if some aspect of the action remains unknown or uncertain. For example, if one knows that a friend had earlier intended to take some action, one might choose to act trusting that they had indeed

done so. Or, if one acts based on known information, one is trusting that the source of that information properly conveyed the accuracy of, and their confidence in, that information.

3.1.2 A Bond of Trust

Trust as a bond is the notion that two or more individuals are able to comfortably rely on the behaviour of each other. Thus a bond of trust is the confidence that each will act in the best interests of the other when placed in a scenario where the utility of the other is contingent on their own actions. In effect, to say a bond of trust exists is to say that, if an ambiguous path of action were encountered, the participants of the bond would be comfortable taking the trusting path and relying on the behaviour of the other participants. This implies that some judgement of trustworthiness – the degree of confidence that something can be relied upon – has already taken place between the two people.

Trust within society arises from the confidence that other members of the society share the same core values and ideals as oneself, and the conjecture that they will therefore behave in a manner which is consistent with these. These behavioural expectations — or social norms — are enforced within the group and breaching them can lead to punishment and exclusion [O'Hara 2004]. As with bonds of trust, this implies that for trust to exist within a society, its average member would judge another equally average member as trustworthy.

3.1.3 The Costs of Trust

Undeniably, there are benefits to be gained from considering trust, however, there are also costs involved in making trusting decisions [O'Hara et al. 2004]. While it is 'cheap' to assess trustworthiness based only on known information, strategies which involve active research in order to better estimate risk may prove more effective. The costs of different techniques can influence how we choose to evaluate trustworthiness; which methods we employ and how rigorous we are in their execution. These costs fall into three categories which come from the broader economic literature:

Operational costs

Operational costs are the expenses of operating a particular trust system, this includes both the cost of setting up and operating the whole trustworthiness assessment system.

Opportunity costs

This is the cost of a missed opportunity, the benefit a trusting path of action would have conferred over the benefit conferred by the non-trusting path.

Deficiency costs

The deficiency cost is the cost of betrayal or the failure of trust. The size of this cost plays a large part in the amount of risk which is taken in a trusting decision.

3.1.4 Motivations to Trust

Deutsch's definition of trust permits us to identify trusting decisions but, appropriately, does not attempt to consider the motivations which might drive someone to choose a particular path. In his later work, Deutsch lists nine different motivations for a single trusting choice, employing the tale of 'The Lady or the Tiger' [Stockton 1884] as a means of demonstrating them.

To briefly summarise the tale: the story tells of a suitor to a princess, who is discovered by the king. Unfortunately, the king is displeased, and orders the suitor thrown into a pit with two exits. Behind one exit is a ferocious tiger, behind the other a beautiful lady. The young man is about to make a choice, when he notices the princess pointing subtly to one of the doors. He immediately chooses that door and the reader is left to imagine the results.

Below, we discuss our understanding of each of the nine motivations and how they relate to trust in our context, summarising from Marsh's review of Deutsch's work [Marsh 1994].

- **Trust as confidence**

  Trust as confidence is perhaps the most obvious trusting decision; it is attractive because the benefits of co-operation often outweigh the benefits of decreased risk through operating alone. In this case the suitor trusts the princess because he is confident that she will have his best interests at heart. Deutsch emphasised the point that, for this to still qualify as a decision of trust, a perceived element of risk must remain a part of the decision – a point which is echoed by others in the field.

- **Trust as social conformity**

  In some social circles, trust may be expected and showing signs of distrust may lead to severe sanctions or punishments. In our running example, displaying distrust might have lead to social exile or shame, to the suitor (were he to survive the result of the choice) or perhaps to his family.

  We note that social conformity would only begin to impact the behaviour of actor once they begin to operate in a sufficiently complex social environment. It might be employed as an incentive to honest behaviour, for example. It is also worth

noting that social conformity concerns need not alter the true beliefs held by an actor, they need only alter the actor's behaviour such that it appears to hold the same beliefs as the rest of the social circle.

- **Trust as innocence**

  A trusting decision founded on the innocence of the trustee is a result of the trustee's lack of understanding of the full dangers or consequences of the course of action. Marsh [1994] suggests that this innocence may be rooted in lack of information, cognitive immaturity, or cognitive defect. In the story, this might perhaps be choosing a particular door simply because it was the first action which was considered.

  In our understanding, a trust choice can only be classified as due to innocence by an external and more informed observer. From the perspective of the actor making the choice, the decision must fall under one of the other categorisations, or may not be perceived to be a trusting decision at all.

- **Trust as impulsiveness**

  Cognitive immaturity, defect, or certain attitudes toward the future may lead to trust as impulsiveness, a result of not giving proper regard and consideration to the consequences of a particular trusting choice. We consider this choice to be similar to a decision of trust based on innocence, except that it may be a conscious decision.

  Deutsch suggests that individuals who are dominated by the pull of immediate gratification may be prone to this form of trusting decision due to their disregard of the future consequences. Again, in the story, this might be embodied in choosing a particular door simply because it was the first action which was considered.

- **Trust as virtue**

  Deutsch [1973] argues that trust and trustworthiness are necessary for co-operative action and friendly social relations, this trust is naturally considered a virtue in social groups. We suggest that, generally, and also in our story, one might trust selfishly simply to demonstrate one's virtue, or altruistically under the ethical motivation to "Do unto others as you would have them do unto you." This ethical code is often known as 'the Golden Rule' and is found in a wide range of cultures, and has also been shown to have an evolutionary basis [Axelrod and Hamilton 1981]. This form of trust has a similar effect to social conformity except that the incentives arise from the hope of later rewards rather than fear of possible sanctions.

- **Trust as masochism**

  Trust as masochism is difficult to rationalise. Our interpretation of it is that the suitor may choose to open the door expecting the tiger and the pain of unfulfilled

trust. Subsequently, a positive outcome (encountering the lady beyond the door) will be perceived as more rewarding than it would have been otherwise.

This form of trust crucially requires the trustee to have low expectations, the cause of which is beyond the scope of our discussion. We do not expect this form of trusting decision to be of immediate use to software agents, however in non-fatal circumstances it could be used as a demonstration of prowess of some kind.

- **Trust as despair**

  Trust as despair may occur when the negative consequences of not reaching a trusting decision (i.e. either actively distrusting or not reaching any decision) are so great or so certain that the trusting choice is the only decision. In the story of The Lady or the Tiger, indecision means certain death by execution thus reaching a decision is preferable. Additionally, this may be an opposite or sorts to 'trust as masochism', if the suitor decides to trust the princess because the other conclusion – that his affections were unrequited – would lead to despair.

  We consider this form of trust to be easy to imagine in interactions with service providers, where the consequences of not procuring a service are far greater than the potential risk. In the context of trusting information this may represent the choice to rely on information not on its own merit, but on the realisation that if it were not true the situation would be dire beyond the point of rescue.

- **Trust as faith**

  Faith, by its very nature, is again difficult for us to rationalise. The decision to trust based on faith is built on the fact that the agent holds core beliefs which offset the consequences of any decision rationalised by faith. In this example the suitor may have faith in 'the gods' or in their 'destiny' such that either he has conviction that the lady will be behind the door, or that whatever lies behind the door is his pre-ordained path. Whether the door is picked impulsively, or the princess' gesture is heeded as a 'sign', is completely down to the convictions of the suitor.

  Trust as faith is likely 'trust as confidence' under another guise. If one's core beliefs include a spiritual force or being which has control of, or influence over, events, then one could well percieve this as a rational descision of confidence. Again it would take an external observer with different beliefs to identify this as a faith-based decision.

  Taking this further, belief in some spiritual force may be no different - in a decision-making context - to other beliefs about the nature of the world, such as gravity. Spiritual belief does not preclude self-consistent rational trusting decisions. The difference is perhaps that we can identify 'belief in a spiritual force or being' as the primary factor in the confidence-based trusting decision.

- **Risk-taking or gambling**

  If the potential gains of a successful trusting decision are subjectively far greater than the potential losses, the gambling suitor may be prepared to take the risk, even in the face of poor odds. The suitor may also not value his life highly or otherwise decide that life is not worth living without the lady, thus further increasing the subjective attractiveness of a gambling choice.

  We note that the estimations of the value of losses or gains are often irrational in practice, thus some gamblers will take ill-advised risks. There is space for gambling decisions in trust mechanisms, when the potential gain is high, and/or the potential losses low, particularly in environments where the failure of a single interaction would not significantly impact the overall outcome.

The diversity of this list highlights how much the motivations for making a trusting choice may vary. It is worth noting that while only one or two of the motives are immediately identifiable as rational motives, in certain situations each of them may offer an advantage, particularly within the context of a complex environment or large society. For example, while trust as social conformity discourages actions which lead to personal gain, such as theft, it often offers a societal benefit, such as reduced crime levels.

We ascribe the trusting behaviour of the agents in our use cases (Chapter 2) to 'trust as confidence'. It is this form of trust that we which we wish to promote through the application of trust mechanisms. In order to be able to trust based on confidence, our systems will require sufficient information to judge the risk of a particular course of action, and also effective strategies and analytical techniques to do so.

3.2 Assessing Trustworthiness

The assessment of risk and trustworthiness are important challenges in confidence-based trusting decisions. Reaching a trusting decision in a high-risk situation would require a higher level of trustworthiness to counteract the risk. Risk assessment has long been a subject of research in management disciplines, such as project management, public health, and safety and security planning. In this thesis we are primarily interested in the assesment of trustworthiness, and so we place the challenge of risk assessment beyond the scope of our current work.

Almost all of our quoted definitions of trust in Section 3.1 describe trust between two or more actors. In the context of our study of trust, we also wish to consider the act of relying on information, as the Semantic Web is an information-driven environment. Thus we consider the challenge of assessing trustworthiness on two fronts: the assessment of information, and of actors.

3.2.1 Trustworthy Information

Trusting in information is fundamentally different to trusting in an actor; information may be inaccurate or incorrect, but it cannot act or change of its own accord as an actor could. Thus the judgement of the trustworthiness of a piece of information must be based on the qualities of the information. This judgement must be made in a given context as different situations may have fundamentally different requirements on the attributes of information, such as accuracy, precision or certainty. We expect this judgement to incorporate the history of the information: its sources, and the trustworthiness of the source in this context.

Naïve approaches have, in the past, simply adopted the perceived trustworthiness of the information source as the sole indicator of the quality of information. This fails to consider a number of important issues; information may have multiple sources, may have been re-published, or may be derived from other information [Hartig 2010]. In some situations, the history of the information may be as important as the information itself [Hartig 2009b].

There are documented occurrences of bogus information being introduced into the popular online encyclopaedia, Wikipedia, which then became difficult to remove.[1] The bogus fact happens to be reproduced by a more traditional publication, which in turn is cited as a justification for the fact's existence in wikipedia, on the assumption of due diligence by the publisher. In circumstances like these, the full information history – if it were available – would bring this incorrect justification to light.

Returning to the subject of judging trustworthiness, any judgement of information trustworthiness must be made in the context of some perceived risk-bearing decision. If there is no perceived difference in risk between paths, then there is no trusting decision to be considered. We note that this is an uncommon choice, because there are almost always paths of inaction which present different risks.

In the absence of a concrete choice between actions, the purpose of information trustworthiness assessment instead becomes the choice of belief. In this new context the judgement criteria would include such factors as account one's readiness to accept facts from that source, and the possible implications of belief.

If this information is accepted as a belief these beliefs might be called upon in a trusting decision at a later date. In which case the beliefs would then be re-evaluated for trustworthiness in the new context.

Work by Gil and Ratnakar [2002] built on early Semantic Web technologies to demonstrate a powerful information annotation system designed to allow information analysts

[1] Reliability of Wikipedia: See `http://en.wikipedia.org/wiki/Reliability_of_Wikipedia#Notable_incidents`

to decompose statements into composite facts which can then be individually assessed on criteria such as credibility and reliability. Gil and Ratnakar's TRELLIS system allows an information analyst to decompose information content into its constituent statements, and justify or dismiss them based on the source of the information as well as any other corroborating evidence, allowing analysts to document their analysis processes.

TRELLIS is a tool which helps people to make the kind of informed information trustworthiness assessment which we discussed briefly above. Although Semantic Web technologies have advanced considerably since this work was carried out, we still consider it relevant, as it demonstrates the capabilities we believe a fully-fledged Semantic Web information management system should have.

3.2.1.1 Information Quality

The form of information trustworthiness assessment we are considering has been studied elsewhere under the guise of Information Quality (IQ) assessment [Naumann 2002]. IQ assessment is a key challenge for any system which integrates information retrieved from many sources, including the unmoderated Web.

Although judgements of information trustworthiness and credibility are undertaken for different purposes, they are both forms of IQ assessment. Naumann presents a range of criteria on which IQ can be assessed (shown in Table 3.1), the results of which, in his system, are combined through some process into an overall IQ score [Naumann 2002].

While Naumann's list covers most common IQ criteria, it should not be thought of as exhaustive; there may exist criteria which have not yet been considered. Also, the selection of criteria to be evaluated may vary depending on the context of the assessment, as some may be neither relevant nor important in certain situations. We note that some of Naumann's criteria, notably *Reputation*, stray beyond the assessment of the information alone, assessing the behaviour and attributes of the information sources. We discuss synthetic reputation mechanisms further in Section 3.4.

3.2.1.2 Information Provenance

Provenance concerns the origins and known history of an artefact; information provenance is specifically concerned with the origins and history of a particular piece of information. Information provenance is vitally important to the assessment of information trustworthiness. Unless we have recorded information from first-hand experiences, we are put in a position where we must rely on information from external sources. Moreover, the information we do receive from others might not be from those who have directly experienced it.

Category	Criteria
Content-related	Accuracy Completeness Customer Support Documentation Interpretability Relevancy Value-added
Technical	Availability Latency Price Quality of Service Response time Security Timeliness
Intellectual	Believability Objectivity Reputation
Instantiation-related	Amount of data Representational conciseness Representational consistency Understandability Verifiability

TABLE 3.1: Naumann's IQ criteria [Naumann 2002]

As we mentioned in Section 3.2.1, it is not sufficient to consider only the immediate source of information when judging its trustworthiness. Information may have been collated or derived from multiple sources, or simply re-published by a third party. Each actor involved in the history of an informational artefact has had some opportunity to influence the final result. Thus, when we consider the trustworthiness of information, we must take into account the actors and processes which have played a role in the creation of the information we have received.

The World Wide Web Consortium (W3C) PROV family of documents describes a framework for interoperable interchange of provenance information [Moreau and Missier 2013, Lebo et al. 2013]. It is the result of a community effort to to achieve inter-operability in provenance information systems.

3.2.2 Trustworthy Actors

Of the two challenges of trust assessment, information and actors, assessing actors is the most important because, as we argued in Section 3.2.1, the trustworthiness of information depends greatly on the trustworthiness of its sources. Generally, we judge the

trustworthiness of actors based on either observable factors, or on information provided by other actors.

3.2.2.1 Observable Signals

Trustworthiness is most often judged on observable factors such as appearance and behaviour. The field of signalling theory, from the discipline of evolutionary biology [Krebs and Davies 1978], demonstrates that there can be an evolutionary advantage in doing so.

When considering a trusting decision, one will almost certainly never have complete information, or the time to examine it, thus we must turn to heuristic-based indicators to help us to judge trustworthiness. Heuristic take advantage of readily accessible, but often loosely applicable, information to provide approximate answers to problems. Appearance is one such heuristic factor, although it does not directly relate to trustworthiness, it can be a strong indicator of trustworthiness in certain circumstances. For example; wearing a smart outfit, and presenting a polite and friendly manner requires a certain level of investment, and this perceived investment is interpreted as a signal of trustworthiness.

Other forms of dress may send different signals; certain outfits might indicate membership of a social group or movement, for which stereotypes may suggest other patterns of behaviour or attitudes. Membership of trade groups or professional bodies is another signal that people employ to suggest trustworthiness, as they suggest certain standards, codes of conduct, or attention to detail as well as some investment of resources towards gaining membership. We may also employ other heuristics which judge behaviour rather than appearance. For example, it may be possible to guess when someone is practicing deception by looking for specific behaviours, or 'tells', which are common in these situations.

Of course, these indicators are not perfect and it is often possible to imitate them and thereby appear more trustworthy than one deserves. Untrustworthy actors may seek to appear trustworthy without making the full investment of resources. To counter this, each new signal aims to be harder to imitate, and to be a better indicator of trustworthiness. Thus there is a perpetual arms-race between the trustworthy population and the untrustworthy population.

3.2.2.2 Reputation

Reputation is a factor on which humans judge trustworthiness; we judge based on recommendations and other reputation information from our peers. Reputation is a heuristic

mechanism to allow us to judge trustworthiness without the costs of building up sufficient direct experience.

Reputation is not an observable characteristic of an individual and must be gleaned from other members of a community. Although is not a directly observable signal, reputation can be a strong indicator of trustworthiness as it has a number of desirable properties:

- It can be linked to past behaviour within a community.
- The investment required is trustworthy behaviour itself.
- Any untrustworthy behaviour puts any previous investment at risk.
- It cannot be faked by a lone individual.

It is worth emphasising that reputation is an indicator of trustworthiness according to community norms; i.e. there may be some variation in the assessment of each interaction, but in aggregate the reputation should reflect the expectations of the community. Thus if one's preferences deviates from the community norm it will be necessary to re-interpret the reputation information before it can be incorporated into a trustworthiness assessment.

3.3 Modelling Trust and Belief

In order to analyse trust, research often has need to measure or quantify it in some way Commonly, researchers employ either discrete or continuous representations of trust:

Discrete range

In systems which interact with humans, trust is often represented as a discrete range, either numeric or otherwise. Such approaches usually include a descriptive label for each value, for example: (1) Distrusts absolutely (2) Distrusts highly (3) Distrusts moderately (4) Distrusts slightly (5) Trusts neutrally (6) Trusts slightly (7) Trusts moderately (8) Trusts highly (9) Trusts absolutely [Golbeck et al. 2003].

Continuous range

Continuous representations of trust often represent trust as a value between 0 and 100, or perhaps 0 and 1. As any continuous range can be converted to any other, given an appropriate transform function, there is little practical difference between them. Continuous ranges are rarely used when humans are asked to rate trust, perhaps because they can be difficult to communicate and measure in practice.

Marsh [1994] chose to represent trust and distrust in a continuous measure in the range $[-1, 1]$, where -1 denotes complete distrust, 1 denotes complete trust, and 0 denotes a neutral stance. Interestingly, Marsh noted that at the edges of this range (-1 or 1), the labels we assign to this measure break down, for example if we define trust to require some degree of risk to be involved, complete trust implies that there is no longer any risk, then the choice is clearly no longer a question of trust. While we agree with this observation, we believe that for practical purposes this technicality can be safely ignored.

The trustworthiness of in information is closely related to belief in information; we would not trust information in which we do not believe, unless perhaps the information comes with strong assurances from the trustworthy source. Marsh's representation of trust can be also be applied to the task of modelling belief; where the three points in the range ($[-1, 0, 1]$) denote belief that it is false (disbelief), belief that the information is true, and a mid-point of no conviction either way.

3.3.1 Untrustworthiness, Disbelief and Uncertainty

When considering information with the aim of making better trusting decisions it is easy to forget the antonyms of trustworthiness and belief: untrustworthiness and disbelief, respectively. If one considers an more active target of trust, such as an agent or person, we are more likely to remember that they may be untrustworthy.

Hartig [2009a] models trustworthiness on a per-statement basis, proposing a similar representation to that of Marsh [1994] but adds to the interpretation of the range. For a statement in Hartig's model, the value of the measure indicates belief or disbelief for the values 1 and -1 respectively, while between these the uncertainty increases such that a value of 0 represents complete uncertainty. This interpretation, as illustrated by Figure 3.1, makes intuitive sense, as one can only completely believe or disbelieve in a statement if one is also completely certain of one's stance. Conversely, if one neither believes nor disbelieves in a statement, one must be wholly uncertain of its validity.

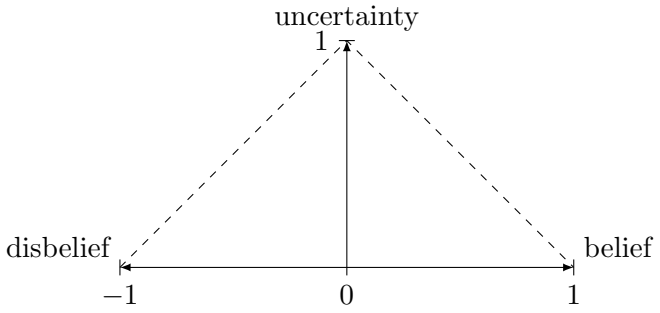


Figure 3.1: Meaning of trust values. Reproduced from [Hartig 2009a]

The exact function of the line illustrated in Figure 3.1 is highly likely to vary between individuals; some will be more willing to believe in the face of uncertainty than others.

3.4 Artificial Trust Environments

In addition to investigating trust among humans and other animals, research has also explored the challenge of trust in artificial environments. These artificial environments can give us insights into how the trust mechanisms we see today might have evolved. Simulated environments and populations give us the opportunity to explore the effects that different changes in behaviour may have on the bonds of trust within a society.

3.4.1 Trust in Multi-Agent Systems

Trust is an active area of research in Multi-Agent Systems (MAS) or Distributed Artificial Intelligence (DAI) research, as, in some systems, the benevolence and co-operation of agents cannot be relied upon. In the field of Multi-Agent Systems (MAS) the term 'Agent' is used to denote a hardware or software-based computer system which exhibits autonomy, social ability, reactivity and pro-activeness [Wooldridge and Jennings 1995]. Research on trust from the multi-agent systems perspective is particularly valuable as it is both analytic and synthetic; environments are constructed, and their emergent properties and behaviour are analysed. In these controlled simulations, it is possible to analyse the macro effects of small changes in incentives or behaviour.

3.4.1.1 Trust Strategies

O'Hara et al. [2004] described the costs and risks of five basic strategies, which serve as a good starting point for consideration:

Optimistic trust systems The optimistic strategy assumes all agents are trustworthy until proven otherwise. In environments where the benefits of co-operation are large, or the costs of betrayal are relatively small, risk may be considered low, and the gains from trusting by default are likely to outweigh the gains of distrust. Systems in these environments are likely to benefit from decentralisation and self-policing mechanisms, to ensure that the shared ethics of the user population determines the policies of trust [Kamvar et al. 2003].

Pure optimism, as a strategy, has very low operational costs, but in non-ideal environments does not perform well. It is, however, useful in bootstrapping systems where there is not yet sufficient information to perform a more complex strategy. Optimism also performs well in distributed environments, as there is no central point of failure.

Pessimistic trust systems Pessimistic strategies, conversely, assume all agents are untrustworthy until proven otherwise. The operational costs of pessimism are as low

as that of optimism, as they share a common simplicity. It also is as well suited to distributed environments as optimistic strategies.

Generally pessimistic strategies are employed in tandem with a whitelist which lists known trusted agents; otherwise a pessimistic agent cannot ever make a trusting decision. While pessimistic strategies will have low, if not zero, deficiency costs, their opportunity costs are likely to be high.

Centralised trust systems Centralised trust strategies delegate the investigation of trustworthiness to a central authority. Centralised strategies do not preclude the need for trust judgements; an agent must instead weigh whether or not to trust the judgements of the central authority. The central authority incurs the costs of investigating the trustworthiness of each agent, but, at a system-level, prevents the duplication of effort through repeated investigations.

By their nature, centralised strategies introduce a central point of failure, however, it can be mitigated to a certain degree so long as there exist multiple, competing, authorities for the same domain. So long as a healthy competitive environment is maintained, having competing authorities may encourage the reduction of costs and the increase of efficiency.

Trust investigative systems Investigative strategies seek to reduce uncertainty, and thereby risk, through active investigation. Investigative agents actively seek information on an unknown agent from third parties, undertaking extra computational tasks in order to determine the trustworthiness of an unknown agent. Iterative credential disclosure negotiation is another example of trust investigation, where both parties are concurrently investigating each other [Winslett et al. 2002].

The operational and opportunity costs of investigative strategies are high; investigating other agents requires time and resources which may cause opportunities to be missed. Deficiency costs are likely to be low, as good investigation ought to have reduced the risk of interaction with malicious agents.

Transitive trust systems Transitive trust systems exploit the trust network built by the links formed by interactions between individual agents to aggregate reputation reports as a single reputation metric [Richardson et al. 2003]. Small-world phenomena, which mean that any two people in a social network may be linked by a relatively short chain of acquaintances, are the main reason transitive trust systems can be effective. However, the degree to which trust may be considered transitive is a contested issue, firstly because the context of trust is generally not modelled for each link, and secondly because the nature of each link in a chain ought to be subtly different; i.e. the difference

between trusting someone to do X, trusting someone to recommend someone to do X and trusting someone to recommend someone who can recommend someone to do X.

The costs of running a transitive trust system depends on the properties of the network in question; larger networks are more likely to find a link between any two individuals, however, risk rises as the trust chain length increases. Transitive trust systems are susceptible to false recommendations, however, over repeated iterations it should be possible to isolate the sources of bad information. The permanence of identities is an important factor here; if it is trivial to abandon tainted identities and coin new ones, malicious agents will be free to supply false recommendations without recrimination, and will likely over time poison the wider network.

3.4.1.2 Reputation Mechanisms

Reputation-based trust systems use reported experience from others to reach a decision of trust about another entity. As well as providing a quality on which to evaluate unknown agents, and generally improving judgements of trustworthiness, functioning reputation systems provide an incentive for good behaviour, and a disincentive against bad. Reputation systems may be constructed with varying degrees of centralisation and distribution. However, the exact balance needed for such a system to function effectively at Web scale, while remaining robust against malice or deception, is difficult to judge.

In the multi-agent systems literature, Huynh et al. [2006] present Certified Reputation (CR), a trustworthiness evaluation mechanism which is informed by reputation information provided by prospective interaction partners. The prospective partner may provide third-party references about its previous performance in order to prove its reliability. The authors acknowledge the inherent problem in relying on third-party referee information sources, and include in CR the means to record and evaluate the credibility of a referee based on the historical accuracy of its recommendations. This strategy appears reasonable, though further work is needed to see how this might perform at Web scale, with a constant turnover in users and actively malicious behaviour.

The Hierarchical And Bayesian Inferred Trust Model (HABIT) trust model presented by Teacy et al. [2008] describes an internal model for representing reported interaction outcomes, a raw measure of reputation, from a variety of sources. The authors have recognised that dishonest agents may report reputation inaccurately and that even benevolent agents may judge the outcomes of an interaction differently. Thus, the key contribution of the HABIT model is that it facilitates the analysis of the the reputation reporter as well as the agent under discussion. From this, the relationships between actual and reported interaction outcomes can be analysed, and generalisations can be made between different sources and contexts.

3.4.2 Trust in a Digital Context

In the past, trusting decisions were generally grounded in ones personal first-hand experiences. As communication technologies have advanced, we have stretched the reach of trust across ever wider distances to support our global economy. Along the way, these communication mediums have, to some extent, lost contact with the factors on which we would normally have based trusting decisions. Mail fraud schemes and telephone scams have been some of the symptoms of this, and anti-fraud legislation and distance selling regulations have been our attempts to address them.

Electronic communications domains such as the Internet and email have stretched the limits of trust further still. Aside from the content of the communication, very few features of identity remain, making it ever harder to judge the identity of remote correspondents. The design of the internet infrastructure also gives no assurance that messages passed among its routers have not been read and/or modified along their travels. In face-to-face communication, one has at least a chance of catching an eavesdropper in the act, and one can be certain that one's words have not been modified in the time they pass from one's lips to the ears of the person receiving them.

As we described in Section 3.2.2.1, we observe certain 'signals' as indicators of trustworthiness. The transition of human interactions to a digital context has forced us to create and adopt new signals. Unfortunately, as copying is trivial in a digital context, imitation is also trivial. This has meant that certain signals which are purely informational – i.e. those not tied to independently observable characteristics – are quickly subverted by untrustworthy actors.

In order to combat this, and to provide assurances of authenticity in a digital context, we have turned to cryptography.

3.4.2.1 Cryptography

Throughout history, cryptographic algorithms have been used to provide secure and private communications. Cryptographic algorithms do not directly enable trust, but they provide four principal assurances, each of which can support trust in certain situations:

Confidentiality

Confidentiality requires that no unauthorised parties may eavesdrop on an exchange.

The assurance of confidentiality allows agents to assume that their communication will not cause agents other than the recipient to change their behaviour, thereby reducing the risk of incorrect action.

Authentication

Authentication ensures that the other party is who they claim to be.

Judgements of trustworthiness often depend on the the identity of the agent; authentication provides a means of verifying identity.

Integrity

The message received has not been tampered with since it was sent by the other party.

Integrity assurances allow agents to assume that they are not operating on information which has been maliciously tampered with.

Non-repudiation

The sender cannot later claim not to have sent the message.

Non-repudiation provides an insurance policy of sorts, the assurance that either side cannot later deny what they have said.

There are two main classes of cryptographic encryption algorithms: symmetric and asymmetric. Symmetric algorithms encrypt a message with a secret key, and can decrypt the resulting cipher-text using the same key. Asymmetric algorithms employ key-pairs, of which either may be used to encrypt information, however, only the key not used to encrypt the message can decrypt the cipher-text.

Generally, one half of an asymmetric key-pair is designated the public key and disseminated freely, and the other half, the private key, is kept only by its owner. Assuming that an agent A acquired the public key for agent B from a trusted source, A can be confident that any message sent to B which is encrypted by B's public key cannot be read by anyone except B. Building on this, bidirectional encrypted communication using asymmetric encryption can be achieved using 2 sets of key-pairs.

Symmetric key algorithms rely on the secrecy of the secret key to assure confidentiality, authentication and integrity; they only weakly provide non-repudiation, as any party privy to the secret key could have sent a particular message. Asymmetric cryptography algorithms generally rely on one half of the key remaining private and the conjecture that it is computationally intractable to compute any private key that corresponds to a given public key, and that no two individuals will generate the same key-pair randomly.

All four assurances are provided more robustly by asymmetric key algorithms than by symmetric key algorithms; both parties must be privy to the key in order to use a symmetric algorithm and thus either party could compromise the communication. In contrast, if the private key of one key-pair from an asymmetrically encrypted communication were compromised, only the communications in one direction would be compromised.

3.4.2.2 Public Key Cryptography

Public key cryptography is a cryptographic approach whose distinguishing characteristic is the use of asymmetric encryption algorithms[2].

Integrity can also be guaranteed for non-encrypted information using public-key cryptography. A one-directional hash of a resource is generated using a known algorithm and the output is encrypted with a private key. From this, anyone possessing the document, the encrypted hash and the public key of the signatory can verify whether the document is unmodified, providing they can perform the same one-way hashing algorithm. This process is generally know as cryptographically, digitally or electronically 'signing' a document, thereby producing a cryptographic, digital or electronic signature.

Non-repudiation is also achievable with public key cryptography by requiring the publisher to cryptographically sign the document before publishing. However, it relies on the assumption that the public key is intrinsically tied to the identity of the signing individual, thus the individual cannot also claim to not have possessed that key pair. In practice, a false denial would be unlikely to go unnoticed as others would have witnessed the individual's use of the disputed key-pair, however it would not be possible to rule out a claim that the key was compromised by a third party.

3.4.2.3 Public Key Infrastructure

The Public Key Infrastructure (PKI) is a set of technologies, policies and procedures used to create, manage, store, distribute, and revoke digital certificates. Public Key Infrastructure (PKI) is built upon the ability of public key cryptography to sign documents, and the affordance that public keys may be published and disseminated widely.

Each PKI scheme has a root Certificate Authority (CA) which maintains and underwrites a collection of unique user identities. In practice, it does so by signing a certificate document containing a user's public key and other information intended to verify the identity of the user. The public nature of PKI schemes deter repudiation of key pairs because a third party, the key signer, can vouch for who that the key was held by.

The most commonly used PKI scheme is the X.509 system specified by the ITU Telecommunication Standardization Sector (ITU-T), which describes a strict hierarchical system of certificate authorities [REC. 2008]. Modern operating systems and Web Browsers are commonly distributed with a set of certificates for root Certificate Authorities (CAs) who are considered trustworthy by the software vendors. These root CAs underwrite the

[2]Commonly, when communicating using asymmetric key encryption, exchanges will actually use a per-session symmetric key, which was negotiated securely using asymmetric key algorithms. This is a optimisation measure as asymmetric cryptography algorithms are generally significantly more computationally expensive than symmetric key encryption

identities of secondary CAs, who either underwrite further CAs or sign identity certificates for users, for things such as internet host names, or email addresses. PKI creates a chain of trust through each CA and is a primarily centralised trust strategy, although the central authority does not need to be involved in each interaction.

The degree of information in an underwritten identity varies considerably between types of certificate. Extended Validation (EV) Certificates offer a stronger layer of assurance compared to standard X.509 certificates; users requesting such a certificate must undergo more extensive investigation of their legal identity and domain name ownership before issuers will issue one, and issuers of EVs must also undergo periodic audits by a recognised and independent third party organisation.

The PKI can be regarded as a set of centralised authorities in which we place our trust. Recently, a series of high-profile security breaches at certain certificate authorities has thrown some doubt onto the trustworthiness of this system [British Broadcasting Corporation 2011, Arthur 2011]. The certificate authorities are perhaps too far divorced from the users who (usually unknowingly) rely on their cryptographic assurances. As a result it can be difficult for users to have confidence in organisations which they are unlikely to ever met, and which does not have a direct incentive to care about their wellbeing.

3.4.2.4 The Web of Trust

In contrast to the hierarchical nature of PKI, the Web of Trust is a grassroots, decentralised, peer-to-peer trust system. Rather than build the foundation of trust on a few select organisations, the Web of Trust relies on a network of people who vouch for each other's credentials.

The Web of Trust predates the recent, high profile, breaches of certificate authorities by many years. The Web of Trust is built on the OpenPGP standards [Callas et al. 2007] which were evolved from the existing Pretty Good Privacy (PGP) computer program, developed by Phil Zimmermann.

The Web of Trust is extended by creating a certificate of a user's identity (including public key and owner information) and then having it cryptographically signed by other users who, by doing so, endorse the association of that public key with the person or entity described in the certificate. While key signing may often be done on a one-off basis, a common method of building this social network of trust is through key signing parties, where a number of users meet to verify each other's credentials.

PGP is often used in email clients as a security and/or privacy measure, but is not often used elsewhere; PKI is, on the other hand, supported by almost every operating system

and Web Browser application. In contrast to PKI, the Web of Trust is a decentralised network and has much in common with transitive trust strategies.

3.4.3 Trust on the Web

The World Wide Web is an unmoderated publishing system with very few technical restrictions on content publication. The Web has no inbuilt mechanism to restrict what content can be published; social norms and the threat of prosecution are the primary forces moderating publised content. As a result, almost anyone can publish anything on the Web (see Section 4.1). As a consequence of this, when browsing the Web, users are faced with questions of trust for every page they visit [Golbeck 2006]. These questions range from decisions such as whether or not the information on the page is true, to complex and critical decisions such as whether it is safe to engage in a financial transaction with a particular website. These questions are all the more difficult because, as we discussed in Section 3.4.2, in digital environments it is easy to copy the appearance of trustworthiness.

The study of trust is continually influenced by our observations of human society; models of human trusting behaviour offer insights into the inner workings of existing trust systems. McKnight et al. [2002] present an e-commerce trust model based on the Theory of Reasoned Action (TRA); they posit that trusting beliefs (perceptions of specific Web vendor attributes) lead to trusting intentions (intention to engage in trust-related behaviours with a specific Web vendor), which in turn result in trust-related behaviours. Figure 3.2 illustrates the relationships between these high-level constructs, and Figure 3.3 illustrates how these are composed of more directly measurable sub-constructs. Models of trust in e-commerce environments, such as the aforementioned model, offer insights into how trusting decisions are made in digital environments, and through analysing them we can hope to identify how cases of misplaced trust might be prevented.

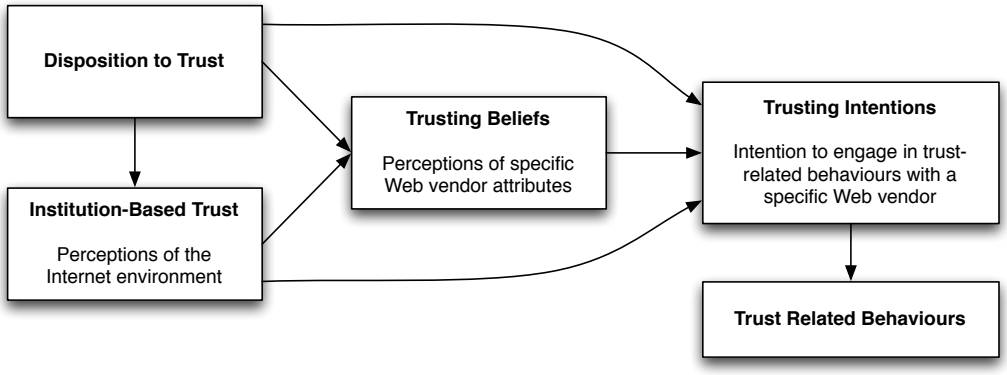


FIGURE 3.2: McKnight et al.'s Web Trust Model - Overview. [McKnight et al. 2002]

From a technological point of view, the Web relies on the Public Key Infrastructure to provide a grounding of trust, assuring the identity of Web servers, and confidentiality

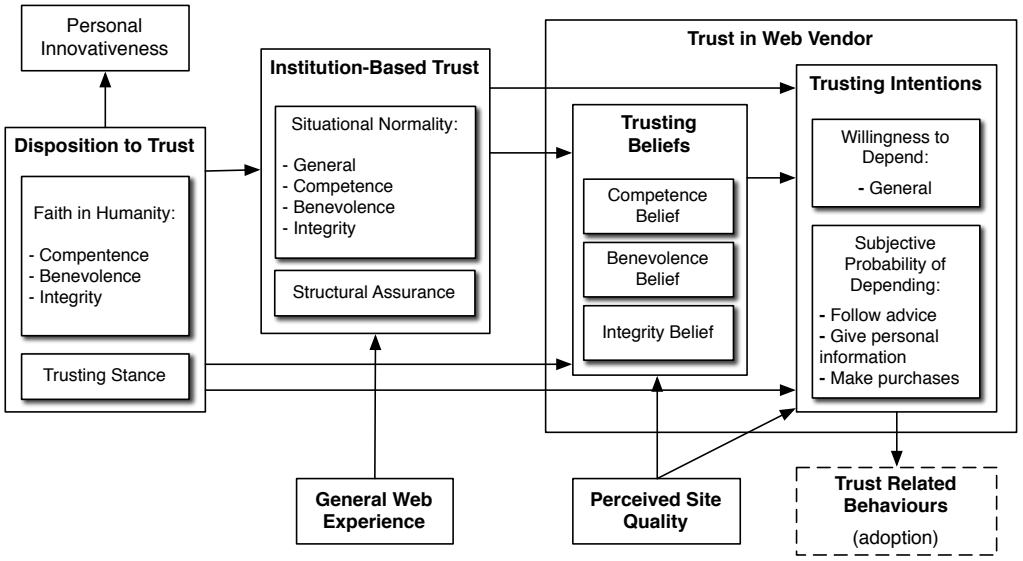


FIGURE 3.3: McKnight et al.'s Web Trust Model - Constructs and Nomological Network. [McKnight et al. 2002]

of communications. Unfortunately its use is optional and users often do not recognise its presence, or absence, nor understand the implications thereof. Dhamija et al. [2006] found that only 23% of participants in their study of online phishing looked at the address bar of the Web browser when asked to comment on the legitimacy of a website. Further, a full 59% of participants did not pay attention to the presence of `https://` in the address bar, which denotes a Hypertext Transfer Protocol Secure (HTTPS) connection.

It is clear from their work that visual cues are a significant factor in people's assessment of Web site identity, indeed 23% of their participants looked only at the content of the website itself to evaluate its authenticity. Wang and Emurian [2005] summarise the design features which have been shown to induce trusting attitudes in users into four categories:

Graphic Design
: Graphical first impressions and professional appearance.

Structure Design
: Information organisation and accessibility, navigational simplicity and consistency.

Content Design
: Informational content including branding, seals-of-approval and professional assurances.

Social-cue Design
: Human touches such as photos of people, live chat, and video pitches.

Unfortunately, while these features demonstrate a certain level of investment and care in the design and upkeep of a website, they are a poor measure of identity. This is due

to the fact that, as we discussed in Section 3.4.2, it is trivial to copy them verbatim from another website.

3.4.3.1 Online marketplaces

Online marketplaces, such as the auction website eBay[3] and Amazon.com's[4] Marketplace, have engineered and integrated reputation systems into their marketplace as a means of enforcing and encouraging trustworthiness among their users. Resnick and Zeckhauser [2002] examine the reputation system in use by eBay, concluding that it does appear to have the desired effect of increasing trustworthy behaviour, even though it may not be mathematically reliable or sound; they postulate that it continues to function, despite being unsound in a mathematical sense, due to the fact that its users believe that it works.

Both these marketplace reputation systems are centralised systems which solicit and aggregate reviews of agents' behaviour.

3.5 Summary

In summary, in this chapter we have examined the concept of trust, the forms in which it manifests, and a range of motivations for trusting. We conclude that 'trust as confidence' is the form of trust which we wish to promote on the Semantic Web (Section 3.1).

Judging trustworthiness is the primary challenge of trust; we divide it into assessing the trustworthiness of information and of actors. Although they are closely related, these two challenges have separate problems (Section 3.2).

Next, we reviewed different methods by which previous research has quantified trust (Section 3.3). Finally, we discussed research into trust in artificial environments, from entirely simulated environments, to the World Wide Web (Section 3.4). We also discussed the technologies which provide the crucial foundations for trust in these domains, including cryptography (Section 3.4.2.1), and the Public Key Infrastructure (Section 3.4.2.3).

Against this background, in the next chapter, we focus on the Semantic Web and the role which trust plays within it. We discuss the current state of the art in that area, and identify a number of research challenges.

[3] eBay.com: See `http://www.ebay.com`
[4] Amazon.com: See `http://www.amazon.com`

Chapter 4

The Semantic Web

Before we can discuss trust in the context of the Semantic Web, it is important that we first understand the nature, purpose and principles of the Semantic Web. To that end, we begin this chapter with a brief history of the World Wide Web, the technology stack which underlies the Semantic Web and which has shaped many of its design decisions (Section 4.1). We then consider the Semantic Web in detail (Section 4.2), discussing its core formats and protocols (Sections 4.2.1, 4.2.2.3), and the Linked Data movement (Section 4.2.3). Against this background, in Section 4.3, we return to the study of trust, now in the context of the Semantic Web. Finally, before summarising this chapter, Section 4.4 concludes with a review of the relevant research challenges which we have identified in this chapter.

4.1 Hypertext and the World Wide Web

The World Wide Web (or colloquially, the Web) [Berners-Lee et al. 1992] was first proposed in March 1989 by Sir Tim Berners-Lee at The European Organization for Nuclear Research (CERN)[1] [Berners-Lee 1989] as a collaboration tool for the High-Energy Physics research community.

The World Wide Web is, at its core, a hypertext information system inter-linked across a global network of computers, the Internet. Hypertext is a form of information presentation that is intended to overcome limitations of linear text — text which has a predefined narrative which runs from beginning to end. Rather than remaining static as plain text must, hypertext makes possible a dynamic organisation of information through links and connections (called hyperlinks). Hypertext systems often go beyond this to incorporate interactive elements (for example when a user 'clicks' or 'hovers' over

[1] CERN: See `http://www.cern.ch/`

an element on the page with a cursor), multiple media (such as images, sound and video) or adaptive content such that they display different information for different users.

The Internet has become an open communication network on which anyone who has the means to communicate may communicate. The cost of Internet access varies across regions, but prices are generally decreasing, and one can often find free access in public facilities such as libraries. The barriers to internet access are low enough that almost anyone can take advantage of the information or services which are offered on the Web.

The only prerequisite to publishing or offering services on the World Wide Web is a publicly routable IP address. Although most do, some low-cost Internet access packages do not include a public IP address. Public IP address can also be leased from online service providers, often in a shared form at a lower cost. In addition, there are now many online services offering free Web publishing platforms, often supported by advertising or services. As a net result of these different factors, any reasonably determined person can publish information or offer services on the Web.

Web pages are requested and transferred over the Hypertext Transfer Protocol (HTTP) [Fielding et al. 1999] — or its counterpart HTTPS [Rescorla 2000] which employs Transport Layer Security (TLS) — and their content is typically described using Berners-Lee et al.'s Hypertext Markup Language (HTML).

The Hypertext Markup Language has undergone a number of revisions: it was first formally specified, at version 2.0, by the Internet Engineering Task Force (IETF) [Berners-Lee and Connolly 1995], and since 1997 updates to the HTML specification have been standardised by the World Wide Web Consortium (W3C)[2]. Extensible Hypertext Markup Language (XHTML) [McCarron and Ishikawa 2010] is a variant of HTML which is based on the Extensible Markup Language (XML), and has a more strict serialisation format than HTML. Recently the W3C's HTML working group has been working towards HTML version 5, which adds many new features including new interactive element types, new semantic markup elements and also refines some existing elements.

Seeking to re-purpose the Web as a universal medium for information and knowledge exchange, the W3C began promoting the research and development of new technologies towards the design of a new 'Semantic' Web [Berners-Lee et al. 2001].

4.2 The Semantic Web

The Semantic Web movement is centred around the desire to expose data on the Web in machine understandable formats, and interlink datasets with each other to create a Web of data as powerful as the World Wide Web is today. As information and meaning

[2] W3C: See `http://www.w3.org/`

are the focus of the Semantic Web, Semantic Web documents do not include the visual formatting information present in traditional HTML documents. The choice of how to visualise information (e.g. as a table, network graph or bar chart) is left as an exercise for the application designer processing the data.

As we discussed in Section 1.1, Berners-Lee et al. [2001]'s article described a World Wide Web where personal software agents could seamlessly, and autonomously, access and exchange data with one another, acting intelligently and benevolently on their owners' behalf. In a distributed environment, where information is exchanged so frequently, and is, more often than not, consumed by autonomous machines – artificial agents – the validity of data becomes increasingly important. As such, security and trust are crucial to the Semantic Web vision, and it is important that we consider their implications to the design of the Semantic Web [Finin and Joshi 2002].

The design of the Web took the simplest features of of hypertext systems, added globally scoped Uniform Resource Identifiers (URIs)(see Section 4.2.1) and relaxed link consistency constraints. Social dynamics are one of the factors which has made the Web so successful, compared to other hypertext systems. Without the overhead of link consistency and the collaborative element it entails, people needed only to agree on the few design constraints of HTML and Hypertext Transfer Protocol (HTTP) in order to get a significant return on their investment. By analogy, the Semantic Web borrowed from simple database and logic systems, and used URIs for column names and symbol terms. [Connolly 2006]

Traditionally, hypertext systems modelled links as either uni- or bidirectional pointers between documents, and in some cases, as typed relationships [Halasz and Schwartz 1994, Halasz et al. 1987]. The significance of adopting simple unidirectional pointers and relaxing link consistency constraints should not be underestimated; more complex link systems require significant degrees of cooperation between systems and people. Simple links greatly simplify the social dynamics involved in creating new Web content, contributing to the Web's potential for exponential growth. Whether there are constraints which need relaxing or social norms which will result in exponential growth are still open questions for the Semantic Web. By using HTTP, a core Web technology, as its delivery platform of choice, the Semantic Web inherits much of the open nature of the Web, and as a result there are few (if any) constraints left to relax.

As an example of the sort of social dynamic that Connolly describes, consider the rise of blogging (a contraction of 'Web logging'). Web logs, or blogs, are websites on which people publish serial entries on any topic, such as descriptions of events and personal experiences, or re-broadcast content which the author finds appealing. The sudden increase in the growth rate of blogging is linked, in part, to the addition of a single feature known as 'trackbacks' [Six Apart 2002] to most common blogging platforms. The trackback system notifies another blog when you directly link to one of its articles.

Although at first glance this seems simple and unexceptional, it dramatically improved the social dynamic of comment on the Web. Instead of comment being constrained to the silo of an individual post, trackback links created an open network of comment across the Web, which encouraged communities to form around common interests.

We do not expect there to be a single feature or application which will alone spur the explosive growth of the Semantic Web, rather a collection of applications and datasets which demonstrate the value of the Semantic Web and bring a variety of incentives for different classes of users to contribute. Taking inspiration from the popularity of blogging; if Semantic Web publishing platforms were able to notify each other when data which links datasets was asserted, it might incentivise the creation of intra-dataset links.

Salvadores et al. [2010] present a back-link discovery service which can be queried to retrieve a list of documents which mention a particular URI; however, its scope is limited to the documents the service has the time and resources to crawl. A trackback-style Semantic Pingback back-link notification system is presented by Tramp et al. [2010]. While this may be a more decentralised and timely alternative than Salvadores et al.'s back-link service, it requires uptake of the Pingback system by all parties, whereas Salvadores et al.'s service does not. Further, publishers newly adding support for Semantic Pingback must wait for back-link information to be reported over time.

In this scenario, Salvadores et al.'s service would provide an efficient means of bootstrapping the back-link database. In addition, pingback can only notify publishers interested in back-link; consumers of data searching for foreign references do not benefit. Alone, back-link services are unlikely to spur the exponential growth of the Semantic Web as they did for blogging, however, they may prove to be part of a larger puzzle of success.

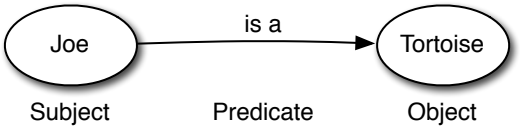
4.2.1 The Resource Description Framework

The core Semantic Web information language, Resource Description Framework (RDF) [Hayes 2004, Manola and Miller 2004], was developed and standardised by the W3C as a platform in which to model information. RDF is primarily a knowledge representation model, based around making factual assertions about resources in the form of subject-predicate-object statements, which are known as "triples" or "statements" in RDF terminology.

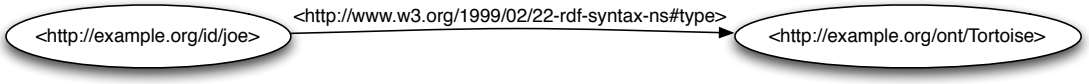
The components of a triple may be one of the following 3 types: [Hayes 2004]

- **Uniform Resource Identifier (URI)** [Berners-Lee et al. 2005]

  Uniform Resource Identifiers are a string of characters used to identify or name a resource on the Internet. Web addresses, or more precisely, HTTP URLs (e.g. `http://www.example.com/`) are the most common type of URI, probably followed closely



(a) A statement represented as a triple



(b) A statement represented as an RDF triple

FIGURE 4.1: The statement "Joe is a Tortoise" represented as a conceptual triple (a), and as an RDF triple (b)

by the `mailto:` URI scheme. URIs encompass URL schemes — which provide a means to de-reference them and resolve an identifier into an informational representation — and Uniform Resource Name (URN) schemes — which are intended to identify resources and do not specify any means of location, such as the ISBN Uniform Resource Name (URN) namespace [Hakala and Walravens 2001].

URIs may form any part of a triple.

- **Anonymous Resources (Blank Nodes or BNodes)**

  Blank Nodes are anonymous identifiers intended for use when it is considered undesirable to coin a URI identifying a particular resource. Blank Nodes be assigned custom identifiers in some serialisations of RDF models, however, these are for convenience only. These identifiers convey no meaning, and cannot be referenced externally as they are valid only within the scope of a particular serialisation.

  Blank Nodes may be used as the subject or object of a triple.

- **Literal Values**

  Literals are a string of characters representing data in a given encoding. In contrast to the other types, literals have two optional attributes in addition to their value: (i) a datatype - denoting, for example, that the value is an integer - and (ii) a language.

  Literals may be found only in the object component of a triple.

Figure 4.1 illustrates a simple statement represented as a triple. Note that in this example the RDF representation uses a URI for each component of the triple.

The advantages of RDF over other Web-friendly technologies, such as XML [Bray et al. 2008] or JavaScript Object Notation (JSON) [Crockford 2006], can be unclear to some. RDF is primarily a knowledge representation model, whereas XML and JSON are data formats. More specifically, RDF formats describe factual assertions, whereas XML and

JSON simply encode information, and do not define how to extract factual assertions from that information. That is not to say that XML or JSON cannot describe factual assertions. Rather, in order for those assertions to be widely understood, it would require a consensus on how to encode and interpret such assertions. Crucially, the RDF data model represents such a concensus; it describes a data model which may be encoded in a number of different serialization formats. Looking beyond information representation, the Linked Data movement (Section 4.2.3) offers a second advantage over plain data serialization formats. The Linked Data movement combines RDF with open-access publishing and community conventions in order to create and promote a commons of open data.

4.2.1.1 Serialisation Formats

There are a number of different serialization formats for the RDF data model; the first, and most common, is RDF/XML [Beckett 2004] (Listings 4.1 and 4.2) an XML-based format [Bray et al. 2008]. A family of non-XML serialisation formats, designed to be easier to write by hand, have arisen. These share many characteristics and differ mainly in their expressiveness. The most prominent non-XML formats are Notation 3 (N3) [Berners-Lee 2006b], Turtle [Beckett and Berners-Lee 2008], and N-Triples [Hayes 2004]. Listing 4.3 demonstrates an unabbreviated N-Triples serialisation (line wrapping and indentation is optional and has been added for the sake of presentation).

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:Description rdf:about="http://example.org/id/joe">
        <rdf:type rdf:resource="http://example.org/ont/Tortoise" />
    </rdf:Description>
</rdf:RDF>
```

Listing 4.1: RDF/XML serialization of the RDF graph depicted in Figure 4.1

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:ont="http://example.org/ont/">

    <ont:Tortoise rdf:about="http://example.org/id/joe" />
</rdf:RDF>
```

Listing 4.2: Abbreviated RDF/XML serialization of the RDF graph depicted in Figure 4.1

```
<http://example.org/id/joe>
    <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
        <http://example.org/ont/Tortoise> .
```

Listing 4.3: NTriples serialization of the RDF graph depicted in Figure 4.1

Fragment identifiers identify fragments of a Web page; when resolved in most Web browsers the fragment will become the initial focus of the browsing window. In RDF

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:ont="http://example.org/ont/">

    <ont:Tortoise rdf:ID="joe">
        <has-name>Joe</has-name>
    </ont:Tortoise>
</rdf:RDF>
```

LISTING 4.4: Example use of a fragment identifier

they are interpreted as identifiers scoped within a particular document, this allows one to declare an identifier and authoritatively state information about it in a single file. Listing 4.4 demonstrates the use of a fragment identifier; the identifier `joe` corresponds to the fragment `#joe` within the document. If the document were to be found at the URL `http://example.org/doc` then the full URI for `joe` would be `http://example.org/doc#joe`.

More recently, to facilitate the embedding of Semantic Web information into human-readable web-pages, RDF in Attributes (RDFa) has been standardised [Adida et al. 2008]. Using XML namespaces, it specifies how HTML content should be embellished with extra mark-up to encode Semantic Web information. Figure 4.5 demonstrates how a simple HTML page might be embellished with semantic information using RDFa.

```
<html
    xmlns="http://www.w3.org/1999/xhtml/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <head></head>
    <body>
        <p about="http://example.org/id/joe">
            <span property="rdfs:label">Joe</span>
            is a
            <span rel="rdf:type">
                <span about="http://example.org/ont/Tortoise" property="rdfs:label">↩
tortoise</span>
            </span>
        </p>
    </body>
</html>
```

LISTING 4.5: A small HTML document embellished with RDFa to include the RDF graph depicted in Figure 4.1, and two additional literal triples.

4.2.1.2 Reification and Named Graphs

In knowledge representation 'reification' involves the representation of factual assertions such that they can be referred to by other factual assertions. Reification is an important feature of advanced modelling languages, as it enables the recording of metadata to an unlimited degree.

Listing 4.6 demonstrates the reification of a single RDF triple; one triple becomes four triples sharing a common identifier. In our example, the reified triple is identified by

a blank node denoted by `_:s`. Unfortunately, as the RDF semantics [Hayes 2004] state that neither a triple nor its reification entails the other, the practical applications of reification are limited.

```
_:s rdf:type        rdf:Statement .
_:s rdf:subject     <http://example.org/id/joe> .
_:s rdf:predicate   rdf:type .
_:s rdf:object      <http://example.org/ont/Tortoise> .
```

LISTING 4.6: Reification of the triple depicted in Figure 4.1, in NTriples format

Named graphs [Carroll et al. 2005] are an alternative, or complement, to reification; they allow an RDF graph – a collection of triples – to be identified by a URI. Unlike reification, named graphs do not allow you to identify a single triple, although it is possible to describe a graph containing a single triple.

4.2.1.3 Signed RDF Graphs

Certain applications of RDF require the assurances of authenticity and non-repudiation that cryptographically signed documents afford. There are two approaches to signing and verifying RDF graphs: the first is to sign a serialisation of the subject graph just as you would a normal document, and then reconstruct a copy of the graph from the signed serialisation and compare with the graph to be verified. Verification involves checking whether the graphs are isomorphic, however, the graph isomorphism problem is assumed to exceed the bounds of polynomial time [Köbler et al. 1993] and so cannot be relied upon to be computationally tractable.

The second approach is to translate both graphs into a canonical serialisation and compare checksums of the canonical serialisations, assuming that one of the checksums was extracted from a cryptographic 'signature'. Unfortunately, generating a canonical serialisation of a graph is also a non-trivial problem. Carroll [2003] demonstrated a pragmatic canonicalisation algorithm which makes small, semantically meaningless, additions to an RDF graph in order to ensure that canonicalisation will operate in $O(n \log n)$ time. While this approach slightly increases the size of graph to be signed, we would argue that the reduction of computational complexity justifies the increase.

The W3C has yet to standardise an official algorithm for signing RDF graphs.

4.2.2 Querying, Reasoning and Inference

To complement the information description capabilities of RDF, we have the ontology languages of RDFS, and OWL, which allow more complex semantic expression, as well as the derivation of additional statements through inference. Perhaps most importantly, to enable the advanced querying of RDF datasets, we have the SPARQL query language.

4.2.2.1 RDF Schema

RDF Schema (RDFS) adds basic ontology description features and limited inference to RDF, including classes, sub-classes and global property domains and ranges [Brickley and Guha 2004], and is designed to be a reflexive vocabulary for defining other RDF vocabularies.

4.2.2.2 The Web Ontology Language

The Web Ontology Language (OWL), also standardised by the W3C, offers more expressive and powerful tools for building and reasoning over ontologies. The Web Ontology Language (OWL) comes in three variants, Lite, DL and Full, which offer different trade-offs between expressiveness and computational tractability [McGuinness and van Harmelen 2004]. OWL has been updated by OWL 2, which also has 3 sublanguages: EL — which has polynomial time reasoning complexity, QL — which is designed to make accessing and querying databases easier, and RL — which is a rule subset of OWL 2 [Motik et al. 2009b,a].

4.2.2.3 The SPARQL Protocol and RDF Query Language

In order to construct applications built on Semantic Web technologies, it is desirable to be able to perform queries over RDF data. The W3C has standardised the SPARQL Protocol And RDF Query Language (SPARQL), which provides a means of querying an RDF data store, and a HTTP based protocol for submitting queries, Listing 4.7 demonstrates an example SPARQL query [Prud'hommeaux and Seaborne 2008, Motik et al. 2009a]. A SPARQL query defines an RDF graph pattern which the query processor tries to match in the data store, binding variables to parts of triples where possible.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE {
    ?x foaf:name ?name .
    ?x foaf:mbox ?mbox
}
```

Listing 4.7: An example SPARQL query

Table 4.1 illustrates the results of the SPARQL query from Listing 4.7 on the data in Listing 4.8.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Johnny Lee Outlaw" .
_:a foaf:mbox <mailto:jlow@example.com> .
_:b foaf:name "Peter Goodguy" .
_:b foaf:mbox <mailto:peter@example.org> .
_:c foaf:mbox <mailto:carol@example.org> .
```

Listing 4.8: Example RDF Data in N-Triples format

name	mbox
“Johnny Lee Outlaw”	<mailto:jlow@example.com>
“Peter Goodguy”	<mailto:peter@example.org>

Table 4.1: The output of the query from Listing 4.7 on the data in Listing 4.8

4.2.3 The Web of Linked Data

The Web of Linked Data is not a new Semantic Web, rather a realisation of where the true power of the Semantic Web lies: its ability to link anything to anything. To begin with, most published Semantic Web data came in the form of custom-tailored ontologies contained within individual archives, using fragment identifiers to identify resources. The premise of the Linked Data movement, drawing on observations of how the Web has grown and prospered, is that the value and usefulness of data increases the more it is interlinked with other data [Berners-Lee 2006a, Heath and Bizer 2011]. This has helped motivate the the adoption of Semantic Web technologies and has shifted the focus of Semantic Web research from theoretical research on reasoning systems into more pragmatic areas such as information integration.

Before the value of publishing as Linked Data became clear, there was little difference between many Semantic Web systems and traditional knowledge based systems. Semantic Web technologies may have had some novel features such as a globally resolvable identifier scheme, but if the scope of Semantic Web technology was limited to private reasoning systems the impact of any innovation it demonstrated would be limited.

Th four principles of Linked Data, as proposed by Berners-Lee [2006a], are:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information in RDF.
4. Include links to other URIs, so that they can discover related things.

These basic principles encourage the organic growth of a commons of open data which, through network effects[Katz and Shapiro 1994], increases in value as people bring new

sources online and link them with existing data. Furthermore, through the use of globally scoped identifiers, a community consensus on term meanings can emerge and evolve into a shared vocabulary.

If the principles are not adhered to, Linked Data would be effectively useless; published data would be in wildly varying formats, reside in isolated silos and it would be near impossible to find authoritative information sources. All of these premises are important to the growth of the Web of Linked Data; a publisher which adheres to some but violates others will diminish the value of their data, rendering its discovery and use difficult or impossible. For example: not using URIs or HTTP URIs to identify things forces anyone interested in your data to implement a custom identifier resolution algorithm for your data; not providing useful information when an identifier is resolved prevents anyone from using your data after stumbling upon an identifier from it; and not linking to other datasets makes it difficult to place your data in a context and makes it difficult to integrate with other data.

The W3C's Linking Open Data community project[3] is an incubator project aiming to bootstrap the Semantic Web data commons with Linked Data. In October 2007, Linked Datasets consisted of over two billion RDF triples, which were interlinked by over two million RDF links. By May 2009 this had grown to 4.2 billion RDF triples[4], and around 142 million RDF links. In September 2011 the number of triples had grown to 31 billion, with around 504 million links[5]. The impact and importance of these incubator groups cannot be underestimated; they address the 'chicken and egg' problem that besets the bootstrapping process of most new technologies.

4.2.3.1 Publishing Linked Data

On the Web, HTTP URIs commonly refer to Web pages. This has led to some discussion and dispute as to how URIs should be interpreted on the Semantic Web, which states they refer to 'resources'. In response to this confusion, the W3C Technical Architecture Group (TAG) identified two different types of resource which might be identified by a URI [Lewis 2007]:

- **Information Resources**

  Information resources identified by URIs are resources whose essential characteristics can be conveyed in a message. Pages and documents on the Web, which users of the Web will be familiar with, are information resources: they typically

[3]Linking Open Data community project: See `http://linkeddata.org/`

[4]Linked Data Dataset Statistics: See `http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/DataSets/Statistics`

[5]Linked Data Linking Statistics: See `http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/DataSets/LinkStatistics`

have one or more representations that can be accessed using HTTP. These representations of the resource are used to convey the informational resource in an HTTP message conversation. Dereferencing a URI refers to the act of retrieving a representation of a resource identified by a URI. Applications, such as browsers, render the retrieved representation so that it can be perceived by a user.

- **Non-Information Resources**

  Non-informational resources are those which do not fall into the above category: things which cannot be converted into an informational representation. For example, if one were to dereference a URI representing the Eiffel Tower, one would not expect to receive a physical copy of the Eiffel Tower somehow transmitted to one's computer. As a rule of thumb, all 'real-world objects' that exist outside of the Web are non-information resources.

  It is, however, not unreasonable to expect to receive a concise bounded description[Stickler 2005] of the resource being dereferenced. For example, a description of its characteristics and relationships with other resources. Crucially, the document bearing information about the resource is distinct from the resource itself, and thus we must be able to distinguish between them with separate identifiers. The W3C TAG advised that the HTTP 303 'See Other' redirect code should be used in order to associate an informational resource with a non-information resource [Lewis 2007].

Thus, we have two means of publishing Semantic Web vocabularies: document fragments (hash namespaces) or redirection (slash namespaces). Following this advisory the Semantic Web Best practices working group published a set of recipes describing and demonstrating the different approaches [Berrueta and Phipps 2008].

4.2.3.2 Fragment Publishing

The original means of publishing RDF data, in a single document using fragment identifiers (as described in Section 4.2.1.1), is still a valid method of publishing. Use of a URI containing a fragment identifier implies agreement to information published at the document returned by resolving that URI after the fragment has been removed [Connolly 2006]. It conforms to the principles of Linked Data publishing, however, it is inadvisable to publish large datasets in this manner as it places a significant burden on clients interested in the data: they must load the entire document even if they are only interested in a fragment of it.

For this reason it is advisable to publish only small vocabularies, such as ontologies, in this manner. In practice, ontologies should be cached at an application level, and thus should only need to be retrieved periodically. It may also be appropriate to pre-load

core RDF vocabularies, such as RDF Schema (RDFS), into Semantic Web applications, as they are unlikely to change often and it is unlikely that an application could cope with their changing. It would also prevent unnecessary requests for those documents from being made, unnecessarily burdening the server publishing them.

4.2.3.3 Redirection Publishing

Redirection based publishing is built upon the redirection pattern proposed by the TAG. When URIs representing non-informational resources are resolved over HTTP, as illustrated in Figure 4.2, the HTTP 303 'See Other' response code is employed to redirect the client to the URI of a related informational resource.

Importantly it is good practice for the redirection step to distinguish between requests based on the 'Content-Type' request headers; if the client prefers RDF data, redirect to an appropriate RDF document, or if the client prefers HTML, redirect them to an HTML page [Heath and Bizer 2011, Berrueta and Phipps 2008].

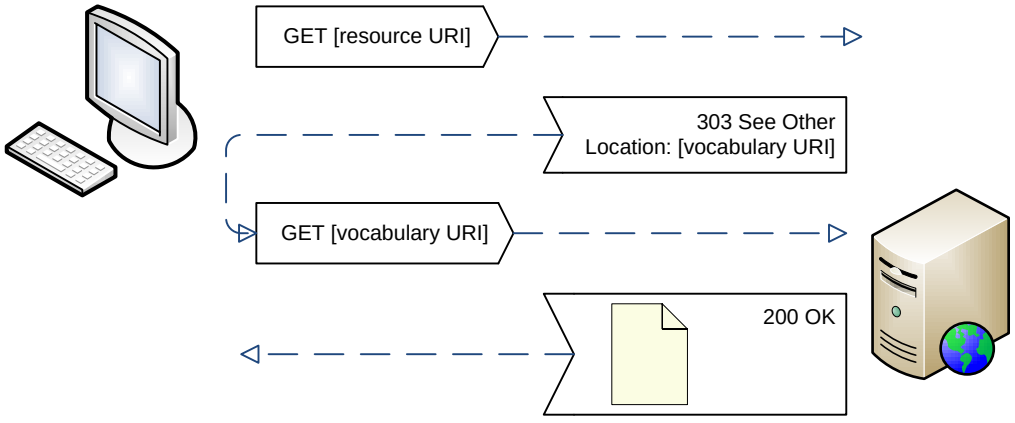


FIGURE 4.2: Illustration of HTTP 303 Redirection publishing technique (Section 4.2.3.3)

Alternatively, in situations where it is undesirable to directly redirect the request, or if the client did not request RDF data, an HTTP 'Link' header could be used to make the client aware of the availability of additional metadata. This negates the need for a client to make two requests to the same URI to find the location for both content and metadata. The HTTP Link header was first specified in HTTP 1.0 [Berners-Lee et al. 1996], however was not included in HTTP 1.1 [Fielding et al. 1999], and has since been formally specified in its own right [Nottingham 2010].

4.3 Trust and the Semantic Web

The Semantic Web inherits many of the features and characteristics of the Web, including its unmoderated publishing nature. We expect that much of what we discussed in

Section 3.4.3, regarding trust on the Web, also applies to the Semantic Web. Indeed, research agendas have expected Semantic Web agents to experience the same quandaries of trust on the Semantic Web as faced by users of the hypertext Web [Kalfoglou et al. 2004]. Against this background, this section focusses on Semantic Web research regarding trust.

4.3.1 Reputation mechanisms

On the reputation front, Semantic Web research has, for the most part, explored the suitability of network-based, transitive trust strategies. Richardson et al. [2003] presented a path-algebra with which network-based trust computation functions can be described. Subsequent works applied similar approaches to a number of different social applications including email filtering and recommender systems [Golbeck 2005, Golbeck and Hendler 2006]. As we discussed in Section 3.4.1.1, the value of trust network analysis as a metric for trustworthiness is a contested issue. The contextuality of trust is ignored by most approaches, and we question the value of assessments which follow long chains as each link in a chain of trust adds additional risk that the assessment is inaccurate.

4.3.2 Policy-based trust

Policy-based trust systems are a means of encoding the criteria of a particular trusting decision as policies in an automated system. Policy-based access controls are often applied within large organisations as they mesh well with organisational structures and responsibilities. Kagal et al. [2003] applied Semantic Web technologies to this field, creating a flexible language for describing rights, credentials, and trust policies, upon which Semantic Web policy based trust systems can be created. Kagal et al. also present a policy engine which harnesses the semantics of their policy language, demonstrating a flexible and dynamic policy management system.

Policy-based systems still demonstrate some undesirable properties, which include the following: First, the verification of any credential almost always involves the credential issuer out of necessity. This necessitates a central point of authority, and of work, which can in turn contribute to problems of security and scalability. Second, the satisfaction of trust policies is often very one-sided – only one party in the exchange has any say on the criteria which the other must meet.

Nejdl et al. [2004] developed the PeerTrust language which goes some way to addressing these problems. PeerTrust recognises that it is desirable for negotiations not to directly involve a centralised credential authority as this is contrary to the decentralised, peer-to-peer nature of the Web and the Semantic Web. Cryptographically signed rules allow them to accomplish this; authorities may issue pre-endorsed credentials through cryptographically signing a set of specially crafted rules. Then, assuming that the other

party recognises the authority of the issuer and that the signature can be verified, the negotiation can include the credential without needing to involve the issuer directly.

PeerTrust also addresses the second shortcoming we highlighted; it considers the whole interaction to be a negotiation by both parties. Both parties may specify criteria which the other must meet before negotiation can advance. In addition to this, PeerTrust recognises that trust policies (or elements thereof) may be sensitive in their own right, and may also have pre-requisites for their disclosure.

The ability to cryptographically sign RDF graphs is a prerequisite for the implementation of decentralised credential systems; unfortunately, signed RDF has not yet been officially standardised (Section 4.2.1.3). Despite this, policy based trust systems may enable us to build sophisticated autonomous agents which can make trust-aware decisions with some semblance of intelligence.

4.3.3 Provenance

As we discussed in Section 3.2.1.2, information provenance is a key factor in assessing information trustworthiness. Provenance remains just as important in the context of the Semantic Web [Halpin 2009]. Good information provenance records should describe the complete history of a dataset, i.e. its original sources, the processes by which sources were combined, and the actors responsible for each process.

Named graphs, which we discussed in Section 4.2.1.2, have been proposed as a means of recording provenance information [Watkins and Nicole 2006]

Information interoperability is another of the primary motivations of the Semantic Web movement. Semantic Web technologies have also been applied to provenance research for the purposes of interoperability.

Hartig and Zhao has contributed a provenance model tailored for the 'Web of Linked Data' [Hartig 2009b, Hartig and Zhao 2010]. Their ontology contains terms in which the actions and concepts of Web interaction patterns can be described.

Hartig and Zhao went on to contribute to the W3C PROV framework as part of the W3C's 'Incubator Group'. The 'PROV' family of documents describes a general-purpose, Semantic Web compatible, provenance framework designed for describing provenance information from any domain [Moreau and Missier 2013].

4.4 Research Challenges

Against the background of our discussion of trust and the Semantic Web, we have identified areas where further research would advance the state of the art:

4.4.1 Identity

Identity is one of the most fundamental parts of any trust system; whether it be identity at an individual or a group level, before anything can be trusted it is first necessary to be able to identify it. Trust on an individual basis is necessary for attitudes such as friendship between two people, whereas trust at a group level is necessary for generalisations such as stereotypes of groups; both of these are desirable mechanisms in robust trust systems.

In systems where agents cannot be relied upon to be truthful, there is risk in relying on their self-stated identities for questions of trust. Most multi-agent systems are either designed with an inbuilt and authoritative identity system, or do not permit deception, and so do not face this problem. In these uncertain environments it is necessary to base identities on observable (and hopefully difficult to spoof) characteristics. We, as humans, generally use visual and audible characteristics to identify people as they are difficult to impersonate well; electronic environments must, however, rely on different characteristics. For example, network addresses and demonstrable possession of cryptographic keys might serve as appropriate characteristics.

The Friend of a Friend (FOAF) ontology is perhaps the most commonly used Semantic Web ontology[Cyganiak 2011] which might be classed as an identity ontology. Using the properties listed in Table 4.2, FOAF facilitates the description of the activities, friendships and acquaintances of people and artificial agents. It is also widely used throughout Semantic Web research [Golbeck 2005, Golbeck and Hendler 2006] and has become the 'de facto' standard for making assertions about people. Unfortunately, as Friend of a Friend (FOAF) documents describe the state of the non-digital world, the information in them is generally not demonstrable. As a result there is little to stop anyone claiming the ownership of a given FOAF identifier, as it is little more than a URI.

Story et al. [2009] proposed the FOAF+SSL extension, which, in a nutshell, allows a FOAF profile to specify a cryptographic public key, and an address on which it can be contacted to prove ownership of the corresponding private key. The Secure Socket Layer (SSL) connection protocol allows both sides of the exchange to present cryptographic certificates, thus the cryptographic identity of both sides of the exchange can be verified. A person can then demonstrate possession of the cryptographic key to demonstrate ownership of the URI (assuming that the key has not been compromised).

WebID is the result of a W3C incubator group[6] which has formed around the FOAF+SSL propsal, to develop vocabularies and protocols for a system of distributed identity for the Semantic Web.

[6] WebID incubator group: See `http://www.w3.org/2005/Incubator/webid/`

account	homepage	pastProject
accountName	icqChatID	phone
accountServiceHomepage	img	plan
age	interest	primaryTopic
aimChatID	isPrimaryTopicOf	publications
based_near	jabberID	schoolHomepage
birthday	knows	sha1
currentProject	lastName	skypeID
depiction	logo	status
depicts	made	surname
dnaChecksum	maker	theme
familyName	mbox	thumbnail
family_name	mbox_sha1sum	tipjar
firstName	member	title
focus	membershipClass	topic
fundedBy	msnChatID	topic_interest
geekcode	myersBriggs	weblog
gender	name	workInfoHomepage
givenName	nick	workplaceHomepage
givenname	openid	yahooChatID
holdsAccount	page	

TABLE 4.2: FOAF properties and relationships

For our objectives, there remain some unsolved issues in FOAF and WebID; 1. FOAF's vocabulary terms apply primarily to people, not software agents 2. there are no disincentives to the discarding of 'tarnished' identities in favour of new ones 3. there remains the problem of coining identities for entities who do not do so themselves, such as Web servers which serve only static RDF documents, or Web servers which serve dynamic RDF documents but do not support WebID .

4.4.2 Reputation

Direct experience is generally accepted as the best grounds on which to make a trusting decision. Unfortunately, as population sizes increase so dramatically do the odds of any two individuals having previously interacted. In addition, any new members of a population will always begin with no direct experiences of other members. Against this background we can see that there is a need for mechanisms which assist in making trusting decisions without a priori experience of the other parties.

Reputation is one such mechanism, one which we as humans are known to employ and which has been studied in a number of disciplines, including Multi-Agent Systems, (see Section 3.4.1.2). As we have yet to see the Semantic Web community widely adopt any particular reputation mechanisms, we consider it be an ongoing challenge. To complicate this challenge, we expect any reputation mechanism to suffer from a similar bootstrap

problem until it achieves a sufficient adoption level. Solving this problem, even partially, would greatly improve any mechanism's chances of adoption.

4.4.3 The Coreference Problem

As people publish more Linked Data, it is inevitable that multiple URIs will be coined for the same resource. However, discovering and publishing URI equivalences is non-trivial, and forms part of the coreference problem.

Coreference is not a new problem, rather the manifestation of an old problem in a new domain; it has been encountered in fields such as natural language processing and Artificial Intelligence (AI). The AI research community often chooses to make the 'unique name assumption', enforcing a one-to-one relationship between resources and identifiers [Russell and Norvig 2003], however, we cannot make this assumption for the Semantic Web.

A centralised registry for URIs could avoid this problem altogether. However there are number of reasons why a centralised approach is not acceptable: (i) it will not scale adequately as the Semantic Web approaches the scale of the Web, (ii) it places a significant burden on a single point of failure, (iii) it invests a single entity with significant power and responsibility, and (iv) crucially, equivalence is generally a subjective decision and thus does not lend itself to centralised administration.

Jaffri et al. [2007] propose and demonstrate a Coreference Service (CRS) which allows the publication an maintenance of coreference information in a single store. They envision that each Linked Data publisher might maintain their own CRS of equivalences they hold to be accurate.

In practice, it remains common for datasets to use the `owl:sameAs` predicate to denote the equivalence of identifiers. Jaffri et al. and Halpin et al. [2010] argue that, due to the subjective nature of this assertion and its implications to reasoning, `owl:sameAs` is often misused. Halpin et al. discuss the subject of similarity at some length, and empirically demonstrate how widely the uses of different similarity predicates vary. This illustration of how subjective things are further supports the case that coreference should be a distributed system.

4.4.4 Provenance

In Section 3.2.1.2 we argued that the trustworthiness of information depends heavily on its provenance. The challenges of recording, maintaining and integrating provenance information remains an active area of research within the Semantic Web (Section 4.3.3).

We still have some way to go before we can expect to see Semantic Web agents which routinely use provenance information in their trusting decisions. It must first become the norm to maintain, and publish, accurate provenance records for RDF data, and also a sufficient portion of the community must converge on a set of vocabularies which are easy to work with or interoperate between. To enable this we should seek to lower the adoption barriers of provenance technology, perhaps by improving support for provenance in popular tools and frameworks, or by finding techniques to reduce the burdens of storing and maintaining provenance information.

4.5 Summary

We began this chapter with a brief introduction to hypertext and the World Wide Web (Section 4.1) before approaching the main focus of this chapter, the Semantic Web (Section 4.2). We then described the fundamental data models underlying Semantic Web technologies (Section 4.2.1), and the more advanced reasoning and query languages which are build upon them (Section 4.2.2). Next, in Section 4.2.3, we discussed the Linked Data movement which has given rise to the Web of Linked Data, and the different publishing practices which are commonly employed. Section 4.3 returns to the subject of trust, discussing Semantic Web research on the subject. Finally, against the background of our discussion so far, Section 4.4 identifies a number of research challenges which arise from our study of trust and the Semantic Web. Principally these challenges are (i) identity, (ii) reputation, (iii) co-reference and (iv) provenance.

Against this background, in the next chapter, we introduce our first contribution towards these challenges; an identity vocabulary for the Semantic Web.

Chapter 5

A Semantic Web Identity Infrastructure

The ability to uniquely identify agents is a key requirement for Semantic Web applications involving trust, provenance or inter-agent interactions. Our use case in Section 2.1 highlighted the importance of identity in inter-agent relations, in particular the need to be able to refer to an entity as a subject of discourse. Without an effective identity infrastructure for the Semantic Web, such discourse is not possible.

Work to date has generally sidestepped the challenges of discovery and representation — assuming that all agent identities are known, and modelling them only as members of the `foaf:Agent` class [Hartig and Zhao 2010, Raimond et al. 2007, Moreau et al. 2010]. If we are to allow our Semantic Web agents off 'in the wild', to crawl the breadth and depth of the Web, and operate autonomously, we must address these challenges.

Due to the nature of the Web, its size, popularity and rate of growth, developing an identity infrastructure based around a centralised registry is not feasible. Instead we seek to provide the necessary tools from which a decentralised identity infrastructure can grow organically. The first, and perhaps most important requirement of an identity infrastructure are vocabularies in which discourse can take place.

In this chapter, we begin by further exploring the challenge of identity description (Section 5.1). We then catalogue the requirements for a Semantic Web agent identity vocabulary (Section 5.2), and discuss related work which has influenced our understanding of and approach to this problem (Section 5.3). Next, we describe our vocabulary in detail, explaining aspects of its design (Section 5.4), demonstrating its usage against one of our earlier use cases (Section 5.6), and reviewing how well it satisfies our original requirements (Section 5.7). Finally, in Section 5.8 we conclude this chapter with a summary.

5.1 Describing Web Agents

As our use case in Section 2.1.1 described, an identity description should allow one agent to recognise another with a high degree of confidence. Identity descriptions must also be communicable without sacrificing recognisability, so that reputation information can be shared among agents to that they can apply it in inter-agent interactions.

There are two different perspectives from which an identity description can be authored: (i) authoritatively – by the subject of the description itself, or (ii) observationally – by an agent which has interacted with or made some observation of the subject .

There are shortcomings to approaches which rely solely on identity descriptions written from one perspective; (i) one can only identify agents which provide their own descriptions, and must trust that the agent included sufficient information for confident identification (ii) the description is limited to information which has been observed by the writer . Thus, to develop a robust identity infrastructure, we seek an approach which can combine information from both perspectives.

While encouraging non-authoritative identity descriptions helps to address the bootstrapping problem an identity infrastructure will face, it will also cause many agents to coin new identifiers for the same entity. This will create an unavoidable coreference problem (Section 4.4.3), and we must take care to encourage the inclusion of information that will enable effective co-reference resolution.

Creating a vocabulary to describe the identity of arbitrary agents of unknown form is most likely an impossible task, since we cannot know the features which adequately describe their identity. Identity descriptions are more likely to combine a number of specialist vocabularies, each maintained by their own set of domain experts.

Pragmatically, we limit the scope of our vocabulary to describing Web agents – agents which understand and communicate over HTTP – which also includes Semantic Web agents. As HTTP clients and servers operate in adherence with open technical specifications, describing them is a much more approachable task. In future we expect that more specialised vocabularies will be combined with ours in order to better describe the nature and identity of the subjects.

5.2 Requirements

In Section 2.1.2, we identified three key challenges for enabling discourse on Semantic Web agents: i) the description of identities, ii) the communication of identity descriptions, iii) the effective comparison of identity descriptions. Marrying these challenges with our research thus far, in this section we identify a number of functional and non-functional requirements for an identity vocabulary and infrastructure.

5.2.1 Functional Requirements

Our decision to limit the scope of our vocabulary to describing only Web-based agents allows us to identify specific functional requirements for the descriptiveness of the vocabulary:

1. **Able to describe a single agent hosted on a single dedicated web server**

   The vocabulary must be able to describe the simplest case: a single web server, hosting only a single web agent.

2. **Able to describe an agent hosted on a shared web server**

   The vocabulary must be able to accurately describe a web agent hosted on a server which also serves a number of other web sites. From the perspective of trustworthiness, shared hosting environments are fundamentally different because: i) the shared server is administered by a third party, ii) flaws in the shared server environment might allow one agent to impersonate another on the same server.

3. **Able to describe a web agent spanning multiple web servers**

   The vocabulary must be able to describe the multi-server environments common in high performance web site architectures. Again, there are specific trust concerns in this environment: i) while they are different servers, they are part of the same agent ii) one or more servers might become compromised, and thus untrustworthy.

4. **Able to describe a web agent which employs public-key encryption**

   The vocabulary must be able to describe the association between an agent and a public-private key pair. Cryptographic keys, if managed with care, can be a strong indicator of identity.

5.2.2 Non-functional Requirements

Next we identify non-functional requirements which cannot be as strictly verified, but are still important in the design of a vocabulary.

Returning briefly to the key challenges we identified in Section 2.1.2; providing that information using our vocabulary can exist in a communicable form (such as the RDF data model), the requirement for communicable descriptions is met. Therefore, we choose not to include this as an explicit requirement.

The issue of coreference, which we discussed in Sections 4.4.3 and 5.1 suggests some non-functional requirements:

5. **Must provide sufficient data for coreference**

   The vocabulary must be sufficiently descriptive so as to enable effective coreference identification.

6. **Must avoid exacerbating the problem of coreference**

   The vocabulary should take measures to discourage the minting of new URIs wherever possible.

In Section 3.4.1.1 and Section 3.4 we reviewed the different strategies and mechanisms which are common in existing trust environments. There are a range of trust strategies, differing in both approach and architecture, which are often suited to particular scenarios or environments. In order not to artificially hamper the effectiveness of different strategies, a strategy neutral vocabulary is desirable.

5.3 Related Work

There are existing specifications which seek to model similar information about web servers. We review them in light of the requirements we identified in the previous section, highlighting their shortcomings where appropriate.

5.3.1 WebID and FOAF

As previously discussed in Section 4.4.1, the FOAF vocabulary allows the description of the activities, friendships and acquaintances of people and artificial agents. WebID builds on FOAF, adding a vocabulary for describing cryptographic key pairs, and an authentication protocol using cryptographic keys. There are two main fronts on which WebID fall short:

Firstly, few of the properties and relationships in the FOAF vocabulary (see Section 4.4.1) can be meaningfully applied to Web agents. Some describe characteristics such agents are unlikely to have, such as depictions or family names, and others describe things which are neither observable or demonstrable – with the exception of some online account identifiers. For this reason, WebID fails to meet requirements #2 through #5.

Secondly, the means of advertising a WebID identifier for casual discovery are insufficient to satisfy requirement #6. An agent can advertise its identity by adding a `foaf:maker` property into response documents. However, this is not appropriate in cases where the agent is not the author of the document in question, or if RDF statements cannot be inserted into the response content. Additionally, if inserted in this way it would confuse the meaning of the `foaf:maker` property. If a document with a `foaf:maker`

statement was received from an agent which had not inserted this information, one might erroneously assume that this was that agent's identifier.

5.3.2 Nokia Web Architecture Vocabulary

The Web Architecture vocabulary [Stickler 2003, Stickler] has similar objectives to ours, in that it is designed to describe web servers and the relationship between them and other resources.

It does not meet our representational requirements #2-4, as it is not able to represent the difference between exclusive and shared use of a Web server. As we discussed in Section 5.2.1, this information may be important to trusting decisions.

The Web Architecture vocabulary predates the practice of Semantic Web URI resolvability. Concequently, its recommendations for URI choices lead to representational problems once we consider URI resolvability.

The vocabulary makes the following recommendations:

> "It is recommended that a web site be denoted by an `http:` URI consisting of only the web authority component, or a path prefix identifying a user-specific web space, followed by a single slash '/' character."[Stickler]

> "It is recommended that a web server be denoted by an `http:` URI consisting of only the web authority component, with no trailing punctuation." [Stickler]

These recommendations take an extensional approach to the coreference problem, giving every web server a canonical URI, and thus passing requirements #5 and #6.

Crucially, in practice, the Web site address (as defined above) is commonly used as the URL of the site's HTML-based homepage, and thus is unlikely to be available for use as an identifier. Using the same URI for both purposes conflates an information resource – the site's homepage – with a non-information resource – the identity of the web server (the service which responds to the HTTP request), which are fundamentally different. However, in the rare case that this URI did not already identify some other resource, it would be an acceptable identifier.

Next, there is a problem when we consider two identifier recommendations in tandem. While at first glance, according to URI specification, the two URIs are different, on closer inspection they are not. The URI specification's scheme-based normalisation rules[Berners-Lee et al. 2005, §6.2.3] state that one must also consult the each scheme

specification, i.e the HTTP URI scheme, in order to normalise a URI. The HTTP specification states that a missing path component should be translated to a single '/' character [Berners-Lee et al. 1996, Fielding et al. 1997, §5.1.2]. Thus, HTTP URLs such as `http://example.org` and `http://example.org/` should be considered equivalent. The result of this is that, as before, one would conflate the identifiers of two fundamentally different resources.

These representational problems render this vocabulary unusable for our purposes.

5.3.3 The IRW ontology

The Identity of Resources on the Web (IRW) ontology [Halpin and Presutti 2009] exists primarily as a tool for modelling the Web architecture, helping to clarify the discussion of proposed changes. It describes the relationship between web servers, informational and non-informational resources, URIs and the results of resolving them. Halpin and Presutti demonstrate this by casting a recent architectural debate in terms of a disagreement over the definition of certain terms in the ontology.

This ontology declares a WebServer class but it has no properties, and thus can only be described in terms of the individual URIs which resolve to it. Whilst technically it meets requirement #1, we cannot see it proving useful in practice as it is impractical to exhaustively create relationships for every URI which resolves to a single server. In addition, the IRW ontology does not consider the delegation of HTTP requests between servers, so it fails to meet requirements #2-4.

In summary, the IRW ontology meets only requirements #1, #5 and #6. Despite these failings, it has a good conceptualisation of the problem, which we draw on for our vocabulary.

5.3.4 The Web of Trust Ontology

The Web of Trust ontology is a lightweight vocabulary for describing the artefacts of public key encryption use (Figure 5.1), and the relationships between them (Figure 5.2). The core classes of the ontology are User, PubKey – a public key, EncryptedDocument – an encrypted document, Endorsement – a document containing a cryptographic signature of another document, SigEvent – an event representing the signing of one public key by another. To lend meaning to its terms, the Web of Trust ontology defines its core classes as subclasses of terms from a number of other ontologies; Figure 5.1 illustrates these relationships.

The Web of Trust ontology should not be confused with the Web of Trust movement which we discussed in Section 3.4.2.4. While they share a very similar name, the Web of

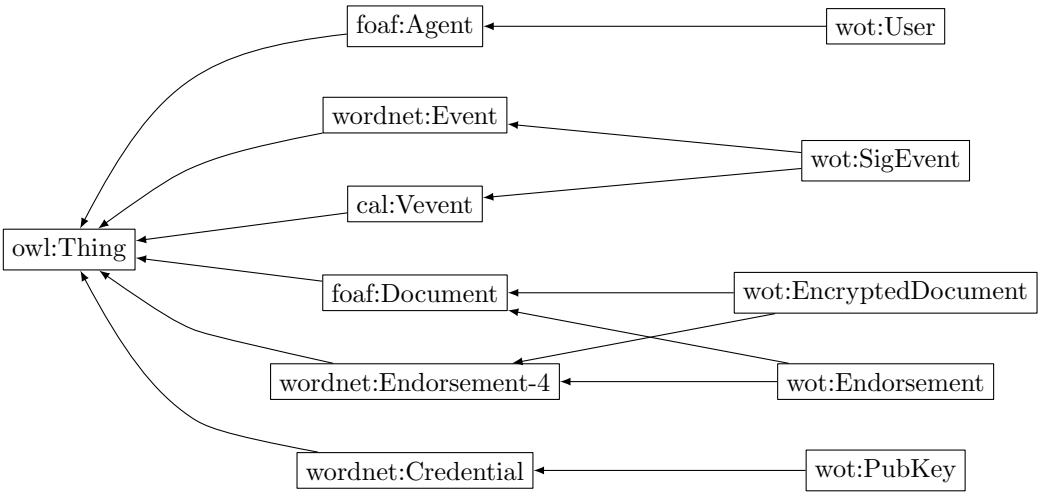


FIGURE 5.1: Web of Trust Ontology class hierarchy

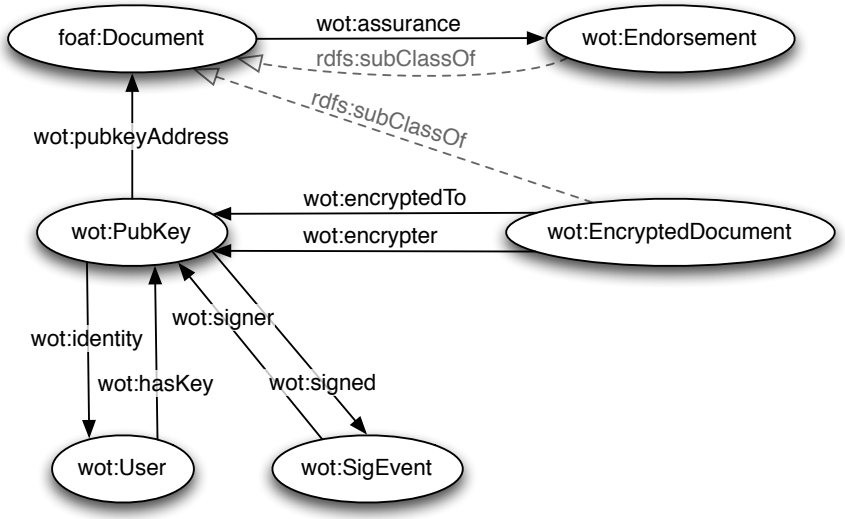


FIGURE 5.2: Web of Trust Ontology object relationships

Trust ontology is equally suited to modelling the hierarchical Public Key Infrastructure as it is a decentralised one.

The Web of Trust ontology does not solve our identity vocabulary problem, it fails to satisfy all but our fourth requirement; the ability to associate a public key with an agent.

5.3.5 POWDER

The Protocol for Web Description Resources (POWDER) is designed to provide a mechanism by which one can record structured metadata about groups of web resources – anything from two web pages, to entire websites. Primarily POWDER is an XML-based format, however Semantic POWDER (POWDER-S) defines transformation rules which

convert POWDER documents into formal OWL semantics. As OWL does not include any means by which to filter or restrict entities based on their URIs, POWDER-S declares an extension to OWL which allows pattern matching over URIs. POWDER-S encodes each filter as a regular expression, using a different expression template for each restriction. The result of this can be seen later in Figure 5.6.

Since it is designed primarily to describe the web resources and their URIs, POWDER does not fully meet any of our requirements for an identity description vocabulary. It does, however, solve the problem of how to group a set of resources, which we require in order to describe certain website structures.

5.3.6 WSDL

The Web Services Description Language (WSDL) allows the description of web-based service types, instances thereof, and the accepted methods by which they may be interacted with. WSDL does support our simplest requirement; defining a service endpoint at a given URI effectively defines an agent with a given contact point. However, while we could likely model request delegation in WSDL, it is not worthwhile as the semantics of this relationship would be obscured by the WSDL definition. Therefore, we consider it to meet only the first of requirements #1-4.

WSDL can be used for simple co-reference discovery, as multiple services may be defined in the same document. Service type definitions might also permit coreference identification, however not in the case of common service types. Therefore we consider WSDL to pass only #6 of our non-functional requirements.

5.3.7 Summary

	1	2	3	4	5	6
WebID & FOAF	✔					
Nokia					✔	✔
IRW	✔				✔	✔
Web of Trust				✔		
POWDER						
WSDL	✔					✔

Table 5.1: Identity vocabulary requirements satisfied by related work

As we have illustrated in Table 5.1, none of the related works discussed above meet all of our requirements for an identity vocabulary. We note that while some of them fulfil our co-reference related requirements (#5 & 6), we believe it is mostly a side-effect of their design, rather than a design choice itself. Therefore, in the next section we present a vocabulary designed to meet those requirements.

5.4 Web Server Identity Vocabulary

Against this background, in this section we propose a new web server identity vocabulary. Our vocabulary is designed as foundation upon which we can build an identity and reputation infrastructure, and upon which a more complete picture of agent identity can grow organically.

Our vocabulary describes agent identity with an intensional definition style; it models web servers as agents described by their relationships to other things, such as the parts of their behaviour which is expressed over web-based channels. In more detail, instead of describing agents directly, our vocabulary describes their observable interfaces, i.e. the parts of them which perform HTTP exchanges.

Our Web Server identity vocabulary consists of 11 classes, most found in two main hierarchies, classifying agents and their behaviours, respectively. It is a minimal vocabulary, designed to act as an interface between existing vocabularies, in order to encourage an intensional approach to identity definition. We reproduce its full definition in Listing A.1, in Appendix A.

5.4.1 Agent Types

The most notable is the agent hierarchy, which consists of classes for 'Agent', 'WebAgent', 'WebClientAgent' and 'WebServerAgent'. As illustrated in Figure 5.3, these are each subclasses of the previous class, except for 'WebClientAgent' and 'WebServerAgent' which are non-disjoint subclasses of 'WebAgent'. This means, for example, that an agent could be both a 'WebClientAgent' and a 'WebServerAgent' at the same time. These classes categorise agents by the behaviours which they exhibit, and are declared as equivalent to terms in a number of other vocabularies, as listed in Section 5.5.4. Examples of how these classes can be used are given in Section 5.7.

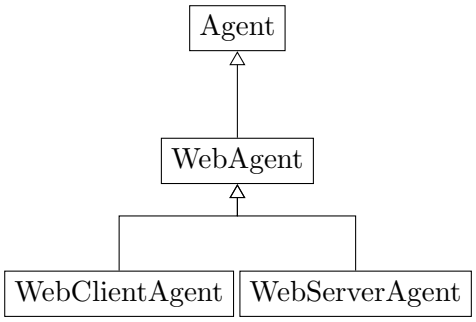


FIGURE 5.3: Subclass relationships within the Agent hierarchy

The vocabulary provides two data properties which link an Agent to a particular behaviour it exhibits, and vice-versa: 'hasBehaviour' and 'isBehaviourOf', illustrated in Figure 5.4.

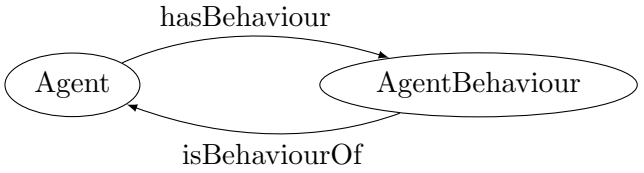


FIGURE 5.4: Agent behaviour properties

5.4.2 Agent Behaviours

Next is the 'AgentBehaviour' hierarchy, which consists of classes for 'AgentBehaviour', 'HTTPBehaviour', 'HTTPClientBehaviour', 'HTTPServerBehaviour' and 'HTTPSServer-Behaviour'. The subsumption hierarchy is defined as illustrated in Figure 5.5. These behaviours are modelled distinctly from the agent itself as an agent might have a number of each behaviours, each operating slightly differently. For example, a single agent might act as an HTTP client, gathering data, and also act as an HTTP server, re-publishing that data in a new form. Or an agent might operate across multiple domains, which we would model with two HTTPServerBehaviour instances.

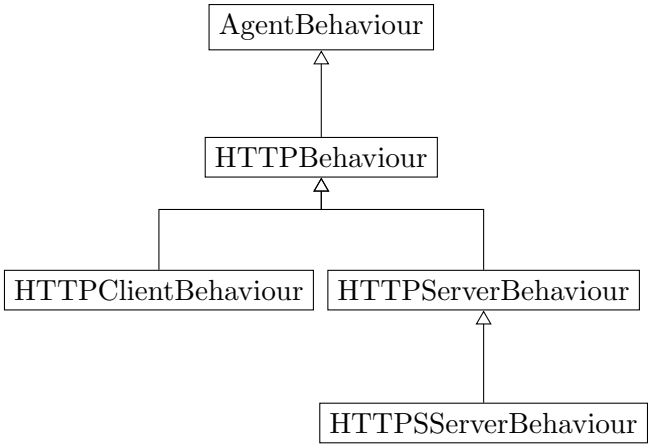


FIGURE 5.5: Subclass relationships within the AgentBehaviour hierarchy

5.4.3 Web Location

The WebLocation class can describe a single URI or a collection of URIs. Unlike the IRW ontology which must describe each URIs individually, we employ the POWDER-S vocabulary to describe groups of URIs which match certain patterns.

Listing 5.1 demonstrates the definition of five different WebLocation instances. First is the definition of `http://www.example.com/`, identified in the listing as `#www_ex_com`, which combines of three separate restrictions on the protocol, hostname and port of the URI. The POWDER-S specification encodes these restrictions as regular expression, which in an Notation 3 (N3) encoding results in many backslash escape characters. For

```
<#www_ex_com> a ident:WebLocation;
    owl:intersectionOf (

        [   # normal http protocol only
            a owl:Restriction;
            owl:onProperty wdrs:matchesregex;
            owl:hasValue "http\\:\\/\\/";
        ]
        [   # hostname of only www.example.com
            a owl:Restriction;
            owl:onProperty wdrs:matchesregex;
            owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)?([^\\:\\/\\?\\#\\@]+\\.)↩
?www.example.com(\\:([0-9]+))?\\/";
        ]
        [   # port of only 80 (the default)
            a owl:Restriction;
            owl:onProperty wdrs:matchesregex;
            owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)?([^\\:\\/\\?\\#\\@]+\\.)↩
*[^\\:\\/\\?\\#\\@]+\\:80\\/";
        ]
    ).


# Class of WebLocations using the http uri scheme.
<#schemehttp> a ident:WebLocation;
    rdfs:subClassOf [

        # normal http protocol only
        a owl:Restriction;
        owl:onProperty wdrs:matchesregex;
        owl:hasValue "http\\:\\/\\/";
    ].

# Class of WebLocations using port 80 (the default).
<#port80> a ident:WebLocation;
    rdfs:subClassOf [

        # port of only 80 (the default)
        a owl:Restriction;
        owl:onProperty wdrs:matchesregex;
        owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)?([^\\:\\/\\?\\#\\@]+\\.)↩
*[^\\:\\/\\?\\#\\@]+\\:80\\/";
    ].

<#www_ex_net> a ident:WebLocation;
    owl:intersectionOf (

        <#schemehttp>
        <#port80>
        [   # hostname of only www.example.net
            a owl:Restriction;
            owl:onProperty wdrs:matchesregex;
            owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)?([^\\:\\/\\?\\#\\@]+\\.)↩
?www.example.net(\\:([0-9]+))?\\/";
        ]
    ).

<#www_ex_org> a ident:WebLocation;
    owl:intersectionOf (

        <#schemehttp>
        <#port80>
        [   # hostname of only www.example.org
            a owl:Restriction;
            owl:onProperty wdrs:matchesregex;
            owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)?([^\\:\\/\\?\\#\\@]+\\.)↩
?www.example.org(\\:([0-9]+))?\\/";
        ]
    ).
```

LISTING 5.1: WebLocations matching the websites `http://www.example.com/`, `http://www.example.org/` and `http://www.example.net/`.

the 'WebLocation' definitions of `http://www.example.org/` and `http://www.example.net/`, we first declare the protocol and port restrictions as separate classes in order to demonstrate how they can be factored out. Then we declare 'WebLocations' for `http://www.example.net/` and `http://www.example.org/`, referencing those common restrictions using `#schemehttp` and `#port80`.

5.4.4 Delegation

In order to model delegation we require a three-way relationship between two 'HTTPServerNatures' and a 'WebLocation'. Since RDF relationships are only two-way, we have reified this relationship as the Delegation class in order to represent its three-sided nature. These relationship properties between 'HTTPBehaviours', 'Delegations' and 'WebLocations', are depicted in Figure 5.6, and we demonstrate the use of this relationship later, in Figure 5.7.

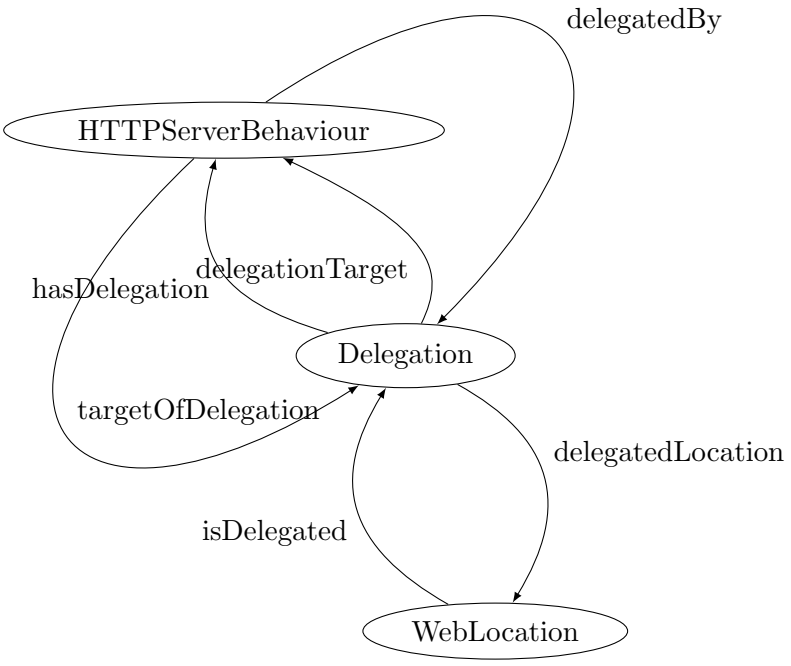


FIGURE 5.6: Agent delegation model

5.5 Coreference

In Section 4.4.3 we discussed the coreference problem faced by the Semantic Web. Loosely, it is the problem of determining whether two different identifiers refer to the same or different things. In order to avoid an adoption bootstrapping problem we have allowed agents to non-authoritatively describe other agents.

In an attempt to mitigate the coreference problem this may cause, we have take certain stances in the design of this vocabulary. Principally, we argue that agents should be defined by their identifying characteristics and relationships, rather than by their identifying URIs.

5.5.1 Distinguishing Information

In almost all cases, coreference analysis cannot be performed over URIs alone, as they are generally opaque identifiers. Therefore, in order to support effective coreference analysis, there need to be sufficient links and data properties surrounding an identifier for them to be compared meaningfully. Unfortunately we cannot define exactly the degree of data or links needed for good coreference analysis as it depends on the context the decision is made in, as well as the degree of certainty required by that context.

In contrast to people, whose identifying characteristics are hard to create unique indexes from, the identifying characteristics of Web-based agents are rooted in unique naming schemes. The HTTP protocol operates over the Domain Name System (DNS) [Mockapetris 1987] and IP [Postel 1981] schemes which each have defined uniqueness properties and central registries (Internet Corporation for Assigned Names and Numbers (ICANN) and Internet Assigned Numbers Authority (IANA)). Therefore, our vocabulary recommends that descriptions of agent identities are accompanied by records containing this data.

Provenance records are another form of information which may be used to identify agents. These can include transcripts of the HTTP exchanges which an agent has been part of, and potentially also records of the DNS query which informed the network connection over which the exchange took place. Although this data may may not be uniquely identifying forever, as IP addresses or domain names may change ownership, we can expect it to change relatively infrequently. When it does change, will be able to identify that from our provenance records, giving us the opportunity to verify that other important characteristics of the agent have remained the same, i.e. that it is still the same agent.

The Semantic Web HTTP vocabulary [Koch et al. 2011] provides us with the tools to associate an HTTP exchange with the network connection over which it took place. Unfortunately, the `connectionAuthority` property provides only limited metadata on the connection: a hostname, and a network port number. There are no properties to describe the connection protocol, the IP address of the actual host connected to, or the DNS request on which the connection was based. At the time of writing, there is no mature Semantic Web vocabulary in which provenance records of DNS exchanges can be recorded. Together, this means that while these provenance records may give us useful information, we have no vocabulary in which to record the key data which cements our

trust in these exchanges. For the moment, the development of this ontology is beyond the scope of our current work.

5.5.2 Canonical URI Discovery

In order to reference an identity description, supporting agents will be forced to coin a URI for the subject of each description. If each supporting agent were to coin a new identifier, this would create a serious problem of coreference.

```
Link: <http://www.example.org/id/HTTPserverNature> ; rel='http://purl.org/↩
mcobden/identity#isServedBy'
```

Listing 5.2: Example agent URI advertisement header

```
Link: <http://www.example.org/id/delegation01> ; rel='http://purl.org/mcobden/↩
identity#delegatedBy'
```

Listing 5.3: Example delegating agent URI advertisement header

To combat this, we propose the use of a custom HTTP Link header, shown in Listing 5.2, with which agents may advertise the canonical URI by which they wish to be identified. In addition, we also propose that agents in the delegation chain may insert a similar header (as in Listing 5.3) into the HTTP headers in order to advertise their presence. Note that this advertises the URI of a delegation record, rather than the URI of the delegating agent, this allows breaks in the delegation chain to be identified. Respectively, these two headers map directly to the `ident:isServedBy` and `ident:delegatedBy` vocabulary terms. They are intended to carry the same implications, where the WebLocation in question is that of the current HTTP Request.

5.5.3 URI Pattern Equivalence

Our vocabulary employs the POWDER-S encoding of the POWDER URI description format. The POWDER specification allows URIs to be filtered by the following components restrictions: schemes, hosts, ports, and paths (exact, contains, starts with, ends with). The XML encoding can encode these directly, however the POWDER-S encoding employs regular expression patterns to define these restrictions. The POWDER formal semantics [Konstantopoulos and Archer 2009] defines a set of regular expression templates which specify how URL component restrictions should be converted into regular expression restrictions.

In order for these URI patterns to be of maximum use to coreference analysis, it may be desirable to extract the original semantics of the restriction, for example, to identify whether different 'WebLocations' are equivalent, overlapping, or disjoint. Therefore, our

vocabulary explicitly states that wherever possible these templates should be employed without modification.

5.5.4 Term Alignment and Interoperability

In order to interoperate well with data expressed in other ontologies and tools which expect such data, our vocabulary declares a number of relationships to terms from other ontologies, which we show in Table 5.2.

Our Term	Relationship	Other Terms
Agent	owl:equivalentClass	foaf:Agent prov:Actor terms:Agent
WebServerAgent	owl:equivalentClass	webArch:Server irw:WebServer
WebClientAgent	owl:equivalentClass	irw:WebClient
WebLocation	owl:equivalentClass	irw:URI
usesPublicKey	rdfs:subPropertyOf	wot:hasKey

Table 5.2: Term equivalences with other vocabularies.

5.6 Worked Example

As an example, below we illustrate an identity description as earlier described by our use case in Section 2.1.1. Suppose Susan's agent responds to HTTP requests under `http://susan.example.com/` Tom's agent would record the identity description shown in Listing 5.4. This description includes Tom's agent's perceived model of Susan's agent, and a record of an HTTP exchange between them.

In the our illustrative scenario, after Susan upgrades her Agent, Hubert's agent observes a different set of agents. Listing 5.5 show the observations of Hubert's agent. Note that after Susan upgraded her Agent it reported a different value for the 'Server' HTTP header, and an additional 'Via' header; it is this kind of information which can indicate a change in the nature of a web agent and indicate relationships with other web agents.

5.7 Requirements Evaluation

In this section, we look back to the requirements we described in Section 5.2, and evaluate the vocabulary we presented in the previous sections against them.

1. **Able to describe a single agent hosted on a single dedicated web server**

```
<#location_susan_example_com> a ident:WebLocation;
    owl:intersectionOf (
        [   # normal http protocol only
            a owl:Restriction;
            owl:onProperty wdrs:matchesregex;
            owl:hasValue "http\\:\\/\\/";
        ]
        [   # hostname of only susan.example.com
            a owl:Restriction;
            owl:onProperty wdrs:matchesregex;
            owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)?([^\\:\\/\\?\\#\\@]+\\.)↩
?susan.example.com(\\:([0-9]+))?\\/";
        ]
        [   # port of only 80 (the default)
            a owl:Restriction;
            owl:onProperty wdrs:matchesregex;
            owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)?([^\\:\\/\\?\\#\\@]+\\.)↩
*[^\\:\\/\\?\\#\\@]+\\:80\\/";
        ]
    ).

<#susan_agent>  a ident:Agent;
    ident:hasBehaviour [
        a ident:HTTPServerBehaviour;
        ident:serves <#location_susan_example_com>;
    ].

<#conn> a httpv:Connection ;
    httpv:connectionAuthority "susan.example.com:80" ;
    httpv:requests ( <#req0> ) .

<#req0> a httpv:Request ;
    httpv:absolutePath "/" ;
    httpv:headers (
        [ a httpv:RequestHeader ;
            httpv:fieldName "Host" ;
            httpv:fieldValue "susan.example.com" ;
            httpv:hdrName <http://www.w3.org/2011/http-headers#host> ]
        [ a httpv:RequestHeader ;
            httpv:fieldName "User-Agent" ;
            httpv:fieldValue "Tom's Agent" ;
            httpv:hdrName <http://www.w3.org/2011/http-headers#user-agent> ]
        ... (full headers omitted) ...
    );
    httpv:httpVersion "1.1" ;
    httpv:methodName "GET" ;
    httpv:mthd <http://www.w3.org/2011/http-methods#GET> ;
    httpv:resp <#resp0> .

<#resp0> a httpv:Response ;
    foaf:maker <#susan_agent>;
    httpv:body <#cont0-bin> ;
    httpv:headers (
        [ a httpv:EntityHeader ;
            httpv:fieldName "Server" ;
            httpv:fieldValue "Susan's agent 1.0" ;
            httpv:hdrName <http://www.w3.org/2011/http-headers#server> ]
        ... (full headers omitted) ...
    );

    httpv:httpVersion "1.1" ;
    httpv:reasonPhrase "OK" ;
    httpv:sc <http://www.w3.org/2011/http-statusCodes#OK> ;
    httpv:statusCodeValue "200" .
```

Listing 5.4: Tom's agent's description of Susan's agent

```
<#susan_agent>  a ident:Agent;
    ident:hasBehaviour [
        a ident:HTTPServerBehaviour;
        ident:serves <#location_susan_example_com>.
    ].

<#server_agent1>    a ident:Agent;
    ident:hasBehaviour [
        a ident:HTTPServerBehaviour;
        ident:delegates <#location_susan_example_com>.
        ident:hasDelegation [
            a ident:Delegation;
            ident:delegatedLocation <#location_susan_example_com>;
            ident:delegationTarget <#susan_agent>.
        ]
    ].

<#server_agent2>    a ident:Agent;
    ident:hasBehaviour [
        a ident:HTTPServerBehaviour;
        ident:delegates <#location_susan_example_com>.
        ident:hasDelegation [
            a ident:Delegation;
            ident:delegatedLocation <#location_susan_example_com>;
            ident:delegationTarget <#susan_agent>.
        ]
    ].

<#conn> a httpv:Connection ;
    httpv:connectionAuthority "susan.example.com:80" ;
    httpv:requests ( <#req0> ) .

<#req0> a httpv:Request ;
    httpv:absolutePath "/" ;
    httpv:headers (
        [ a httpv:RequestHeader ;
            httpv:fieldName "Host" ;
            httpv:fieldValue "susan.example.com" ;
            httpv:hdrName <http://www.w3.org/2011/http-headers#host> ]
        [ a httpv:RequestHeader ;
            httpv:fieldName "User-Agent" ;
            httpv:fieldValue "Hubert's Agent" ;
            httpv:hdrName <http://www.w3.org/2011/http-headers#user-agent> ]
        ... (full headers omitted) ...
    );
    httpv:httpVersion "1.1" ;
    httpv:methodName "GET" ;
    httpv:mthd <http://www.w3.org/2011/http-methods#GET> ;
    httpv:resp <#resp0> .

<#resp0> a httpv:Response ;
    foaf:maker <#susan_agent>;
    httpv:body <#cont0-bin> ;
    httpv:headers (
        [ a httpv:EntityHeader ;
            httpv:fieldName "Via" ;
            httpv:fieldValue "1.1 svr1.susan.example.com (Some Enterprise Proxy)↩
" ;
            httpv:hdrName <http://www.w3.org/2011/http-headers#via> ]
        [ a httpv:EntityHeader ;
            httpv:fieldName "Server" ;
            httpv:fieldValue "Enterprise Agent 1.3" ;
            httpv:hdrName <http://www.w3.org/2011/http-headers#server> ]
        ... (full headers omitted) ...
    );
    httpv:httpVersion "1.1" ;
    httpv:reasonPhrase "OK" ;
    httpv:sc <http://www.w3.org/2011/http-statusCodes#OK> ;
    httpv:statusCodeValue "200" .
```

LISTING 5.5: Hubert's agent's description of Susan's agent

Our vocabulary must be able to describe the simplest class of webservers, those serving only single web agent from a single web server. Listing 5.6 is an example of such a description, describing an agent which serves requests for `http://www.example.com/`.

```
[]  a ident:Agent;
    ident:hasBehaviour [

        a ident:HTTPServerBehaviour;
        ident:serves [

            a ident:WebLocation;
            owl:intersectionOf (

                [   # normal http protocol only
                    a owl:Restriction;
                    owl:onProperty wdrs:matchesregex;
                    owl:hasValue "http\\:\\/\\/";
                ]
                [   # hostname of only www.example.com
                    a owl:Restriction;
                    owl:onProperty wdrs:matchesregex;
                    owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)↩
?([^\\:\\/\\?\\#\\@]+\\.)?www.example.com(\\:([0-9]+))?\\/";
                ]
                [   # port of only 80 (the default)
                    a owl:Restriction;
                    owl:onProperty wdrs:matchesregex;
                    owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)↩
?([^\\:\\/\\?\\#\\@]+\\.)*[^\\:\\/\\?\\#\\@]+\\:80\\/";
                ]
            ).
            ].
        ].
    ].
```

Listing 5.6: An RDF description, in N3 format, of an agent serving a single website.

Providing the POWDER WebLocation descriptions are correct, this captures the essential characteristics of a single website hosted on a single server.

2. **Able to describe an agent hosted on a shared web server**

Modelling a shared-hosting environment proves to be more complex than our first requirement. We model this using the delegation terms in our identity vocabulary; a single agent handles all incoming requests for co-hosted websites, which delegates each request to an agent responsible for the website in question. Listing 5.7 demonstrates this form of description, though for the sake of brevity, this example references the WebLocations defined in Listing 5.1, and includes only a single delegation record, where there might normally be more.

Our vocabulary is sufficiently flexible to describe shared-hosting environments, and thus it meets our requirements in this area.

3. **Able to describe a web agent spanning multiple web servers**

```
<#agent1>   a ident:Agent;
    ident:hasBehaviour [

        a ident:HTTPServerBehaviour;
        ident:delegates
            <#www_ex_com>,
            <#www_ex_org>,
            <#www_ex_net>.
        ident:hasDelegation [

            a ident:Delegation;
            ident:delegatedLocation <#www_ex_com>;
            ident:delegationTarget <#agent2.
        ]
    ].

<#agent2>   a ident:Agent;
    ident:hasBehaviour [

        a ident:HTTPServerBehaviour;
        ident:serves
            <#www_ex_com>.
    ].
```

LISTING 5.7: An RDF description, in N3 format, of an agent hosted on a shared server.

Next, our vocabulary must be capable of describing agents in high-performance environments where an array of servers may be employed to handle requests. Our model for this resembles the inverse of the previous requirement's example; rather than a single agent delegating to one agent per website, we model it as many separate agents delegating to a single common agent. In our example, the different webservers fielding the request each run an instance of the same software to which they delegate the request. This shared software embodies the common agent, despite the fact it is running on different machines. We demonstrate a description of this example in Listing 5.8, also, for brevity, referencing the WebLocations defined in Listing 5.1.

Again, our vocabulary is able to describe complex hosting environments, and thus meets our requirements in this area.

4. **Able to describe a web agent which employs public key encryption**

The vocabulary must be capable of describing an agent employing a particular public key, perhaps to provide connection security assurances to clients. Our representation of this is simply an `ident:usesPublicKey` relationship between an agent and a public key resource. An agent wishing to lend weight to such an assertion could employ a signed RDF graph (see Section 4.2.1.3), or implement the WebID authentication protocol. We demonstrate how to describe this relationship in Listing 5.9, again, for brevity, referencing the WebLocations defined in Listing 5.1.

5. **Must provide sufficient data for coreference**

```
<#site_agent>   a ident:Agent;
    ident:hasBehaviour [

        a ident:HTTPServerBehaviour;
        ident:serves
            <#www_ex_com>.
    ].

<#server_agent1>    a ident:Agent;
    ident:hasBehaviour [

        a ident:HTTPServerBehaviour;
        ident:delegates
            <#www_ex_com>.
        ident:hasDelegation [
            a ident:Delegation;
            ident:delegatedLocation <#www_ex_com>;
            ident:delegationTarget <#site_agent>.
        ]
    ].

<#server_agent2>    a ident:Agent;
    ident:hasBehaviour [

        a ident:HTTPServerBehaviour;
        ident:delegates
            <#www_ex_com>.
        ident:hasDelegation [

            a ident:Delegation;
            ident:delegatedLocation <#www_ex_com>;
            ident:delegationTarget <#site_agent>.
        ]
    ].

<#server_agent3>    a ident:Agent;
    ident:hasBehaviour [

        a ident:HTTPServerBehaviour;
        ident:delegates
            <#www_ex_com>.
        ident:hasDelegation [

            a ident:Delegation;
            ident:delegatedLocation <#www_ex_com>;
            ident:delegationTarget <#site_agent>.
        ]
    ].
```

Listing 5.8: An RDF description, in N3 format, description of an agent hosted across multiple servers.

```
<#agent>    a ident:Agent;
    ident:hasBehaviour [
        a ident:HTTPServerBehaviour;
        ident:serves <#www_ex_com>;
        ident:usesPublicKey <#key>.
    ].

<#key>  a wot:PubKey;
    wot:hex_id  "01234567";
    wot:length  1024;
    wot:fingerprint "0123456789ABCDEF0123456789ABCDEF01234567";
    wot:pubkeyAddress   <http://www.example.com/pubkey.asc>.
```

Listing 5.9: An RDF description, in N3 format, of an agent employing public-key encryption.

As we argue in Sections 5.5 and 5.1, it is important to have sufficient distinguishing data in an identity description to enable effective coreference resolution.

In an Internet setting, the primary identifying information of corresponding agents are network addresses and cryptographic keys. Our vocabulary describes Web agents primarily in terms of the HTTP locations they accept requests for, the cryptographic keys they hold, and any delegation relationships they may be involved in. Delegation may only be observable in some circumstances, but it is important relationship in the context of trustworthiness, as it potentially involves other agents in a trusting decision.

A limitation of our ontology is that there remains a 'gap' between network address information and HTTP locations served. The process of a URI resolution usually includes at least one DNS query and a network connection to the resulting address. The result of both of these depends on the state of the user's Internet connection when the attempt is made, and they may be – knowingly or otherwise – behind a network firewall or connection proxy. Fortunately, in most cases, one can assume that these resolution steps proceeded normally, and then treat the host-part of HTTP URIs in a manner akin to network address.

If, however, in order to perform effective coreference resolution, one must be certain about the details of the lower-level URI resolution steps, then we suggest the use of provenance information. Specifically, we suggest that identity description of the subject should be linked to, and distributed with, provenance records of the resolution stages on which interactions with the subject were performed. They would provide provide the necessary information to fill this 'gap', without abandoning the temporal nature of URI resolution.

The additional metadata captured in interaction provenance records might also provide useful information for coreference analysis; for example: changes in HTTP 'User-Agent' may indicate changes in identity. Unfortunately, as we highlighted in Section 5.5.1, there are key areas for which we do not have a mature vocabulary in which to record this provenance information – DNS exchanges, and connection initialisation and encryption negotiation. For the moment, we deem it sufficient to trust that domain name resolution and connection initialisation are not manipulated, and do not play a large role in agent identity, and thus we do not seek to create vocabularies for these actions in this thesis.

6. **Must avoid exacerbating the problem of coreference**

The minting of new Semantic Web URIs for existing concepts is inevitable; one will not always be able to find an existing URI, coined by another entity, which one can be sure represents the same understanding of a concept. Nevertheless, our vocabulary makes some recommendations to in an attempt to reduce rate at which new URIs are coined. First, our recommendations to facilitate effective coreference, in Section 5.5, assist the

identification and conflation of duplicated identifiers. In addition, we suggest a URI advertisement protocol, designed to facilitate discovery of existing identifiers. These measures, described in Section 5.5.2, employ custom HTTP headers to advertise the URIs of identity-aware Semantic Web agents. The design of this vocabulary, and our recommendations for its use, is sufficient to fulfil this requirement.

#	Requirement	Verdict
1	Able to describe a single agent hosted on a single dedicated web server	Met
2	Able to describe an agent hosted on a shared web server	Met
3	Able to describe a web agent spanning multiple web servers	Met
4	Able to describe a web agent which employs public key encryption	Met
5	Must provide sufficient data for coreference	Met with limitations
6	Must avoid exacerbating the problem of coreference	Met

Table 5.3: Summary of requirements analysis

In summary, we have almost completely met our original requirements for this vocabulary, as shown in Table 5.3.

5.8 Summary

Our objective in this thesis is to advance us towards the vision of the Semantic Web as an ecosystem of intelligent, and trust-aware, autonomous actors.

The ability to uniquely identify agents is a key requirement for many Semantic Web challenges involving trust, provenance or inter-agent interactions. Work to date has often sidestepped this representation problem — modelling agents only as members of the `foaf:Agent` class [Hartig and Zhao 2010, Raimond et al. 2007, Moreau et al. 2010]. Therefore, in this chapter we take a pragmatic approach to the socio-technical problem of describing identity, presenting an identity vocabulary for describing web-based agents by their observable characteristics. This is intended to act as a minimal foundation of identity upon which a more complete picture of identity can grow organically.

In order to avoid an adoption bootstrapping problem, our vocabulary encourages the non-authoritative description of web-based agents, whilst encouraging co-reference identification through analysis and canonical URI discovery.

We began by outlining the requirements a web agent identity vocabulary must meet (Section 5.2). Against this background, we reviewed relevant existing work (Section 5.3).

Next, we presented our identity vocabulary and demonstrated its usage in Section 5.4, and discussed our coreference considerations in Section 5.5. Finally, in Section 5.7, we reviewed the vocabulary against our original requirements.

In the next chapter, to complement our work towards a better notion of identity on the Semantic Web, we propose a novel approach for bootstrapping trust in new identities by harnessing open information sources.

We follow this, in Chapter 7, by investigating the information-management challenges that trust-aware agents will face if they are to manage information as we described in Section 2.3. Specifically, we identify the requirements it places on our knowledge-bases and propose effective storage strategies for recording inferred information and its provenance.

Chapter 6

Grounding Trust on the Web and Semantic Web

In the previous chapter, we proposed a Web server identity description vocabulary to underpin a Semantic Web identity infrastructure, and encourage the growth of trust and reputation on the Semantic Web.

Unfortunately, as research agendas have recognised [O'Hara and Hall 2008], systems of this nature are known to suffer from a bootstrapping problem, and we have no reason to conclude that the Semantic Web will be an exception. A bootstrapping problem exists where a certain adoption threshold must be reached in order for adoption itself to have a positive return on investment. Early adoption is discouraged by the investment required, and so the threshold is unlikely to be reached.

In this chapter, we propose a novel solution to this bootstrapping problem which can help to build trust in new environments. Our approach is founded on the observation that neither the Web, nor the Semantic Web exist in a vacuum – they exist as extensions of human societies. We can address the bootstrapping problem by finding ways to encourage trust transfer from other environments. Crucially, this chapter harnesses free information sources in order to grow and strengthen the links between an online identity and the identity of its real-world counterpart, thereby encouraging trust transfer. This enables people to apply their existing knowledge of trustworthiness in a new environment, something we described in Use Case 2 (Section 2.2.1).

This chapter continues with a discussion of trust transfer in 6.1, and a discussion of how this applies to the Web domain in Section 6.2.

We then, in Section 6.3, present a website identity service which integrates a range of publicly available information to provide a Website identity dashboard. This dashboard is intended to enable users to quickly cross reference Website identity information from

different sources, and aims to thereby enable trust transfer, and help prevent misplaced trust through mistaken identity.

Following this, Section 6.4 presents an discussion of our identity service, including its successes, its shortcomings and avenues for future work. We also discuss how this work relates to the Semantic Web research agenda.

Finally, Section 6.5 concludes this chapter with a summary of what we have discussed.

6.1 Bootstrapping through trust transfer

Trust transfer describes the act of a person (the trustor) basing their initial trust in a target entity on trust in some other related entity, or on a context other than the one in which the target is encountered [Stewart 2003]. Stewart's study investigated the trust transfer between online e-commerce organisations, and found empirical evidence to support the hypothesis that greater perceived similarity and interaction between entities enables greater trust transfer.

As we discussed in Chapter 3, human society has many ways with which to advertise trustworthiness, and many factors upon which to judge it. Compared to the Web, human society has had a long time to develop effective trust mechanisms and establish long-standing reputations. Trust transfer provides a means by which we can harness these trust mechanisms and reputations in new environments.

Thus, if we can encourage trust transfer between old and new environments, we can combat the bootstrapping problem faced by new trust networks. While this approach does require a pre-existing trust environment from which to transfer, only artificial, simulated, environments are truly this isolated. The Web and the Semantic Web are certainly not isolated systems; their content and use are both expressions of human society, and thus we expect there to be many opportunities by which we can achieve trust transfer.

The concept of 'entitativity' describes a continuum in the extent to which collections of indiviuals are perceived as forming a cohesive unit [Hamilton et al. 1997]. Stewart's study demonstrated that perceptions of similarity (i.e. entitativity) between individuals correlates with greater trust transfer in the perceiver. Our approach to the trust bootstrapping problem is thus to encourage trust transfer by supporting the perception entitativity.

More practically, we will assist individuals in identifying groups of high entitativity by providing them with information from reputable channels which may highlight ties between individuals. Specifically, in the Web and Semantic Web domains, we are referring to entitativity between a web-based agent, and its real-world presence.

6.2 Entitativity and Trust on the Web

As we discussed in Section 3.4.3, the world wide web is by no means immune to the machinations of unsavoury individuals; individuals dishonestly exploit others online just as they do in the offline world. Impersonation attacks attempt to induce incorrect perceptions of high entitativity between the attacker and a trusted party, and then take advantage of the trusted status which this affords them.

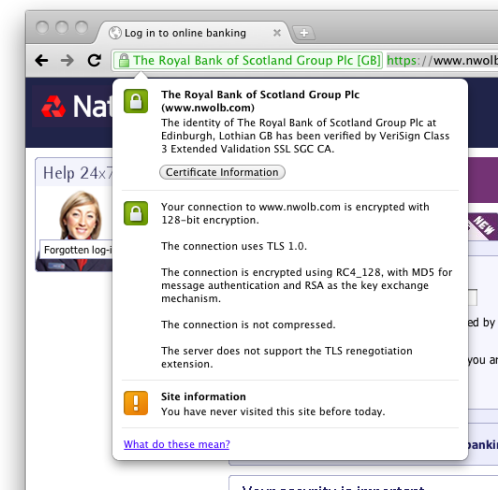
In Section 3.4.3 we also discussed the factors which influence users' trusting decisions on the Web. Dhamija et al. [2006] demonstrated that people rely heavily on the visual features of Web page content, rather than the technical facts presented by their Web browser. They found that 23% of their participants judged website legitimacy only on the page content, and only 13% participants in their study had ever checked the content of an SSL certificate. This suggests that one of the reasons impersonation attacks are so successful on the Web, is that a significant portion of Web users are not aware of, or do not understand, the technical information which identifies a website. The adoption of Web technologies has likely outpaced the widespread education of its underlying principles, such as urls, hostnames and public-key infrastructure. More needs to be done to teach users these skills, and to equip them with information which can help them to identify legitimate identities.

Since Dhamija et al.'s study, Web Browser vendors have begun to address these problems; newer versions of their software have included better interfaces to website identity information. In addition, the use of Extended Validation (EV) certificates (see Section 3.4.2.3), which provide better identity information and assurance, has become more common. For example: Google's Chrome browser and Microsoft's Internet Explorer 8 highlight the domain name portion of the URL in their address bars, and Mozilla Firefox and Google Chrome include a menu which displays a readable representation of the encryption certificate presented by the Website. Figure 6.1 illustrates different views on the same information from three different web browsers.

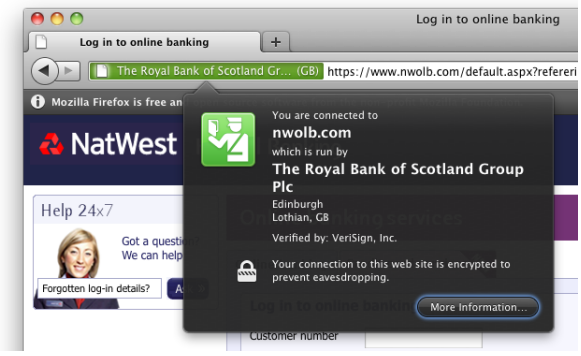
These interface changes help to highlight the presence of identity information, which may in turn help to prevent false perceptions of entitativity. While this is undoubtedly an improvement, we believe that there is much more to be gained from providing users with additional identity-reated information.

6.3 Website Identity Service

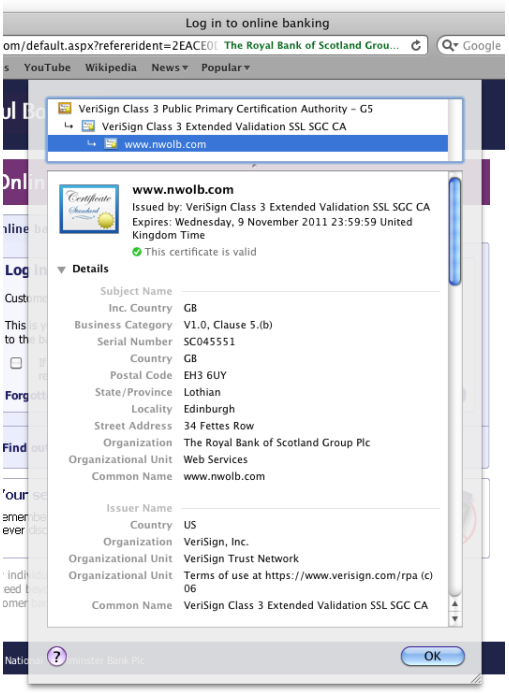
To demonstrate how we can bootstrap trust in new environments through trust transfer, we present a novel website identity service which aims to foster trust transfer to websites from their offline identities. This service combines information from a range of sources



(a) Google Chrome



(b) Mozilla Firefox



(c) Safari

FIGURE 6.1: Encryption certificate information shown in three popular Web browsers.

to describe a websites' identity and to highlight the ties a web presence has with offline entities. These ties, if sufficient, may increase the level of entitativity perceived by users, and thereby support the transfer of a user's trust between the entities. Transfer of trust from outside the Web onto the actors operating on this Web, which we earlier described in Section 2.2.1, will directly address the bootstrapping problem by increasing the number of trustworthy actors in the population.

In addition to promoting trust transfer, this service helps to bootstrap a trust ecosystem by other means. By improving access to identity-centric information, it will help to prevent cases of misplaced trust through mistaken identity. This will improve users' ability to judge the identity of websites, so that they may reach better trusting decisions, with more confidence. In addition, the ability to quickly cross-reference identity information from different sources will help users to identify inconsistencies, which may in turn highlight deceptive behaviour. Decreasing the incidence of misplaced identity would, over time, help to improve the level of trust in the Web as a platform.

Our website identity service combines (i) host IP records and canonical DNS names, (ii) connection security information, (iii) domain name registration records, and (iv) national company registry information. Figure 6.2 demonstrates the information gathered for the internet host `www.nwolb.com`; the online banking portal of National Westminster Bank.

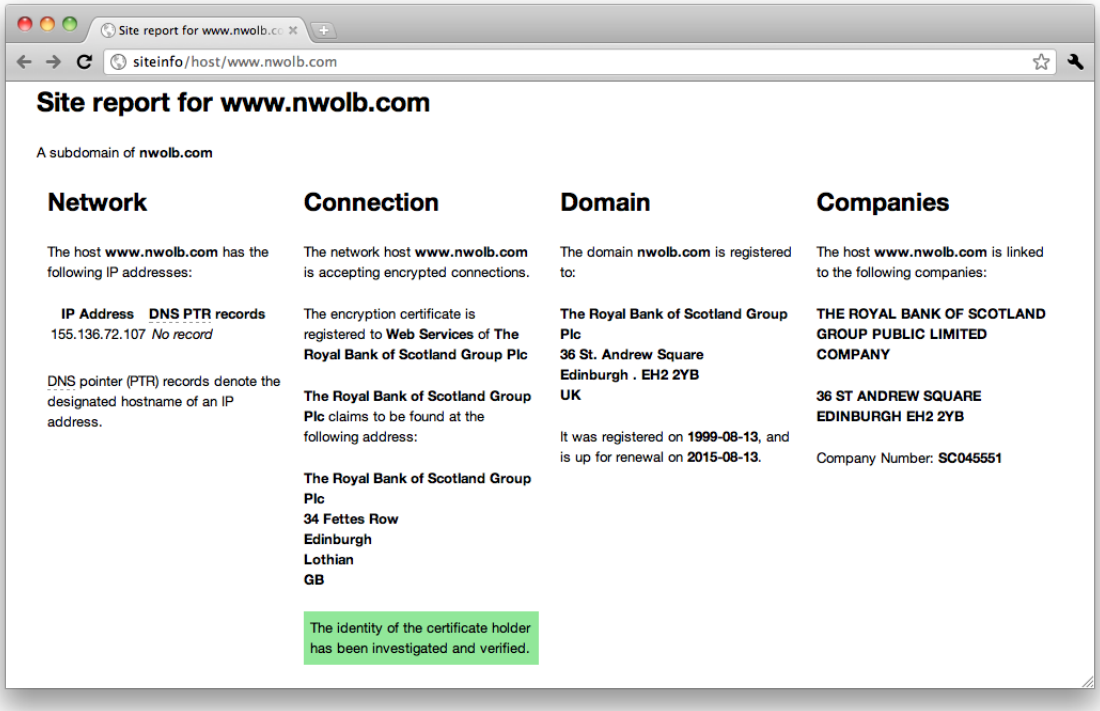
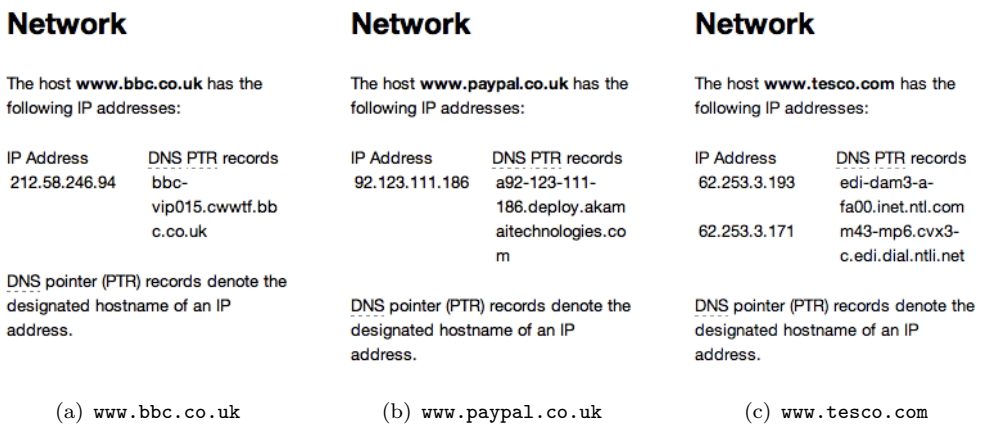


FIGURE 6.2: Website identity information for the internet host `www.nwolb.com`

6.3.1 Network Information

The first panel contains information on the internet address of the website in question, listing all IP addresses referenced by the website's DNS address record. We also include the result of a canonical hostname (DNS PTR) record query for each address, as it may provide some information about who operates the website host, or owns the address at which it resides. Figure 6.3 illustrates the content of this panel for three different hosts.



(a) `www.bbc.co.uk` (b) `www.paypal.co.uk` (c) `www.tesco.com`

FIGURE 6.3: Example network information for three different hosts

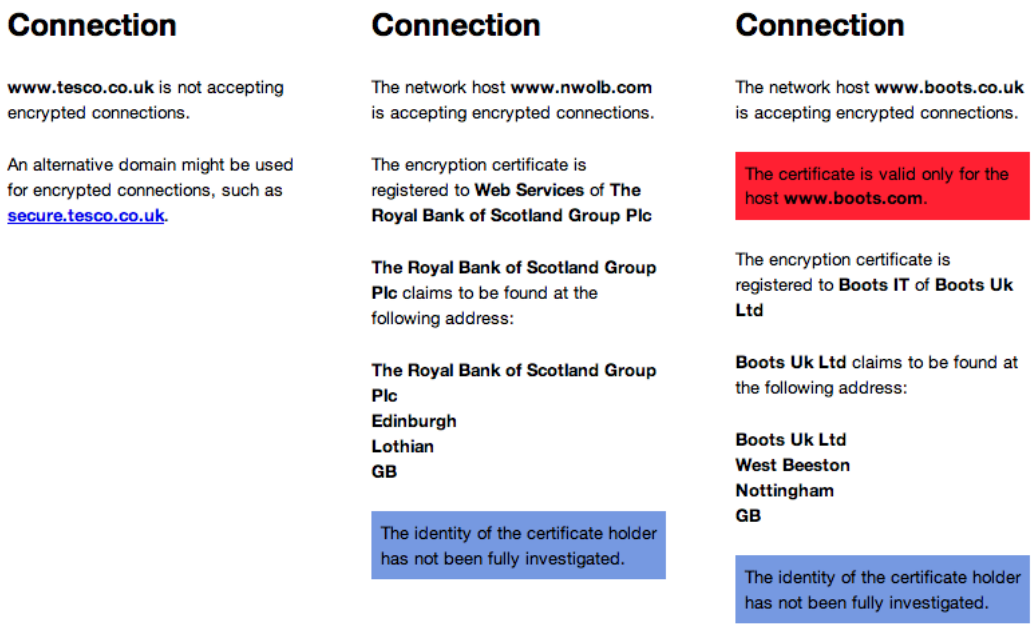
This panel primarily allows users to identify whether the website in question owns the IP addresses which answer requests for this hostname. It can also be used to identify shared-hosting environments, or the use of load-balancing or caching services, as in Figure 6.3(b), where `www.paypal.co.uk` is served by a host under the `akamaitechnologies.com` domain.

6.3.2 Connection Information

In the 'Connection' pane we include information about the possible security of the network connection to the website. Specifically, we identify whether the remote host will accept HTTPS connections, which are negotiated over an encrypted protocol. This, alone, can give users some assurance of privacy and security – assuming of course that the encryption certificates are legitimate and have not been compromised.

Similarly to popular Web browsers (as discussed in Section 6.2), this panel displays identity information extracted from the connection encryption certificate. Figure 6.4 illustrates the information displayed in this panel for three different hosts.

Aside from the benefits of encrypted communication, this panel also provides further information which can be compared with the other panels. Name and address information, if present, can be cross-referenced with the other information available. Certain certificates also include registered company numbers in their metadata, which enables further



(a) `www.tesco.co.uk` (b) `www.nwolb.com` (c) `www.boots.co.uk`

FIGURE 6.4: Example connection information for four different hosts

investigation and cross-referenceing. In addition, holders of EV certificates (such as the connection certificates pictured in Figure 6.1, 6.5 and the connection pane of Figure 6.2) will have been subject to an identity investigation by the certificate issuer, which further endorses the validity of the information present in the certificate.

When an website offers encrypted connections, we embed machine-readable RDF data into this area of the page using RDFa (see Section 4.2.1.1). This information, illustrated in Listing 6.1, employs our identity ontology from Chapter 5 to assert that the agent answering requests at this domain possesses a certain cryptographic key pair.

6.3.3 Domain Registration Information

The domain panel displays information from the Internet WHOIS record [Daigle 2004] of the host's domain name. The Internet WHOIS directory contains contact information for the owners and technical contacts for every registered domain name, as well as the creation, start and expiry dates of the registration. Figure 6.6 illustrates the information displayed in this part of the interface. Domain registration information is another data point which can be correlated and compared against other information.

While the correctness of WHOIS information is often not rigorously verified, it raises the bar for impersonation attempts. It is common for phishing attacks to be hosted from computer systems whose security has been compromised; attacks are then staged from within the websites that these systems were originally hosting. Between October

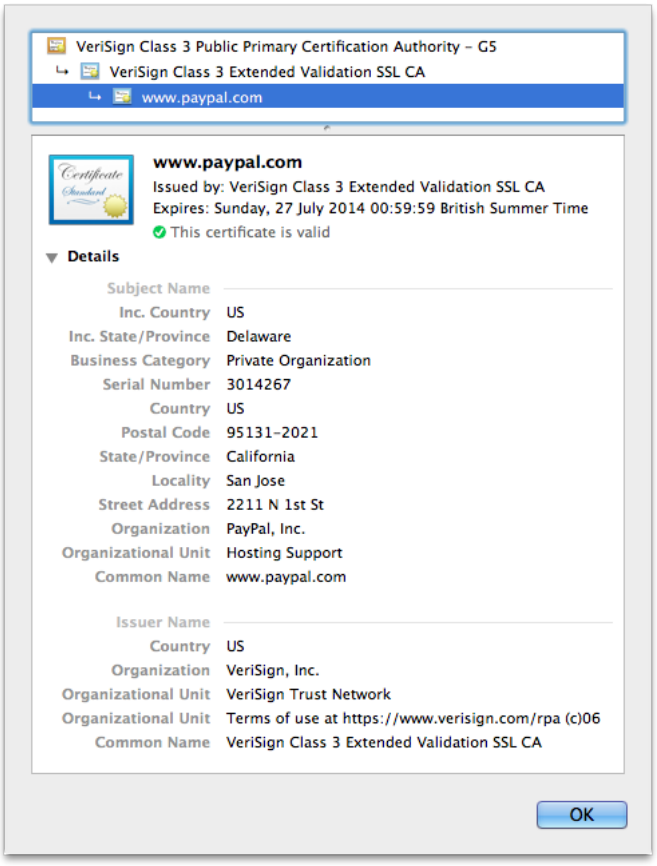
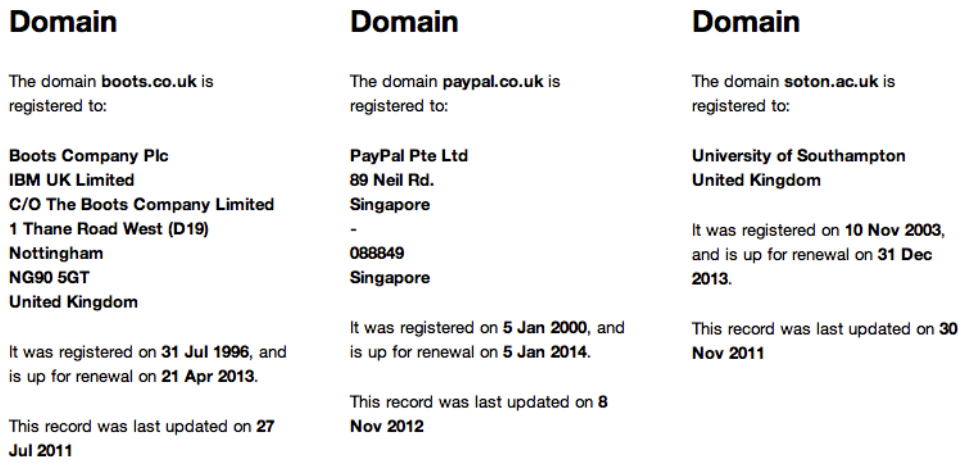


FIGURE 6.5: Encryption certificate for `www.paypal.com`

```
<#nwolb_agent>  a ident:Agent;
    ident:hasBehaviour [
        a ident:HTTPServerBehaviour;
        ident:usesPublicKey <#nwolb_key>;
        ident:serves [
            a ident:WebLocation;
            owl:intersectionOf (
                [   # https protocol only
                    a owl:Restriction;
                    owl:onProperty wdrs:matchesregex;
                    owl:hasValue "https\\:\\/\\/";
                ]
                [   # hostname of only www.nwolb.com
                    a owl:Restriction;
                    owl:onProperty wdrs:matchesregex;
                    owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)↩
?([^\\:\\/\\?\\#\\@]+\\.)?www.nwolb.com(\\:([0-9]+))?\\/";
                ]
                [   # port of only 443
                    a owl:Restriction;
                    owl:onProperty wdrs:matchesregex;
                    owl:hasValue "\\:\\/\\/(([^\\/\\?\\#]*)\\@)↩
?([^\\:\\/\\?\\#\\@]+\\.)*[^\\:\\/\\?\\#\\@]+\\:443\\/";
                ].
    ].

<#nwolb_key>    a wot:PubKey;
    wot:hex_id  "4E43C81D76EF37537A4FF2586F94F338E2D5BDDF";
    wot:length  2048;
    wot:fingerprint "DF85EAE713C5FFC71CFE803421F46644C26A9E2C";
```

LISTING 6.1: Website identity information embedded as RDFa



(a) `www.boots.co.uk` (b) `www.paypal.co.uk` (c) `www.ecs.soton.ac.uk`

FIGURE 6.6: Example domain information for three different hosts

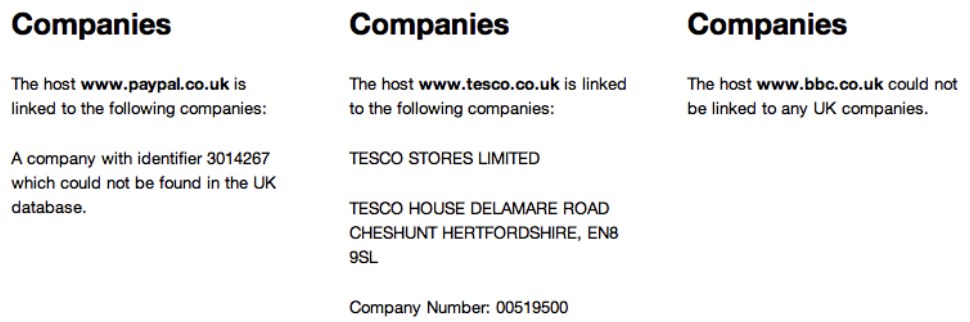
2007 and March 2008 Moore and Clayton [2009] found that 75% of phishing websites encountered were hosted on compromised systems. Compromising a web serving system is unlikely to grant the attacker permission to also edit the WHOIS records, as they are generally only configurable via the domain agent's website, through a separate credential system. This means that although compromised websites may be used as an impersonation platform, the domain's WHOIS records are unlikely to match the impersonation.

6.3.4 Company Information

The final pane in the interface displays, where possible, information on companies registered in the United Kingdom. In some cases we are able to link a hostname to a specific company; through information in either the domain name registration records, or in the encryption certificate. The first relies on the fact that UK domain name registrations (ending `.co.uk`), often include the company number in the registrant name field. The second approach harnesses a field in the encryption certificate metadata which is used to record company registration numbers.

When this information is found, the website identity service queries the UK Companies House database to retrieve the company's registered name and address to display in the *companies* pane. This information provides a further opportunity to cross-reference address data. Currently, our identity service only has means to query for records of companies registered in the UK, so while other panels may show information about non-UK hostnames, it cannot provide company information for non-UK companies.

Figure 6.7 illustrates the company information which is displayed in this pane. Note that in Figure 6.7(a), we were unable to identify the company in question from the



(a) `www.paypal.co.uk` (b) `www.tesco.co.uk` (c) `www.bbc.co.uk`

FIGURE 6.7: Example company information for three different hosts

identifier which was found. In this case, the identifier in question was gleaned from an SSL certificate for `www.paypal.com` (see Figure 6.5) issued to a United States company, so is likely to have been issued by a United States authority. Manual inspection of the certificate indicates that this company was allegedly incorporated in the US state of Delaware. A manual check through the Delaware state Division of Corporations website verifies that this identifier does correspond to the registration of Paypal, Inc.

In the case of Figure 6.7(c), neither the SSL certificate, nor the domain registration records contained any company registration numbers.

6.4 Discussion

Our service demonstrates a novel approach to bootstrapping new trust environments, harnessing existing information sources to support identities by assisting the perception of entitativity to encourage trust transfer. The Website Identity service significantly reduces the time and effort in cross-referencing website identity information. The ability to quickly perform a cross-reference check may be sufficient to prevent a large proportion of online impersonation attacks. If it remains the case that 75% of phishing attacks are hosted on compromised servers (see Section 6.3.3), a cross-reference check may be sufficient to identify the impersonation attempt. We do, however, still rely on the user having the inclination and the skills to corroborate the information presented.

We believe that the website identity service has demonstrated the potential in the ability to view and corroborate information across these sources; however, we acknowledge that there remains room for improvement. We have identified a number of shortcomings in our work; the most significant arise from issues beyond our control, while the others arise from our choice of implementation platform and data sources.

In order for our service to be of use, users must have the inclination to check the identity of the website they are visiting and be aware of our service and how to use it. Next,

users must be able to understand the information which our service presents. While some of the information displayed is highly technical, the bulk it consists of names and addresses, which should be understood by most people. We have made every effort to present the information in an accessible manner, however we cannot be assured our efforts were successful without formal evaluation.

Our service is somewhat disadvantaged due to the platform on which it is currently implemented: as a web-based service. As it is not integrated directly into users' browsers, it cannot ensure that it has access to the exact information the user has received, such as the network connection metadata. It is, however, significantly more difficult to create a web browser extension for a range web browsers, than it is to write a web-based service. For this reason we maintain that it was the right approach for demonstrating its potential, but we discuss integration as an avenue for future work in Section 6.4.1.

Additionally, as we mentioned in Section 6.3.4 our coverage of company information is limited to companies registered in the United Kingdom. While this will reduce the impact and usefulness of our service to users from outside the United Kingdom, we do not believe it undermines the success of this work. Our aim was to demonstrate the potential of this approach, rather than to develop a globally supported service. Adding support for company information in new jurisdictions, where that information is publicly available, is another area for further development. Where this information is not available, this kind of service may help to motivate its publication.

6.4.1 Future Work

We have identified a number of avenues for future work, some of which aim to address these shortcomings, while others aim to improve its overall value.

Formal evaluation

Before we take our identity service any further, we wish to undertake some formal evaluation of its effectiveness. We are conscious that trust transfer is highly subjective, therefore a formal study to determine the degree of transfer afforded is necessary.

We plan to investigate the degree to wish our service affects the perceived trustworthiness of a range of entity and website pairs, while also measuring for changes in perceived entitativity. In addition, varying the combinations of information panel presented would allow us to measure their effectiveness, as well as which work well together.

Further work could also measure the service's effectiveness in thwarting phishing attacks, by encouraging a decrease in percieved entitativity, though this is perhaps beyond the scope of our core research aims.

Semantic Web adaptation

Another principal area for future work is adapting the architecture of our system for the Semantic Web trust ecosystem. We would re-architect our system as a set of independent Linked Data publishing websites, each focusing on a particular information domain. We would then supplement this with tools and/or aggregation platforms to help bring this information together. This would isolate each component and allow other services to make use of the individual information sources for other purposes.

Another opportunity for future work is the upcoming 'Registered Organization Vocabulary' [Archer et al. 2013], currently under discussion by a W3C Working Group. It defines a vocabulary for describing organisations that have gained legal entity status through formal registration. Our service could be extended both publish and consume information in this vocabulary.

Public records publication

Official records by national agencies are generally a trusted source of information; our service demonstrates how the online identity of a business can be supported by these records. This further supports the case that more government datasets should be publicly available online, as they may have unidentified uses which benefit the general population.

Public records improvements

In addition to this, our service has highlighted an area where the creation of new government datasets could add significant value to existing ones. Automated and robust identification of the company responsible for a website is a non-trivial task which could be made a lot easier through legislation.

Were the government to curate a dataset mapping domain names to company numbers, by mandating their routine disclosure, the task would become trivial, at least for UK companies. This would also make it possible to identify all domains owned by a single company.

Currently, by UK law, companies are required to publish their company number on their websites. However, as the legislation does not specify how or where, there is no standard location in which to find it. If the creation of additional datasets is undesirable, standardising a means of disclosing company numbers in a machine readable form might be a sufficient alternative. Note that this approach would not allow one to identify all domains owned by a single company.

A lightweight approach based around a well-known url scheme, like the Web Robots exclusion standard [Koster 1996], is one possible implementation of this. Also, a recent W3C 'Working Group Note' presents a 'Registered Organization Vocabulary' [Archer et al. 2013] which may be an answer to the representational needs of this problem.

In Section 6.3.4 we note that `.co.uk` registrations often include a company registration number. Regulation here could be tightened in order to mandate this information, however this would not affect other domains hierarchies such as `.com` or `.org`, so other approaches may be more worthwhile.

To take either of these approaches further would involve contacting the agencies responsible for setting policies in both areas. The UK company register is maintained by Companies House, an Executive Agency of the UK government, whereas the `.co.uk` domain tree is managed by Nominet UK, a non-profit company limited by guarantee.

New information sources or targets

Another expansion point for future development is the addition of entirely new information sources. Any form of information which adds credibility to the subject's identity may prove a worthwhile addition.

The scope of the service could also be expanded to build confidence and trust in other online entities, such as people. In which case, we could provide links between people's online identities and membership endorsements for professional organisations, or to proofs of qualification and certification.

Browser integration

As discussed briefly above, one possible avenue for improvement is Web browser integration. Instead of a pure web service based implementation, part or all of the functionality could be implemented as a web browser extension. This would allow direct access to network connection metadata, thereby addressing the problem of inaccurate information. Additionally, a browser extension could proactively present users with the identity dashboard in certain situations. These could include, when a user visits a website for the first time, or when they attempt to enter information into a web page form.

Increased jurisdiction coverage

As we discussed above, our service currently only integrates information from the United Kingdom Companies House database. Future work could improve its ability to identify the jurisdiction of a company identifier, and add support for querying the company records in these new jurisdictions.

6.5 Summary

In this chapter, we turned our attention to the trust bootstrapping problem which affects the Semantic Web. We proposed the use of trust transfer as a means of addressing this chicken-and-egg problem, to build trust in a new identity environment.

To enable trust transfer to occur, our approach encourages the growth and promotion of identity links between trust environments. These links have the potential to boost perceived entitativity and thereby encourage trust transfer.

To demonstrate this approach, in Section 6.3, we presented a website identity service which brings together a range of information sources relevant which each describe an aspect of a website's identity. This service combines network address information, domain name records, encryption certificate metadata and, where possible, company registration information. Each of these information sources acts as a link to some real-world authority, and its role and reputation in society. These authorities act as the crucial links back to the existing trust networks of the offline world.

In Section 6.4 we reviewed the shortcomings of our identity service, and discussed avenues for future work. Chiefly, these are a) the need to formally evaluate which information sources, or combinations thereof, successfully promote entitativity, b) the identification and integration of further information sources, and c) the adaption of this approach for the Semantic Web environment.

In the next chapter we build upon our contributions thus far; a lightweight vocabulary in which to describe the identity of web-based agents, and a strategy by which we can bootstrap trust in their identities. Against this background, the next chapter focuses on the challenge of combining Semantic Web information from a range of sources whilst maintaining the ability to judge trustworthiness of inferred or queried information.

Chapter 7

Effective Trust-Aware Information Management

The primary aim of our work is to enable people, or Semantic Web agents, to make effective trustworthiness judgements of Semantic Web information. Thus far, towards this goal, we have proposed a vocabulary to form the basis of an identity ecosystem for the Semantic Web (Chapter 5), and demonstrated how we might bootstrap the Semantic Web trust ecosystem through trust transfer from existing trust ecosystems (Chapter 6.

In this chapter we consider the information management challenges which arise once we have established systems for identity and reputation on the Semantic Web. These systems will not, themselves, solve the problem of trust, but instead provide a stable foundation for advanced information management systems.

As we described in our third use case (Section 2.3.1), one of the key challenges which arises when handling and acting upon information from many heterogenous sources, is judging the trustworthiness of information. Robust provenance information, both of external information and information derived from it, is fundamental requirement of dependable trustworthiness assessments.

In Section 7.1 we identify a critical incompatibility with quad-based statement storage engines, the named graph data model's notion of acceptance, and the need to represent potentially incorrect claims by other agents (Section 7.1.1). Section 7.1.2 discusses a potential solution to this problem using unique graph identifiers.

Next, in Section 7.2, we explore the need to answer questions of provenance for derived statements, looking at existing approaches such as statement-based truth maintenance (Section 7.2.3). In Section 7.2.4 we propose a novel graph-based approach, which allows questions of provenance to be answered but scales with the number of graphs in

the knowledge-base, rather than the number of triples. Then in Section 7.2.5 we propose a 'graph delta'-based storage strategy which extends this, reducing the storage requirements of this approach, and enabling certain optimisations.

Finally, we wrap-up this chapter with a summary in Section 7.3.

7.1 Named Graph Acceptance and Provenance

In order to combine, and query or reason over Semantic Web information, we import our RDF documents into knowledge-bases, known as triplestores. Early knowledge-base implementations stored RDF statements as triples, hence why they became known as triplestores. However, if information from multiple source documents was imported into the same triplestore, there was no way to determine the source of a single triple.

Since then, most triplestores now instead store RDF statements as quads – each triple is stored along with an identifier commonly used to describe the source graph – however they are still often referred to as triplestores. This addition enables much more effective information management, as it allows sets of statements to share a common identifier which, in turn, enables us to record provenance records for those statements. Recent versions of the SPARQL query language [Harris and Seaborne 2013] have embraced the quad form, to permit queries involving this graph identifier.

The ORDI SG triple-store goes one further, storing statements as 5-tuples [ORDI-Triplesets]. Their implementation uses the fifth element to identify a list of 'Tripleset' memberships for a particular statement. This 'tripleset' mechanism is employed to implement Access Control List (ACL) features on RDF graphs, restricting viewing and/or editing permissions.

In tandem with quad-based triplestores, we have the proposed Named Graph RDF semantics [Carroll et al. 2005], which we discussed in Section 4.2.1.2. The RDF semantics [Hayes 2004], as written, do not consider anything beyond triples – Named graphs, quads and 5-tuples are not discussed. The named graph semantics associate every RDF statement with a named graph, and allow a document to contain named subgraphs and reference other named graphs. The semantics also allow globally scoped URIs as graph names, allowing documents to assert statements about the content of other documents.

```
@prefix ex: <http://www.example.org/>

ex:doc/other {
    ex:doc/other#alice foaf:knows ex:doc/other#bob .
    }
```

Listing 7.1: An RDF document which makes a claim about the contents of another document

Just as it is possible in RDF to assert statements about identifiers over which you do not have authority, as we illustrate in Listing 7.1, using named graphs allows one to claim that certain statements are present in a document over which you may not necessarily have authority. The document shown in Listing 7.1 claims that the document identified by `http://www.example.org/doc/other` contains one particular statement. Unless additional precautions are taken, importing named graphs would allow malicious documents to insert arbitrary information into other documents. The Named Graph model suffices when communication only uses traditional RDF documents, it is the use of document formats which support named graphs which highlights this information separation problem.

To combat this, Carroll et al. [2005] define the notion of graph 'acceptance'; the information consumer chooses whether or not to 'accept' a graph based on their individual policies. Only the graphs 'accepted' by these policies will be used by the information consumer. The graph acceptance policies are left as an exercise for the information consumer as they are expected to be both subjective and highly contextual. Carroll et al. do, however, suggest that these policies may include some form of trustworthiness assessment. The work of Kagal et al. [2003] presents a Semantic Web policy language which may be suitable for this application.

Unfortunately this information separation problem resurfaces in applications where graph acceptance cannot reasonably be decided before information is imported. If a document were to assert statements into the graph of another document, we would have no way of correcting, or even identifying this change. As a result, unless we take special precautions, we cannot import unknown RDF documents employing the named graph semantics without the risk of undermining provenance records.

7.1.1 Limitations of Graph Acceptance

There are a number of situations where it may not be possible or desirable to determine whether or not a graph should be accepted before information is imported:

All graphs accepted

The simplest case is that of an information consumer whose acceptance policy accepts all graphs. Legitimate applications of this policy include caching services and search engines. The function of the information consumer requires that it accept all graphs, so it cannot enact any policy that would protect its provenance records.

Cannot judge acceptance early

An information consumer may not be able to judge acceptance of the information at the time it encounters it. Trustworthiness decisions are understood to be highly

contextual. If information is to be used more than once, then it may need to undergo two different acceptance assessments, as the context of the assessment will change for each use. Thus, although information might not be accepted in the first context, it may still be accepted in the second, so we cannot simply discard it after failing the first time.

Acceptance may vary over time

Finally, our perception of an information source's trustworthiness may change over time, as we learn more about its behaviour and reputation. Thus there may be situations where we wish to retract our acceptance of certain graphs, or accept a graph previously retracted. To achieve this we require an accurate means of identifying the sources of all statements in our knowledge-base.

While these situations demonstrate that this problem is of some importance, graph acceptance still serves a legitimate purpose. Graph acceptance's primary purpose is to filter untrustworthy information from a knowledge-base before it is interpreted by an application. Thus it remains applicable for applications which cannot work with named graphs, or cannot themselves decide graph acceptance.

7.1.2 Representational Limitations of Named Graphs

The crux of the problem is that we cannot uniquely identify two named graphs, asserted in different places, which share a common name. Suppose documents B and C (Listings 7.2 and 7.3) make some claims about document A (identified by the URI `http://www.example.com/doc/a`), some of which are common, and some of which are distinct. How do we separate which statements were in each?

```
@prefix ex: <http://www.example.org/>

ex:doc/a {
    ex:doc/a#alice foaf:knows ex:doc/a#bob .
    ex:doc/a#alice foaf:knows ex:doc/a#charlie .
    }
```

LISTING 7.2: Document B with the URI: `http://www.example.com/doc/b`.

```
@prefix ex: <http://www.example.org/>

ex:doc/a {
    ex:doc/a#alice foaf:knows ex:doc/a#bob .
    ex:doc/a#bob foaf:knows ex:doc/a#charlie .
    }
```

LISTING 7.3: Document C with the URI: `http://www.example.com/doc/c`.

A standard quad-based triplestore is not sufficient here; the value of the graph component of the quad can contain either the identifier of the named graph (A), or the identifier

of the carrier document (B or C), not both. With either choice there is some loss of information, the former means we lose track of where each triple came from, and the latter prevents us from supporting named graphs.

A 5-tuple approach is the next most obvious answer to this problem, adding both the graph and the carrier identifier to the triple; for example (*subject*, *predicate*, *object*, *graph name*, *document name*). Unfortunately this approach fails to solve our problem in more complex scenarios. Consider another document D (Listing 7.4) which describes the claims which were made in documents B and C. If we use the graph and document names to annotate each triple then we have no means to distinguish between the separate claims of B and C as described in D.

```
@prefix ex: <http://www.example.org/>

ex:doc/b {
    ex:doc/a {
        ex:doc/a#alice foaf:knows ex:doc/a#bob .
        ex:doc/a#alice foaf:knows ex:doc/a#charlie .
        }
    }

ex:doc/c {
    ex:doc/a {
        ex:doc/a#alice foaf:knows ex:doc/a#bob .
        ex:doc/a#bob foaf:knows ex:doc/a#charlie .
        }
    }
```

LISTING 7.4: Document D with the URI: `http://www.example.com/doc/d`.

Another question is how would D assert that the claims in C were bogus, whereas the claims in B were not? This requires an identifier within the document D which can separate the two graphs. The fact that this problem also affects documents encoding named graphs, and is not confined to knowledge-base implementations, suggests that is a wider problem with the Named Graph data model.

7.1.3 A Unique Graph Identifier

To solve these representational issues, we propose that each graph is given a unique and minimally scoped name, and the name present in the existing model should instead be expressed as an RDF statement.

In more detail, our proposed solution is to replace the graph name with an anonymous RDF blank-node identifier (see Section 4.2.1) – blank-node identifiers have defined meaning only within the scope of a single encoding or knowledge-base. We note that in this model the root graph within a document also receives a unique identifier. With this, we then have separate identifiers for the notion of a document and for the graph it contains.

We also propose that the existing method of naming a graph with a globally resolvable URI be replaced by RDF statements to the same effect. We illustrate in our examples which follow, employing the straw-man vocabulary prefix `rdfg2` to propose some provisional terms.

```
@prefix ex: <http://www.example.org/>

ex:doc/e a foaf:Document .

:_graphE {
    ...
    :_graphE2 {
        ...
        } .
    } .

ex:doc/e rdfg2:encodingOf :_graphE .
:_graphE rdfg2:encodedBy ex:doc/e .

ex:doc/e rdfg2:includesEncodingOf :_graphE2 .
:_graphE2 rdfg2:encodedWithin ex:doc/e .
```

Listing 7.5: Statements recorded about a document and its root graph

Listing 7.5 demonstrates the statements that a knowledge-base might store about a document `ex:doc/e`:

- `ex:doc/e` is a document
- There is a graph `:_graphE`
- There is a graph `:_graphE2` which is a subgraph of `:_graphE`
- `ex:doc/e` is an encoding of the graph `:_graphE` (and the converse)
- `ex:doc/e` includes an encoding of the graph `:_graphE2` (and the converse)

We can re-use these properties in order to represent claims about other documents, such as in Listings 7.6 and 7.7.

```
:_graphF { ... } .
:_graphF rdfg2:encodedBy <http://www.example.com/doc/f> .

:_graphG { ... } .
:_graphG rdfg:equivalentGraph :_graphH
:_graphH rdfg2:encodedBy <http://www.example.com/doc/f> .
```

Listing 7.6: Two equivalent claims over the content of a document

Crucially, by looking at the graph that these triples are asserted in, we can treat some as claims, and others as fact. For example, the triples we discuss in Listing 7.5 might be asserted in a graph which we created to record provenance information, and therefore be treated as facts. Whereas the triples in Listings 7.6 and 7.7, when asserted within a graph from an external source, can be treated as claims.

```
:_graphI { ... } .
:_graphI rdfg2:encodedWithin <http://www.example.com/doc/i> .

:_graphJ { ... } .
:_graphJ rdfg:subGraphOf :_graphK .
:_graphK rdfg2:encodedBy <http://www.example.com/doc/k> .
```

LISTING 7.7: Two equivalent claims that some sub-graph exists within a document

Finally, we note that, since we retain a quad-based data model, with some interworking this approach may be backwards compatible with existing quad-based triplestores. This would involve ensuring graph identifiers are unique, perhaps through a private URI scheme, translating graph names and converting them to statements.

7.1.4 Future Work

Fundamental changes to core data models are not to be taken lightly, thus it goes almost without saying that we must undertake future work in this area. Furthermore, we cannot make these changes alone, these changes will require the scrutiny, review and approval of the Semantic Web community, through academic publication and collaboration with W3C working groups.

Future work is also warranted in exploring how we might enable references to named subgraphs within other documents. It is desirable to reference them directly, however we have deliberately removed the ability to assign global identifiers to them. One possible solution is to permit allow graph identifiers to also be fragment identifiers. We wish to undertake further work here in order to investigate the importance of this requirement, and to explore and evaluate different approaches.

7.2 Managing Derived Information and Provenance

As Semantic Web agents increase in sophistication, we can expect them to do more than simply collate facts from downloaded documents. They may gain the ability to publish new information, perhaps from first-hand observations, formal reasoning, or from data aggregation and analysis. To retain the ability to answer questions of trustworthiness we will require provenance information recording the origin of this new information.

7.2.1 Inference

Performing complete inference over a knowledge-base can be a very resource intensive task, in terms of both computation and memory. As a result, reasoning is perhaps employed less often than we might think. To combat this, there are certain optimisation techniques which can help to mitigate these resource costs:

Forward chaining & materialisation

Forward chaining involves performing inference ahead of time and materialising (or asserting) the inferred triples in the knowledge-base. Queries will then be able to take advantage of inference by virtue of the fact that every entailment is already present in the knowledge-base.

Backward chaining & query rewriting

Backward chaining works from a given goal query, applying the inference rules against the goal in order to find matches in the knowledge-base. While this may make queries more expensive, it avoids the up-front cost of reasoning, as well as the costs associated with storing the materialised statements. However, the amortised additional query cost may eventually exceed the up-front cost of reasoning.

Hybrid

A hybrid approach, which combines forward and backward chaining, is also possible. Harris and Gibbins [2003] describe one such hybrid implementation in the 3store knowledge-base. They identified certain inference rules for which the results were seldom queried, or had significant storage costs. These rules, specifically rdfs inference rules 6 and 9 concerning sub-property and sub-class inference, were applied only through backward chaining. Their engine then performed forward chaining for all inference rules.

Some degree of forward-chaining is attractive in most applications, as amortised costs and query execution times are lower. As we discuss in Section 7.1.1, there are certain factors, such as trust, which may prompt us to add or remove information from our knowledge-base. Unfortunately, the addition or removal of information from a knowledge-base has the potential to invalidate all inferred and materialised information. Unless the reasoning engine can determine the full implications of the change, it will necessitate a complete re-computation of materialised inference results. If a complete re-computation happens too frequently the costs of approaches involving forward-chaining will exceed that of a pure backward-chaining approach.

As an aside, Fensel and van Harmelen [2007] suggest a novel optimisation, noting that often applications do not necessarily require complete inference over a knowledge-base. Instead, an application could query and reason over a small subset of the knowledge-base, iteratively growing the size of that subset until sufficient result quality is achieved, or some time limit is reached.

Against this background, there is a need for strategies which minimise the need to perform a full re-computation of inferred statements.

7.2.2 Provenance

Assuming that an application has performed some derivation over its knowledge-base, such as inference, in order for us to judge the trustworthiness if its results, we again require good provenance records. The degree of provenance information recorded for inferred information can vary considerably. Some implementations will not record any provenance information at all. Some will indicate only that information was the result of inference. Some may go further, detailing which graphs were included in the inference task. The most extensive (and also expensive) that we have seen is a system which recorded the complete set of deductive dependencies between statements (see Section 7.2.3).

The degree of this information constrains our ability to make trustworthiness assessments. The first level is insufficient for any worthwhile trustworthiness assessment. The second allows us to judge the trustworthiness of inferred information separately from the rest of the knowledge-base, however, the result will only measure the aggregate trustworthiness of the whole knowledge-base. The third is better again, but only measures the aggregate trustworthiness of the graphs included in the inference, rather than the true trustworthiness of an inferred statement. Finally, while the fourth is sufficient for a good, well focused, trustworthiness assessment, it provides an order of magnitude more information than is required for our purposes.

7.2.3 Statement Truth Maintenance

The approach of storing a complete deductive dependancy graph is very similar to what Doyle [1979] described as 'justifications' stored by his Truth Maintenance Systems (TMSs). Doyle's TMS operated over two types of data structure; 'nodes' representing beliefs, and 'justifications' which represent reasons for beliefs. Broekstra and Kampman [2003] explore the empirical performance of a per-statement provenance based retraction-management system, which is effectively a restricted version of Doyle's TMS. This implementation tracks deductive dependencies between statements so that when it removes statements from the knowledge-base it can also easily remove any statements which were deduced from them and so forth.

While this approach is named 'truth' maintenance, it does not mean that it can only be used to track trustworthy and 'true' information. Rather, it tracks dependencies between beliefs, facts or statements – true or otherwise. The approach is so named because, as originally concieved, it existed within a system which assumed all stored information to be true [Doyle 1979].

Table 7.1 illustrates this form of dependency record. Statement 2 is dependent on statement 1, and statement 3 is dependent on both statements 1 and 2. Therefore if

Statement ID	Dependency ID
2	1
3	1
3	2
5	4

TABLE 7.1: Example dependency table

statement 1 were to be removed, so too would statements 1 and 2. We note that these records are a tabular representation of a directed acyclic graph, which we may refer to as an entailment graph.

This approach has a space complexity of $\mathcal{O}(|I|)$ where I is the set of inferred statements, as it must create at least one extra record (though likely more) for every entailment stored. Performing the removal of a single statement and its consequences requires the traversal of its full entailment graph, this has a time and space complexity of $\mathcal{O}(|E|)$ and $\mathcal{O}(|V|)$ where E is the set of dependency records, and V is the set of statements which are dependent on the statement in question. Thus the complexity of removing a single statement varies depending on the number of statements which have been inferred from it.

Perhaps as a result of this complexity, very few triplestores offer any form of 'justification' records or truth maintenance features. The need for truth maintenance in information management systems has, perhaps for the time being, been overshadowed by the desire to build triplestores capable of storing, retrieving and querying large numbers of triples, with high performance. The implementation described by Broekstra and Kampman [2003] has since dropped this feature, due to the computational and space costs of calculating and storing dependencies in large scale environments[1]. The Bigdata RDF Database[2] retains per-statement truth maintenance capabilities, however at the time of writing the documentation states that it does not support 'quads', so therefore does also not support named graphs.

7.2.4 A Graph-Based Approach

Rather than recording provenance on a per-statement basis, instead we propose to record provenance on a per-graph basis. In more detail, we assert inferred statements in different graphs depending on the set of asserted graphs which were inferred from, and it is against these graphs which provenance is recorded. The granularity of per-statement provenance information goes beyond the requirement of most Semantic Web

[1]Sesame mailing list post: See `http://sourceforge.net/mailarchive/message.php?msg_id=28761869`

[2]Bigdata: See `http://www.systap.com/bigdata.htm`

applications. Since reasoning is repeatable, if we find that we require fine-grained, per-statement, information, we can simply repeat the reasoning process – so long as we store sufficient information to do so.

Unless fine-grained information is required regularly, the amortised cost of repeating the reasoning process may be less than the long term cost of storing and maintaining the information. This approach is likely to have improved scaling properties as metadata is recorded about groups of statements, rather than single statements.

In most cases, unless constraints prevent it, reasoning agents aim to operate over the total deductive closure of their knowledge base. However, as we discussed in Section 7.1.1, reasoning will not always be a one-off process. Semantic Web agents will need to add and remove graphs from their knowledge bases over time as documents are updated or corrected, or if new reputation informations alters trustworthiness judgements.

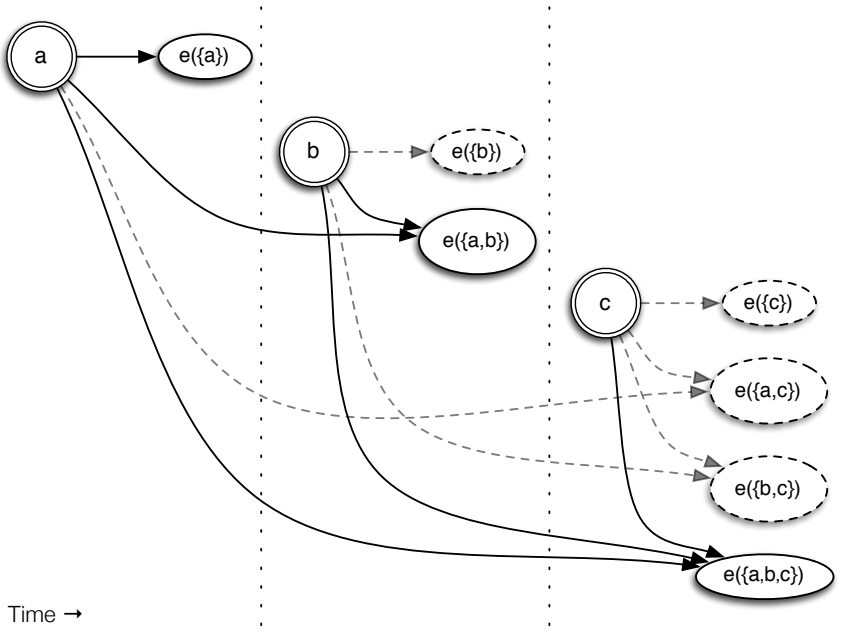


FIGURE 7.1: Global entailment graph construction over time

We use the function $e(G)$ to denote the deductive closure over the set of graphs G, where the entailment is performed over the union of the graphs in G. Figure 7.1 illustrates the successive global entailment graphs through the incremental addition of 3 graphs a, b and c to a knowledge base. Each time a graph is added the global entailment graph must be recomputed. In this figure, the dashed arrows and ovals indicate the possible entailment graphs which do not represent the global entailment graph and its computation. As new documents are added, a complete re-computation of the global entailment closure will involve a certain amount of repeated computation. That is, at step t, a re-computation will duplicate the computation undertaken at step $t-1$, resulting in a significant overhead, which continues to grow as the knowledge-base grows.

Given a set S of 3 graphs a, b and c, the possible entailment graphs of S is the powerset of S less the empty set $(\mathcal{P}(S) \setminus \emptyset)$[3]. Thus for a set S of asserted graphs there will be a maximum of $2^{|S|} - 1$ different entailment graphs. Figure 7.2(a) illustrates the 7 possible entailment graphs in this scenario, indicating the entailment relationships between graphs with arrows between nodes, from source graph to entailment graph.

Figures 7.2(b) and 7.2(c) illustrate the subset relationships between the graphs from Figure 7.2(a). A set of statements (a graph) is by definition a subset of its own entailment. When considering graphs, we refer to subset relationships between graphs as subgraph relationships.

Given a set of graphs S, our graph-based approach has the potential to create $2^{|S|} - 1$ inferred graphs in the worst case. For our approach, the worst case is when the inference results in each possible inference graph containing only one statement each, as this results in the highest storage overhead per inferred statement. Thus, in the worst case, the space complexity of our approach is $\mathcal{O}(2^{|S|}-1)$, as we store a constant number of statements to describe the provenance of a graph. Therefore, our approach will, in the worst case, have the same complexity as Broekstra and Kampman's justification records, $\mathcal{O}(n)$, where n is the number of inferred triples.

In any case but the worst, either some inferred graphs will be empty, or some inferred graphs will contain multiple triples. The first will effectively terminate the dependency tree early, which has the potential to significantly reduce the number of inferred graphs. The second will result in a lower per inferred statement overhead.

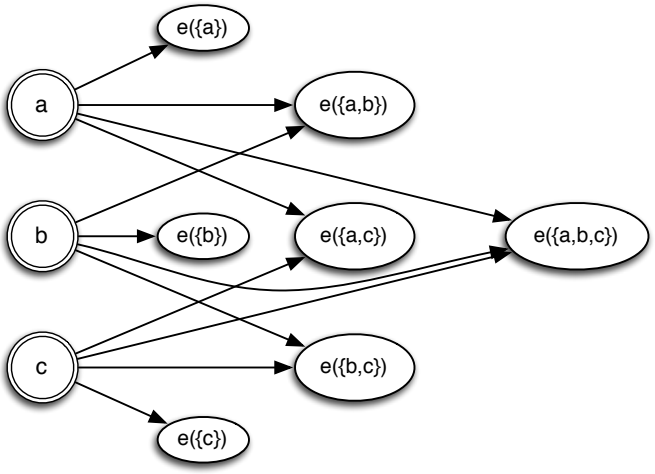
Thus, the benefits of this graph-based approach are two-fold; first, it gives us the means by which to record provenance data at higher granularity, scaling with the number of graphs rather than the number of triples. Second, it provides us with the means of identifying which subsets of a global entailment graph are affected by any changes to the knowledge-base. Enabling us to perform a much smaller inference task than computing the entire deductive closure.

For example, with graphs a, b and c, where c has been added, the optimal approach would compute only $e(\{a, b, c\}) \setminus e(\{a, b\})$ to ensure the global deductive closure is re-computed (which includes $e(\{c\})$, $e(\{a, c\})$ and $e(\{b, c\})$). The shaded portion of Figure 7.3 demonstrates the portion of the entailment graph which needs to be freshly computed.

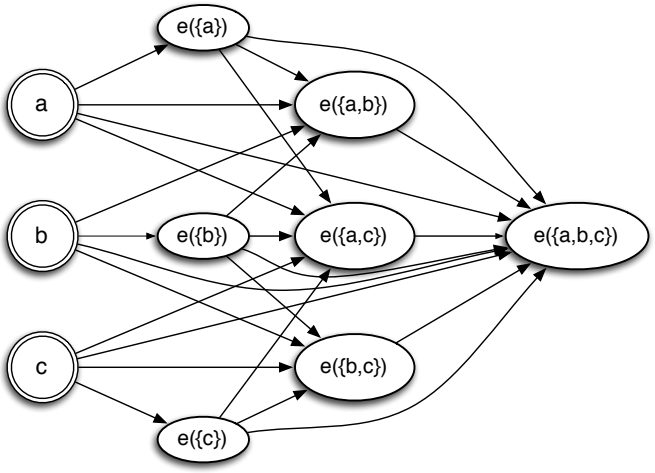
7.2.5 Graph Delta Storage

Building on this graph-based provenance approach, we identify another opportunity for optimisation. Given the antecedents of an entailment graph are each subgraphs of it, if these graphs are all stored independently there is a significant information duplication.

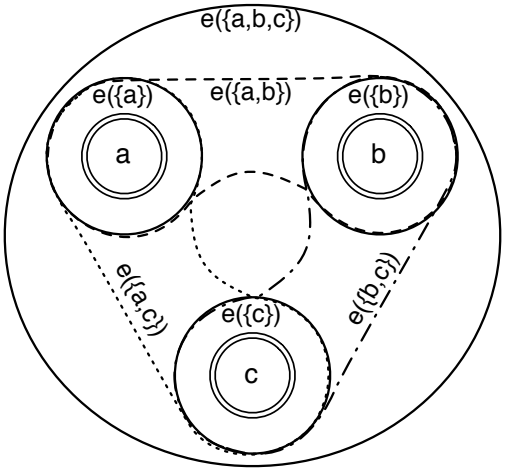
[3]The power-set of a set S is the set of all possible subsets of S, according to set theory.



(a) Graphs a, b and c and their possible entailment graphs.



(b) Subgraph relationships between a, b and c and their entailment graphs.



(c) Venn diagram of subgraph relationships in figure 7.2(b)

FIGURE 7.2: Relationships between S and its entailment graphs.

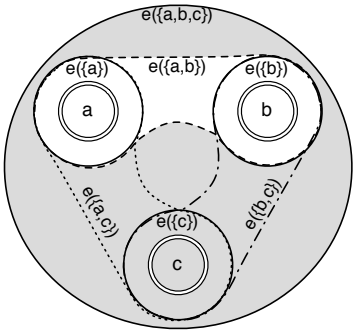


FIGURE 7.3: The portions of Figure 7.2(c) which are dependent on c (shaded).

We can exploit this subgraph relationship to store only newly-inferred triples in each graph, using the subgraph tree to retrieve inherited statements.

$$I(X) = e(X) \setminus \bigcup_{\substack{Y \in \mathcal{P}(X) \\ Y \subsetneq X}} e(Y) \tag{7.1}$$

Equation 7.1 formally defines this approach; for a set of graphs X, the result of the function is the deductive closure of X, minus any statements in the deductive closure of any subset of X. For example, where X is a three graph set, as in Figure 7.2(c), this is represented by the region inside $e(a, b, c)$ which is not also inside $e(a, b)$, $e(b, c)$ and $e(a, c)$.

Succinctly, this approach stores only the delta between the entailment graph and the union of the subgraphs, thus reducing duplicate information. Without this optimisation the storage requirements of entailed graphs monotonically increase with each new graph added to the knowledge-base.

7.2.6 Implementation

In order to explore the viability of our two proposals we developed a prototype implementation built upon the Apache Jena[4] RDF knowledge-base, and the Named Graphs API for Jena (NG4J)[5] support library.

We created a Semantic Web client agent which demonstrated our unique graph naming proposals. When our agent retrieves RDF content through URI resolution, it applies our naming proposals to the data before it is added to our knowledge-base. In addition, it records provenance information for retrieved data, expressed using the `prov` vocabulary.

[4]Apache Jena: See `https://jena.apache.org/`
[5]NG4J: See `http://wifo5-03.informatik.uni-mannheim.de/bizer/ng4j/`

Our prototype reasoner separated inferred triples into separate graphs based on their antecedent graphs, and recorded in a provenance log that those graphs were the result of an inference action.

```
_:g1 { test:a test:some_prop test:b . }
_:g2 { test:some_prop rdfs:subPropertyOf test:other_prop . }
_:g3 { test:some_prop rdfs:subPropertyOf test:third_prop . }

_:ig1 { test:a test:other_prop test:b . }
_:ig2 { test:a test:third_prop test:b . }
_:ig3 {}

:provenance {
    _:g1 rdf:type rdfg:Graph .
    _:g1 prv:retrievedBy _:da1 .
    _:g1 rdfg2:encodedBy <http://www.example.org/1.rdf> .

    _:g2 rdf:type rdfg:Graph .
    _:g2 prv:retrievedBy _:da2 .
    _:g2 rdfg2:encodedBy <http://www.example.org/2.rdf> .

    _:g3 rdf:type rdfg:Graph .
    _:g3 prv:retrievedBy _:da3 .
    _:g3 rdfg2:encodedBy <http://www.example.org/3.rdf> .

    _:da1 rdf:type prv:DataAccess .
    _:da1 prv:performedAt "..." .
    _:da1 prv:accessedResource <http://www.example.org/1.rdf> .
    _:da1 prv:performedBy :self .

    _:da2 rdf:type prv:DataAccess .
    _:da2 prv:performedAt "..." .
    _:da2 prv:accessedResource <http://www.example.org/2.rdf> .
    _:da2 prv:performedBy :self .

    _:da3 rdf:type prv:DataAccess .
    _:da3 prv:performedAt "..." .
    _:da3 prv:accessedResource <http://www.example.org/3.rdf> .
    _:da3 prv:performedBy :self .

    _:ig1 rdf:type rdfg:Graph .
    _:ig1 :inferredFromGraph _:g1 .
    _:ig1 :inferredFromGraph _:g2 .

    _:ig2 rdf:type rdfg:Graph .
    _:ig2 :inferredFromGraph _:g1 .
    _:ig2 :inferredFromGraph _:g3 .

    _:ig3 rdf:type rdfg:Graph .
    _:ig1 rdfg:subGraphOf _:ig3 .
    _:ig2 rdfg:subGraphOf _:ig3 .
    _:ig3 :inferredFromGraph _:g1 .
    _:ig3 :inferredFromGraph _:g2 .
    _:ig3 :inferredFromGraph _:g3 .
}
```

LISTING 7.8: Inferred triples separated into graphs (TriG Syntax)

Listing 7.8 illustrates, using TriG syntax[6], the separation of inferred information into graphs, our unique graph naming proposals, and the provenance records we mentioned above.

[6]TriG Syntax specification: See `http://wifo5-03.informatik.uni-mannheim.de/bizer/trig/`

While we proved that out approach was viable, unfortunately our implementation was far from optimal. An ideal design would be a reasoner which accepted 'quads' as input, and flexibly allowed the 'graph' of newly-inferred quads to be set through a pluggable mechanism.

However, due to the architecture and complexity of the Jena Reasoner codebase, we were unable to achieve our idealised design goals. There were a number of issues which contributed to this:

- Internally, the reasoner uses a hybrid algorithm combining forward and backward chaining; which results in the creation of 'rules' which are no longer associated with a named graph.
- Derivation logging does not record the creation of rules from triples, only the creation of triples from rules.
- At the point where new rules or triples are derived, there remains no reference to the original antecedent triples, so their source graphs can not be easily identified.

While we were able to work around these issues in order to build our prototype, it was not a trivial process. Addressing the first two issues involved fairly invasive changes to the Jena Reasoner, as there were no good extension points for our purposes. Our workaround for the last of the above issues amounted to searching our existing named graphs to see which of them contained each antecedent triple. As one might suspect, this is a particularly expensive operation, which scales poorly as the size of the knowledge-base grows.

As a result of this, while we understand the scaling properties of our provenance records in terms of triples stored, we were not able to explore the performance implications on the reasoning process.

7.2.7 Future Work

Further work is required to explore the empirical performance of our approach as it will vary depending on the properties of the input data. We plan to undertake this on two fronts:

First, after addressing the implementation issues of our prototype, we plan to perform some benchmarking of our proposals. Popular reasoning-based benchmarks, such as Lehigh University Benchmark (LUBM) [Guo et al. 2005], are generally intended to test OWL-based reasoners, and so are not immediately applicable. Therefore, we may either have to create new benchmark datasets for RDFS reasoning, or explore the compatibility of OWL-based reasoning with our approach. To test our approach, an RDFS reasoning

benchmark would need to provide a variable frequency distribution of entailment types, entailment chain lengths, the number of graphs and statements per graph.

Secondly, in order to investigate how our two proposals perform with real-world data we wish to study the inference characteristics of real-world RDF datasets. We wish to better understand how much information is inferred in practice, and how deep the entailment trees grow to be. Also we would seek to identify how much or little these characteristics vary across subject domains and data sources.

Finally, our work makes the assumption that the statements entailed from a given graph are always a superset of the original graph. Future work could investigate the ramifications of this assumption as other forms of reasoning, or other application processes may not have this property.

7.3 Summary

In this chapter we targeted challenges involving the management of information and provenance records, which are critical prerequisites for well-founded trustworthiness judgements. Communicating provenance information is one of many applications for named graphs, however we identify some fundamental problems with the named graph proposal. First, that the named graph 'acceptance' filter is not appropriate in a range of applications, and second, that there are situations in the current data model where it is impossible to distinguish two separate graphs sharing the same name. To address this, we proposed replacing the graph name in the RDF Named Graph data model with a unique identifier, representing the name as a separate statement.

Next, we explored the provenance of information derived through processes such as inference. We proposed a novel graph-based approach to truth maintenance records, which trades the granularity of per-statement metadata for the computational and memory savings afforded by working with groups of statements. Building on this, we proposed a delta-based storage strategy for derived data, as a further storage space optimisation.

Chapter 8

Conclusions and Future Work

Trust, in the context of the Semantic Web, presents a number of challenges, many of which are socio-technical in nature. In this thesis, we took an end-to-end look at these problems, identifying areas where technical innovation can make progress towards larger socio-technical problems. We took a pragmatic approach to some of these problems, developing a number of contributions across the breadth of this research area. In brief, our contributions are: i) an identity ontology for describing web-based agents, ii) a novel approach to addressing the trust network bootstrapping problem, iii) an improvement to the RDF Named Graph semantics, and iv) a novel graph-based approach to recording provenance, which enables us to make intelligent computational and storage savings.

In the sections which follow we summarise and discuss our contributions in more detail, also discussing avenues of future work which build upon them.

8.1 Identity

Work to date has generally represented Semantic Web agents only as members of the `foaf:Agent` class, providing little other information. A notion of agent identity is critical to enabling effective inter-agent discourse, and the growth of trust and reputation within the Semantic Web environment. Thus, we identified a strong need for a more descriptive Semantic Web agent identity vocabulary.

Against this background, we developed an identity ontology for describing web-based agents. It is designed to allow the description of any web-based agent, not just those which adopt support for it, so that it is not beset with a bootstrapping problem. This ontology provides a foundation for future research and for the growth of trust and reputation within the Semantic Web.

We plan to apply this vocabulary in future applications, in order to further evaluate its suitability. We anticipate that co-reference (Section 4.4.3) will be an issue with our

vocabulary, as we have taken an intensional definition stance, so this will involve the development of co-reference resolution strategies for use with our vocabulary. Future work could also address the lack of an ontology for describing DNS queries and responses, as this information underpins the operation of the Web.

8.2 Bootstrapping Trust

Against the background of our contribution thus far, we focussed on the acknowledged research problem of bootstrapping trust in new environments, such as the Semantic Web. We proposed a novel approach which harnesses open data to bootstrap trust in new trust environments. This approach brings together public records published by a range of trusted institutions in order to build confidence in an identity in this new environment. It is this confidence in identity which encourages trust transfer into the trust environment.

To demonstrate this approach, we developed a website identity service which integrates this freely available information into a common dashboard. We combine network address information, domain name records, encryption certificate metadata and, where possible, company registration information. Each of these information sources acts as a link to some real-world authority, and its role and reputation in society. Through this dashboard we demonstrate how records from these institutions can be correlated to build confidence in an identity and ultimately encourage trust transfer.

Summarising from Section 6.4.1, our work has given rise to a number of different avenues for future work:

Formal Evaluation

We are conscious that trust transfer is highly subjective, therefore a formal study to determine the effectiveness of our system is desirable. In addition, varying the combinations of information panel presented would allow us to measure their effectiveness, as well as which work well together.

Public records improvement and open data activism

Promoting open data publishing, Semantic Web data publishing, and seeking new applications for existing datasets represents a significant avenue for future work. Our work has demonstrated a new situation where we can draw value from existing public records. This is another argument in favour of the open publishing of government data.

In addition to this, our service has highlighted an opportunity to add value to existing datasets. The task of mapping domain names to companies would be significantly easier if there was canonical method for doing so. Currently, by UK

law, companies are required to publish their company number on their websites, however the format and means of doing so is not specified. Mandating centralised records, or specifying a machine-readable disclosure process, would greatly simplify this task.

New information sources or subjects

The integration of information sources of new types or in new jurisdictions is an obvious area of future work which we may investigate. We could also extend our platform to help build confidence and trust in other online entities, such as people. In which case, we could provide links between people's online identities and membership endorsements for professional organisations, or to proofs of qualification and certification.

8.3 Information Integrity and Provenance

Lastly, we presented contributions in the areas of information integrity and provenance, which are both critical prerequisites for well-founded judgements of information trustworthiness.

Named graphs are necessary for a range of advanced tasks such as inter-agent discourse, and recording or communicating provenance information. We identify serious representational limitations with the named graph proposal which affect the ability to cleanly represent claims of other agents.

Against this background, in order to address these concerns we proposed changes to the graph naming mechanisms in the RDF Named Graph data model. It goes almost without saying that future work is required in this area. Our proposed changes to the Named Graph data model will require peer review by the Semantic Web community, and collaboration with W3C working groups.

Next, we explored the provenance of information derived through processes such as inference. We proposed a novel graph-based approach for recording the provenance of information derived from a range of sources, such as inference. It trades the granularity of per-statement metadata for the computational and memory savings afforded by working with groups of statements. This approach maintains the ability to answer graph-level provenance questions and allows new optimisations through which we can avoid needless repeat computation. Building on this further, we proposed a delta-based storage strategy for derived data which allow us to avoid duplication.

We plan to undertake further work to explore the empirical performance of our approach, as its performance will vary depending on the properties of the input data. In addition, we wish to study the inference characteristics of real-world RDF datasets, to look for patterns and characteristics which may affect our performance with real data.

Appendix A

Identity Ontology

This appendix reproduces our Web Server identity introduced in Chapter 5. An electronic copy can be retrieved from http://purl.org/mcobden/identity.

```
<?xml version="1.0"?>


<!DOCTYPE rdf:RDF [
    <!ENTITY wot "http://xmlns.com/wot/0.1/" >
    <!ENTITY terms "http://purl.org/dc/terms/" >
    <!ENTITY foaf "http://xmlns.com/foaf/0.1/" >
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY WebArch "http://sw.nokia.com/WebArch-1/" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY prov "http://purl.org/net/provenance/ns#" >
    <!ENTITY powder-s "http://www.w3.org/2007/05/powder-s#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdfg "http://www.w3.org/2004/03/trix/rdfg-1/" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY contact "http://www.w3.org/2000/10/swap/pim/contact#" >
    <!ENTITY irw "http://www.ontologydesignpatterns.org/ont/web/irw.owl#" >
]>


<rdf:RDF xmlns="http://purl.org/mcobden/identity#"
     xml:base="http://purl.org/mcobden/identity"
     xmlns:irw="http://www.ontologydesignpatterns.org/ont/web/irw.owl#"
     xmlns:prov="http://purl.org/net/provenance/ns#"
     xmlns:foaf="http://xmlns.com/foaf/0.1/"
     xmlns:terms="http://purl.org/dc/terms/"
     xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#"
     xmlns:wot="http://xmlns.com/wot/0.1/"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:powder-s="http://www.w3.org/2007/05/powder-s#"
     xmlns:WebArch="http://sw.nokia.com/WebArch-1/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:rdfg="http://www.w3.org/2004/03/trix/rdfg-1/">
    <owl:Ontology rdf:about="http://purl.org/mcobden/identity">
        <owl:imports rdf:resource="http://purl.org/net/provenance/ns#"/>
```

```
        <owl:imports rdf:resource="http://sw.nokia.com/schemas/general/WebArch↩
-1.4.owl"/>
        <owl:imports rdf:resource="http://www.ontologydesignpatterns.org/ont/web↩
/irw.owl"/>
        <owl:imports rdf:resource="http://xmlns.com/foaf/0.1/"/>
        <owl:imports rdf:resource="http://xmlns.com/wot/0.1/"/>
    </owl:Ontology>



    <!--
    ↩
///////////////////////////////////////////////////////////////////////////////////↩

    //
    // Annotation properties
    //
    ↩
///////////////////////////////////////////////////////////////////////////////////↩

     -->

    <owl:AnnotationProperty rdf:about="&rdfs;label"/>
    <owl:AnnotationProperty rdf:about="&rdfs;comment"/>



    <!--
    ↩
///////////////////////////////////////////////////////////////////////////////////↩

    //
    // Datatypes
    //
    ↩
///////////////////////////////////////////////////////////////////////////////////↩

     -->




    <!-- http://www.w3.org/2001/XMLSchema-datatypes#string -->

    <rdfs:Datatype rdf:about="http://www.w3.org/2001/XMLSchema-datatypes#string↩
"/>



    <!--
    ↩
///////////////////////////////////////////////////////////////////////////////////↩

    //
    // Object Properties
    //
    ↩
///////////////////////////////////////////////////////////////////////////////////↩
```

```
     -->



    <!-- http://purl.org/mcobden/identity#delegatedBy -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#delegatedBy↩
">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:label xml:lang="en">delegated by</rdfs:label>
        <rdfs:comment xml:lang="en">Denotes the HTTP server which enacts this ↩
delegation.</rdfs:comment>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#Delegation↩
"/>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#↩
HTTPServerBehaviour"/>
        <owl:inverseOf rdf:resource="http://purl.org/mcobden/identity#↩
hasDelegation"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#delegatedLocation -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#↩
delegatedLocation">
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#Delegation↩
"/>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#WebLocation↩
"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#delegates -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#delegates">
        <rdfs:label xml:lang="en">delegates</rdfs:label>
        <rdfs:comment xml:lang="en">This HTTP Server Behaviour delegates ↩
requests matching specified HTTP Location to another Server.</rdfs:comment>
        <rdfs:subPropertyOf rdf:resource="http://purl.org/mcobden/identity#↩
serves"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#delegationTarget -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#↩
delegationTarget">
        <rdf:type rdf:resource="&owl;FunctionalProperty"/>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#Delegation↩
"/>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#↩
HTTPServerBehaviour"/>
        <owl:inverseOf rdf:resource="http://purl.org/mcobden/identity#↩
targetOfDelegation"/>
    </owl:ObjectProperty>
```

```
    <!-- http://purl.org/mcobden/identity#encompasses -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#encompasses↩
">
        <rdf:type rdf:resource="&owl;AsymmetricProperty"/>
        <rdf:type rdf:resource="&owl;IrreflexiveProperty"/>
        <rdf:type rdf:resource="&owl;TransitiveProperty"/>
        <rdfs:label xml:lang="en">encompasses</rdfs:label>
        <rdfs:comment xml:lang="en">This Web location specification encompasses ↩
the following Web location specification.</rdfs:comment>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#WebLocation↩
"/>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#WebLocation↩
"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#hasBehaviour -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#hasBehaviour↩
">
        <rdfs:label xml:lang="en">has behaviour</rdfs:label>
        <rdfs:comment xml:lang="en">This Agent Behaviour comprises a part of ↩
this Agent.</rdfs:comment>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#Agent"/>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#↩
AgentBehaviour"/>
        <owl:inverseOf rdf:resource="http://purl.org/mcobden/identity#↩
isBehaviourOf"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#hasDelegation -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#↩
hasDelegation">
        <rdfs:label xml:lang="en">delegates HTTP Location</rdfs:label>
        <rdfs:comment xml:lang="en">This HTTP Server enacts the specified ↩
delegation.</rdfs:comment>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#Delegation"/>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#↩
HTTPServerBehaviour"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#isBehaviourOf -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#↩
isBehaviourOf">
        <rdfs:label xml:lang="en">is behaviour of</rdfs:label>
        <rdfs:comment xml:lang="en">This Agent Behaviour comprises a part of the↩
 specified Agent.</rdfs:comment>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#Agent"/>
```

```
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#↩
AgentBehaviour"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#isDelegated -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#isDelegated↩
">
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#Delegation"/>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#WebLocation↩
"/>
        <owl:inverseOf rdf:resource="http://purl.org/mcobden/identity#↩
delegatedLocation"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#isEncompassedBy -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#↩
isEncompassedBy">
        <rdf:type rdf:resource="&owl;AsymmetricProperty"/>
        <rdf:type rdf:resource="&owl;IrreflexiveProperty"/>
        <rdf:type rdf:resource="&owl;TransitiveProperty"/>
        <rdfs:label xml:lang="en">is encompassed by</rdfs:label>
        <rdfs:comment xml:lang="en">This Web location specification is ↩
encompassed by the following Web location specification.</rdfs:comment>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#WebLocation↩
"/>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#WebLocation↩
"/>
        <owl:inverseOf rdf:resource="http://purl.org/mcobden/identity#↩
encompasses"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#isServedBy -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#isServedBy">
        <rdfs:label xml:lang="en">is served by</rdfs:label>
        <rdfs:comment xml:lang="en">The specified HTTP Sever Behaviour serves ↩
requests matching this HTTP Location specification.</rdfs:comment>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#↩
HTTPServerBehaviour"/>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#WebLocation↩
"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#serves -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#serves">
        <rdfs:label xml:lang="en">serves</rdfs:label>
        <rdfs:comment xml:lang="en">This HTTP Server Behaviour serves requests ↩
matching specified HTTP Location.</rdfs:comment>
```

```
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#↩
HTTPServerBehaviour"/>
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#WebLocation↩
"/>
        <owl:inverseOf rdf:resource="http://purl.org/mcobden/identity#isServedBy↩
"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#targetOfDelegation -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#↩
targetOfDelegation">
        <rdfs:range rdf:resource="http://purl.org/mcobden/identity#Delegation"/>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#↩
HTTPServerBehaviour"/>
    </owl:ObjectProperty>



    <!-- http://purl.org/mcobden/identity#usesPublicKey -->

    <owl:ObjectProperty rdf:about="http://purl.org/mcobden/identity#↩
usesPublicKey">
        <rdfs:label rdf:datatype="&xsd;string">uses public key</rdfs:label>
        <rdfs:comment xml:lang="en">This Agent Behaviour is known to employ a ↩
key-pair with the specified Public Key.</rdfs:comment>
        <rdfs:domain rdf:resource="http://purl.org/mcobden/identity#↩
AgentBehaviour"/>
        <rdfs:range rdf:resource="&wot;PubKey"/>
        <rdfs:subPropertyOf rdf:resource="&wot;hasKey"/>
    </owl:ObjectProperty>



    <!-- http://www.w3.org/2002/07/owl#topObjectProperty -->

    <owl:ObjectProperty rdf:about="&owl;topObjectProperty"/>



    <!--
    ↩
///////////////////////////////////////////////////////////////////////////////↩

    //
    // Data properties
    //
    ↩
///////////////////////////////////////////////////////////////////////////////↩

     -->



    <!-- http://www.w3.org/2007/05/powder-s#matchesregex -->
```

```
    <owl:DatatypeProperty rdf:about="&powder-s;matchesregex"/>



    <!--
    ↩
///////////////////////////////////////////////////////////////////////////////////////↩

    //
    // Classes
    //
    ↩
///////////////////////////////////////////////////////////////////////////////////////↩

     -->



    <!-- http://purl.org/dc/terms/Agent -->

    <owl:Class rdf:about="&terms;Agent">
        <owl:equivalentClass rdf:resource="http://purl.org/mcobden/identity#↩
Agent"/>
    </owl:Class>



    <!-- http://purl.org/net/provenance/ns#Actor -->

    <owl:Class rdf:about="&prov;Actor">
        <owl:equivalentClass rdf:resource="http://purl.org/mcobden/identity#↩
Agent"/>
    </owl:Class>



    <!-- http://sw.nokia.com/WebArch-1/Server -->

    <owl:Class rdf:about="&WebArch;Server">
        <owl:equivalentClass rdf:resource="http://purl.org/mcobden/identity#↩
WebServerAgent"/>
    </owl:Class>



    <!-- http://www.ontologydesignpatterns.org/ont/web/irw.owl#URI -->

    <owl:Class rdf:about="&irw;URI">
        <owl:equivalentClass rdf:resource="http://purl.org/mcobden/identity#↩
WebLocation"/>
    </owl:Class>



    <!-- http://www.ontologydesignpatterns.org/ont/web/irw.owl#WebClient -->

    <rdf:Description rdf:about="&irw;WebClient">
```

```
        <owl:equivalentClass rdf:resource="http://purl.org/mcobden/identity#↩
WebClientAgent"/>
    </rdf:Description>



    <!-- http://www.ontologydesignpatterns.org/ont/web/irw.owl#WebServer -->

    <owl:Class rdf:about="&irw;WebServer">
        <owl:equivalentClass rdf:resource="http://purl.org/mcobden/identity#↩
WebServerAgent"/>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#Agent -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#Agent">
        <rdfs:label xml:lang="en">Agent</rdfs:label>
        <owl:equivalentClass rdf:resource="&foaf;Agent"/>
        <owl:disjointWith rdf:resource="http://purl.org/mcobden/identity#↩
AgentBehaviour"/>
        <owl:disjointWith rdf:resource="http://purl.org/mcobden/identity#↩
Delegation"/>
        <owl:disjointWith rdf:resource="http://purl.org/mcobden/identity#↩
WebLocation"/>
        <rdfs:comment xml:lang="en">An Agent</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#AgentBehaviour -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#AgentBehaviour">
        <rdfs:label xml:lang="en">Agent Behaviour</rdfs:label>
        <owl:disjointWith rdf:resource="http://purl.org/mcobden/identity#↩
Delegation"/>
        <owl:disjointWith rdf:resource="http://purl.org/mcobden/identity#↩
WebLocation"/>
        <rdfs:comment xml:lang="en">An aspect or part of the inherent behaviour ↩
of an Agent.</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#Delegation -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#Delegation">
        <owl:disjointWith rdf:resource="http://purl.org/mcobden/identity#↩
WebLocation"/>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#HTTPBehaviour -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#HTTPBehaviour">
        <rdfs:label xml:lang="en">HTTP Behaviour</rdfs:label>
        <owl:equivalentClass>
```

```
            <owl:Class>
                <owl:unionOf rdf:parseType="Collection">
                    <rdf:Description rdf:about="http://purl.org/mcobden/identity↩
#HTTPClientBehaviour"/>
                    <rdf:Description rdf:about="http://purl.org/mcobden/identity↩
#HTTPServerBehaviour"/>
                </owl:unionOf>
            </owl:Class>
        </owl:equivalentClass>
        <rdfs:subClassOf rdf:resource="http://purl.org/mcobden/identity#↩
AgentBehaviour"/>
        <rdfs:comment xml:lang="en">A facet of an agent which specialises in ↩
HTTP.</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#HTTPClientBehaviour -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#HTTPClientBehaviour">
        <rdfs:label xml:lang="en">HTTP Client Behaviour</rdfs:label>
        <rdfs:subClassOf rdf:resource="http://purl.org/mcobden/identity#↩
HTTPBehaviour"/>
        <owl:disjointWith rdf:resource="http://purl.org/mcobden/identity#↩
HTTPServerBehaviour"/>
        <rdfs:comment xml:lang="en">A facet of an agent which is responsible for↩
 making HTTP requests.</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#HTTPSServerBehaviour -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#HTTPSServerBehaviour↩
">
        <rdfs:label xml:lang="en">HTTPS Server Behaviour</rdfs:label>
        <rdfs:subClassOf rdf:resource="http://purl.org/mcobden/identity#↩
HTTPServerBehaviour"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="http://purl.org/mcobden/identity#↩
usesPublicKey"/>
                <owl:onClass rdf:resource="&wot;PubKey"/>
                <owl:minQualifiedCardinality rdf:datatype="&xsd;↩
nonNegativeInteger">1</owl:minQualifiedCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:comment xml:lang="en">A facet of an agent which responds to HTTP ↩
Requests using Transport Layer Security (TLS).</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#HTTPServerBehaviour -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#HTTPServerBehaviour">
        <rdfs:label xml:lang="en">HTTP Server Behaviour</rdfs:label>
        <rdfs:subClassOf rdf:resource="http://purl.org/mcobden/identity#↩
HTTPBehaviour"/>
```

```
        <rdfs:comment xml:lang="en">A facet of an agent which responds to HTTP ↩
Requests.</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#WebAgent -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#WebAgent">
        <rdfs:label xml:lang="en">Web Agent</rdfs:label>
        <rdfs:subClassOf rdf:resource="http://purl.org/mcobden/identity#Agent"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="http://purl.org/mcobden/identity#↩
hasBehaviour"/>
                <owl:someValuesFrom rdf:resource="http://purl.org/mcobden/↩
identity#HTTPBehaviour"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:comment xml:lang="en">An agent which interats with the web via ↩
HTTP.</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#WebClientAgent -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#WebClientAgent">
        <rdfs:label xml:lang="en">WebClient Agent</rdfs:label>
        <rdfs:subClassOf rdf:resource="http://purl.org/mcobden/identity#WebAgent↩
"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="http://purl.org/mcobden/identity#↩
hasBehaviour"/>
                <owl:someValuesFrom rdf:resource="http://purl.org/mcobden/↩
identity#HTTPClientBehaviour"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:comment xml:lang="en">An agent which makes HTTP requests to HTTP ↩
servers.</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#WebLocation -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#WebLocation">
        <rdfs:label xml:lang="en">Web Location</rdfs:label>
        <rdfs:comment xml:lang="en">A resource which describes a specific Web ↩
IRI, or matches defined group thereof.</rdfs:comment>
    </owl:Class>



    <!-- http://purl.org/mcobden/identity#WebServerAgent -->

    <owl:Class rdf:about="http://purl.org/mcobden/identity#WebServerAgent">
        <rdfs:label xml:lang="en">WebServer Agent</rdfs:label>
```

```
        <rdfs:subClassOf rdf:resource="http://purl.org/mcobden/identity#WebAgent↩
"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="http://purl.org/mcobden/identity#↩
hasBehaviour"/>
                <owl:someValuesFrom rdf:resource="http://purl.org/mcobden/↩
identity#HTTPServerBehaviour"/>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:comment xml:lang="en">An agent which understands and responds to ↩
HTTP requests.</rdfs:comment>
    </owl:Class>



    <!-- http://xmlns.com/foaf/0.1/Agent -->

    <rdf:Description rdf:about="&foaf;Agent"/>
</rdf:RDF>



<!-- Generated by the OWL API (version 3.3.1957) http://owlapi.sourceforge.net ↩
-->
```

Listing A.1: Our Web Server identity ontology

Bibliography

B Adida, M Birbeck, S McCarron, and S Pemberton. RDFa in XHTML: Syntax and Processing. Recommendation, W3C, October 2008. http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014. Latest version available at http://www.w3.org/TR/rdfa-syntax.

R Allbery. DNS SRV Resource Records for AFS. RFC 5864 (Proposed Standard), April 2010. URL: http://www.ietf.org/rfc/rfc5864.txt.

P Almquist. Type of Service in the Internet Protocol Suite. RFC 1349 (Proposed Standard), July 1992. URL: http://www.ietf.org/rfc/rfc1349.txt. Obsoleted by RFC 2474 (Nichols et al. [1998]).

M Andrews. Negative Caching of DNS Queries (DNS NCACHE). RFC 2308 (Proposed Standard), March 1998. URL: http://www.ietf.org/rfc/rfc2308.txt. Updated by RFCs 4035 (Arends et al. [2005c]), 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]).

P Archer, M Meimaris, and A Papantoniou. Registered Organization Vocabulary. Working group note, W3C, August 2013. http://www.w3.org/TR/2013/NOTE-vocab-regorg-20130801/. Latest version available at http://www.w3.org/TR/vocab-regorg/.

R Arends, R Austein, M Larson, D Massey, and S Rose. DNS Security Introduction and Requirements. RFC 4033 (Proposed Standard), March 2005a. URL: http://www.ietf.org/rfc/rfc4033.txt. Updated by RFC 6014 (Hoffman [2010]).

R Arends, R Austein, M Larson, D Massey, and S Rose. Resource Records for the DNS Security Extensions. RFC 4034 (Proposed Standard), March 2005b. URL: http://www.ietf.org/rfc/rfc4034.txt. Updated by RFCs 4470 (Weiler and Ihren [2006]), 6014 (Hoffman [2010]).

R Arends, R Austein, M Larson, D Massey, and S Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035 (Proposed Standard), March 2005c. URL: http://www.ietf.org/rfc/rfc4035.txt. Updated by RFCs 4470 (Weiler and Ihren [2006]), 6014 (Hoffman [2010]).

C Arthur, Rogue web certificate could have been used to attack Iran dissidents, August 2011. URL: http://www.guardian.co.uk/technology/2011/aug/30/faked-web-certificate-iran-dissidents.

D Artz and Y Gil. A survey of trust in computer science and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):58–71, 2007.

R Axelrod and W. D Hamilton. The evolution of cooperation. *Science*, 211(4489): 1390–1396, 1981.

D Beckett. RDF/XML Syntax Specification (Revised). Recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/. Latest version available at http://www.w3.org/TR/rdf-syntax-grammar/.

D Beckett and T Berners-Lee. Turtle - Terse RDF Triple Language, W3C Team Submission, 2008. URL: http://www.w3.org/TeamSubmission/turtle/.

J Berger, N Noorderhaven, and B Nooteboom. *Determinants of supplier dependence: An empirical study*, pages 195–212. Edward Elgar, 1995.

T Berners-Lee and D Connolly. Hypertext Markup Language - 2.0. RFC 1866 (Historic), November 1995. URL: http://www.ietf.org/rfc/rfc1866.txt. Obsoleted by RFC 2854 (Connolly and Masinter [2000]).

T Berners-Lee, R Fielding, and H Frystyk. Hypertext Transfer Protocol – HTTP/1.0. RFC 1945 (Informational), May 1996. URL: http://www.ietf.org/rfc/rfc1945.txt.

T Berners-Lee, R Fielding, and L Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (Standard), January 2005. URL: http://www.ietf.org/rfc/rfc3986.txt.

T Berners-Lee. Information management: A proposal, 1989. URL: http://www.w3.org/History/1989/proposal-msw.html.

T Berners-Lee. Linked Data - Design Issues, 2006a. URL: http://www.w3.org/DesignIssues/LinkedData.html.

T Berners-Lee. Notation 3: A readable language for data on the Web. Available online at http://www.w3.org/DesignIssues/Notation3.html, March 2006b.

T Berners-Lee, R Cailliau, J.-F Groff, and B Pollermann. World-Wide Web: The Information Universe. *Electronic Networking: Research, Applications and Policy*, 1(2): 74–82, 1992.

T Berners-Lee, J Hendler, and O Lassila. The Semantic Web. *Scientific American*, 284 (5):28–37, 2001.

D Berrueta and J Phipps. Best Practice Recipes for Publishing RDF Vocabularies. Technical report, 2008. URL: `http://www.w3.org/TR/2008/NOTE-swbp-vocab-pub-20080828/`. Latest version available at `http://www.w3.org/TR/swbp-vocab-pub/`.

T Bray, J Paoli, M Sperberg-McQueen, E Maler, and F Yergeau. Extensible Markup Language (XML) 1.0 (Fifth Edition). Recommendation, W3C, November 2008. `http://www.w3.org/TR/2008/REC-xml-20081126/`. Latest version available at `http://www.w3.org/TR/xml`.

D Brickley and R Guha. RDF Vocabulary Description Language 1.0: RDF Schema. Recommendation, W3C, February 2004. `http://www.w3.org/TR/2004/REC-rdf-schema-20040210/`. Latest version available at `http://www.w3.org/TR/rdf-schema/`.

B Briscoe. Tunnelling of Explicit Congestion Notification. RFC 6040 (Proposed Standard), November 2010. URL: `http://www.ietf.org/rfc/rfc6040.txt`.

British Broadcasting Corporation, Iran accused in 'dire' net security attack, March 2011. URL: `http://www.bbc.co.uk/news/technology-12847072`.

J Broekstra and A Kampman. Inferencing and Truth Maintenance in RDF Schema: Exploring a naive practical approach. In *Workshop on Practical and Scalable Semantic Systems (PSSS)*, 2003.

J Callas, L Donnerhacke, H Finney, D Shaw, and R Thayer. OpenPGP Message Format. RFC 4880 (Proposed Standard), November 2007. URL: `http://www.ietf.org/rfc/rfc4880.txt`. Updated by RFC 5581 (Shaw [2009]).

J. J Carroll. Signing RDF Graphs. In *The SemanticWeb - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 369–384. Springer Berlin / Heidelberg, 2003.

J. J Carroll, C Bizer, P Hayes, and P Stickler. Named graphs, provenance and trust. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2005. ACM.

D Connolly and L Masinter. The 'text/html' Media Type. RFC 2854 (Informational), June 2000. URL: `http://www.ietf.org/rfc/rfc2854.txt`.

D Connolly. A Pragmatic Theory of Reference for the Web. In *Proceedings of Identity, Reference, and the Web Workshop at the 15th WWW Conference*, 2006.

J Cook and T Wall. New work attitude measures of trust, organizational commitment and personal need non-fulfilment. *Journal of Occupational Psychology*, 53(1):39–52, March 1980.

D Crockford. The application/json Media Type for JavaScript Object Notation (JSON). RFC 4627 (Informational), July 2006. URL: http://www.ietf.org/rfc/rfc4627.txt.

S Currall and T Judge. Measuring trust between organizational boundary role persons. *Organizational behavior and human decision processes*, 64(2):151–170, 1995.

R Cyganiak, Top 100 most popular RDF namespace prefixes, February 2011. URL: http://richard.cyganiak.de/blog/2011/02/top-100-most-popular-rdf-namespace-prefixes/.

L Daigle. WHOIS Protocol Specification. RFC 3912 (Draft Standard), September 2004. URL: http://www.ietf.org/rfc/rfc3912.txt.

C Davis, P Vixie, T Goodwin, and I Dickinson. A Means for Expressing Location Information in the Domain Name System. RFC 1876 (Experimental), January 1996. URL: http://www.ietf.org/rfc/rfc1876.txt.

M Deutsch. *Cooperation and trust: Some theoretical notes.*, pages 275–320. University of Nebraska Press, Oxford, England, 1962.

M Deutsch. *The Resolution of Conflict*. Yale University Press, 1973.

R Dhamija, J. D Tygar, and M Hearst. Why phishing works. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590, New York, NY, USA, 2006. ACM.

J Doyle. A Truth Maintenance System. *Artificial Intelligence*, 12(3):231 – 272, 1979. ISSN 0004-3702.

D Eastlake 3rd. Domain Name System Security Extensions. RFC 2535 (Proposed Standard), March 1999. URL: http://www.ietf.org/rfc/rfc2535.txt. Obsoleted by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]), updated by RFCs 2931 (Eastlake 3rd [2000]), 3007 (Wellington [2000a]), 3008 (Wellington [2000b]), 3090 (Lewis [2001]), 3226 (Gudmundsson [2001]), 3445 (Massey and Rose [2002]), 3597 (Gustafsson [2003]), 3655 (Wellington and Gudmundsson [2003]), 3658 (Gudmundsson [2003]), 3755 (Weiler [2004]), 3757 (Kolkman et al. [2004]), 3845 (Schlyter [2004]).

D Eastlake 3rd. DNS Request and Transaction Signatures (SIG(0)s). RFC 2931 (Proposed Standard), September 2000. URL: http://www.ietf.org/rfc/rfc2931.txt.

D Eastlake 3rd. Domain Name System (DNS) Case Insensitivity Clarification. RFC 4343 (Proposed Standard), January 2006. URL: http://www.ietf.org/rfc/rfc4343.txt.

D Eastlake 3rd. Domain Name System (DNS) IANA Considerations. RFC 5395 (Best Current Practice), November 2008. URL: http://www.ietf.org/rfc/rfc5395.txt.

D Eastlake 3rd and C Kaufman. Domain Name System Security Extensions. RFC 2065 (Proposed Standard), January 1997. URL: `http://www.ietf.org/rfc/rfc2065.txt`. Obsoleted by RFC 2535 (Eastlake 3rd [1999]).

R Elz and R Bush. Serial Number Arithmetic. RFC 1982 (Proposed Standard), August 1996. URL: `http://www.ietf.org/rfc/rfc1982.txt`.

R Elz and R Bush. Clarifications to the DNS Specification. RFC 2181 (Proposed Standard), July 1997. URL: `http://www.ietf.org/rfc/rfc2181.txt`. Updated by RFCs 4035 (Arends et al. [2005c]), 2535 (Eastlake 3rd [1999]), 4343 (Eastlake 3rd [2006]), 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 5452 (Hubert and van Mook [2009]).

E Erikson. *Identity: Youth & Crisis*. Austen Riggs Monograph. W.W. Norton & Co., 1994.

C Everhart, L Mamakos, R Ullmann, and P Mockapetris. New DNS RR Definitions. RFC 1183 (Experimental), October 1990. URL: `http://www.ietf.org/rfc/rfc1183.txt`. Updated by RFCs 5395 (Eastlake 3rd [2008]), 5864 (Allbery [2010]).

D Fensel and F van Harmelen. Unifying Reasoning and Search to Web Scale. *Internet Computing, IEEE*, 11(2):96 –95, march-april 2007. ISSN 1089-7801.

R Fielding, J Gettys, J Mogul, H Frystyk, and T Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2068 (Proposed Standard), January 1997. URL: `http://www.ietf.org/rfc/rfc2068.txt`. Obsoleted by RFC 2616 (Fielding et al. [1999]).

R Fielding, J Gettys, J Mogul, H Frystyk, L Masinter, P Leach, and T Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. URL: `http://www.ietf.org/rfc/rfc2616.txt`. Updated by RFCs 2817 (Khare and Lawrence [2000]), 5785 (Nottingham and Hammer-Lahav [2010]).

T Finin and A Joshi. Agents, trust, and information access on the semantic web. *SIGMOD Rec.*, 31:30–35, December 2002.

F Fukuyama. Trust: The Social Virtues and the Creation of Prosperity. 1995.

Y Gil and V Ratnakar. Trusting Information Sources One Citizen at a Time. volume 2342/2002, pages 162–176, 2002.

J Golbeck. Trust on the World Wide Web: a survey. *Foundations and Trends Web Science*, 1(2):131–197, 2006.

J Golbeck and J Hendler. Inferring binary trust relationships in Web-based social networks. *ACM Transactions on Internet Technology (TOIT)*, Jan 2006.

J Golbeck, B Parsia, and J Hendler. Trust Networks on the Semantic Web. In M Klusch, A Omicini, S Ossowski, and H Laamanen, editors, *Cooperative Information Agents VII*, volume 2782 of *Lecture Notes in Computer Science*, pages 238–249. Springer Berlin / Heidelberg, 2003.

J. A Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, College Park, MD, USA, 2005.

T Grandison and M Sloman. A Survey of Trust in Internet Applications. *IEEE Communications Surveys and Tutorials*, 3(4):2–16, 2000.

D Grossman. New Terminology and Clarifications for Diffserv. RFC 3260 (Informational), April 2002. URL: `http://www.ietf.org/rfc/rfc3260.txt`.

O Gudmundsson. DNSSEC and IPv6 A6 aware server/resolver message size requirements. RFC 3226 (Proposed Standard), December 2001. URL: `http://www.ietf.org/rfc/rfc3226.txt`. Updated by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]).

O Gudmundsson. Delegation Signer (DS) Resource Record (RR). RFC 3658 (Proposed Standard), December 2003. URL: `http://www.ietf.org/rfc/rfc3658.txt`. Obsoleted by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]), updated by RFC 3755 (Weiler [2004]).

Y Guo, Z Pan, and J Heflin. LUBM: A benchmark for OWL knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2 - 3):158 – 182, 2005. ISSN 1570-8268.

A Gustafsson. Handling of Unknown DNS Resource Record (RR) Types. RFC 3597 (Proposed Standard), September 2003. URL: `http://www.ietf.org/rfc/rfc3597.txt`. Updated by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]), 5395 (Eastlake 3rd [2008]).

J Hakala and H Walravens. Using International Standard Book Numbers as Uniform Resource Names. RFC 3187 (Informational), October 2001. URL: `http://www.ietf.org/rfc/rfc3187.txt`.

F Halasz and M Schwartz. The Dexter hypertext reference model. *Communications of the ACM*, 37(2):30–39, 1994.

F. G Halasz, T. P Moran, and R. H Trigg. Notecards in a nutshell. In *CHI+GI 1987: Proceedings of the SIGCHI/GI Conference on Human Factors in Computing systems and Graphics Interface*, pages 45–52, Toronto, Ontario, Canada, 1987. ACM.

H Halpin. Provenance: The Missing Component of the Semantic Web. In *Proceedings of the First Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2009)*, 2009.

H Halpin and V Presutti. An Ontology of Resources: Solving the Identity Crisis. In L Aroyo, P Traverso, F Ciravegna, P Cimiano, T Heath, E Hyvönen, R Mizoguchi, E Oren, M Sabou, and E Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 521–534. Springer Berlin / Heidelberg, 2009.

H Halpin, P Hayes, J McCusker, D McGuinness, and H Thompson. When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data. In P Patel-Schneider, Y Pan, P Hitzler, P Mika, L Zhang, J Pan, I Horrocks, and B Glimm, editors, *The Semantic Web – ISWC 2010*, volume 6496 of *Lecture Notes in Computer Science*, pages 305–320. Springer Berlin / Heidelberg, 2010.

D Hamilton, S Sherman, and B Lickel. Perceiving Social Groups: The Importance of the Entitativity. *Intergroup cognition and intergroup behavior*, page 47, 1997.

S Harris and N Gibbins. 3store: Efficient Bulk RDF Storage. In R Volz, S Decker, and I. F Cruz, editors, *PSSS1 - Practical and Scalable Semantic Systems, Proceedings of the First International Workshop on Practical and Scalable Semantic Systems, Sanibel Island, Florida, USA, October 20, 2003*, volume 89 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

S Harris and A Seaborne. SPARQL 1.1 Query Language. Recommendation, W3C, March 2013. `http://www.w3.org/TR/2013/REC-sparql11-query-20130321/`. Latest version available at `http://www.w3.org/TR/sparql11-query/`.

O Hartig. Querying Trust in RDF Data with tSPARQL. In L Aroyo, P Traverso, F Ciravegna, P Cimiano, T Heath, E Hyvönen, R Mizoguchi, E Oren, M Sabou, and E Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 5–20. Springer Berlin / Heidelberg, 2009a.

O Hartig. Provenance Information in the Web of Data. In C Bizer, T Heath, T Berners-Lee, and K Idehen, editors, *Proceedings of the WWW2009 Workshop on Linked Data on the Web, Madrid, Spain, April 20, 2009*, volume 538 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2009b.

O Hartig. Towards a Data-Centric Notion of Trust in the Semantic Web. In P Kärger, D Olmedilla, A Passant, and A Polleres, editors, *Proceedings of the Second Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2010)*, volume 576 of *CEUR Workshop Proceedings*. CEUR-WS.org, May 2010.

O Hartig and J Zhao. Publishing and Consuming Provenance Metadata on the Web of Linked Data. In D McGuinness, J Michaelis, and L Moreau, editors, *Provenance and Annotation of Data and Processes*, volume 6378 of *Lecture Notes in Computer Science*, pages 78–90. Springer Berlin / Heidelberg, 2010. 10.1007/978-3-642-17819-1_10.

P Hayes. RDF Semantics. Recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/. Latest version available at http://www.w3.org/TR/rdf-mt/.

T Heath and C Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 1st edition, 2011.

P Hoffman. Cryptographic Algorithm Identifier Allocation for DNSSEC. RFC 6014 (Proposed Standard), November 2010. URL: http://www.ietf.org/rfc/rfc6014.txt.

A Hubert and R van Mook. Measures for Making DNS More Resilient against Forged Answers. RFC 5452 (Proposed Standard), January 2009. URL: http://www.ietf.org/rfc/rfc5452.txt.

T. D Huynh, N. R Jennings, and N. R Shadbolt. Certified reputation: how an agent can trust a stranger. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, AAMAS '06, pages 1217–1224, New York, NY, USA, 2006. ACM.

A Jaffri, H Glaser, and I Millard. URI Identity Management for Semantic Web Data Integration and Linkage. In *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*, volume 4806 of *Lecture Notes in Computer Science*, pages 1125–1134. Springer Berlin / Heidelberg, 2007.

L Kagal, T Finin, and A Joshi. A Policy Based Approach to Security for the Semantic Web. In *The SemanticWeb - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 402–418. Springer Berlin / Heidelberg, 2003.

Y Kalfoglou, H Alani, M Schorlemmer, and C Walton. On the Emergent Semantic Web and Overlooked Issues. In S McIlraith, D Plexousakis, and F van Harmelen, editors, *The Semantic Web – ISWC 2004*, volume 3298 of *Lecture Notes in Computer Science*, pages 576–590. Springer Berlin / Heidelberg, 2004.

S. D Kamvar, M. T Schlosser, and H Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

M. L Katz and C Shapiro. Systems Competition and Network Effects. *Journal of Economic Perspectives*, 8(2):93–115, 1994.

S Kent and K Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), December 2005. URL: http://www.ietf.org/rfc/rfc4301.txt. Updated by RFC 6040 (Briscoe [2010]).

R Khare and S Lawrence. Upgrading to TLS Within HTTP/1.1. RFC 2817 (Proposed Standard), May 2000. URL: `http://www.ietf.org/rfc/rfc2817.txt`.

J Köbler, U Schc6ning, and J Torán. *The Graph Isomorphism Problem: Its Structural Complexity*. Progress in Theoretical Computer Science Series. Birkhäuser, 1993.

J Koch, C. A Velasco, and P Ackermann. HTTP Vocabulary in RDF 1.0. Working draft, W3C, May 2011. `http://www.w3.org/TR/2011/WD-HTTP-in-RDF10-20110510/`. Latest version available at `http://www.w3.org/TR/HTTP-in-RDF10/`.

O Kolkman, J Schlyter, and E Lewis. Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag. RFC 3757 (Proposed Standard), April 2004. URL: `http://www.ietf.org/rfc/rfc3757.txt`. Obsoleted by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]).

S Konstantopoulos and P Archer. Protocol for Web Description Resources (POWDER): Formal Semantics. Recommendation, W3C, September 2009. `http://www.w3.org/TR/2009/REC-powder-formal-20090901/`. Latest version available at `http://www.w3.org/TR/powder-formal/`.

M Koster. A Method for Web Robots Control. *Network Working Group, Internet Draft*, 1996.

J Krebs and N Davies. *Behavioural ecology: an evolutionary approach*. Sinauer Associates, 1978.

T Lebo, S Sahoo, and D McGuinness. PROV-O: The PROV Ontology. Recommendation, W3C, April 2013. `http://www.w3.org/TR/2013/REC-prov-o-20130430/`. Latest version available at `http://www.w3.org/TR/prov-o/`.

E Lewis. DNS Security Extension Clarification on Zone Status. RFC 3090 (Proposed Standard), March 2001. URL: `http://www.ietf.org/rfc/rfc3090.txt`. Obsoleted by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]), updated by RFC 3658 (Gudmundsson [2003]).

E Lewis. The Role of Wildcards in the Domain Name System. RFC 4592 (Proposed Standard), July 2006. URL: `http://www.ietf.org/rfc/rfc4592.txt`.

E Lewis and A Hoenes. DNS Zone Transfer Protocol (AXFR). RFC 5936 (Proposed Standard), June 2010. URL: `http://www.ietf.org/rfc/rfc5936.txt`.

R Lewis. Dereferencing HTTP URIs. Technical report, World Wide Web Consortium, May 2007. URL: `http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14`.

B Manning. DNS NSAP RRs. RFC 1348 (Experimental), July 1992. URL: `http://www.ietf.org/rfc/rfc1348.txt`. Obsoleted by RFC 1637 (Manning and Colella [1994a]).

B Manning and R Colella. DNS NSAP Resource Records. RFC 1637 (Experimental), June 1994a. URL: http://www.ietf.org/rfc/rfc1637.txt. Obsoleted by RFC 1706 (Manning and Colella [1994b]).

B Manning and R Colella. DNS NSAP Resource Records. RFC 1706 (Informational), October 1994b. URL: http://www.ietf.org/rfc/rfc1706.txt.

F Manola and E Miller. RDF Primer. Recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/. Latest version available at http://www.w3.org/TR/rdf-primer/.

S Marsh and M Dibben. Trust, Untrust, Distrust and Mistrust – An Exploration of the Dark(er) Side. In P Herrmann, V Issarny, and S Shiu, editors, *Trust Management*, volume 3477 of *Lecture Notes in Computer Science*, pages 17–33. Springer Berlin / Heidelberg, 2005.

S. P Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, April 1994. URL: http://homepage.mac.com/smarsh2003/SteveMarsh/Publications_files/Trust-thesis.pdf.

D Massey and S Rose. Limiting the Scope of the KEY Resource Record (RR). RFC 3445 (Proposed Standard), December 2002. URL: http://www.ietf.org/rfc/rfc3445.txt. Obsoleted by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]).

S McCarron and M Ishikawa. XHTML 1.1 - Module-based XHTML - Second Edition. Recommendation, W3C, November 2010. http://www.w3.org/TR/2010/REC-xhtml11-20101123. Latest version available at http://www.w3.org/TR/xhtml11/.

D McGuinness and F van Harmelen. OWL Web Ontology Language Overview. Recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/. Latest version available at http://www.w3.org/TR/owl-features/.

D. H McKnight and N. L Chervany. The Meanings of Trust. Working Paper, 1996. URL: http://www.misrc.umn.edu/workingpapers/fullPapers/1996/9604_040100.pdf.

D. H McKnight, V Choudhury, and C. J Kacmar. Developing and Validating Trust Measures for e-Commerce: An Integrative Typology. *Information Systems Research*, 13(3):334–359, 2002.

P Mockapetris. Domain names - concepts and facilities. RFC 1034 (Standard), November 1987. URL: http://www.ietf.org/rfc/rfc1034.txt. Updated by RFCs 1101 (Mockapetris [1989]), 1183 (Everhart et al. [1990]), 1348 (Manning [1992]), 1876 (Davis et al. [1996]), 1982 (Elz and Bush [1996]), 2065 (Eastlake 3rd and Kaufman [1997]),

2181 (Elz and Bush [1997]), 2308 (Andrews [1998]), 2535 (Eastlake 3rd [1999]), 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]), 4343 (Eastlake 3rd [2006]), 4035 (Arends et al. [2005c]), 4592 (Lewis [2006]), 5936 (Lewis and Hoenes [2010]).

P Mockapetris. DNS encoding of network names and other types. RFC 1101, April 1989. URL: `http://www.ietf.org/rfc/rfc1101.txt`.

T Moore and R Clayton. Evil Searching: Compromise and Recompromise of Internet Hosts for Phishing. In R Dingledine and P Golle, editors, *Financial Cryptography and Data Security*, volume 5628 of *Lecture Notes in Computer Science*, pages 256–272. Springer Berlin Heidelberg, 2009.

L Moreau and P Missier. PROV-DM: The PROV Data Model. Recommendation, W3C, April 2013. `http://www.w3.org/TR/2013/REC-prov-dm-20130430/`. Latest version available at `http://www.w3.org/TR/prov-dm/`.

L Moreau, B Clifford, J Freire, J Futrelle, Y Gil, P Groth, N Kwasnikowska, S Miles, P Missier, J Myers, B Plale, Y Simmhan, E Stephan, and J. V den Bussche. The Open Provenance Model core specification (v1.1). *Future Generation Computer Systems*, July 2010.

B Motik, B. C Grau, I Horrocks, Z Wu, and A Fokoue. OWL 2 Web Ontology Language Profiles. Recommendation, W3C, October 2009a. `http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/`. Latest version available at `http://www.w3.org/TR/owl2-profiles/`.

B Motik, B Parsia, and P Patel-Schneider. OWL 2 Web Ontology Language XML Serialization. Recommendation, W3C, October 2009b. `http://www.w3.org/TR/2009/REC-owl2-xml-serialization-20091027/`. Latest version available at `http://www.w3.org/TR/owl2-xml-serialization/`.

F Naumann. Information Quality Criteria. In *Quality-Driven Query Answering for Integrated Information Systems*, volume 2261 of *Lecture Notes in Computer Science*, pages 269–289. Springer Berlin / Heidelberg, 2002.

W Nejdl, D Olmedilla, and M Winslett. PeerTrust: Automated Trust Negotiation for Peers on the Semantic Web. In W Jonker and M Petković, editors, *Secure Data Management*, volume 3178 of *Lecture Notes in Computer Science*, pages 118–132. Springer Berlin / Heidelberg, 2004.

K Nichols, S Blake, F Baker, and D Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474 (Proposed Standard), December 1998. URL: `http://www.ietf.org/rfc/rfc2474.txt`. Updated by RFCs 3168 (Ramakrishnan et al. [2001]), 3260 (Grossman [2002]).

M Nottingham. Web Linking. RFC 5988 (Proposed Standard), October 2010. URL: http://www.ietf.org/rfc/rfc5988.txt.

M Nottingham and E Hammer-Lahav. Defining Well-Known Uniform Resource Identifiers (URIs). RFC 5785 (Proposed Standard), April 2010. URL: http://www.ietf.org/rfc/rfc5785.txt.

K O'Hara. *Trust: From Socrates to Spin*. Icon Books, 2004.

K O'Hara and W Hall. Trust on the Web: Some Web Science Research Challenges. *UoC Papers: E-Journal on the Knowledge Society*, (7), Oct 2008.

K O'Hara, H Alani, Y Kalfoglou, and N Shadbolt. Trust Strategies for the Semantic Web. *Proceedings of Workshop on Trust*, Jan 2004.

D Olmedilla, O. F Rana, B Matthews, and W Nejdl. Security and Trust Issues in Semantic Grids. In *Semantic Grid: The Convergence of Technologies*, volume 05271 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum (IBFI), Schloss Dagstuhl, Germany, July 2005.

ORDI-Triplesets, ORDI SG Tripleset Model. URL: http://www.ontotext.com/ordi/tripleset-model.

Oxford English Dictionary, "trust, noun.", September 2011. URL: http://www.oed.com/view/Entry/207004.

J Postel. Internet Protocol. RFC 791 (Standard), September 1981. URL: http://www.ietf.org/rfc/rfc791.txt. Updated by RFC 1349 (Almquist [1992]).

E Prud'hommeaux and A Seaborne. SPARQL Query Language for RDF. Recommendation, W3C, January 2008. http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/. Latest version available at http://www.w3.org/TR/rdf-sparql-query/.

Y Raimond, S Abdallah, M Sandler, and F Giasson. The music ontology. In *Proceedings of the International Conference on Music Information Retrieval*, pages 417–422. Citeseer, 2007.

K Ramakrishnan, S Floyd, and D Black. The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168 (Proposed Standard), September 2001. URL: http://www.ietf.org/rfc/rfc3168.txt. Updated by RFCs 4301 (Kent and Seo [2005]), 6040 (Briscoe [2010]).

I.-T REC. X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks. Technical report, International Telecommunication Union, March 2008.

E Rescorla. HTTP Over TLS. RFC 2818 (Informational), May 2000. URL: `http://www.ietf.org/rfc/rfc2818.txt`. Updated by RFC 5785 (Nottingham and Hammer-Lahav [2010]).

P Resnick and R Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. *The Economics of the Internet and E-Commerce*, Jan 2002.

M Richardson, R Agrawal, and P Domingos. Trust Management for the Semantic Web. In D Fensel, K Sycara, and J Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 351–368. Springer Berlin / Heidelberg, 2003.

S. J Russell and P Norvig. *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall, 2003.

M Salvadores, G Correndo, M Szomszor, Y Yang, N Gibbins, I Millard, H Glaser, and N Shadbolt. Domain-Specific Backlinking Services in the Web of Data. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 318 –323, 31 2010-sept. 3 2010.

J Schlyter. DNS Security (DNSSEC) NextSECure (NSEC) RDATA Format. RFC 3845 (Proposed Standard), August 2004. URL: `http://www.ietf.org/rfc/rfc3845.txt`. Obsoleted by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]).

S. P Shapiro. The Social Control of Impersonal Trust. *American Journal of Sociology*, 93(3):pp. 623–658, 1987. ISSN 00029602.

D Shaw. The Camellia Cipher in OpenPGP. RFC 5581 (Informational), June 2009. URL: `http://www.ietf.org/rfc/rfc5581.txt`.

Six Apart, TrackBack Technical Specification, August 2002. URL: `http://www.sixapart.com/pronet/docs/trackback_spec`.

K. J Stewart. Trust transfer on the World Wide Web. *Organization Science*, pages 5–17, 2003.

P Stickler, Definition of the Web Architecture Vocabulary v1.4. URL: `http://sw.nokia.com/schemas/general/WebArch-1.4.owl`.

P Stickler, URIQA: The URI Query Agent Model, 2003. URL: `http://sw.nokia.com/uriqa/URIQA.html`.

P Stickler, CBD - Concise Bounded Description, June 2005. URL: `http://www.w3.org/Submission/CBD/`. W3C Member Submission.

F. R Stockton. The Lady or the Tiger? *The Century*, 25(1):83–86, November 1884.

H Story, B Harbulot, I Jacobi, and M Jones. FOAF+ SSL: RESTful Authentication for the Social Web. In *Proceedings of the First Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2009)*, 2009.

W. T. L Teacy, N. R Jennings, A Rogers, and M Luck. A Hierarchical Bayesian Trust Model based on Reputation and Group Behaviour. In *6th European Workshop on Multi-Agent Systems (EUMAS 2008)*, December 2008.

S Tramp, P Frischmuth, T Ermilov, and S Auer. Weaving a Social Data Web with Semantic Pingback. In P Cimiano and H Pinto, editors, *Knowledge Engineering and Management by the Masses*, volume 6317 of *Lecture Notes in Computer Science*, pages 135–149. Springer Berlin / Heidelberg, 2010.

Y. D Wang and H. H Emurian. An overview of online trust: Concepts, elements, and implications. *Computers in Human Behavior*, Jan 2005.

E. R Watkins and D. A Nicole. Named Graphs as a Mechanism for Reasoning About Provenance. In X Zhou, J Li, H Shen, M Kitsuregawa, and Y Zhang, editors, *Frontiers of WWW Research and Development - APWeb 2006*, volume 3841 of *Lecture Notes in Computer Science*, pages 943–948. Springer Berlin Heidelberg, 2006.

S Weiler. Legacy Resolver Compatibility for Delegation Signer (DS). RFC 3755 (Proposed Standard), May 2004. URL: `http://www.ietf.org/rfc/rfc3755.txt`. Obsoleted by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]), updated by RFCs 3757 (Kolkman et al. [2004]), 3845 (Schlyter [2004]).

S Weiler and J Ihren. Minimally Covering NSEC Records and DNSSEC On-line Signing. RFC 4470 (Proposed Standard), April 2006. URL: `http://www.ietf.org/rfc/rfc4470.txt`.

B Wellington. Secure Domain Name System (DNS) Dynamic Update. RFC 3007 (Proposed Standard), November 2000a. URL: `http://www.ietf.org/rfc/rfc3007.txt`. Updated by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]).

B Wellington. Domain Name System Security (DNSSEC) Signing Authority. RFC 3008 (Proposed Standard), November 2000b. URL: `http://www.ietf.org/rfc/rfc3008.txt`. Obsoleted by RFCs 4035 (Arends et al. [2005c]), 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), updated by RFC 3658 (Gudmundsson [2003]).

B Wellington and O Gudmundsson. Redefinition of DNS Authenticated Data (AD) bit. RFC 3655 (Proposed Standard), November 2003. URL: `http://www.ietf.org/rfc/rfc3655.txt`. Obsoleted by RFCs 4033 (Arends et al. [2005a]), 4034 (Arends et al. [2005b]), 4035 (Arends et al. [2005c]).

M Winslett, T Yu, K. E Seamons, A Hess, J Jacobson, R Jarvis, B Smith, and L Yu. Negotiating trust in the Web. *Internet Computing, IEEE*, 6(6):30–37, nov/dec 2002.

M. J Wooldridge and N. R Jennings. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.